CS 525M – Mobile and Ubiquitous Computing Seminar

Presented by Devanshu Mehta

Introduction

- **Improving TCP Performance over Wireless Links**
  - Motivation
  - Types of Solutions
  - Related Work
  - Implementation
  - Results
  - Conclusions

Motivations

- TCP is geared towards handling packet loss due to congestion.
- Losses on wireless connections are usually just due to the nature of the links.
  - Sporadic high error rates
  - Intermittent connectivity
- In such a situation, TCP assumes congestion and goes in to slow start.
- This degrades performance since the connection can actually handle a much higher throughput.

Proposed Solutions

- End to End
  - Attempt to make TCP sender handle losses using:
    - Selective ACKs (SACK)
    - Explicit Loss Notification (ELN)
- Split Connection
  - Hide the wireless link from the sender; different protocol over wireless hop.
- Link-Layer
  - Hide loss from sender through local retransmissions and forward error correction.

Purpose of Experiments

- What combination of mechanisms results in the best performance for each of the protocol classes?
- How important is it for link-layer schemes to be aware of TCP algorithms to achieve high end-to-end throughput?
- How useful are selective acknowledgements in dealing with lossy links, especially in the presence of bursty losses?
- Is it important for the end-to-end connection to be split in order to effectively shield the sender from wireless losses and obtain best performance?

Implementation Details

TABLE I
SUMMARY OF PROTOCOLS STUDIED IN THIS PAPER

Name	Category	Special Mechanisms
E2E	end-to-end	standard TCP-Reno
E2E-NEWRENO	end-to-end	TCP-NewReno
E2E-SMART	end-to-end	SMART-based selective acks
E2E-IETF-SACK	end-to-end	IETF selective acks
E2E-ELN	end-to-end	Explicit Loss Notification (ELN)
E2E-ELN-RXMT	end-to-end	ELN with retransmit on first dupack
LL	link-layer	none
LL-TCP-AWARE	link-layer	duplicate ack suppression
LL-SMART	link-layer	SMART-based selective acks
LL-SMART-TCP-AWARE	link-layer	SMART and duplicate ack suppression
SPLIT	split-connection	none
SPLIT-SMART	split-connection	SMART-based wireless connection

Results: Methodology

- •BSD/OS TCP Reno
- •Focuses on traffic TO mobile device.
- •Exponentially distributed bit error model
- •Losses generated in both directions
- •No losses due to congestion
- •First tested with average error rate of 1 in 64kb
- •The tested with bursty errors
- •The results should be consistent for other patterns of losses as well

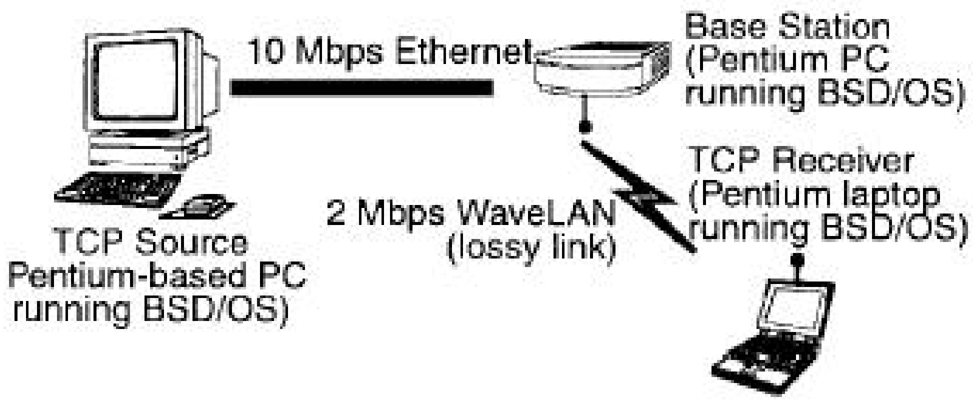


Fig. 2. Experimental topology. There were an additional 16 Internet hops between the source and base station during the WAN experiments.

