DEMOCRITUS UNIVERSITY OF THRACE
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SOFTWARE AND APPLICATION DEVELOPMENT SECTOR

EXTENDING INTERNET ACCESS AND ENHANCING MOBILE DATA OFFLOADING THROUGH PEER - AND VEHICLE- ASSISTED COMMUNICATIONS

IOANNIS KOMNIOS

PhD thesis

Περίληψη διατριβής

Στην παρούσα διδακτορική διατριβή μετράμε το πρόβλημα της πρόσφατης έκρηξης του όγκου δεδομένων που μεταφέρονται μέσα από τα δίκτυα κινητής τηλεφωνίας, καθώς επίσης και την αυξανόμενη δυσαναλογία ανάμεσα στα άτομα που έχουν πρόσβαση στο Διαδίκτυο και τα άτομα που δεν έχουν τη δυνατότητα να συνδεθούν στις υπηρεσίες του Διαδικτύου. Για την αντιμετώπιση των παραπάνω προκλήσεων, στο πλαίσιο της παρούσας διατριβής προτείνουμε λύσεις για την εκφόρτωση δεδομένων από τα δίκτυα κινητής τηλεφωνίας και την επέκταση της υπάρχουσας διαδικτυακής κάλυψης. Συγκεκριμένα, αξιοποιούμε της αρχές των ανεκτικών δικτύων σε καθυστερήσεις και διακοπές (Delay/Disruption Tolerant Networks), από τη στιγμή που πολλές εφαρμογές για κινητές συσκευές είναι ανεκτικές σε καθυστερήσεις, όπως ο συγχρονισμός δεδομένων που αποθηκεύονται στο υπολογιστικό νέφος, και λειτουργούν αποδοτικά σε περίπτωση καθυστερήσεων χωρίς να επηρεάζουν την εμπειρία χρήσης.

Αρχικά, επικεντρωνόμαστε σε ασύρματα κοινοτικά δίκτυα (wireless community networks) σε αναπτυσσόμενες χώρες που συχνά βασίζονται σε συνδέσεις χαμηλού εύρους ζώνης τα οποία μοιράζονται από ένα μεγάλο αριθμό χρηστών, οδηγώντας τις συνδέσεις στη συνθήκη του υπο-πακέτου, όπου η απόδοση ανά ροή είναι μικρότερη από ένα πακέτο ανά RTT. Έχει αποδειχθεί ότι το Transmission Control Protocol (TCP) δεν λειτουργεί αποδοτικά σε τέτοιες συνθήκες, οδηγώντας σε μη δικαιοσύνη, μεγάλο ποσοστό απώλειας πακέτων και συνεχόμενες λήξεις του χρονικού περιθωρίου (timeout). Ως εναλλακτική στο TCP, μελετάμε την απόδοση μιας μεθόδου πρόσβασης less-than-best-effort (LBE), την Low Extra Delay Background Transport (LEDBAT) και τη παραλλαγή της που εγγυάται δικαιοσύνη ανάμεσα στις ροές fLEDBAT, στη συνθήκη του υπο-πακέτου για συνδέσεις που μοιράζονται από πολλούς χρήστες. **Σκοπός μας είναι να εξερευνήσουμε τη δυνατότητα χρήσης μιας μεθόδου πρόσβασης LBE, που έχει σχεδιαστεί ώστε να είναι λιγότερο επιθετική από το TCP, στη θέση του TCP για την μετάδοση δεδομένων σε συνδέσεις με περιορισμένο εύρος ζώνης.**

Στη συνέχεια στοχεύουμε σε αστικά περιβάλλοντα και αναζητούμε λύσεις ώστε να αυξήσουμε τον όγκο των δεδομένων που εκφορτώνονται από τα δίκτυα κινητής τηλεφωνίας μέσω σημείων πρόσβασης WiFi. Σήμερα οι χρήστες εφαρμόζουν είτε επί τόπου εκφόρτωση δεδομένων (on-the-spot offloading) όταν ένα σημείο πρόσβασης WiFi είναι άμεσα διαθέσιμο, είτε ανεκτική στην καθυστέρηση εκφόρτωση δεδομένων (delay-tolerant offloading) όταν η μετάδοση καθυστερεί για κάποιο χρονικό περιθώριο σε περίπτωση που κάποιος χρήστης έχει την ευκαιρία να εκφορτώσει δεδομένα μέσω WiFi αργότερα. Πολλές εφαρμογές, όπως το ηλεκτρονικό ταχυδρομείο και η μεταφορά αρχείων, μπορούν να καθυστερήσουν τη μετάδοση δεδομένων χωρίς να επηρεάσουν την εμπερία χρήσης. Παρόλα αυτά, ο περιορισμένος αριθμός ανοιχτών σημείων πρόσβασης WiFi περιορίζει τα οφέλη των υπαρχόντων μεθόδων εκφόρτωσης δεδομένων από δίκτυα κινητής τηλεφωνίας. Ως λύση προτείνουμε τον μηχανισμό Cost-Effective Multi-Mode Offloading (CEMMO) που ενισχύει την επί τόπου εκφόρτωση δεδομένων και την ανεκτική στην καθυστέρηση εκφόρτωση δεδομένων με μια μέθοδο εκφόρτωσης δεδομένων πολλαπλών βημάτων μέσω χρηστών. Όλες οι μέθοδοι εκφόρτωσης δεδομένων μέσω χρηστών ως τώρα βασίζονται στην αρχή των ενδιαφερόντων και των συνδρομών σε δημοφιλές περιεχόμενο. Η νέα μέθοδος εκφόρτωσης δεδομένων μέσω χρηστών που προτείνουμε είναι η πρώτη λύση του είδους που επιτρέπει την εκφόρτωση δεδομένων από τις συνδέσεις ανεξάρτητα από το περιεχόμενο και τη δημοφιλία. **Σκοπός μας είναι να αναπτύξουμε έναν μηχανισμό εκφόρτωσης δεδομένων που ενισχύει τη συνολική δυνατότητα εκφόρτωσης δεδομένων και αυξάνει το συνολικό όγκο δεδομένων που εκφορτώνονται από τα δίκτυα κινητής τηλεφωνίας.**

Στο επόμενο βήμα, επικεντρωνόμαστε στην παροχή δωρεάν πρόσβασης στο Διαδίκτυο, η οποία είναι ανεκτική στην καθυστέρηση, σε όσους βρίσκονται αποκομμένοι από τον σημερινό ψηφιακό κόσμο. Η εγκατάσταση ανοιχτών σημείων πρόσβασης WiFi σε μεγάλη κλίμακα έχει αποδειχθεί μη εφαρμόσιμη λόγω του τεράστιου οικονομικού κόστους που συνεπάγεται. Τα δίκτυα που παρέχονται από τους χρήστες (user-provided networks) έχουν προταθεί ως εναλλακτική, ωστόσο τέτοιες προσεγγίσεις παρέχουν πρόσβαση στο Διαδίκτυο μόνο σε συγκεκριμένες περιοχές χωρίς να παρέχουν εκτεταμένη κάλυψη. Ως λύση, σχεδιάζουμε μια δικτυακή αρχιτεκτονική, ο κορμός της οποίας αποτελείται από κόμβους ανεκτικούς στην καθυστέρηση που τοποθετούνται τόσο σε μέσα μαζικής μεταφοράς, όπως λεωφορεία και τραμς, όσο στις αντίστοιχες στάσεις και λειτουργούν ως μεταφορείς δεδομένων και πύλες δικτύων, αντίστοιχα. Για να επιτύχουμε μέγιστο κέρδος, αναπτύσσουμε το πρωτόκολλο δρομολόγησης γράφου συνδεσιμότητας (Connectivity Plan Routing Protocol), ένα DTN πρωτόκολλο δρομολόγησης που αξιοποιεί το γράφο συνδεσιμότητας των δικτύων μαζικής μεταφοράς και επιλέγει τη βέλτιστη διαδρομή που εκτιμάται πως θα παρέχει το γρηγορότερο χρόνο παράδοσης ανάμεσα στον τελικό χρήστη και μια συνδεδεμένη πύλη δικτύου. Στη συνέχεια προτείνουμε το ενισχυμένο πρωτόκολλο δρομολόγησης γράφου συνδεσιμότητας (Enhanced Connectivity Plan Routing Protocol), το οποίο αξιοποιεί όχι μόνο την εκ των προτέρων γνώση συνδέσεων ανάμεσα σε πύλες δικτύων και μεταφορείς δεδομένων, αλλά αξιοποιεί ακόμη και ευκαιριακές επαφές ανάμεσα σε μεταφορείς δεδομένων για να υπολογίσει διαδρομές, ενώ ταυτόχρονα είναι ανεκτικό σε καθυστερήσεις που δεν μπορούν να προβλεφθούν, όπως για παράδειγμα λόγω οδικών ατυχημάτων. **Σκοπός μας είναι να αναπτύξουμε μια ολοκληρωμένη δικτυακή αρχιτεκτονική που αξιοποιεί τα μέσα μαζικής μεταφοράς σε αστικά περιβάλλοντα για να επεκτείνει την υπάρχουσα συνδεσιμότητα στο Διαδίκτυο.**

Στο τελευταίο μέρος της παρούσας διατριβής, επικεντρωνόμαστε στο σχεδιασμό μιας αρχιτεκτονικής για το Μελλοντικό Διαδίκτυο (Future Internet) η οποία συνδυάζει την ικανότητα των Δικτύων που βασίζονται στην Πληροφορία (Information Centric Networks) να «σπρώχνουν» δεδομένα στα άκρα του δικτύου ώστε να παρέχουν τοπική πρόσβαση σε σημαντικό περιεχόμενο, με μια στρατηγική μετάδοσης μόνο όταν χρειάζεται που παρέχεται από τα δίκτυα ανεκτικά στις καθυστερήσεις και διακοπές. Μια τέτοια αρχιτεκτονική, όχι μόνο ωθεί την καινοτομία σε ένα μεγάλο εύρος νέων υπηρεσιών και εφαρμογών, αλλά συμπεριλαμβάνει και τις υπάρχουσες επιτυχημένες υπηρεσίες του Διαδικτύου. **Σκοπός μας είναι να χτίσουμε μια αρχιτεκτονική Διαδικτύου η οποία παρέχει καθολική κάλυψη, αξιοποιώντας όλες τις επιλογές συνδεσιμότητας, και επιτρέπει μεγαλύτερη ενσωμάτωση των επίγειων και των δορυφορικών επικοινωνιών.** Ως ένα πρώτο βήμα προς την εφαρμογή της αρχιτεκτονικής αναπτύσσουμε το SPICE testbed, ένα DTN testbed που αξιοποιεί την τελευταία λέξη της τεχνολογίας και βρίσκεται στο Κέντρο Διαστημικής Διαδικτύωσης στην Ξάνθη. Το SPICE DTN testbed είναι ένα πειραματικό ερευνητικό περιβάλλον το οποίο έχει σχεδιαστεί συγκεκριμένα για επίγειες, δορυφορικές και διαστημικές επικοινωνίες και περιλαμβάνει εκτός από πολλούς κόμβους DTN, εξειδικευμένα εξαρτήματα τα οποία εξομοιώνουν με ακρίβεια τη λειτουργεία τυπικών σταθμών βάσης, διαστημικών συνδέσεων και δορυφόρων.

To my father,

who would be so proud

to hold this thesis in his hands

Acknowledgments

With the completion of the manuscript of the present PhD thesis, I would like to thank all these people who stood beside me during this journey and contributed, each in his/her own way, to its realisation.

First of all, I would like to sincerely thank my supervisor *Prof. Vassilis Tsaoussidis* for the help, guidance and plenty of opportunities he has given me throughout the last 8 years that we have been collaborating. Prof. Tsaoussidis has never limited me into strict frameworks; instead he has granted me the freedom to identify various solutions to the problems addressed in the present thesis. Moreover, he has motivated me to identify skills and capabilities that I would have never acknowledged myself and I will forever be grateful for that.

I would also like to thank *Prof. Katos* and *Prof. Antonakopoulos*, the members of the advisory committee, for the useful feedback they have provided me regarding the present thesis, as well as all *members of the examining committee* for the precious time they devoted to the assessment of this thesis.

The road to this thesis has given me the opportunity to meet, collaborate and more importantly spend great time with several people who are not only colleagues, but also great friends. *Fanoula*, you know that if it hadn't been for you, this thesis would not be structured as it is now. Thank you for everything! I miss a lot our endless arguments.

Gpapaste, I proved you wrong – plenty of times – and here is the greatest proof! You know how to push my buttons, just like I know how to push yours. Keep doing that please! *Soto*, you know that you are perfect with words (as well as hair and looks), now time to grab it in paper for your own thesis too. *Beze*, in the initial acknowledgment I had written that the same goes for you (without the hair), but since then you got your PhD and I'm so happy about it, even though I could not be there to celebrate. Guys, thank you all three for all the help and support throughout these years, as well as the relentless mouhabeti.

Agapi, whatever I may write about you will never be enough. You are definitely one of the smartest people I have ever met and definitely the best administration. You deserve not just one, but plenty of Administration Oscars! I am so grateful that I have you in my life, not only as a colleague, but also as a friend. Thanks for all the understanding, the perfect collaboration and your patience with everything that came up during these past years. We made it!

Sotiri, you have been the greatest officemate I could have ever asked for. Maybe a little too quiet, but never a distraction. *Yianni*, if it hadn't been for you, I would still be trying to find my way through the endless virtual machines. *Maria*, thank you for all the help with SPICE deliverables. *Kiki*, thank you for all the advice; you've been through it all and your advice has always been useful. I will miss your smile everyday when I enter the office, even though you may have not slept all night! *Georgina*, this thesis is the proof that I am not the guy working at the university cafeteria, like I made you believe on the first day we met. Thanks for the deeper discussions (and the food you bring to the office). Hold strong and take it easy!

Aggeliki and *Christo*, you might have finished you PhD when I was still in the beginning, but still your guidance and advice helped me get through everything. *Aggeliki*, I have to follow your advice regarding my future. Keep reminding me that! *Christo*, I still keep in mind a discussion we had around 5 years ago on the train returning from Athens. Remember that?

One of the most amazing opportunities that this PhD has given me was to meet and collaborate with great people abroad. *Arjuna*, my brother, thank you for every single opportunity and piece of advice you have given me! It has really opened up my horizons. *Jon*, you are always an inspiration, not only to me, but to so many people around the world. *Sergey*, I am really glad I finally met you in Xanthi last year. Skype can never replace face-to-face meeting. Thank you for the great collaboration! *Jörg*, thank you for the hospitality and the useful discussions that we held when I visited you in Helsinki. *Jose*, working with you was amazing and really helped me understand what I really love to do. And from now on I plan to chase this stronger than ever!

Pat Pat, *Gargaliane*, *Tsibleki*, *Voulgare*, *Halia*, *Trelopaido*, *Aleko*, *Karra* and *Tasouli*, you know that we are one really big family! If it hadn't been for your support and encouragement (especially at the beginning of this PhD when I was about to give up after my dad passed away) this thesis would have never been published. It doesn't matter if we cannot physically meet often, long live tiletaverna!

Eirini and *Christina*, you have always been there for me for some decades (that I should probably not count – there is no point in giving out our age). Thank you for all the support throughout these years. I know I can always count on you! And you should know the same.

Giota, *Gianni* and *Alexi*, thank you for all the support (including the home-made food) and constantly believing in me!

Mom, Anna and *Giagia* thank you for all the support and trust throughout these years. You should know that you are my motivation and inspiration and I will always do anything I can to make you happy. I hope this thesis makes you a little happier! As long as we are together, we can make everything happen. Never forget that!

Last, but definitely not least, *Stefane*, I still smile at your reaction whenever someone asks you what my job is. Thank you for all the support, help, advice, understanding and patience throughout these years. Sharing my life with the most talented, creative and hardworking person I have ever met has given me endless courage and helped me get through everything. Everyone says that the time someone spends to get a PhD is really hard. Well, for me this time was amazing. And it's all because of you!

Abstract

In the present thesis, we study the problem of the recent explosion in mobile data traffic, which pushes the capacity limits of cellular networks, as well as the issue of the growing digital divide, i.e. the growing disparity between those with sufficient access to the Internet and those who cannot afford access to Internet services. To solve these challenges, we propose solutions that offload data from cellular networks and extend existing Internet coverage. To achieve that, we exploit the principles of Delay/Disruption Tolerant Networking (DTN), since the delay-tolerant nature of several mobile applications, such as e-mail, file transfers or cloud storage synchronisation, allows for delays without significantly affecting user experience.

Initially, we focus on wireless community networks in developing regions that usually rely on low-bandwidth backhaul links that are shared amongst a large user base, driving these links to sub-packet regimes where the per-flow throughput is less than one packet per Round-Trip Time (RTT). It has been proven that Transmission Control Protocol (TCP) does not perform effectively in such conditions, leading to unfairness, high packet loss rates and consecutive timeouts. As an alternative to TCP, we investigate the performance of a less-than-best-effort access method, namely Low Extra Delay Background Transport (LEDBAT) and its fair modification fLEDBAT, in the sub-packet regime of shared backhaul links. Our evaluation results show that fLEDBAT flows achieve higher link efficiency and increased fairness compared to TCP flows. However, when TCP and fLEDBAT flows share the same link in the sub-packet regime, fLEDBAT flows become aggressive, consuming more and more resources. In order for fLEDBAT to function properly, shared bottleneck detection mechanisms and a conservative reaction to consecutive timeouts need to be incorporated into its core.

We then target dense urban environments and seek methods to increase the amount of cellular data offloaded through WiFi Access Points (APs). So far users perform either on-the-spot offloading (OTSO) when a WiFi AP is immediately available, or delay-tolerant offloading (DTO) when the transmission is delayed for some time in case the user encounters an offloading opportunity later. We highlight the fact that the limited amount of available open WiFi APs restrains the gains of existing offloading approaches and propose the Cost-Effective Multi-Mode Offloading (CEMMO) mechanism that enhances pure OTSO and pure DTO with a multi-hop peer-assisted offloading mode (PAO). The newly introduced PAO mode is the first solution of its kind that offloads data from the uplink, independent of content and popularity, since all peer-assisted offloading solutions so far rely on the concept of interests and subscriptions for popular content. Our evaluation results show that CEMMO

successfully selects the most cost-effective transfer method and offloads significantly more data than pure OTSO and pure DTO, even when WiFi availability is limited.

As a next step, we focus on providing free delay-tolerant Internet access to the under-privileged society that is currently excluded from today's digital world. Large-scale deployments of open WiFi APs have been proven unfeasible due to the soaring costs they incur. User-provided networks have been proposed as an alternative, however such approaches only grant Internet access in specific areas, failing to provide extended coverage. As a solution, we design a networking architecture whose backbone framework consists of DTN nodes deployed on both public transport vehicles (data ferries), such as buses and trams, and their corresponding stops (gateways). To achieve data delivery within the Delay Tolerance Threshold that each user defines, we develop the Connectivity Plan Routing Protocol (CARPOOL), a DTN routing protocol that exploits the connectivity graph of public transport networks to select a route that is expected to achieve earliest delivery time between the end-user and an online gateway. We later introduce the Enhanced Connectivity Plan Routing Protocol (CARPOOL+) that exploits not only *a priori* knowledge of contacts between gateways and ferries, but also opportunistic contacts among ferries to compute routes, and also mitigates the impact of typical unpredictable delays caused by traffic, road accidents etc. Our evaluation results show that, compared to other well-known DTN routing protocols, CARPOOL+ achieves the highest delivery ratio with low latency and the lowest overhead.

In the last part of this thesis, we focus on the development of a Future Internet architecture that combines the inherent ability of Information Centric Networking (ICN) to push content to the edges, in order to provide more localised access to important content, with a transmit-when-needed policy provided by DTN. The proposed I-DTN architecture ensures universal coverage by traversing the entire range of connectivity options and enables tighter integration of satellite and terrestrial communications. As a first step towards its implementation, we develop SPICE DTN testbed, a state-of-the-art DTN testbed for terrestrial, satellite and space communications. SPICE testbed includes not only several DTN nodes, but also specialised components that accurately emulate the functionality of typical ground stations (GSs), space links and satellites. Several new protocols and mechanisms mostly for space DTNs have already been implemented and evaluated using SPICE DTN testbed.

Table of contents

List of figures

List of tables

List of acronyms

2D – Two Dimensions

2G – Second Generation Wireless Telephone Technology

3D – Three Dimensions

3G - Third Generation of Mobile Telecommunications Technology

ALARMS - Look-Ahead Routing and Message Scheduling

AMS - Asynchronous Message Service

AOS - Advanced Orbiting Systems

AP – Access Point

API – Application Programming Interface

AQM – Active Queue Management

AWMN - Athens Wireless Metropolitan Network

BDTE – Bundle Delivery Time Estimation tool

BE – Best-Effort

BER – Bit Error Rate

BGP - Border Gateway Protocol

BLER - Bus Line-based Effective Routing

BP - Bundle Protocol

BSP - Bundle Security Protocol

BSS – Bundle Streaming Service

CAPEX - Capital Expenditure

CAPS - Cell-ID Aided Positioning System

CARPOOL - Connectivity Plan Routing Protocol

CARPOOL+ - Enhanced Connectivity Plan Routing Protocol

CCDF – Complementary Cumulative Distribution Function

CCN - Content Centric Networking

CCSDS - Consultative Committee for Space Data Systems

CEMMO - Cost-Effective Multi-Mode Offloading

CFDP - CCSDS File Delivery Protocol

CGR – Contact Graph Routing

COMET – Content Mediator Architecture for Content-Aware Networks

CONIC - Content-Oriented Network with Indexed Caching

COPSS - Content-Oriented Pub/Sub System

COTS – Commercial Off-The-Shelf

CPU - Central Processing Unit

CROWD - Consumer-generated Mobile Wireless Media

CRT - Command Ranging and Telemetry

D2D – Device-to-Device

DECADE - Decoupled Application Data Enroute

DINET - Deep Impact Network

DiPRoPHET – Distance-based Probabilistic ROuting Protocol using History of Encounters and Transitivity

DONA - Data-Oriented Network Architecture

DTI – Delay Tolerance Interval

DTLSR - Delay Tolerant Link State Routing

DTN – Delay/Disruption Tolerant Network(ing)

DTN2 – Delay Tolerant Networking reference implementation

DTNRG – Delay Tolerant Networking Research Group

DTO – Delay-Tolerant Offloading

DTPC – Delay Tolerant Payload Conditioning protocol

E2E – End-to-End

EC – European Commission

ECN - Explicit Congestion Notification

ECSS - European Cooperation for Space Standardisation

EMMA – Environmental Monitoring in Metropolitan Areas

ESA – European Space Agency

fLEDBAT – fair Low Extra Delay Background Transport

FP7 – European Union's 7th Framework Programme for Research and Technological Development

FTP - File Transfer Protocol

GEO – Geosynchronous

GeOpps - Geographical Opportunistic Routing for Vehicular Networks

GeoSpray - Geographic Routing Protocol for Vehicular Delay-Tolerant Networks

GPS – Global Positioning System

GS – Ground Station

GSM – Global System for Mobile communications

GUI – Graphical User Interface

H2020 – HORIZON 2020 European Union's Research and Innovation Programme

HTML5 – HyperText Markup Language 5

HTTP - HyperText Transfer Protocol

I-DTN - Information Centric Delay Tolerant Networking

IBR-DTN – DTN implementation for embedded systems

ICN - Information Centric Networking

ID – Identity

IEEE - Institute of Electrical and Electronics Engineers

IETF – Internet Engineering Task Force

ION-DTN - Interplanetary Overlay Network reference implementation

IP – Internet Protocol

IPSec – Internet Protocol Security

ISP - Internet Service Provider

ITU – International Telecommunications Union

JAXA - Japan Aerospace Exploration Agency

JPL - Jet Propulsion Laboratory

LBE – Less-than-Best-Effort

LCD-Net - Lowest Cost Denominator Networking

LEDBAT - Low Extra Delay Background Transport

LEO - Low Earth Orbit

LoS - Line-of-Sight

LTE - Long Term Evolution

LTP - Licklider Transmission Protocol

LTPCL – LTP Convergence Layer

MAC - Media Access Control

MEO - Medium Earth Orbit

MOC – Mission Operation Center

N4C - Networking for Communications Challenged Communities

NASA - National Aeronautics and Space Administration

NETEM – Network Emulator

NGO - Non-Governmental Organisation

ns-2 – Network Simulator 2

ONE - Opportunistic Network Environment

OPEX – Operating Expenses

OPTraCom - Opportunistic Public Transport Communication

OTSO - On-The-Spot Offloading

OWD – One-Way Delay

P2P – Peer-to-Peer

PAO - Peer-Assisted Offloading

POP3 - Post Office Protocol 3

PRoPHET - Probabilistic ROuting Protocol using History of Encounters and Transitivity

PSIRP - Publish Subscribe Internet Routing Paradigm

PSM - Power Save Mode

PSS - Portable Satellite Simulator

PUB/SUB - Publish/Subscribe

PURSUIT – Publish/Subscribe Internet Technology

QoS - Quality of Service

RAM – Random Access Memory

RED - Random Early Detection

RFC – Request For Comments

RSS – Rich Site Summary

RTT - Round-Trip Time

SALSA - Stable and Adaptive Link Selection Algorithm

SCAMPI – Service Platform for Social Aware Mobile and Pervasive Computing

SDR - Space Data Routers

SeNDT - Sensor Network with Delay Tolerance

SIPTO - Selective IP Traffic Offload

SLE - Space Link Extension

SMS – Short Message Service

SMTP - Simple Mail Transfer Protocol

SPICE - Space Internetworking Center

STK - Satellite Tool Kit

TAQ - Time-out Aware Queuing

TC – Telecommand

TCP - Transmission Control Protocol

TCPCL – TCP Convergence Layer

TibTec - Tibetan Technology Center

TM – Telemetry

TTL – Time-To-Live

TV - Television

UCAN - Unified Cellular and Ad Hoc Network

UDP - User Datagram Protocol

UDPCL – UDP Convergence Layer

UI – User Interface

UK-DMC – British Disaster Monitoring Constellation satellite

uTP – uTorrent Transport Protocol

V2V - Vehicle-to-Vehicle

VoD - Video-on-Demand

VPN - Virtual Private Network

VSAT - Very Small Aperture Terminal

WCN - Wireless Community Network

WDM - Working Day Movement

WLAN – Wireless Local Area Network

ZBMF - Zone Based Message Ferrying

1. Introduction

1.1 Description and context

In the present thesis, we study the problem of the recent explosion in mobile data traffic, which pushes the capacity limits of cellular networks, as well as the issue of the growing digital divide, i.e. the growing disparity between those with sufficient access to the Internet and those who cannot afford access to Internet services. To solve these challenges, we propose solutions that offload data from cellular networks and extend existing Internet coverage. To achieve that, we exploit the principles of Delay/Disruption Tolerant Networking (DTN), since the delay-tolerant nature of several mobile applications, such as e-mail, file transfers and cloud storage synchronisation, allows for delays without significantly affecting user experience.

Initially, we focus on wireless community networks (WCNs) in developing regions that usually rely on low-bandwidth backhaul links that are shared amongst a large user base, driving these links to sub-packet regimes where the per-flow throughput is less than one packet per Round-Trip Time (RTT). It has been proven that Transmission Control Protocol (TCP) does not perform effectively in such conditions, leading to unfairness, high packet loss rates and consecutive timeouts. As an alternative to TCP, we investigate the performance of a less-than-best-effort (LBE) access method, namely Low Extra Delay Background Transport (LEDBAT) and its fair modification fLEDBAT, in the sub-packet regime of shared backhaul links. **Our intention is to explore the feasibility of using LBE or scavenger access methods, which have been designed to be less aggressive than TCP, instead of TCP for uploading content over bandwidth constrained backhauls.**

We then target dense urban environments and seek methods to increase the amount of cellular data offloaded through WiFi Access Points (APs). So far users perform either on-the-spot offloading (OTSO) when a WiFi AP is immediately available, or delay-tolerant offloading (DTO) when the transmission is delayed for some time in case the user encounters an offloading opportunity later. However, the limited amount of available open WiFi APs restrains the gains of existing offloading approaches. As a solution, we propose the Cost-Effective Multi-Mode Offloading (CEMMO) mechanism that enhances pure OTSO and pure DTO with a multi-hop peer-assisted offloading mode (PAO). All peer-assisted offloading solutions so far rely on the concept of interests and subscriptions for popular content. The newly introduced PAO mode is the first solution of its kind that offloads data from the uplink, independent of content and popularity. **Our intention is to develop an offloading**

mechanism that leverages the overall offloading capability and increases the total amount of data that is offloaded from cellular networks.

As a next step, we focus on providing free delay-tolerant Internet access to the under-privileged society that is currently excluded from today's digital world. Large-scale deployments of open WiFi APs have been proven unfeasible due to the soaring costs they incur. User-provided networks have been proposed as an alternative, however such approaches only grant Internet access in specific areas, failing to provide extended coverage. As a solution, we design a networking architecture whose backbone framework consists of DTN nodes deployed on both public transport vehicles, such as buses and trams, and their corresponding stops to operate as data ferries and gateways, respectively. To achieve data delivery within the Delay Tolerance Threshold that each user defines, we develop the Connectivity Plan Routing Protocol (CARPOOL), a DTN routing protocol that exploits the connectivity graph of public transport networks to select a route that is expected to achieve earliest delivery time between the end-user and an online gateway. We later introduce the Enhanced Connectivity Plan Routing Protocol (CARPOOL+) that exploits not only *a priori* knowledge of contacts between gateways and ferries, but also opportunistic contacts among ferries to compute routes, and also mitigates the impact of unpredictable delays caused by traffic, road accidents etc. **Our intention is to develop a networking architecture that leverages public transport networks in urban environments to extend existing Internet connectivity.**

In the last part of this thesis, we focus on the development of a Future Internet architecture that combines the inherent ability of Information Centric Networking (ICN) to push content to the edges, in order to provide more localised access to important content, with a transmit-when-needed policy provided by DTN. Such an architecture not only spurs innovation for a wide range of new services and applications, but also encompasses existing successful Internet services. **Our intention is to build an Internet architecture that ensures universal coverage by traversing the entire range of connectivity options and enables tighter integration of satellite and terrestrial communications.** As a first step towards its implementation, we develop SPICE testbed, a state-of-the-art DTN testbed deployed at the Space Internetworking Center in Xanthi, Greece. SPICE DTN testbed is an experimental research environment that has been specifically designed for terrestrial, satellite and space communications. SPICE testbed includes not only several DTN nodes, but also specialised components that accurately emulate the functionality of typical ground stations (GSs), space links and satellites.

1.2 Problem definition

The Internet has crossed new frontiers with new applications and services being offered that enable remote health care, education, employment, e-governance, digital economy, social networking etc. Providing efficient Internet access to everyone is considered as one of the fundamental requirements in today's digital age, such clean water, roads, schools etc. Internet access needs to be universally available in terms of affordability and ability to contribute to the wider Internet community, providing true digital inclusion of any member of the society. Such an inclusion will not only enable a multitude of societal benefits, but also play a vital role in improving the safety and security of societies. In the reality of today's Internet, the vision of digital inclusion faces two major challenges: **the challenge of the continuously growing global mobile data traffic** and **the challenge of a growing digital divide**.

Due to the recent proliferation of smartphones and tablets, mobile data traffic has been explosively growing and pushing the capacity limits of cellular networks. The data traffic growth threatens to congest cellular networks severely and, therefore, impair Quality of Service (QoS). In highly congested wireless networks, many operators already perform de-prioritisation of heavy users for congestion management. The current economic models for accessing the Internet are built on the basic Best-Effort (BE) model and the network protocols that govern the transmission of data are adapted to suit the BE nature of the Internet to contend for available resources, which makes it impossible to support service models that could lower the cost of Internet access, e.g. through adaptive QoS. This raises **the issue of investigating the performance of scavenger or less-than-best-effort (LBE) access methods, which were introduced in an effort to share the unused capacity of backhaul links with the underprivileged without affecting the performance of paid customers, in the heavily congested sub-packet regime of shared backhaul links**. With the global mobile data traffic forecasted to increase 13 times until 2017, cellular network providers need to find cost-effective ways to handle the growing traffic demands with expected QoS levels. Infrastructure upgrade is the intuitive solution to deal with the congestion problem; however, infrastructure upgrades are costly and, by themselves, insufficient as a financially sustainable solution for maintaining the expected QoS for the growing traffic. Operators need to balance end-user satisfaction, infrastructure investments, i.e. capital expenditure (CAPEX), and operating expenses (OPEX).

Offloading mobile data through WiFi networks tackles the congestion problem by reducing the amount of cellular network traffic. From the perspective of cellular network providers, offloading seems to be a practical solution because the cost of WiFi APs needed for the offloading process is relatively low, WiFi technology uses unlicensed bands and offloading can satisfactorily serve delay-tolerant types of data traffic. DTN architecture and its supporting Bundle Protocol (BP) is an emerging technology to support the new era in internetworking by providing delay-tolerant access even when traditional continuous end-to-end (E2E) connectivity fails. DTN supports E2E message transfer over a variety of networking technologies and constitutes an overlay layer, called the bundle layer. In order to deal with constant disruptions in DTNs, BP utilises store-and-forward mechanisms, along with the notion of persistent message storage. So far, offloading research efforts have been focused on either OTSO schemes, or DTO solutions. However, the performance of both offloading types is highly dependent on the amount of available WiFi APs and their vicinity to a user. This calls for **the need for an offloading solution that leverages the overall offloading capability and, therefore, increases the total amount of data that is being offloaded from cellular networks regardless of its content and popularity**.

As far as the growing digital divide is concerned, several economic models, such as providing restricted Internet access during night at a lower price, have been proposed in the recent past, in an effort to provide Internet access to all members of the society. Nonetheless, these models are not affordable to all, leaving certain members of the society with the only alternative of using random WiFi hotspots when available. Several governments and local administrations have undertaken the initiative to deploy WiFi hotspots in points of interests, however cost-efficiency is a critical factor that hinders extended deployments. User-provided networks, where an Internet connection is shared freely and transparently among end-users in a way that is technically and legally independent of access or infrastructure providers, have also been proposed as a solution. For example, BT FON initiative encourages FON members to share their home broadband connection and get in return free access at millions of other FON hotspots worldwide. Even though the aforementioned solutions can provide free Internet access in specific areas, they fail to provide extended coverage. This highlights **the need for an architecture that enhances the overall basic Internet connectivity in urban environments with minimum cost.** To achieve that, the inherent capabilities of DTNs can be combined with mobile applications that serve non-critical data to provide basic delay-tolerant Internet access. Such an architecture can be built using low-cost components, such as Raspberry Pi, and can be further coupled with concepts that have already been successfully applied in remote regions, such as message ferrying, to increase communication opportunities.

Overall, the existing Internet architecture is seriously challenged to ensure universal service provisioning due to geographic, socio-economic or technological reasons. Access problems are typical in sparsely spread populations living in physically remote locations, since it not cost-effective for Internet Service Providers (ISPs) to install the required infrastructure for broadband Internet access to these areas. Even in developed countries, several members of the society remain disconnected, since they find themselves unable to pass a necessary credit check or live in circumstances that are too unstable to commit to lengthy costly broadband services. From the technological perspective, the current Internet is progressively reaching a saturation point in meeting increasing user expectations and is progressively showing its inability to efficiently respond to new challenges. This calls for **the need for an Internet architecture that seamlessly integrates multiple transmission technologies to reduce transmission cost and increase efficiency, flexibility and dependability.** This architecture needs to ensure universal coverage by traversing the entire range of connectivity options through a single unifying communication architecture with a single set of abstractions.

1.3 Contributions of the present thesis

In the present thesis, we target the challenges of the growing cellular data traffic and the growing digital divide by **designing, implementing and evaluating networking architectures that increase cellular data offloading opportunities and extend existing Internet access to both areas typically not covered (i.e. geographically) and people that cannot afford the cost. To achieve that, we exploit the delay-tolerant nature of several applications and couple the proposed architectures with new peer-assisted and vehicle-assisted forwarding methods, which are based on the DTN paradigm.** Our research is focused on environments where the problem of cellular network congestion is mostly evident; initially we focus on the heavily-shared backhaul links of WCNs, while later we target dense urban environments. Regarding the open research issues that were presented in the previous subsection, the contributions of the present thesis are divided into the following areas:

- The performance evaluation of BE and LBE access methods in the sub-packet regime of shared backhaul links, as well as their interaction.
- The design, implementation and evaluation of an architecture that incorporates different modes of operation to enhance cellular network offloading with a new multi-hop peer-assisted offloading method.
- The design, implementation and evaluation of a networking architecture that leverages public transport networks in urban environments to extend overall Internet connectivity with minimum cost, since it exploits existing WiFi hotspots and is built using low-cost components, and its coupling with a routing protocol that exploits every contact opportunity.
- The design of an architectural framework that efficiently exploits all possible communication opportunities, while providing a unified abstraction to application

developers for supporting current Internet-based services and enabling innovative future solutions, such as the tighter integration of terrestrial and satellite networks.

Below we describe the steps that we followed to carry out our research and we highlight the scientific contribution of this thesis in the form of four issues.

Issue 1: The need to investigate the performance of scavenger or LBE access methods in the heavily congested sub-packet regime of shared backhaul links.

In the first part of our thesis, we focus on the sub-packet regime, i.e. the result of heavy sharing on the order of several competing flows operating over low-bandwidth backhaul links of WCNs. The sub-packet regime has not been a traditionally important region of operation for network flows and, as a result, this space has remained relatively unexplored. TCP and other common congestion control protocols break down in the sub-packet regime, resulting in severe unfairness, high packet loss rates and repetitive timeouts. In addition, none of the standard TCP variants or known queuing mechanisms offer substantial performance gains in the sub-packet regime.

Within this framework, we first confirm the performance of TCP in the sub-packet regime and, then, evaluate the performance of LEDBAT access method, as well as its fair modification fLEDBAT, under the same conditions. LEDBAT is one of the most popular scavenger transport methods, proposed by BitTorrent Inc., and is already an experimental Request For Comments (RFC) by the Internet Engineering Task Force (IETF). TCP, LEDBAT and fLEDBAT have been evaluated using Network Simulator 2 (NS-2). Our findings show that LEDBAT achieves higher link efficiency and fairness when compared to TCP in a variety of sub-packet regime scenarios. However, when TCP and fLEDBAT flows share the same link in the sub-packet regime, fLEDBAT flows fail to measure the actual base delay due to the standing queue and become aggressive, consuming more resources than TCP. We also investigate whether active queue management (AQM) can be the solution to sub-packet regime and observe that AQM cancels the gains of fLEDBAT by making fLEDBAT flows more aggressive. We conclude that fLEDBAT needs to react more conservatively after consecutive timeouts and also include shared bottleneck detection mechanisms, in order to correctly adjust its congestion window in the sub-packet regime.

Issue 2: The need for an offloading solution that enhances the existing delay-tolerant functionality in order to leverage the overall offloading capability and, therefore, increase the total amount of data that is being offloaded from cellular networks.

In the second part of our thesis, we elaborate on the concept of data offloading from cellular networks, since cellular networks will not be able to handle the upcoming explosion in mobile data growth mainly due to economic reasons that pose barriers to wide infrastructure upgrades. First, we highlight the fact that pure OTSO and pure DTO cannot offer extended offloading capabilities since their performance strongly depends on the availability of WiFi APs. We also elaborate on the existing mobile data offloading solutions through opportunistic communications that have been introduced in the context of disseminating popular content, a restriction that hinders their wide adoption. Overall, we highlight the need for a peer-assisted method for offloading traffic from the uplink, regardless of the offloaded content and its popularity.

Within this framework, we design, develop and evaluate CEMMO, a cost-effective multi-mode offloading mechanism that offloads mobile traffic from the uplink and incorporates three modes of operation: cellular delivery, DTO and PAO. OTSO is always exploited when available. The introduction of the first peer-assisted offloading mode, where the offloaded traffic is delivered through intermediate mobile devices regardless of content and popularity, leverages the overall offloading capability and increases the total amount of data that is offloaded from cellular networks. For PAO, we develop a new data-forwarding scheme with low storage and energy overhead that uses regionally and temporally restricted flooding. To achieve that, we introduce a mobility and connectivity prediction model based on a Markov process. CEMMO allows the cellular operator to select the most effective mode for communicating mobile data with different delay constraints based on the estimated costs of delivery through each of the three modes. The overall cost, as defined by each operator, can include transfer costs, energy costs, as well as incentives to motivate user participation. Moreover, CEMMO provides cellular operators with knowledge on the amount of data offloaded by each user. Such knowledge is currently unavailable and could help cellular operators design the expansion of their networks accordingly.

CEMMO has been implemented and evaluated using the Opportunistic Network Environment (ONE) simulator. We evaluate the performance of CEMMO under a wide variety of scenarios and conclude that CEMMO increases offloading ratio and reduces cost compared to pure OTSO and pure DTO, by switching between transfer policies according to the needs of each transfer. Our results also show that CEMMO reduces the total energy consumption of the mobile devices involved in the offloading process, since less data are offloaded through the energy-intensive third generation of mobile telecommunications technology (3G).

Issue 3: The need for an architecture that enhances the overall basic Internet connectivity in urban environments with minimum cost.

In the next part of our research, we focus on the challenge of the growing digital divide. Our research targets dense urban networks, where members of the society cannot afford an Internet connection even though several connectivity alternatives exist. We aim at solutions that can have a dual role: provide free Internet connectivity to the digitally excluded through delay-tolerant applications and help offload congested cellular networks. To achieve that, we exploit the inherent characteristics of DTNs; *store-and-forward* and *custody transfer*.

Within this framework, we design, implement and evaluate a networking architecture that leverages public transport networks in urban environments to extend overall Internet connectivity in a delay-tolerant way. To achieve that, we deploy DTN nodes on public transport vehicles, such as buses and trams, (data ferries) and their corresponding stops (gateways), forming the backbone framework of the proposed architecture. End-users (i.e. pedestrians) in the vicinity of ferry stops forward Internet access requests from their mobile devices to offline DTN gateways, which in turn exploit DTN-enabled ferries to relay these user requests to gateways with Internet connectivity that are capable of handling them. In order to select a path that is expected to achieve earliest data delivery, we develop CARPOOL, a DTN routing protocol designed to route data based on the knowledge of the public transport vehicles schedule. Based on our experience, we further enhance CARPOOL and develop CARPOOL+, a DTN routing protocol that has been specifically designed for urban transport networks and has the ability to exploit not only *a priori* knowledge of contacts between gateways and ferries, but also opportunistic contacts among ferries, to compute routes. Unlike other routing proposals that exploit public transport networks as DTN backbone, CARPOOL+ does not rely on strictly deterministic values and has been designed to achieve high delivery ratio even when significant delays occur, e.g. due to road traffic.

Both CARPOOL and CARPOOL+ have been implemented and evaluated using the ONE simulator. The initial performance evaluation of CARPOOL shows that an acceptable level of service can be provided, since CARPOOL outperforms other popular DTN routing protocols in terms of delivery ratio (i.e. service probability) and overhead (i.e. potential to accommodate more users). Our extended simulation results show that CARPOOL+ improves delivery ratio and average latency compared to CARPOOL not only when ferries follow the predefined schedule, but also when delays occur. Moreover, the comparison of CARPOOL+ with the most well-known DTN routing protocols shows that CARPOOL+ achieves the highest delivery ratio with low latency and minimum overhead.

Issue 4: The need for an Internet architecture that seamlessly integrates multiple transmission technologies to reduce transmission cost and increase efficiency, flexibility and dependability.

In the last part of our thesis, we highlight the need for a universal communication platform that encompasses a working set of protocols, systems, services and tools. Following the evaluation results of the present thesis, we provide an overview of our joint work on the design of an architecture that efficiently exploits all possible communication opportunities, from fixed or mobile broadband networks to disruptive networks and satellite communications, while providing a unified abstraction to application developers for supporting current Internet-based services and enabling innovative future solutions. Such a platform helps achieve full digital inclusion within and across societies by reaching out to people who are socially and digitally isolated due to socio-economic or geographical disparity.

The proposed framework is an Information Centric Delay Tolerant Networking (I-DTN) architecture that combines the advantages of DTN with the advances in the area of ICN and its inherent ability to push content to the edges, providing more localised access to important content and reducing access cost per bit through the enablement of a transmit-when-needed policy. In particular, the framework of the proposed I-DTN architectural framework combines the Internet Protocol (IP), ICN and DTN solutions into a single system architecture, exposing a common information-centric abstraction to applications, while supporting a range of networking protocols over different underlying heterogeneous networks. I-DTN architecture also enables tighter integration of terrestrial and satellite communications, in order to provide ubiquitous coverage. As a first step towards the implementation of I-DTN architecture, we have developed SPICE DTN testbed, a state-of-the-art DTN testbed for terrestrial, satellite and space communications. Several new protocols and mechanisms mostly for space DTNs have already been developed and evaluated using SPICE DTN testbed.

1.4 Thesis structure

In **Chapter 2**, we study the relevant background and related work to this thesis. We begin by presenting the key characteristics of the DTN and ICN paradigms. We then provide an overview of WCNs and focus on the extreme conditions of the sub-packet regime. We continue by describing LEDBAT access method and discuss its potential to perform more efficiently than TCP in congestion conditions. Next, we present enabling technologies for cellular network decongestion and focus on specific cellular offloading solutions. Finally, we elaborate on the applicability of DTN by presenting several DTN use cases and applications, as well as DTN routing approaches.

In **Chapter 3**, we first describe the mobility and connectivity prediction model that CEMMO utilises. We then present CEMMO mechanism that enhances the existing offloading modes with PAO mode and detail its operation. A sample scenario is also included.

In **Chapter 4**, we present a DTN architecture for public transport networks. We first describe the initial architecture that was coupled with CARPOOL DTN routing protocol. Then, we detail the enhanced architecture that exploits not only scheduled, but also opportunistic contacts among DTN nodes and present CARPOOL+ routing protocol, the evolution of CARPOOL.

In **Chapter 5**, we present the methodology that we follow for the evaluation of the architectures proposed in the present thesis. We describe the evaluation scenarios, the evaluation metrics, as well as the simulation tools that we use.

In **Chapter 6**, we present our experimental results. We verify the operation of the proposed mechanisms and architectures and we compare their performance with relevant mechanisms and protocols that have been proposed in literature. We analyse our results and draw meaningful conclusions.

In **Chapter 7**, we provide an overview of the I-DTN architecture that ensures universal Internet coverage. We then describe the core of SPICE DTN testbed, a platform that allows for extensive experiments involving terrestrial, satellite and space communications.

