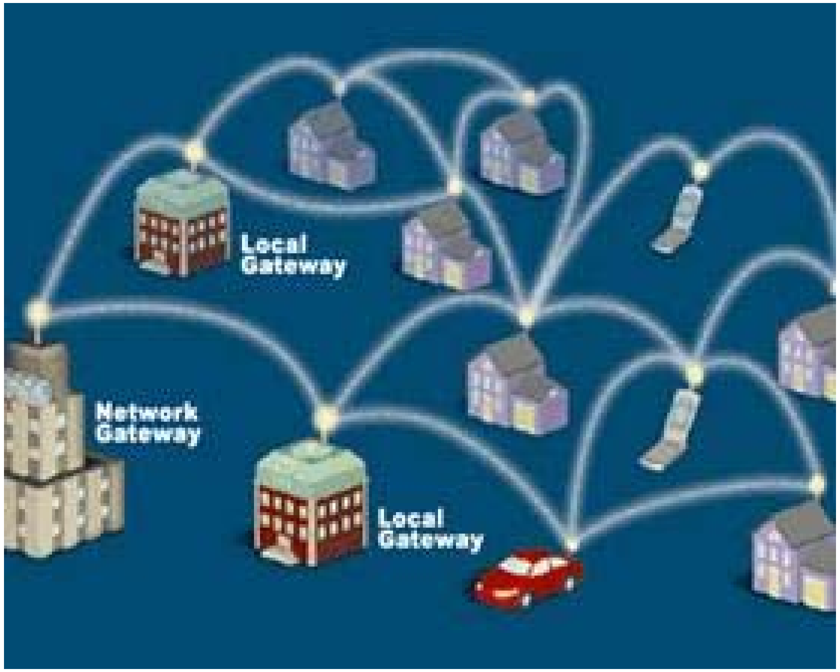
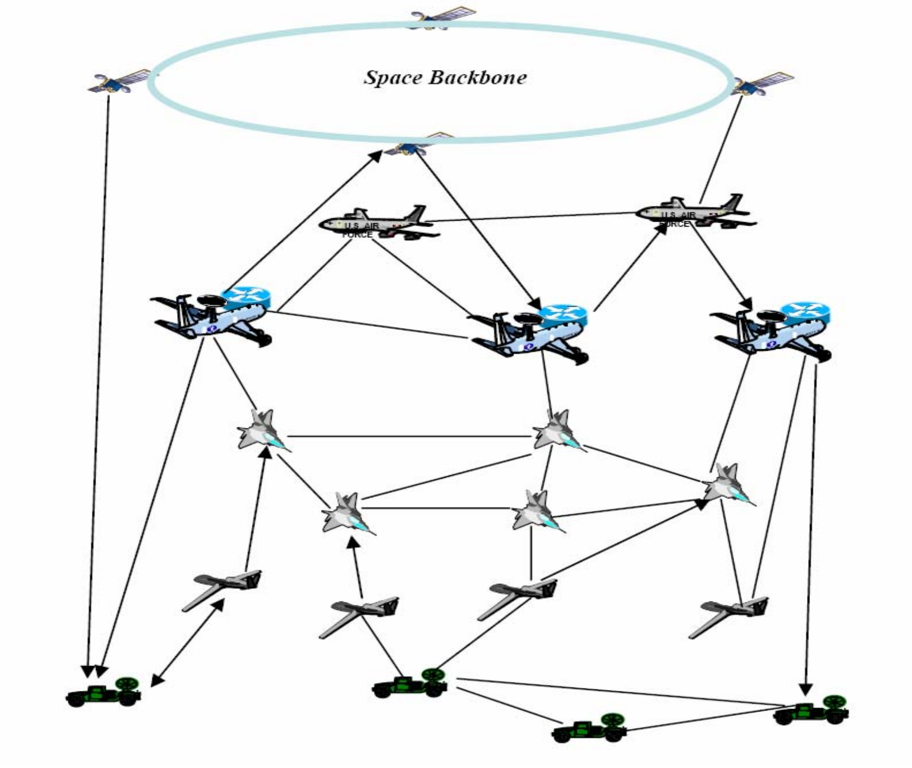
Experimental Study of Link and Transport Protocols in Interference-Prone Wireless LAN Environments

Vijay Subramanian, RPI
K.K.Ramakrishnan, ATT Labs
Shivkumar Kalyanaraman, RPI/IBM

Emerging Conditions for TCP



Meshed Wireless Networks



Multi-tier MANETs, VANETs

Objectives

- End-end TCP performance suffers non-linearly with increasing loss rates.
- We study the nature of wireless losses in the presence of interference (modeled by noise injection at receiver).

