DS-TP: Deep-Space Transport Protocol

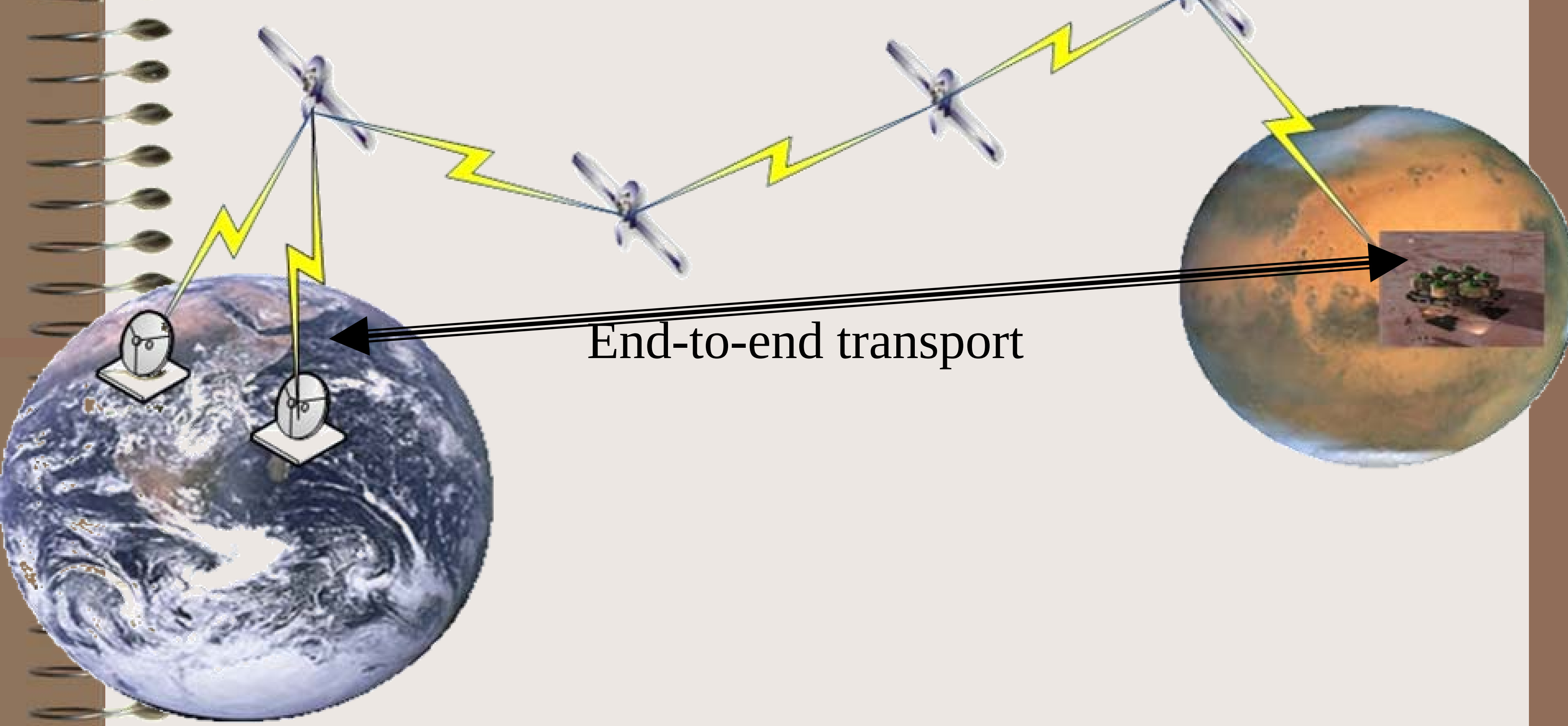
I. Psaras, G. Papastergiou & *V. Tsaoussidis*
Democritus University of Thrace, Xanthi, Greece
Computer Networks Laboratory
comnet.ee.duth.gr/comnet
(ipsaras, gpapaste, vtsaousi)@ee.duth.gr

Nestor Peccia
ESA/ESOC
Darmstadt,Germany
nestor.peccia@esa.int

Why transport?