In **Chapter 8**, we conclude the present thesis. We discuss the most important outcomes of the present thesis and we highlight our contributions on the problems that this thesis targets. We also discuss open issues for future work.

2. Background and related work

In this chapter, we study the relevant background and related work that constitutes the basis of our research. In the first part, we discuss the inherent characteristics of the DTN paradigm (Section 2.1), as well as the benefits of the ICN paradigm (Section 2.2); both technologies are key to the solutions proposed in the present thesis. In the second part, our research is divided into two major network types that both face the challenges studied in the present thesis: the low-bandwidth backhaul links in developing regions that are shared by a large user-base leading to severe congestion, and the typical cellular networks in developed regions that cannot handle the constantly growing cellular data traffic. As far as the former are concerned, we provide an overview of wireless community networks (WCNs) and focus on the extreme conditions of the sub-packet regime (Section 2.3). We then present the new LCD-Net Internet paradigm, describe LEDBAT access method and discuss its potential to perform more efficiently than TCP in congestion conditions (Section 2.4). As far as the latter are concerned, we first present enabling technologies for cellular network decongestion and, later, focus on specific cellular offloading solutions (Section 2.5). In the last part of this chapter, we present several DTN use cases and applications, as well as DTN routing approaches, that are relevant to the problem of extending existing Internet connectivity (Section 2.6).

2.1. Delay-Tolerant Networking architecture

DTN [1][2] is an emerging technology to support a new era in internetworking and interoperable communications, either on Earth, or in Space. Like IP, DTN operates on top of existing network architectures, creating a DTN overlay. The key advantage over IP is that DTN allows interconnecting networks with very diverse characteristics. In particular, DTN extends internetworking in the time domain: rather than assuming a continuous E2E path as IP networks do, DTN operates in a store-and-forward fashion: intermediate nodes assume temporary responsibility for messages and keep them until the next opportunity arises to forward them to the next hop. While stored, messages may even be physically carried within a node as the node is transported: the model is also termed *store-carry-forward*. This inherently deals with temporary disconnections or disruptions and allows the connection of nodes that would else be disconnected in space at any point in time.

DTN can bind heterogeneous networks and incorporate devices and applications with limited functionality. The achieved service flexibility can unify diverse environments under a common context. Imagine for example, space sensors, which produce useful data for Earth observation, underwater environments, seismographic sensors, safety authorities and mobile users, all participating opportunistically in a single network, formed dynamically and on the fly in an emerging situation. Such a network incorporates different devices, protocols, user groups and organisations, which, however, present a common characteristic under specific circumstances: for example, they could all assist in an emergency situation. Although the situation might only involve a local geographical area, an efficient operation requires a combined effort of collecting remote data, adjusting remote devices to monitor the specific region, synchronising authorities also beyond the region, and informing users in and around the region about the situation. Key design assumptions of Internet protocols such as short RTT, absence of disruptions and continuous E2E path availability are challenged by such a concept. In particular, DTN operates successfully over challenged networks that may be characterised by:

- *Disruptive connectivity,* which means that a connection between the sender and the receiver is not always available (and may never be);
- *High propagation delay*, which ranges from a few seconds to several minutes;
- *High bit error rates (BER)* up to 10^{-3} for deep-space networks, and
- *High bandwidth asymmetry,* which can reach 1000:1 for space links.

In essence, DTN technology enables seamless communication between diverse devices by hiding the complexity, the diversity, and the potential discontinuity of the heterogeneous E2E communication from the service. This is accomplished by the combination of an inherently asynchronous interaction service offered to applications and the use of numerous underlying convergence layer protocols, which map to underlying layers. These convergence layer protocols offer a communication framework that allows for communication among a great variety of devices that range from common sensors and smart mobile devices to deep space sensors and embedded routers. Various options on convergence layers for different types of common underlying transport and network protocols are already available in the existing DTN reference implementations, which are discussed later in this subsection.

The core of the DTN architecture lies in the BP [3]. BP defines the bundle as the core unit of the DTN architecture; a *bundle* is a series of data blocks that is routed in a store-and-forward manner between nodes over various transport networks. In order to improve the efficiency of transfers, the DTN architecture supports several features, such as *fragmentation* and *custody transfer*. DTN fragmentation and reassembly ensures that contact volumes are fully utilised, avoiding retransmission of partially-forwarded bundles. Bundle fragmentation can be performed either proactively at the sender or reactively at an intermediate node, if the reduction of the size of a bundle is required to forward it. Reassembly can be performed either at the destination or at some other node on the route to the destination. Custody transfer is another important feature of the DTN architecture. In essence, custody transfer moves the responsibility for reliable delivery of a bundle among different DTN nodes in the network. Whenever a node accepts the reliable delivery responsibility of a bundle, it is called the "custodian" of this bundle.

As far as the existing DTN reference implementations are concerned, several implementations have been developed during the last years, each targeting different applicability scenarios. The goal of DTN2 reference implementation (DTN2) [4] is to clearly embody the components of the DTN architecture, while also providing a robust and flexible software framework for experimentation, extension, and real-world deployment in terrestrial environments. Digital communication between interplanetary spacecraft and space flight control centers on Earth, however, is subject to constraints that differ in some ways from those that characterise terrestrial communications. For this reason, the Jet Propulsion Laboratory (JPL) has developed "Interplanetary Overlay Network" (ION-DTN) [5], an alternative implementation of BP, which is aimed at addressing those constraints and enabling DTN communications in interplanetary mission operations. IBR-DTN [6] is another implementation of the bundle protocol designed for embedded systems. IBR-DTN can be used as framework for DTN applications; its module-based architecture with miscellaneous interfaces makes it possible to change functionalities like routing or bundle storage just by inheriting a specific class. Regarding opportunistic communications, the Service Platform for Social Aware Mobile and Pervasive Computing (SCAMPI) router [7] was developed as an opportunistic application development platform for mobile embedded devices with support for both native application development and HyperText Markup Language 5 (HTML5). The platform includes an opportunistic router based on the Delay Tolerant Networking Research Group (DTNRG) [8] specifications, a framework for developing pure HTML5 applications that take advantage of the router, and an opportunistic application distribution system. Its evolution, the Liberouter [9], is a complete communication system that serves as WLAN access points, individual or (dis)connected ones, and offers message storage and relaying in combination with a distributed app store. All DTN reference implementations are based on the BP and related convergence layer specifications. An IETF [10] working group has been recently formed for the standardisation of DTN [11].

Regarding space communications, DTN is already being standardised by the Consultative Committee for Space Data Systems (CCSDS) [12]. The DTN2 reference implementation has been tested in a DTN testbed in orbit using British Disaster Monitoring Constellation satellite (UK-DMC) [13], while the ION-DTN implementation has been loaded onto the EPOXI spacecraft in deep space as part of the Deep Impact Network (DINET) experiment [14]. Our research group has significantly contributed to the evaluation of existing DTN protocols in space environment [15][16][17], the development of new protocols [18][19][20], mechanisms [21][22][23][24] and services [25]. Moreover, we have developed a state-of-the-art DTN testbed for terrestrial and space communications [26][27][28] that can be exploited to evaluate a variety of protocols under several communication scenarios. More details on DTN use cases and applications are provided in sub-section 2.5, while SPICE DTN testbed is described in Chapter 7.

All key mechanisms developed, implemented and evaluated in the framework of the present thesis (i.e. CEMMO, CARPOOL/CARPOOL+ and I-DTN) are based on the DTN architecture and exploit its core characteristics.

2.2. Information Centric Networking

ICN is an approach to evolve the Internet infrastructure away from a host-centric paradigm based on the assumption of the continuous connectivity and the E2E principle, to a network architecture, which is based on named content or data. In ICN, data become independent from location, application, storage, and means of transportation, enabling in-network caching and replication. The expected benefits are improved efficiency, better scalability with respect to information/bandwidth demand and better robustness in challenging communication scenarios.

Given the aforementioned benefits, ICN has been increasingly attracting attention in the wider research community, fueled by research efforts in various parts of the world. Data-Oriented Network Architecture (DONA) [29] is one of the first clean-slate ICN proposals. DONA uses flat, self-identifying and unique names for information objects and binds the act of resolving requests for information to locating and retrieving information. The Content Centric Networking (CCN) [30] project developed a new network layer based on an information centric approach. CCN proposes a name-based routing system for locating and delivering named data packets. The fundamental entities in CCN are interest and data packets. The namespace is used to route requests for content, called interest, toward the producer using longest prefix matching. Namespace prefixes are announced in the network in a similar way as IP prefixes are announced using the Border Gateway Protocol (BGP) today. When an interest packet meets the corresponding data packet, either a cached copy found en-route to the producer, or ultimately at the producer, the data packet is sent back following the reverse path. Although CCN has been mostly developed to retrieve content stored in the network, its use in case of real-time applications such as audio-conference has been also investigated in [31]. CCN is the basis of several relevant research projects. Content-Oriented Publish/Subscribe System (COPSS) [32] presents an extension to the basic CCN architecture, by introducing a multicast-based publish/subscribe (PUB/SUB) capability, and extends the naming framework with the introduction of the concept of content descriptors to enable efficient large-scale information dissemination. The Content Mediator Architecture for Content-aware Networks (COMET) [33] closely follows CCN, differentiated in two ways: it modifies the IPv4 packet header in order to make the architecture backwards compatible and it reduces the state maintained by routers via caching; a routing table has a finite size and a router stores the most recently used entries.

In Publish/Subscribe Internet Technology (PURSUIT) [34][35] and Publish Subscribe Internet Routing Paradigm (PSIRP) [36], a PUB/SUB system is proposed based on the context of separating the architecture into the core functions of *rendezvous* (which matches supply of information to its demand and results in some form of information that is used for binding the information delivery to a network location), *topology management* (which realises the management of the overall delivery topology and the formation of specific delivery graphs) and *forwarding* (which receives publications and forwards them to the network and/or the local node). Content-Oriented Network with Indexed Caching (CONIC) [37] is another network architecture designed for efficient data dissemination using storage and bandwidth resources in end-systems, i.e. available storage in end-hosts is used for caching. A similar approach, where content is cached in routers is the Cache and Forward architecture [38]. MultiCache [39] is another information-centric overlay network architecture aiming to improve network utilisation via resource sharing. In MultiCache, operators deploy and control proxy overlay routers that enable the joint provision of multicast and caching, targeting both synchronous and asynchronous requests.

As we move from the core towards the edges of the network, many opportunities for performance and efficiency improvements can be exploited through ICN. One such opportunity is caching in the *last mile,* taking advantage of the huge number of devices with spare memory and bandwidth existing at the periphery right next to the users. Caching itself is a very old topic in networking research. Past research has focused on overlay, or web proxy caching [40], while hierarchical and cooperative caching [41] has examined file replication in a string of caches, as well as cache placement techniques [42][43]. Some form of cooperation between caches at different levels is assumed, with different levels having different responsibilities. The ICN paradigm provides the opportunity of caching inside the network, because it names content instead of end-hosts and, thus, abstracts from the location where the content or its copies is kept. Caching of chunks instead of whole files [44], raises issues of fair resource allocation and efficient content multiplexing, directly affecting packet-level network dynamics. In CCN, content packets are cached by default in every router that the packet traverses. An attempt to model the behaviour of in-network caching trees can be found at [45], whereas in [46] a Markovian system model capturing the essential dynamics of in-network caching is defined as a way to calculate the survival time of information items in an ICN. In [47], the authors propose an autonomic cache management architecture that dynamically (re-)assigns information items to caches in an ICN. Cooperative content caching between ISP networks has also been investigated, aiming to reduce overall content distribution costs, in particular for peer-to-peer (P2P) traffic [48]. Packet caching in routers has also been proposed to exploit the benefits of packet-level redundant content elimination [49]. In-network caching techniques are also being investigated in the IETF for potential standardisation in the Decoupled Application Data Enroute (DECADE) Working Group [50].

In Chapter 7, we provide an overview of our joint work on the design of the first universal communication framework (I-DTN) that combines two emerging architecture and connectivity approaches: ICN and DTN. Such a unified architecture aggressively seeks to widen the connectivity options and provide flexible service models beyond what is currently pursued in the game around universal service provisioning.

2.3. Wireless Community Networks and the sub-packet regime

Having described the core characteristics of the DTN and the ICN paradigm that are key to the solutions proposed in the present thesis, we now focus on the challenge of the growing mobile data traffic in developing regions. Even though the growth in the popularity of the Internet during the last decade is unprecedented, providing sustainable, cost-effective and high-quality Internet connection with coverage for all citizens is still a challenging problem. Access problems often result from sparsely spread populations living in physically remote locations; it is simply not cost-effective for ISPs to install the required infrastructure for broadband Internet access to these areas. Coupled with physical limitations of terrestrial infrastructures to provide last mile access, remote communities also incur higher costs for connection between the exchange and the backbone network mainly due to distance. Internet statistics reports from sources such as Akamai [51] and the International Telecommunications Union (ITU) [52] indicate that for a variety of reasons, broadband growth in many areas of the world, including most of Africa and South America, continues to lag behind.

The socio-economic development of rural regions in the third world highly depends on access to information. Low-cost wireless local area network (WLAN) equipment operating in unlicensed spectrum has revolutionised local area communications, introducing novel schemes for open wireless connectivity and even new business models. The ease of deployment of WiFi has made it ubiquitous in densely populated urban areas, leading to the development of the first wireless communities. Using inexpensive wireless technology to communicate, autonomous wireless internetworks have been built, offering a variety of broadband services. WCNs that provide cost-effective Internet access both in developed and emerging regions are usually run by non-profit organisations which can also cooperate with local stakeholders to develop community services, including local networking, voice connections and Internet access.

Three of the largest wireless community networks have been developed in Europe. Guifi.net [53], located in Spain, is an open, free and neutral telecommunications network built through a P2P agreement where everyone can join the network by providing their connection, and therefore, extending the network and gaining connectivity to all. Similarly, Funkfeuer [54] is a free, experimental, unregulated network in Austria, which has the potential to bridge the digital valley between the social layers and deliver the infrastructure and the knowledge for it. Athens Wireless Metropolitan Network (AWMN) [55] is another grassroots wireless community, taking advantage of new, state-of-the-art wireless technologies, to connect people and services.

Rural wireless networks, which usually share a costly link to the Internet with high level of loss rates and packet reordering [56][57] among a large user base, have also been deployed. For example, there exists a large number of low-bandwidth community network environments in the developing world with high levels of network sharing [58][59] where a 128 Kbps – 2 Mbps access link may be shared by 50 – 200 users [60]. One of the largest rural wireless mesh networks, the Dharamshala Wireless Mesh Network that is a community network for the Tibetan Technology Center (TibTec), interconnects over 10000 rural users through 2000 computers in a difficult mountainous terrain within a radius of 70 km around Dharamshala, a town located in the Himalayan region of northern India. This network is part of the AirJaldi networks [61] that utilise small-size, low power nodes mounted on low masts and are characterised by a very low ecological footprint and the ability to operate under the most demanding environmental conditions. In South Africa, the Peebles Valley mesh network [62] consists of long distance wireless links covering 15 sq. km and the Internet bandwidth is provided by an HIV/AIDS clinic which has a sponsored very small aperture terminal (VSAT) providing 256 kbps / 64 kbps connection since no other connectivity options were available. Similarly, the LinkNet mesh network [63] is located in the very remote village of Macha, Zambia. A distance of 70 km from a tarred road or landline phone, Macha provides rural connectivity to the John’s Hopkins Malaria Institute at Macha, the Macha Mission Hospital and the community at large within Macha.

With the ubiquitous availability of low-cost mobile devices, cloud-storage, applications that capitalise on the benefits offered by the cloud, and the increase in the complexity of websites, backhaul links in WCNs are often driven into the sub-packet regime, an environment where the per-flow throughput is less than one packet per RTT. For example, it is typical for web pages to contain content loaded from several different servers. Standard browsers have evolved to optimise for opening multiple TCP connections to each web server yielding many competing TCP connections to download one page. Users of high bandwidth connections do not notice any negative effects. However, when congestion occurs, users in low bandwidth environments experience TCP and other common BE congestion control protocols break down in the sub-packet regime, resulting in:

- Severe *unfairness* over the short time scale;
- *High packet loss rates* that trigger a reduction in window sizes, and
- *Repetitive timeouts* that force a significant fraction of flows to observe long silence periods with no packet transmissions. Repetitive timeouts are triggered by the loss of retransmitted packets [64][65][66].

This topic has only recently attracted the research community, after findings showed that the performance of TCP seriously degrades in the sub-packet regime, since existing congestion control schemes, such as TCP NewReno, assume the fair-share bandwidth of a flow is at least 1 packet per RTT [67]. In addition, none of the standard TCP variants or known queuing mechanisms offer substantial performance gains in the sub-packet regime.

The sub-packet regime is defined as an environment where the per-flow throughput is less than 1 packet per RTT. A flow with segment size S and round-trip time of RTT is in the sub-packet regime if both of the following conditions hold at the bottleneck link, which has capacity C:

1. Number of competing flows, $N \gg 1$, and
2. Per-flow fair share less than S/RTT.

The fair share of a flow on a bottleneck link is inversely proportional to its RTT and is also dependent on the RTT and the number of competing flows on that link. If all flows have the same RTT, then the fair share is C/N. The first condition alone holds in the aggregate links that most Internet traffic goes through, but per-flow fair share on these high-bandwidth links is usually large enough to keep TCP out of the sub-packet regime. The second condition alone holds when one or a few TCP connections use a low-bandwidth link, such as a WiFi or cellular link, but the minimal sharing keeps these links out of the sub-packet regime. It is the combination of heavy sharing, of the order of several tens to hundreds or thousands of competing flows, operating over low-bandwidth networks that causes the sub-packet regime.

The sub-packet regime has not been a traditionally important region of operation for network flows, and as a result this space has remained relatively unexplored. The concept of a sub-packet regime arises in prior work in the context of understanding the behaviour of TCP in the face of many competing flows. The authors of [68] propose an analytical model to characterise the equilibrium behaviour of TCP in the sub-packet regime. The model is a simpler variant of a full Markov model of TCP operating in traditional regimes [69], but gives more careful attention to modelling repetitive timeouts, an extremely common state experienced by TCP flows in sub-packet regimes. As an evolution of this work, the same authors propose Time-out Aware Queuing (TAQ), a readily deployable in-network middlebox approach that uses a multi-level adaptive priority queuing algorithm to reduce the probability of timeouts, improve fairness and performance predictability in sub-packet regimes [70].

2.4. Low Extra Delay Background Transport access method

In this subsection, we provide an overview of the LCD-Net Internet paradigm that aims to support several LBE access methods and focus on one of the most popular LBE access methods, namely LEDBAT, that is already implemented and deployed in BitTorrent P2P clients. LCD-Net [71] is a new Internet paradigm to architect or bring together multi-layer resource pooling Internet technologies to support new low-cost Internet access methods. LCD-Net proposes to use resource pooling at many levels such as:

- Efficient wireless spectrum use;
- More efficient network use through the exploitation of caches and multicast, which can reduce both the transmission of redundant traffic and the average transmission cost per service access, and
- Exploitation of the available unused capacity in broadband networks by mandating LBE access to these resources.

These technologies currently exist in a variety of forms such as standards, successful deployments, in research and development phase etc. LCD-Net brings together several of these Internet technologies to ensure that donors who share their resources are not affected and at the same time are incentivised for sharing their resources, thereby creating new access methods that could potentially solve a problem of digital exclusion. Examples of LCD-Net concepts include home users that freely share their Internet connection and network operators that distribute the unused capacity. Within the framework of LCD-Net, several protocols and mechanisms that achieve LBE services have been proposed.

In general, LBE or scavenger access methods are designed to have a smaller bandwidth and/or delay impact on standard TCP flows than standard TCP flows themselves, when they share a bottleneck with it. LBE solutions have been proposed on both transport and network layers; the authors of [72] provide a survey of LBE access protocols and congestion control mechanisms. The challenges that a LBE access method faces at various network layers are described in [73].

LEDBAT is one of the most popular scavenger transport methods, being already an experimental RFC [74] by the IETF. LEDBAT is a delay-based congestion control mechanism that seeks to utilise the available bandwidth on a path while limiting the consequent increase in queuing delay on the path; to limit congestion LEDBAT exploits knowledge on the changes in one-way delay (OWD) measurements that the flow itself induces in the network. Delay-based congestion control protocols, such as TCP Vegas, are generally designed to achieve more, not less throughput than standard TCP, and often outperform TCP under particular network settings. In contrast, LEDBAT is a scavenger congestion control mechanism that seeks to utilise all available bandwidth and aims at interfering as little as possible with standard TCP flows, by being less aggressive than TCP when probing for additional bandwidth, and trying to keep a small standing queue. LEDBAT has been designed to be used either as part of an existing transport protocol with the appropriate extensions or as an application-layer mechanism running on top of User Datagram Protocol (UDP); the latter approach is followed by uTorrent Transport Protocol (uTP) [75] as long as the data transmission mechanisms are capable of carrying timestamps and acknowledging data frequently. In general, LEDBAT has been designed for use by background bulk-data applications, such as software updates, file sharing applications and P2P file transfers, since standard TCP may be too aggressive for use with such background applications.

LEDBAT congestion control interprets the increase in the queuing delay at the bottleneck link as an early signal of congestion, thus responding to congestion earlier than standard TCP and yielding bandwidth to a competing TCP flow. When the estimated queuing delay is less than a predetermined target, LEDBAT flows infer that the network is not yet congested and increase their sending rate to utilise any spare capacity in the network. When the estimated queuing delay becomes greater than the predetermined target, LEDBAT flows decrease their sending rate as a response to potential congestion in the network. To ensure no impact on competing non-LEDBAT flows, LEDBAT flows decrease their sending rate at least as quickly as competing flows increase their sending rate; a higher decrease can also be applied.

The congestion algorithm of LEDBAT is illustrated in (2.1), (2.2) and (2.3). First, the current queuing delay (*queuing_delay*) is estimated by subtracting the minimum delay (*base_delay*) from the measured OWD (*current_delay*) in (2.1). Then, distance $\Delta(t)$ from a predefined target queuing delay (*TARGET*) is calculated in (2.2). If there is no packet loss, the congestion window is recomputed based on the upper part of (2.3), where α is the increase/decrease factor. If there is a packet loss, LEDBAT performs like TCP by halving the congestion window.

$$\text{queuing\_delay} = \textit{current\_delay} - \textit{base\_delay} \qquad (2.1)$$

$$\Delta(t) = queuing\_delay - TARGET \quad (2.2)$$

$$cwnd(t+1) = \begin{cases} cwnd(t) + \alpha \dfrac{TARGET\text{-}\Delta(t)}{TARGET} \dfrac{1}{cwnd(t)}, \text{ if no loss} \\ \dfrac{1}{2} cwnd(t), \text{ if loss} \end{cases} \quad (2.3)$$

LEDBAT has been evaluated in a variety of simulation scenarios, including large bandwidth delay product scenarios [76], as well as real implementations [77]. The performance of LEDBAT in high-speed access networks, the tuning of its parameters, its interaction with TCP flows and the impact of route changes on throughput and fairness have been investigated [78][79][80][81][82]. Different decrease schemes for LEDBAT have been presented in [83]. Extensive evaluations showed that LEDBAT presents a few malfunctions [84][85][86] with the most important being the so-called "late-comer advantage", where a second, newly starting LEDBAT flow can starve the first, already running one. The last flow arriving at the bottleneck is more aggressive due to a wrong estimation of the base delay and finally takes over all resources.

Several solutions have been proposed to solve this problem, with the most prominent being fLEDBAT, a modification to the LEDBAT algorithm that introduces multiplicative decrease of the congestion window continuously driven by the estimated distance from TARGET [87]. The multiplicative window reduction response to congestion allows the source sending rate to slow down enough to make a stable and fair point always reachable. Clearly, to guarantee at the same time fairness and protocol efficiency, a proper choice of the decrease factor has to be made, in order to prevent significant and unnecessary drops in the congestion window. Moreover, an additive increase according to a constant factor α is introduced, as in TCP Reno.

The congestion algorithm of fLEDBAT is illustrated in (2.4). If Δ(t) is negative or zero, the measured delay has not yet surpassed the target delay, thus there is room for increase and the congestion window is additively increased (α is the increase factor). If Δ(t) is positive, the measured OWD is already larger than target delay, thus the congestion window needs to be multiplicatively reduced (ζ is the decrease factor). If there is a packet loss, fLEDBAT performs like TCP by halving the congestion window.

$$cwnd(t+1) = \begin{cases} cwnd(t) + \alpha \dfrac{1}{cwnd(t)}, \text{ if no loss and } \Delta(t) \le 0 \\ cwnd(t) + \alpha \dfrac{1}{cwnd(t)} - \dfrac{\zeta}{TARGET} \Delta(t), \text{ if no loss and } \Delta(t) > 0 \\ \dfrac{1}{2} cwnd(t), \text{ if loss} \end{cases} \quad (2.4)$$

Evaluation results have showed that fLEDBAT not only solves the fairness issue for backlogged flows, but also maintains the same good properties of LEDBAT, i.e. it yields to TCP while exploiting the spare bandwidth.

As described in the beginning of this chapter, one part of our research focuses on addressing the challenge of heavily shared low-bandwidth backhaul links in developing regions. In particular, we aim to address this problem by enabling flows to use scavenger transport methods instead of the traditional TCP access methods. Our intention is to explore the feasibility of using such scavenger access methods for uploading content over bandwidth constrained backhauls. Given the conservative approach of LBE access methods, we claim that they can achieve higher link efficiency and fairness than TCP in such extreme conditions. By being less aggressive and transmitting less data, we expect LBE access methods to timeout less often than TCP. Along these lines, in Chapter 6 we investigate the performance of LEDBAT and its fair modification fLEBDAT in the sub-packet regime of shared backhaul links of WCNs in emerging regions [88]. All work on LEDBAT so far has been focused on scenarios where the network is assumed to have sufficiently large capacity and is never driven into the sub-packet regime, where a low-bandwidth link is heavily shared. To the best of our knowledge, this is the first time LEDBAT and fLEDBAT are evaluated in the sub-packet regime.

2.5. Cellular data offloading

The second part of our research focuses on the typical cellular networks in developed regions that cannot handle the constantly growing cellular data traffic; due to the recent proliferation of smartphones and tablets, mobile data traffic has been explosively growing and pushing the capacity limits of cellular networks. With the global mobile data traffic forecasted to increase 13 times until 2017 [89], cellular network providers need to find cost-effective ways to handle the growing traffic demands with expected QoS levels. In highly congested wireless networks, many operators already perform de-prioritisation of heavy users for congestion management [90]. As an alternative, complementary network technologies are used for delivering data originally targeted for cellular networks. End-users perform data offloading for data service cost control and the availability of higher bandwidth, while operators perform offloading to ease congestion of cellular networks. The concept of offloading cellular networks has recently gained increased popularity due to the relevant advances in certain fields; fast switching or concurrent exploitation of multiple wireless interfaces allows users to transmit data over WiFi whenever available, while user mobility and connectivity prediction allows users to estimate their probability to encounter a WiFi AP in the near-future.

In particular, the Application Programming Interfaces (APIs) of mobile operating systems today enable seamless transition from 3G to WiFi networks that is transparent to the user. For example, MultiNets [91] is a client-based solution that provides real-time switching of wireless interfaces on mobile devices, i.e. activating a new network interface and deactivating the current one without interrupting existing connections. MultiNets provides three interface selection policies: energy saving, data offloading and performance. MultiNets does not require any changes to the network protocols and enables existing applications to run transparently without any modification. Nowadays, switching of wireless interfaces is not only fast, but also energy-efficient. Stable and Adaptive Link Selection Algorithm (SALSA) [92] is another control algorithm that mainly targets to minimise the total energy expenditure. SALSA guarantees to achieve near-optimal power consumption, while keeping the average queue finite. Another context-sensitive energy-efficient framework for wireless data transfers is presented in [93]. The authors propose to leverage the complementary strength of WiFi and cellular networks by choosing wireless interfaces for data transfers based on network condition estimation; the selection of wireless interfaces is formulated as a statistical decision problem. The use of different context information including time, history, cellular network conditions and device motion are considered. WiFisense [94] is another system that employs user mobility information retrieved from low-power sensors, e.g. accelerometer, in smartphones and further includes adaptive WiFi sensing algorithms to conserve battery power while improving WiFi usage. Similar solutions have also been proposed to enable switching between WiFi and Bluetooth [95]. The joint utilisation of the existing wireless interfaces further facilitates the development of advanced techniques to boost the performance of wireless networks and enhance the experience of mobile users. The authors of [96] propose a solution that exploits multiple wireless interfaces; to evaluate its performance the authors deploy an experimental mobile cooperative video distribution testbed. Nesto [97] is another network selection system for Android mobile devices that is designed to support two primary connectivity modes: a traditional single connectivity mode and a full dual mode, where both the cellular and ad hoc WiFi networks are used simultaneously. *We expect CEMMO mechanism, described in Chapter 3, to benefit from these solutions.*

In the field of user mobility and connectivity, several models and approaches have been proposed. For example, the authors of [98] evaluate mobility predictors with real user mobility data to explore the predictability of the time of user mobility in the context of mobile handoffs. In particular, a series of predictors based on Markov models that reflect possible dependencies across time and space are evaluated. However, a sequence of cell associations rather than a record of client position and velocity is considered as movement, restricting the accuracy of the model. Along the same lines, NextPlace [99] is an approach based on the use of non-linear time series for the prediction of the future location of users, as well as their arrival and residence time. This work focuses on the predictability of users to visit their most important places, rather than on the transitions between different locations, similar to [100]. In [101], a system that automatically clusters Global Positioning System (GPS) data taken over an extended period of time is translated into meaningful locations at multiple scales. To achieve that, the authors apply a first-order Markov model to predict user transitions between significant places. Breadcrumbs [102] is a comparative evaluation of different mobility predictors. In particular, Breadcrumbs system tracks the movement of the device's owner and customises a predictive mobility model for that specific user. Combined with past observations of wireless network capabilities, Breadcrumbs generates connectivity forecasts. The authors conclude that sufficiently accurate predictions can be achieved by applying first- or second-order Markov models. Several solutions in the area of mobility prediction are also based on social contacts [103]. To achieve accurate positioning, several alternatives to energy-intensive GPS have been proposed. The Cell-ID Aided Positioning System (CAPS) [104] leverages mobility and the position history of a user to achieve significantly better accuracy than the cell tower-based approach, while keeping energy overhead low. CAPS is designed based on the insight that users exhibit consistency in routes traveled, and that cell-ID transition points that the user experiences can uniquely identify position on a frequently traveled route. To this end, CAPS uses a cell-ID *sequence matching* technique to estimate current position based on the history of cell-ID and GPS position sequences that match the current cell-ID sequence. The authors of [105] identify the possibility of using electronic compasses and accelerometers in mobile phones, as a simple and scalable method of localisation without war-driving (i.e. the collection of information on the availability and status of WiFi networks by a person in a vehicle using a mobile device). The idea is similar to ship or air navigation systems, known for centuries. Nonetheless, directly applying the idea to human-scale environments is non-trivial, since noisy phone sensors and complicated human movements present practical research challenges. The authors cope with these challenges by recording a person's *walking patterns*, and matching it against possible *path signatures* generated from a local electronic map. Electronic maps enable greater coverage, while eliminating the reliance on WiFi infrastructure and expensive war-driving.

In Chapter 3, we develop a mobility and connectivity prediction model that is used by CEMMO mechanism; our model follows a similar approach to Breadcrumbs. In particular, user mobility and the existence of WiFi connectivity is modelled within each region during any time interval using first-order Markov models. Markov models are ideal for mobile devices, since their Central Processing Unit (CPU) and storage needs are low; Markov models only involve reading and writing individual entries in arrays. Our mobility model also considers temporal accuracy. In order to capture temporal user behaviour, we model user mobility separately during small time intervals.

Following our overview on the enabling technologies for cellular data offloading, we focus on the existing offloading solutions that can be divided into three main types based on their requirements: *OTSO*, *DTO* and *data offloading through opportunistic communications*.

2.5.1. On-the-spot offloading

OTSO refers to the immediate exploitation of an alternative network to cellular network, such as WiFi, whenever available. OTSO is the first offloading solution that has been implemented due to its simplicity. Mobile users prefer to offload data through WiFi networks, since typical WiFi data rates are significantly higher than cellular networks. From the service provider's perspective, WiFi is attractive because it allows data traffic to be shifted from expensive licensed bands to free unlicensed bands. In particular, mobile data offloading has been studied over both 3G [106] and Long Term Evolution (LTE) [107] networks and several feasibility studies in urban areas have been conducted [108]. A collaborative WiFi-based mobile data offloading architecture that targets at improving the energy-efficiency of smartphones is described in [109]. As described in [110], operators follow three approaches to offload data traffic onto WiFi networks, depending on the level of integration between WiFi and cellular networks:

- *Network bypass or unmanaged data offloading*: User data are transparently moved onto the WiFi networks, whenever WiFi coverage exists, completely bypassing the cellular core networks for data services. Only voice services continue to be delivered via the core network.
- *Managed data offloading*: This approach is adopted by operators that do not want to lose control of their subscribers, by placing an intelligent session-aware gateway through which the subscriber's WiFi session traverses on its way to the Internet.
- *Integrated data offloading*: This approach provides the operator with full control over subscribers, as well as the ability to deliver any subscribed content, while users are on the WiFi network. This is achieved by the integration of cellular and WiFi networks so that a bridge can be formed between the two networks through which data flow can be established.

The concept of data offloading through femtocells has also been proposed to alleviate the cellular burden by using an alternative access network. The authors of [111] designed a multiple femtocell offloading scheme without modifying the existing core network systems and femtocell base stations. In the proposed environment, the offloading scheme can be a cost-effective way to minimise core network investment by balancing the network load and lowering network congestion. With no impact on the commercial core network nodes, the offloading scheme does not only apply to multiple mobile home networks, but also optimises core network resources while rapidly increasing mobile user data. The main drawback of data offloading through femtocells is that it contends for limited licensed spectrum with cellular networks. The economics of mobile data offloading through third-party WiFi or femtocell access points have also been studied [112].

2.5.2. Delay-tolerant offloading

The notion of DTO has also been introduced as a method to achieve increased data offloading. In this type of cellular data offloading, users do not directly offload data to another network; instead users are willing to wait for a certain period of time in an effort to encounter an offloading opportunity later. Delay-tolerant applications can greatly benefit from DTO. Usually operators provide incentives to users, such as lower pricing, in order to motivate their participation in DTO. The authors of [113] present a quantitative study on the viability of delay-tolerant traffic offloading through WiFi APs. Their results conclude that operators can significantly increase the amount of data offloaded from cellular networks by providing the right incentives to users. As delay increases, user dissatisfaction decreases; the tradeoff between the amount of traffic being offloaded and user dissatisfaction has been studied. A novel incentive framework to motivate users to leverage their delay tolerance for cellular traffic offloading has been presented [114], while the authors of [115] followed a cooperative game theory approach to identify the ideal incentive structure.

Delayed transfer, especially with long delay deadlines, is likely to enable traffic dispersion over time in an effort to shift the high daytime demand for networking resources to the nighttime. In [116], Balasubramanian et al. introduce *Wiffler*, a system that enables fast switching from 3G to WiFi and utilises predictions about node connectivity through WiFi hotspots in order to perform offloading of data that tolerate delays up to 100 s. Wiffler only uses these predictions if delaying reduces 3G usage and the transfers can be completed within the application's tolerance threshold. In case of delay or loss sensitive applications, Wiffler quickly switches to 3G, if WiFi is unable to transmit the packet within a time window. The problem of WiFi offloading with delay-tolerant applications is formulated as a finite-horizon sequential decision problem, where the objective is to minimise the total cellular usage plus the potential penalty for deadline violation, in [117]. The optimal placement of the WiFi APs in order to achieve increased offloading has also been studied in [118]. The number of APs varies for different requirements of QoS for data delivery in large metropolitan areas.

2.5.3. Data offloading through opportunistic communications

Mobile data offloading through opportunistic communications in the context of disseminating popular content has also been introduced. According to this approach, only a small subset of users receives some popular data through the available 3G network; afterwards, these users deliver the data to a larger set of users who are interested in the data by exploiting opportunistic communications and social interactions among users. The main objective of this technique is to define an optimal subset of users, who receive the data directly from the infrastructure, in order to maximise the total number of interested users who receive this content through opportunistic P2P communications. One of the first concepts is 7DS [119], a P2P dissemination and sharing system for mobile devices, aiming at increasing the communication availability of users with intermittent connectivity. Unified Cellular and Ad Hoc Network (UCAN) [120] architecture was later proposed to enhance cell throughput while maintaining fairness. In UCAN, a mobile client has both 3G interface and the Institute of Electrical and Electronics Engineers (IEEE) 802.11-based P2P links; the 3G base station forwards packets for destination clients with poor channel quality to proxy clients with better channel quality. The proxy clients then use an ad hoc network composed of other mobile clients and 802.11 wireless links to forward the packets to the appropriate destination, thereby improving cell throughput. The authors of [121] establish a mathematical framework to study the problem of multiple-type mobile data offloading under the realistic assumptions that mobile data are heterogeneous in terms of size and lifetime, mobile users have different data subscribing interests and the storages of offloading helpers are limited. Similarly, the authors of [122] investigate the target-set selection problem for information delivery as a case study and propose three algorithms: Greedy, Heuristic and Random. TOMP [123] is the first opportunistic traffic offloading system that uses movement predictions of mobile users to analyse the prospective inter-device connectivity; the authors propose three different metrics for analysing movement predictions, as well as an algorithm that uses these metrics to utilise an efficient opportunistic traffic offloading scheme. The authors of [124] extend the concept of opportunistic data offloading by introducing acknowledgements as feedback on data diffusion; based on this information, the source node adjusts the number of replicas that are propagated to users. The proposed Push-and-Track content dissemination framework is flexible and can be applied to a variety of scenarios, including periodic message flooding.

Unlike the previous formulation of mobile data offloading as a target-set selection problem, which essentially asks the question "who will download the content through the cellular link", the authors of [125] ask "who" and "when". In particular, they present methods for individual users to locally estimate their own and their acquaintances' topological importance on the opportunistic proximity-link-based networks and aggregated interests in content. These factors are consolidated into a time-dependent function that embodies the concept of user patience for the content.

In contrast to previous work on offloading popular content only through opportunistic networks, the dissemination of popular data through frequently visited public WiFi hotspots is presented in [126]; users retain this content in cache and participate in the data propagation through opportunistic device-to-device (D2D) communications. In particular, the authors design two algorithms for DTO of bulky, socially recommended content from 3G networks. The first one, called "MixZones", uses opportunistic, ad hoc transfers between users, and is assisted by predictions made by the network operator. The second one, called "HotZones", exploits delay tolerance, tries to download content when users are close to WiFi access points and is also assisted by predictions made by the operator. Along these lines, the authors of [127] present a detailed evaluation of procedures that exploit mobility prediction and prefetching to enhance offloading of traffic from mobile networks to WiFi hotspots, for both delay-tolerant and delay-sensitive traffic. Overall, work on opportunistic P2P offloading so far only supports offloading of data with popular content (e.g. music/video sharing, electronic newspapers, etc.). Consequently, their offloading success strongly depends on data popularity.

In Chapter 3, we present CEMMO, the only offloading mechanism that enhances OTSO and DTO with a multi-hop peer-assisted offloading mode, where the offloaded traffic is delivered through intermediate mobile devices. To the best of our knowledge, the proposed PAO is the first peer-assisted offloading solution that is implemented not only for offloading data from the uplink, but also independent of its content or popularity.

2.6. DTN use cases, applications and routing

DTO and data offloading through opportunistic communications described in the previous subsection constitute only a small part of the services that the DTN architecture has enabled. In particular, projects such as Haggle [128] and SCAMPI [129] have explored ways to offer communication and computing services on top of mobile opportunistic networks. On the one hand, Haggle diverges from the DTN stack structure but enables the automated propagation of information across concatenated time-disjoint communication links in ways that focus on specific application constraints. On the other hand, SCAMPI has chosen to retain the DTN bundle protocol, but adds infrastructure that is particularly suitable for mobile opportunistic communications.

DTN has also been tested in numerous stressed terrestrial environments, such as sensors in underwater environments [130][131]. One of the first projects on the field was the Sensor Network with Delay Tolerance (SeNDT) project [132], a Trinity College Dublin project running real-world pilots using a sensor node for environmental monitoring designed for public authorities, Non-Governmental Organisations (NGOs) and/or corporates. SeNDT applied DTN technology to fill a niche for sensor nodes that cannot use more typical networks (e.g. those assuming IP or Global System for Mobile communications/Short Message Service (GSM/SMS) connectivity) and the pilots involved lake water quality monitoring and road-side noise monitoring. Similarly, the Environmental Monitoring in Metropolitan Areas (EMMA) [133] project developed a decentralised and cost-efficient architecture for area-wide measurement of air pollutants. Vehicles of existing public transportation systems were used to continuously acquire environmental data. The measured values were exchanged between different vehicles with the help of WLAN. Since vehicles only meet each other sporadically, techniques from the fields of Vehicle-to-Vehicle (V2V) communications and DTNs were used for the data exchange. The measured values were then delivered to a central gateway that forwarded the messages to the evaluation server. The results of several measurements in the city of Braunschweig, Germany, showed that the concept of exchanging information between vehicles works very well. Communication is possible even without direct line-of-sight (LoS) and at higher distances between vehicles. As an evolution of this work, the Opportunistic Public Transport Communication (OPTraCom) [134] project studied routing and data dissemination in this network through real-world experiments. The Consumer-generated Mobile Wireless Media (CROWD) [135] project aimed to study PUB/SUB based exchanges of user-generated multimedia information within the framework of a Web 2.0 online portal, while users are on the move and can communicate opportunistically. The ultimate goal of this project was to unleash interactivity and make users capable of collecting and publishing information with the help of (or within) the communities they belong to, and with the help of a new generation of online services that use opportunistic communications.

Throwboxes, i.e. small and inexpensive stationary devices equipped with wireless interfaces and storage that act as relays between nodes, were also introduced to improve data delivery in DTNs [136]. Evolving from this concept, ferries were proposed for data collection from sensors [137] and, later, for data transmission in DTNs. One of the key DTN use cases is the provision of delay-tolerant Internet connectivity in remote villages that have no access to the backbone Internet infrastructure [138][139]. KioskNet [140] was one of the first proposals on the field with a main goal to employ buses and cars as “mechanical backhaul” devices to carry data to and from a remote village and an Internet gateway. This ‘mechanical backhaul’ avoids the cost of trenches, towers, and satellite dishes, allowing Internet access even in remote areas. In areas where dialup, long-range wireless or cellular phone service is available, the kiosk controller can be configured to use these communication links in conjunction with the mechanical backhaul. The Networking for Communications Challenged

Communities (N4C) [141] project also developed solutions for basic Internet access in remote regions. The project covered the whole spectrum from theory to applications and most importantly experimented with developments in the field using “testbeds” with real users.

The concept of exploiting data mules or message ferries for transfers among DTN nodes has not only been applied in sparse mobile networks [142], but also dense urban environments [143]. The authors of [144] studied how messages could be carried over a realistic large-scale global network between airports based upon scheduled flight connections. In particular, they investigated the interaction with different routing protocols, the impact of scheduling uncertainties, and the limiting factors by means of simulations and analysis. Along the same lines, TrainNet [145] is a vehicular network that uses trains to transport latency insensitive data. TrainNet augments a railway network by equipping stations and trains with mass storage devices; e.g., a rack of portable hard disks. TrainNet has two applications. First, it provides a low-cost, very high bandwidth link that can be used to deliver non real-time data. In particular, cable television (TV) operators can use TrainNet to meet the high bandwidth requirement associated with Video-on-Demand (VoD) services. TrainNet is able to meet this requirement easily because its links are scalable, meaning that their capacity can be increased inexpensively due to the continual fall of hard disk price. Secondly, TrainNet provides an alternative, economically viable, broadband solution to rural regions that are reachable via a railway network. Therefore, using TrainNet, rural communities are able to gain access to bandwidth intensive digital contents such as music, video, television programs, and movies cheaply. Similarly, MetroNet [146] exploits the precise schedules of metro systems to pre-fetch data and facilitate mobile Internet downloads, even if an E2E path between the source (i.e. content source) and the destination (i.e. user) does not exist contemporaneously. Methods to improve communication management in ferry-assisted DTNs have also been proposed [147].

In Chapter 4, we develop and present a DTN-based architecture for public transport networks, equipped with smart delivery mechanisms that utilises every contact opportunity to provide delay-tolerant Internet access to users. The proposed architecture can exploit a variety of Internet applications that have been adapted to run over DTNs [148], including:

- *Delay-tolerant web search from a bus* [149]: Thedu system was developed to adapt the interactive process of web search and retrieval to vehicular networks with intermittent Internet access. Thedu mobile nodes use an Internet proxy to collect search engine results and prefetch result pages. The mobiles nodes download the pre-fetched web pages from the proxy.
- *Opportunistic web access via WLAN hotspots* [150]: The DTN-enabled web server is a server that accepts bundles containing Hypertext Transfer Protocol (HTTP) requests and returns responses of bundled resources; it also supports plain HTTP access. The

server obtains the resources to be bundled up in a specific response either from a dependency file stored on the server, which might be generated by a web authoring tool, or parses the requested resource, if it is HTML, and determines the other resources to be included. A separate proxy is provided to allow arbitrary web servers to access the DTN web server using bundles.

- *DTN-enabled blogging* [151]: The application provides a proxy for a mobile device which is contacted by the mobile user's web browser. The application encapsulates blog-postings with optional attachments in DTN bundles, transfers them to a backend which unpacks them and posts them to a regular blogging site.
- *DTN E-mail* [152]: A set of e-mail applications was developed in support of DTN-based communications. This includes a personal mail proxy connecting existing mail clients, such as Thunderbird [153], via the Simple Mail Transfer Protocol (SMTP) and the Post Office Protocol 3 (POP3), a simple native DTN mail client for the Nokia Internet tablet, and an infrastructure mail gateway to translate between traditional SMTP and DTN-based mail traffic.
- *fbDTN* [154]: fb-DTN is a gateway that allows users to access Facebook services over a DTN in a convenient and secure manner. The system enables users to read their news feed, post status updates and photos, and comment and *like* the posts of other people.
- *Twitter* [155]: This proof-of-concept application uses Twitter social network to demonstrate the feasibility behind DTN principles using the DTN2 implementation.
- *DTN-based voice communications* [156]: The DT-Talkie is a push-to-talk-style application for the Nokia N800 [157] and N810 Internet tablets. It allows holding delay-tolerant conversations over a DTN network, e.g. using a WLAN, by encapsulating talk spurts into bundles and exchanging them between peers. Simple conversations can make use of the hardware buttons of the N810; a Graphical User Interface (GUI) allows managing contacts and maintaining multiple conversation contexts.