- High ARQ at link layer: how does it impact both raw and residual loss rates? What is the impact of latency on TCP?
- Understand the three-way trade-off between Goodput, Latency and Loss Rate at the link layer; how can we optimize this trade-off?

- Developed link traces for loss rate and latency – and used it to test our proposed link and transport layer protocols (LT-TCP and LL-HARQ).

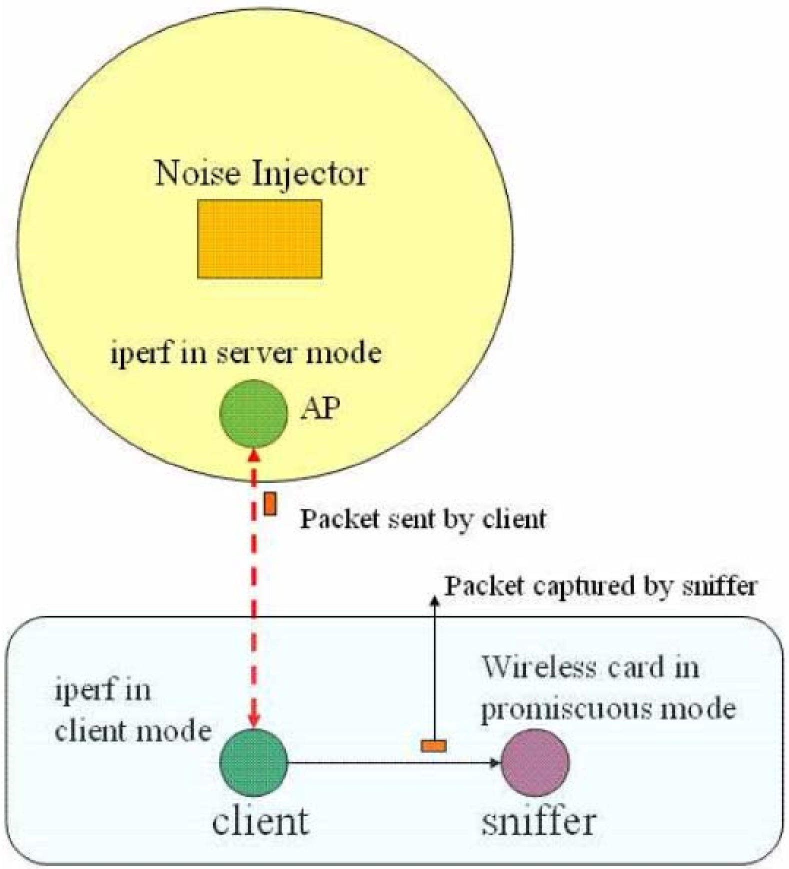
Observations

- There exists a three-way tradeoff between goodput, latency and residual loss.

- We find that the division of reliability burden between link and transport layers does indeed matter.
  - Link layer must significantly decrease the residual loss rate to avoid accumulation of end-to-end loss rate.
  - However, persistent link-layer ARQ in highly lossy wireless networks can lead to large and variable delays for end-to-end TCP.
  - Hybrid ARQ at link layer can be designed to dramatically reduce residual loss rate even with one retry attempt
  - End-to-end Loss-tolerant TCP can compensate for the residual accumulated end-to-end loss rate

- We demonstrate these issues & validate our proposed solutions through ORBIT experiments, and ORBIT traces-driven simulations

Experimental Topology/Setup

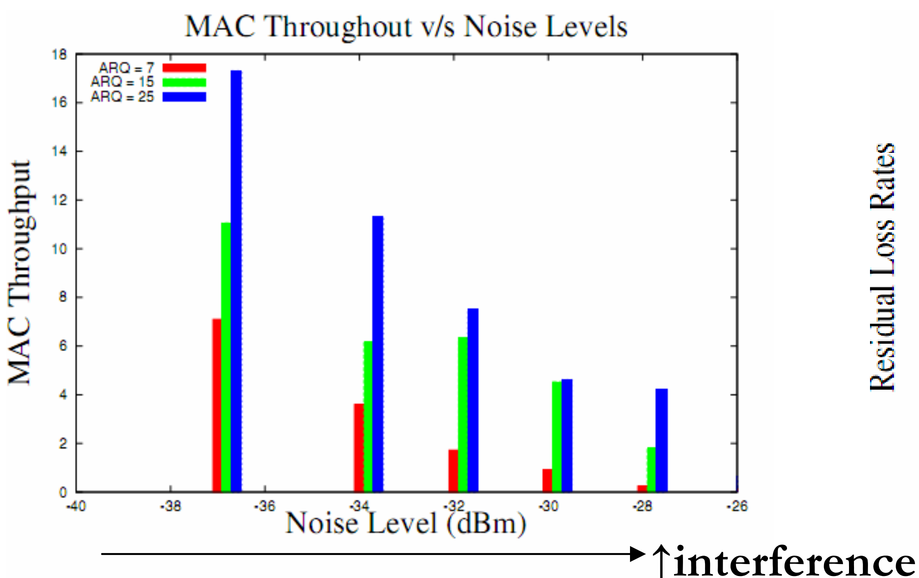


- **Experiments:** conducted on the ORBIT testbed on the 802.11a frequency range using Linux (2.6.18) and the *madwifi* driver for an Atheros card.