Results: Link-Layer

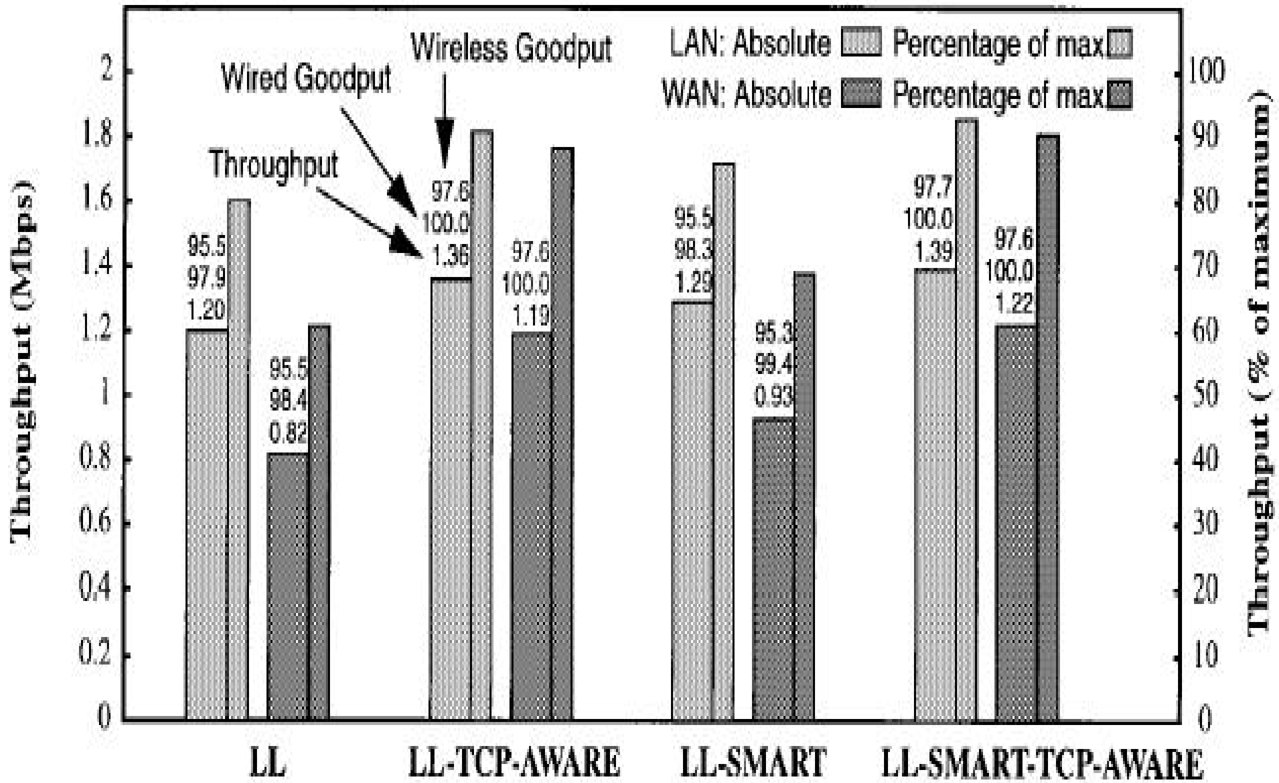


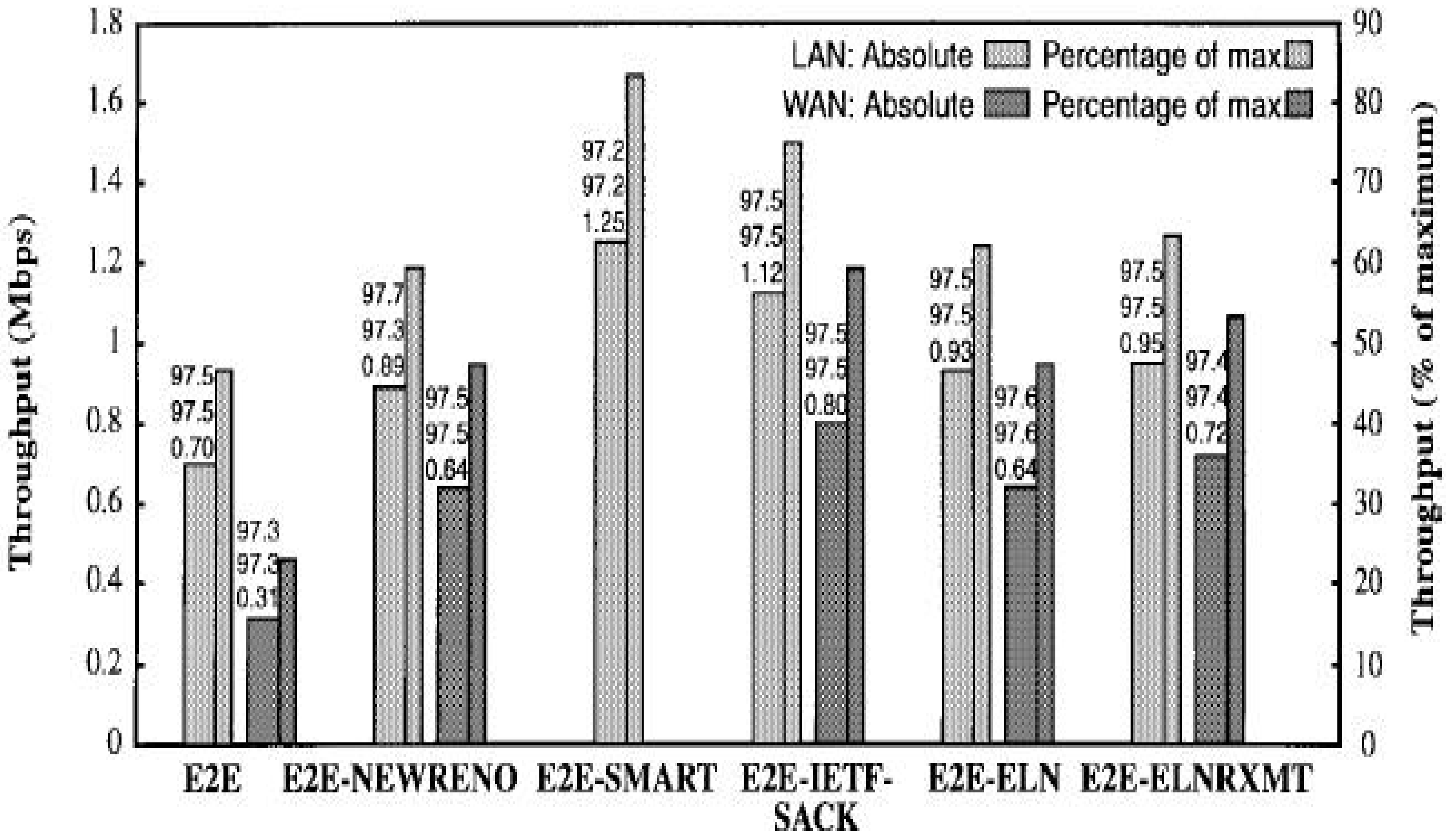
Fig. 3. Performance of link-layer protocols: bit-error rate = 1.9×10^{-6} (1 error/65536 bytes), socket buffer size = 32 kbytes. For each case, there are two bars: the thick one corresponds to the scale on the left and denotes the throughput in megabits per second; the thin one corresponds to the scale on the right and shows the throughput as a percentage of the maximum, i.e., in the absence of wireless errors (1.5 Mb/s in the LAN environment and 1.35 Mb/s in the WAN environment).

Results: Link-Layer

- LL-TCP-AWARE has better performance than LL because of in-order transmission of packets.
- In pure LL, out of order packets cause duplicate ACKs and hence invoke fast retransmit.
- This degradation is more acute on WANs.