A survey of delay-tolerant networking applications is provided in [158].

Numerous new concepts were designed leveraging the DTN paradigm at their core. One of the most popular is Floating Content [159], where a message, such as a text message or an image, deemed to be of interest to other people at a certain area is tagged with geographical coordinates of that area. This area is referred to as the anchor-zone of the message. The message is disseminated in opportunistic manner whenever two nodes meet within the anchor-zone. Outside the anchor-zone, nodes are free to delete it. The unique characteristics are that the scheme does not rely on any infrastructure, a message can only be created and distributed locally, and it cannot be deleted afterwards. Therefore, assuming there is a sufficient number of nodes in the anchor-zone, a message remains available within the anchor-zone until it expires. *The D2D forwarding mechanism presented in Chapter 3 is partially inspired by the idea of floating content and relies on the notion of data storage in a specific region for a restricted time interval.*

One of the most challenging DTN topics that attracted the interest of the research community is routing in disconnected environments. Several DTN routing protocols have been proposed, the majority of which differ in two main aspects: *the amount of available information to make the routing decisions* and *the number of copies to create per bundle*.

Epidemic routing [160] was one of the first proposals in this area; Epidemic routing relies upon carriers that come into contact with another portion of the network through node mobility and spread the messages they hold without making any assumptions regarding the topology and connectivity of the underlying network. More specifically, Epidemic employs pair-wise exchanges of messages among all nodes that connect to each other. This transitive transmission of data leads to a high probability of the messages eventually reaching their destination, since all nodes will eventually see all replicas in a bounded amount of time (i.e., the system will reach eventual consistency). The overall goal of Epidemic Routing is to maximise message delivery rate and minimise message delivery latency, while also minimising the aggregate system resources consumed in message delivery. This can be accomplished by placing an upper bound on message hop count and per-node buffer space. Overall, Epidemic routing is able to deliver nearly all messages in scenarios where existing ad hoc routing protocols fail to delivery any messages because of limited node connectivity. Its main disadvantage is the extreme overhead it creates, which leads to high contention for buffer space and bandwidth.

Extending the idea of Epidemic routing, the authors of [161] proposed a Probabilistic ROuting Protocol using History of Encounters and Transitivity (PRoPHET) that uses the history of previous encounters and the transitive property inherent in human contacts to estimate a delivery probability, kept in a table, at each node for all known destinations. As nodes meet, they exchange their delivery probability tables and then update their own tables based on the identity of the other node and the received table, using simple mathematical equations. The authors have chosen a rather simple forwarding strategy – when two nodes meet, a message is transferred to the other node if the delivery predictability of the destination of the message is higher through the other node. Minor modifications to the routing metric calculations of PRoPHET have led to an updated version of the protocol, called PRoPHETv2 [162]. A distance-based modified version (DiPRoPHET) [163], that introduces the concept of cross-layer implementation to retrieve the distance value from the lower layer for use in the upper layers in a DTN, has also been proposed. PRoPHET has motivated the introduction of a variety of social-based DTN routing protocols. For example, Bubble [164] exploits two social and structural metrics, namely *centrality* and *community*, using real human mobility traces to enhance delivery performance.

In an effort to reduce the transmission overhead while keeping delivery probability high, the authors of [165] proposed Spray-and-Wait routing protocol; Spray-and-Wait bounds the total number of copies and transmissions per message without compromising performance. Spray-and-Wait routing consists of the following two phases:

- *Spray phase*: For every message originating at a source node, L message copies are initially spread (i.e. forwarded by the source and possibly other nodes receiving a copy) to L distinct "relays". Any node A that has n > 1 message copies (source or relay), and encounters another node B with no copies, hands over to B [n/2] messages and keeps [n/2] messages for itself. When a node is left with only one copy, it switches to direct transmission.
- *Wait phase*: If the destination is not found in the spraying phase, each of the L nodes carrying a message copy performs direct transmission, i.e. forwards the message only to its destination.

Despite its simplicity, Spray-and-Wait achieves comparable delays to an optimal scheme, and is very scalable as the size of the network or connectivity level increase. The main disadvantage of Spray-and-Wait is that it spreads all its copies quickly to the node's immediate neighbourhood, but then few, if any, of the nodes carrying a copy might ever see the destination.

Spray-and-Focus [166] solves this problem by modifying the second phase, now called "focus" phase. Rather than waiting for the destination to be encountered, each relay can forward its copy to a potentially more appropriate relay, using a carefully designed utility-based scheme. Spray-and-Focus can successfully recognise and take advantage of potential opportunities to forward a message "closer" to its destination. As a result, it can achieve very good performance also in situations where existing spraying schemes may suffer significantly.

As far as vehicular DTNs are concerned, MaxProp routing [167] is one of the most promising solutions based on prioritising the schedules of packets to be transmitted and to be dropped. In particular, MaxProp uses a ranked list of the peer's stored packets based on a cost assigned to each destination. These priorities are built on path likelihood to peers according to historical data and some enhancement mechanisms, including acknowledgments, a head-start for new packets and lists of previous intermediaries. Packets that are ranked with highest priority are the first to be transmitted during a transfer opportunity, while packets ranked with lowest priority are the first to be deleted to make room for an incoming packet. When two packets have destinations with the same cost, the packet that has traveled fewer hops is given higher priority. MaxProp also uses hop counts in packets as a measure of network resource fairness.

Geographic vehicular routing solutions, that exploit location information and other mobility parameters provided by positioning devices to make routing decisions, have also been proposed. These protocols aim to reduce the geographic distance to the destination node at each step and are intended to be used on sparse vehicular scenarios where communication opportunities are based on sporadic and intermittent contacts. The Geographical Opportunistic Routing for Vehicular Networks (GeOpps) [168] is a geographical delay tolerant routing algorithm that exploits information from the vehicles' navigation system to route messages to a specific location. To achieve that, neighbour vehicles that follow suggested routes to their driver's destination calculate the nearest point that they will get to the destination of the packet. Afterwards, they use the nearest point and their map in a utility function that expresses the minimum estimated time that this packet would need in order to reach its destination; the vehicle that can deliver the packet closer to its destination becomes the next packet carrier. The geographic routing protocol for vehicular delay-tolerant networks (GeoSpray) [169] is inspired in the general guidelines of GeOpps and uses geographic position information and other mobility parameters, together with bundle destination addresses, making sure that bundles are forwarded towards the destination. In contrast to GeOpps that maintains at most one copy of a bundle in the network, GeoSpray combines selected replication and forwarding with explicit delivery acknowledgment. In particular, the GeoSpray routing protocol employs the concept of ''spray phase'' from binary Spray-and-Wait, where a fixed number of bundle copies are distributed to distinct nodes in the network. However, instead of doing blind replication as proposed in Spray-and-Wait, GeoSpray guarantees that bundle copies are only spread to network nodes that go closer to the bundle's destination. Furthermore, instead of waiting until one of these network nodes meets the destination and delivers its bundle copy as proposed in the Spray-and-Wait ''wait phase'', GeoSpray allows each node to forward its bundle copy further to another node that can take the data closer to the destination.

Along the same lines, the Bus Line-based Effective Routing (BLER) protocol [170] uses a route contact oracle to find a shortest path among bus lines instead of two buses. The main problem when using a contacts oracle in a vehicular environment is that the buses rarely respect the meetings schedule, due to inconstant speed and waiting times at bus stops or delays due to car accidents etc. For this reason, BLER considers paths among bus lines, instead of paths among buses. Delay Tolerant Link State Routing (DTLSR), an adapted version of Dijkstra's shortest-part algorithm that works with time and space, has also been proposed [171]. As the network state changes, link state announcements are flooded throughout the network. Each node maintains a graph representing its current view of the state of the network and uses a shortest path computation to find routes for messages. Each node in the system is assigned to an administrative area and a link state protocol instance operates only within a single area. This helps constrain the size of the network graph and limits the scope of announcement messages. However, DTLSR uses the planned schedule instead of the exact contact times and, thus, cannot handle unexpected delays and is inefficient when severe changes in the network occur. Look-Ahead Routing and Message Scheduling approach (ALARMS) [172] was introduced to deliver bundles through message ferries that only hold information regarding their path for the next rounds. ALARMS assumes a variation of the well-known ferry model, in which there are ferry nodes moving along pre-defined routes to exchange messages with the gateway node of each region on the route. Each ferry also passes to the gateway nodes look-ahead routing information about when it will arrive at each gateway node on the route in the next two rounds and how long it will stay. The gateway nodes use this information to estimate the delivery delay of each message when being delivered by different ferries, and schedule the messages to be delivered by the ferry that arrives earliest at the destination. This work, however, does not consider global knowledge of the network and a path to the destination can only be found if a ferry exists to directly connect the source and the destination.

Cluster or zone-based routing protocols have also been proposed. Zone Based Message Ferrying (ZBMF) [173] is a hybrid protocol for DTNs that transforms the disrupted network space into a connected network topology by dividing it into overlapping ferry routing zones and giving DTN message routing a sense of direction, thus providing both contained dissemination of copies of messages and faster delivery of messages in the network. Due to its low overhead, ZBMF can be used to cover large areas of disrupted communication, such as cities and large disaster areas. Along the same lines, the authors of [174] investigate a distributed clustering scheme and propose a cluster-based routing protocol for DTNs. The basic idea is to group mobile nodes with similar mobility pattern into a cluster, which can then interchangeably share their resources, such as buffer space, for overhead reduction and load balancing, aiming to achieve efficient and scalable routing. Due to both the lack of continuous communication among mobile nodes and the possible errors in the estimation of nodal contact probability, convergence and stability become major challenges in distributed clustering in DTNs.

The main characteristic of the majority of the aforementioned protocols is that they follow a multiple-copy routing strategy to increase delivery probability. Replication typically leads to excessive overhead and poor network performance. *In Chapter 4, we propose a single-copy DTN routing protocol, CARPOOL, and its evolution, CARPOOL+, which we couple with the proposed DTN-based architecture for public transport networks. Both protocols exploit a priori knowledge on the route of public transport vehicles to route bundles based on the* *earliest estimated delivery time, while CARPOOL+ also exploits opportunistic contacts among ferries and mitigates the impact of typical schedule delays, e.g. due to road traffic.*

3. Cost-Effective Multi-Mode Offloading with peer-assisted communications

In this chapter, we describe CEMMO, the Cost-Effective Multi-Mode Offloading mechanism, which is proposed to enhance pure on-the-spot (OTSO) and pure delay-tolerant offloading (DTO) with a multi-hop peer-assisted offloading mode (PAO) where the offloaded traffic is delivered through intermediate mobile devices. In Section 3.1, we introduce the mobility and connectivity prediction model that CEMMO utilises. Based on this model, in Section 3.2 we detail DTO and PAO transfer policies. The operation of CEMMO and the process of selecting the transfer method that minimises cost are presented in Section 3.3, while Section 3.4 includes a sample operational scenario of CEMMO. Finally, in Section 3.5 we discuss issues concerning the adoption of CEMMO by cellular operators and users.

3.1. Mobility and connectivity prediction model

Human mobility is characterised by periodicity (e.g. daily or weekly), as people usually follow recurrent schedules (e.g. go to work in the morning, return home in the afternoon, have some evening activity, etc.). In this thesis, we propose a mobility and connectivity prediction model according to which a large area is divided into smaller *regions* with unique identifiers, and a time period (e.g. a day or a week) is split into smaller *time intervals* (e.g. 10-minute intervals). The region size and the time interval duration affect the spatio-temporal accuracy of the prediction model. Defining small regions and time intervals improves the accuracy of the model at the expense of increased computational complexity. Users store their own mobility information for each time interval. For example, if an one-hour time interval is selected, users store their mobility information for 24 distinct time intervals. In particular, users keep records of their location, their transitions from one region to another and the frequency at which they move between regions.

For each user, we define:

- N(X,t) as the number of visits in region X during time interval t (i.e. the time interval that starts at time t);
- N(A→X,t) as the number of transitions from A to a neighbouring region X during t;
- $t_{duration}$ as the duration of each time interval, and
- $t_{next} = t_{duration} + t$ as the start time of the next time interval.

Considering the aforementioned definitions, the probability of a user located in region A to move to region X directly within t is:

$$P(X \mid A, t) = N(A \rightarrow X, t) / N(A, t) \quad (3.1)$$

The probability that the user stays in region X until the end of time interval t is:

$$P_{stay}(X, t_{next}) = P(X \mid X, t) = N(X \rightarrow X, t) / N(X, t) \quad (3.2)$$

User mobility during any time interval t is modelled as a Markov process. The next region that will be visited by a user during a time interval is assumed to solely depend on the previous one. Thus, the probability of a user to move from current region A to region X through any neighbouring region $i \in NR(A)$ during t is:

$$P(X|A, t) = \sum_{\forall i \in NR(A)} P(i|A, t) \times P(X|i, t) \quad (3.3)$$

where NR(A) is the set of neighbouring regions of A.

Starting from the current region ($L_{current}$) and time interval ($t_{current}$) and based on Equation 3.1 and Equation 3.3, we estimate the probabilities of a user to visit any other region during any time interval $t \leq t_{end}$, where t_{end} is the Delay Tolerance Interval (DTI), i.e. the time interval until which the user is willing to wait for the data to be offloaded through a WiFi AP. We use Equation 3.2 to compute the set of regions where the user may be located at the end of each time interval, along with the corresponding probabilities. We recursively use these regions as potential starting points of user movement during the next time interval and repeat this process to predict user mobility during each time interval. We provide a detailed description of this process in Table 1.

Each user also keeps statistics regarding WiFi connectivity within each region, since a region may only have partial WiFi connectivity. NW(X) denotes the number of times WiFi connectivity was available in region X. With the selection of small time intervals, we do not need to measure the duration of the WiFi connectivity. The probability that a user located in region X during time interval t has access to a WiFi access point is defined as:

$$P(WiFi|X) = NW(X) / N(X, t) \quad (3.4)$$

Input: *Probabilities P* as extracted from historical data, $L_{current}$, $t_{current}$, t_{end}

Output: *Probabilities* P_{est} to visit any region for each time interval before t_{end}

Initialise P_{est}(Region, Interval) ← 0 ∀ region and interval ≤ t_{end}

$P_{est}(L_{current}, t_{current})$ ← 1

Add $L_{current}$ to StartingPoints *//Initial starting point is current location*

$t_i \leftarrow t_{current}$

$t_{next} \leftarrow t_i + t_{duration}$

While $t_i \le t_{end}$ **do**

    For each SP ∈ StartingPoints **do**

        For each region R: $P(R|SP, t_i) > 0$ **do** *//For all possible transitions to another region*

        $P_{est}(R, t_i) \leftarrow P(R|SP, t_i)$ *//Based on Equation 3.1*

            For each region N: $P(N|R, t_i) > 0$ **do**

                newProb ← $P(N|R, t_i) \times P(R|SP, t_i)$ *//Based on Equation 3.3*

          $P_{est}(N, t_i)$ += newProb *//Transitions to a region through different paths*

            End *// are mutually exclusive and, therefore, probabilities are added*

        End

    End

    For each region R:($P_{est}(R, t_i) > 0$ AND $P(R, t_{next}) > 0$) **do** *//Regions to be used as*

      Add R to StartingPoints *//starting points during the next time interval*

        $P_{est}(R, t_{next}) \leftarrow P_{stay}(R, t_{next}) \times P_{est}(R, t_i)$ *//P_{stay} is based on Equation 3.2*

    End

    $t_i \leftarrow t_{next}$

    $t_{next} \leftarrow t_i + t_{duration}$

End

Table 3.1 Mobility prediction model

3.2. Delay-tolerant and peer-assisted offloading transfer policies

In the following subsections, we present DTO and PAO transfer policies based on the proposed mobility and connectivity prediction model. We assume that each node (i.e. mobile device) is equipped with both IEEE 802.11 and 3G wireless interfaces that can run simultaneously, is capable of generating content for upload and is willing to accept certain delays in applications, such as video upload, without sacrificing too much user satisfaction. For both policies, we define R as the set of regions that the source node S (i.e. the user who initiated the upload request) is likely to visit before t_{end} and U the set of nearby users. We define the following events:

$$E^{S}_{r,t_S} \rightarrow \text{source node S visits region } r \in R \text{ during time interval } t_S$$

$$E^{u}_{r,t} \rightarrow \text{user } u \in U \text{ visits region } r \in R \text{ during time interval } t$$

$$E^{u}_{WiFi,\, t_w} \rightarrow \text{user } u \in U \text{ has WiFi access during } t_w$$

3.2.1. Delay-tolerant offloading

According to this policy, mobile users attempt to transfer their data directly through a WiFi network within a specific timeframe. The success probability of DTO is defined as the probability of the source node S to have WiFi connectivity within a interval t_w:

$$P_{DTO_{success}} = P(E^{S}_{WiFi,t_w}) \quad (3.5)$$

For data upload requests, the time interval t_w, during which source node S is expected to have access to a WiFi network, needs to be prior to t_{end}. Thus:

$$t_w \leq t_{end}$$

3.2.2. Peer-assisted offloading

According to this newly proposed PAO transfer policy, other mobile users are utilised as intermediate data carriers to offload data through a WiFi AP. The main concept of our approach lies in storing data locally within a specific region for a short time interval, in a way that any mobile user who enters this region during this interval receives a replica of the data. Data storage enables the data exchange between the source node and intermediate users, who will actually perform the transfer through a WiFi AP, using ad hoc technologies such as WiFi Direct. The cellular operator divides the overall region into sub-regions and predetermines the limits of each region using GPS coordinates, selects the optimal storing region ($R_{storing}$) and storing time interval ($T_{storing}$) based on the probabilities of nearby nodes to visit any region in the near future, and notifies the source node about the GPS coordinates of the flooding region. Users can detect when they have entered the flooding region using location services such as Geofences [175].

When the source node enters the region that has been specified by the operator, replicas of the data are being flooded. Eventually, all users within the region during the storing interval receive a replica of the data. The operator can use a control channel, similar to the one used for call setup or sending text messages, to signal if the WiFi network interface of the mobile device needs to be switched on. The flooding process is regionally and temporally restricted; nodes that have received a replica of the data are only allowed to forward replicas to other users located within the storing region during the storing interval. The lifetime of the data is dependent on the storing time interval. The geographical and temporal narrowing of this flooding process limits the total number of replicas in the network and, thus, does not create excessive overhead. CEMMO provides the operator with the flexibility to adjust the region size and, therefore, control the overall overhead. Flooding content in a large region (or a small region that is highly dense) would result in high overhead, since many users would act as relays for the data. Flooding to more regions would result in excessive overhead for the operator, making the scheme infeasible.

When users holding a replica leave the storing region, they are only allowed to upload the data directly through a WiFi AP. When an intermediate data carrier successfully uploads a data replica through a WiFi AP, he/she informs the operator, who in turn sends a notification to the source node. In case another intermediate node attempts to transfer another replica, the transfer is aborted, and the node is notified to discard its replica. All other nodes drop their replicas once DTI expires. In case the source node does not receive any notification of successful upload until t_{end}, data are transferred through the available 3G network.

A user u serves an upload request successfully, if he/she manages to transfer the data between the source user and the WiFi AP (i.e. by transferring a replica from the storing region). Thus, user u is a successful relay node if the following event occurs:

$$E_{Success}^{u,r,t} \rightarrow E_{r,t}^{u} \cap E_{r,t_S}^{S} \cap E_{WiFi,\, t_w}^{u}$$

For data upload requests, the time interval t, when user u visits region r, needs to be equal to or subsequent of time interval t_s. Moreover, time interval t_w, when node u has access to a WiFi network, needs to be equal to or subsequent of time interval t. Thus:

$$t_s \leq t \leq t_w \leq t_{end}$$

The operator selects to store data in the storing region until the expiration of the storing interval in order to maximise the probability that data will be successfully transferred before DTI expires (i.e. before t_{end}). Thus:

$$\{R_{storing}, T_{storing}\} = \underset{r,t}{\operatorname{argmax}}\, P\left(\cup_{u \in U} E_{Success}^{u,r,t}\right) \quad (3.6)$$

The success probability of PAO is:

$$P_{PAOsuccess} = P\left(\cup_{u \in U} E_{Success}^{u,R_{storing},T_{storing}}\right) \quad (3.7)$$

3.3. Cost-Effective Multi-Mode Offloading mechanism

The overall assumption of CEMMO [176] is that there exists a centralised mechanism, provided by the cellular operator, that collects data transfer requests from users, performs online predictions on user mobility and connectivity and decides on the transfer policy that minimises cost. It should be noted that we are not really concerned with predicting the user's mobility perfectly. If our model predicts that a user will move to one location and the user finally moves to another, this wrong estimation does not affect the performance of CEMMO, as long as there is available network connectivity through a WiFi AP in this region. The purpose of CEMMO is to assist cellular operators increase the amount of traffic offloaded by their customers by discovering WiFi APs in their vicinity and offloading data either directly or through other peers. CEMMO helps balance the load and relieve 3G access networks from excessive usage, thus enhancing their total network capacity and meeting the increasing traffic demands.

The forwarding of content through WiFi APs may happen in different ways. We assume that users perform OTSO when they have direct access to a WiFi network. We also assume that mobile users are willing to accept a delay in the delivery of some non-urgent data, when proper incentives are offered to them, such as reduced service pricing and improved services. Web server synchronisation, such as Dropbox and Flickr, are typical mobile applications where CEMMO undertakes the responsibility to transfer the data within the specified DTI by applying one of the three data transfer policies:

i. *Delay-Tolerant Offloading (DTO)*: According to this policy, mobile users attempt to transfer their data directly through a WiFi network within a specific timeframe. CEMMO evaluates the probability of a user to have access to a WiFi AP before DTI expires, in order to decide on the optimal data transfer policy.

ii. *Peer-Assisted Offloading (PAO)*: In this case, the mobile devices of nearby users are utilised as intermediate carriers in order to accomplish the data transfer through a WiFi AP. This technique does not require direct access of the user that generates data to a WiFi network.

iii. *Transfer through the available 3G network*: If neither of the aforementioned techniques achieves data offloading within the specified DTI or CEMMO mechanism estimates that data offloading either through opportunistic P2P communications or directly through a WiFi AP is not feasible (i.e. success probability is zero) or financially efficient, data are transferred through 3G. In essence, the 3G network is used as an alternative. We assume that 3G connectivity is always available and, therefore, the success probability of this approach is one.

All mobile nodes apply the proposed mobility and connectivity prediction model and CEMMO decides on the transfer policy by evaluating information on the environment and mobility of users. Each time a mobile user needs to upload data, a request is sent to the cellular operator. This request includes:

- the data size;
- the DTI field as set by the user;
- his/her current location;
- his/her probabilities to visit any other region within DTI, and
- WiFi connectivity probabilities.

Given the small size of the data transfer requests, this process can be accomplished through the available GSM network in order to reduce the energy consumption of the mobile devices. When the cellular operator receives a request for data transfer, CEMMO exploits the provided information to estimate the success probability of DTO (Equation 3.5). In order for the operator to estimate the success probability of PAO, a signal is sent to all users who are associated with the same cell tower requesting information on their approximate position and probability to visit other regions within DTI, along with their WiFi connectivity probabilities. It is noted that a single cell tower might cover more than one regions. Based on this feedback, the operator identifies the storing region and storing interval (Equation 3.6) and estimates the success probability of PAO (Equation 3.7).

Both data transfers through 3G and WiFi networks involve a cost for the operator. In regard to 3G based data transfers, this cost may correspond to the financial cost of the transfer, the associated user dissatisfaction due to increased pricing, network congestion and increased energy consumption that data transfer through 3G incurs. As far as data transfer through WiFi networks is concerned, the cost corresponds to user dissatisfaction due to the delayed transmission, as well as a financial cost for the operator, including any reward the operator needs to pay to users in order to motivate them to delay their transmissions. Data transfer through opportunistic P2P networks incurs an additional cost, since the limited resources of each user involved in the process are exploited for the transfer of data from other users. In order to quantify the data transfer cost of each policy, we define α as the transfer cost per MB through 3G and β as the transfer cost per MB through WiFi. Since all data that are not successfully delivered over WiFi are eventually be transmitted over 3G, CEMMO calculates the estimated transfer cost of DTO and PAO based on the success probability in each case as follows:

$$\text{Est. Cost}_{DTO} = \alpha \times (1\text{-}P_{DTOsuccess}) + \beta \times P_{DTOsuccess} \quad (3.8)$$

$$\text{Est. Cost}_{PAO} = \alpha \times (1\text{-}P_{PAOsuccess}) + \beta \times N \times P_{PAOsuccess} \quad (3.9)$$

where N is the number of participants in PAO. CEMMO selects the transfer policy that minimises the estimated transfer cost. By increasing the β/α ratio, the cost of relaying data to users increases compared to the cost of data transfer through 3G. Thus, the amount of data that is offloaded through P2P communications is reduced.

Our prediction model models mobility and connectivity within each region during any time interval using first-order Markov models. Markov models are ideal for mobile devices, since their CPU and storage needs are low; they only involve reading and writing individual entries in arrays. Therefore, the complexity and computational overhead from the perspective of users is marginal. From the operator side, the required signalling overhead is limited in comparison to the amount of data that is being offloaded. Moreover, it can be further reduced if all users periodically transmit their probabilities to the operator or the operator holds all collected mobility and connectivity probabilities until each time interval expires. In this case, the operator does not need to request the same probabilities from users for consecutive requests.

3.4. Sample scenario

We consider a region covered by a cellular network, as depicted in Figure 3.1. According to our model, the operator divides the greater region into smaller regions (i.e. Regions 1-16). Users hold their own mobility and connectivity predictions based on their historical data. We consider that a user (Node 1), located in Region 7, has recorded a video of 100MB using the camera of his/her mobile device and wants to upload it to a cloud storage service within the next 30 minutes, in order to share it to his/her friends. Therefore, Node 1 sends a request to the cellular operator that includes the data size, the DTI, his/her location, his/her probabilities to visit any other region and the connectivity probabilities before DTI expires. On the reception of the request, the operator requests probabilities from the N nodes currently connected to the same cellular base station, collects them and calculates the success probabilities of both DTO and PAO. Based on the calculated success probabilities, the operator estimates the overall transfer cost of each policy. If $\text{Est.Cost}_{DTO} < \text{Est.Cost}_{PAO}$, the operator opts for DTO and informs Node 1 to hold the data until DTI expires. If Node 1 identifies an available WiFi connection prior to DTI expiration, data are offloaded. Else data are transferred over 3G when DTI expires. If $\text{Est.Cost}_{PAO} < \text{Est.Cost}_{DTO}$, the operator opts for PAO and informs Node 1 on the storing region (in this case Region 11) and storing interval. When Node 1 enters Region 11, the video is opportunistically flooded to all nodes located within Region 11 (i.e. Nodes 2, 3 and 4) using an ad hoc interface of the mobile devices, such as WiFi Direct. The operator signals all K nodes (K<<N) within Region 11 to switch on their WiFi interface during the storing interval. Node 2 moves to Region 14, connects to the available WiFi AP and uploads the video to the corresponding server. When the video is successfully uploaded, Node 2 informs the operator that, in turn, informs Node 1. All other nodes drop the video once DTI expires. In case the video has not been successfully uploaded when DTI expires, it is transferred over 3G. The maximum overall signalling overhead associated with CEMMO is not significant compared to the amount of data that is being offloaded; signals only include GPS coordinates and probabilities and, therefore, their size does not exceed a few bytes.

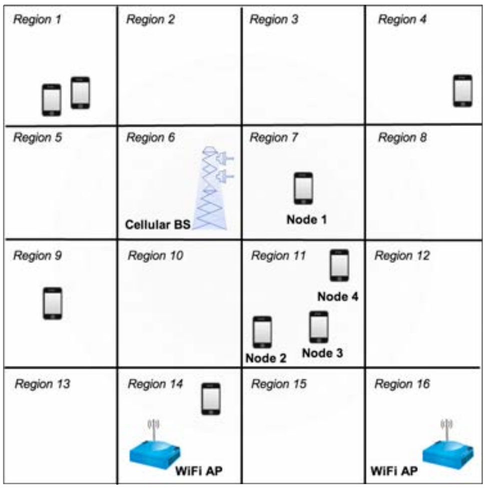


Figure 3.1 Sample scenario

3.5. Adopting CEMMO

In this section, we discuss some issues related to the adoption of CEMMO. First, there exists a need for a mechanism that allows users to set the desired DTI for each application according to its requirements. This can be implemented by exposing a simple API per application, similar to e-mail refresh rate settings in mobile devices. Alternatively, CEMMO can use application port information to infer a predefined delay tolerance to specific ports. Intermediate carriers also have to be able to opt-in and opt-out of the information dissemination process anytime they want and set their own battery power thresholds, in order to tune their participation in PAO. For example, when their battery level is high, they can fully collaborate; when their battery level is intermediate, they can partially participate, while below a specific battery power threshold they no longer participate in the offloading process. Moreover, as users tend to maintain regular mobility patterns daily, mobile phones can perform scanning only when they are in a pre-recorded area of WiFi stations.

In regard to the proposed mobility and connectivity prediction model, changes in a user's schedule may take long time to be reflected in the model. Users should be able to reset their history when their schedule changes (e.g. between semesters when the everyday schedule of students completely changes). Moreover, CEMMO could also measure the throughput of WiFi networks and include this aspect in the offloading decision, so that users select the most efficient connection. War-driving information can help operators assess the probability of WiFi availability in a region, as well as the quality of the connection. Operators can exploit such information, along with user measurements to extract more accurate connectivity predictions [105]. In order to convince other clients to participate in PAO and share their handheld resources, it is fundamental that every participant node perceives some level of fairness and profit sharing within the network in the long term. Economic frameworks for peer-assisted services that create the right incentives for both users and providers to participate have been proposed [177].

In order for CEMMO to handle download requests, the cellular operator has to be the operator of the WiFi APs as well, in order to be able to push data to the intermediate carriers that in turn deliver it to the source node that issued a request. The gains of CEMMO can be even greater when downloading content in regions where 3G coverage is not strong and mobile devices switch to the second-generation wireless telephone technology (2G). Downloading large volumes of content through CEMMO, instead of 2G or low-rate 3G networks, can lead to faster download times and reduced costs for operators.

Resolving any security and privacy issues that CEMMO may involve is essential for engaging user participation and maintaining their trust. CEMMO requires transmission of private data about user mobility to the cellular operator. Users may be reluctant to disclose sensitive information even though such information is nowadays open-handedly provided to a vast amount of popular apps for smartphones. In order to preserve privacy, providers store this information temporarily, utilise it only to decide on the most efficient data forwarding strategy, and subsequently discard it. In case offloading is performed through unmanaged, untrusted WiFi APs, Internet Protocol Security (IPsec) or selective IP traffic offload (SIPTO) [178] can provide effective security solutions. Several solutions have been proposed for making WiFi APs as secure and easy to use as cellular networks [179][180]. Moreover, all messages in PAO need to be encrypted, so that no intermediate carrier can gain access to them.

4. Connectivity Plan Routing Protocol

In this chapter, we describe the proposed DTN-based architecture for dense urban environments that extends the existing infrastructure of public transport networks to provide low-cost Internet services through applications that can withstand certain delays without significantly affecting user experience. In Section 4.1, we describe the proposed DTN access model and detail the Connectivity Plan Routing Protocol (CARPOOL), our proposal to achieve data delivery within the Delay Tolerance Threshold that each user defines. In Section 4.2, we present an enhancement of the proposed architecture that exploits not only scheduled, but also opportunistic contact opportunities, and describe the Enhanced Connectivity Plan Routing Protocol (CARPOOL+) that includes smart mechanisms for route calculation.

4.1. DTN-based architecture for CARPOOL

The ultimate goal of the present work is to provide free delay-tolerant Internet access in metropolitan environments to the under-privileged society that is currently excluded from today's digital world. To achieve that, we extend the existing free Internet access provided by public hotspots that are usually scattered around a city. Several governments and local administrations have undertaken the initiative to deploy hotspots in points of interests, however cost-efficiency is a critical factor that hinders extended deployments [181]. Actually, we broaden connectivity options by deploying DTN nodes both on typical means of public transport (ferries), such as buses and trams, and their corresponding stops (gateways). Offline DTN gateways located near ferry stops collect Internet access requests from end-users in that area and DTN ferries act as relays between offline gateways or designated gateways that have access to the Internet and are capable of handling such requests (online gateways). In this context, we introduce CARPOOL [182][183], a DTN routing protocol that utilises the connectivity plan between ferries and gateways (i.e. ferry stops) to compute routes to online gateways. CARPOOL utilises *a priori* knowledge of contacts between gateways and ferries, in order to achieve high delivery ratio. Our approach shares the philosophy of Contact Graph Routing (CGR) [184], which is the most prominent routing solution in space internetworking. CGR extracts a path for space data transmission utilising *a priori* knowledge of contacts between space assets. It is also noted that, in contrast to most solutions proposed in literature, CARPOOL is not a replication scheme. Only a single copy of each bundle exists in the network at any given time, keeping overhead to minimum.

4.1.1. DTN access model

The proposed access model consists of three major components:

- The *mobile devices* of the end-users that are data sources/sinks;
- *DTN gateways* that are responsible for handling requests from end-users within their radius, and
- *DTN ferries* that are responsible for transferring bundles across the gateways. We adopt the most frequently used term "data or message ferry"; typical examples of DTN data ferries include buses and trams.

All components communicate with each other through wireless network interfaces (e.g. WiFi Direct) with certain connectivity range *R*, have certain storage capacity to store bundles and alternate between two basic operations: neighbour discovery and opportunistic transfer. While all components are crucial for our access model, initially we delegated all computational tasks to the gateways, assuming that DTN ferries have restricted energy and computational capabilities. In contrast, DTN gateways are resource-capable fixed nodes located near ferry stops. A typical example of such a DTN node is the Liberouter [9], a low-cost router platform that serves as WLAN access point. We assume that certain gateways have access to the Internet (*online DTN gateways*) through a hotspot that exists in the area, while the majority is offline. Online gateways are typically available at the terminals of public transport vehicles.

The mobile devices of the end-users are expected to communicate with the gateways seamlessly. Once the mobile device of an end-user device discovers a DTN gateway in its radius, an access request to the Internet is transferred from the relevant application. A variety of applications that can benefit from the proposed access model have been proposed in Chapter 2; applications that can withstand greater delays without significantly affecting user experience, such as back-up applications or cloud storage offloading and synchronisation, can benefit greatly from our architecture. Each user application is managed through a dedicated interface, where users can also define the desired Delay Tolerance Threshold, i.e. the amount of time users are willing to wait until their data are delivered. Users can always access the available user interface (UI) to monitor data whose transmission has been delegated to an offline gateway and data that are still waiting for a connection, in order to be transmitted. For example, if a user has prepared three e-mails and delegated their transmission to a delay-tolerant E-mail application, the user can always check whether one or more E-mails have been transmitted to an offline DTN gateway that is now responsible for their transmission. When an Internet access request, along with its Delay Tolerance Threshold, is received by an offline gateway, the gateway executes CARPOOL route selection algorithm to identify valid paths between itself and an online gateway based on the connectivity plan. Typically, the travel plan of buses, trams and trains is predefined and only minor delays can occur. Therefore, in our model we assume that all gateways have global knowledge of the connectivity plan. In case of a major delay, the updated traffic schedule can be flooded into the network through a central administrative node. The selection of the online gateway is performed in a round-robin manner; however, if more information on network traffic or gateway location is available, the online gateways can be selected in a way that network load balancing is also performed.

In case no available route is identified before the Delay-Tolerance Threshold or Time-To-Live (TTL) expires, or the gateway has no available buffer space to store the bundle, the mobile device is notified that data delivery has not been accepted by this gateway and the bundle is stored in the mobile device of the user until another gateway is met. For increased Delay Tolerance Threshold, the offline gateway is more likely to identify a valid route to an online gateway and, therefore, the probability that the bundle is accepted is higher. All other gateways on the path can only refuse to undertake the responsibility to serve an Internet access request, if they do not have adequate buffer space to store the bundle. If the offline gateway has adequate buffer space and CARPOOL identifies valid paths, a path that is expected to achieve earliest bundle delivery is selected and the offline gateway extracts the identity (ID) of the next gateway on this path, the ID of the ferry that will transfer the bundle and the estimated forwarding time. The gateway notifies the mobile device that the data transmission has been accepted and requests the data. Once the gateway receives the data, it updates the relevant bundle header fields with the extracted values and stores the bundle in its buffer. Bundles are, then, forwarded to the selected ferries as soon as the respective contacts become available, i.e. when the ferry is in range of the gateway. It is noted that in order to establish a connection between any two components, a typical handshake procedure is required; we assume that the handshake procedure, along with neighbour discovery, is performed by link layer communication protocols (e.g. using the WiFi P2P Device Discovery service).

Message ferries move on predetermined paths and pause at each gateway for a certain period of time, allowing for the data exchange. We assume that a ferry stops at every stop on its path for a short period of time. When a ferry and a gateway are in range, data are exchanged as depicted in Figure 4.1. The ferry downloads all bundles destined for this gateway, while the gateway uploads all bundles that are destined for this ferry. When an online gateway receives a new bundle, the bundle is forwarded to the receiving application through the Internet. When an offline gateway receives a bundle, CARPOOL route selection algorithm is re-executed and the corresponding fields in the header of the bundle are updated. We assume that both gateways and ferries have radios that allow simultaneous upload and download. The amount of data that is exchanged between nodes depends on the contact time. Overall, our access model faces two limitations: *the finite buffer size of gateways and ferries* and *the small window of communication opportunities between gateways and ferries*. In the event that this window does not suffice for all bundles to be delivered to an offline or online gateway, the path for the unserviced bundles is re-calculated. In order to assure that all bundles are finally delivered to the Internet before their Delay Tolerance Threshold expires, one approach would be to equip all gateways with basic cellular connectivity that will only be exploited in case a bundle cannot be delivered within the Delay Tolerance Threshold. If equipping all gateways with basic cellular connectivity is not cost-effective, the proposed access model could prioritize or flood certain bundles, as soon as a ferry or gateway estimates that they cannot be delivered before their Delay Tolerance Threshold expires. Both approaches are only introduced as fallback mechanisms, in order to guarantee that all data are finally delivered to their destination.

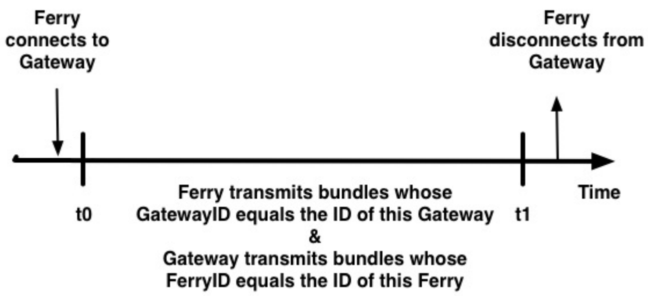


Figure 4.1 Data exchange between a gateway and a ferry

It is noted that, instead of storing the E2E path through the network, we only store the next gateway on the path. This approach ensures that our model takes into consideration and proactively handles changes to the initial connectivity plan. If the full path to an online gateway were stored, the time-shift of an intermediate contact, even for a few seconds, would lead to a significant delay, let alone bundle expiration. The proposed method reacts to changes in the state of the network by re-evaluating the best route for a bundle at every gateway. In this way, CARPOOL recalculates routes each time a bundle misses its expected contact due to high load in the network or short connectivity time between gateways and ferries.

Figure 4.2 contains a sample topology corresponding to our model. We highlight that the majority of gateways do not have access to the Internet. Municipalities or local organisations can adopt this model, in order to extend the coverage area of the free Internet services they offer. Instead of installing new costly infrastructure all over a city to provide Internet access, a municipality can opt for this model, simply by installing the relevant components to bus stops and buses. The proposed architecture can be built using typical low-cost platforms, such as Raspberry Pi [185], equipped with storage devices. Moreover, its low energy requirements can be covered using a solar panel. The architecture is mainly expected to serve users that freely access the Internet though delay-tolerant applications, but can be further extended to support users that need to offload data from the cellular network, due to cellular volume restrictions. We expect applications such as E-mail [152], fbDTN [153] and Twitter [154] that can withstand certain delays without significantly affecting user experience to benefit from the proposed architecture.

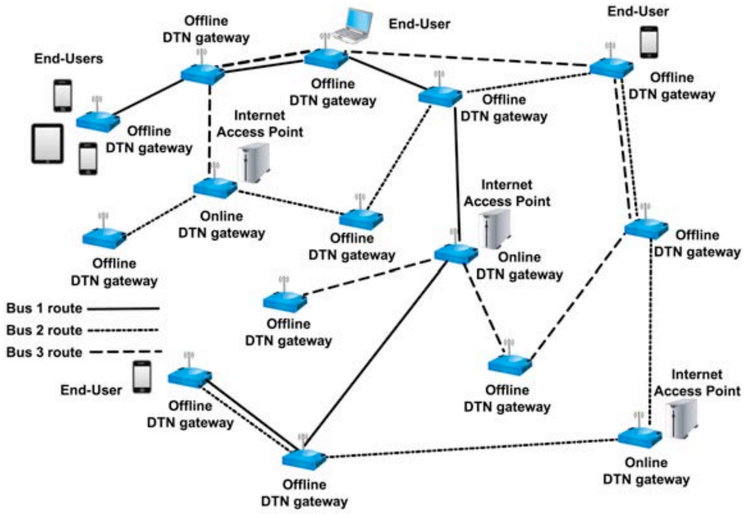


Figure 4.2 Sample topology

Unlike upload operations, downloading data from the Internet requires an additional PUB/SUB session layer (e.g. similar to the one presented in [186]), in order to allow for applications such as Rich Site Summary (RSS) content distribution and web access over DTN. The proposed architecture can also be the enabler of new types of applications, such as the identification of nearby people (as showcased in [129]) or device-to-device communication when a disaster occurs and typical online communications are not available. The proposed architecture is an independent network that is highly resistant to both expected and non-anticipated disruptions. Regarding cost, we expect municipalities and non-governmental organisations (NGOs) to cover the installation cost in an effort to provide free delay-tolerant Internet access to the digitally excluded, enable new types of services and own a resilient opportunistic communications networks that can be exploited in case of disasters. For download requests, a user can explicitly state the gateway where data need to be delivered; this is not necessarily the same gateway that issued the request. The proposed routing protocol can function efficiently in both cases. We also note that the proposed model can be extended to support data transfers between end-users that are located near gateways, as well. In this work, we do not study issues of privacy and security. We assume that relevant mechanisms, such as the Bundle Security Protocol (BSP) [187], exist.

4.1.2. CARPOOL route selection algorithm

In order to support the proposed access model, we have designed and implemented CARPOOL, a DTN routing protocol that utilises *a priori* knowledge of the connectivity plan to deliver bundles among DTN gateways. The proposed routing protocol works as follows: all gateways hold *the list of online gateways* and *the overall connectivity plan*, which includes all contacts between gateways and ferries along with the scheduled start of each contact. In particular, each contact entry (C) of the connectivity table (ConnectivityTable) has the following 3-tuple structure (GatewayID, FerryID, ContactTime). We also assume that the entries of the ConnectivityTable are sorted based on FerryID and, for each ferry, based on ContactTime. Thus, the ConnectivityTable first includes contacts for Ferry1, sorted based on ContactTime, followed by contacts for Ferry2, sorted based on ContactTime etc. When an offline gateway receives a new bundle from an end-user, CARPOOL identifies the most suitable next gateway for this bundle, in terms of estimated bundle delivery time to an online gateway. CARPOOL identifies paths between an offline gateway and an online gateway, starting from the destination, and moving towards the source in a hop-by-hop manner. The values required as input to the algorithm are PreviousGateway and NewArrivalTime. Initially, PreviousGateway is set to the ID of an online gateway and NewArrivalTime equals to bundle creation time plus TTL. The current gateway first identifies all contacts in the overall connectivity table that satisfy the following requirements:

- GatewayID equals PreviousGateway, and
- ContactTime is greater than CurrentTime and less than the latest arrival time (NewArrivalTime).

For each of the aforementioned contacts, we store a set of 3-Tuples: the contact itself and the exact previous contact (in terms of time) between the same ferry and another gateway. When the previous gateway that this ferry has traversed becomes the current gateway, we have identified a direct contact, where the current gateway is only one hop away from an online gateway. Otherwise, the algorithm re-executes using as input the GatewayID and the ContactTime of the previous contact. Thus, we now search for valid contacts that are two hops away from an online gateway. This process is continued until a path is found.

In order to reduce the complexity and the associated computational overhead of our algorithm, instead of identifying all possible paths and selecting one that is expected to achieve earliest delivery, we first sort valid contacts to an online gateway starting from the earliest, prior to applying our selection algorithm. This way, we need not calculate all paths from the current gateway to all online gateways; instead we simply select the first plausible path to an online gateway, which is also a path that is expected to achieve earliest delivery. Once a path has been discovered, the GatewayID of the next gateway on the path (*NextGateway*), the FerryID of the ferry that will transfer the bundle (*NextFerry*) and the *TimeToForward* that corresponds to the time that the bundle will be forwarded are added to the header of the bundle; the bundle is then stored in the gateway buffer, until a connection between the gateway and *this* ferry exists. CARPOOL route selection algorithm is presented in Table 4.1.

```
Input:    Initially, set PreviousGateway equal to the final recipient (i.e. online gateway)
          and
          newArrivalTime = Bundle Creation Time + TTL
Output:  Bundle header fields:  NextGateway, NextFerry, TimeToForward

For each Ferry F do
    For Contact C of ConnectivityTable_F do
        If (C.GatewayID = PreviousGateway) AND
                (C.ContactTime ≥ CurrentTime) AND
                (C.ContactTime ≤ NewArrivalTime) then
                    //A valid contact has been found. We store a set of 3-tuples: this
                //contact and the previous contact the ferry has with another gateway
            Add Set(Cprev,C) to ValidContacts
        Endif
    Endfor
Endfor
//Sort valid contacts starting from the earliest contact
C Sort ValidContacts
//Identify a path from source to destination
For Set(Sprev,S) of ValidContacts do
    If (Sprev.Gateway = CurrentGateway) then
    //Path found. Store header fields and exit algorithm
        NextGateway = S.GatewayID
        NextFerry = S.FerryID
        TimeToForward = Sprev.ContactTime
        Exit
    Else
    //We have not found a path from current gateway to the online gateway.
    //Re-run the algorithm moving one hop further from the online gateway
        PreviousGateway = Sprev.GatewayID
        NewArrivalTime = Sprev.ContactTime
        Algorithm (PreviousGateway, NewArrivalTime)
    Endif
Endfor
```

Table 4.1 CARPOOL route selection algorithm

CARPOOL re-calculates the path for all bundles in the network with *TimeToForward* greater than CurrentTime aligned within a fixed threshold, set according to the arrival time of the next ferry to this gateway, in order not to miss this contact opportunity. This allows to cancel the impact of the schedule deviation and typically suffices to accommodate minor schedule drifts.

