INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2006; **19**:381–405
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/dac.789

Assessment of Internet voice transport with TCP

Panagiotis Papadimitriou*,† and Vassilis Tsaoussidis‡

Electrical and Computer Engineering Department, Demokritos University of Thrace, Xanthi 67100, Greece

SUMMARY

We investigate whether the current best-effort technology of Internet stands up to the high quality standards of real-time voice communication. More precisely, we assess VoIP quality within the context of transport protocol support and efficiency. Initially, we focus on TCP and UDP supportive role from the perspective of VoIP performance. Applying our metric for real-time performance, we reach the outcome that UDP is always not the protocol of choice for VoIP, since it occasionally exhibits inadequate performance. Beyond that, we evaluate a solution-framework based on TCP protocols, which favour real-time traffic. Based on our measurements, we assess the efficiency of the associated congestion control and congestion avoidance mechanisms in terms of VoIP performance. We also study the effect of packet size on protocol behaviour and VoIP quality. Furthermore, we investigate VoIP traffic friendliness, as well as potential tradeoffs between protocol performance and fairness. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: voice over IP; QoS; TCP; congestion control; performance evaluation

1. INTRODUCTION

Internet is evolving into a universal communications network, hosting several types of traffic including traditional data, voice and video. *Voice over IP* (*VoIP*) has emerged as an alternative to the traditional *public switched telephone network* (*PSTN*) and is steadily gaining popularity. Unlike traditional telephony, VoIP takes advantage of packet network properties, achieving more effective bandwidth utilization. Furthermore, VoIP exploits advanced voice compression techniques, which reduce the size of the transmitted stream. A fundamental advantage of VoIP is that it enables the creation of applications, which integrate voice with data. Consequently, the services provided by VoIP are not restricted to voice communication, but often include other media (e.g. video) and data applications, such as white boarding and application/file sharing.

In terms of traffic characteristics, voice streams have low data rates and exhibit low burstiness [1]. VoIP, as a high quality real-time voice communication, has stringent end-to-end delay and

---

*Correspondence to: P. Papadimitriou, Electrical and Computer Engineering Department, Demokritos University of Thrace, Xanthi 67100, Greece.
†E-mail: ppapadim@ee.duth.gr
‡E-mail: vtsaousi@ee.duth.gr

Received May 2005
Revised October 2005
Accepted October 2005

loss rate requirements. Delays above 150 ms are perceived by most users, while delays exceeding 300 ms usually render the conversation annoying. Inline with high latencies, delay variations often impact voice quality. More precisely, a significant proportion of packets delivered very late either cause conversational gaps or generate a confusing conversation. Therefore, variations of throughput and delay, along with reliability parameters, such as packet loss and packet errors usually degrade the performance of such applications. Even in the situation of excellent VoIP end-systems, the varying delays and loss characteristics of the Internet may still cause perceptible degradation of voice quality.

VoIP usually runs over *user datagram protocol* (*UDP*), due to the impression that UDP is more suitable for real-time applications. UDP is a fast, lightweight and free-transmitting protocol, which appears to meet the demanding *quality of service* (*QoS*) requirements of time-sensitive applications. However, UDP cannot guarantee reliability and certainly, is not able to deal with network delays either, since it lacks all basic mechanisms for error recovery and flow/congestion control. Despite the impression that congestion control is not mandatory for real-time applications, such as VoIP, relevant research work has revealed that congestion control does not necessarily degrade performance. For example, in Reference [25], we have shown that UDP occasionally achieves worse performance than *transmission control protocol* (*TCP*).

Although TCP dominates the Internet, it is apparently a less common option for real-time applications. The most fundamental design principle of TCP is reliability. The *AIMD* algorithm [2], incorporated in standard TCP versions, achieves stability and converges to fairness when the demand of competing flows exceeds the channel bandwidth. TCP is further enhanced with a series of mechanisms for congestion control, including *congestion avoidance*, *slow start*, *fast retransmit* and *fast recovery* [3, 4]. However, standard TCP exhibits limited efficiency in heterogeneous wired/wireless environments, since it is not able to detect the nature of the errors that result in packet losses and consequently, to determine the appropriate error-recovery strategy. More precisely, authors in Reference [5] outline three major shortfalls of TCP: (i) ineffective bandwidth utilization, (ii) unnecessary congestion-oriented responses to wireless link errors (e.g. fading channels) and operations (e.g. handoffs), and (iii) wasteful window adjustments over asymmetric, low-bandwidth reverse paths. Since the Internet provides services for various types of applications, flow contention inevitably appears in varying traffic, spatial and temporal patterns. VoIP flows competing with traditional data traffic (e.g. web, mail) is a common scenario. In the event of limited bandwidth availability, coexisting traffic often impacts the timely delivery of VoIP packets degrading application performance. Inversely, VoIP flows may also hurt congestion-sensitive traffic when they compete for scarce bandwidth. Although the mixture of voice with data traffic may call for complex admission policies, in this study, we assume that voice does not get any prioritization.

Contributions to efficient Internet voice applications meeting the toll quality standards set by traditional telephony providers usually focus on network support, adaptive playback buffers at the receiver or enhanced voice coding standards. Our objective is to explore the performance of current end-to-end solutions from the perspective of VoIP QoS. End-to-end schemes are promising, since significant performance gains can be achieved without any extensive support from intermediate nodes in the network. More precisely, we evaluate UDP versus TCP, investigating whether UDP exhibits superior performance, as implied. Beyond UDP and TCP supportive role, we study the particular methodology for congestion control by evaluating a series of TCP protocols, which deal with congestion from different perspectives (i.e. congestion control, congestion avoidance). Based on comprehensive results, we explore the potential of these protocols focusing on the associated impact on VoIP performance. We also study the effect of packet size on protocol behaviour and voice quality. Furthermore, we investigate VoIP traffic friendliness, as well as potential tradeoffs between protocol performance and fairness. Since voice quality is subjective and expressed primarily with respect to the individual user, we demonstrate additional results of VoIP performance versus diverse deadlines for packet inter-arrival times.

We organize the rest of the paper as follows. In the sequel, we refer to research studies and proposals dealing with various aspects of VoIP QoS, followed by an overview of the most remarkable end-to-end solutions towards the improvement of real-time performance. In Section 3, we analyse selected end-to-end congestion control mechanisms, and briefly discuss how they impact protocol behaviour in the context of VoIP performance. In Section 4, we describe the particularity of voice traffic in terms of characteristics and requirements, and present our evaluation methodology. In Section 5, we analyse the results of the experiments we performed, and in the last section, we highlight our conclusions.

2. RELATED WORK

VoIP is rapidly evolving and has eventually attracted the required attention of the research community. Numerous research studies have been presented, which span several aspects of VoIP. Relevant work includes [6], where VoIP services are evaluated focusing on the ability of the Internet to effectively support interactive voice communication. Authors in Reference [7] investigate performance issues associated with mixing voice and self-similar data traffic within the Internet. In Reference [8], the voice quality of a VoIP system is evaluated when the voice-data length and the network conditions change. Authors in Reference [9] investigate the behaviour of UDP and VoIP over 802.11 networks from the perspective of number of connections that a single access point can support. Furthermore, Reference [10] includes an overview of perceptual QoS assessment methodologies for VoIP systems.

Since VoIP exhibits the characteristics of real-time traffic, we also refer to research efforts, which deal with efficient QoS management of real-time applications focusing on protocol design. *Real-time transport protocol* (*RTP*) [11] is the standard for transmitting delay-sensitive traffic across packet-based networks. The protocol runs on top of UDP or TCP and provides end-to-end network transport functions, which are suitable for real-time applications over multicast or unicast networks. RTP, in a complementary role, uses the sequence information to determine whether the packets are arriving in order, and the time-stamping information to determine packet inter-arrival times. The data transport is augmented by *real-time control transport protocol* (*RTCP*), which allows the monitoring of data delivery. RTP and RTCP are designed to be independent of the underlying transport and network layers. The inclusion of IP, UDP/TCP and RTP headers in a voice-data packet inevitably increases transmission overhead. The transmission efficiency of VoIP can be enhanced by omitting redundancy in the reported header information. Relevant techniques [12] compress the header by not resending the information that does not change after call connection setup, and multiplex the voice data of two or more call channels in one IP packet with a sub-header identifying call channels.

Authors in References [13–16] proposed a family of TCP compatible protocols, called *TCP-friendly*. Generally, TCP-Friendliness is defined as the average throughput of non-TCP-transported applications over a large time scale does not exceed that of any conformant

TCP-transported ones under the same circumstance [17]. TCP-Friendly protocols achieve smooth window adjustments, while they manage to compete fairly with TCP flows. In order to achieve smoothness, this family of protocols use gentle backward adjustments upon congestion. However, this modification has a negative impact on the protocol responsiveness. *TCP-Friendly rate control* (*TFRC*) is a representative TCP-Friendly protocol, which adjusts its transmission rate in response to the level of congestion, as it is estimated based on the calculated loss rate. Unlike standard TCP, the instantaneous throughput of TFRC has a much lower variation over time and consequently only smooth adjustments are needed. Furthermore, multiple packet losses in the same *round trip time* (*RTT*) are considered as a single loss event by TFRC and hence, the protocol follows a more gentle congestion control strategy. TFRC eventually achieves the smoothing of the transmission gaps and therefore, is suitable for applications requiring a smooth sending rate. However, this smoothness has a negative impact, as the protocol becomes less responsive to bandwidth availability [18]. TFRC has another major constraint: it is designed for applications transmitting fixed-sized packets and consequently its congestion control is unsuitable for applications, which use packets with variable size. In order to overcome this inconvenience, a TFRC variant called *TFRC-packetsize* (*TFRC-PS*) has been alternatively proposed.