	LL	LL-TCP-AWARE	LL-SMART	LL-SMART-TCP-AWARE
LAN (8 KB)	1.20 (95.6%,97.9%)	1.29 (97.6%,100%)	1.29 (96.1%,98.9%)	1.37 (97.6%,100%)
LAN (32 KB)	1.20 (95.5%,97.9%)	1.36 (97.6%,100%)	1.29 (95.5%,98.3%)	1.39 (97.7%,100%)
WAN (32 KB)	0.82 (95.5%,98.4%)	1.19 (97.6%,100%)	0.93 (95.3%,99.4%)	1.22 (97.6%,100%)

Results: End-to-End



Results: End-to-End

TABLE III

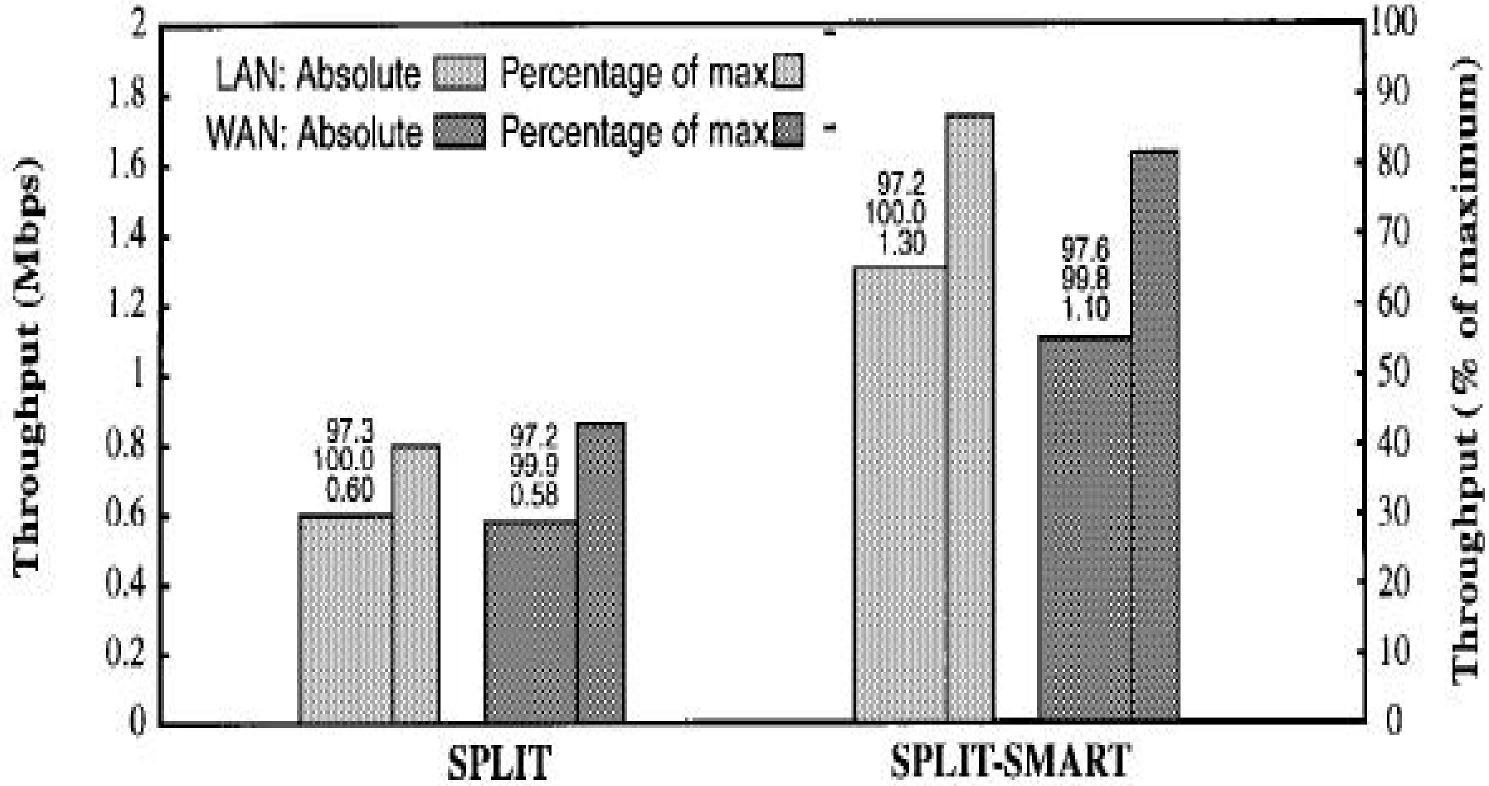
SUMMARY OF RESULTS FOR THE END-TO-END SCHEMES FOR AN AVERAGE ERROR RATE OF ONE EVERY 65536 BYTES OF DATA. THE NUMBERS IN THE CELLS FOLLOW THE SAME CONVENTION AS IN TABLE II