4.2. DTN-based architecture for CARPOOL+

The movement of ferries in a public transportation network exhibits some form of regularity and, therefore, routing in a deterministic way is in theory possible. A real vehicular network, however, does not follow the exact time schedule due to variable speeds and waiting times at stops, as well as delays due to unpredicted events such as road accidents and roadside activities. Based on the experience gained from the design of CARPOOL, we now design a more sophisticated routing protocol with mechanisms specifically designed for dynamic public transportation networks. In particular, we develop CARPOOL+ [188], an Enhanced Connectivity Plan Routing Protocol, which exploits not only predefined contacts, but also opportunistic contacts between ferries, and deals with schedule changes by recalculating the path when a major delay occurs. In the proposed architecture, we assume that all gateways and ferries have global knowledge of the scheduled connectivity plan; however ferry mobility is quasi-deterministic since deviations regularly occur, e.g. due to road traffic congestion. Our goal is to adaptively identify routes that improve the overall delivery ratio with low delay and minimum overhead. Similar to CARPOOL, CARPOOL+ is a single-copy DTN routing protocol specifically designed for highly dense networks that exploits existing knowledge on node connectivity to smartly store, carry and forward bundles from an offline gateway to a gateway connected to the Internet.

4.2.1. Enhanced DTN access model

The DTN access model of CARPOOL+ is based on the access model of CARPOOL, as described in Section 4.1.1, enhanced with two dynamic mechanisms:

- The exploitation opportunistic contacts among ferries whenever they become available and provide a better route in terms of estimated delivery time, and
- Path recalculation en route whenever a deviation from schedule occurs, e.g. due to road traffic congestion.

In particular, when a ferry detects another ferry within its range, it calculates the estimated delivery time of each bundle it holds through the second ferry. In case the estimated delivery time is earlier than the value stored in the header of the bundle (i.e. the estimated delivery time through the original ferry), then the bundle is copied to the buffer of the second ferry. The original ferry then discards its copy, while the second ferry is now responsible for the delivery of the bundle to the final recipient. This approach is similar to the inherent characteristic of DTNs; *custody transfer* [2]. Even though we assume that the estimation of contacts between ferries and gateways is pretty accurate, a similar assumption regarding intra-ferry encounters is less reasonable, since both nodes move dynamically and the communication window between two ferries depends on their speed, the direction towards which they are moving, their distance, the data rate of their network cards, etc. The inclusion of intra-ferry encounter estimations in the route selection algorithm would lead to increased computational complexity and overhead, as well as more prediction errors, since the slightest ferry delay would result in wrong estimations.

An example of this procedure is provided in Figure 4.3. In this scenario, gateway GW1 is the source of bundle M, while gateway GW3 is the destination. Based on the available connectivity plan, ferry1 provides the fastest route from GW1 to GW3, through GW2, and, therefore, bundle M is transferred to ferry1 when it comes within range of GW1. While on route from GW1 to GW2, ferry1 and ferry2 opportunistically come into range and ferry1 requests ferry2 to calculate the estimated delivery time of bundle M to GW3 through ferry2. After the execution of CARPOOL+ route selection algorithm, ferry2 responds with the calculated estimated delivery time, which is earlier than the original estimated delivery time (through ferry1) stored in the header of bundle M. For this reason, ferry1 transfers a copy of bundle M to ferry2, which is now responsible for its transmission, drops the original bundle it holds, and both ferries follow their predetermined paths. When ferry2 and GW3 come into range, bundle M is successfully delivered to its destination GW3, earlier than initially expected.

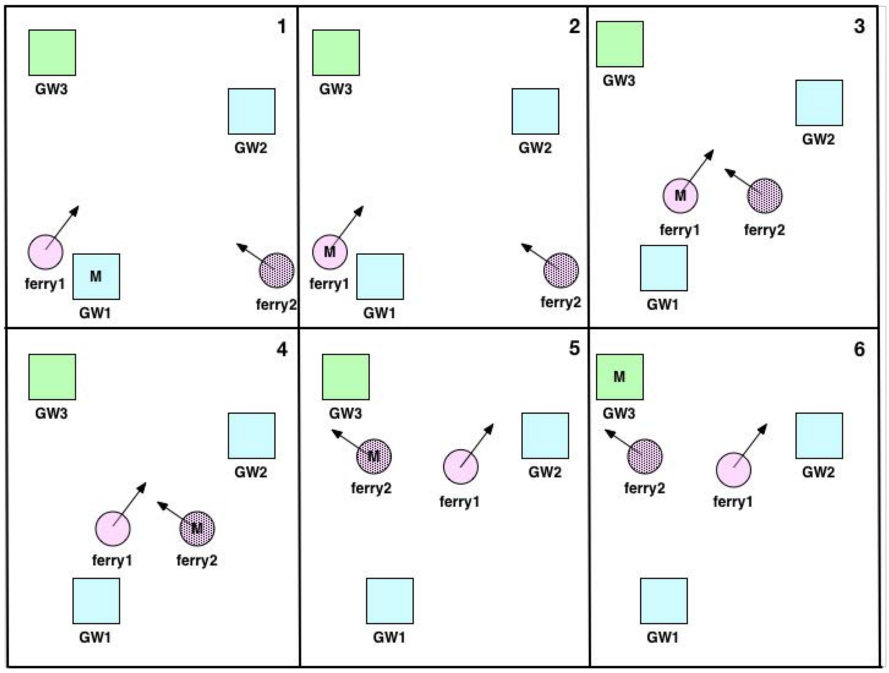


Figure 4.3 Data exchange between two ferries during an opportunistic contact

Our architecture also deals with minor or major delays by not sticking to the predetermined contact plan, but allowing ferries to recalculate the path whenever a delay occurs. In particular, each time a ferry arrives at an offline gateway, it checks the estimated arrival time, as given in the connectivity plan. In case the ferry arrives later than expected, the ferry forwards all bundles destined to that gateway and recalculates the estimated delivery time for the remaining bundles given the delayed arrival. If another path that is expected to achieve earliest delivery time is identified for certain bundles, then the relevant bundle header fields are updated and the bundles follow the new route.

This procedure is depicted in Figure 4.4. In this scenario, ferry1 holds bundle M, in order to deliver it to its destination GW3, after visiting GW1 and GW2 on its path. However, due to a road accident, ferry1 arrives to GW1 significantly later than expected. For this reason, ferry1 recalculates the estimated delivery time of bundle M, concludes that a new fastest route can be found through GW1, transfers bundle M to GW1, which is now responsible for its delivery, and moves along its path. GW1 executes CARPOOL+ route calculation algorithm and concludes that the fastest route for bundle M to GW3 is through ferry2. As soon as ferry2 and GW1 come into range, GW1 transfers a copy of bundle M to ferry 2 and drops the original bundle it holds. Ferry 2 successfully delivers bundle M to GW3, prior to the arrival of ferry1 (i.e. earlier than the initial estimated delivery time).

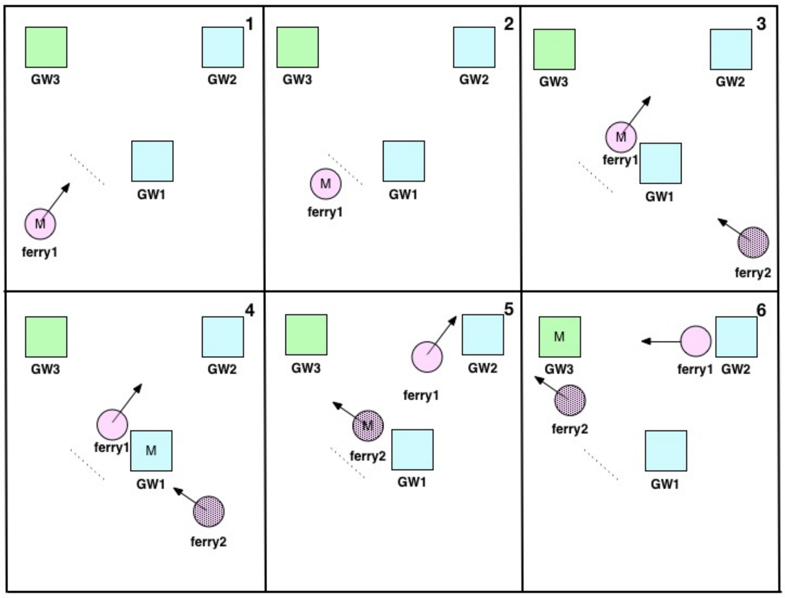


Figure 4.4 Path recalculation en route in case of delayed ferry arrival

The incorporation of the aforementioned dynamic mechanisms can lead to the exploitation of several new routes that were previously unavailable, resulting in an increase of the amount of successfully delivered data, as well as reduction of the average latency.

4.2.2. CARPOOL+ route selection algorithm

The input and output of CARPOOL+ route selection algorithm is depicted in Figure 4.5.

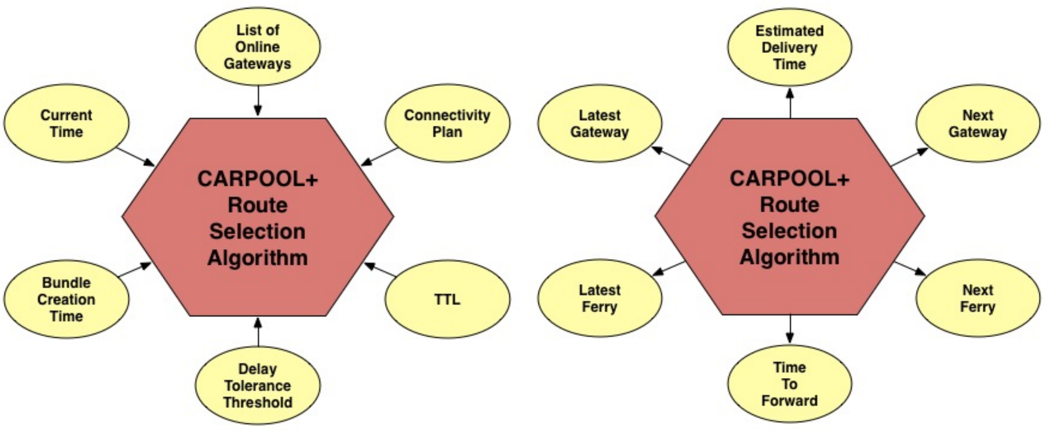


Figure 4.5 Input and output of CARPOOL+ route selection algorithm

In contrast to most DTN routing solutions, CARPOOL+ is a single-copy protocol that does not flood the network with excessive bundle copies, minimising transmission overhead and resource consumption. Similar to CARPOOL, CARPOOL+ does not perform source routing. CARPOOL+ computes the whole path from the source to the destination, but, given the dynamic nature of the topology, instead of storing the E2E path through the network, CARPOOL+ only stores the next gateway on the path and the route is re-evaluated at every step. In case another route that is expected to achieve earliest bundle delivery time is identified en route, it is exploited. This approach ensures that our protocol proactively handles changes to the initial connectivity plan. If the full path to an online gateway were stored, the time-shift of an intermediate contact (i.e. delay of a ferry due to road traffic congestion), even for a few minutes, would possibly lead to a significant delay, let alone bundle expiration. In order to further reduce complexity and re-calculations, bundles stored in the same offline gateway or ferry and destined to the same online gateway could be aggregated.

We assume that all gateways hold *the list of online gateways* and *the overall connectivity plan*, which includes all contacts between gateways and ferries along with the scheduled start of each contact. In particular, each contact entry (C) of the connectivity table (ConnectivityTable) has the following 3-tuple structure (GatewayID, FerryID, ContactTime). We also assume that the entries of the ConnectivityTable are sorted based on FerryID and, for each ferry, based on ContactTime. Thus, the ConnectivityTable first includes contacts for Ferry1, sorted based on ContactTime, followed by contacts for Ferry2, sorted based on ContactTime etc. CARPOOL+ identifies routes between an offline gateway and an online gateway starting from the destination (i.e. the online gateway) and moving towards the source in a hop-by-hop manner.

Assuming that the network topology is known, a reasonable approach to select the online gateway that will serve an Internet access request is to select an online gateway that is geographically closer to the source node. However, this approach leads to central online gateways serving notably more Internet access requests than online gateways located close to the edges of the network; thus certain links are congested, while others are under-utilised. Therefore, we opt for the round-robin selection of the online gateway that will serve an Internet access request, so that traffic load is equally shared among the available online gateways and each online gateway serves the same amount of Internet access requests. For applications with tighter TTLs and Delay Tolerance Thresholds, the online gateway selection algorithm can be modified to select only online gateways that are able to achieve timely data delivery. Route calculation is performed in six cases:

- When an offline gateway receives an Internet access request from an end-user, CARPOOL+ selects a gateway from the list of online gateways as the destination in a round-robin manner and identifies the most suitable next gateway for this bundle, in terms of earliest estimated bundle delivery time.
- When an intermediate gateway receives a bundle, it recalculates the path to the online gateway and updates the relevant header fields.

- When a ferry arrives at a gateway later than expected, it downloads bundles destined to this gateway and recalculates the path for the remaining bundles it holds.
- When an offline gateway holds bundles to be transmitted through a ferry that has been delayed, it recalculates the route for these bundles prior to the next scheduled contact with another ferry, in order not to miss the next available contact opportunity.
- When two ferries are in range, each ferry requests the other ferry to calculate the estimated delivery time for the bundles it holds in its buffer through the other ferry.
- When a gateway tries to transmit bundles to a ferry whose buffer is already full and vice versa.

We assume that both gateways and ferries have knowledge of the current time and their ID, while gateways also know the ID of the next gateway (i.e. stop) on their path. The variables required to initialise the algorithm are PreviousGateway, NewArrivalTime, CurrentGateway and DestinationID. Initially, PreviousGateway is set to the ID of the online gateway (DestinationID) that will serve this request and NewArrivalTime equals to the minimum value among Bundle Creation Time plus TTL or Bundle Creation Time plus Delay Tolerance Threshold for these data. CurrentGateway is the ID of the gateway that holds the bundle to be transmitted or, in case of route calculation through another ferry, the ID of the next gateway (i.e. stop) on the path of the ferry. The gateway or ferry that holds the bundle to be transmitted first identifies all contacts in the overall connectivity table that satisfy the following requirements:

- GatewayID equals PreviousGateway, and
- ContactTime is greater than CurrentTime and less than NewArrivalTime.

For each of these contacts, we store in a list of valid contacts (ValidContacts) a set of 3-Tuples: the contact itself (C) and the exact previous contact (in terms of time) between the same ferry and another gateway (C_{prev}) (i.e. the contact that refers to the previous stop on the path of the ferry). As an example, we consider the simplest case in which the source node is only one hop away from the destination and there exists only one ferry that connects them. We assume that source ID is GW1, destination ID is GW2 and, according to the connectivity plan Ferry1 visits GW1 at t_1 and its next stop is GW2 at t_2. In this simple case, C equals to (GW2, Ferry1, t_2) and C_{prev} equals to (GW1, Ferry1, t_1).

In order to reduce the complexity and the associated computational overhead of the route selection algorithm when more than one contacts that fulfil the aforementioned requirements exist in the connectivity table, instead of identifying all possible routes and selecting one that is expected to achieve the earliest delivery time, we first sort valid contacts to the online gateway starting from the contact that arrives earliest at the destination, prior to applying our selection algorithm. This way, the calculation of all routes from the current gateway to the online gateway is not required. Since our search for a route always begins with the earliest valid contacts and we search for routes starting from the destination and moving backwards towards the source in a hop-by-hop manner, the first plausible route to the online gateway that we identify is also a route that is expected to achieve the earliest delivery time. Therefore, the selected route is optimal in terms of earliest delivery time, without taking into account unscheduled delays. Of course, more than one route may exist that achieves the same delivery time, but this does not constitute the selected route less optimal.

In particular, having sorted the list of valid contacts, we search for paths between CurrentGateway and PreviousGateway for each entry of ValidContacts. If the ID of the next gateway in ValidContacts equals to DestinationID, we store the ContactTime of the entry as the *EstimatedDeliveryTime*, since this is a direct contact to the online gateway (i.e. the last hop on the path). If the ID of the previous gateway that this ferry has traversed is equal to CurrentGateway, we have identified a direct contact, where the current gateway is only one hop away from the online gateway. In this case, we first confirm that this contact does not lead to a loop and then update the relevant bundle header fields. Otherwise, CARPOOL+ route selection algorithm is re-executed using as starting point the GatewayID and the ContactTime of the previous contact. Thus, we now search for valid contacts that are two hops away from the online gateway. This process is repeated until a route is found. CARPOOL+ bundle header includes six fields in total:

- *EstimatedDeliveryTime*, the time when the bundle is expected to be delivered to its final recipient (i.e. the online gateway);
- *TimeToForward,* the time when the next gateway or ferry is expected to receive this bundle;
- *NextFerry,* the next ferry that will carry the bundle;
- *NextGateway,* the next gateway where the bundle will be delivered;
- *LatestGateway,* the latest gateway where the bundle was stored, and
- *LatestFerry,* the latest ferry that carried the bundle.

EstimatedDeliveryTime serves a dual role: initially to identify if a bundle can be delivered within the Delay Tolerance Threshold as set by the user, and later, in case of an opportunistic contact between two ferries, to identify if a bundle can be delivered earlier through another ferry. Using *TimeToForward, NextFerry* and *NextGateway* ferries and gateways know when and to whom each bundle should be forwarded next. In case of delays, some bundles may be caught up in a loop between a gateway and a ferry. By holding the ID of the latest gateway where a bundle was stored (*LatestGateway*) in the header of each bundle and checking it prior to each transmission, we avoid transfers to that node again and, thus, typical loops. Similarly, we hold the ID of the latest ferry that transferred a bundle (*LatestFerry*) in the header of each bundle and check it prior to each transmission in order to avoid transfers over the reverse path. A notation table (Table 4.2) and CARPOOL+ route selection algorithm (Table 4.3) are included below.

Symbol	**Definition**
PreviousGateway	A variable that stores the ID of a gateway
CurrentGateway	The ID of the gateway that currently holds a bundle. If the bundle is held by a ferry, CurrentGateway equals to the ID of the next gateway on the path of the ferry
DestinationID	The ID of the destination node (i.e. the online gateway)
CurrentTime	The current time
NewArrivalTime	A variable that stores a contact time value
BundleCreationTime	The time when a bundle was created
DelayToleranceThreshold	The time that a user is willing to wait until the data are delivered to an online gateway
TTL	Bundle Time-To-Live
ConnectivityTable	The connectivity table. All entries are sorted based on FerryID and, for each ferry, based on ContactTime
ValidContacts	A table where all contacts that fulfill the requirements of CARPOOL+ route selection algorithm are stored
C(GatewayID, FerryID, ContactTime)	An entry in the connectivity table
C_{prev}(GatewayID, FerryID, ContactTime)	The exact previous entry in the connectivity table that refers to the previous stop on the path of FerryID
Set(C_{prev},C)	A set of contacts

Table 4.2 Notation table

The core of CARPOOL+ route selection algorithm is Dijkstra's algorithm using delivery time as cost metric, similar to [184]. Assuming that the network consists of V nodes and E contacts and using Fibonacci heap as a priority queue, each iteration at each node is theoretically executable in O(E+VlogV) time. Since CARPOOL+ route selection algorithm is run at most V times, the time complexity of the algorithm is $O(VE+V^2 \log V)$. This is the worst case scenario; typically, the complexity of CARPOOL+ route selection algorithm is significantly smaller since we do not calculate all plausible routes, but stop searching for other routes once we identify a route that is expected to achieve the earliest delivery time, as explained above.

```
Input:    ConnectivityTable, ListOfOnlineGateways,BundleCreationTime,
          DelayToleranceThreshold, TTL, CurrentTime
Output: EstimatedDeliveryTime, NextGateway, NextFerry, TimeToForward, LatestGateway,
          LatestFerry

//Algorithm Initialization
ValidContacts = CreateList();
EstimatedDeliveryTime = 0;
NextGateway = NULL;
NextFerry = NULL;
TimeToForward = 0;
LatestFerry = NULL;
LatestGateway = NULL;
DestinationID = SelectInRoundRobin(ListOfOnlineGateways);
//We select an online gateway as the destination in a round robin manner
PreviousGateway = DestinationID;
NewArrivalTime = min(BundleCreationTime + TTL,
                    BundleCreationTime + DelayToleranceThreshold);
switch (getLocalNodeType())
          case NodeType.GATEWAY:
          //If the route is calculated by a gateway, CurrentGateway equals to the ID of this gateway
                    CurrentGateway =  ThisGatewayID;
                    break;
          case NodeType.FERRY:
          //If the route is calculated by a ferry,
          //CurrentGateway equals to the ID of the next gateway on the path of the ferry
                    CurrentGateway = NextGatewayIDOnFerryPath;
                    break;
end switch
//Required variables have been initialized
```

```
CARPOOL+(PreviousGateway, NewArrivalTime, CurrentGateway, DestinationID)
//CARPOOL+ core algorithm
{
        for (i=1; i<ConnectivityTable.size; i++)
        //We search all entries of the connectivity table. We assume that the entries of the
        //ConnectivityTable are sorted based on FerryID and, for each ferry, based on ContactTime
                C = ConnectivityTable[i];
                //Get contact C with fields {C.GatewayID, C.FerryID, C.ContactTime}
                if ((C.GatewayID == PreviousGateway) &&
                   (C.ContactTime >= CurrentTime) &&
                   (C.ContactTime <= NewArrivalTime))
                        Cprev = ConnectivityTable[i-1];
                        //Get the previous entry of the ConnectivityTable
                        if (C.FerryID == Cprev.FerryID)
                        //These are contacts of the same ferry, so Cprev refers to the previous
                        //stop on the path of the ferry. Therefore, we have identified a valid hop
                        //between PreviousGateway and another gateway
                                ValidContacts.add(Cprev, C);
                        end if
                end if
        end for
        SortBasedOnCContactTime(ValidContacts);
        //We first sort all valid contacts starting from the earliest contact C that arrives at the
        //destination and then search for routes. Since we search for routes backwards, the first
        //route that we identify is also a route that is expected to achieve the earliest delivery time
        for each ([Sprev, S] in ValidContacts)
        //We search all entries of ValidContacts, starting from the earliest
                if (S.GatewayID == DestinationID)
                //This is the last hop on the path, so we store the estimated delivery time
                        EstimatedDeliveryTime = S.ContactTime;
                end if
                if (Sprev.GatewayID == CurrentGateway)
                //The previous gateway on the path of the ferry is CurrentGateway. This means
                //that we have found a route between CurrentGateway and the destination
                        if ((S.GatewayID!=LatestGateway) && (S.FerryID != LatestFerry))
                        //To avoid simple loops, we make sure that a bundle is not delivered to
                        //the latest ferry that carried it or the latest gateway where it was stored.
                        //If this is not a loop, we update bundle header fields and exit algorithm
                                NextGateway = S.GatewayID;
                                NextFerry = S.FerryID;
                                TimeToForward = Sprev.ContactTime;
                                LatestGateway = Sprev.GatewayID;
                                LatestFerry = Sprev.FerryID;
                                return (EstimatedDeliveryTime, NextGateway, NextFerry,
                                        TimeToForward, LatestGateway, LatestFerry);
                        end if
                else
                //We have not found a route between CurrentGateway and PreviousGateway
                        PreviousGateway = Sprev.GatewayID;
                        NewArrivalTime = Sprev.ContactTime;
                        CARPOOL+(PreviousGateway, NewArrivalTime,
                                    CurrentGateway, DestinationID);
                        //We re-run the algorithm moving one hop further from the destination
                        //and closer to CurrentGateway. The first time we run CARPOOL+ core
                        / algorithm we look for one-hop routes between CurrentGateway and
                        //the destination. If no one-hop route has been found, we re-run
                        //CARPOOL+ core algorithm to look for two-hop routes,
                        //then three-hop routes etc.
                end if
        end for
        //No route has been found
        return (-1, NULL, NULL, -1, NULL, NULL);
}
```

Table 4.3 CARPOOL+ route selection algorithm

4.2.3. Communication diagrams

The communication diagrams of the mobile device of an end-user, an offline DTN gateway and a ferry using the proposed architecture are depicted in Figures 4.6-4.8, respectively. In order to establish a connection between any two assets, a typical handshake procedure is required. All functions that are provided by lower communication layers and are not integral part of the proposed architecture have been marked within a thick black box in the communication diagrams between end-users, gateways and ferries that follow. When a mobile device detects an online gateway within its radius, it exploits the available connection to access the Internet. When a mobile device identifies an offline gateway within its range, it only uploads bundles that can be delivered to the Internet within the Delay Tolerance Threshold that the user has defined through the user application (Figure 4.6). We note that in case the mobile device has more than one Internet access requests to transmit, the whole bundle list is transferred from the mobile device of the end-user to the offline gateway that calculates routes for all bundles.

When an offline gateway connects to a ferry, it only uploads bundles whose NextFerryID, as calculated through CARPOOL+ and stored in the header of the bundle, is this ferry (Figure 4.7).

The overall communication diagram of a ferry that includes opportunistic communication between two ferries, as well as communication between a ferry and a gateway (online or offline), is shown in Figure 4.8. It is underlined that a ferry always checks buffer space availability of a gateway prior to each transmission and vice versa. We also note that a bundle carried by a ferry may be transferred to a gateway and then transferred back to the same ferry to be carried to the next hop, but this does not necessarily degrade the performance of the solution. Ferries only have a subset of global knowledge of the network, while gateways have an updated view of the network status since they know if ferries have been delayed or arrived earlier. If we aim to eliminate possible transfers back and forth a ferry and a gateway, a solution would be for gateways to transmit their latest view of the network to a ferry when they are in range, so that the ferry performs path re-calculation (instead of the gateway), updates the relevant header fields for the bundles it holds and sends to the gateway only bundles that need to be transferred to the next hop through another ferry. However, we did not follow this approach for two reasons. First, since the proposed architecture aims at dense urban environments where several ferries traverse each gateway, providing many alternative routes that constantly change given the dynamic movement of ferries, the case of a bundle being transferred back and forth a ferry and a gateway is rare. Second, gateways, which have an updated view of the network status, can make better routing decisions than ferries. As far as ferries are concerned, there are three alternatives as depicted in Figure 4.8:

- When a ferry connects to an online gateway, it delivers all bundles it holds to the Internet.
- When a ferry connects to an offline gateway, it delivers all bundles whose GatewayID, as calculated through CARPOOL+ and stored in the header of the bundle, is this gateway. In case the ferry is delayed, e.g. due to road traffic, it recalculates the estimated delivery time and updates the relevant header fields of the remaining bundles.
- When a ferry connects to another ferry, it delivers all bundles whose estimated delivery time through the other ferry is earlier than current estimated delivery time.

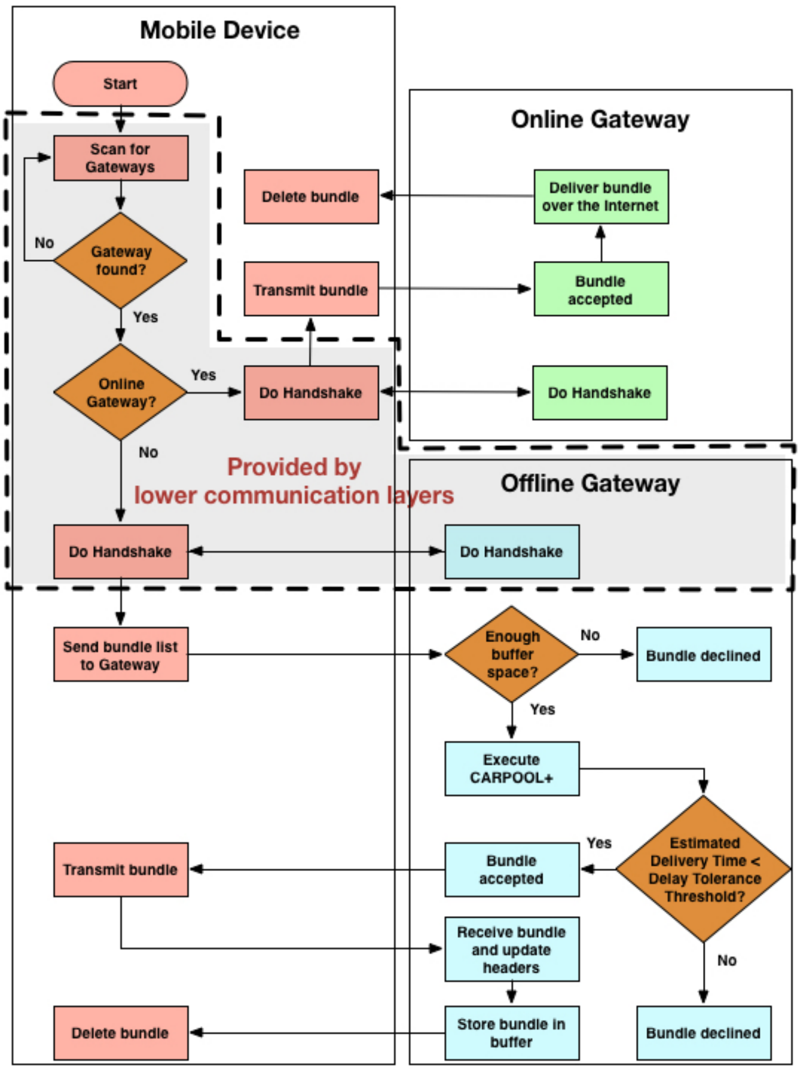


Figure 4.6 Communication diagram of the mobile device of a user

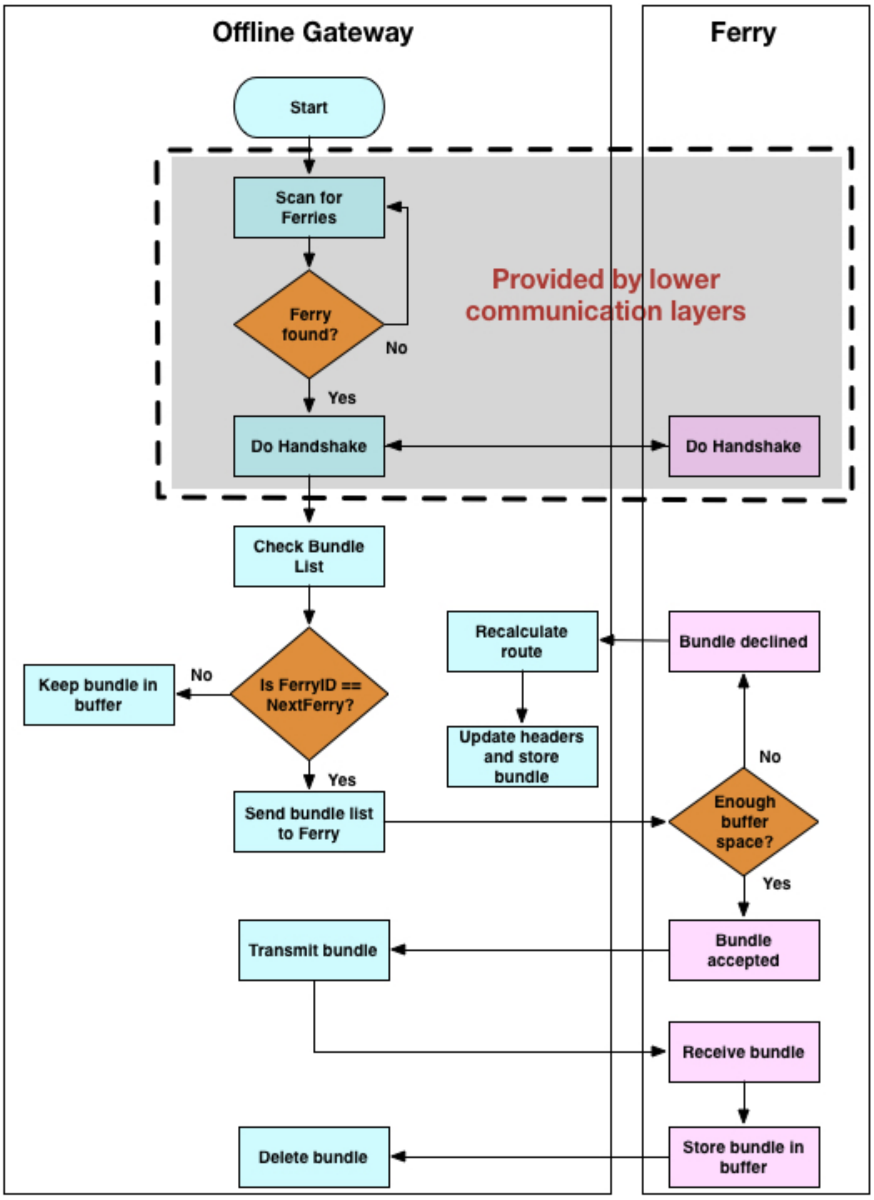


Figure 4.7 Communication diagram of an offline gateway

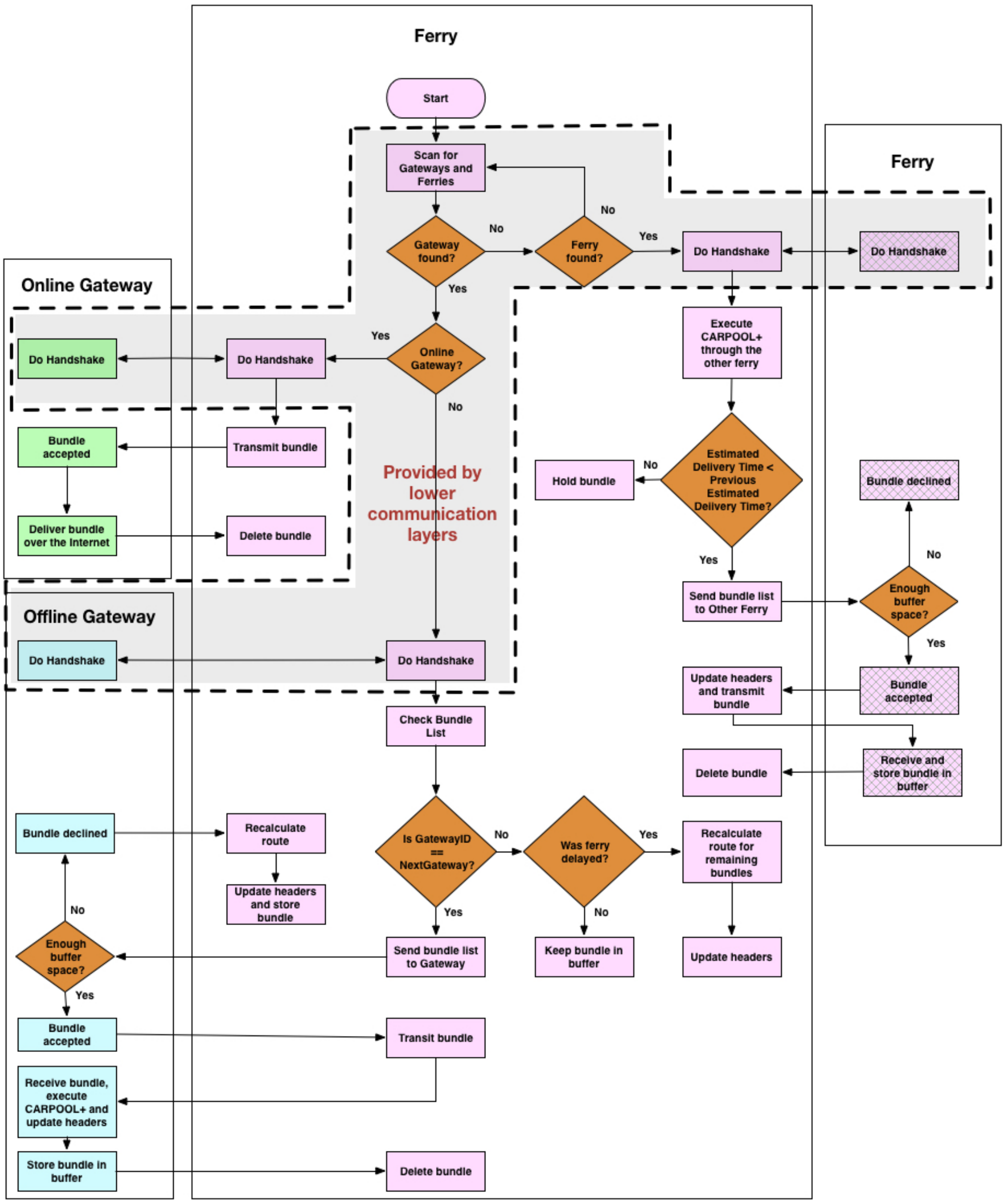


Figure 4.8 Communication diagram of a ferry

5. Evaluation methodology

In this chapter, we present the methodology we follow for the performance evaluation of the protocols and mechanisms that we have designed and implemented as part of the present thesis. The evaluation methodology includes the goals of the evaluation analysis (Section 5.1), the networking environments used in the evaluation of our research proposals (Section 5.2), the metrics used in the performance evaluation (Section 5.3) and the experimentation tools that we used (Section 5.4).

5.1. Goals of the evaluation analysis

It is important to define the goals of the evaluation analysis, which are directly linked to the goals of the protocols and mechanisms designed and implemented within the framework of the present thesis. The overall goal of the evaluation analysis is to identify whether our peer- and vehicle-assisted research proposals are capable of effectively covering the existing gaps in providing extended Internet coverage and data offloading as studied in the present thesis. Considering the design goals and the relevant research directions set by the solutions we propose, the goals of the evaluation analysis are summarised as follows:

1. Evaluate the performance of TCP NewReno, LEDBAT and fLEDBAT access methods in the sub-packet regime when flows of a single type occupy a link and investigate any unfairness issues that may arise when both TCP and fLEDBAT flows share the same link. (*Scenario 1*)
2. Investigate the impact of public WiFi APs availability in the performance of CEMMO mechanism, pure OTSO and pure DTO, highlight the gains of CEMMO in terms of amount of offloaded data, study the impact of β/α ratio in the performance of CEMMO and investigate the storage requirements of CEMMO per user. (*Scenario 2*)
3. Investigate the energy efficiency of CEMMO mechanism. (*Scenario 3*)
4. Evaluate the performance of CARPOOL in comparison to the most prominent DTN routing protocols. (*Scenario 4*)
5. Highlight the performance gains of CARPOOL+ over CARPOOL with the incorporation of the two new mechanisms, evaluate the impact of online gateway availability and buffer size on the performance of CARPOOL+ and compare the performance of CARPOOL+ to the other DTN routing protocols. (*Scenario 5*)

In order to draw meaningful conclusions, the experimental scenarios used in our evaluation analysis are based on networking conditions that are close to reality and far from arbitrary assumptions. We also note that the evaluation methodology we follow does not lead to specific, but general conclusions that are not restricted by the specific environmental conditions set in our evaluation scenarios. Therefore, the research results included in this thesis present an overall perspective of the issues we study and provide knowledge that can be exploited as the base for further research.

5.2. Networking scenarios under evaluation

We evaluate the proposed protocols and mechanisms under different networking conditions, depending on the goals of each evaluation scenario. In particular, we divide our analysis into 5 evaluation scenarios, as described in the following subsections.

5.2.1. Scenario 1: Performance evaluation of TCP New Reno, LEDBAT and fLEDBAT access methods in the sub-packet regime

The goal of Scenario 1 is to evaluate the performance of TCP NewReno, LEDBAT and fLEDBAT access methods in the sub-packet regime of shared backhaul links, i.e. when a low-bandwidth link is heavily shared among many competing flows. We aim to investigate if a LBE access method that is typically less aggressive than TCP can better utilise the limited available bandwidth in highly contention conditions. In particular, we evaluate LEDBAT and fLEDBAT for an increasing number of concurrent flows in the uplink of a shared backhaul link, beginning with only a few flows that are not enough to drive the link in the sub-packet regime; this way, we evaluate their performance both in and out of the sub-packet regime. All work on LEDBAT so far has been focused on scenarios where the network is assumed to have sufficiently large capacity and is never driven into the sub-packet regime. To the best of our knowledge, this is the first time LEDBAT and fLEDBAT are evaluated in the sub-packet regime. To achieve that, we utilise Network Simulator 2 (NS-2) version 2.35 [189]. LEDBAT code is already available [190], while fLEDBAT is implemented by modifying LEDBAT code. We evaluate the aforementioned protocols by simulating three different backhaul links with varying capacity and delay, as depicted in Table 5.1. The characteristics of the backhaul links are extracted from measurements reported in [191][192].

Case	Uplink Capacity	Downlink Capacity	RTT	Uplink Buffer	Downlink Buffer
(a)	58 Kbps	135 Kbps	900 ms	4 pkts	10 pkts
(b)	600 Kbps	1.2 Mbps	450 ms	22 pkts	45 pkts
(c)	800 Kbps	2.2 Mbps	450 ms	30 pkts	82 pkts

Table 5.1 Backhaul link characteristics for each case

The backhaul uplink is occupied by an increasing number of flows (N) ranging from 2 to 96, which is realistic considering typical backhaul links in emerging regions [6060]. In all cases, buffer size is set equal to the bandwidth-delay product and all flows use a fixed packet size equal to 1500 Bytes, including 40 Bytes header. LEDBAT parameter α is set to 1, while parameter ζ of fLEDBAT is set to 5. TARGET is set equal to 100 ms. Both DropTail and Random Early Detection (RED) [193] without Explicit Congestion Notification (ECN) [194] are used for queue management; when RED is used, maximum threshold is set equal to buffer size divided by two, while minimum threshold is set equal to maximum threshold divided by three. All flows are File Transfer Protocol (FTP) flows, we randomise the start times of each flow uniformly between 0 s and 30 s and each simulation lasts for 300 s. Each set of experiments is repeated 30 times, since we randomise the start times of each flow, and the mean values for each evaluation metric are extracted.

Our analysis is divided into two main categories:

- Flows of a single type (i.e. TCP, LEDBAT or fLEDBAT) occupy the backhaul link, and
- A single link is shared among an equal number of TCP and fLEDBAT flows, in order to investigate whether fLEDBAT satisfies its design principles by yielding to TCP flows.

5.2.2. Scenario 2: Performance evaluation of CEMMO mechanism compared to pure OTSO and pure DTO

The goal of Scenario 2 is to extensively evaluate CEMMO mechanism in comparison to pure OTSO and pure DTO, which is the state-of-the-art approach in mobile data offloading. All DTO solutions in literature share the same functionality; only the Delay Tolerance Threshold changes. In this scenario, we aim to highlight the advantage of a mechanism that selects the most cost-effective offloading policy in terms of cost, as defined by each operator, as well as the gains of the new PAO method that offloads data through other peers irrespective of content and popularity. All offloading methods, as well as CEMMO mechanism, are implemented and evaluated in the Opportunistic Network Environment (ONE) simulator [195], which has been designed for terrestrial delay-tolerant communications.

We evaluate the proposed mechanism under the Working Day Movement (WDM) model [196], a realistic mobility model that simulates the daily mobility of average people. We assume that a total of 200 users perform map-based movement within a 4.5 km x 3.4 km section of Helsinki, Finland. Moreover, there exist 5 buses moving on specific routes within this area. People may commute on foot, by car or bus. Mobile users are allocated to a home, a workplace and places of interest (e.g. bars, stores etc.) that they often visit.

We assume that the 3G network is ubiquitous provided by a single operator. Furthermore, there is a private Internet connection available at any home or office; only authorised users utilise private Internet connections. A number of open public WiFi APs is available in popular locations. The uplink data rate of each WiFi AP is 4 Mbps.

We evaluate the impact of traffic with various characteristics to the offloading efficiency of CEMMO, DTO and OTSO. In this work, we emphasise future scenarios where big data uploads become common and CEMMO becomes more relevant. The rise in personal assistance, health monitoring and behavioural intervention applications is expected to lead to an increase in the amount of mobile sensor data (GPS, accelerometer, audio and video) that need to be transferred to remote servers for further processing. We categorise users in three classes based on the amount of data traffic that they generate:

- Low (around 100 MB per day);
- Moderate (around 200 MB per day), and
- Heavy (around 400 MB per day) traffic users.

Each user class is generated using a Gaussian distribution and the traffic load values are based on [88]. In this scenario, we assume that:

- 30% of the users generate low traffic;
- 40% generate moderate traffic, and
- The remaining 30% generate heavy traffic.

The data production rate follows a Pareto distribution. We ignore data that may be produced when a user is at work or home, where an Internet connection is available. We also assume that the storage capacity of the mobile devices is sufficient in all cases, in order to investigate the storage requirements of CEMMO. We consider three traffic types:

- *High priority*. This traffic type comprises 20% of the total mobile traffic and its size varies from 1 KB to 5 MB. Users do not tolerate any delay on the transfer of this traffic type; only OTSO is feasible for this traffic type. Webpage requests and chat conversations fall into this traffic type.
- *Medium priority.* 60% of the total mobile traffic is medium priority. Its size varies from 1 MB to 200 MB and users accept moderate delays. Users accept delays up to 30 minutes for medium priority traffic. Non-urgent e-mails and social networking updates are typical medium priority traffic.
- *Low priority.* This traffic type comprises 20% of the total mobile traffic. Its size varies from 10 MB to 300 MB and users accept increased delays. Users accept delays

ranging from 10 to 40 minutes for low priority traffic, i.e. no low priority traffic is transmitted over 3G prior to the 10-minute threshold. Cloud storage services and mobile sensor data transfers are examples of low priority traffic.

In all experiments, we set region size equal to 300 m x 340 m, similar to [116], and time interval duration equal to 5 minutes; this corresponds to 150 regions and 288 distinct time intervals. The total duration of our experiments is 20 days, which corresponds to 10 days of training, in order to build the prediction model, and 10 days of actual evaluation. The results are presented as averages over 10 simulation runs.