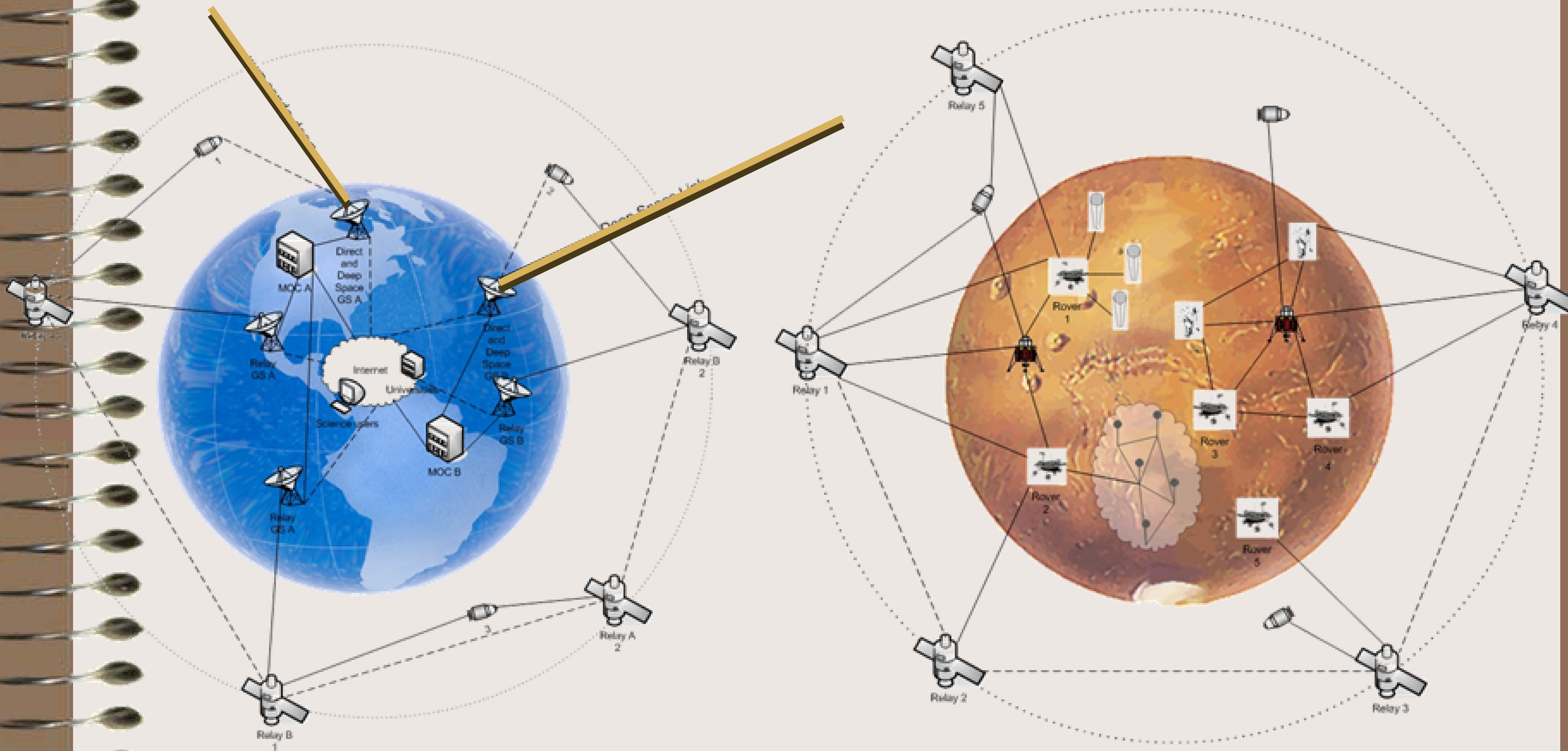
Deep space=>multiple hops

End-to-end reliability



Why transport?