TCP-Westwood is a sender-side-only modification of *TCP-Reno* congestion control, which exploits end-to-end bandwidth estimation to properly set the values of slow-start threshold and congestion window after a congestion episode [19]. TCP-Westwood significantly improves fair sharing of high-speed networks capacity. The protocol performs an end-to-end estimate of the bandwidth available along a TCP connection to adaptively set the control windows after congestion [20]. Although TCP-Westwood does not incorporate any mechanisms to support error classification and the corresponding recovery tactics for wired/wireless networks, the proposed mechanism appears to be effective over asymmetric wireless links due to its efficient congestion control.

TCP-Real is a high-throughput transport protocol that incorporates congestion avoidance mechanism in order to minimize transmission-rate gaps. Therefore, this protocol is suited for real-time applications, since it enables better performance and reasonable playback timers. TCP-Real [21, 22] employs a receiver-oriented and measurement-based congestion control mechanism that promotes TCP to a reliable solution over heterogeneous networks and asymmetric paths. In TCP-Real, the receiver decides with better accuracy about the appropriate size of the congestion window. Slow start and timeout adjustments are present, but they are only used whenever congestion avoidance fails. However, rate and timeout adjustments are aborted whenever the receiving rate indicates sufficient availability of bandwidth [22].

Congestion avoidance mechanisms usually prevent congestion episodes, which damage the timely delivery of packets and consequently degrade real-time application performance. The objective of these mechanisms is to estimate the level of congestion before it takes place, and hence avoid it. In this context, network traffic is constantly monitored in an effort to anticipate and avoid congestion at common network bottlenecks. Hence, protocols, which incorporate such mechanisms are efficient for time-sensitive applications, such as VoIP. Congestion avoidance may be achieved through packet dropping (i.e. RED) or otherwise through bandwidth and delay estimation, which trigger transport-level adjustments prior to congestion. Alternatively, *ECN* is proposed [23], where packets are marked rather than dropped when congestion is about to happen. A well-designed, congestion avoidance mechanism is *TCP Vegas* [24, 5].

3. APPROACHES TO END-TO-END CONGESTION CONTROL

In the sequel, we perform an analysis of some representative end-to-end congestion control mechanisms, which are highly advisable for preserving the fundamental QoS guarantees for time-sensitive traffic, including VoIP.

3.1. TCP-Friendly congestion control

The differences between standard TCP and TCP-Friendly congestion control lie mainly in the specific values of additive increase rate α and multiplicative decrease β; their similarities lie in their *AIMD* based congestion control (a characteristic that enables us to include them both in the family of TCP (α, β) protocols). Standard TCP is therefore viewed as a specific case of TCP (α, β) with $\alpha = 1, \beta = 0.5$. On the other hand, TCP-Friendly protocols are designed to satisfy the requirements of time-sensitive applications. However, they may exhibit further weaknesses when bandwidth becomes available rapidly. Apparently, the tradeoff between responsiveness and smoothness can be controlled to favour some applications, but it will cause some other damages. The choice of parameters α and β has a direct impact on the '*responsiveness*' of the protocols to conditions of increasing contention or bandwidth availability.

Several studies are focused on the quantitative modelling of the TCP long-term throughput, as a function of the measured packet loss rate (p), RTT and initial transmission timeout value (T_0). According to Reference [16], TCP (α, β) throughput can be modelled as

$$T_{\alpha,\beta}(p, \mathrm{RTT}, T_0, b) = \frac{1}{\mathrm{RTT}\sqrt{\dfrac{2b(1-\beta)}{\alpha(1+\beta)}p} + T_0 \min\left(1, 3\sqrt{\dfrac{(1-\beta^2)b}{2\alpha}p}\right)p(1+32p^2)} \tag{1}$$

where b is the number of packets acknowledged by each *acknowledgement* (*ACK*) and α, β are the congestion control parameters. Observations of the window dynamics and event losses are frequently assumed within a time period of a *congestion epoch*. A congestion epoch is defined in Reference [22] as the time period that reflects the '*uninterrupted growing lifetime of a window*'. More precisely, a congestion epoch begins with βW packets, increased by α packets per RTT and reaching a congestion window of W packets, when a packet is dropped. The congestion window is then decreased to βW. Hence, a congestion epoch involves:

$$n = (\beta/\alpha W + 1)\mathrm{RTTs} \tag{2}$$

TCP-Friendly (α, β) protocols approximate the throughput of standard TCP ($\alpha = 1$, $\beta = 0.5$), which means that Equation (3), which is derived from (1) [16], provides a rough guide to achieve friendliness:

$$T_{\alpha,\beta}(p, \mathrm{RTT}, T_0, b) = T_{1,\,0.5}(p, \mathrm{RTT}, T_0, b) \tag{3}$$

However, since the network or application conditions may change rapidly, friendliness might not be attained. More precisely, based on (2), we conclude that (3) can be achieved later than expected, since multiple drops extend further the time of convergence. Furthermore, according to (2), the time period required for (3) to hold is in reverse proportion to the number of flows within a fixed bandwidth channel; the smaller the number, the larger the window.

TCP-Real approximates a receiver-oriented approach beyond the balancing trade of the parameters of additive increase and multiplicative decrease. The protocol introduces another parameter, namely γ, which determines the window adjustments during congestion avoidance. More precisely, the receiver measures the data-receiving rate and attaches the result to its ACKs, directing the transmission rate of the sender. When new data is acknowledged and the congestion window is adjusted, the current data-receiving rate is compared against the previous one. If there is no receiving rate decrease, the congestion window is increased by 1 *Maximum segment size* (*MSS*) every RTT ($\alpha = 1$). If the magnitude of the decrease is small, the congestion window remains temporarily unaffected; otherwise, the sender reduces the congestion window multiplicatively by γ. In Reference [22] a default value of $\gamma = \frac{1}{8}$ is suggested. However, this parameter can be adaptive to the detected conditions. Generally, TCP-Real can be viewed as a TCP (α, β, γ) protocol where γ captures the protocol's behaviour prior to congestion, when congestion boosts up [18].

3.2. TCP-Westwood estimation scheme

TCP-Westwood incorporates a recovery mechanism which avoids the blind halving of the sending rate of TCP-Reno after packet losses and enables TCP-Westwood to achieve a high link-utilization in the presence of wireless errors. The specific mechanism considers the sequence of bandwidth samples sample_BWE[n] obtained using the ACKs arrivals and evaluates a smoothed value, BWE[n], by low-pass filtering the sequence of samples, as described by the following pseudocode [20]:

Algorithm 1
TCP-Westwood

```
if an ACK is received
   sample_BWE[n] = (acked*pkt_size*8)/(now−last_ACK_time);
   BWE[n] = (1−beta)*(sample_BWE[n] + sample_BWE[n−1])/2 + beta*BWE[n−1];
end if
```

where acked is the number of segments acknowledged by the last ACK; pkt_size is the segment size in bytes; now is the current time; last_ACK_time is the time the previous ACK was received; beta is the pole used for the filtering (a value of 19/21 is suggested). However, the outcome of this linear filtering technique is usually a biased estimate, since bandwidth-samples filtering is based on a fixed pole filter. According to simulation tests performed in Reference [25], TCP-Westwood filtering algorithm tends to overestimate the available bandwidth.

3.3. TCP-Vegas congestion avoidance mechanism

In TCP-Vegas, every RTT, the sender calculates the throughput rate, which subsequently is compared to an expected rate. Depending on the outcome of this comparison, the transmission rate of the sender is adjusted accordingly. More precisely, let RTT_{min} denote the minimum RTT measured by the TCP source. Whenever an ACK is received, TCP-Vegas computes the quantity:

$$\text{diff} = (\text{expected\_Rate} - \text{actual\_Rate})\text{RTT}_{\min}$$

The size of congestion window (cwnd) is then increased by 1 if diff < 1, decreased by 1 if diff > 3 and left unchanged if $1 \leqslant \text{diff} \leqslant 3$. Based on Reference [26] admissions, Vegas achieves better transmission rates than TCP-Reno and TCP-Tahoe. Although the protocol is compliant to the rules of fairness (AIMD algorithm), according to Reference [24], Vegas cannot guarantee fairness, since competing connections may converge to different cwnd parameter values. Another shortfall of this protocol is that it is not able to distinguish the nature of error. However, this constraint is common to the most TCP versions.

4. EVALUATION METHODOLOGY

4.1. VoIP characteristics

Although voice transfer originally calls for strict requirements on end-to-end delay, jitter and packet loss, the presence of sophisticated encoding algorithms allows for more relaxed bandwidth administration (usually up to 64 kbps). A widely deployed coding standard that achieves good speech quality is ITU-T G.711 [27], which uses *pulse code modulation* (*PCM*) to generate 8-bit samples per 125 μs, achieving a rate of 64 kbps. Table I summarizes the most widely-used speech coding standards along with its associated characteristics (we refer to *mean opinion score* (*MOS*) in a following subsection). In the sequel, we discuss the major QoS parameters that affect VoIP quality.