In Scenario 2, we particularly focus on:

a) Evaluating the impact of the public WiFi APs availability on the performance of CEMMO, DTO and OTSO (Subsection 6.2.1). To achieve that, we set β/α ratio equal to 0.04 (i.e. $\alpha=1$, $\beta=0.04$), which is a moderate value and we repeat our simulations for an increasing number of public WiFi APs, ranging from 4 to 20. We keep the number of WiFi APs relatively low since measurements have shown that many "open" APs are not accessible as they apply Media Access Control (MAC) address filtering or web-based authentication for access control.

b) Examining the sensitivity of CEMMO to the overall cost as defined by the operator, which involves the financial and energy cost of the transfers and user dissatisfaction due to delayed transmissions, congestion and increased pricing (Subsection 6.2.2). In particular, we assume that there exist 12 open public WiFi APs and we investigate the sensitivity of the CEMMO performance to the β/α ratio. The increasing β/α ratio corresponds to an increase in the cost of offloading traffic through WiFi APs compared to the cost of 3G data transfer. It is impossible to make accurate estimations on the overall cost of offloading data through WiFi or 3G networks, since we do not have any knowledge on the costs as seen from the perspective of an operator. However, CEMMO provides the mechanism to set and adjust α and β values according to the needs of each operator.

c) Investigating the storage requirements of CEMMO (Subsection 6.2.2). For this reason, we calculate the average and maximum cache size per user required to serve relaying requests.

5.2.3. Scenario 3: Energy optimisation of CEMMO mechanism

The goal of Scenario 3 is to evaluate the energy efficiency of CEMMO, since it is important for user participation. If an offloading scheme only aims to increase the overall offloaded traffic without considering energy consumption on mobile devices, it can quickly drain the battery of the devices. In particular, in this set of experiments, we investigate the capability of CEMMO to adapt its operation based on energy criteria that correspond to the energy consumption of each transfer policy. Table 5.2 summarises the energy consumption values for WiFi-based and 3G-based communications based on [93]. The cellular network interface on mobile devices is typically always on, therefore the total energy consumption for scanning and connection is zero. While the energy consumption per MB of data transfer through WiFi is significantly lower than the corresponding energy consumption of 3G, WiFi-based communications consume additional energy for network discovery and connection. When WiFi operates in the power save mode (PSM), we assume that the total energy required for scanning and connection is, on average, 20 J for each data transfer.

	Data Transfer (J/MB)	**Scanning-Connection (J)**
3G	130	0
WiFi	5	20

Table 5.2 Energy consumption

The estimated energy cost for the data transfer of a single message of size M is:

$$\text{Est. Cost}_{DTO} = M \times (130 \times (1 - P_{DTOsuccess}) + 5 \times P_{DTOsuccess}) + 20$$

$$\text{Est. Cost}_{PAO} = M \times (130 \times (1 - P_{PAOsuccess}) + 2 \times 5 \times N \times P_{PAOsuccess}) + 2 \times N \times 20$$

We note that energy consumption is doubled for PAO since any two peers both consume energy for the restricted flooding of data in the storing region during the storing interval. The energy optimisation of CEMMO is also implemented and evaluated in the ONE simulator.

We evaluate CEMMO, pure DTO and pure OTSO for three distinct delay tolerance profiles: small, medium and large. Table 5.3 summarises the time restrictions for each of the three traffic types. Delays are random and distributed uniformly between the two thresholds, while the lower threshold in each profile signals the threshold prior to which no traffic is transmitted over 3G. We do not use excessive delay tolerance intervals; such intervals would significantly affect user satisfaction.

	High Priority	**Medium Priority**	**Low Priority**
Small	0 sec.	0 - 10 min.	10 - 20 min.
Medium	0 sec.	10 - 20 min.	20 - 30 min.
Large	0 sec.	20 - 30 min.	30 - 40 min.

Table 5.3 Delay tolerance profiles

We assume that there exist 12 open public WiFi APs in popular locations. The uplink data rate of each WiFi AP is 4 Mbps. The remaining communication parameters are identical to Scenario 2.

5.2.4. Scenario 4: Performance evaluation of CARPOOL routing protocol

The goal of Scenario 4 is to evaluate the performance of CARPOOL routing protocol in a dense urban environment and study the impact of increased traffic load on its performance comparatively with four widely-used routing protocols, namely, Epidemic [160], PRoPHET (α=0.75, β=0.25 and γ=0.98) [161], binary Spray-and-Wait with 10 bundle copies [165] and MaxProp [167]. CARPOOL is implemented and evaluated using the ONE simulator. Initially, we create the connectivity plan for the entire simulation using as input:

- The ID and the coordinates of each gateway;
- The ID and the speed of each ferry, along with the gateways on the path of the ferry in the order it transverses them;
- The waiting time at each stop, and
- The start times of each ferry.

We assume that all ferries follow the reverse path once they reach their destination and pause at every stop for a certain period of time. All gateways are aware of the connectivity plan.

For our simulations, we select a topology that corresponds approximately to an abstraction of the transport service of Thessaloniki, Greece, that includes both the city center and the suburbs. In total, our simulation environment covers an area of approximately 100 km^2 that includes 106 offline gateways and 15 online gateways. Our scenarios follow 60 ferries travelling on 20 routes. The speed of the ferries ranges from 5 m/s to 14 m/s. All gateways and ferries are equipped with 2 GB storage size and wireless network cards with 10 Mbps available transmission rate and 50 m communication radius. The overall duration of all simulations is 48 hours, including a sufficient training period for protocols to initialise themselves. The traffic load varies from 2500 to 50000 bundles per 12 hours. Bundle size ranges from 500 KB to 2 MB. Given the delay-tolerant nature of the applications, bundle TTL is set to 20 h, sufficiently large to accommodate all communication attempts by all protocols. An instance of the simulation topology is depicted in Figure 5.1.

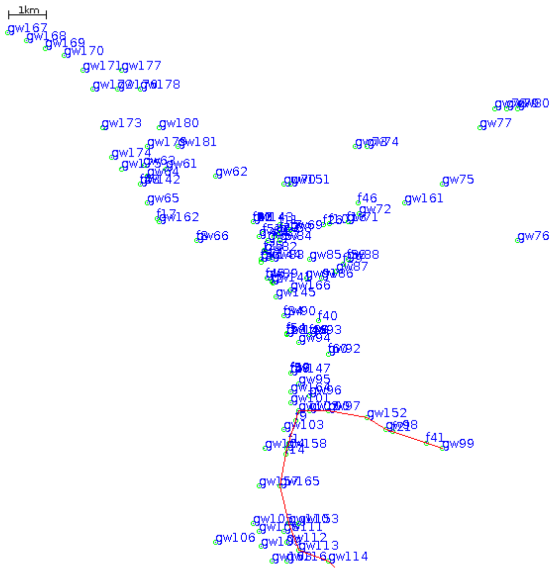


Figure 5.1 An instance of the simulation topology

5.2.5. Scenario 5: Performance evaluation of CARPOOL+ routing protocol

The goal of Scenario 5 is to perform a performance assessment of CARPOOL+ routing protocol in comparison to the original CARPOOL protocol, Epidemic, PRoPHET (α=0.75, β=0.25 and γ=0.98), binary Spray-and-Wait with 10 bundle copies and MaxProp. ONE simulator is used for all experiments. The simulation scenario is based on a map-based model of a city, where 10 ferries follow different paths and traverse 28 gateways in total (25 offline gateways and 3 online gateways). A screenshot of the topology is depicted in Figure 5.2; Table 5.4 shows the route of each ferry. We assume that all ferries follow the reverse path once they reach their destination, pause at every stop for a certain period of time and their speed ranges from 5 m/s to 14 m/s. Offline gateways collect Internet access requests from users within their radius and transfer them to one of the online gateways; the number of online gateways varies for each set of experiments as shown in Table 5.5. For each bundle, a single online gateway is selected as destination. The online gateway that will serve an Internet access request is selected in a round-robin manner, so that traffic load is equally shared among the available online gateways and each online gateway serves the same amount of Internet access requests. In order to ensure that the comparison between CARPOOL+ and other routing solutions is fair, all simulations have been conducted using exactly the same scenario and parameters, including the selection of destination nodes. Bundle size ranges from 500 KB to 20 MB and TTL is set to 1800 s. For all gateways and ferries communication throughput equals to 10 Mbps and communication range is 50 m; these are reasonable values for high-speed, long-range interfaces such as WiFi Direct. We opt for large buffer size since the proposed architecture supports delay-tolerant applications that need to store large volumes of data for long periods, in order to achieve high delivery ratio; the impact of buffer size on the performance of CARPOOL+ is also evaluated. The overall duration of all simulations is 10800 s, while protocols that require initialisation (such as PRoPHET and MaxProp) also include a 5-day training period; through simulations and given the regular ferry schedule and the size of the topology, we have concluded that 5 days of training are enough for these protocols to initialise their probabilities. The initialisation parameters for PRoPHET have been extracted from [161]. In particular, we set the initialisation constant α equal to 0.75, the scaling constant β equal to 0.25 and the aging constant γ equal to 0.98. The remaining protocols do not require any input, since they identify the network topology through training. In all evaluation results, we only consider the amount of data that has been transmitted through our access model, and not through the suggested fallback mechanism. Our aim is to evaluate the performance of CARPOOL+ and compare its ability to efficiently transfer data from the mobile device of an end-user to an online gateway with other popular DTN routing solutions. In order to achieve fair comparison between CARPOOL+ and other DTN routing protocols, Delay Tolerance Threshold was set infinite, since the other routing protocols do not use Delay Tolerance Thresholds. Thus, all protocols are only restricted by TTL. The specific parameters for each set of experiments are summarised in Table 5.5. All evaluation results include 95% confidence intervals extracted over 20 simulation runs.

FerryID	Route
f1	gw11-gw13-gw14-gw21-gw23-gw28-gw27
f2	gw12-gw14-gw16-gw18-gw26-gw29-gw33
f3	gw22-gw24-gw27-gw30-gw33-gw34-gw35-gw32
f4	gw38-gw37-gw36-gw31-gw32-gw30-gw28-gw23
f5	gw19- gw18-gw20-gw25-gw27-gw24-gw17-gw15-gw14-gw12
f6	gw31-gw36-gw37-gw38-gw32-gw30-gw27-gw24-gw22
f7	gw34-gw35-gw33-gw29-gw26-gw25-gw21-gw20
f8	gw19-gw11-gw12-gw13-gw16-gw15-gw17-gw22-gw23-gw28
f9	gw15-gw17-gw22-gw24-gw23-gw30-gw31-gw37
f10	gw19-gw18-gw20-gw21-gw25-gw26-gw34-gw35

Table 5.4 Ferry routes

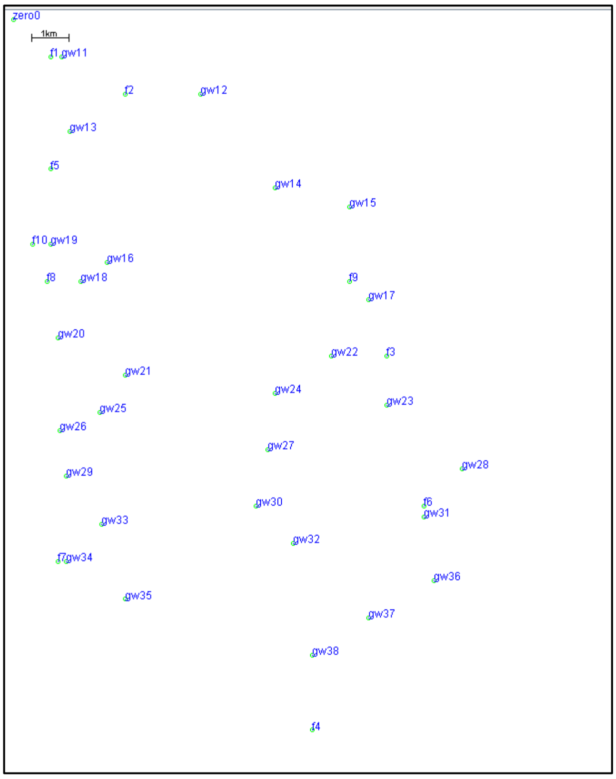


Figure 5.2 Screenshot of the topology

Set of experiments	Number of online gateways	Traffic load (bundles/ gateway/ hour)	Buffer size (GB)	Deviations from schedule due to traffic jams	Protocols under evaluation
1st	3	25-200	2	No	CARPOOL+, CARPOOL
2nd	3	25-200	2	Yes	CARPOOL+, CARPOOL
3rd	3-8	125-200	2	Yes	CARPOOL+
4th	3	25-200	0.5-2	Yes	CARPOOL+
5th	3	25-200	2	Yes	CARPOOL+, Epidemic, PRoPHET, Spray-and-Wait, MaxProp
6th	3-8	125	2	Yes	CARPOOL+, Epidemic, PRoPHET, Spray-and-Wait, MaxProp

Table 5.5 Parameters for each set of experiments

5.3. Evaluation metrics

In this subsection, we describe the metrics that are used in the evaluation analysis of the protocols and mechanisms studied in the framework of the present thesis.

5.3.1. General performance metrics

We evaluate performance using a variety of metrics, such as link efficiency, fairness index, packet loss probability and queuing delay index. Each of the aforementioned metrics is defined as follows:

1. *Average link efficiency (η)* expresses the average link utilisation as the ratio between the throughput sum of all flows over the available capacity, as shown below:

$$\eta = \frac{\sum_{i=1}^{N} \text{Throughput}(i)}{C} \qquad (5.1)$$

2. *Average Jain's fairness index (F)* determines whether flows are receiving a fair share of the available resources and is calculated as follows:

$$F = \frac{\left(\sum_{i=1}^{N} \text{Throughput}(i)\right)^2}{N \sum_{i=1}^{N} \text{Throughput}(i)^2} \qquad (5.2)$$

3. *Average packet loss probability (P_l)* is calculated as the average ratio of dropped packets over the total number of packets sent over the link, as shown below:

$$P_l = \frac{\text{Total packets dropped}}{\text{Total packets sent}} \qquad (5.3)$$

4. *Average queuing delay index (D_Q)* is calculated normalising the mean queuing delay during the simulation over the maximum theoretical queuing delay (which depends on buffer size). The maximum D_Q value is 1 and it is calculated as follows:

$$D_Q = \frac{\text{Average Queuing Delay}}{\text{Maximum Queuing Delay}} \qquad (5.4)$$

5. *Average traffic load distribution* is the percentage of resources that each type of flow consumes, when flows that use different access methods co-exist in the same link.

5.3.2. Offloading performance metrics

We evaluate offloading performance using the following metrics:

1. *Offloading ratio* expresses the fraction of the total amount of generated data that is offloaded through WiFi networks and is calculated as shown below:

$$\text{Offloading Ratio} = \frac{\text{Total Data Offloaded through WiFi}}{\text{Total Data Generated}} \qquad (5.5)$$

2. Average Cost per MB is calculated as the sum of the total data (in MB) that was transferred through each transfer policy multiplied by the cost of each policy, divided by the total data transferred, as follows:

$$\text{Avg. Cost per MB}_{OTSO} =$$

$$\text{Avg. Cost per MB}_{DTO} = \frac{\alpha \text{ x Data Transmitted 3G} + \beta \text{ x Data transmitted WiFi}}{\text{Total Data Transmitted}} \quad (5.6)$$

$$\text{Avg. Cost per MB}_{CEMMO} = \frac{\text{Avg. Cost per MB}_{DTO} + \beta \text{ x N x Data transmitted PAO}}{\text{Total Data Transmitted}} \quad (5.7)$$

3. *Average Cache Size* is calculated as a ratio of the maximum storage used in all mobile devices for PAO to the number of the mobile devices involved in PAO, as shown below:

$$\text{Avg. Cache Size} = \frac{\sum_{i=1}^{\text{Number of devices}} \text{Maximum device storage used for PAO}_i}{\text{Mobile users involved in PAO}} \quad (5.8)$$

4. *Maximum Cache Size* is the maximum storage used by a mobile device for PAO, as shown below:

$$\text{Max. Cache Size} = \max_{\text{Number of devices}} \text{Device storage used for PAO} \quad (5.9)$$

5.3.3. Routing performance metrics

We evaluate the performance of routing protocols using the following metrics:

1. *Delivery ratio* expresses the fraction of the total generated bundles that are successfully delivered and is calculated as shown below:

$$\text{Delivery ratio} = \frac{\text{Number of bundles successfully delivered}}{\text{Number of bundles generated}} \quad (5.10)$$

   The amount of data that cannot be delivered through the proposed access model and needs to be transferred over the available basic cellular connectivity, or prioritized as soon as a gateway or ferry estimates that data cannot be delivered before their Delay Tolerance Threshold or TTL expires, can be calculated as (1 – Delivery probability); we expect these data to constitute only a small fraction of the overall transmitted data.

2. *Overhead ratio* is calculated as the number of bundles relayed minus the number of bundles delivered to the number of bundles delivered, as follows:

$$\text{Overhead Ratio} = \frac{\text{Number of bundles relayed-Number of bundles delivered}}{\text{Number of bundles delivered}} \quad (5.11)$$

3. *Median latency* is calculated as the numerical value separating the higher half of all bundle latencies from the lower half.

4. *Average latency* is measured as the average time between each bundle creation and delivery time and is calculated as follows:

$$\text{Median Latency} = \frac{\sum_{i=1}^{\text{Number of bundles successfully delivered}} \text{Latency}_i}{\text{Number of bundles successfully delivered}} \quad (5.12)$$

5. *Average hop count* is calculated as the average number of hops a bundle crosses before it is delivered to its recipient.

$$\text{Avg.Hop Count} = \frac{\sum_{i=1}^{\text{Number of bundles successfully delivered}} \text{Number of hops until destination}_i}{\text{Number of bundles successfully delivered}} \quad (5.13)$$

5.4. Experimentation tools

In the framework of the evaluation analysis, fLEDBAT access method is implemented and evaluated in NS-2 version 2.35 based on [190]. All offloading access methods (i.e. OTSO, DTO and PAO), as well as CEMMO mechanism are implemented and evaluated in the ONE simulator. CARPOOL and CARPOOL+ routing protocols are also implemented and evaluated in ONE simulator. Below we summarise the most important characteristics of these tools.

5.4.1. Network Simulator 2

NS-2 is a popular open-source network simulator that supports a wide variety of applications, transport and routing protocols, data link protocols, error models and network topologies. NS-2 is a discrete event simulator, where simulation actions are connected to a sequence of time events controlled by a simulation clock. NS-2 supports event monitoring and tracing during simulation and allows for the extraction of useful conclusions during the evaluation of the results. This simulator is easily extendable and allows users to implement new mechanisms and protocols in different network layers, in order to support diverse research targets.

NS-2 is implemented in two object-oriented programming languages: *C++* and *OTcl*. OTcl is the object-oriented version of Tcl language and acts as UI to set up simulations. C++ is used as backend and runs the actual simulation. The reason behind using two programming languages instead of one is the fact that each language is used for different purposes. On the one hand, C+++ is used for the fast and efficient conduction of the experiments. On the other hand, OTcl may be slower, but has the advantage that it allows for faster code changes, since it is an interpreted programming language and changes in code do not require compilation.

Therefore, OTcl is used for the quick completion of tasks such as network topology creation, network condition setting and selection of various protocols.

5.4.2. Opportunistic Network Environment simulator

The ONE simulator has been specifically designed for evaluating DTN routing and application protocols. The ONE simulator is a java-based tool offering a broad set of DTN protocol simulation capabilities in a single framework. At its core, ONE is an agent-based discrete event simulation engine. At each simulation step the engine updates a number of modules that implement the main simulation functions.

The main functions of the ONE simulator are the modelling of node movement, inter-node contacts, routing and message handling. Result collection and analysis are performed through visualisation, reports and post-processing tools. Movement models, which can be either synthetic models or existing movement traces, implement node movement. Connectivity between nodes is based on their location, communication range and transmission rate. The routing function is implemented by routing modules that decide which messages to forward over existing contacts. Finally, the messages themselves are generated through event generators.

Simulation results are collected primarily through reports generated by report modules during the simulation run. Report modules receive events (e.g., message or connectivity events) from the simulation engine and generate results based on them. The results generated may be logs of events that are then further processed by external post-processing tools, or aggregate statistics calculated in the simulator. A GUI displays a visualisation of the simulation state showing the locations, active contacts and messages carried by the nodes.

Overall, the ONE simulator offers an extensible simulation framework that supports mobility and event generation, message exchange, DTN routing and application protocols, a basic notion of energy consumption, visualisation and analysis, as well as interfaces for importing and exporting mobility traces, events, and entire messages. The ONE simulator is designed in a modular fashion, allowing extensions of virtually all functions to be implemented using well-defined interfaces.

6. Evaluation results

In this chapter, we evaluate the performance of the mechanisms and protocols proposed in the present thesis. In particular, we present our evaluation results separately for the five scenarios described in our evaluation analysis.

6.1. Scenario 1: Performance evaluation of TCP New Reno, LEDBAT and fLEDBAT access methods in the sub-packet regime

The goal of Scenario 1 is to investigate if a LBE access method that is typically less aggressive than TCP can better utilise the limited available bandwidth in heavily shared backhaul links. Given the bandwidth-delay product in each case, the number of parallel flows might not be enough to drive the link into the sub-packet regime. For this reason, each figure is separated into two parts through a vertical dotted line: in the left part the network has not reached the sub-packet regime yet, while in the right part the number of flows is enough to drive the network in the sub-packet regime.

6.1.1. Comparison of TCP, LEDBAT and fLEDBAT access methods when flows of a single type occupy the link

We start our analysis by considering an increasing number of concurrent flows that use the same access method (i.e. TCP NewReno, LEDBAT or fLEDBAT) and we repeat the experiments using both DropTail and RED as queue management algorithms, in order to validate the impact of AQM on low priority congestion control as briefly described in [197]. The parameters of this scenario are provided in Chapter 5.

6.1.1.1. Average link efficiency

In order to demonstrate the effect of increasing traffic load on link utilisation, we first plot link efficiency for the uplink (Equation 5.1). Figure 6.1 shows that the uplink is almost always fully utilised in all scenarios described in Table 5.1. When we are not in the sub-packet regime, fLEDBAT underutilises bandwidth that can be exploited by other flows. When we enter the sub-packet regime, we notice that both LEDBAT and fLEDBAT achieve higher link efficiency than TCP. In essence, TCP tries to transmit a significant amount of data, but fails due to the fully utilised link, resulting in timeouts and retransmissions. LEDBAT and fLEDBAT do not timeout as often as TCP by being less aggressive and transmitting less data. This leads to significantly less retransmitted packets and higher link efficiency. The results are similar when we use RED as an AQM scheme, as shown in Figure 6.2.

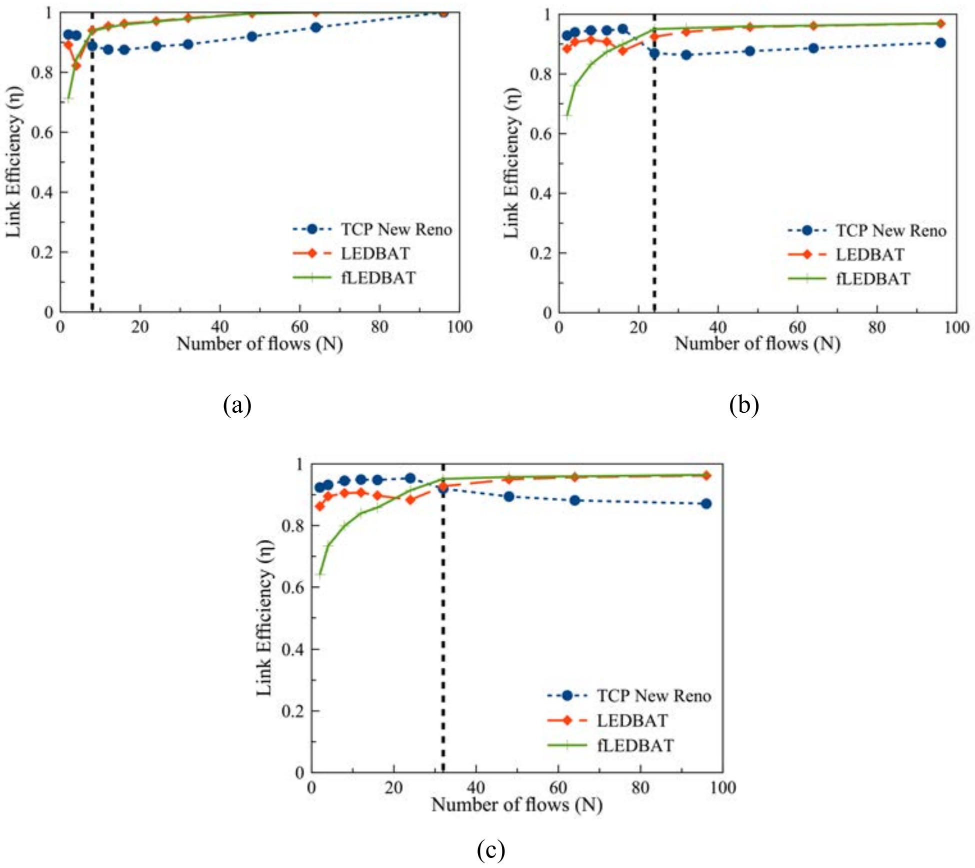


Figure 6.1 Average link efficiency (η) using DropTail for cases (a)-(c)

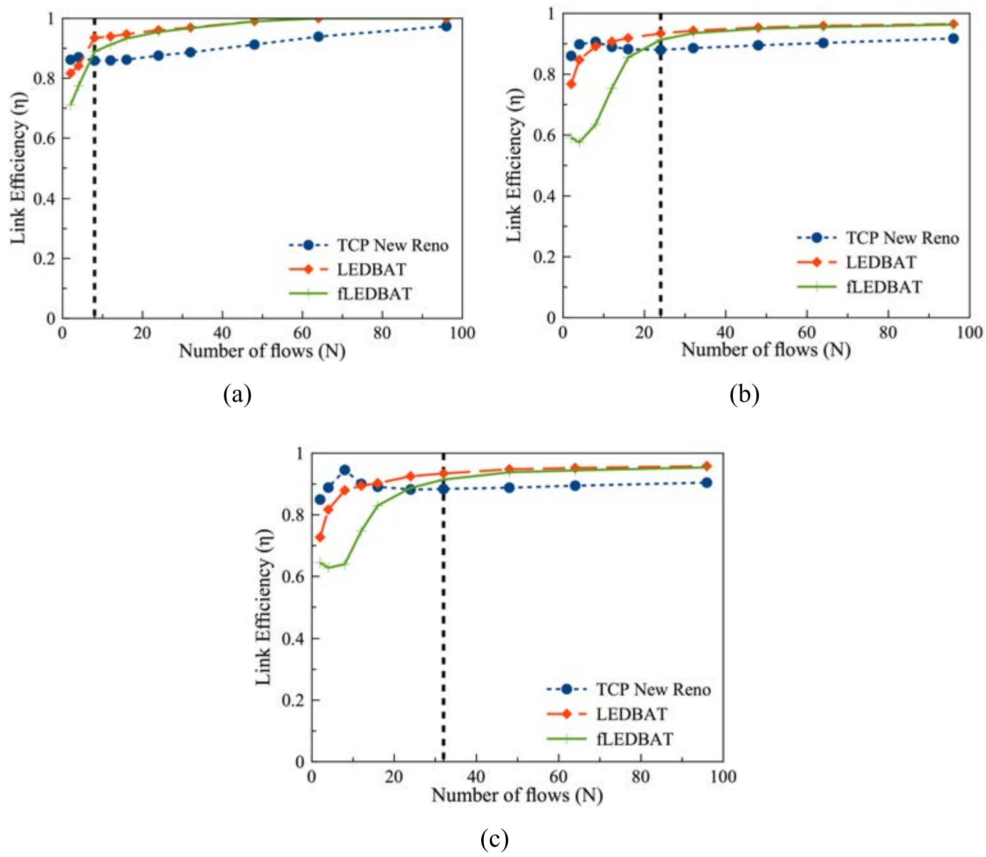


Figure 6.2 Average link efficiency (η) using RED for cases (a)-(c)

6.1.1.2. Average Jain's fairness index

To explore the effect of increasing traffic load on the fairness of each protocol, we plot the average Jain's fairness index (Equation 5.2) of 30 runs for an increasing number of flows for all three backhaul links described in Table 5.1. Figure 6.3 shows that in all cases fairness among fLEDBAT flows is significantly higher than TCP NewReno flows; by being less aggressive, fLEDBAT manages to distribute the available resources more equally among flows. When we are not in the sub-packet regime, LEDBAT fairness is particularly low due to the "late-comer advantage" that was discussed in Chapter 2. fLEDBAT was proposed as a solution to the "late-comer advantage" and, as seen in Figure 6.3, it indeed solves this problem.

When we enter the sub-packet regime, the network is full and packets from all flows are being dropped. Therefore, the "late-comer advantage" is not present in the sub-packet regime. Another important observation from Figure 6.3(a) is that fairness significantly decreases as the number of flows increases. This is the result of a large number of flows sharing a small bandwidth-delay product link; small buffers cannot hold enough packets from all flows, thus resulting in significant unfairness among flows [198]. We also notice that RED solves the "late-comer advantage" problem, as shown in Figure 6.4.

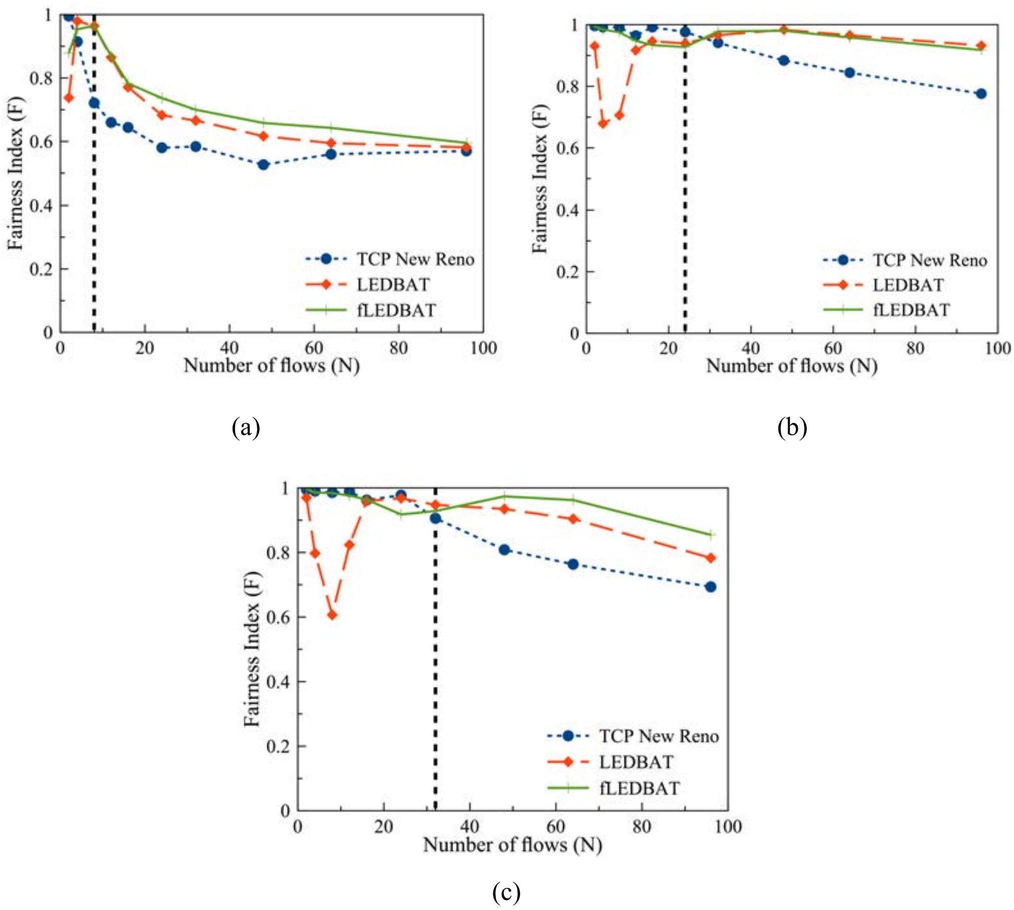
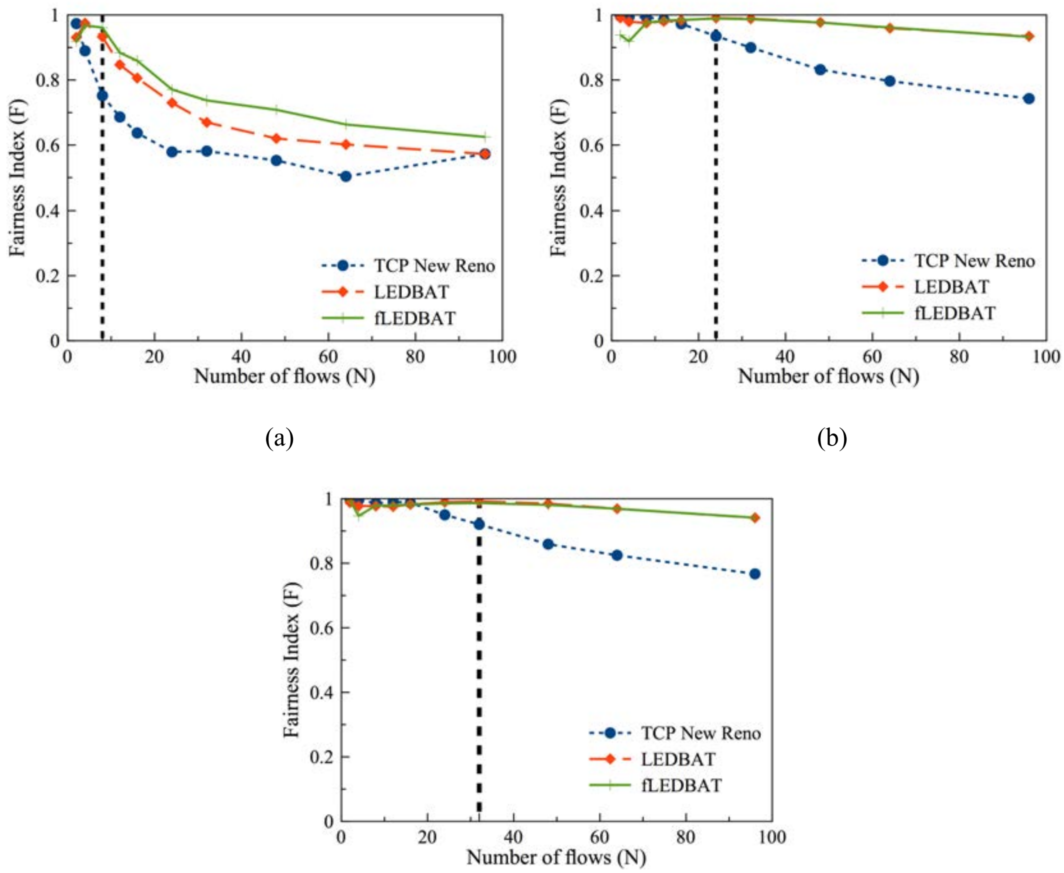


Figure 6.3 Average fairness index (F) using DropTail for cases (a)-(c)



(a) (b)

(c)

Figure 6.4 Average fairness index (*F*) using RED for cases (a)-(c)

6.1.1.3. Average packet loss probability

Figure 6.5 and Figure 6.6 capture the average packet loss probability (Equation 5.3) for each scenario by increasing traffic load for DropTail and RED, respectively. In Figure 6.5, when we are not in the sub-packet regime, the results show that TCP flows encounter higher packet loss probability than LEDBAT and fLEDBAT flows in all scenarios described in Table 5.1, since TCP flows produce significantly more data. When we are in the sub-packet regime, the total data rate produced by all flows surpasses the capacity of the link, leading to increased packet loss probability.

All access methods converge towards the same average packet loss probability for increasing number of flows, since flows transmit more and more packets that the buffers cannot store. This convergence is more gradual for cases (b) and (c), where the bandwidth-delay product and the buffer sizes are larger. In all cases, given the small buffer sizes and the sub-packet regime, packet loss probability is significant. All simulations are repeated using RED as depicted in Figure 6.6. As expected, in this case the average packet loss probability is higher, since RED drops packets even if the buffer is not full yet. Due to the small buffer size in case (a), there is no significant difference between Figure 6.5(a) and Figure 6.6(a).

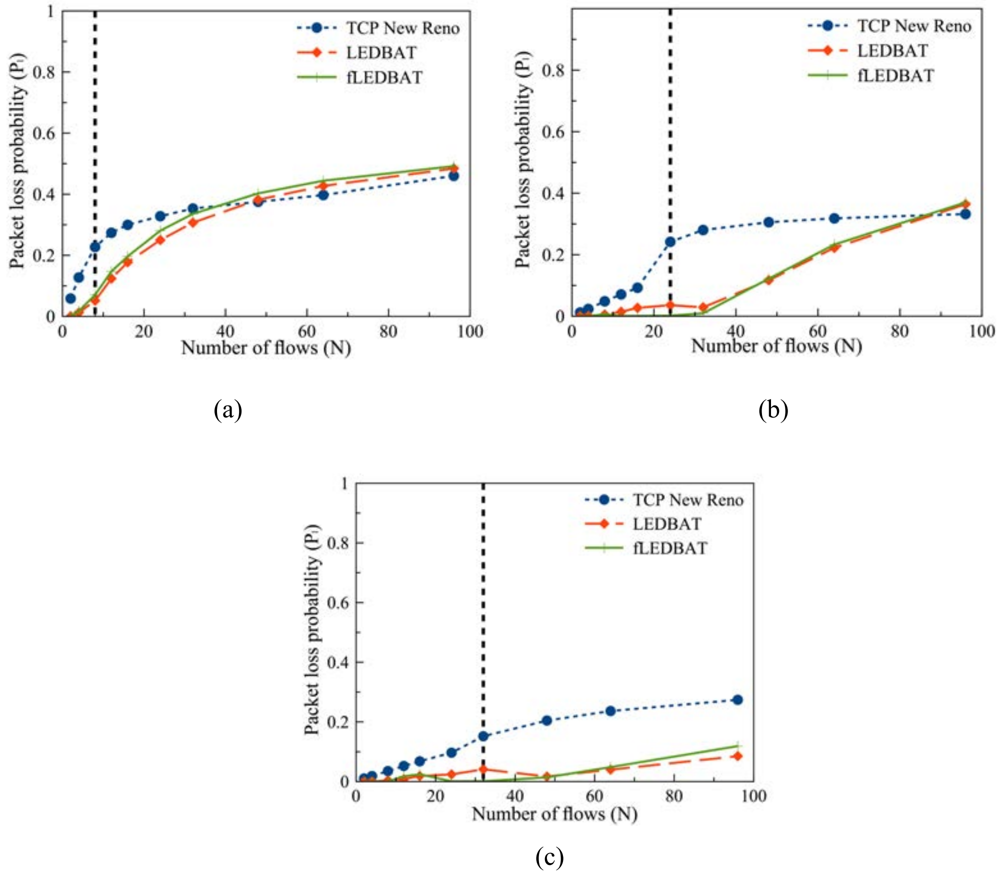


Figure 6.5 Average packet loss probability (P_l) using DropTail for cases (a)-(c)

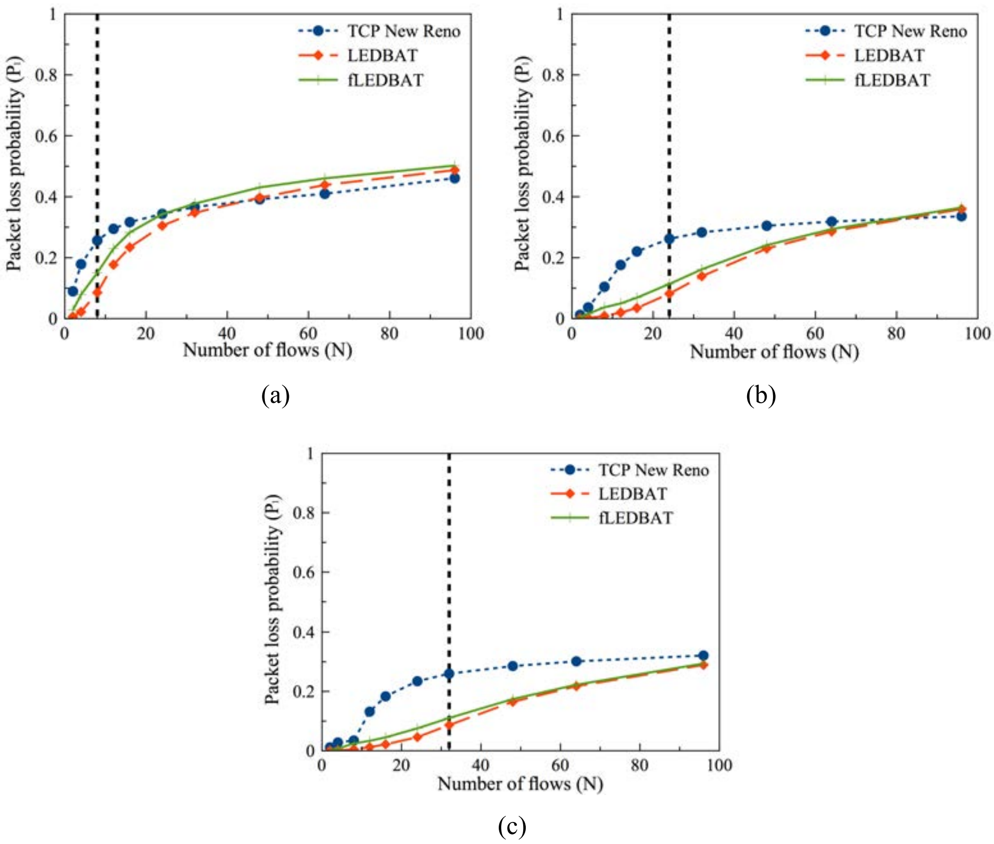
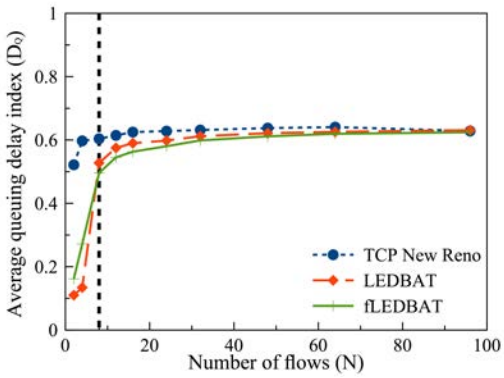


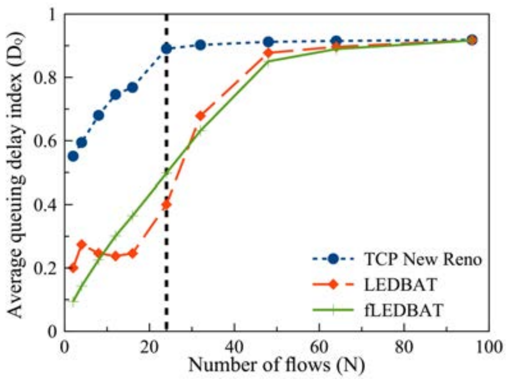
Figure 6.6 Average packet loss probability (P_l) using RED for cases (a)-(c)

6.1.1.4. Average queuing delay index

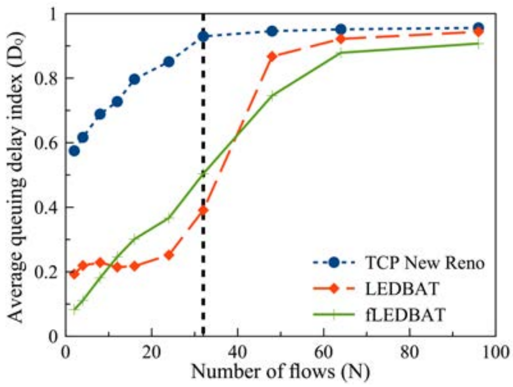
In this subsection, we study how the increasing traffic load in the uplink affects the average queuing delay of the flows (Equation 5.4). Figure 6.7 depicts the average queuing delay index when we use DropTail. When we are not in the sub-packet regime, in all cases LEDBAT and fLEDBAT achieve significantly less queuing delay than TCP. When we enter the sub-packet regime, all access methods converge to the same average queuing delay. Another important observation is the fact that in Figure 6.7(a), the maximum queuing delay index is never reached. As explained in [198], if the bottleneck buffer is not large enough to accommodate an identical number of packets from all competing flows, there are difficulties in measuring the equilibrium queuing delay. Cases (b) and (c) that satisfy the aforementioned requirement, reach the maximum queuing delay index, which depends on the buffer size, as shown in Figure 6.7(b) and Figure 6.7(c).



(a)



(b)



(c)

Figure 6.7 Average queuing delay index (D_Q) using DropTail for cases (a)-(c)

The simulations are repeated using RED in Figure 6.8. Compared to Figure 6.7, we notice that in all cases the average queuing delay index is significantly lower when RED is used, since buffer capacity is not fully utilised, rather packets are dropped even when the buffer is not full. The average queuing delay index in Figure 6.8 is dependent on RED minimum and maximum thresholds. When we are not in the sub-packet regime, TCP NewReno presents slightly higher average queuing delay. Using RED in the sub-packet regime, all access methods present the same average queuing delay.