End-to-end, flexible paths



What do we expect from a transport protocol

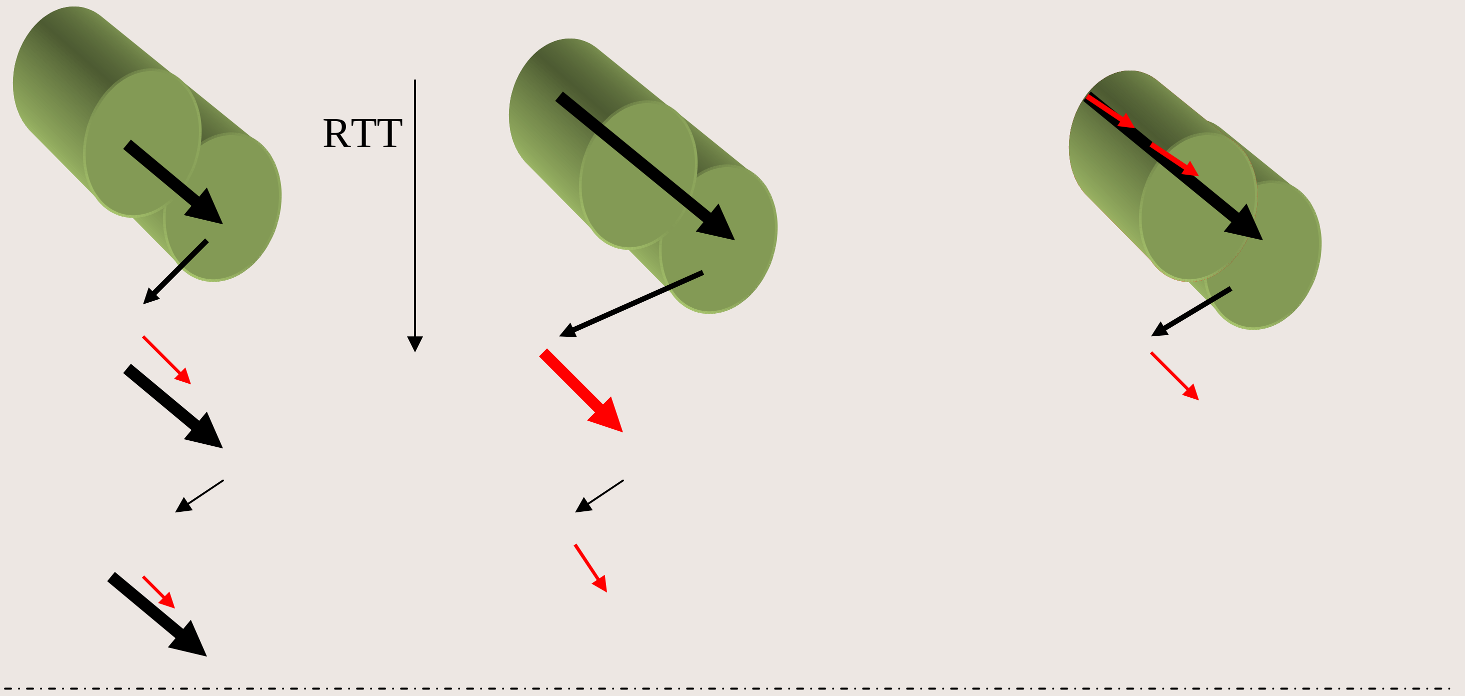
- Add end-to-end reliability, not just next hop
  - *Therefore, reliability is extended over flexible paths*
- Handle retransmissions when correction techniques fail
  - *Typically, implemented with sequence numbers, timeouts, and packet_in_flight buffering* in a <u>closed loop</u> system
- Exploit available resources fully
  - *The classic tradeoff among bandwidth and delay*
- Handle flow and congestion control when necessary
  - *Prescheduled connections do not typically face this challenge*

What gap does DS-TP fill in

- The scheduling for retransmissions in an open loop system, which decouples feedback from transmission scheduling
  - Exploiting bandwidth fully, does not mean we exploit bandwidth well.
  - When delay is the dominant factor trading some bandwidth wouldn't damage efficiency
  - When the error pattern (e.g. periodic burst) allows for responsive strategies:
    - How intensively and when to retransmit?