4.1.1. End-to-end delay. End-to-end delay consists of the delay incurred by the voice signal from the instant it is produced by the speaker until it is heard by the listener at the destination. Initially, the analog signal is encoded, followed by the packetization phase, incurring an encoding (D_{enc}) and packetization (D_{pack}) delay, respectively. Voice packets are then transmitted on the network. Network delay is expressed by the summation of propagation (P_h), transmission (T_h), and the variable queuing and processing delays (Q_h) for each hop h in the path from the source to the destination. If we include a playback delay (D_{play}) and we ignore the processing delays at both sender and receiver, the end-to-end delay D for a packet is expressed, as

$$D = D_{\text{enc}} + D_{\text{pack}} + \sum_{h \in \text{Path}} (T_h + Q_h + P_h) + D_{\text{play}} \tag{4}$$

We observe that for a certain codec and voice connection, the only random component in (4) is queuing delay. Therefore, end-to-end latency is mostly affected by queuing delays.

Table I. Speech codecs.

Codec	Rate (kbps)	MOS	Complexity
ITU-T G.711	64	Above 4	—
ITU-T G.722	48–64	3.8	Low
ITU-T G.726	32	3.8	Low
ITU-T G.728	16	3.6	Low
ITU-T G.729 (A/B)	8	3.9	Medium
ETSI GSM 06.10	13	3.5	Low
ETSI GSM 06.20	5.6	3.5	High
ETSI GSM 06.60	12.2	Above 4	High
ETSI GSM 06.70	4.8–12.2	Above 4	High
Nokia AMR-WB	12.5–24.8	Above 4	Very high

Table II. Typical delay guidelines for VoIP.

Delay	Effect in perceived quality
Less than 150 ms	Delay is not noticeable
150–250 ms	Acceptable quality with slight speech impairments
Over 250–300 ms	Unacceptable delay, conversation is inefficient

Table III. Typical jitter guidelines for VoIP.

Delay variation	Effect in perceived quality
Less than 40 ms	Jitter is not noticeable
40–75 ms	Acceptable quality with minor impairments
Over 75 ms	Unacceptable quality, too much jitter

The typical delay guidelines for VoIP are shown in Table II. End-to-end delays exceeding 300 ms [1] affect the timely delivery of voice data and have a negative impact on conversation quality.

4.1.2. Delay variation. Delay variation is usually caused by the variable queuing and processing delays on routers during periods of increased traffic and occasionally by routing changes. Delay variation is responsible for the phenomenon called network jitter, which has unpleasant effects in interactive communication, as packets often reach the receiver later than required. Practically, delayed packets are either discarded and considered lost, or at the worst they obstruct the proper reconstruction of oncoming packets. Table III indicates the perceptual effect of jitter on VoIP. Generally, interactive communication does not afford delay variations above 75 ms. Excessive jitter usually generates a confusing conversation, which is intolerable by most users.

4.1.3. Packet loss. Packet loss is a major impairment factor, since it causes a perceptible degradation in voice quality. Packet drops are typically the result of excessive congestion in the network. However, in a heterogeneous wired/wireless environment, apart from congestion, hand-offs and fading channels may result in packet loss. Standard TCP is unable to successfully detect the nature of the errors in such a network environment. As a result, TCP is not able to determine the appropriate error-recovery strategy with a negative impact on VoIP performance. Several VoIP codecs use *forward error correction* (*FEC*) or *packet loss concealment* (*PLC*) [28] in order to ameliorate the impact of packet losses on voice quality. Despite the presence of such mechanisms, VoIP still suffers when the transport protocol is unable to restrict packet drops. In Reference [29], Paxson investigated the loss rate for a number of Internet paths and found that it ranged from 0.6 to 5.2%. A recent study [30] confirmed Paxson's earlier results, but also showed that the average loss rate for the measurements was 0.42%. However, due to an average *burstiness* of 72%, packet loss can be occasionally much higher and therefore, efficient transport services are required, so that VoIP traffic can compete with the existing PSTN.

4.2. Scenarios and parameters

The evaluation plan was implemented on the *NS-2* network simulator. The experiments were conducted based on realistic VoIP scenarios. More precisely, we used a network topology (Figure 1), which includes multiple bottlenecks, cross traffic and wireless links. The router R1 is the bottleneck for VoIP traffic (flows between source nodes and sink nodes), while the router R2 is another bottleneck for competing VoIP and cross traffic (flows between peripheral source and peripheral sink nodes). Cross traffic includes various FTP flows over TCP-Reno: a common application running over a widely used transport protocol. We used two configurations for cross traffic: light cross traffic (20 flows) and heavy cross traffic (60 flows). The number of source and sink nodes is always equal. We also attached an equal number of peripheral source and sink nodes. In all experiments, we used droptail routers.

During a conversation, speakers alternate between activity and idle periods. Taking into consideration the ON and OFF periods [31], as well as the heavy-tailed characteristics and self-similarity of VoIP traffic [32], we used the *Pareto* distribution for modelling the call holding times. We configured *Pareto* with a mean rate of 64 kbps and the shape parameter was set to 1.5. According to Reference [31], we distributed the ON and OFF periods with means of 1.0 and 1.35 s, respectively. Despite the probabilistic flows entering times and the randomly generated wireless errors, the deterministic nature of *NS-2* provided insignificant variation in the results, leading to almost perfect statistical confidence. Along these lines, we simulated VoIP streams of 64 kbps (inline with the widely-used ITU-T G.711 coding standard) and we set the packet size at 160 bytes (20 ms of speech). However, in the second scenario, we adjusted the packet size at 160, 300 and 500 bytes (20, 40 and 65 ms of speech, respectively) successively, in order to investigate how packet size affects VoIP performance.

NS-2 error models were inserted into the access links to the main sink nodes. Error models were configured on both (forward and reverse) directions of the link traffic. We used the *Bernoulli* model in order to simulate the link-level errors with packet error rate (PER) adjusted at 0.01. The fourth scenario is the only exception, where PER varies from experiment to experiment. The simulation time was fixed at 60 s, an appropriate time-period for all the protocols to demonstrate their potential. We performed our experiments running VoIP over UDP and TCP versions Vegas, Westwood and Real.

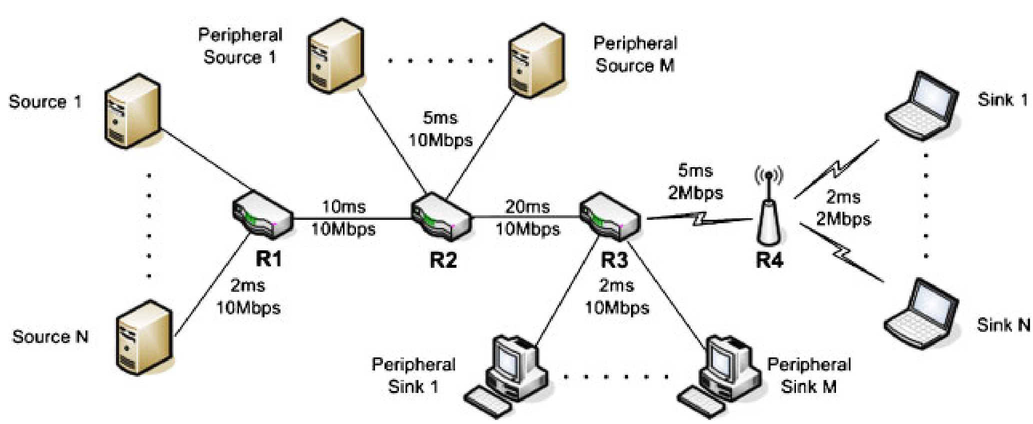


Figure 1. Simulation topology.

4.3. Measuring performance

Since the simulated topology includes cross traffic, our performance metrics are applied separately to main VoIP traffic and cross FTP traffic. Exploiting this issue, we are able to perform a thorough study of the implications between interfering traffic. Along these lines, we refer to the performance metrics supported by our simulation model. System goodput is used to measure the overall system efficiency in bandwidth utilization. The system goodput is defined as

$$\text{Goodput} = \text{Original\_Data}/\text{Connection\_time}$$

where Original_Data is the number of bytes delivered to the high-level protocol at the receiver (i.e. excluding retransmitted packets and overhead) and Connection_time is the amount of time required for the data delivery. Fairness is measured by the Fairness index, derived from the formula given in Reference [2], and defined as

$$\text{Fairness index} = \frac{(\sum_{i=0}^{n} \text{Throughput}_i)^2}{n(\sum_{i=0}^{n} \text{Throughput}_i^2)}$$

where Throughput_i is the throughput of the ith flow and n is the total number of flows.

Evaluating the performance of real-time traffic based on traditional performance metrics (i.e. goodput) may produce misleading results, since such metrics do not account for variable delays, which impact the efficiency of time-sensitive applications. In voice communications, several quality evaluation methods have been proposed [10]. MOS provides a numerical measure of human speech quality at the receiving end. MOS virtually indicates the speech quality perceived by the listener on a scale from 1 (worst) to 5 (best). The most popular objective measurements include *perceptual evaluation of speech quality* (*PESQ*) [33] and *E-model* [34]. PESQ is an objective measurement tool that predicts the results of subjective listening tests on telephony systems. PESQ score is estimated by processing both the input reference and the degraded output speech signal. PESQ takes into account coding distortions, errors, packet loss, delays and variable delays, and filtering in analogue network components. The resulting quality score is analogous to the subjective MOS and ranks on a scale from -0.5 (worst) to 4.5 (best). E-model is a computational model that uses transmission parameters to predict the subjective quality of packetized voice. It has several input parameters that represent the terminal, network, and environmental quality factors. E-model assumes that the perceived effect of impairments, such as echo, delay or distortion, is additive. Based on this principle, E-model outputs a single rating (R) on a scale from 0 to 100, which can be further translated into *MOS*. Table IV includes a classification of *MOS* and R ratings, along with the corresponding user perception.