	E2E	E2E-NEWRENO	E2E-SMART	E2E-IETF-SACK	E2E-ELN	E2E-ELN-RXMT
LAN (8 KB)	0.55 (97.0,96.0)	0.66 (97.3,97.3)	1.12 (97.6,97.6)	0.68 (97.3,97.3)	0.69 (97.3,97.2)	0.86 (97.4,97.3)
LAN (32 KB)	0.70 (97.5,97.5)	0.89 (97.7,97.3)	1.25 (97.2,97.2)	1.12 (97.5,97.5)	0.93 (97.5,97.5)	0.95 (97.5,97.5)
WAN (32 KB)	0.31 (97.3,97.3)	0.64 (97.5,97.5)	N.A.	0.80 (97.5,97.5)	0.64 (97.6,97.6)	0.72 (97.4,97.4)

Results: End-to-End

- E2E < E2E w/Partial ACK < E2E w/ELN < E2E w/Selective ACKs
- ELN performs better because of sender’s awareness of wireless link.
- E2E based on Selective Acknowledgement (SMART and IETF) schemes work best among E2E (1.25Mbps).
- Still, they do not perform as well as the best of the LL schemes (1.39Mbps).

Results: Split Connection



Results: Split Connection

TABLE IV
SUMMARY OF RESULTS FOR THE SPLIT-CONNECTION SCHEMES AT AN AVERAGE ERROR RATE OF 1 EVERY 64 kbytes

	SPLIT	**SPLIT-SMART**
LAN (8 KB)	0.54 (97.4%,100%)	1.30 (97.6%,100%)
LAN (32 KB)	0.60 (97.3%,100%)	1.30 (97.2%,100%)
WAN (32 KB)	0.58 (97.2%,100%)	1.10 (97.6%,100%)

Results: Wrapping Up

- Burst Errors:
- SMART Selective ACKs better than simple LL-TCP-AWARE

TABLE V

THROUGHPUTS OF LL-TCP-AWARE AND LL-SMART-TCP-AWARE AT DIFFERENT BURST LENGTHS. THIS ILLUSTRATES THE BENEFITS OF SACK'S, EVEN FOR A HIGH-PERFORMANCE, TCP-AWARE LINK PROTOCOL

Burst Length	LL-TCP-AWARE (Mbps)	LL-SMART-TCP-AWARE (Mbps)
2	1.25	1.28
4	1.02	1.20
6	0.84	1.10

Purpose – Reloaded!

- **What combination of mechanisms results in the best performance for each of the protocol classes? – LL-SMART-TCP-AWARE**
- **How important is it for link-layer schemes to be aware of TCP algorithms to achieve high end-to-end throughput? – Important!**
- **How useful are selective acknowledgements in dealing with lossy links, especially in the presence of bursty losses? – Very Useful!**
- **Is it important for the end-to-end connection to be split in order to effectively shield the sender from wireless losses and obtain best performance? – Yes!**

Conclusions

- A reliable link-layer protocol that uses knowledge of TCP (LL-TCP-AWARE) is best among LL protocols as it gives best throughput and least retransmissions.
- LL protocols also perform better than Split Connection schemes proving the split is not necessary for improved performance.
- SMART schemes with SACK perform best among end-to-end; but not as good as LL.
- End-to-End provide improved performance and are promising as they require no support at intermediate nodes.