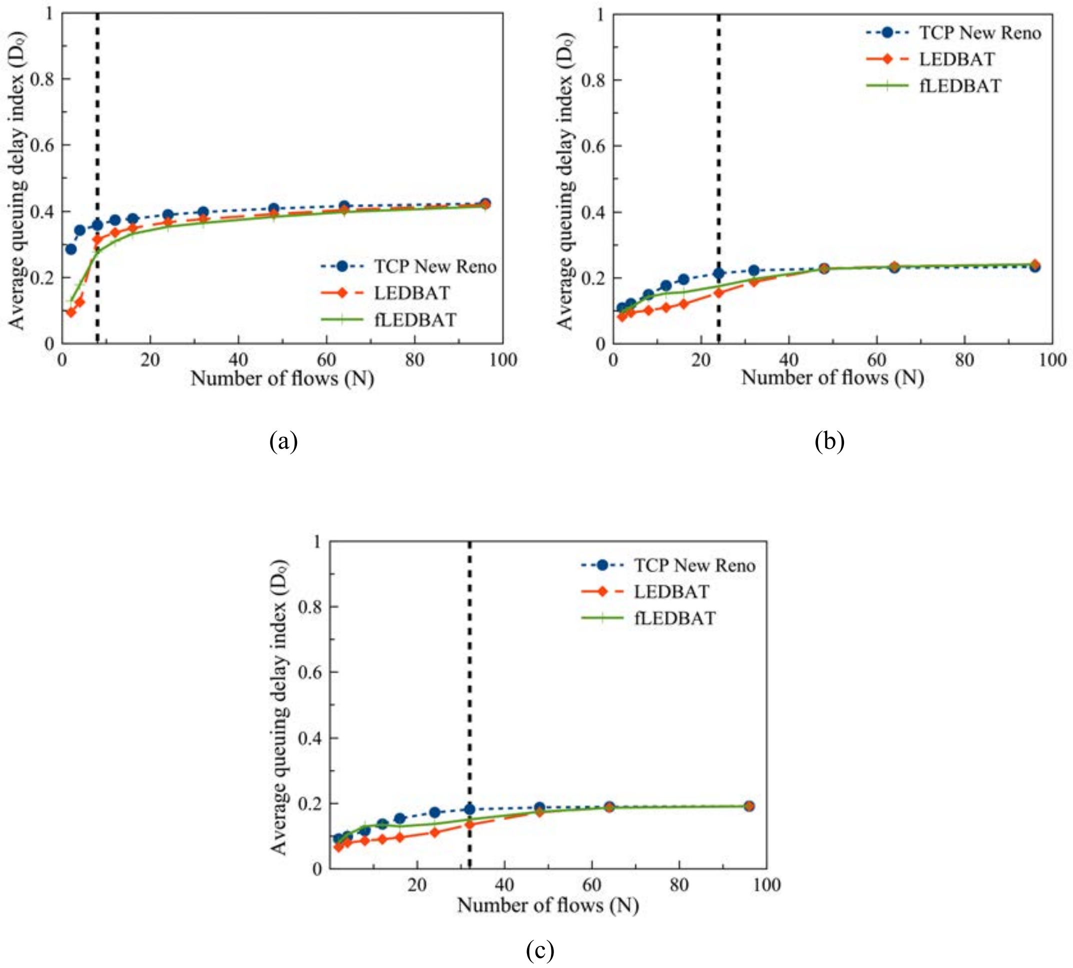


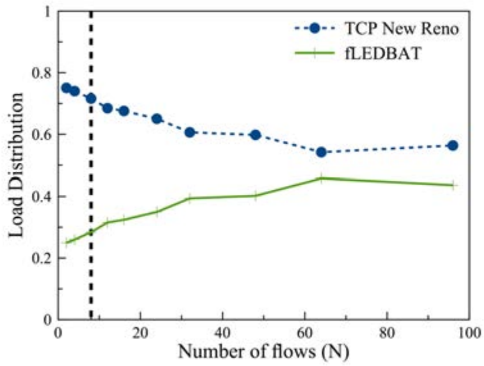
Figure 6.8 Average queuing delay index (D_Q) using RED for cases (a)-(c)

6.1.2. Distribution of resources when TCP and fLEDBAT flows share the same link

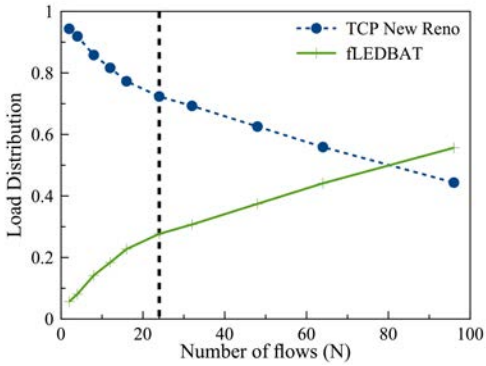
In the second set of experiments, flows are equally divided into TCP NewReno and fLEDBAT flows that share the same link; the number of flows in the following figures refers to the total number of flows, equally divided into TCP NewReno and fLEDBAT flows. The aim of these scenarios is to investigate whether fLEDBAT satisfies its design principles by yielding to TCP flows in the sub-packet regime. For this reason, we study the load distribution between TCP and fLEDBAT flows in cases (a)-(c) for increasing traffic load using DropTail, as depicted in Figure 6.9. Intuitively, we would expect a high TCP share and a low LEDBAT share. This holds only for Figure 6.9(c), where both buffer size and bandwidth are significant. Due to the small buffer size and the restricted resources available in Figure 6.9(a), we notice that even a few fLEDBAT flows consume a significant part of resources. Moreover, Figure

6.9(a) shows that as the number of total flows increases, fLEDBAT flows become more aggressive, consuming almost equal share of resources to TCP.

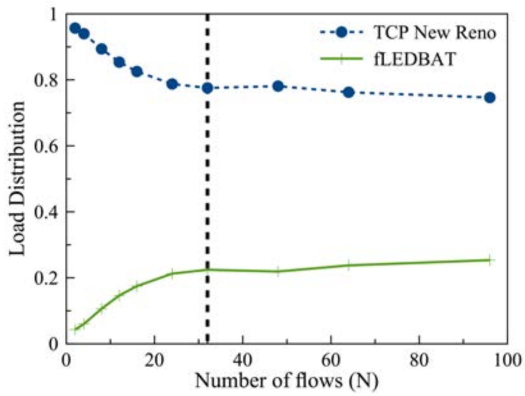
Even worse, in Figure 6.9(b), fLEDBAT flows become extremely aggressive, consuming even more resources than TCP when more than 80 flows share the link. This aggressiveness is the result of the incorrect base delay estimation of fLEDBAT, due to the standing buffer queue. In essence, fLEDBAT flows that enter the network when we are already in the sub-packet regime do not measure the actual base delay (i.e. when the buffer is empty), but the OWD when the buffer is already full. Therefore, these flows assume that there is room to increase their congestion window, becoming very aggressive when they should not. This incorrect base delay estimation [86] is obvious in all cases, where the load share of fLEDBAT gradually increases.



(a)



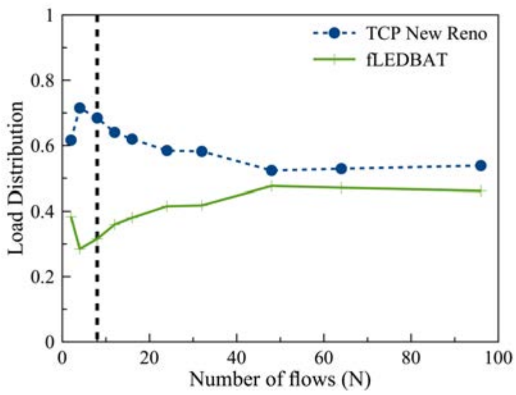
(b)



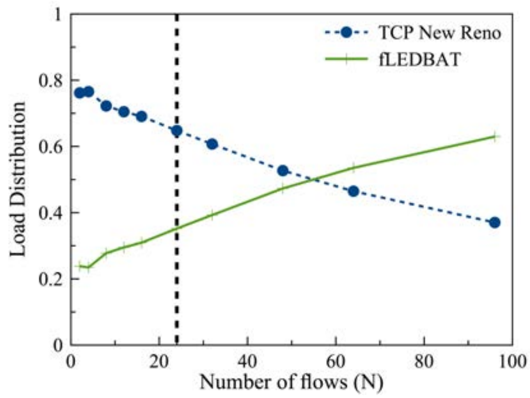
(c)

Figure 6.9 Load Distribution between TCP NewReno and fLEDBAT flows using DropTail for cases (a)-(c)

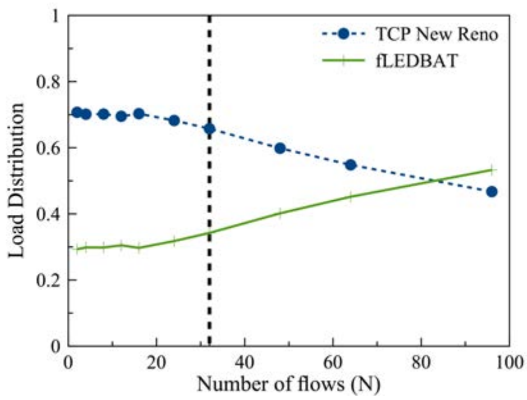
All simulations are repeated using RED for AQM and the results are depicted in Figure 6.10. Due to the small buffer size in case (a), we observe no significant change in the load distribution between Figure 6.9(a) and Figure 6.10(a). The packets drops induced by RED in Figure 6.10(b) and Figure 6.10(c), result in fluctuations in the OWD measured by fLEDBAT flows. This delay variability is misinterpreted and fLEDBAT flows constantly increase their congestion window, becoming more and more aggressive. We see that fLEDBAT flows consume more resources than TCP NewReno flows for 50 flows in case (b) (Figure 6.10(b)) or 80 flows in case (c) (Figure 6.10(c)). The fact that AQM totally jeopardises the mechanisms of scavenger transport methods has been first proposed in [197].



(a)



(b)



(c)

Figure 6.10 Load Distribution between TCP NewReno and fLEDBAT flows using RED for cases (a)-(c)

6.1.3. Discussion and recommendations

The motivation behind this work is to investigate the suitability of a scavenger access method as an access method for backhaul links of WCNs in developing regions, where a low-bandwidth link is usually shared among a large user base, resulting in sub-packet regimes. Based on the simulation results, we show that while out of the sub-packet regime LEDBAT and fLEDBAT flows are more conservative than TCP flows, thus underutilising the available resources, when we enter the sub-packet regime they present less retransmissions (that increase contention in an already congested link) and, therefore, timeout less often achieving higher link efficiency. Moreover, fLEDBAT access method performs better resource distribution among its flows compared to TCP NewReno, achieving increased fairness in all cases. If the link buffers are large enough to accommodate packets from all flows, LEDBAT and fLEDBAT also achieve lower packet loss probability.

Through simulations we show that when we are not in the sub-packet regime, fLEDBAT flows satisfy their design principles and yield to TCP flows. When TCP NewReno and fLEDBAT flows share the same link in the sub-packet regime, fLEDBAT flows fail to measure the actual base delay due to the standing queue and become aggressive, consuming more and more resources. In order for fLEDBAT to function properly in the sub-packet regime when competing with TCP flows, new ways to estimate base delay need to be developed. Shared bottleneck detection mechanisms have been proposed in literature [199], however no real-life validation has taken place so far. Moreover, a conservative reaction to consecutive timeouts, which are typical in the sub-packet regime, needs to be incorporated into fLEDBAT.

Due to their ack-clocking design, both TCP and LEDBAT access methods encounter high packet loss rates and variable timeout periods when multiple flows share a highly congested link. The authors of [70] propose to prioritise retransmitted packets in order to improve performance in the sub-packet regime, since retransmissions are important and repeated drops cause TCP to collapse. Admission control could also be used to keep TCP in good operation range with loss rates less than 10 percent.

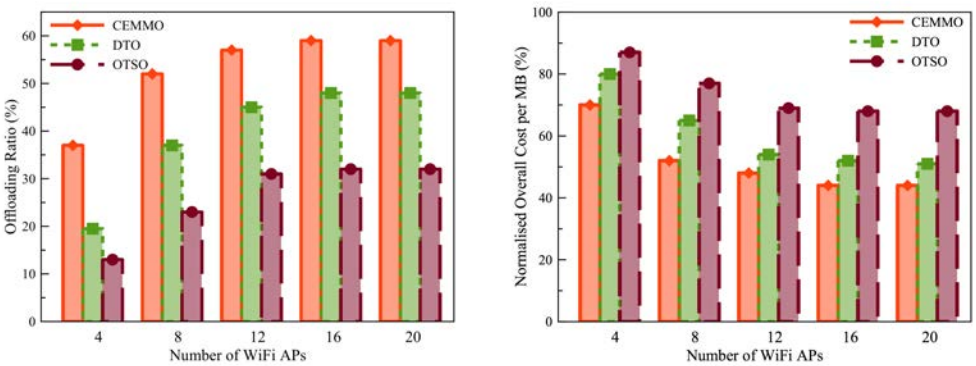
We conclude that a LBE access method cannot be the solution to the increasing network load, especially as far as cellular networks are concerned. For this reason, mechanisms to decongest cellular networks need to be developed. Data offloading is so far the most popular approach to offload highly congested cellular networks through WiFi APs. However, the overall gains of cellular data offloading depend on the amount of available WiFi APs. In an effort to increase offloading opportunities, we have proposed the first peer-assisted offloading transfer policy independent of content and popularity and we have coupled it with CEMMO mechanism that selects the most cost-effective policy for each transfer. Both PAO and CEMMO are evaluated in Scenario 2.

6.2. Scenario 2: Performance evaluation of CEMMO mechanism compared to pure OTSO and pure DTO

The goal of Scenario 2 is to extensively evaluate CEMMO mechanism in comparison to pure OTSO and pure DTO. We aim to highlight the advantage of a mechanism that selects the most cost-effective offloading policy in terms of cost, as defined by each operator, as well as the gains of the new PAO transfer policy. First, we focus on evaluating the impact of the public WiFi APs availability on the performance of CEMMO, DTO and OTSO. Next, we examine the sensitivity of CEMMO to the overall cost as defined by the operator, which involves the financial and energy cost of the transfers and user dissatisfaction due to delayed transmissions, congestion and increased pricing.

6.2.1. Impact of public WiFi APs availability

In this experiment, we evaluate the impact of the public WiFi APs availability on the performance of CEMMO, pure DTO and pure OTSO. In particular, we increase the number of available WiFi APs from 4 to 20 using a step of 4; we consider the obtained results sufficient to draw meaningful conclusions. In Figure 6.11, we present the offloading ratio (Equation 5.5) and the normalised data transfer cost per MB for the three transfer policies (Equations 5.6-5.7) for an increasing number of public WiFi APs, ranging from 4 to 20. According to Figure 6.11, CEMMO is less sensitive to WiFi APs availability and manages to offload around 36% of the mobile traffic even in scenarios with sparse availability. Under the same conditions, DTO offloads 19% of the total mobile traffic, while OTSO only offloads 12% of the traffic. In scenarios with extended WiFi coverage, CEMMO offloads up to 59% of mobile data traffic. CEMMO also outperforms the other mechanisms under investigation in terms of normalised overall data transfer cost per MB. The performance of all mechanisms improves as WiFi availability increases, since more data are offloaded through low-cost WiFi networks.



(a) (b)

Figure 6.11 Impact of WiFi APs availability on offloading ratio (a) and overall transfer cost (b)

Figure 6.12 depicts the distribution of the data offloaded through CEMMO. When the number of available WiFi APs is low, almost half of the overall traffic is offloaded through PAO. For increased WiFi availability, the probability that a user will encounter a WiFi AP to offload data directly increases and, therefore, more data are offloaded through DTO. The gains of incorporating a PAO method that offloads data independent of content and popularity are obvious in Figure 6.12.

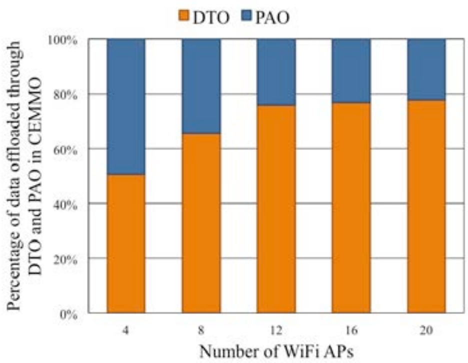
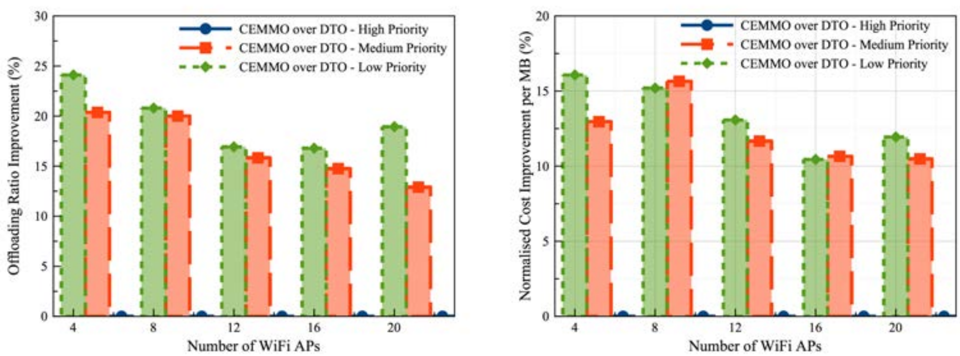


Figure 6.12 Distribution of the offloaded data in CEMMO

Figure 6.13 depicts the relative improvement of the offloading ratio and transfer cost per MB of CEMMO over pure DTO for the three different traffic types. Users accept delays up to 30 minutes for medium priority traffic, and delays ranging from 10 to 40 minutes for low priority traffic, i.e. no low priority traffic is transmitted over 3G prior to the 10-minute threshold. Users do not tolerate any delay for high priority traffic; only OTSO is feasible for this traffic type. The performance improvement of CEMMO over DTO for high priority traffic is zero or even slightly negative, since in some cases users may unsuccessfully attempt to transfer data during a short connection to a public WiFi hotspot. CEMMO significantly outperforms DTO for medium and low priority traffic types. We observe that CEMMO offloads more traffic when users are willing to accept larger delays. CEMMO offloads up to 24% more low priority traffic than DTO and reduces the transfer cost per MB up to 16%. We also notice that the gains of CEMMO over DTO slightly vary depending on the number of available WiFi APs; these measures depict the results given by a specific model and we expect them to follow a smooth pattern when simulations are repeated multiple times.

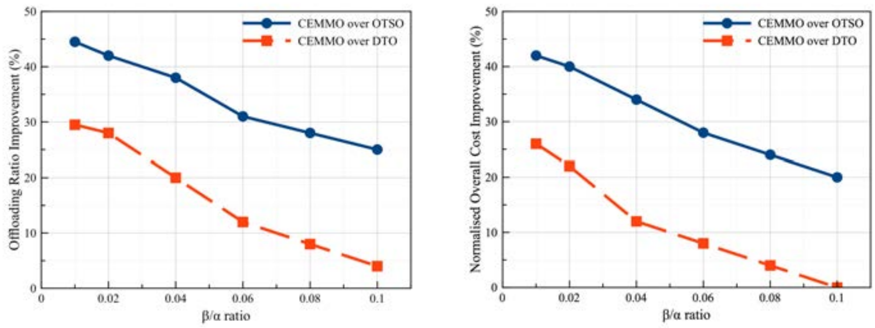


(a) (b)

Figure 6.13 Offloading ratio (a) and cost improvement (b)

6.2.2. Impact of β to α ratio

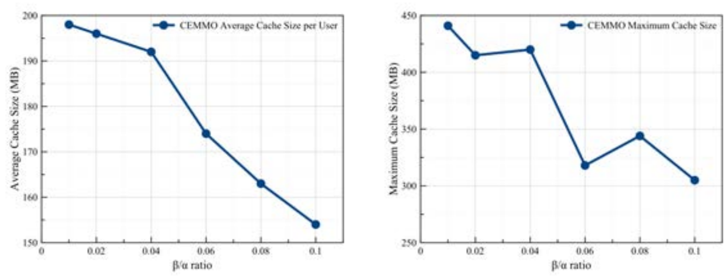
In this set of experiments, we investigate the improvement of CEMMO in terms of offloading ratio and overall transfer cost over pure DTO and pure OTSO for an increasing β/α ratio. Figure 6.14 shows that when β is small (i.e. β/α=0.01), CEMMO offloads 44% more data than OTSO and 29% more data than DTO and reduces data transfer cost up to 42% and 26%, respectively. For the increasing β/α ratio, we observe that the number of mobile users that are utilised to achieve cost-efficient PAO is reduced and, therefore, the offloading ratio of CEMMO is reduced. For large β values, CEMMO significantly reduces the amount of traffic that is offloaded through PAO and matches DTO in terms of data transfer cost per MB.



(a) (b)

Figure 6.14 Offloading ratio (a) and overall cost (b) improvement for the increasing β/α ratio

Considering the limited storage resources of mobile devices, it is essential to keep low the amount of data that needs to be cached during PAO. In Figure 6.15, we evaluate CEMMO in terms of storage requirements. In particular, we depict the average and maximum cache size (Equations 5.8-5.9) per user required to serve data relaying requests. On average, CEMMO requires up to 200 MB of cache for small β/α ratio, while the required cache size is reduced up to 45 MB for larger β values, since less data are offloaded through PAO. We observe that CEMMO has low storage requirements due to the spatio-temporal restrictions of the flooding process; in the worst case scenario CEMMO only requires 440 MB of cache for its operation. We also notice variations in the maxumum cache size; this is justified by the fact that we consider the maximum cache size used by any user in a simulation. In contrast, we notice that the average cache size per user follows a smooth pattern for increasing β/α ratio.



(a) (b)

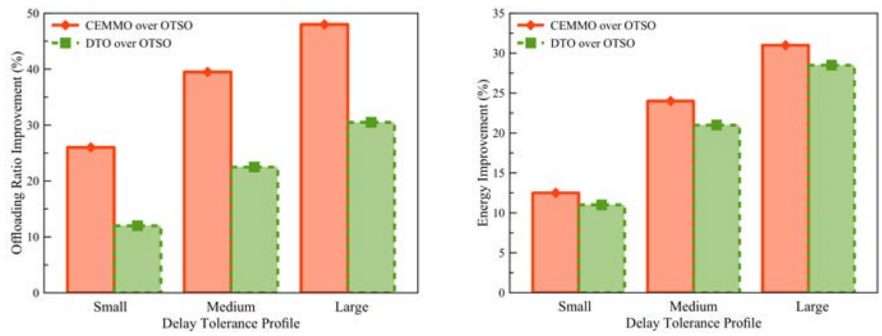
Figure 6.15 Average (a) and maximum (b) cache size of CEMMO for the increasing β/α ratio

We conclude that CEMMO has low storage requirements and can offload significantly more data than pure OTSO and pure DTO with reduced cost. In order to satisfy the requirements of urgent or real-time data, CEMMO opts for direct delivery over the available cellular network in these cases. We also observe that in the worst-case scenario, CEMMO matches the gains DTO. Moreover, the cellular network operator has the ability to control the amount of data offloaded through other peers by adjusting the β/α ratio.

6.3. Scenario 3: Energy optimisation of CEMMO mechanism

The goal of Scenario 3 is to evaluate the energy efficiency of CEMMO, since it is important for user participation. If an offloading scheme only aims to increase the overall offloaded traffic without considering energy consumption on mobile devices, it can quickly drain the battery of the devices. In particular, in this set of experiments, we investigate the capability of CEMMO to adapt its operation based on energy criteria that correspond to the energy consumption of each transfer policy.

In Figure 6.16, we present the relative improvement of CEMMO and DTO over OTSO, in terms of offloading ratio and energy. CEMMO and DTO improve the offloading ratio up to 47% and 30%, respectively, over OTSO for the large delay tolerance profile. While PAO consumes additional energy for P2P transfers, CEMMO consumes less total energy than OTSO and DTO, since less data need to be transferred over energy-intensive 3G communications. CEMMO achieves up to 31% improvement on energy consumption over OTSO for the large delay tolerance profile, while the corresponding improvement for DTO over OTSO is 28%. Similar conclusions apply for small and medium delay tolerance profiles.



(a) (b)

Figure 6.16 Offloading ratio (a) and energy (b) improvement

Next, we investigate the energy consumption separately for each user. In Figure 6.17, we present the Complementary Cumulative Distribution Function (CCDF) of the energy improvement of CEMMO and DTO over OTSO for each delay tolerance profile. According to our simulation results, CEMMO consumes less energy than DTO and OTSO for around 80% of the users. For the large delay tolerance profile, around 30% of the users consume more than 40% less energy, when CEMMO is used. Moreover, around 95% of the users observe improved battery lifetime compared to OTSO. We conclude that CEMMO not only offloads significantly more data than DTO and OTSO, but also improves energy consumption, by including the energy cost of each transfer policy in the offloading decision. CEMMO improves energy consumption for more than 90% of the users, independent of the delay tolerance profile.

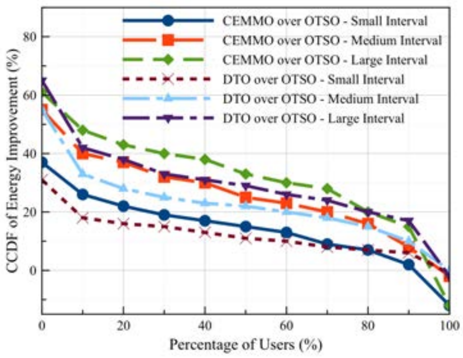


Figure 6.17 CCDF of energy improvement

6.4. Scenario 4: Performance evaluation of CARPOOL routing protocol

The goal of Scenario 4 is to evaluate the performance of CARPOOL routing protocol in a dense urban environment and study the impact of increased traffic load on its performance comparatively with four widely-used routing protocols, namely, Epidemic [160], PRoPHET [161], binary Spray-and-Wait with 10 bundle copies [165] and MaxProp [167].

Results in Figure 6.18 illustrate the delivery ratio (Equation 5.10) of the five routing schemes for increasing traffic load. We notice that in low traffic load conditions (less than 20000 bundles in 12h) only CARPOOL and MaxProp manage to deliver all bundles; the three other protocols fail to achieve maximum delivery ratio. CARPOOL achieves increased delivery ratio, since contacts between gateways and ferries are known *a priori* and in the event of unexpected delays, new paths to online gateways are being re-discovered. It should be noted that in contrast to other protocols, CARPOOL does not exploit short contacts between ferries. Through its scheduling tactics at the gateways and the ferries, MaxProp achieves high delivery ratio. Epidemic routing suffers from its excessive overhead and experiences worst performance. PRoPHET and Spray-and-Wait present decreasing offloading ratio with increasing traffic load. In high traffic load conditions (more than 25000 bundles in 12h) the delivery ratio of all protocols decreases when traffic load increases. Even in the worst scenario, CARPOOL performs significantly better than all other protocols, managing to successfully deliver 82% of the created bundles despite heavy congestion. Unlike other protocols, the delivery ratio of CARPOOL, even in worst-case scenarios, suffices in its own right to guarantee some level of service. A user may feel confident that even if one attempt fails, most likely this will not be repeated.

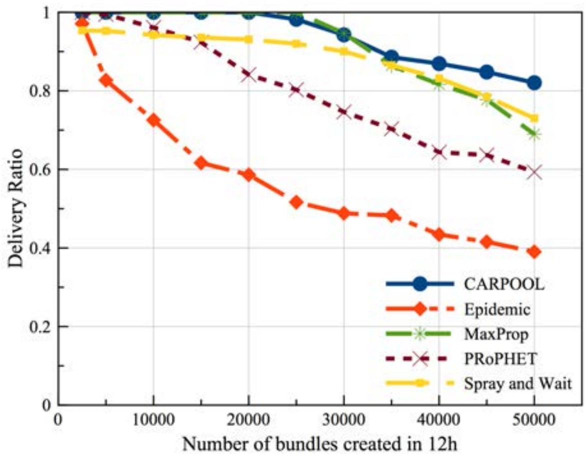


Figure 6.18 Delivery ratio for increasing number of bundles

In Figure 6.19, we show the overhead ratio (Equation 5.11) observed for each routing protocol for increasing traffic load. Given the density of the ferries and the gateways, along with the limited energy capacity of the ferries, overhead becomes important. As expected, CARPOOL presents minimum overhead, since there exists only one copy of each bundle in the network at any given time. Since CARPOOL keeps a single copy per bundle, it minimises energy consumption of battery-powered devices, but also allows for better bandwidth utilisation, which practically means our network can accommodate more users. Spray-and-Wait also keeps overhead low (as defined in its simulation settings), while the other three protocols suffer from increased overhead. As shown in Figure 6.19, overhead decreases for the rest of the protocols when traffic load increases. This is justified by the protocols' failure to operate in regular mode, since they cannot create their typical number of copies. The result of this overhead reduction is their functional blackout as it appears in the corresponding heavy-traffic delivery ratio results in Figure 6.18.

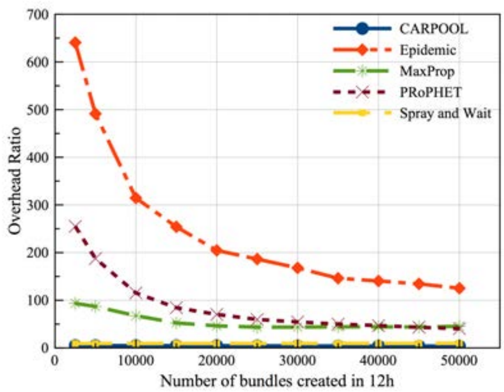


Figure 6.19 Overhead ratio for increasing number of bundles

In Figure 6.20, we show the median latency (Equation 5.12) of each protocol for increasing traffic load. The median latency of all protocols presents a steady increase for increasing traffic load with the exception of MaxProp, which presents a rapid increase in latency as traffic load increases. CARPOOL performs sufficiently well. Spray-and-wait outperforms other protocols at the expense of higher overhead and less delivery ratio. CARPOOL achieves exactly the same median latency with Spray-and-Wait in highly congested networks and manages to deliver 10% more packets with 1/10 of the overhead of Spray-and-Wait (i.e. CARPOOL keeps only one copy of each bundle in the network at any given time, while with Spray-and-Wait up to 10 copies of each bundle can co-exist in the network). When the network is not congested, Spray-and-Wait achieves lower median latency than CARPOOL (at the cost of delivering significantly less data) by exploiting opportunistic contacts between ferries as well.

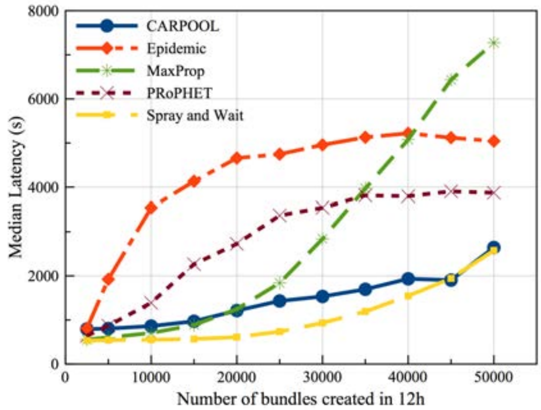


Figure 6.20 Median latency for increasing number of bundles

The motivation behind this work was to design a platform for delay-tolerant Internet access that extends the existing coverage of free WiFi access points in an urban environment. Indeed, we conclude that CARPOOL can offload significantly more data than other popular DTN routing protocols with minimum overhead. Being confident that CARPOOL can achieve even higher performance, when opportunistic contacts between ferries are also exploited, we proceed with the development of CARPOOL+ that extends CARPOOL towards this direction. The performance of CARPOOL+ is evaluated in the following subsection.

6.5. Scenario 5: Performance evaluation of CARPOOL+ routing protocol

The goal of Scenario 5 is to perform a performance assessment of CARPOOL+ routing protocol in comparison to the original CARPOOL protocol, Epidemic, PRoPHET (α=0.75, β=0.25 and γ=0.98), binary Spray-and-Wait with 10 bundle copies and MaxProp. We first focus on the gains of the two new mechanisms (i.e. exploitation of opportunistic contacts among ferries and en route path recalculation in case of delays) incorporated into

CARPOOL+, then evaluate the impact of online gateway availability and buffer size on the performance of CARPOOL+ and, finally, compare the performance of CARPOOL+ with the most popular DTN routing protocols, as described in Table 5.5.

6.5.1. Performance comparison between CARPOOL+ and CARPOOL routing protocols

First, we compare the performance of CARPOOL+ and CARPOOL routing protocols for increasing traffic load, when no schedule deviation occurs (1^{st} set of experiments of Table 5.5). Our evaluation results in Figure 6.21 show that CARPOOL+ achieves not only increased delivery ratio (Figure 6.21(a)), but also significantly reduced average latency (Figure 6.21(b)) compared to CARPOOL; even the small communication window between two ferries, leads to better performance. These gains are the outcome of the exploitation of opportunistic contacts among ferries, since routes that were not previously available are now exploited. When traffic load is high, we notice in Figure 6.21(a) and Figure 6.21(b) that delivery ratio and average latency are reduced. In high congestion conditions, bundles are lost because buffer size is not adequate to store all bundles, leading to reduced delivery ratio. At the same time, bundles whose destination is closer to the source gateway are more likely to be successfully delivered, thus leading to reduced average delay. Average hop count and overhead ratio are also almost halved, leading to significant resource saving (Figure 6.21(c) and Figure 6.21(d)). It should be noted that here we investigate routing overhead as defined in Section 4.2. As detailed in the description of our architecture, we assume that neighbour discovery is performed by lower layer communication protocols (i.e. link layer) and, therefore, we do not evaluate the overhead that corresponds to the exchange of periodic "hello" packets that need to be transmitted for ferries and gateways to identify if they are in range of each other. We note that CARPOOL+ incurs an additional overhead for the transfer of the bundle list from one node to another, when the second one needs to execute CARPOOL+ route selection algorithm to investigate whether a route to the destination can be found for these data. However, we expect the bundle list to consist of only a few Bytes, while the overall transferred data are in the order of MB, at least. Therefore, this additional overhead that CARPOOL+ incurs is insignificant.

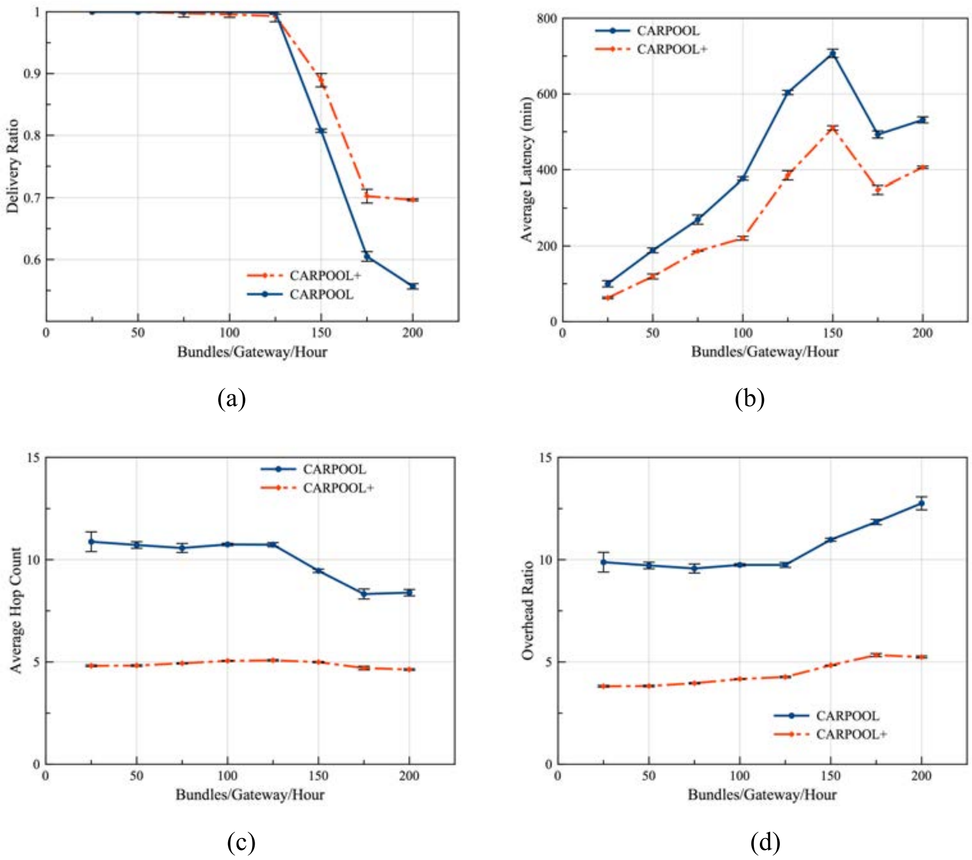


Figure 6.21 Delivery ratio (a), average latency (b), average hop count (c) and overhead ratio (d) when no schedule deviation occurs

Next, we extensively evaluate the performance of CARPOOL and CARPOOL+ for increasing traffic load, when schedule deviations occur due to traffic jams (2nd set of experiments of Table 5.5). As depicted in Figure 22(a) and Figure 22(b), CARPOOL+ achieves significantly higher delivery ratio with lower average latency than CARPOOL; thus more bundles are delivered to their recipients faster than before. In particular, for increased traffic load, CARPOOL+ manages to deliver 16% more bundles than CARPOOL with the same average delay. The overall bundle delivery latency strongly depends on the availability of online gateways. In these two sets of experiments, the availability of online gateways is limited (i.e. only 10% of the gateways are online), in order to highlight the benefits of the proposed architecture in terms of delivery ratio even when online gateways are scarce. This approach leads to high latency values for increased traffic load, as shown in Fig. 22(b). Moreover, in Figure 6.22(c) and Figure 6.22(d) we notice that both overhead ratio and average hop count have been significantly reduced. Bundles delivered through CARPOOL+ traverse two to three times less hops than those delivered through CARPOOL until they arrive to an online gateway. This can be translated into significant energy and computational gains, since increased hop count requires more recalculations and transmissions.

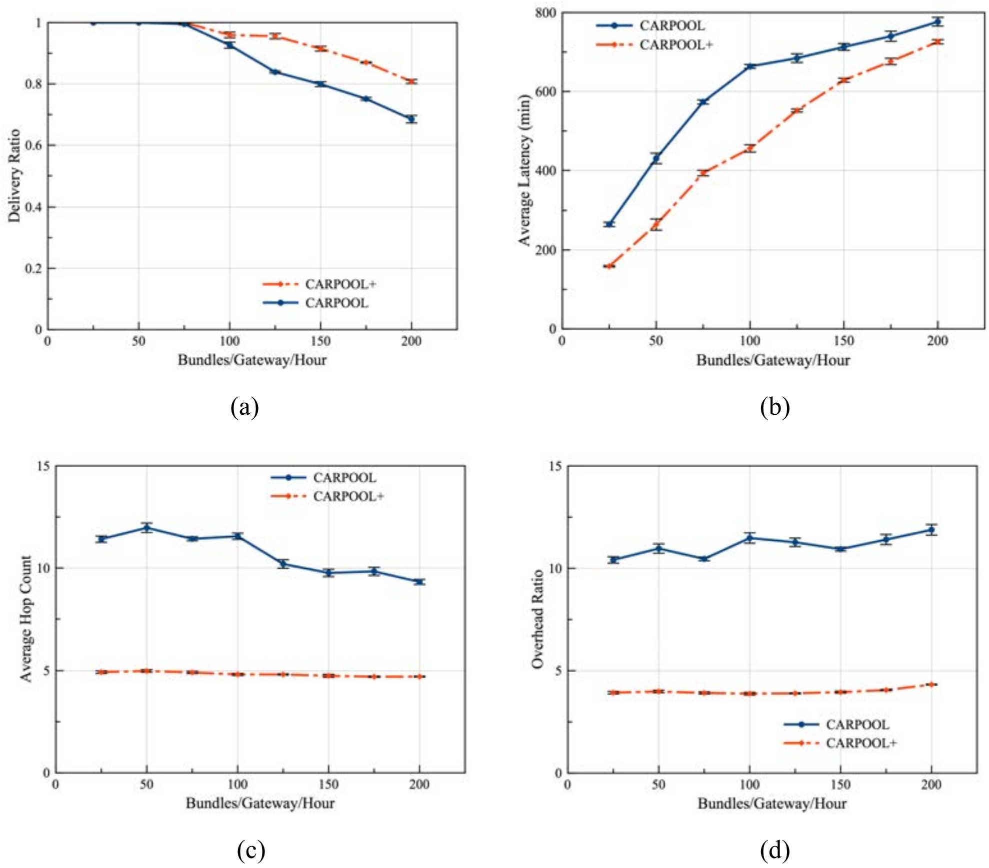
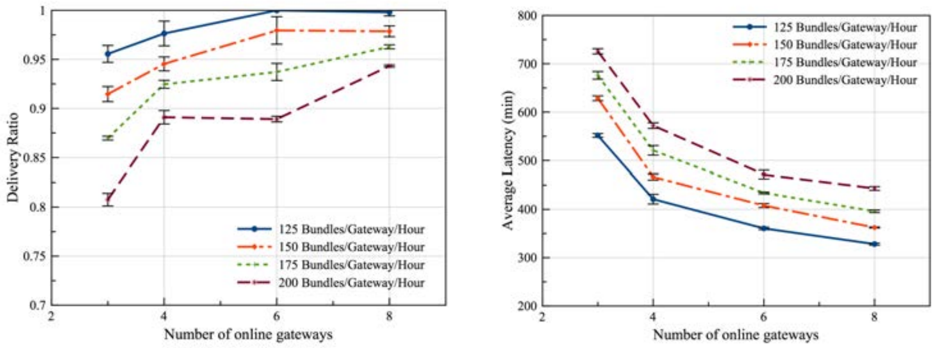


Figure 6.22 Delivery ratio (a), average latency (b), average hop count (c) and overhead ratio (d) in schedule deviation conditions

6.5.2. Impact of online gateway availability and buffer size on the performance of CARPOOL+

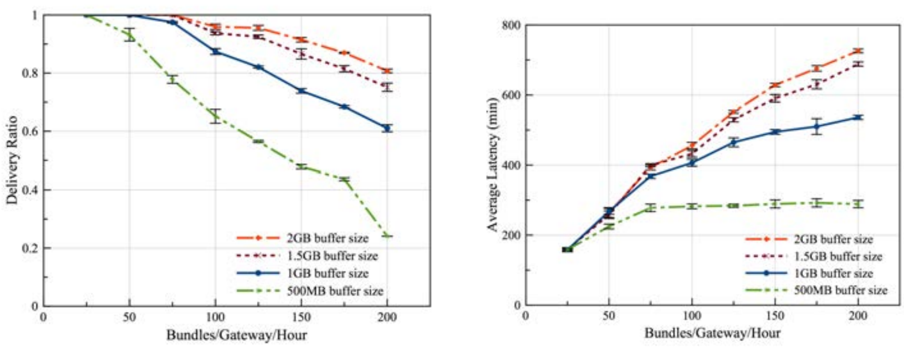
Following the experiments of the previous subsection, where the availability of online gateways is limited, in the next set of experiments we investigate how the number of online gateways affects the performance of CARPOOL+. In particular, we increase the number of online gateways from 3 to 8, keeping the total number of gateways unchanged, and we repeat the experiments for traffic load that varies from 125 to 200 bundles/gateway/hour (3rd set of experiments of Table 5.5). As depicted in Figure 23(a), CARPOOL+ achieves higher delivery ratio for increased online gateway availability, independent of the traffic load. At the same time, average latency is significantly reduced, as shown in Figure 23(b), since an online gateway is located closer to each sender and the same traffic load is split among more online gateways. We also note that we do not use an excessive number of online gateways; in the best case the online gateways comprise less than 30% of all gateways.



(a) (b)

Figure 6.23 Delivery ratio (a) and average latency (b) for increasing number of online gateways

We also investigate the impact of buffer size on the performance of CARPOOL+. In particular, we repeat the set of experiments when deviations from schedule occur for CARPOOL+ when buffer size equals to 1.5 GB, 1 GB and 500 MB (4th set of experiments of Table 5.5). As shown in our evaluation results in Figure 6.24, restricted buffer size leads to reduced delivery ratio and average latency. When traffic load is low, even small buffer size is adequate to handle all Internet access requests. As the network load increases, however, delivery ratio is reduced since the available storage capacity is not sufficient to handle the increased traffic load, leading to bundles being dropped. As shown in Figure 6.24(a), the smaller the buffer size, the stronger the reduction in delivery ratio. Average latency is also reduced when buffer size is small since the network is only able to successfully deliver bundles whose destination is close to the source node and, thus, require less overall buffering time. Given the amount of data that is being transferred, the frequency and the duration of the contacts, as well as the capacity of the links, buffer size in the order of 2 GB ensures that buffers hold enough bundles to achieve high contact utilisation, nonetheless without being overbooked with data that cannot be transmitted. We also note that given the store-and-forward nature of DTNs, buffering in DTNs is in the form of permanent storage.



(a) (b)

Figure 6.24 Delivery ratio (a) and average latency (b) for different buffer sizes

6.5.3. Performance comparison between CARPOOL+ and other DTN routing protocols

In this set of experiments, we extensively evaluate CARPOOL+ in comparison to the most prominent DTN routing protocols for increasing traffic load (5^{th} set of experiments of Table 5.5). We note that in all evaluation results we only consider the amount of data that has been transferred through the proposed access model, not the fallback mechanism, and, therefore, the comparison between CARPOOL+ and the other protocols is fair. As depicted in Figure 6.25(a), Spray-and-Wait is not suitable for a dense communication scenario with many opportunistic contacts among ferries. Given its approach to spray half of the bundles each node holds at every contact opportunity, in a city-wide scenario Spray-and-Wait distributes bundles only to a few nodes that are close to the source node; if these nodes are offline gateways or ferries that do not cross an online gateway, bundles are not delivered to their recipients. This is also obvious from the low average delay for Spray-and-Wait in Figure 6.25(a); Spray-and-Wait only manages to deliver bundles to online gateways that are located close to the original offline gateway. Epidemic manages to deliver only a small fraction of the overall data, given its greedy replication at all contact opportunities. Intuitively, we would expect Epidemic to achieve low average delay. However, the limited contact duration among ferries or gateways and ferries, along with the restricted storage size, poses barriers to the greedy replication approach of Epidemic. PRoPHET cannot accurately estimate probabilities for highly loaded dynamic networks and, therefore, presents decreasing delivery probability for increasing traffic load. As shown in Figure 6.25(a), CARPOOL+ and MaxProp achieve significantly higher delivery ratio than all other protocols. However, CARPOOL+ outperforms MaxProp in terms of average delay and routing overhead, since it manages to deliver bundles faster (Figure 6.25(b)) and with significantly lower overhead ratio (Figure 6.25(c)), especially when traffic load is high. We also note that the overhead that relates to the discovery of other nodes in range and the initiation of a connection between any two nodes, which we assume is provided by lower communication layers, applies to all routing protocols and, therefore, does not constitute bias for the evaluation of CARPOOL+.