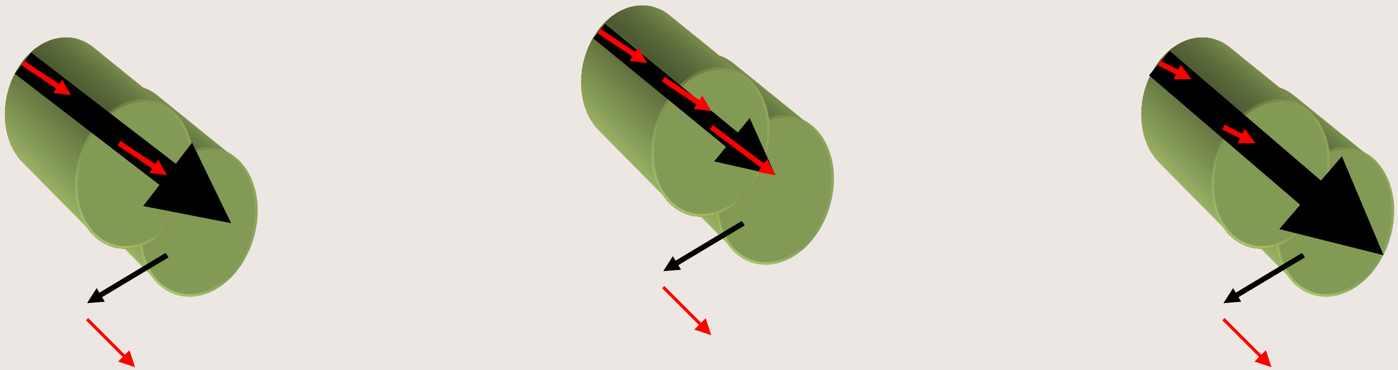
Main contribuion

Traditional – space - suggestion



Zoom in: volume and timing

Error nature may determine policy



How DS-TP works

- There are at least two possible ways to implement DS-TP's strategy

1. To apply redundancy on a *per packet basis*, and depending on the error rate to regulate the redundancy pattern - > this will also determine the associated delay of retransmitted packets.
   - Requirement: to avoid delay longer than the RTT

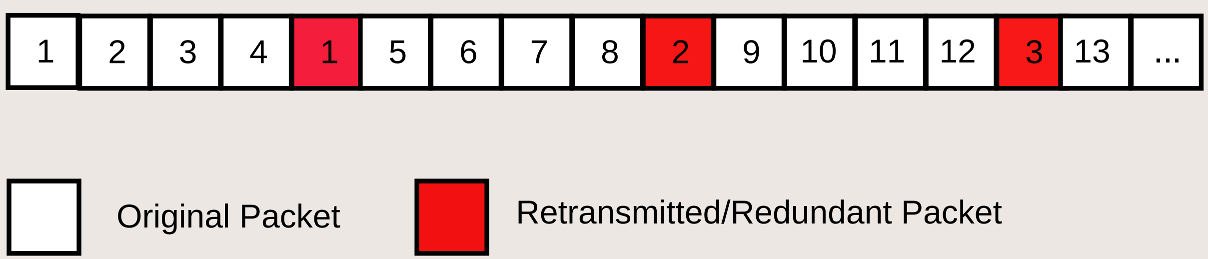
How DS-TP works

2. To retransmit data on a per-window basis, for the whole or a portion of the window and regulate the trailer with delay that corresponds best to the `min` *probability of loss* –
   - occasionally relying on error detection strategies for large files.

Deep-Space Transport Protocol: DAR

One redundant packet is sent every (1/PER) – 1 original packets.

For example, if PER = 20%, the transmission sequence is:



Why DS-TP

- **TP-Planet**: somewhat over-qualified.
- **RCP-Planet**: un-reliable.
- **SCPS-TP**: the space version of TCP
- **Saratoga**: pretty simple and efficient but slow compared to DS-TP.
- **CFDP**: application layer protocol, with transport layer functionalities. Similar to Saratoga, but less efficient.
- **LTP**: includes a unique mechanism to differentiate between blocks of data that need 100% reliability and blocks of data that do not.