Table IV. Voice quality classification.

User perception	MOS score	R
Very satisfied	4.3–5.0	90–100
Satisfied	4.0–4.3	80–90
Some users satisfied	3.6–4.0	70–80
Most users dissatisfied	3.1–3.6	60–70
Almost all users dissatisfied	2.6–3.1	50–60
Not recommended	1.0–2.6	0–50

In order to quantify the impact of individual impairments on voice quality, we exploited the *real-time performance* metric, which we initially proposed in Reference [35] and revised in Reference [36]. The metric achieves an efficient performance evaluation of real-time traffic and can be easily configured according to individual application requirements, such as VoIP. More precisely, *real-time performance index* is defined as the ratio of the number of *timely received packets* over the total number of packets sent by the application:

$$\text{Real-time performance index} = \frac{\#\ \text{timely received packets}}{\#\ \text{sent packets}} \leqslant 1$$

Real-time performance monitors packet inter-arrival times and distinguishes packets arriving on time from excessively delayed packets (according to a configurable inter-arrival threshold). The proportion of the number of delayed packets is denoted by *delayed packets rate*. Since VoIP is sensitive to packet losses, we additionally define *packet loss rate* as the ratio of the number of lost packets over the number of packets sent by the application. In accordance with VoIP delay requirements, we adjusted the inter-arrival threshold at 300 ms. Consequently, arriving packets exceeding this deadline are marked as *delayed packets*, since they do not effectively participate in the reconstruction of voice data. Since our experiments were performed on several VoIP flows, we present the average of the real-time performance of each VoIP flow.

5. RESULTS AND DISCUSSION

In the sequel, we demonstrate and analyse results from the experiments we performed based on five distinct scenarios. The basic parameters of each simulation scenario were described in the previous section.

5.1. TCP vs UDP

In the first scenario, we performed a series of experiments in order to evaluate the performance of VoIP over TCP and UDP. We measured: *goodput*, *fairness index* and *real-time performance*. In addition, we selected statistics from delayed and lost packets, since all are influencing factors, which impact VoIP performance. Our experiments were performed, as described in the previous section. However, in this scenario, we avoided any interference from other (non-VoIP) flows in order to have a thorough exploration of all VoIP performance issues. Therefore, we did not include any cross traffic. We present some conclusive results from TCP-Vegas, TCP-Westwood, TCP-Real and UDP (Figures 2–6).

Inline with our expectations, UDP achieves slightly higher goodput performance (Figure 2) in comparison with the TCP protocols. More precisely, UDP transmits steadily at application rate, while TCP purposely backs off in periods of congestion. In the context of TCP, the demand for smoothness rather than responsiveness determines the choice of α, β and consequently designates the protocol's behaviour and strategy. However, rendering smoothness as a dominant factor may sacrifice application throughput under awkward conditions (e.g. congestion). More precisely, the reported goodput results (Figure 2) outline the deficiency of TCP-Westwood, especially with increased contention. The high packet loss rate (Figure 6) of TCP-Westwood indicates that the protocol is unable to effectively recover from excessive congestion, due to its smooth window adjustments. Apart from the conservative choice of

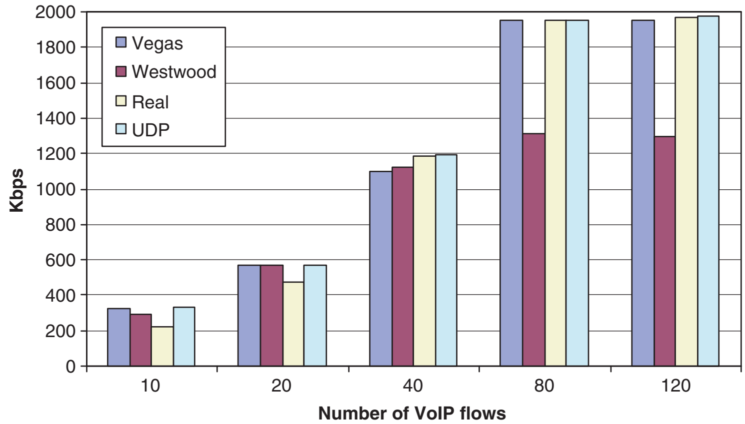


Figure 2. System goodput.

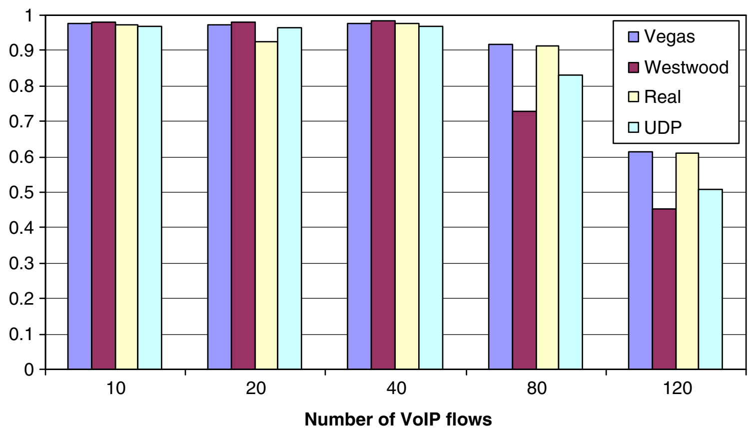


Figure 3. Average real-time performance.

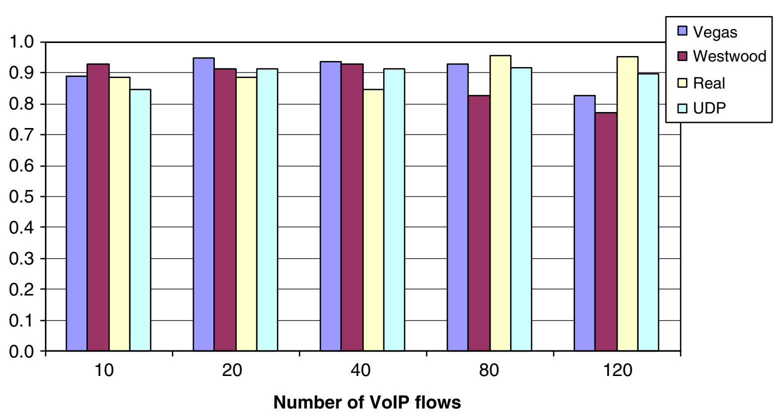


Figure 4. Fairness index.

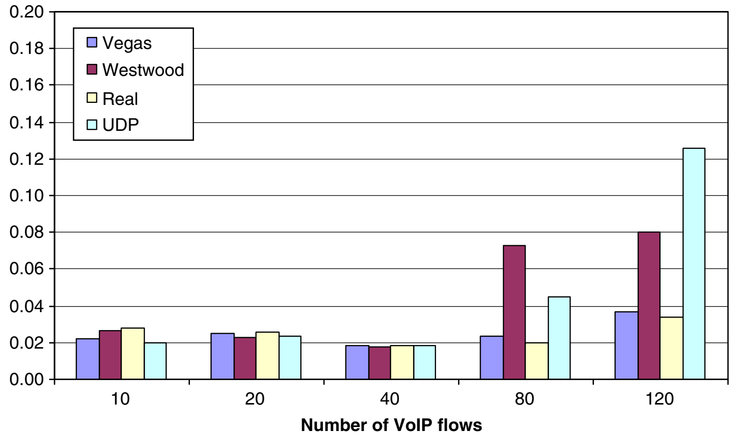


Figure 5. Delayed packets rate.

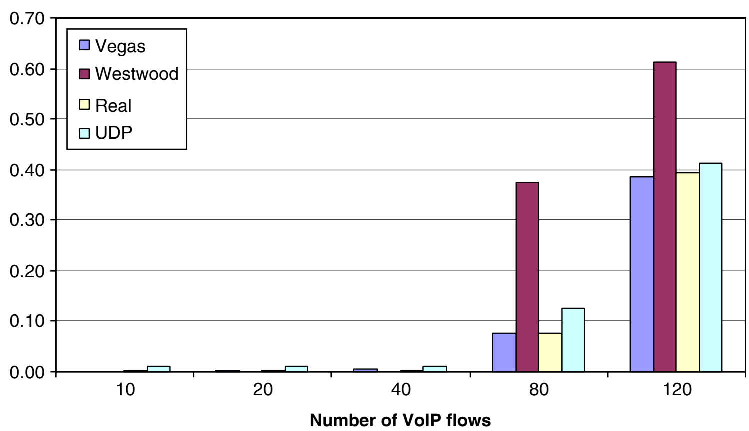


Figure 6. Packet loss rate.

parameters (i.e. α, β), the protocol is not assisted by its inefficient bandwidth-estimation algorithm. Furthermore, TCP-Westwood, which does not incorporate any mechanisms for error classification and the corresponding recovery tactics, fails to deliver high performance in wired/wireless scenarios; its performance is further diminished due to the unnecessary congestion-oriented responses to wireless link errors. On the contrary, Vegas and especially Real, which appears able to detect the nature of the error, respond appropriately to wireless errors and achieve remarkable goodput rates (Figure 2). The receiver-oriented congestion control of TCP-Real is able to classify packet drops: whenever the receiver observes data delivery with low jitter and high receiving-rate, missing packet(s) indicate a random (and usually wireless) error.