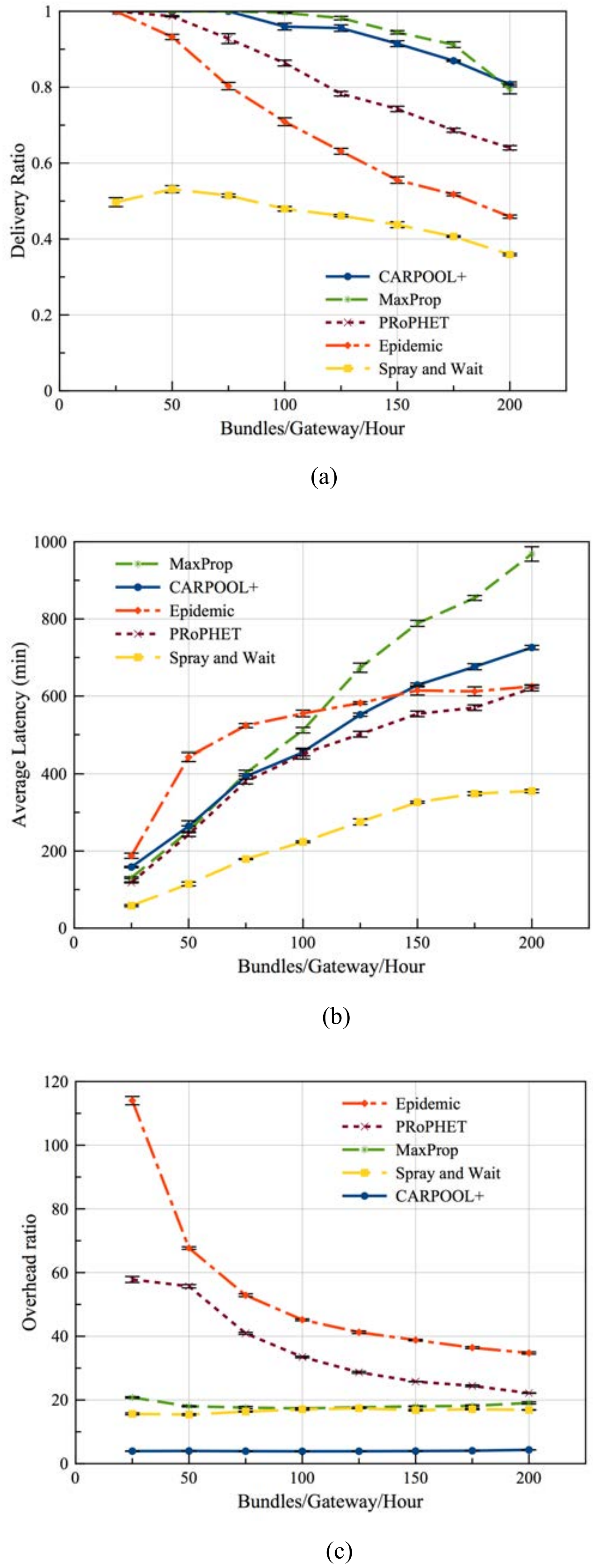
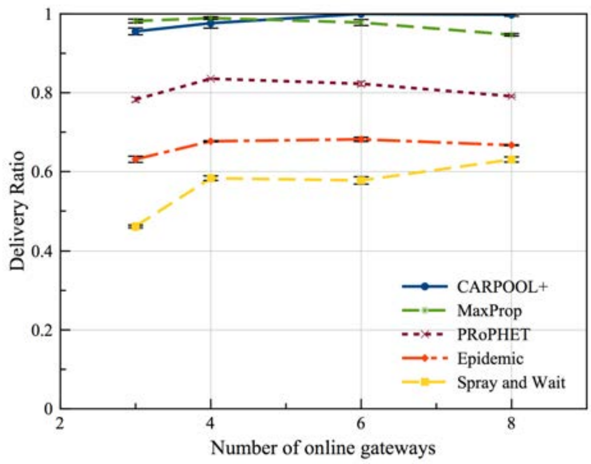
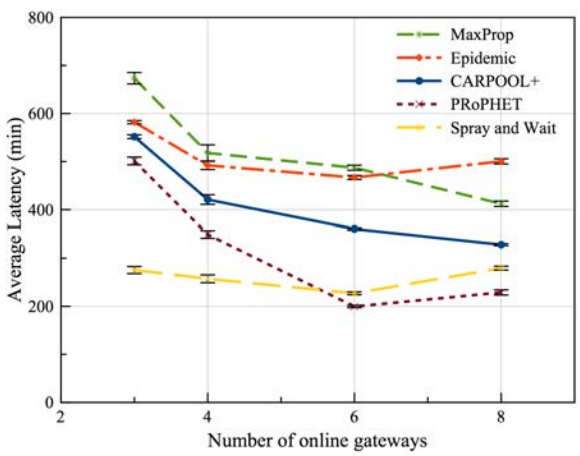


Figure 6.25 Delivery ratio (a), average latency (b) and overhead ratio (c) for increasing traffic load

Finally, we compare the performance of CARPOOL+, Epidemic, PRoPHET, Spray-and-Wait and MaxProp for an increasing amount of available online gateways (6th set of experiments of Table 5.5). Our evaluation results in Figure 6.26(a) show that CARPOOL+ is the only protocol that delivers all bundles independent of the amount of available gateways, followed closely by MaxProp. We also notice that increasing the number of online gateways leads to better performance for Spray-and-Wait, while the performance of Epidemic and PRoPHET is left almost unaffected. The increase in the number of online gateways also has a significant impact on average latency, as depicted in Figure 6.26(b); all protocols present decreasing average latency for increasing number of online gateways. CARPOOL+ is the only protocol that delivers all bundles, while keeping latency moderate. MaxProp presents significantly higher average latency than CARPOOL+, while PRoPHET and Spray-and-Wait achieve lower average latency than CARPOOL+ at the expense of transmitting only a portion of the traffic load.



(a)



(b)

Figure 6.26 Delivery ratio (a) and average latency (b) for increasing number of online gateways

Summarising, we observe that CARPOOL+ improves delivery ratio and reduces average delay, even when all ferries follow the predefined schedule. When deviations from schedule occur, CARPOOL+ achieves higher delivery ratio with lower latency and minimum overhead compared to the most prominent DTN routing protocols. Therefore, we conclude that it is feasible to build the proposed DTN-based architecture for public transport networks using low-cost components to extend existing Internet coverage and provide free delay-tolerant Internet access in urban environments. The core of the proposed architecture lies in exploiting not only scheduled contacts, but also opportunistic contacts among ferries, and withstanding typical deviations from schedule, e.g. due to road traffic.

7. Enablers for universal coverage

In this chapter, we provide an overview of our joint work on the design of an Internet architecture that ensures universal coverage, by traversing the entire range of connectivity options, and enables tighter integration of satellite and terrestrial communications. In Section 7.1, we summarise the needs that the proposed Information Centric Delay Tolerant Networking (I-DTN) architecture aims to address and describe its framework. The core of the architecture lies in the unification of the two key networking technologies that were described in Chapter 2; ICN and DTN. As a first step towards the implementation of I-DTN architecture, we have developed SPICE DTN testbed, a state-of-the-art DTN testbed for terrestrial, satellite and space communications deployed at the Space Internetworking Center in Xanthi, Greece. SPICE DTN testbed is described in Section 7.2.

7.1. I-DTN architectural framework

All solutions to the problem of achieving digital inclusion so far focus on single technologies that fail to apply universally. It has, therefore, become clear that in order to achieve true digital inclusion, all available resources across the spectrum of connectivity options need to be exploited. This can only be achieved by an Internet architecture that seamlessly integrates multiple transmission technologies, provides flexible QoS and supports delay tolerance (for disruptive communication and time-shifted content delivery) and information centricity (for increased content delivery efficiency). Such an architecture also needs to support heterogeneous networks, ranging from terrestrial to satellite networks, in order to provide extended Internet coverage and a variety of new services.

In [200][201], we propose an Internet architecture that can ensure universal coverage by traversing the entire range of connectivity options through a single unifying communication architecture with a single set of abstractions. Such an architecture not only spurs innovation for a wide range of new services and applications, but also encompasses existing successful Internet services. To achieve that, we make use of advances in the area of ICN and its inherent ability to push content to the edges, providing more localised access to important content and reducing access cost per bit through the enablement of a transmit-when-needed policy, as well as DTN as a complementary connectivity option.

The main goal of the proposed unified I-DTN architectural framework is to efficiently exploit all possible communication opportunities, from fixed or mobile broadband networks to disruptive networks and satellite links, while providing a unified abstraction to application developers for supporting current Internet-based services and enabling innovative future solutions. The framework of I-DTN architectural framework combines IP, ICN and DTN solutions into a novel system architecture, exposing a common information-centric abstraction to applications, while supporting a range of networking protocols over different transport networks.

Figure 7.1 presents the I-DTN architectural framework. As shown in Figure 7.1, the proposed architecture supports both native applications, such as App3, or backwards-compatible applications through socket abstraction, such as App1 and App2. The PUB/SUB information-centric layer below provides the required service abstraction and supports interfaces for various dissemination strategies, including native ICN, IP, as well as various DTN approaches, such as BP, SCAMPI and Haggle. Different types of networking technologies, such as satellite and WiFi networks, are supported at the lower layers.

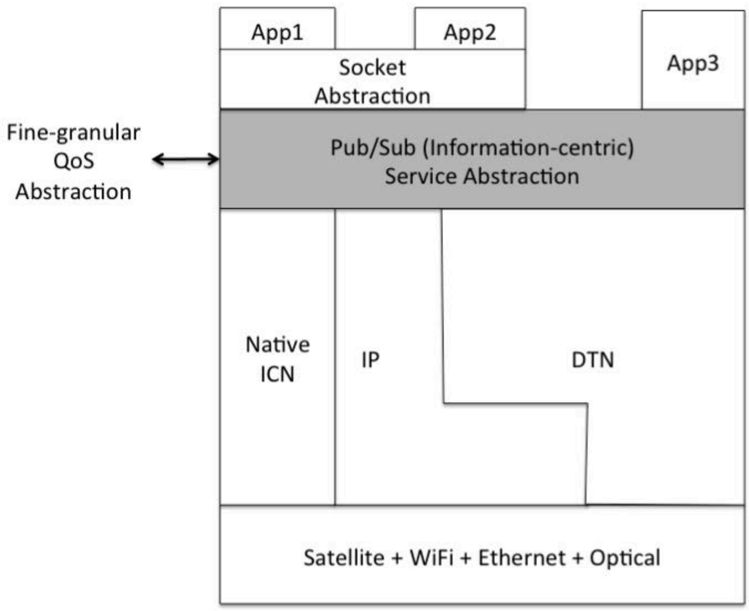


Figure 7.1 I-DTN architecture

This proposal is an integrative architectural framework that brings IP, ICN and DTN together into a single framework, in which DTN complements current IP and ICN solutions as an ideal candidate for communication in disconnected network environments that face increased delays and/or disruption. I-DTN framework also provides interfaces that allow for defining and manipulating the QoS parameters in alignment with the service model that is exposed to the applications.

7.2. SPICE DTN testbed

As a first step towards the implementation of the proposed I-DTN architecture, we have developed SPICE DTN testbed [26], a state-of-the-art DTN testbed for terrestrial, satellite and space communications deployed at the Space Internetworking Center in Xanthi, Greece.

SPICE testbed is an experimental research environment for developing and evaluating a variety of new architectures and protocols for space communications. In particular, SPICE testbed presents the following key features:

i) *Realistic emulation of space communications.* SPICE testbed provides a realistic experimental environment for terrestrial, satellite and space communications, including real and flight-ready components. Indeed, specialised hardware and software components have been incorporated into the testbed, enabling the testing, evaluation and validation of implemented mechanisms and protocols. Furthermore, a link with a geostationary satellite, namely HellasSat 2, is utilised on demand to provide real satellite link characteristics for experimental purposes.

ii) *Compliance with typical equipment of major space agencies.* SPICE testbed incorporates typical components used by space agencies for the evaluation of protocols prior to mission launch. In particular, the Portable Satellite Simulator (PSS) [202] is built in compliance with the requirements of the European Space Agency (ESA), while CORTEX Command Ranging and Telemetry (CRT) [203] is used by all major space agencies in their ground station facilities to support their missions. Finally, Satellite Tool Kit (STK) [204] is employed by mission designers as a tool to calculate not only exact satellite trajectories and contact durations, but also detailed communication characteristics, and perform link-budget analysis.

iii) *Interface provision for multiple underlying protocols.* SPICE testbed not only supports a variety of convergence layers for underlying protocols that comply with CCSDS standards and major space agencies, but also facilitates the development of new routing, transport, and management schemes. Taking advantage of this functionality, SPICE researchers are able to validate such schemes against standardised protocols and perform interoperability testing.

iv) *Scalability*. SPICE testbed includes numerous nodes for the evaluation of complex communication scenarios and can be further enhanced with virtual nodes installed on a high-performance server. Therefore, complex scenarios involving constellations of satellites (e.g., cubesats) and several end-users can be realistically modelled. It is also noted that this scalability comes without adding any complexity, since the testbed is easily configured and controlled through dedicated workstations.

SPICE testbed is an ideal platform to evaluate different DTN implementations, protocols, applications and services. Its architecture and components have already contributed to the design, implementation, and optimisation of new algorithms and protocols, with respect to the challenging conditions of satellite and space communications [15][16][17][18][23][25] [205][206][207]. Moreover, SPICE testbed has been used as the key testing platform in several European and ESA funded projects including the European Union's 7th Framework Programme for Research and Technological Development (FP7) SPICE [208], FP7 Space

Data Routers (SDR) [209], ESA's Extending Internet Into Space, ESA's BitTorrent study and more.

Notionally, the testbed comprises two distinct parts, namely the data plane and the control plane, and its architecture is depicted in Figure 7.2. Data are transferred between nodes to emulate communication among space and ground assets through the data plane, while configuration scripts, control messages, and reports related to the emulation are managed through the control plane.

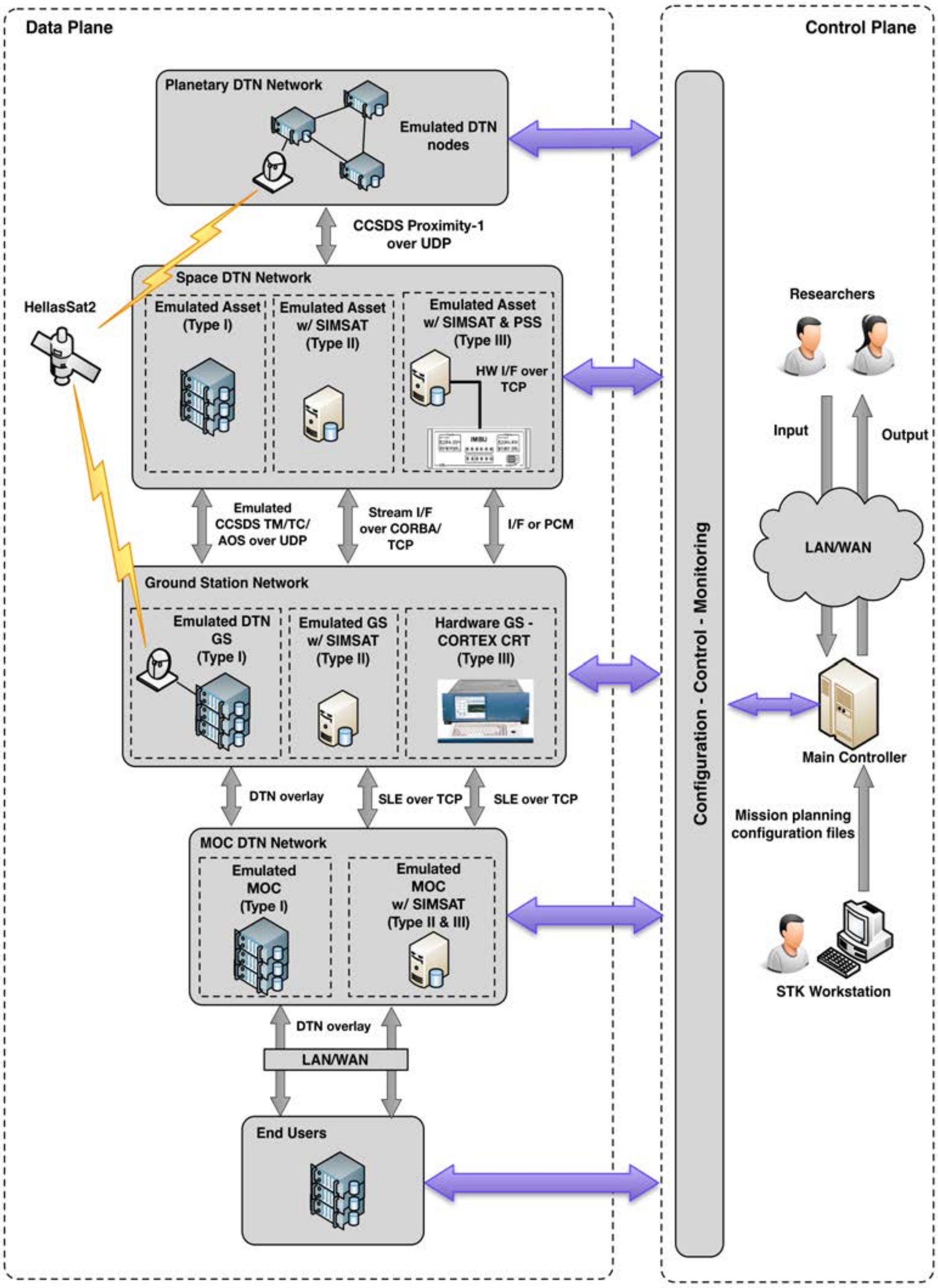


Figure 7.2 SPICE testbed architecture

In particular, the control plane is responsible for:

- Configuring and controlling the testbed nodes in real time based on user input;
- Monitoring the correct node operation;
- Collecting any associated performance statistics, and
- Delivering the experimental results to the researchers.

These operations are coordinated by a main controller accessible via the internal network or the Internet. A hardware firewall restricts remote access, allowing only encrypted Virtual Private Network (VPN) connections. Researchers configure the experiments to be conducted through a UI, available at the main controller. Link characteristics and emulation parameters are either imported directly by the users or provided by the STK workstation after conducting the relevant simulations. Upon the completion of an experiment, results are collected and stored in the main controller.

The data plane of SPICE testbed supports the emulation of a wide variety of space and satellite communication scenarios, including present and future missions. These scenarios may involve:

(a) A number of landed assets, such as landers and rovers, that generate scientific data and can possibly form a planetary network;

(b) A set of space assets near Earth or in deep Space (e.g. Low Earth Orbit (LEO) satellites, Medium Earth Orbit (MEO) satellites, Geosynchronous (GEO) satellites, spacecraft, planetary relay satellites etc.) that can produce and/or relay data, and

(c) Terrestrial facilities such as typical ground stations (GS), mission operation centers (MOC) and end-users.

Researchers are able to emulate all these types of space communications taking advantage of the diverse protocol stack configurations supported by SPICE testbed. As shown in Figure 7.3, SPICE DTN testbed supports a variety of user applications, such as the Bundle Delivery Time Estimation (BDTE) tool [23], application-independent protocols, such as the Delay Tolerant Payload Conditioning (DTPC) protocol [18] and several convergence layers for the underlying protocols including:

- The UDP convergence layer protocol (UDPCL);
- The TCP convergence layer protocol (TCPCL);
- The LTP convergence layer protocol (LTPCL), and
- The Bundle Streaming Service (BSS) [25].

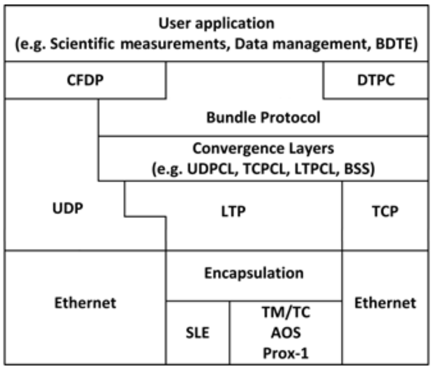


Figure 7.3 SPICE testbed protocol stack

At the lower layers, an implementation of Proximity-1 [210] is employed as a CCSDS data link protocol to interconnect planetary nodes with relay satellites. Depending on the objective of the emulation, researchers may use one of the three available CCSDS data link protocol implementations to interconnect space and GS DTN networks, as shown in Figure 7.2:

I. *Type I:* Software-based emulation of the basic functionality of Telemetry (TM) [211], Telecommand (TC) [212] and Advanced Orbiting Systems (AOS) [213] protocols. Space assets and GSs are emulated using only ION-DTN implementation.

II. *Type II:* Software-based emulation of the full functionality of TM/TC/AOS protocols including Space Link Extension (SLE). Space assets are emulated using ION-DTN, as well as SIMSAT Software. GSs do not support DTN and receive TM/TC/AOS packets using SIMSAT.

III. *Type III:* Hardware-based emulation of the full functionality of TM/TC protocols. Space assets are emulated utilising ION-DTN and the combination of SIMSAT and PSS. In this case, ground stations do not support DTN and only receive TM/TC frames using CORTEX CRT system.

Communication between a GS and a MOC can be either DTN-based (Type I) or SLE-based (Type II and Type III). Space data are then transferred to the end-users using ION-DTN. HellasSat 2 satellite may also be employed to provide real satellite link characteristics between DTN nodes.

The most important hardware components of SPICE DTN testbed are described below:

- *DTN Nodes:* The DTN nodes are fifteen rack-mounted servers used as distinct emulation nodes in experiments with network protocols of the DTN architecture. Each node is equipped with a quad-core Intel Xeon CPU operating at 2.4 GHz with 4 GB of Random Access Memory (RAM) and 1 TB of storage, running a Linux distribution. Private IP addresses are assigned to these servers so that they can communicate directly with each other locally. Additionally, they are divided into three groups of five, with a public IP address used by each group for inbound/outbound Internet traffic. Inbound traffic is strictly limited to a few ports needed for remote access and the DTN frameworks. Users have the ability to gain access by means of an IPSec VPN, which is configured on a hardware firewall. Each server constitutes a standalone DTN node implementing the full DTN stack. In certain scenarios, where more DTN nodes are needed, the testbed core can be extended by employing a number of virtual machines. For this purpose a high-performance computer is used, featuring two hexa-core Intel Xeon CPUs, 24 GB of RAM and 12 TB of redundant storage. The high-performance computer runs a bare-metal hypervisor, VMware vSphere, which sets up the virtualisation layer. This makes for a scalable testbed core capable of accommodating more than 35 nodes, enough to emulate most space missions.

- *PSS:* PSS is a generic system capable of injecting TM into the downlink chain of a GS and receiving TC from the uplink chain, complying with CCSDS recommendations and European Cooperation for Space Standardisation (ECSS) and ESA standards. PSS offers several monitoring and control interfaces and a maintenance interface, which allows controlling and monitoring the PSS locally at the GS or remotely from the control center. PSS is deployed in the testbed as a hardware satellite model, incorporating the link layer protocol stack of a real satellite.
- *CORTEX CRT*: CORTEX CRT is a TM and TC base-band commercial off-the-shelf (COTS) solution. CORTEX CRT system allows a continuous improvement of the signal processing and supports future standards through telemetry processing, CCSDS TC processing, ranging measurements etc. In essence, CORTEX CRT is able to decode and process TM data received from a satellite through an antenna and encode TC data transmitted to a satellite. CORTEX CRT has field-proven compatibility with most of satellites, high level of reliability with no preventive maintenance, and has been extensively used by many space agencies, including the National Aeronautics and Space Administration (NASA), ESA and Japan Aerospace Exploration Agency (JAXA). Within SPICE testbed, CORTEX CRT emulates the functionality of a real GS collecting and transmitting data from/to satellites.
- *HellasSat 2:* A satellite link over HellasSat 2 has been set up at the premises of SPICE for the evaluation of BP over a real satellite link, subject to errors and disruptions due to weather conditions.

The key software components of SPICE DTN testbed include:

- *DTN Implementations:* As described in Chapter 2, ION-DTN is an implementation of the DTN architecture developed by JPL and released as open source software. It includes implementations of the BP, the Licklider Transmission Protocol (LTP)

[214], Bundle Security Protocol (BSP) [187], and two CCSDS application protocols that have been adapted to run over the BP/LTP stack: class-1 (unacknowledged) CCSDS File Delivery Protocol (CFDP) [215] and Asynchronous Message Service (AMS) [216]. Several protocols that have been developed by researchers of the Space Internetworking Center have been already incorporated in the latest ION-DTN release, and other are planned to be released in the following versions. DTN2**,** which is the reference implementation of the DTN architecture, and IBR-DTN, an implementation of the BP designed for embedded systems and smartphones are also included in the SPICE Testbed, mainly for interoperability testing purposes.

- *CFDP:* The ESA CFDP ground segment implementation provides a full implementation of the CFDP. ESA's CFDP provides a Java library and a daemon implementation for reliable and unreliable file transfer in space and on the ground.
- *SIMSAT:* SIMSAT [217] is a general-purpose real-time simulation infrastructure developed for ESA. SIMSAT supports standard simulation services such as cyclic and event-based real-time scheduling of models, logging of simulation events etc. The SIMSAT UI is used to coordinate experiments that utilise the PSS.
- *STK:* STK is a COTS mission modelling and analysis software for space, defence and intelligence systems and is used as an external component to the DTN testbed. STK Professional Edition is used to create and manage high-level objects (satellites, aircraft, facilities, etc.), propagate and orient vehicles, analyse relationships between objects, visualise objects in two dimensions (2D) and three dimensions (3D) and animate in real or simulated time. With STK Communications detailed transmitter and receiver elements with antenna pointing are defined, direct or bent pipe communication links over time are analysed and link budget analysis of each communication link is performed, contact periods among communicating elements are calculated, accidental/intentional jamming effects are analysed etc. Finally, the STK Integration Module integrates with other applications in order to develop custom applications to automate repetitive tasks from outside of the application. Information like bandwidth, error rates, propagation delay, disruption periods and connectivity schedule constitute network parameters that are imported to SPICE testbed prior to each experiment.
- *Netem:* The Network Emulator (NETEM) tool [218], which is included in recent Linux kernel versions (2.6+), is used to alter networking properties and emulate variable delay, loss, duplication and re-ordering.

8. Conclusions

In the present thesis, we highlighted the challenges of the continuously growing global mobile data traffic and the growing digital divide, and proposed mechanisms, protocols and architectures as solutions to these problems. In particular, our work consisted of:

- The evaluation of a less-than-best-effort access method, compared to a typical best-effort protocol, in the sub-packet regime of shared backhaul links in developing regions.
- The design, implementation and evaluation of a peer-assisted offloading method that offloads data from the uplink independent of content and popularity, along with the design, implementation and evaluation of a cost-effective multi-mode offloading mechanism that selects the most effective transfer method based on the cost defined by each operator.
- The design and implementation of a DTN architecture for urban environments that exploits means of public transport to enhance existing Internet connectivity through delay-tolerant applications, as well as the design, implementation and evaluation of a new DTN routing protocol that achieves high delivery ratio with low latency and minimum overhead by utilising the existing connectivity plan of means of public transport and exploiting opportunistic contacts.
- The design of a Future Internet architecture that ensures universal coverage, by seamlessly integrating multiple transmission technologies, and enables tighter integration of terrestrial and satellite networks.

In the following subsections, we describe the main contributions of the thesis, we summarise our conclusions and provide guidelines for future research.

8.1. Access methods in the sub-packet regime

Previous work has shown that the implementation of a typical best-effort (BE) protocol, such as TCP, in the sub-packet regime of shared backhaul links in developing regions leads to severe unfairness, high packet loss rates and repetitive timeouts. Within this framework, we claimed that a less-than-best-effort (LBE) protocol, that was initially introduced to exploit the unused capacity, could perform significantly better than TCP, given its conservative approach to resource consumption.

Based on our claims, we investigated the performance of LEDBAT access method and its fair modification fLEBDAT in the sub-packet regime of shared backhaul links of wireless community networks. In particular, we evaluated the performance of TCP NewReno, LEDBAT and fLEDBAT access methods through simulations using realistic backhaul links with various characteristics in terms of capacity, RTT and buffer size. Evaluation scenarios were split into two sets of experiments.

In the first set of experiments, all flows used a single type access method, i.e. either TCP NewReno, LEDBAT or fLEDBAT. The goal of these experiments was to compare the performance of a BE and a LBE protocol in the sub-packet regime. Based on the simulation results, we showed that, when we enter the sub-packet regime, fLEDBAT:

- Presents less retransmissions achieving higher link efficiency;
- Performs better resource distribution among its flows compared to TCP NewReno, achieving increased fairness, and
- Achieves lower packet loss probability, if the link buffers are large enough to accommodate packets from all flows.

In the second set of experiments, equal number of TCP NewReno and fLEDBAT flows co-exist in the backhaul link. The goal of these experiments was to investigate the operation of fLEBDAT in the presence of TCP flows in the sub-packet regime. Our results showed that, when TCP NewReno and fLEDBAT flows share the same link in the sub-packet regime, fLEDBAT flows fail to measure the actual base delay due to the standing queue and become aggressive, consuming more and more resources.

We concluded that, in order for LEDBAT flows to function properly in the sub-packet regime when competing with TCP flows, new ways to estimate base delay need to be developed and a conservative reaction to consecutive timeouts needs to be adopted.

8.2. Cellular data offloading

In the next part of our research, we focused on mechanisms to decongest cellular networks in developed regions; data offloading through WiFi APs is so far the most popular approach. Studying the existing cellular data offloading transfer methods, we identified a major gap; all peer-assisted offloading solutions so far are based on data interests and subscribers.

Within this framework, we developed the first peer-assisted offloading (PAO) transfer method that offloads cellular data from the uplink, independent of its content and popularity. To achieve that, the mobile devices of nearby nodes are utilised as intermediate data carriers to offload data through a WiFi AP. The main concept of our approach lies in storing data locally within a specific region for a short time interval, in a way that any mobile user who enters this region during this interval receives a replica of the data. Data storage enables the data exchange between the source node and intermediate users, who actually perform the transfer through a WiFi AP, using ad hoc technologies such as WiFi Direct. The cellular operator manages the overall procedure.

In order to enable PAO, we proposed a mobility and connectivity prediction model based on a Markov process and developed a forwarding scheme with low storage and energy overhead. PAO was implemented along with the other two most popular offloading methods: pure on-the-spot offloading (OTSO) and pure delay-tolerant offloading (DTO).

All three offloading methods were implemented and evaluated as distinct transfer policies of CEMMO, a new cost-effective multi-mode offloading mechanism that we proposed. CEMMO helps cellular operators increase the amount of traffic offloaded by their customers by discovering WiFi APs in their vicinity and automatically offloading data either directly or through other peers, depending on the estimated transfer cost of each method. The overall cost, as defined by each operator, includes transfer costs, energy costs, as well as incentives to motivate user participation. CEMMO helps balance the load and relieve 3G access networks from increased usage, thus increasing their total network capacity and meeting the increasing traffic demands. In particular, CEMMO incorporates three modes of operation:

- *Delay-tolerant offloading*;
- *Peer-assisted offloading,* and
- *Transfer over the available 3G network*, when other transfer methods fail or their estimated cost is higher than the transfer cost over the 3G network.

OTSO is always exploited when available. We also note that CEMMO provides cellular operators with knowledge on the amount of data offloaded by each user. Such knowledge is currently unavailable and should help cellular operators design the expansion of their network accordingly.

CEMMO was evaluated in a variety of scenarios; even in scenarios with limited available WiFi APs CEMMO not only offloaded significantly more data than pure DTO and pure OTSO, but also outperformed the other mechanisms under investigation in terms of normalised overall data transfer cost per MB. Our evaluation results showed that a significant portion of data transmitted over CEMMO were delivered over the newly proposed PAO transfer method. We also observed that PAO has low storage requirements due to the spatio-temporal restrictions of the flooding process.

An energy optimisation of CEMMO was also designed, implemented and evaluated. If an offloading scheme only aims to increase the offloading traffic ratio without considering energy consumption on mobile devices, it can quickly drain the battery of the devices. We concluded that CEMMO improves energy consumption for more than 90% of the users (independent of the delay tolerance profile) by including the energy cost of each transfer policy in the offloading decision.

8.3. DTN routing in urban environments

As far as the growing digital divide is concerned, we observed that even in well-connected environments several members of the society remain disconnected due to socio-economic reasons. Based on this observation, we focused on metropolitan environments with an ultimate goal to extend existing free Internet access by providing delay-tolerant Internet access to the under-privileged society that is currently excluded from today's digital world.

In particular, we proposed a solution that extends the existing free Internet access provided by public hotspots that are usually scattered around a city. Actually, we broadened connectivity options by exploiting typical means of public transport as message ferries. Offline DTN gateways located near ferry stops collect Internet access requests from end-users in that area and DTN ferries act as relays between offline gateways or designated gateways that have access to the Internet and are capable of handling such requests. To achieve that, we designed and implemented an easy-to-deploy architecture that exploits both prescheduled contacts in public transport networks and opportunistic contacts among ferries to provide delay-tolerant Internet access to users. The proposed architecture can be built using low-cost components, such as Raspberry Pi, and is mainly expected to serve users that access the Internet through delay-tolerant applications at no cost.

In order to select a route that is expected to achieve high delivery ratio with low latency, we also designed and implemented:

- Connectivity Plan Routing Protocol (CARPOOL), a DTN routing protocol that routes data based on the knowledge of the public transport vehicles schedule and
- Its enhanced version, CARPOOL+, a dynamic routing protocol for public transport networks that also exploits opportunistic contacts among ferries and incorporates mechanisms to select a route that is expected to achieve earliest data delivery with minimum overhead, even when significant deviations from schedule occur.

We extensively evaluated the proposed architecture in several dense urban scenarios and compared both CARPOOL and CARPOOL+ with the most prominent DTN routing solutions to highlight their efficiency. Our results showed that CARPOOL+ improves delivery ratio and average latency compared to CARPOOL not only when ferries follow the predefined schedule, but also when delays occur. The comparison of CARPOOL+ with the most well-known DTN routing protocols also showed that CARPOOL+ achieves the highest delivery ratio with low latency and minimum overhead.

8.4. Future Internet

In the last part of this thesis, we provided an overview of our joint work on the design of a unified Information Centric Delay Tolerant Networking (I-DTN) architectural framework to efficiently exploit all possible communication opportunities, from fixed or mobile broadband networks to disruptive networks and satellite links, while providing a unified abstraction to application developers for supporting current Internet-based services and enabling innovative future solutions. The framework of the I-DTN architectural framework combines IP, ICN and DTN solutions into a novel system architecture that exposes a common information-centric abstraction to applications and supports a range of networking protocols over different transport networks.

As a first step towards the implementation of I-DTN architecture, we have developed SPICE DTN testbed, a state-of-the-art DTN testbed for terrestrial, satellite and space communications; our next step is to integrate the ICN characteristics of the I-DTN architecture into SPICE DTN testbed. Towards this direction, we have received funding from the European Commission (EC) for a new H2020 research project (UMOBILE, Grant Agreement No 646124) that aims to develop an integrated ICN/DTN architecture [219].

Credits

I would like to acknowledge the contributions of all colleagues who co-authored publications whose research outcomes have been included in the present thesis and deeply thank every single one of them for our fruitful cooperation.

1. *I. Komnios, A. Sathiaseelan, and J. Crowcroft, "LEDBAT Performance in Sub-packet Regimes", 11th IEEE/IFIP Annual Conference on Wireless On-Demand Network Systems and Services (WONS 2014), Obergurgl, Austria, April 2014.*

   Arjuna Sathiaseelan proposed the idea to investigate the performance of a LBE access method in sub-packet regime and contributed to the discussion on the research results. Jon Crowcroft contributed to the discussion on how to approach the problem under investigation.

2. *I. Komnios, F. Tsapeli, and S. Gorinsky, "Cost-Effective Multi-Mode Offloading with peer-assisted communications", Ad Hoc Networks, Elsevier, Volume 25, Part B, pp.370-382, ISSN 1570-8705, February 2015.*

   Fani Tsapeli and Sergey Gorinsky contributed to the development of the concept of CEMMO. Fani Tsapeli also contributed to the implementation of CEMMO in ONE simulator, while Sergey Gorinsky provided input regarding the notion of transfer cost from the prespective of a cellular operator, as well as the required incentives to motivate user participation.

3. *I. Komnios, and V. Tsaoussidis, "CARPOOL: Extending Free Internet Access over DTN in Urban Environments", in ACM MobiCom Workshop on Lowest Cost Denominator Networking for Universal Access (LCDNet 2013), Miami, Florida, USA, September 2013.*

   Vassilis Tsaoussidis contributed to the definition of the networking scenario that CARPOOL routing protocol aims to address, as well as the description of the evaluation results.

4. *I. Komnios, and V. Tsaoussidis, "CARPOOL: Connectivity Plan Routing Protocol", 12th International Conference on Wired & Wireless Internet Communications (WWIC 2014), Paris, France, May 2014.*

   Vassilis Tsaoussidis contributed to the description of the evaluation results.

5. *I. Komnios and E. Kalogeiton, "A DTN-based architecture for public transport networks", Annals of Telecommunications, Springer, ISSN 0003-4347, August 2015.*

   Eirini Kalogeiton contributed to the implementation of CARPOOL+ routing protocol, based on original CARPOOL implementation, in ONE simulator, as well as the description of the two new mechanisms CARPOOL+ incorporates.

6. *A. Sathiaseelan, D. Trossen, I. Komnios, J. Ott, and J. Crowcroft, "An Internet Architecture for the Challenged", IAB Workshop on Internet Technology Adoption and Transition (ITAT), Cambridge, UK, December 2013.*

   Arjuna Sathiaseelan contributed to development of the concept of a Future Internet architecture that provides extended Internet coverage. Dirk Trossen contributed to the ICN aspects of I-DTN architecture by providing input on PURSUIT ICN. Jörg Ott contributed to the DTN aspects of I-DTN architecture by providing input on SCAMPI DTN. Jon Crowcroft contributed to the overall discussion on I-DTN architecture.

7. *A. Sathiaseelan, D. Trossen, I. Komnios, J. Ott, and J. Crowcroft, "Information centric delay tolerant networking: An Internet Architecture for the challenged", Technical Report: UCAM-CL-TR-841, ISSN 1476-2986.*

   Arjuna Sathiaseelan contributed to development of the concept of a Future Internet architecture that provides extended Internet coverage. Dirk Trossen contributed to the ICN aspects of I-DTN architecture by providing input on PURSUIT ICN. Jörg Ott contributed to the DTN aspects of I-DTN architecture by providing input on SCAMPI DTN. Jon Crowcroft contributed to the overall discussion on I-DTN architecture.

8. *I. Komnios, I. Alexiadis, N. Bezirgiannidis, S. Diamantopoulos, S.-A. Lenas, G. Papastergiou, and V. Tsaoussidis, "SPICE Testbed: A DTN Testbed for Satellite and Space Communications", 9th International Conference on Testbeds and Research*

Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2014), Guangzhou, China, May 2014.

All co-authors contributed to the development of SPICE DTN testbed and helped co-author the research paper.

References

[1] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets", IRB-TR-03-003, February 2003.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture", IETF RFC 4838, April 2007, http://www.ietf.org/rfc/rfc4838.txt.

[3] K. Scott and S. Burleigh, "Bundle Protocol Specification", IETF RFC 5050, November 2007, http://www.ietf.org/rfc/rfc5050.txt.

[4] DTN2 - Delay Tolerant Networking Reference Implementation, http://sourceforge.net/project/showfiles.php?group_id=101657.

[5] ION-DTN - Interplanetary Overlay Network, produced by NASA's Jet Propulsion Laboratory and partially maintained by Ohio University, https://ion.ocp.ohiou.edu.

[6] M., Doering, S., Lahde, J., Morgenroth, and L. Wolf, "IBR-DTN: an efficient implementation for embedded systems", ACM MobiCom workshop on Challenged Networks (CHANTS 2008), pp.117–120, New York, ACM, NY, 2008, http://www.ibr.cs.tu-bs.de/projects/ibr-dtn/.

[7] T. Kärkkäinen, M. Pitkänen, and J. Ott, "SCAMPI Application Platform Demo paper", ACM MobiCom workshop on Challenged Networks (CHANTS 2012), August 2012, http://dx.doi.org/10.1145/2348616.2348636.

[8] Delay Tolerant Networking Research Group, http://www.dtnrg.org/wiki/Home.

[9] T. Kärkkäinen, and J. Ott, "Liberouter: Towards Autonomous Neighborhood Networking", IEEE/IFIP Annual Conference on Wireless On-demand Network Systems and Services (WONS 2014), 2014, http://dx.doi.org/10.1109/WONS.2014.6814739.

[10] Internet Engineering Task Force, http://www.ietf.org.

[11] DTN Working Group, http://datatracker.ietf.org/wg/dtn/charter/.

[12] Consultative Committee for Space Data Systems, http://www.ccsds.org.

[13] L. Wood, W. Ivancic, W. M. Eddy, D. Stewart, J. Northam, C. Jackson, and A. Curiel, "Use of the Delay-Tolerant Networking Bundle Protocol from Space", 59th International Astronautical Congress, Glasgow, September 2008.

[14] J. Wyatt, S. Burleigh, R. Jones, L. Torgerson and S. Wissler, "Disruption Tolerant Networking Flight Validation Experiment on NASA's EPOXI Mission", 1st

International Conference on Advances in Satellite and Space Communications (SPACOMM 2009), July 2009, http://dx.doi.org/10.1109/SPACOMM.2009.39.

[15] I., Komnios, S., Diamantopoulos, and V. Tsaoussidis, "Evaluation of Dynamic DTN Routing Protocols in Space Environment", 5th International Workshop on Satellite and Space Communications (IWSSC 2009), Sienna-Tuscany, Italy, September 2009, http://dx.doi.org/10.1109/IWSSC.2009.5286392.

[16] E. Koutsogiannis, G. Papastergiou, and V. Tsaoussidis, "Evaluation of CCSDS File Delivery Protocol over Delay Tolerant Networks", 5th International Workshop on Satellite and Space Communications (IWSSC 2009), Sienna-Tuscany, Italy, September 2009, http://dx.doi.org/10.1109/IWSSC.2009.5286338.

[17] G. Papastergiou, N. Bezirgiannidis, and V. Tsaoussidis, "On the Performance of Erasure Coding over Space DTN", 10th International Conference on Wired/Wireless Internet Communications (WWIC 2012), Santorini, Greece, June 2012, http://dx.doi.org/10.1007/978-3-642-30630-3_23.

[18] G. Papastergiou, I. Alexiadis, S. Burleigh, and V. Tsaoussidis, "Delay Tolerant Payload Conditioning Protocol", Elsevier Computer Networks, vol.59, pp.244-263, February 2014, http://dx.doi.org/10.1016/j.bjp.2013.11.003.

[19] C. V. Samaras, and V. Tsaoussidis, "Design of Delay-Tolerant Transport Protocol (DTTP) and its evaluation for Mars", Acta Astronautica, vol.67, issue 7-8, October-November 2010, pp.863-880, ISSN 0094-5765, http://dx.doi.org/10.1016/j.actaastro.2010.05.025.

[20] G. Papastergiou, I. Psaras, and V. Tsaoussidis, "Deep-Space Transport Protocol: A Novel Transport Scheme for Space DTNs", Computer Communications (COMCOM), Elsevier Science, Special Issue on Delay-/Disruption-Tolerant Networks, vol.32, issue 16, pp.1757‐1767, October 2009, http://dx.doi.org/10.1016/j.comcom.2009.02.012.

[21] N. Bezirgiannidis, C. Caini, D.D. Padalino Montenero, M. Ruggieri, and V. Tsaoussidis, "Contact Graph Routing Enhancements for Delay Tolerant Space Communications", in 7th Advanced Satellite Multimedia Systems Conference (ASMS 2014), Livorno, Italy, September 2014, http://dx.doi.org/10.1109/ASMS-SPSC.2014.6934518.

[22] N. Bezirgiannidis, and V. Tsaoussidis, "Predicting Queueing Delays in Delay Tolerant Networks with Application in Space", 12th International Conference on Wired & Wireless Internet Communications (WWIC 2014), Paris, France, May 2014, http://dx.doi.org/10.1007/978-3-319-13174-0_18.

[23] N. Bezirgiannidis, S. Burleigh, and V. Tsaoussidis, “Delivery Time Estimation for Space Bundles”, IEEE Transactions on Aerospace and Electronic Systems, vol.49, no.3, pp.1897-1910, July 2013, http://dx.doi.org/10.1109/TAES.2013.6558026.

[24] N. Bezirgiannidis, F. Tsapeli, S. Diamantopoulos, and V. Tsaoussidis, “Towards Flexibility and Accuracy in Space DTN Communications”, in 8th ACM MobiCom Workshop on Challenged Networks (CHANTS 2013), Miami, Florida, USA, September 2013, http://dx.doi.org/10.1145/2505494.2505499.

[25] S.-A. Lenas, S. Burleigh, and V. Tsaoussidis, “Bundle Streaming Service: Design, Implementation and Performance Evaluation”, Transactions on Emerging Telecommunications Technologies, ISSN 2161-3915, November 2013, http://dx.doi.org/10.1002/ett.2762.

[26] I. Komnios, I. Alexiadis, N. Bezirgiannidis, S. Diamantopoulos, S.-A. Lenas, G. Papastergiou, and V. Tsaoussidis, “SPICE Testbed: A DTN Testbed for Satellite and Space Communications”, 9th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2014), Guangzhou, China, May 2014, http://dx.doi.org/10.1007/978-3-319-13326-3_20.

[27] E. Koutsogiannis, S. Diamantopoulos, G. Papastergiou, I. Komnios, A. Aggelis, and N. Peccia, “Experiences from architecting a DTN testbed”, Journal of Internet Engineering, vol.3, no.1, December 2009.

[28] C. V. Samaras, I. Komnios, S. Diamantopoulos, E. Koutsogiannis, V. Tsaoussidis, G. Papastergiou, and N. Peccia, “Extending Internet Into Space - ESA DTN Testbed Implementation and Evaluation”, 1st International Conference on Mobile Lightweight Wireless Systems (MOBILIGHT 2009), Athens, Greece, May 2009, http://dx.doi.org/10.1007/978-3-642-03819-8_37.

[29] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A Data-Oriented (and Beyond) Network Architecture”, SIGCOMM 2007, http://dx.doi.org/10.1145/1282380.1282402.

[30] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, “Networking Named Content”, ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT 2009), Rome, Italy, December 2009, http://dx.doi.org/10.1145/1658939.1658941.

[31] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, “ACT: Audio Conference Tool Over Named Data Networks”, ACM SIGCOMM workshop in Information-Centric Networking, Toronto, Canada, August 2011, http://dx.doi.org/10.1145/2018584.2018601.

[32] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. K. Ramakrishnan, "COPSS: An Efficient Content Oriented Publish/Subscribe System", ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS 2011), Brooklyn, NY, USA, October 2011, http://dx.doi.org/10.1109/ANCS.2011.27.

[33] A. Detti, N. Blefari-Melazzi, S. Salsano, and M. Pomposini, "CONET: A Content Centric InterNetworking Architecture", ACM SIGCOMM workshop in Information-Centric Networking, 2011, http://dx.doi.org/10.1145/2018584.2018598.

[34] EC PURSUIT, http://www.fp7-pursuit.eu/, 2013.

[35] D. Trossen, and G. Parisis, "Designing and Realizing an Information-Centric Internet", IEEE Communications Magazine, vol.50, no.7, pp.60-67, July 2012, http://dx.doi.org/10.1109/MCOM.2012.6231280.

[36] EC PSIRP project, http://www.psirp.org/.

[37] Y. Zhu, M. Chen, and A. Nakao, "CONIC: Content-Oriented Network with Indexed Caching", IEEE Conference on Computer Communications Workshops (INFOCOM 2010), pp.1-6, March 2010, http://dx.doi.org/10.1109/INFCOMW.2010.5466658.