Some (potentially-interesting) concepts

- SNACKs
  - (1 – 2)

SNACK 1 reports missing packets-
Trigger no retransmission
SNACK 2 request retransmission

- Diffs
  - Adjust Time and distance
  - Shift SN based on RTT

Diff_seq_no -> retr. Packet gap
Diff_time -> corresp. Time gap
Elapsed_t+expected_t <2RTT

- Tradeoff
  - Bandwidth delay and error rate

Diff_time large=>policy canceled
Diff_time small=>suff. bandwidth

Deep-Space Transport Protocol

- Actual Rate and `c_seqno`
- Retransmission Rate and `r_seqno`
- Line Rate = Actual Rate + Retransmission Rate
- Retransmission Rate = Packet Error Rate

- Actual Rate = (1 - PER%)*Link Rate

Deep-Space Transport Protocol: DAR

- According to r_seqno, the packet with sequence number c_seqno will be retransmitted after diff_pkts:

$$diff\_pkts = [(\frac{1}{error\_rate} - 1) \cdot c\_seqno] - r\_seqno$$

DSTP Sender

- Transmits original and redundant packets at line rate
  - Redundant transmission rate depends on measured packet error rate
  - Redundant transmission rate = packet error rate
- Calculates error rate using Snack-1 and Snack-2 information
- Retransmits lost packets immediately at line rate upon reception of Snack-2
  - Snack2 retransmissions do not count for redundant transmission rate

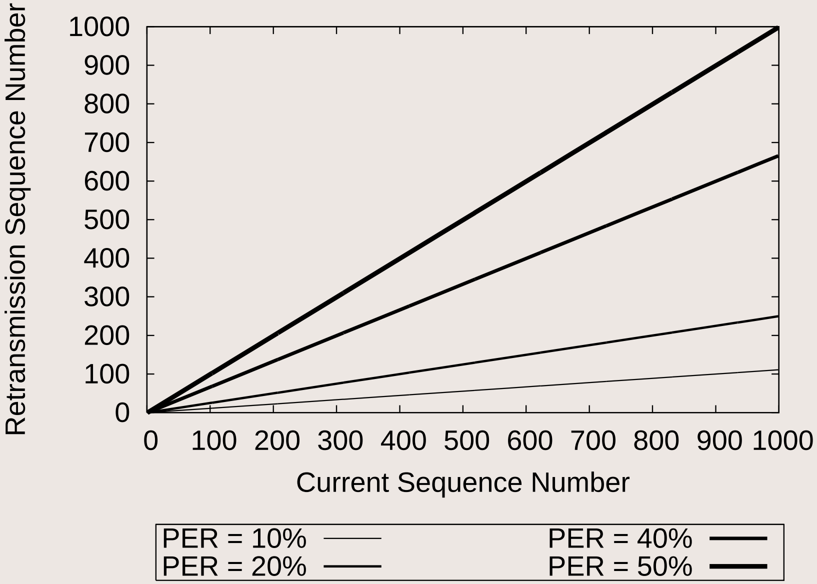
DSTP Receiver

Calculates error rate using packet sequence number information

- Estimates redundant transmission rate based on measured error rate
- Informs sender for missing packets with SNACKs
  - Continuous blocks of receiver's buffer
  - Sends Snack-1 for missing packets that a redundant packet is pending
  - Sends Snack-2 for missing packets that the redundant packet is lost or for missing packets that no redundant packet is transmitted
- Snack-2 triggers packet retransmissions
- A timer is set for every Snack-2 sent