The high goodput rates achieved by UDP are not reflected in the real-time performance results. UDP does not yield satisfactory performance, especially during increased contention.

A comparative view of the results in Figures 5 and 6 justifies this unexpected result. Although, UDP does not exhibit excessive packet loss (relatively to TCP), Figure 5 indicates that a significant proportion of voice-data (especially for 80 and 120 flows) reaches the recipient exceeding the delay requirements of VoIP. The significant delay in voice-data delivery is primarily due to the increased queuing delays in our multi-hop network topology. Based on Equation (4), we showed that queuing delay is a critical factor for the end-to-end latency of VoIP. The aggressive UDP results in rapidly growing queues and essentially in bottleneck buffer overflows. Since the protocol never reduces its transmission rate, the buffers remain overflowed and a considerable proportion of the packets that are not dropped reach the recipient later than required. Therefore, UDP results in long and variable delays, which usually cause conversational gaps and generally degrade the quality of online communication.

Inline with UDP, TCP-Westwood exhibits inadequate real-time performance, due to low goodput rates and increased number of delayed packets. On the other hand, TCP-Real achieves superior performance (Figures 2 and 3). This appears to be the result of its receiver-oriented congestion control and its advanced error detection. More precisely, the protocol exploits the additional parameter γ: the desired smoothness is counterbalanced with responsiveness, which is critical during congestion episodes. Apart from goodput and real-time performance, TCP-Real excels in bandwidths sharing, even for a large number of flows (Figure 4). TCP-Vegas takes advantage of its congestion avoidance mechanism providing efficient transport services for VoIP. However, the protocol occasionally trades fairness for a remarkable performance (Figure 4: 120 flows); its mechanism cannot handle bandwidths sharing efficiently, since competing connections may converge to different cwnd parameter values, as we reported in our analysis.

5.2. The impact of voice-packet size on VoIP performance

We repeated the experiments of the previous scenario using packets of 160, 300 and 500 bytes successively (20, 40 and 65 ms of speech, respectively). However, apart from VoIP flows, we included light FTP cross traffic (20 flows) over TCP-Reno. Inline with the first scenario, we evaluated the transport services of TCP-Vegas, TCP-Westwood, TCP-Real and UDP. Our objective is to investigate the impact of different packet sizes on VoIP performance. We hereby present the associated goodput and real-time performance results for each protocol separately (Figures 7–14).

As expected, nearly all protocols achieve higher goodput rates for longer voice-data segments. This phenomenon is more intense in TCP-Westwood (Figure 9), due to its congestion control mechanism. Apparently, the bandwidth estimation mechanism of the protocol is more effective with larger segments, since the number of bandwidth samples is decreased and filtering is more efficient. Large packet sizes occasionally improve VoIP performance (Figure 10), although the benefits are not so noticeable. Inversely, TCP-Vegas demonstrates a different behaviour, since it does not exhibit any performance gains for packet sizes above 160 bytes (Figure 7). Relatively small segments favour frequent throughput estimations and thus, Vegas adjusts its transmission rate more accurately avoiding packet drops. In terms of VoIP performance, long voice-data segments degrade voice quality (Figure 8). This observation is also addressed to TCP-Real (Figure 12) and UDP (Figure 14), despite the slight gains in goodput for both protocols (Figures 11 and 13).

The resulting VoIP quality degradation for longer voice-data segments is associated with the impact of packet errors on large voice segments. More precisely, concerning the wireless errors

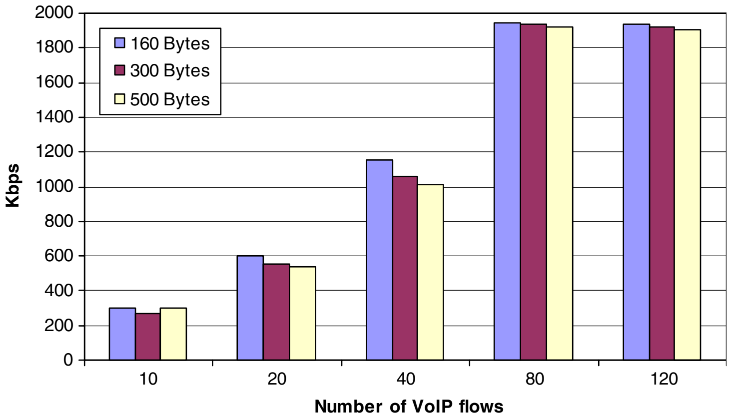


Figure 7. Goodput of VoIP flows (Vegas).

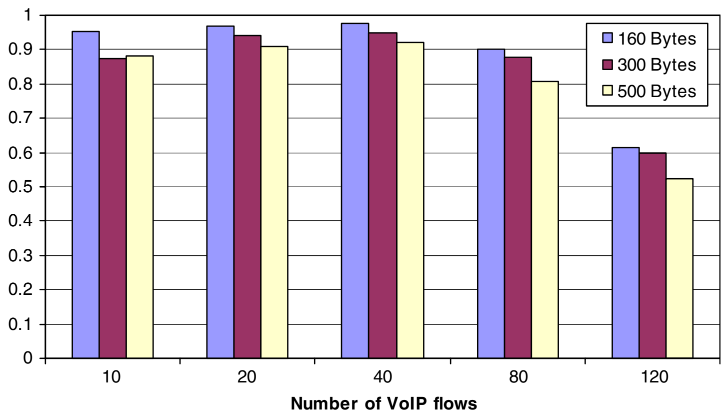


Figure 8. Average real-time performance (Vegas).

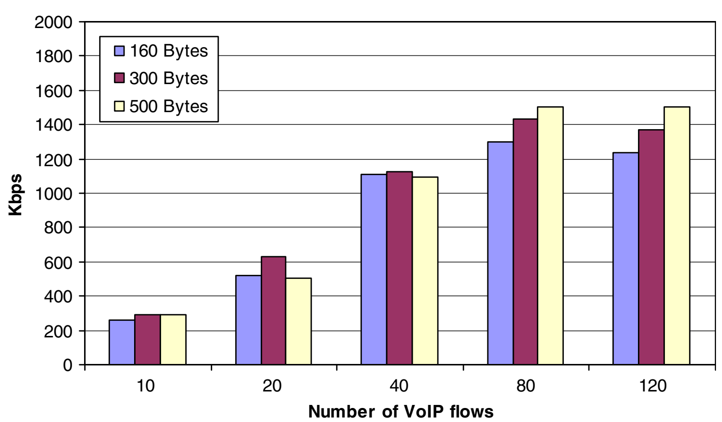


Figure 9. Goodput of VoIP flows (Westwood).

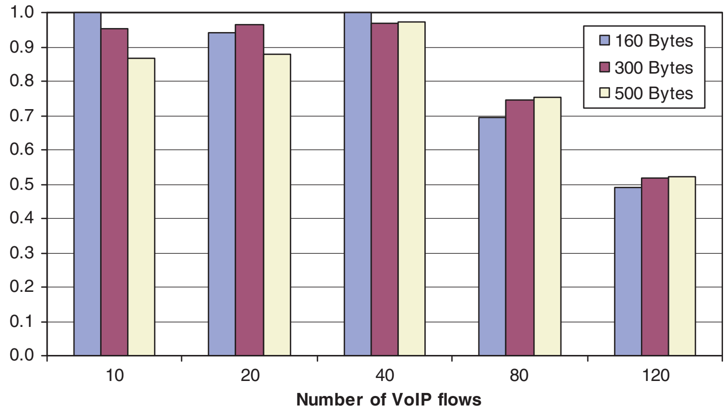


Figure 10. Average real-time performance (Westwood).

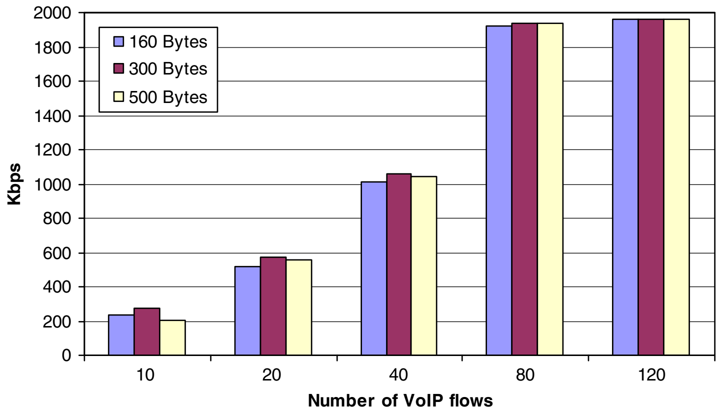


Figure 11. Goodput of VoIP flows (Real).

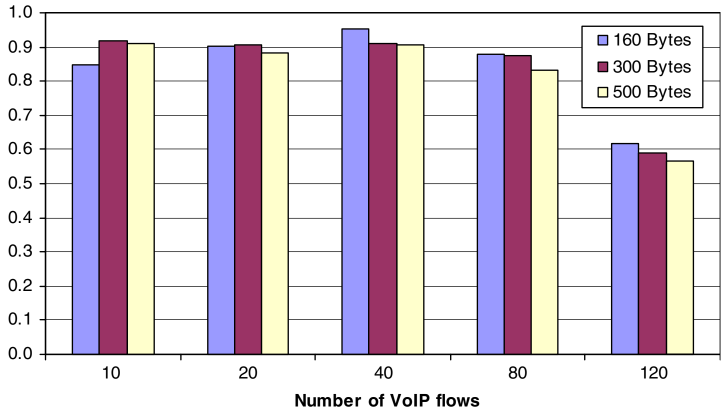


Figure 12. Average real-time performance (Real).