[38] L. Dong, H. Liu, Y. Zhang, S. Paul, and D. Raychaudhuri, "On the Cache-and-Forward Network Architecture", IEEE International Conference on Communications (ICC 2009), June 2009, http://dx.doi.org/10.1109/ICC.2009.5199249.

[39] K. Katsaros, G. Xylomenos, and G. C. Polyzos: "MultiCache: An overlay architecture for information-centric networking", Elsevier Computer Networks (COMNET), vol.55, issue 4, pp.936-947, March 2011, http://dx.doi.org/10.1016/j.comnet.2010.12.012.

[40] N. Fujita, Y. Ishikawa, A. Iwata, and R. Izmailov, "Coarse-grain replica management strategies for dynamic replication of Web contents", Elsevier Computer Networks (COMNET), vol.45, pp. 19-34, 2004, http://dx.doi.org/10.1016/j.comnet.2004.02.006.

[41] Y. Chen, L. Qiu, W. Chen, L. Nguyen, R.H. Katz, "Efficient and adaptive Web replication using content clustering", IEEE Journal on Selected Areas in Communications, vol.21, issue 6, pp.979-994, August 2003, http://dx.doi.org/10.1109/JSAC.2003.814608.

[42] I.D. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems", SIAM Journal on Computing, vol.38, 2008, http://dx.doi.org/10.1137/080715421.

[43] D. Applegate, A. Archer, and V. Gopalakrishnan, "Optimal content placement for a large-scale VoD system", ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT 2010), Philadelphia, USA, December 2010, http://dx.doi.org/10.1145/1921168.1921174.

[44] E. J. Rosensweig, and J. Kurose, "Breadcrumbs: Efficient, Best-Effort Content Location in Cache Networks", IEEE Conference on Computer Communications (INFOCOM 2009), 2009, http://dx.doi.org/10.1109/INFCOM.2009.5062201.

[45] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and Evaluation of CCN-Caching Trees", IFIP International Conferences on Networking (NETWORKING 2011), 2011, http://dx.doi.org/10.1007/978-3-642-20757-0_7.

[46] V. Sourlas, G. S. Paschos, P. Mannersalo, P. Flegkas, and L. Tassiulas, "Modeling the dynamics of caching in content-based publish/subscribe systems", 26th ACM Symposium On Applied Computing (SAC), Taiwan, March 2011, http://dx.doi.org/10.1145/1982185.1982291.

[47] V. Sourlas, P. Flegkas, L. Gkatzikis, and L. Tassiulas, "Autonomic Cache Management in Information- Centric Networks", 13th IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), Hawaii, USA, April 2012, http://dx.doi.org/10.1109/NOMS.2012.6211890.

[48] G. Dan, "Cache-to-Cache: Could ISPs Cooperate to Decrease Peer-to-peer Content Distribution Costs?", IEEE Transactions on Parallel and Distributed Systems, vol.22, no.9, pp.1469-1482, September 2011, http://dx.doi.org10.1109/TPDS.2011.28.

[49] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, "Packet caches on routers: the implications of universal redundant traffic elimination", SIGCOMM Computer Communication Review, October 2008, http://dx.doi.org/10.1145/1402958.1402984.

[50] The IETF DECADE Working Group, https://datatracker.ietf.org/wg/decade/charter/.

[51] Akamai, http://www.akamai.com/.

[52] International telecommunication union, http://www.itu.int/.

[53] Guifi.net, http://guifi.net.

[54] Funkfeuer Free Net, http://www.funkfeuer.at.

[55] Athens Wireless Metropolitan Network, http://www.awmn.net.

[56] A. Sathiaseelan, and T. Radzik, "RD-TCP: Reorder Detecting TCP", 6th IEEE International Conference on High Speed Networks and Multimedia

Communications (HSNMC 2003), Estoril, Portugal, July 2003, http://dx.doi.org/10.1007/978-3-540-45076-4_47.

[57] A. Sathiaseelan, and T. Radzik, "Reorder Notifying TCP (RN-TCP) with Explicit Packet Drop Notification (EPDN)", International Journal of Communication Systems, Wiley, vol.19, issue 6, pp.659-678, September 2005, http://dx.doi.org/10.1002/dac.759.

[58] B. Oyelaran-Oyeyinka, and C.N. Adeya, "Internet access in Africa: Empirical evidence from Kenya and Nigeria", Telematics and Informatics, vol.21, issue 1, pp.67–81, February 2004, http://dx.doi.org/10.1016/S0736-5853(03)00023-6.

[59] E. Huerta, and R. Sandoval-Almaz, "Digital literacy: Problems faced by telecenter users in Mexico", Information Technology for Development, vol.13, issue 3, pp.217-232, 2007, http://dx.doi.org/10.1002/itdj.20071.

[60] J. Chen, "Re-architecting web and mobile information access for remote regions", Department of Mathematics, New York University, New York, September 2011.

[61] S. Surana, R. Patra, S. Nedevschi, M. Ramos, L. Subramanian, Y. Ben-David, and E. Brewer, "Beyond pilots: Keeping rural wireless networks alive", 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2008), San Francisco, CA, USA, April 2008.

[62] D. Johnson, "Evaluation of a single radio rural mesh network in South Africa", IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD 2007), Bangalore, India, December 2007, http://dx.doi.org/10.1109/ICTD.2007.4937415.

[63] J. Backens, G. Mweemba, and G. van Stam, "A rural implementation of a 52 node mixed wireless mesh network in Macha, Zambia", International Conference on e-Infrastructure and e-Services for Developing Countries (AFRICOMM 2009), December 2009, http://dx.doi.org/10.1007/978-3-642-12701-4_4.

[64] R. Morris, "TCP behavior with many flows", International Conference on Network Protocols (ICNP 1997), Washington, DC, USA, IEEE Computer Society, pp.205, 1997, http://dx.doi.org/10.1109/ICNP.1997.643715.

[65] C.M. Pazos, J.C. Sanchez Agrelo, and M. Gerla, "Using back-pressure to improve TCP performance with many flows", 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1999), vol.2, pp.431-438, March 1999, http://dx.doi.org/10.1109/INFCOM.1999.751375.

[66] L. Qiu, Y. Zhang, and S. Keshav, "Understanding the performance of many TCP flows", Elsevier Computer Networks (COMNET), vol.37, issues 3-4, pp.277–306, 2001, http://dx.doi.org/10.1016/S1389-1286(01)00203-1.

[67] L.A. Grieco, and S. Mascolo, “Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control”, SIGCOMM Computer Communication Review, vol.34, issue 2, pp.25-38, April 2004, http://dx.doi.org/10.1145/997150.997155.

[68] J. Chen, J. Iyengar, L. Subramanian, and B. Ford, “TCP behavior in sub packet regimes”, ACM joint international conference on Measurement and modeling of computer systems (SIGMETRICS 2011), ACM, New York, USA, pp.157-158, 2011, http://dx.doi.org/10.1145/1993744.1993804.

[69] S. Fortin-Parisi, and B. Sericola, “A Markov model of TCP throughput, goodput and slow start”, Journal on Performance Evaluation, vol.58, issue 2, pp.89-108, November 2004, http://dx.doi.org/10.1016/j.peva.2004.07.016.

[70] J. Chen, L. Subramanian, J. Iyengar, and B. Ford, “TAQ: Enhancing Fairness and Performance Predictability in Small Packet Regimes”, 9th European Conference on Computer Systems (EuroSys 2014), Amsterdam, Netherlands, April 2014, http://dx.doi.org/10.1145/2592798.2592819.

[71] A. Sathiaseelan, and J. Crowcroft, “LCD-Net: lowest cost denominator networking”, SIGCOMM Computer Communication Review, vol.43, issue 2, pp.52-57, April 2013, http://dx.doi.org/10.1145/2479957.2479966.

[72] D. Ros, and M. Welzl, “Less-than-Best-Effort service: A Survey of end-to-end approaches”, IEEE Communications Surveys & Tutorials, vol.15, no.2, pp.898-908, 2013, http://dx.doi.org/10.1109/SURV.2012.060912.00176.

[73] M. Welzl, S. Gjessing, and N. Khademi, “Less-than-best-effort service for Community Wireless Networks: Challenges at three layers”, 11th IEEE/IFIP Annual Conference on Wireless On-demand Network Systems and Services (WONS 2014), pp.148-153, April 2014, http://dx.doi.org/10.1109/WONS.2014.6814737.

[74] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, “Low Extra Delay Background Transport (LEDBAT)”, IETF RFC 6817, December 2012, https://tools.ietf.org/html/rfc6817

[75] uTorrent Transport Protocol, http://www.bittorrent.org/beps/bep_0029.html.

[76] N. Kuhn, O. Mehani, A. Sathiaseelan, and E. Lochin, “Less-than-Best-Effort capacity sharing over high BDP networks with LEDBAT”, IEEE 78th Vehicular Technology Conference (VTC Fall 2013), pp.1-5, September 2013, http://dx.doi.org/10.1109/VTCFall.2013.6692266.

[77] J. Schneider, J. Wagner, R. Winter, and H. Kolbe, “Out of my way - evaluating low extra delay background transport in an ADSL access network”, 22nd

International Teletraffic Congress (ITC 2010), pp.1-8, September 2010, http://dx.doi.org/10.1109/ITC.2010.5608714.

[78] D. Rossi, C. Testa, and S. Valenti, "Yes, we LEDBAT: playing with the new BitTorrent congestion control algorithm", Passive and Active Measurement (PAM 2010), Zurich, Switzerland, April 2010, http://dx.doi.org/10.1007/978-3-642-12334-4_4.

[79] D. Rossi, C. Testa, S. Valenti, and L. Muscariello, "LEDBAT: the new BitTorrent congestion control protocol", International Conference on Computer Communication Networks (ICCCN 2010), Zurich, Switzerland, August 2010, http://dx.doi.org/10.1109/ICCCN.2010.5560080.

[80] A.J. Abu, and S. Gordon, "Performance analysis of LEDBAT in the presence of TCP New Reno", International Conference on Network Technologies and Communications (NTC 2010), pp.65-70, Phuket, Thailand, November 2010, http://dx.doi.org/10.5176/978-981-08-7656-2_NTC2010-41.

[81] A.J. Abu, and S. Gordon, "A framework for dynamically selecting gain in LEDBAT algorithm for high speed networks", Information Technology Journal, vol.10, pp.358-366, 2011, http://dx.doi.org/10.3923/itj.2011.358.366.

[82] A.J. Abu, and S. Gordon, "Impact of delay variability on LEDBAT performance", IEEE International Conference on Advanced Information Networking and Applications (AINA 2011), pp.708–715, March 2011, http://dx.doi.org/10.1109/AINA.2011.98.

[83] M. Kühlewind, and S. Fisches, "Evaluation of different decrease schemes for LEDBAT congestion control", 17th international conference on Energy-aware communications (EUNICE 2011), Springer-Verlag, Berlin, Heidelberg, pp.112-123, 2011, http://dx.doi.org/10.1007/978-3-642-23541-2_13

[84] G. Carofiglio, L. Muscariello, D. Rossi, and C. Testa, "A hands-on Assessment of Transport Protocols with Lower than Best Effort Priority", 35th IEEE Conference on Local Computer Networks (LCN 2010), Denver, CO, USA, October 2010, http://dx.doi.org/10.1109/LCN.2010.5735831.

[85] G. Carofiglio, L. Muscariello, D. Rossi, and S. Valenti, "The quest for LEDBAT fairness", IEEE GLOBECOM 2010, Miami, FL, USA, December 2010, http://dx.doi.org/10.1109/GLOCOM.2010.5683559.

[86] D. Ros, and M. Welzl, "Assessing LEDBAT's delay impact", IEEE Communications Letters, vol.17, no.5, pp.1044-1047, May 2013, http://dx.doi.org/10.1109/LCOMM.2013.040213.130137.

[87] G. Carofiglio, L. Muscariello, D. Rossi, C. Testa, and S. Valenti, "Rethinking low extra delay backtround transport protocols", Elsevier Computer Networks (COMNET), vol.57, pp.1838-1852, June 2013, http://dx.doi.org/10.1016/j.comnet.2013.02.020.

[88] I. Komnios, A. Sathiaseelan, and J. Crowcroft, "LEDBAT Performance in Sub-packet Regimes", 11th IEEE/IFIP Annual Conference on Wireless On-Demand Network Systems and Services (WONS 2014), Obergurgl, Austria, April 2014, http://dx.doi.org/10.1109/WONS.2014.6814738.

[89] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017.

[90] H. Zhou, K. Sparks, N. Gopalakrishnan, P. Monogioudis, F. Dominique, P. Busschbach, and J. Seymour, "Deprioritization of heavy users in wireless networks", IEEE Communications Magazine, vol.49, no.10, pp.110-117, 2011, http://dx.doi.org/10.1109/MCOM.2011.6035824.

[91] S. M. S. Nirjon, A. Nicoara, C.-H. Hsu, J. Singh, and J. Stankovi, "MultiNets: Policy Oriented Real-Time Switching of Wireless Interfaces on Mobile Devices", ACM Transactions on Embedded Computer Systems, vol.13, issue 4, April 2014, http://dx.doi.org/10.1145/2489788.

[92] M.-R. Ra, J. Paek, A.B. Sharma, R. Govindan, M.H. Krieger, and M.J. Neely, "Energy-Delay Tradeoffs in Smartphone Applications", 8th International conference on Mobile systems, applications, and services (MobiSys 2010), ACM, New York, NY, USA, pp.255-270, 2010, http://dx.doi.org/10.1145/1814433.1814459.

[93] A. Rahmati, and L. Zhong, "Context-for-wireless: context-sensitive energy-efficient wireless data transfer", 5th ACM International conference on Mobile systems, applications and services (MobiSys 2007), New York, NY, USA, pp.165-178, http://dx.doi.org/10.1145/1247660.1247681.

[94] K.-H. Kim, A.W. Min, D. Gupta, P. Mohapatra, and J.P. Singh, "Improving energy efficiency of Wi-Fi sensing on smartphones", IEEE Conference on Computer Communications (INFOCOM 2011), pp.2930-2938, April 2011, http://dx.doi.org/10.1109/INFCOM.2011.5935133.

[95] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "CoolSpots: Reducing the Power Consumption of Wireless Mobile Devices with Multiple Radio Interfaces", International conference on Mobile systems, applications, and services (MobiSys 2006), pp.220-232, 2006, http://dx.doi.org/10.1145/1134680.1134704.

[96] S. Sharafeddine, K. Jahed, N. Abbas, E. Yaacoub, and Z. Dawy, "Exploiting Multiple Wireless Interfaces in Smartphones for Traffic Offloading", IEEE BlackSeaCom, 2013, http://dx.doi.org/10.1109/BlackSeaCom.2013.6623398.

[97] A. Bar, D. Mimran, L. Chekina, Y. Elovici, and B. Shapira, "Nesto - Network Selection and Traffic Offloading System for Android Mobile Devices", 9th International Wireless Communications and Mobile Computing Conference (IWCMC 2013), pp.337-342, 2013, http://dx.doi.org/10.1109/IWCMC.2013.6583582.

[98] L. Song, U. Deshpande, U. C. Kozat, D. Kotz, and R. Jain, "Predictability of WLAN Mobility and its Effects on Bandwidth Provisioning", IEEE Conference on Computer Communications (INFOCOM 2006), 2006, http://dx.doi.org/10.1109/INFOCOM.2006.171.

[99] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems", 9th international conference on Pervasive computing (Pervasive 2011), USA, 2011, http://dx.doi.org/10.1007/978-3-642-21726-5_10.

[100] D. Fang, "Moving Object Trajectory Based Spatio-Temporal Mobility Prediction", Master's of Science Thesis in Geoinformatics, KTH, Stockholm, Sweden, 2012.

[101] D. Ashbrook, and T. Starner, "Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users", Journal of Personal and Ubiquitous Computing, vol.7, no.5, 2003, http://dx.doi.org/10.1007/s00779-003-0240-0.

[102] A. J. Nicholson, and B. D. Noble, "BreadCrumbs: Forecasting Mobile Connectivity", ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2008), pp.46–57, ACM, New York, NY, USA, 2008, http://dx.doi.org/10.1145/1409944.1409952.

[103] W. Moreira, P. Mendes, and S. Sargento, "Opportunistic Routing Based on Daily Routines", IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2012), 2012, http://dx.doi.org/10.1109/WoWMoM.2012.6263749.

[104] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using Cell-ID sequence matching", 9th international conference on Mobile systems, applications, and services (MobiSys 2011), ACM, New York, NY, USA, pp.293-306, http://dx.doi.org/10.1145/1999995.2000024.

[105] I. Constandache, R.R. Choudhury, I. Rhee, "Towards Mobile Phone Localization without War-Driving", IEEE Conference on Computer Communications (INFOCOM 2010), pp.1-9, March 2010, http://dx.doi.org/10.1109/INFCOM.2010.5462058.

[106] S.-I. Sou, "Mobile Data Offloading With Policy and Charging Control in 3GPP Core Network", IEEE Transactions on Vehicular Technology, vol.62, no.7, pp.3481-3486, September 2013, http://dx.doi.org/10.1109/TVT.2013.2255899.

[107] D.H. Hagos, and R. Kapitza, "Study on performance-centric offload strategies for LTE networks", 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2013), pp.1-10, April 2013, http://dx.doi.org/10.1109/WMNC.2013.6548999.

[108] V.F.S. Mota, D.F. Macedo, Y. Ghamri-Doudane, and J.M.S. Nogueira, "On the feasibility of WiFi offloading in urban areas: The Paris case study", IFIP Wireless Days, pp.1-6, November 2013, http://dx.doi.org/10.1109/WD.2013.6686530.

[109] A.Y. Ding, B. Han, Y. Xiao, P. Hui, A. Srinivasan, M. Kojo, and S. Tarkoma, "Enabling energy-aware collaborative mobile data offloading for smartphones", 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2013), pp.487-495, June 2013, http://dx.doi.org/10.1109/SAHCN.2013.6645020.

[110] A. Aijaz, H. Aghvami, and M. Amani, "A survey on mobile data offloading: technical and business perspectives", IEEE Wireless Communications, vol.20, no.2, pp.104-112, April 2013, http://dx.doi.org/10.1109/MWC.2013.6507401.

[111] J. Roh, Y. Ji, Y. G. Lee, and T. Ahn, "Femtocell Traffic Offload Scheme for Core Networks", 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2011), Paris, France, pp.1–5, 2011, http://dx.doi.org/10.1109/NTMS.2011.5721037.

[112] L. Gao, G. Iosifidis, J. Huang, and L. Tassiulas, "Economics of mobile data offloading", IEEE Conference on Computer Communications (INFOCOM 2013), pp.3303-3308, April 2013, http://dx.doi.org/10.1109/INFCOM.2013.6567155.

[113] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. "Mobile data offloading: how much can WiFi deliver?", IEEE/ACM Transactions on Networking, vol.21, issue 2, pp.536-550, April 2013, http://dx.doi.org/10.1109/TNET.2012.2218122.

[114] X. Zhuo, W. Gao, G. Cao, and S. Hua, "An Incentive Framework for Cellular Traffic Offloading", IEEE Transactions on Mobile Computing, vol.13, no.3, pp.541-555, March 2014, http://dx.doi.org/10.1109/TMC.2013.15.

[115] V. Misra, S. Ioannidis, A. Chaintreau, and L. Massouli, "Incentivizing Peer-Assisted Services: A Fluid Shapley Value Approach", ACM SIGMETRICS 2010, NY, USA, pp.215-226, 2010, http://dx.doi.org/10.1145/1811039.1811064.

[116] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting Mobile 3G using WiFi", International conference on Mobile systems, applications, and services (MobiSys 2010), 2010, http://dx.doi.org/10.1145/1814433.1814456.

[117] M.H. Cheung, and J. Huang, “Optimal Delayed Wi-Fi Offloading”, 11th IEEE International Symposium on Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt 2013), 2013.

[118] S. Dimatteo, P. Hui, B. Han, and V. Li, “Cellular Traffic Offloading through wi-fi Networks”, IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS 2011), 2011, http://dx.doi.org/10.1109/MASS.2011.26.

[119] M. Papadopouli, and H. A. Schulzrinne, “Performance Analysis of 7DS a Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users”, IEEE Sarnoff Symposium, 2001.

[120] H. Luo, X. Meng, R. Ramjee, P. Sinha, and L. Li, “The Design and Evaluation of Unified Cellular and Ad-Hoc Networks”, IEEE Transactions on Mobile Computing, vol.6, no.9, pp.1060-1074, September 2007 http://dx.doi.org/10.1109/TMC.2007.1035.

[121] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, “Multiple mobile data offloading through delay tolerant networks”, ACM MobiCom workshop on Challenged Networks (CHANTS 2011), Las Vegas, Nevada, USA, September 2011, http://dx.doi.org/10.1145/2030652.2030665.

[122] B. Han, P. Hui, V. Kumar, M. Marathe, J. Shao, and A. Srinivasan, “Mobile data offloading through opportunistic communications and social participation”, IEEE Transactions on Mobile Computing, vol.11, no.5, pp.821-834, May 2012, http://dx.doi.org/10.1109/TMC.2011.101.

[123] P. Baier, F. Durr, and K. Rothermel, “Tomp: “Opportunistic traffic offloading using movement predictions”, IEEE Conference on Local Computer Networks (LCN 2012), pp.50–58, 2012, http://dx.doi.org/10.1109/LCN.2012.6423668.

[124] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, and M. De Amorim, “Push-and-track: Saving infrastructure bandwidth through opportunistic forwarding”, Pervasive and Mobile Computing, vol.8, issue 5, pp.682-697, October 2012, http://dx.doi.org/10.1016/j.pmcj.2012.02.001.

[125] W. Peng, F. Li, X. Zou, and J. Wu, “The Virtue of Patience: Offloading Topical Cellular Content through Opportunistic Links”, 10th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2013), pp.402-410, Octobet 2013, http://dx.doi.org/10.1109/MASS.2013.69.

[126] N. Ristanovic, J. Le Boudec, A. Chaintreau, and V. Erramilli, “Energy efficient offloading of 3g networks”, IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2011), 2011, http://dx.doi.org/10.1109/MASS.2011.27.

[127] V.A. Siris, and M. Anagnostopoulou, “Performance and energy efficiency of mobile data offloading with mobility prediction and prefetching”, 14th IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2013), pp.1-6, June 2013, http://dx.doi.org/10.1109/WoWMoM.2013.6583450.

[128] J. Scott, P. Hui, J. Crowcroft, and C. Diot, “Haggle: A Networking Architecture Designed Around Mobile Users”, IEEE/IFIP Annual Conference on Wireless On-demand Network Systems and Services (WONS 2006), 2006.

[129] M. Pitkänen, T. Märkkäinen, J. Ott, M. Conti, A. Passarella, S. Giordano, D. Pucinelli, F. Legendre, S. Trifunonic, K. Hummel, M. May, N. Hedge, and A. Spyropoulos, “SCAMPI: Service platform for soCial Aware Mobile and Pervasive computing”, ACM SIGCOMM workshop on Mobile Cloud Computing (MCC), August 2012, http://dx.doi.org/10.1145/2342509.2342512.

[130] D. Merani, A. Berni, J. Potter, and R. Martins, “An Underwater Convergence Layer for Disruption Tolerant Networking”, IEEE Baltic Congress on Future Internet Communications (BCFIC 2011), pp.103-108, February 2011, http://dx.doi.org/10.1109/BCFIC-RIGA.2011.5733227.

[131] A. Berni, D. Merani, J. Potter, and R. Been, “Heterogeneous System Framework For Underwater Networking”, IEEE Military Communications Conference 2011, pp. 2050-2056, November 2011, http://dx.doi.org/10.1109/MILCOM.2011.6127620.

[132] Sensor Networking with Delay Tolerance (SeNDT) project, http://down.dsg.cs.tcd.ie/sendt/.

[133] Environmental Monitoring in Metropolitan Areas project, https://www.ibr.cs.tu-bs.de/projects/emma/.

[134] Opportunistic Public Transport Communication project, https://www.ibr.cs.tu-bs.de/projects/optracom/.

[135] Consumer Generated Mobile Wireless Media project, http://anr-crowd.lip6.fr/index.php/Main/HomePage.

[136] N. Banerjee, M.D. Corner, and B.N. Levine, “An Energy-Efficient Architecture for DTN Throwboxes”, IEEE Conference on Computer Communications (INFOCOM 2007), pp.776-784, 2007, http://dx.doi.org/10.1109/INFCOM.2007.96.

[137] M. Ho, and K. Fall, “Poster: Delay tolerant networking for sensor networks”, 1st IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004), 2004.

[138] S. Jain, K. Fall, and R. Patra, “Routing in a Delay Tolerant Network”, SIGCOMM, Portland, Oregon, USA, August-September 2004, http://dx.doi.org/10.1145/1015467.1015484.

[139] A. Doria, M. Uden, D.P. Pandey, “Providing connectivity to the Saami nomadic community”, International Conference on Open Collaborative Design for Sustainable Innovation, 2002.

[140] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, “Low-cost communication for rural Internet kiosks using mechanical backhaul”, ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2006), pp.334–345, 2006, http://dx.doi.org/10.1145/1161089.1161127.

[141] Networking for Communications Challenged Communities: Architecture, Test Beds and Innovative Alliances project, http://www.n4c.eu.

[142] W. Zhao, M. Ammar, and E. Zegura, “A message ferrying approach for data delivery in sparse mobile ad hoc networks”, 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc 2004), ACM, New York, NY, USA, pp.187-198, http://dx.doi.org/10.1145/989459.989483.

[143] L.D. Hartog, T. Spyropoulos, and F. Legendre, “Using Public Transportation as a DTN Backbone: Mobility Properties and Performance Analysis”, IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2010), Montreal, Canada, June 2010.

[144] A. Keränen, and J. Ott, “DTN over aerial carriers”, 4th ACM workshop on Challenged networks (CHANTS 2009), ACM, New York, NY, USA, pp.67-76, 2009, http://doi.acm.org/10.1145/1614222.1614234.

[145] M. Zarafshan-Araki, and K.-W. Chin, “TrainNet: A transport system for delivering non real-time data”, Elsevier Computer Communications (COMCOM), vol.33, issue 15, pp.1850-1863, ISSN 0140-3664, September 2010, http://dx.doi.org/10.1016/j.comcom.2010.06.008.

[146] L.-P. Tung, Y.-C. Chen, K. Lan, and L.-J. Chen, “MetroNet: a disruption-tolerant approach for mobile downloads on metro systems”, Transactions on Emerging Telecommunications Technologies, vol.25, issue 8, pp.835-851, 2014, http://dx.doi.org/10.1002/ett.2695.

[147] F. Yasmeen, N. Huda, S. Yamada, and C. Borcea, “Ferry access points and sticky transfers: Improving communication in ferry-assisted DTNs”, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2012), pp.1-7, June 2012, http://dx.doi.org/10.1109/WoWMoM.2012.6263746.

[148] T. Kärkkäinen, J. Ott, and M. Pitkänen, "Applications in Delay-Tolerant and Opportunistic Networks", Book chapter in S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic: "Mobile Ad Hoc Networking: The Cutting Edge Directions", John Wiley & Sons, Inc., ISBN 978-1-118-08278-2, 2013, http://dx.doi.org/10.1002/9781118511305.ch9.

[149] A. Balasubramanian, Y. Zhou, W.B. Croft, B.N. Levine, and A. Venkataramani, "Web Search from a Bus", ACM MobiCom workshop on Challenged Networks (CHANTS 2007), 2007, http://dx.doi.org/10.1145/1287791.1287803.

[150] M. Pitkänen, T. Kärkkäinen, and J. Ott, "Opportunistic Web Access via WLAN Hotspots", 8th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2010), Mannheim, Germany, March 2010, http://dx.doi.org/10.1109/PERCOM.2010.5466997.

[151] DTN-Blogging, http://www.netlab.tkk.fi/u/jo/dtn/2007-Lauri-Peltola-DTN-Blogging.pdf.

[152] T. Hyyrylainen, T. Karkkainen, C. Luo, V. Jaspertas, J. Karvo, and J. Ott, "Opportunistic Email Distribution and Access in Challenged Heterogeneous Environment s", ACM Workshop on Challenged Networks (CHANTS 2007), 2007, http://dx.doi.org/10.1145/1287791.1287811.

[153] Thunderbird E-mail Client, https://www.mozilla.org/en-US/thunderbird/.

[154] A. Lindgren, "Social Networking in a Disconnected Network: fbDTN: Facebook over DTN", 6th ACM Workshop on Challenged Networks (CHANTS 2011), pp.69–70, 2011, http://dx.doi.org/10.1145/2030652.2030674.

[155] K. Zaragoza, N. Thai, and T. Christensen, "An implementation for accessing Twitter across challenged networks", 6th ACM Workshop on Challenged Networks (CHANTS 2011), pp.71–72, 2011, http://dx.doi.org/10.1145/2030652.2030675.

[156] DT-Talkie, http://www.netlab.tkk.fi/tutkimus/dtn/dttalkie/.

[157] NOKIA N800, http://www.microsoft.com/en-us/mobile/support/product/nokia-n800/.

[158] A. Voyiatzis, "A Survey of Delay- and Disruption-Tolerant Networking Applications", Journal of Internet Engineering, vol.5, issue 1, pp.331-344, June 2012, Kleidarithmos Press.

[159] J. Ott, E. Hyytiä, P. Lassila, J. Kangasharju, and S. Santra, "Floating Content for Probabilistic Information Sharing", Elsevier Personal Wireless Communications (PMC), vol.7, issue 6, pp.671-689, December 2011, http://dx.doi.org/10.1016/j.pmcj.2011.09.001.

[160] A. Vahdat, and D. Becker, "Epidemic routing for partially connected ad hoc networks", Technical Report CS-200006, Duke University, 2000).

[161] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic routing protocol for intermittently connected networks", RFC 6693, 2011, https://tools.ietf.org/html/rfc6693.

[162] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a DTN routing protocol - PRoPHETv2", 6th ACM workshop on Challenged networks (CHANTS 2011), New York, USA, pp.27-30, http://dx.doi.org/10.1145/2030652.2030661.

[163] P. Sok, S. Tan, and K. Kim, "PRoPHET Routing Protocol Based on Neighbor Node Distance Using a Community Mobility Model in Delay Tolerant Networks", IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC 2013), pp.1233-1240, November 2013, http://dx.doi.org/10.1109/HPCC.and.EUC.2013.175.

[164] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks", IEEE Transactions on Mobile Computing, vol.10, no.11, pp.1576-1589, November 2011, http://dx.doi.org/10.1109/TMC.2010.246.

[165] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks", ACM SIGCOMM Workshop on Delay Tolerant Networking (WDTN 2005), 2005, http://dx.doi.org/10.1145/1080139.1080143.

[166] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility", 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops 2007), pp.79-85, March 2007, http://dx.doi.org/10.1109/PERCOMW.2007.108.

[167] J. Burgess, B. Gallagher, and B.N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks", IEEE Conference on Computer Communications (INFOCOM 2006), 2006, http://dx.doi.org/10.1109/INFOCOM.2006.228.

[168] I. Leontiadis, and C. Mascolo, "GeOpps: Geographical Opportunistic Routing for Vehicular Networks", IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), pp.1-6, June 2007, http://dx.doi.org/10.1109/WOWMOM.2007.4351688.

[169] V. Soares, J. Rodrigues, and F. Farahmand, "GeoSpray: A geographic routing protocol for vehicular delay-tolerant networks", Information Fusion, vol.15, pp.102-113, 2014, http://dx.doi.org/10.1016/j.inffus.2011.11.003.

[170] M. Sede, X. Li, D. Li, M-Y. Wu, M. Li, W. Shu, "Routing in Large-Scale Buses Ad Hoc Networks", IEEE Wireless Communications and Networking Conference (WCNC 2008), 2008, http://dx.doi.org/10.1109/WCNC.2008.475.

[171] M. Demmer, and K. Fall, "DTLSR: delay tolerant routing for developing regions", Workshop on Networked systems for developing regions (NSDR 2007), 2007, http://dx.doi.org/10.1145/1326571.1326579.

[172] Y. Xian, C. Huang, and J. Cobb, "Look-ahead routing and message scheduling in delay-tolerant networks", IEEE Conference on Local Computer Networks (LCN 2010), 2010, http://dx.doi.org/10.1109/LCN.2010.5735753.

[173] B. K. Polat, M. Khan, N. Tuli, and P. Kailay, "ZBMF: Zone Based Message Ferrying for Disruption Tolerant Networks", Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-642-00205-2_6.

[174] H. Dang and H. Wu, "Clustering and cluster-based routing protocol for delay-tolerant mobile networks", IEEE Transactions on Wireless Communications, vol.9, no.6, pp.1874-1881, June 2010, http://dx.doi.org/10.1109/TWC.2010.06.081216.

[175] Android Geofences, http://developer.android.com/training/location/geofencing.html.

[176] I. Komnios, F. Tsapeli, and S. Gorinsky, "Cost-Effective Multi-Mode Offloading with peer-assisted communications", Ad Hoc Networks, Volume 25, Part B, pp.370-382, ISSN 1570-8705, February 2015, http://dx.doi.org/10.1016/j.adhoc.2014.07.028.

[177] J. Lee, Y. Yi, S. Chong, and Y. Jin, "Economics of WiFi Offloading: Trading Delay for Cellular Capacity", IEEE Conference on Computer Communications (INFOCOM 2013), 2013, http://dx.doi.org/10.1109/INFCOMW.2013.6562888.

[178] 3GPP Technical Report 23.829 "Local IP Access and Selected IP Traffic Offload (LIPA-SIPTO)".

[179] D. Leroy, G. Detal, J. Cathalo, M. Manulis, F. Koeune, and O. Bonaventure, "SWISH: Secure WiFi Sharing", Elsevier Computer Networks, vol.55, no7, pp.1614-1630, 2011, http://dx.doi.org/10.1016/j.comnet.2011.01.012.

[180] CISCO, "The Future of Hotspots: Making Wi-Fi as Secure and Easy to Use as Cellular", White Paper, 2012.

[181] G. Thraves, A. Urueta, P. Vidales, and M. Solarski, "Driving the deployment of citywide ubiquitous WiFi access", 1st International Conference on Simulation Tools and Techniques for Communications, 2008.

[182] I. Komnios, and V. Tsaoussidis, "CARPOOL: Extending Free Internet Access over DTN in Urban Environments", ACM MobiCom Workshop on Lowest Cost

Denominator Networking for Universal Access (LCDNet 2013), Miami, Florida, USA, September 2013, http://dx.doi.org/10.1145/2502880.2502891.

[183] I. Komnios, and V. Tsaoussidis, "CARPOOL: Connectivity Plan Routing Protocol", 12th International Conference on Wired & Wireless Internet Communications (WWIC 2014), Paris, France, May 2014, http://dx.doi.org/10.1007/978-3-319-13174-0_21.

[184] S. Burleigh, "Contact Graph Routing", Internet Draft, 2010.

[185] Raspberry Pi, http://www.raspberrypi.org.

[186] M. Demmer, and K. Fall, "The design and implementation of a session layer for delay-tolerant networks", Elsevier Computer Communications (COMCOM), vol.32, issue 16, pp.1724-1730, 2009, http://dx.doi.org/10.1016/j.comcom.2009.02.010

[187] S. Symington, S. Farrell, H. Weiss, and P. Lovell, "Bundle Security Protocol Specifications", RFC 6257, 2011.

[188] I. Komnios and E. Kalogeiton, "A DTN-based architecture for public transport networks", Annals of Telecommunications, Springer, ISSN 0003-4347, August 2015, http://dx.doi.org/10.1007/s12243-015-0473-8

[189] S. McCanne, and S. Floyd, "Network Simulator - ns (version 2)", http://www.isi.edu/nsnam/ns.

[190] D. Rossi, "Ledbat module for ns-2", http://perso.telecom-paristech.fr/~drossi/index.php?n=Software.LEDBAT.

[191] Z. Koradia, G. Mannava, A. Raman, G. Aggarwal, V. Ribeiro, A. Seth, S. Ardon, A. Mahanti, and S. Triukose, "First impressions on the state of cellular data connectivity in India", 4th ACM Annual Symposium on Computing for Development (DEV-4 2013), ACM, New York, NY, USA, 2013, http://doi.acm.org/10.1145/2537052.2537064.

[192] M. Chetty, S. Sundaresan, S. Muckaden, N. Feamster, and E. Calandro, "Measuring broadband performance in South Africa", 4th ACM Annual Symposium on Computing for Development (DEV-4 2013), ACM, New York, NY, USA, 2013, http://dx.doi.org/10.1145/2537052.2537053.

[193] S. Floyd, and V. Jacobson, "Random Early Detection (RED) gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, vol.1, issue 4, pp.397–413, August 1993, http://dx.doi.org/10.1109/90.251892.

[194] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and

congestion avoidance in the Internet", IETF RFC 2309, April 1998, https://tools.ietf.org/html/rfc2309.

[195] A. Keranen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation", 2nd International Conference on Simulation Tools and Techniques (SIMUTools 2009), New York, 2009, http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5674.

[196] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working Day Movement Model", 1st ACM/SIGMOBILE Workshop on Mobility Models for Networking Research, 2008, http://dx.doi.org/10.1145/1374688.1374695.

[197] Y. Gong, D. Rossi, C. Testa, S. Valenti, and D. Taht, "Interaction or interference: can AQM and low priority congestion control successfully collaborate?", ACM ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT 2012), Student Workshop, pp.25-26, December 2012, http://dx.doi.org/10.1145/2413247.2413263.

[198] M. Chen, X. Fan, M.N. Murthi, T.D. Wickramarathna, and K. Premaratne, "Normalized queueing delay: congestion control jointly utilizing delay and marking", IEEE/ACM Transactions on Networking, vol.17, issue 2, pp.618-631, April 2009, http://dx.doi.org/10.1109/TNET.2008.926508.

[199] M. Yousaf, M. Welzl, and B. Yener, "Accurate shared bottleneck detection and file transfer delay prediction for grid scheduling", University of Innsbruck, Austria, December 2008.

[200] A. Sathiaseelan, D. Trossen, I. Komnios, J. Ott, and J. Crowcroft, "An Internet Architecture for the Challenged", IAB Workshop on Internet Technology Adoption and Transition (ITAT 2013), Cambridge, UK, December 2013.

[201] A. Sathiaseelan, D. Trossen, I. Komnios, J. Ott, and J. Crowcroft, "Information centric delay tolerant networking: An Internet Architecture for the challenged", Technical Report UCAM-CL-TR-841, ISSN 1476-2986.

[202] Portable Satellite Simulator, http://www.spacelinkngt.com/PSSIMBU.html.

[203] CORTEX CRT system, http://www.zds-fr.com/en/products/10/satellite-ttc.html.

[204] Satellite Tool Kit, http://www.agi.com/products/.

[205] N. Bezirgiannidis, and V. Tsaoussidis, "Packet size and DTN transport service: Evaluation on a DTN Testbed", International Congress on Ultra Modern Telecommunications and Control Systems, Moscow, October 2010, http://dx.doi.org/10.1109/ICUMT.2010.5676669.

[206] S.-A. Lenas, S. Burleigh, and V. Tsaoussidis, "Reliable Data Streaming over Delay Tolerant Networks", 10th International Conference on Wired/Wireless

Internet Communications (WWIC 2012), Santorini, Greece, June 2012, http://dx.doi.org/10.1007/978-3-642-30630-3_33.

[207] N.L. Clarke, V. Katos, S.-A. Menesidou, B. Ghita, and S. Furnell, "A novel security architecture for a space-data DTN", 10th International Conference on Wired/Wireless Internet Communications (WWIC 2012), Santorini, Greece, June 2012, http://dx.doi.org/10.1007/978-3-642-30630-3_31.

[208] EC's FP7 Space Internetworking Center (SPICE) project, http://www.spice-center.org.

[209] FP7 Space-Data Routers for Exploiting Space Data project, http://www.spacedatarouters.eu.

[210] Proximity-1 Space Link Protocol, CCSDS 211.0-R-3.1, 2002.

[211] CCSDS Telemetry (TM) Space Data Link protocol Recommendation for Space Data Systems Standards, CCSDS 132.0-B, 2003.

[212] CCSDS Telecommand (TC) Space Data Link protocol Recommendation for Space Data Systems Standards, CCSDS 232.0-B-2, 2010.

[213] AOS Space Data Link Protocol, CCSDS 732.0-B-1, Blue Book, Issue 1, 2003.

[214] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol", IETF RFC 5326, 2008.

[215] CCSDS File Delivery Protocol (CFDP) Recommendation for Space Data System Standards, CCSDS 727.0FBF4, 2007.

[216] Asynchronous Message Service (AMS), http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/37837/1/05-0923.pdf.

[217] SIMSAT simulator, http://www.terma.com/space/ground-segment/satellite-simulators/.

[218] Network Emulator, http://www.linuxfoundation.org/collaborate/workgroups/networking/netem.

[219] UMOBILE project, http://www.umobile-project.org

Cirriculum Vitae

Ioannis Komnios was born in Thessaloniki, Greece, on October 5^{th}, 1984. In 2007, he received his Diploma in Electrical and Computer Engineering from the Democritus University of Thrace, Xanthi, Greece, and two years later, in 2009, he received a distinction for his MSc in Computer Networks from the same department.

Since 2010 Ioannis is a researcher at the Space Internetworking Center (SPICE), while in 2011 he served as a visiting researcher in AALTO University, Finland, where he collaborated with the computer networking research group on the applicability of delay-tolerant networking technology in maritime scenarios.

Ioannis has worked on five research project funded by the European Commission and the European Space Agency and has also keenly participated in the preparation of research proposals, being a key author in several research projects.

Ioannis has published several research papers mostly focusing on Delay Tolerant Networks, cellular data offloading and extending Internet access, which are his main research interests.

Journal Papers

1. **I. Komnios** and E. Kalogeiton, "A DTN Architecture for Public Transport Networks", Annals of Telecommunications, Springer, August 2015, doi: 10.1007/s12243-015-0473-8.
2. **I. Komnios**, T. Tsapeli and S. Gorinsky, "Cost-Effective Multi-Mode Offloading with peer-assisted communications", Ad Hoc Networks, Elsevier, Available online 7 August 2014, doi: 10.1016/j.adhoc.2014.07.028.
3. E. Koutsogiannis, S. Diamantopoulos, G. Papastergiou, **I. Komnios**, A. Aggelis and N. Peccia, "Experiences from architecting a DTN Testbed", Journal of Internet Engineering, Klidarithmos Press, Vol. 3, No. 1.

Conference Papers

1. **I. Komnios** and V. Tsaoussidis, "CARPOOL: Connectivity Plan Routing Protocol", 12th International Conference on Wired & Wireless Internet Communications (WWIC 2014), Paris, France, May 26-28, 2014.
2. **I. Komnios**, I. Alexiadis, N. Bezirgiannidis, S. Diamantopoulos, S.-A. Lenas, G. Papastergiou and V. Tsaoussidis, "SPICE Testbed: A DTN Testbed for Satellite and Space Communications", 9th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2014), Guangzhou, China, May 5-7, 2014.
3. **I. Komnios**, A. Sathiaseelan and J. Crowcroft, "LEDBAT Performance in Sub-packet Regimes", 11th IEEE/IFIP Annual Conference on Wireless On-Demand Network Systems and Services (IEEE WONS 2014), Obergurgl, Austria, April 2-4, 2014.
4. A. Sathiaseelan, D. Trossen, **I. Komnios**, J. Ott and J. Crowcroft, "An Internet Architecture for the Challenged", IAB Workshop on Internet Technology Adoption and

Transition (ITAT), Cambridge, UK, December 4-6, 2013.

5. **I. Komnios** and V. Tsaoussidis, "CARPOOL: Extending Free Internet Access over DTN in Urban Environments", in ACM MobiCom Workshop on Lowest Cost Denominator Networking for Universal Access (ACM LCDNet '13), Miami, Florida, USA, September 30, 2013.
6. C. Samaras, **I. Komnios**, S. Diamantopoulos, E. Koutsogiannis, V. Tsaoussidis, G. Papastergiou and N. Peccia, "Extending Internet into Space – ESA DTN Testbed Implementation and Evaluation", 1st International Conference on Mobile Lightweight Wireless Systems, Workshop on Research activities funded by ESA in Greece, Athens, Greece, May 18-20, 2009.
7. **I. Komnios**, S. Diamantopoulos and V. Tsaoussidis, "Evaluation of Dynamic DTN Routing Protocols in Space Environment", 5th International Workshop on Satellite and Space Communications, Sienna, Italy, 2009.
8. S. Lenas, **I. Komnios** and V. Tsaoussidis, "Comparison Between Static and Dynamic Routing on Networks that Utilize Constantly Changing Connectivity Maps", Eurhka 2009, Corfu, Greece, 2009.

Research Projects

- **UMOBILE** - Universal, mobile-centric and opportunistic communications architecture, H2020 Project funded by the European Commission (Grant Agreement No 645124, H2020-ICT-2014-1),
  Role: *Main Author of the Proposal, Technical Manager and WP leader*
- **SPICE** - Space Internetworking Center, FP-7 Project funded by the European Commission (Grant Agreement No 264226, FP7-REGPOT-2010-1),
  Role: *Main Author of the Proposal, Technical Manager and WP leader*
- **SDR** - Space Data Routers, FP-7 Project funded by the European Commission (Grant Agreement No 263330, FP7-SPACE-2010-1),
  Role: *Contributing Author to the Proposal, Researcher*
- **Extending Internet Into Space – Phase 3 –** ESA/ESOC DTN/IP Testbed Deployment and Optimization, Funded by the European Space Agency (ESA), 2010-2011,
  Role: Researcher
- **Extending Internet Into Space – Phase 2** – DTN/IP Testbed Implementation and Evaluation, Funded by the European Space Agency (ESA), 2008-2010, *Role: Researcher*

Editor, Organiser and Reviewer Activities

- TPC in Mobiquitous 2014 Poster Session
- Editor of SPICE Update newsletter
- Editor of SPICE project activity summary
- Organizer of Workshop on terrestrial and space DTN
- Co-organizer of Delay- and Disruption- Tolerant Networks (DTNs) workshop
- Organizer of Workshop on DTN communications
- Organizer of SPICE project final event on Space Internetworking
- Outstanding Reviewer in Computer Communications (Elsevier)
- Outstanding Reviewer in Ad Hoc Networks (Elsevier)
- Reviewer in Pervasive and Mobile Computing (Elsevier), IEEE Transactions on Mobile Computing, IEEE Communications Letters, Wireless Communications and Mobile Computing (Wiley) and more

2015