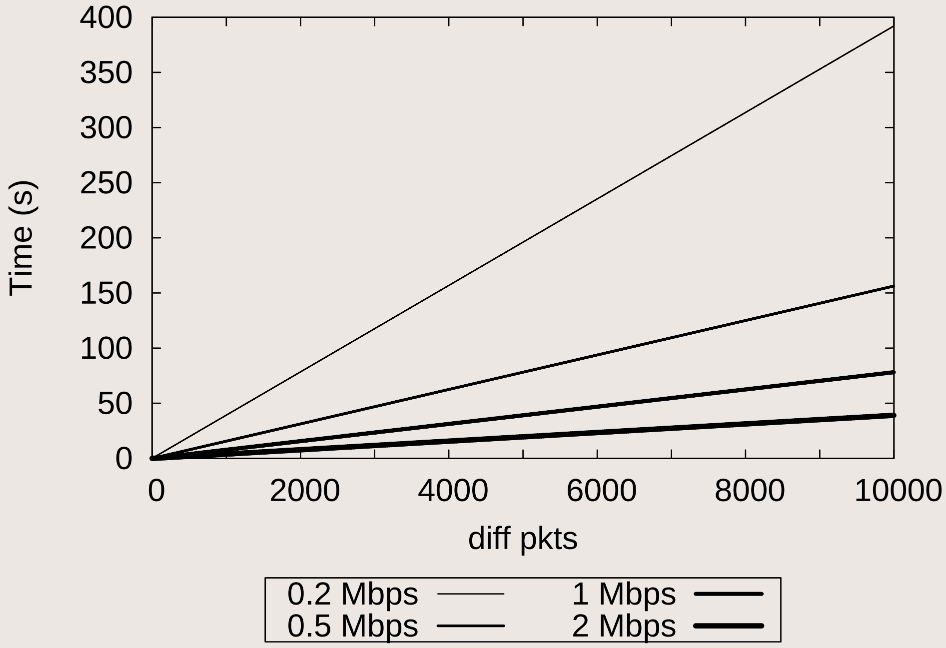
DS-TP Scenario

- We graph *r_seqno* in conjunction with the current sequence number (*c_seqno*):

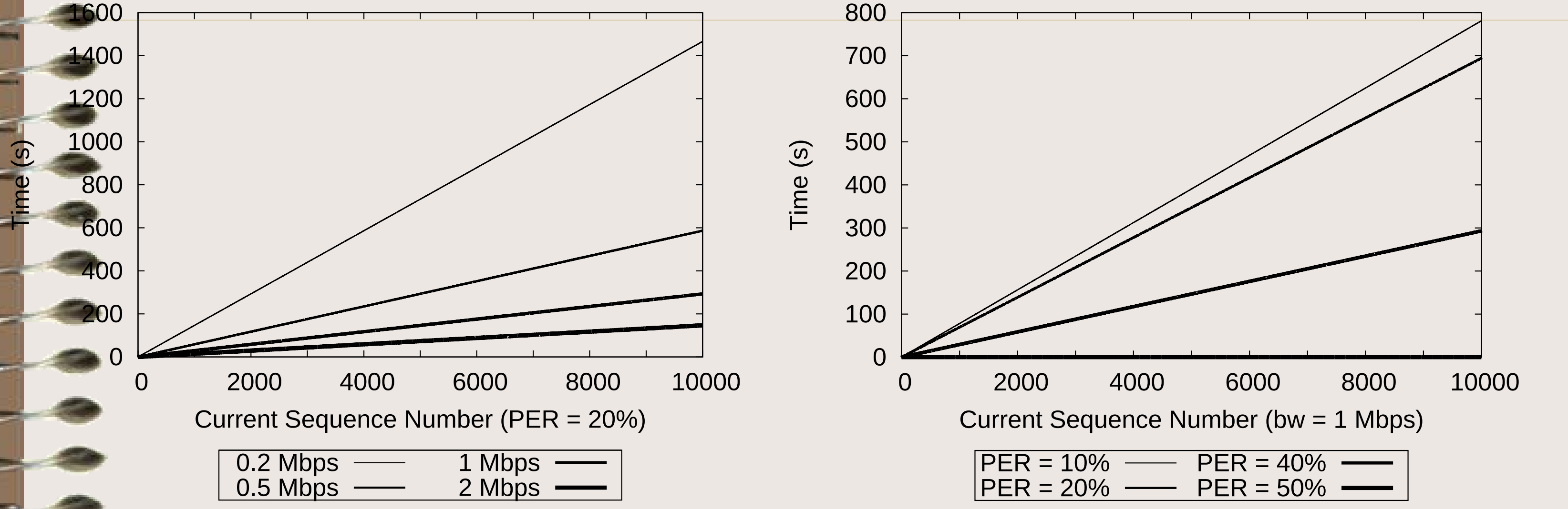


DS-TP Scenario

- A *x* Mbps link can transfer *x/8*MBps or (1024**x)*/8KB/s.
- Therefore *diff_pkts* require: $diff\_time = \frac{8 \cdot diff\_pkts}{1024 \cdot x}$



DS-TP Evaluation



- We see that the DAR retransmission interval may be up to 25mins for the 10,000th packet.
- If the reverse link propagation delay is smaller than the retransmission interval, then the DAR's functionality is cancelled.