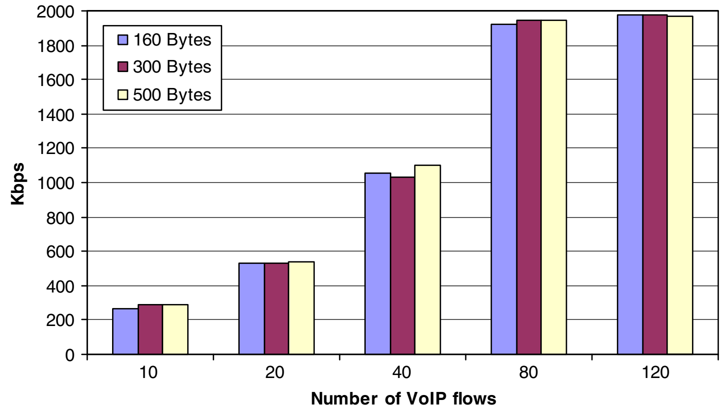


Figure 13. Goodput of VoIP flows (UDP).

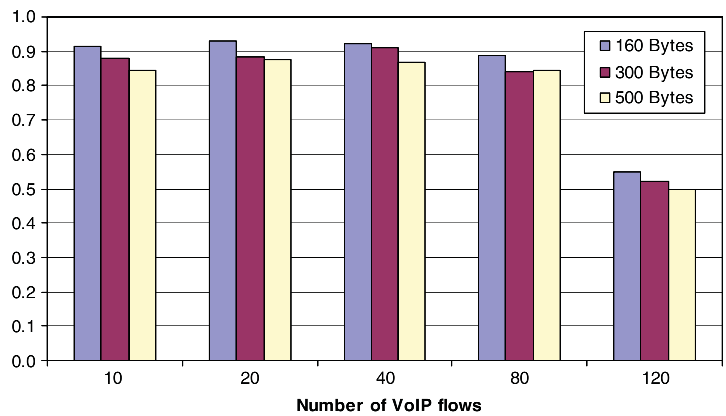


Figure 14. Average real-time performance (UDP).

introduced in our scenario, we observe that large packets exhibit an increased sensitivity to packet errors. Similar findings were reported in Reference [8]. In a following scenario, we introduce diverse packet error rates and we investigate the impact on VoIP performance.

5.3. The impact of VoIP on corporate FTP traffic

Departing from a comparative analysis of protocol support for VoIP, we investigate the impact of VoIP on corporate FTP traffic. Our objective is to explore whether there is a tradeoff between protocol performance and efficiency of interfering traffic. Along these lines, we simulated various VoIP flows competing with cross FTP traffic over TCP-Reno. We repeated the experiment for VoIP over Vegas, Westwood, Real and UDP and we measured the aggregated goodput of all FTP flows, correspondingly. We present goodput results of both light and heavy cross FTP traffic (Figures 15 and 16).

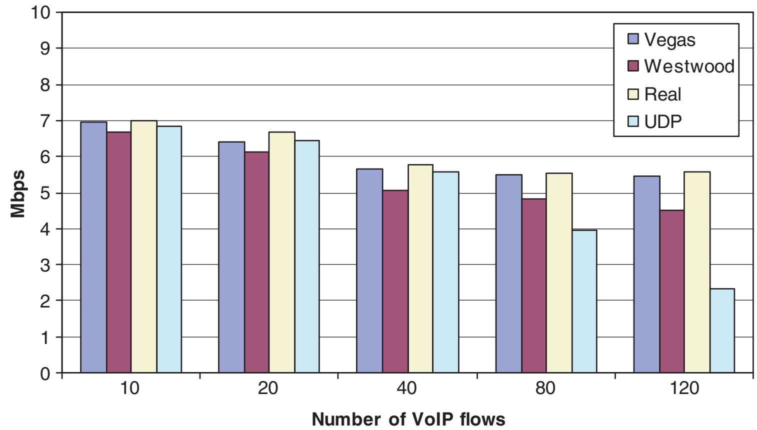


Figure 15. Goodput of light cross FTP traffic.

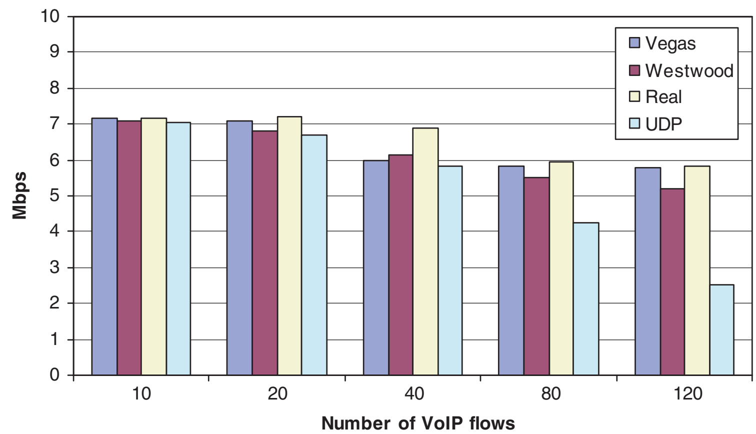


Figure 16. Goodput of heavy cross FTP traffic.

UDP has a negative impact on the interfering FTP flows either with light or heavy cross traffic. This impact becomes more significant when contention increases and UDP allocates excessive network resources ‘‘stealing’’ bandwidth from the FTP flows. More precisely, UDP’s free-rate transmitting policy forces the interfering TCP flows to sharp window adjustments. Therefore, all applications sharing the same network resources with UDP face an unfavourable situation. On the contrary, TCP is designed to converge to fairness, and indeed, all TCP protocols included are relatively fair to the FTP flows. TCP-Real achieves a remarkable performance, since it mostly favours other applications sharing common network resources.

5.4. *VoIP performance vs packet errors*

In this scenario, we performed the experiments using variable packet error rate adjustments (PER: 0.01–0.05). We simulated 80 and 120 VoIP flows competing with light FTP cross traffic (20 flows) over TCP-Reno. We also carried out the same experiment without packet errors and used it as a reference. Our purpose is to demonstrate the impact of diverse packet error rates on goodput (Figures 17 and 19) and mostly on VoIP performance (Figures 18 and 20).

The results from this scenario lead to the overall conclusion that the performance of TCP is inevitably diminished in the situation of increasing wireless errors. TCP responds to all losses by invoking congestion control responses, resulting in degraded end-to-end performance. As the error rate increases, system goodput (Figures 17 and 19) naturally decreases and, consequently, voice quality is degraded (Figures 18 and 20). Goodput and real-time performance degradation are more intense in the situation of 80 flows, since the relatively high sending rate (due to lower contention) is diminished by the reduction of the congestion window. TCP-Vegas and especially

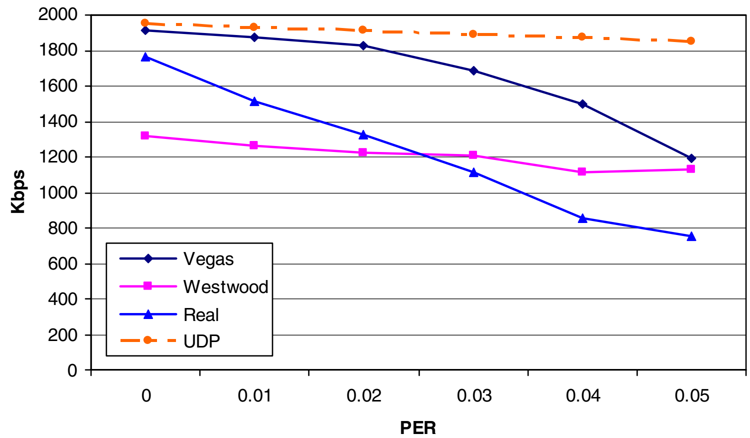


Figure 17. Goodput (80 VoIP flows).

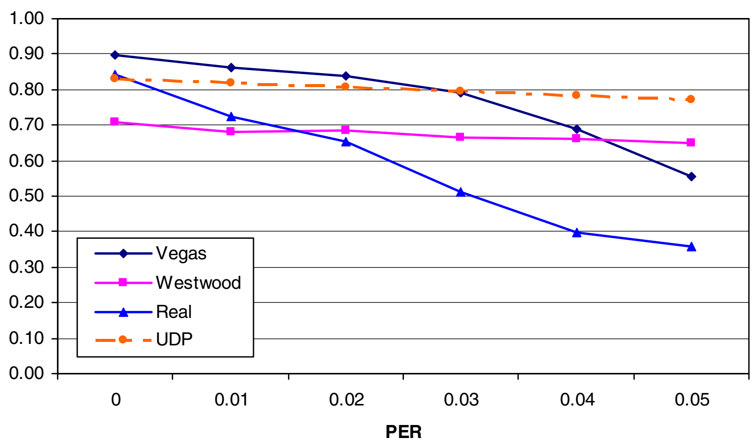


Figure 18. Average real-time performance (80 VoIP flows).