Protocol Evaluation Framework

- We compare the performance of DS-TP with the *Fixed-Rate Transport Protocol* (FR-TP).
- FR-TP is similar to Saratoga and CFDP.
- FR-TP transmits data on a fixed, predetermined rate, equal to the line rate.
- SNACKs are sent to the sender only after the file transfer is complete (i.e., the sender has transmitted all data into the transmission link).

DS-TP vs FR-TP

- Therefore, FR-TP needs *n_frtp* rounds in order to complete the file transfer:

$$n_{frtp} = \log_y(y^n) = \log_y(\frac{1}{fs}) = \frac{\log \frac{1}{fs}}{\log y}$$

DS-TP vs FR-TP

- During the 1st round, DS-TP transmits *fs* + *r_1* MBs, in total, where *r_1* are the DAR retransmissions.
- During the 1st round, *fs* - *r_1* KBs are sent once and *r_1* KBs are sent twice.
- Provided that the channel PER applies uniformly for the total number of packets:
  - *fs* - *r_1* are lost with probability *y*, and
- *r_1* are lost with probability *y*^2, where:

$$r_1 = fs \cdot \frac{y}{1 - y}$$

DS-TP vs FR-TP

- Therefore, during the 1st round, *a_1* packets are lost:

$$a_1 = (fs - r_1)\cdot y + r_1 \cdot y^2$$

- Substituting *r_1* to the above Equation, we have:

$$a_1 = fs \cdot y \cdot (1 - y)$$

DS-TP vs FR-TP

- During the 2nd round, DS-TP transmits *a_1 + r_2* MBs, in total, where *r_2* are the DAR retransmissions.
- During the 2nd round, *a_1 - r_2* KBs are sent once and *r_2* KBs are sent twice, where
- *a_1 - r_2* are lost with probability *y*, and
- *r_2* are lost with probability *y*^2, where

$$r_1 = fs \cdot \frac{y}{1-y}$$

DS-TP vs FR-TP

- Therefore, during the 2nd round, *a_2* packets are lost:

$$a_2 = (a_1 - r_2)\cdot y + r_2 \cdot y^2$$

- Substituting *r_2* into the previous Equation, we have:

$$a_2 = fs\cdot y^2 \cdot (1 - y)^2$$

DS-TP vs FR-TP

- DS-TP will complete the file transfer, when:

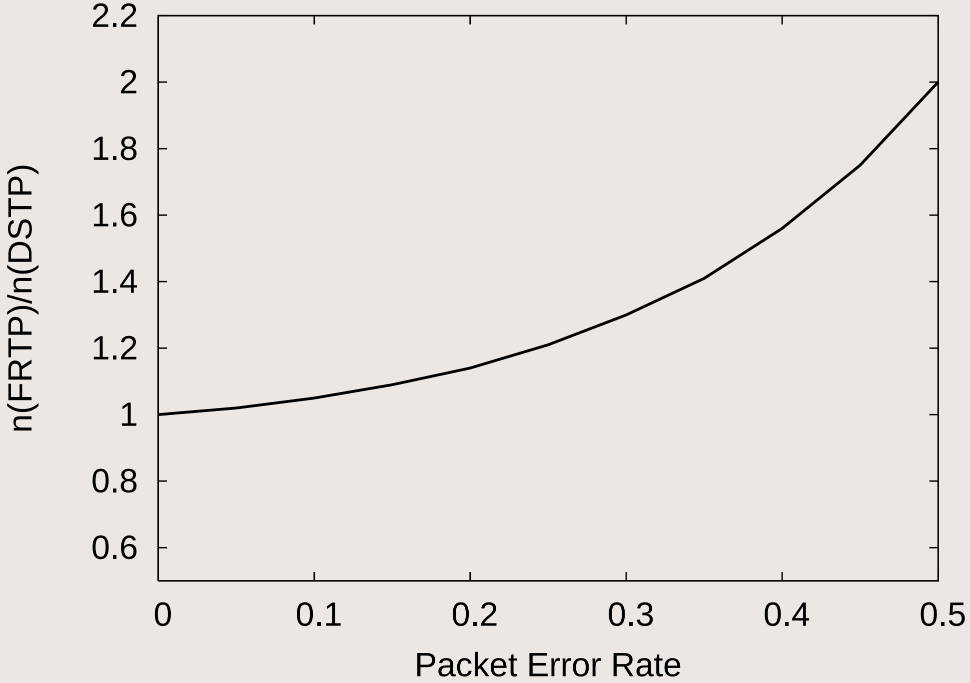
$$fs \cdot y^n \cdot (1-y)^n < 1 \ packet$$

- Hence, DS-TP needs *n_dstp* rounds to transfer a *fs* MBs file:

$$n_{dstp} = \log_{[y\cdot(1-y)]}[y\cdot(1-y)]^n = \log_{[y\cdot(1-y)]}(\frac{1}{fs}) \Rightarrow$$

$$n_{dstp} = \frac{\log \frac{1}{fs}}{\log(y\cdot(1-y))}$$

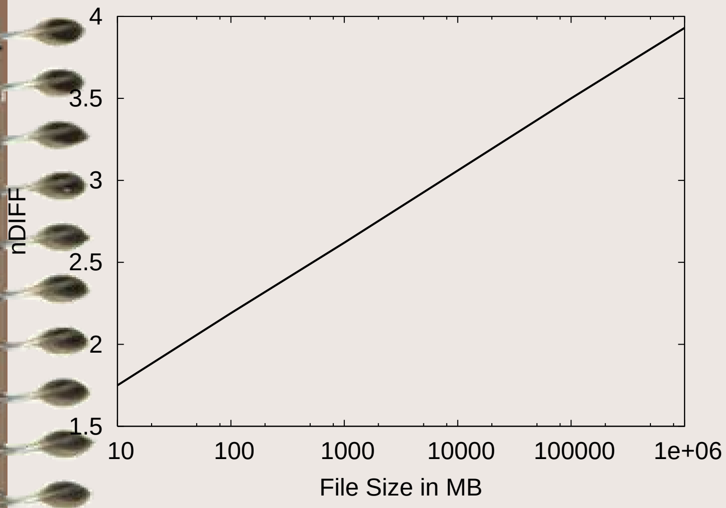
DS-TP vs FR-TP

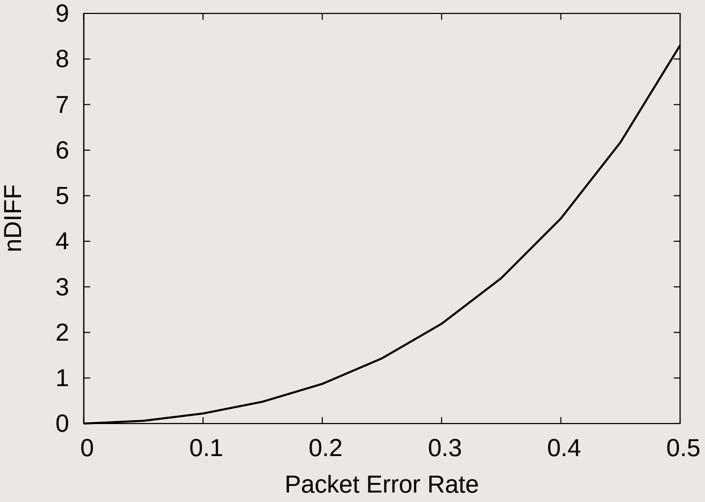


FRTP requires more RTT as the error rate grows

DS-TP vs FR-TP

- DS-TP is up to 8 rounds faster than FR-TP.
- The difference increases even more for larger file sizes.





Conclusions – DSTP:

- Reduces communication/connectivity time (though minimizing # of RTTs required), which also cancels the need for extraordinary buffering requirements iff

$$DSTP\ Tr.D. - FRTP\ Tr.D. \leq n_{diff} \cdot RTT$$

- Requires no extension of infrastructure and therefore minimizes cost of deployment.

<u>Ultimate goal</u>: Increase the amount of data transferred within the given timeframes

<u>Working assumption</u>: Delay is the problem – not bandwidth