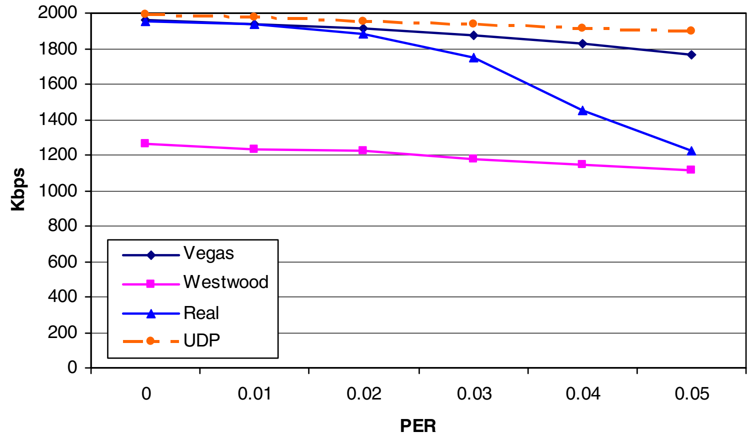


Figure 19. Goodput (120 VoIP flows).

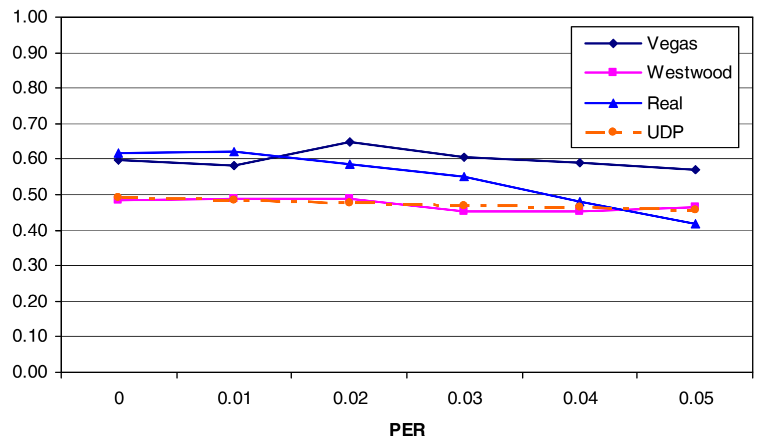


Figure 20. Average real-time performance (120 VoIP flows).

TCP-Westwood are less responsive to the diverse packet error rates, although they are not able to detect the nature of a packet error. On the contrary, TCP-Real surprisingly demonstrates limited efficiency at relatively high packet error rates (PER: 0.03–0.05) despite the incorporated error classification mechanism. Apparently, the mechanism operates inefficiently for increased wireless errors. However, the performance of TCP-Real is noticeably improved when contention is increased (Figure 20). Generally, a novel strategy is needed for TCP-Real, where detection and recovery should additionally account for specific characteristics of the error pattern, such as frequency and duration. On the other hand, UDP exhibits minor implications in the event of increasing wireless errors, as the protocol transmits at a steady rate and does not account for any type of error.

5.5. *VoIP performance vs a range of inter-arrival thresholds*

In the last scenario, we performed a series of experiments using variable adjustments of the inter-arrival time threshold (100–400 ms). Our primary goal is to investigate protocol sensitivity under these conditions. Furthermore, we show that VoIP performance is tightly coupled with the individual perception of quality. Along these lines, we simulated 80 and 120 VoIP flows competing with light cross FTP traffic (20 flows). Diverse threshold adjustments affect delayed packets exclusively and, consequently real-time performance, so we present only the corresponding results (Figures 21–24).

TCP-Vegas and TCP-Real are most sensitive to the adjustments of the inter-arrival time threshold. The configuration of threshold at 200 ms and above leads to respectable real-time performance gains for these protocols (Figures 21 and 23). Generally, all TCP

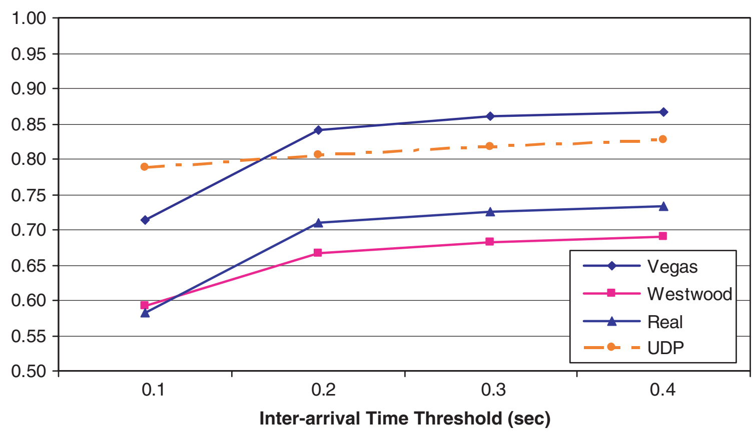


Figure 21. Average real-time performance (80 VoIP flows).

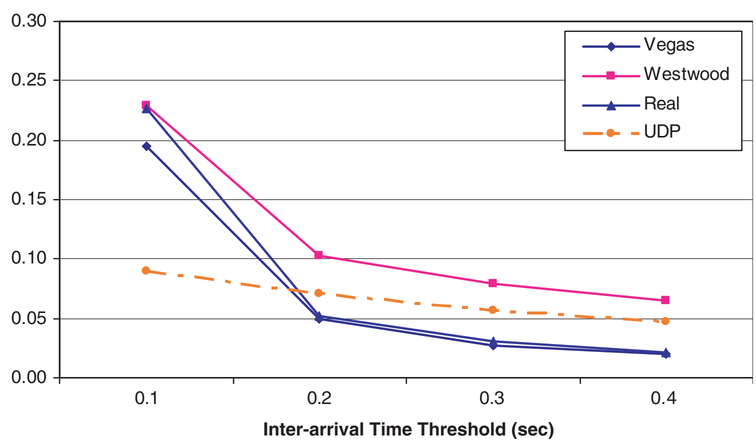


Figure 22. Delayed packets rate (80 VoIP flows).

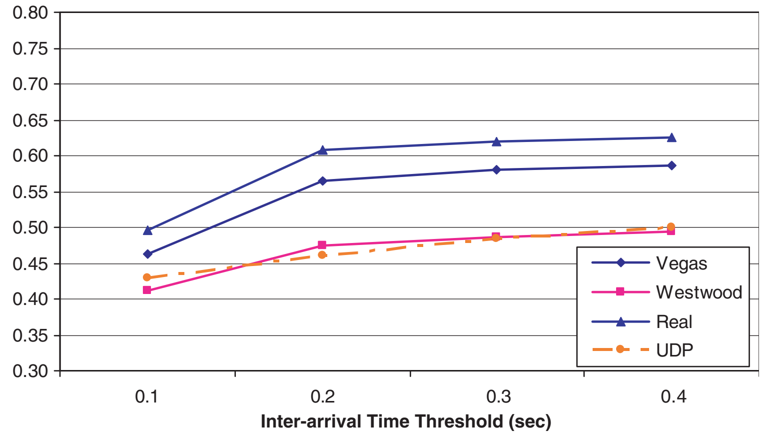


Figure 23. Average real-time performance (120 VoIP flows).

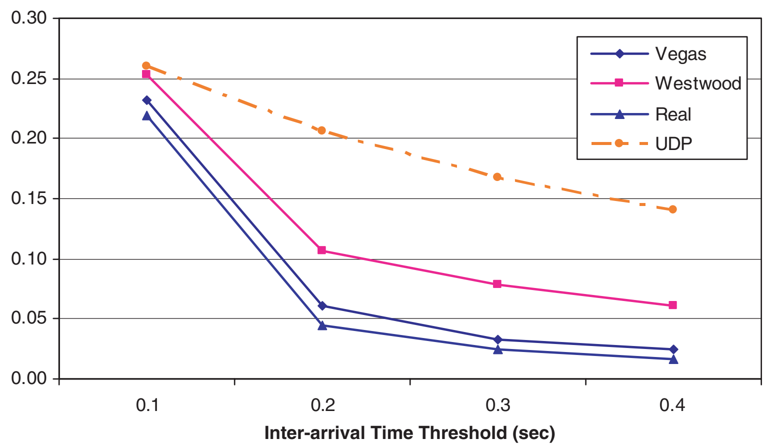


Figure 24. Delayed packets rate (120 VoIP flows).

protocols exhibit a slight inefficiency for threshold adjustments below 200 ms, which is reflected in the delayed packets results (Figures 22 and 24). In the situation of 80 VoIP flows (Figures 21 and 22), a user who is not annoyed by the conversation impairments caused by the adjustment at 200 ms will be generally satisfied by the overall quality. On the contrary, another user with stringent QoS requirements (e.g. threshold at 100 ms) will perceive only acceptable quality. The difference in perceived quality is significant, if VoIP runs over TCP-Vegas or TCP-Real. UDP generally exhibits less sensitivity to such adjustments in comparison with the TCP protocols. Hence, VoIP quality over UDP is less subjective to user perception of QoS parameters.

6. CONCLUSIONS AND FUTURE WORK

We explored the supportive role of selected transport protocols in terms of VoIP performance. Our research efforts were motivated from the (occasionally false) impression that UDP is the most prominent solution. However, our experimental results demonstrate the limited efficiency of UDP from the perspective of VoIP performance, as well as its destructive impact on interfering data traffic. UDP yields satisfactory performance only at conditions of bandwidth availability and low contention. Nevertheless, it is more likely that this unresponsive protocol endangers network stability and integrity. The inefficiency of UDP outlines the importance of congestion control, since the high packet loss ratio at periods of congestion inevitably degrades voice quality.

We further explored the potential of TCP protocols with different characteristics, such as congestion control and error detection mechanisms. A comparative overview of the associated results reveals that congestion control does not hurt time-sensitive applications. On the contrary, it occasionally results in performance gains. Generally, an efficient congestion control/avoidance mechanism achieves remarkable performance during variable network conditions; it effectively reduces transmission rates at periods of congestion, while it does not restrict application performance at periods of bandwidth availability. TCP-Real, which incorporates a sophisticated congestion-avoidance mechanism is the most prominent solution, since it achieves superior performance; yet it excels in bandwidths sharing. TCP-Real incorporates an error classification mechanism in order to perform efficiently over wireless links. However, we showed that the protocol exhibits limited performance in the event of increased wireless errors. Exploring tradeoffs between performance and friendliness, we observed that TCP-Real has the slightest impact on corporate TCP traffic. TCP-Vegas also provides efficient transport services. However, this efficiency comes at a cost: Vegas occasionally does not achieve fairness. Our experiments outline the limited performance of TCP-Westwood, due to its inefficient bandwidth-estimation mechanism along with the absence of error classification. We also identified that long voice-data segments usually degrade voice quality. This observation is mostly addressed to TCP-Vegas.

Studying the behaviour of these TCP protocols, we reach the conclusion that congestion avoidance is more suitable for VoIP and generally for time-sensitive applications, which are relatively intolerant to packet losses. Therefore, efforts for improving today's VoIP performance from the perspective of end-to-end support should focus on the design of more effective and sophisticated congestion avoidance mechanisms. We have not yet outlined or deployed a specific mechanism towards efficient Internet voice transport. However, we have shown that TCP-Real provides satisfactory services, and it can be certainly tuned to respond more efficiently to packet errors caused by wireless links. Along these lines, future work will be focused on the design of a partially reliable protocol, as an extension of current TCP-Real, which will provide improved support for time-sensitive applications, including real-time voice communications.

REFERENCES

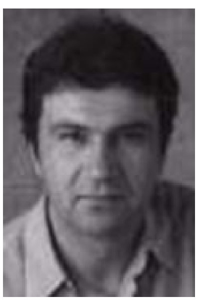
1. Karam M, Tobagi F. Analysis of the delay and jitter of voice traffic over the Internet. *Proceedings of IEEE INFOCOM 2001*, Anchorage, AK, U.S.A., April 2001.
2. Chiu D, Jain R. Analysis of the increase/decrease algorithms for congestion avoidance in computer networks. *Journal of Computer Networks* 1989; **17**(1):1–14.
3. Jacobson V. Congestion avoidance and control. *Proceedings of ACM SIGCOMM '88*, Stanford, U.S.A., August 1988.

4. Stevens W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. *RFC 2001*, January 1997.
5. Tsaoussidis V, Matta I. Open issues on TCP for mobile computing. *Journal of Wireless Communications and Mobile Computing* 2002; **2**(1):3–20.
6. Markopoulou A, Tobagi F, Karam M. Assessment of VoIP quality over Internet backbones. *Proceedings of IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000.
7. Noel E, Tang KW. Performance analysis of a VoIP access architecture. *Proceedings of IEEE ICPPW '04*, Canada, August 2004.
8. Oouchi H, Takenaga T, Sugawara H, Masugi M. Study on appropriate voice data length of IP packets for VoIP network adjustment. *Proceedings of IEEE GLOBECOM 2002*, Taipei, Taiwan, November 2002.
9. Garg S, Kappes M. An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks. *Proceedings of IEEE Wireless Communications and Networking Conference* (WCNC), New Orleans, LA, U.S.A., March 2003.
10. Takahashi A, Yoshino H. Perceptual QoS assessment technologies for VoIP. *IEEE Communications Magazine* 2004; **42**(7):28–34.
11. Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: a transport protocol for real-time applications. *RFC 1889*, IETF, January 1996.
12. Casner S, Jacobson V. RFC2508: compressing IP/UDP/RTP headers for low-speed serial links. IETF, Internet Draft, February 1999.
13. Floyd S, Handley M, Padhye J. A Comparison of Equation-based and AIMD Congestion Control. May 2000. URL: http://www.icir.org/floyd/papers.html
14. Floyd S, Handley M, Padhye J, Widmer J. Equation-based congestion control for unicast applications. *Proceedings of ACM SIGCOMM 2000*, Stockholm, Sweden, August 2000.
15. Yang YR, Kim MS, Lam SS. Transient behaviours of TCP-friendly congestion control protocols. *Proceedings of IEEE INFOCOM 2001*, Anchorage, AK, U.S.A., April 2001.
16. Yang YR, Lam SS. General AIMD congestion control. *Proceedings of the 8th International Conference on Network Protocols* (*ICNP*), Osaka, Japan, November 2000.
17. Floyd S, Fall K. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking* 1999; **7**(4):458–472.
18. Tsaoussidis V, Zhang C. The dynamics of responsiveness and smoothness in heterogeneous networks. *IEEE Journal on Selected Areas in Communications (JSAC)* 2005; **23**(6):1178–1189.
19. Grieco LA, Mascolo S. TCP Westwood and easy RED to improve fairness in high-speed networks. *Proceedings of the 7th International Workshop on Protocols for High-Speed Networks*, Berlin, April 2002.
20. Mascolo S, Casetti C, Gerla M, Sanadidi M, Wang R. TCP Westwood: bandwidth estimation for enhanced transport over wireless links. *Proceedings of MobiCom '01*, Rome, Italy, July 2001.
21. Zhang C, Tsaoussidis V. TCP real: improving real-time capabilities of TCP over heterogeneous networks. *Proceedings of the 11th IEEE/ACM NOSSDAV*, Port Jefferson, NY, U.S.A., June 2001.
22. Tsaoussidis V, Zhang C. TCP real: receiver-oriented congestion control. *Computer Networks* 2002; **40**(4):477–497.
23. Ramakrishnan K, Floyd S. A proposal to add explicit congestion notification (ECN) to IP. *RFC 2481*, January 1999.
24. Hengartner U, Bolliger J, Cross T. TCP Vegas revisited. *Proceedings of IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000.
25. Capone A, Fratta L, Martignon F. Bandwidth estimation schemes for TCP over wireless networks. *IEEE Transactions on Mobile Computing* 2004; **3**(2):129–143.
26. Brakmo L, Peterson L. TCP Vegas: end to end congestion avoidance on a global internet. *IEEE Journal on Selected Areas Communications (JSAC)* 1995; **13**(8):1465–1480.
27. ITU-T Recommendation G.711. Pulse code modulation (PCM) of voice frequencies. http://www.itu.int/rec/recommendation.asp?type = items&lang = E&parent = T-RECG.711-98811-I
28. ITU-T Recommendation G.711 Appendix I. A high quality low-complexity algorithm for packet loss concealment with G.711. http://www.itu.int/rec/recommendation.asp?type = items&lang = E&parent = T-REC-G.711-199909-I!AppI
29. Paxson V. End-to-end packet dynamics. *IEEE/ACM Transactions on Networking* 1999; **7**(3):277–292.
30. Andersen DG, Snoeren AC, Balakrishnan H. Best-path vs. multi-path overlay routing. *Proceedings of Internet Measurement Conference (IMC) 2003*, Miami, FL, U.S.A., October 2003.
31. Brady PT. A statistical analysis of on–off patterns in 16 conversations. *Bell System Technical Journal* 1968; **47**: 73–91.
32. Dang TD, Sonkoly B, Molnár S. Fractal analysis and modelling of VoIP traffic. *Proceedings of NETWORKS 2004*, Austria, June 2004.
33. ITU-T Recommendation P.862. Perceptual evaluation of speech quality (PESQ). http://www.itu.int/rec/recommendation.asp?type = items&lang = e&parent = T-REC-P.862-200102-I
34. ITU-T Recommendation G.107. The E-model, a computational model for use in transmission planning. December 1998.

35. Papadimitriou P, Tsaoussidis V, Tsekeridou S. The impact of network and protocol heterogeneity on real-time application QoS. *Proceedings of the 10th IEEE ISCC*, Cartagena, Spain, June 2005.
36. Papadimitriou P, Tsaoussidis V. Evaluation of transport services for VoIP. *Technical Report*: *TR-EE-DUTH-2005-12*.

AUTHORS' BIOGRAPHIES

Panagiotis Papadimitriou obtained his BSc in Computer Science from University of Crete, Greece; and MSc in Information Technology from University of Nottingham. He is currently a PhD Candidate in Electrical and Computer Engineering in Demokritos University, Greece. Panagiotis is a Lecturer in the Information Management Department of Technological Educational Institute (TEI) of Kavala, Greece. His research interests include Internet QoS, transport protocols and streaming media.



Vassilis Tsaoussidis received a BSc in Applied Mathematics from Aristotle University, Greece; a Diploma in Statistics and Computer Science from the Hellenic Institute of Statistics; and a PhD in Computer Networks from Humboldt University, Berlin, Germany (1995). Vassilis held faculty positions in Rutgers University, New Brunswick, SUNY Stony Brook and Northeastern University, Boston. In May 2003, Vassilis joined the Department of Electrical and Computer Engineering of Demokritos University, Greece. His research interests lie in the area of transport/network protocols, i.e. their design aspects and performance evaluation. Vassilis is an editor for *IEEE Transactions in Mobile Computing*, the Journal of *Computer Networks*, the Journal of *Wireless Communications and Mobile Computing* and the Journal of *Mobile Multimedia*. He participates in several Technical Program Committees in his area of expertise, such as INFOCOM, GLOBECOM, ICCCN, ISCC, EWCN, WLN, and several others.