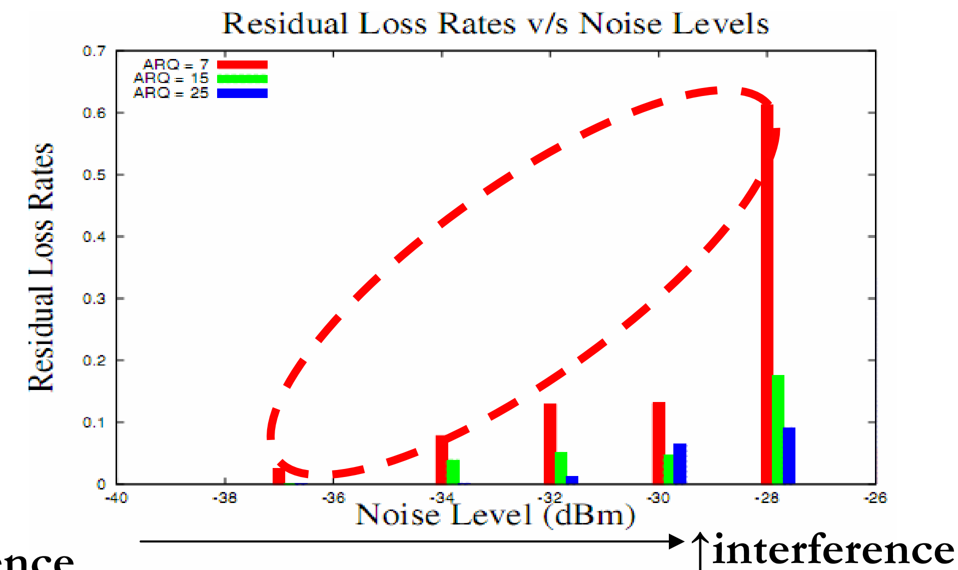
- One sniffer (sufficient to monitor loss and latency).
- Goal: obtain a model of a wireless link that characterizes the retransmission and latency in the face of interference.
- Noise level is varied from -40 dBm (no noise) to -25 dBm (high noise).

- Multiple runs were conducted and a representative set is discussed in detail. (default ARQ =15 unless specified)
- Various link and TCP metrics were recorded.

MAC Throughput vs Residual Loss Rates: effect of ARQ persistence



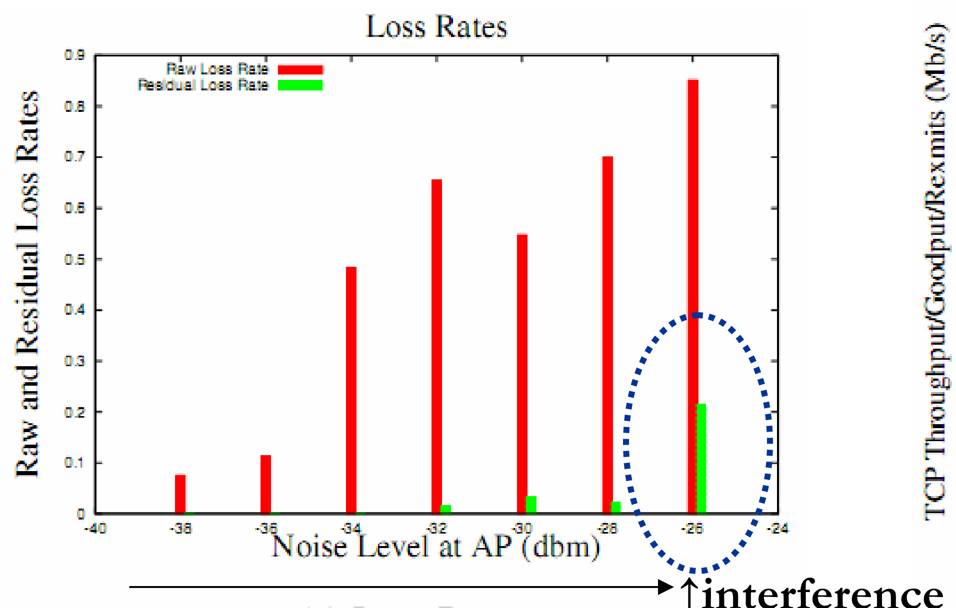
(a) MAC Throughput



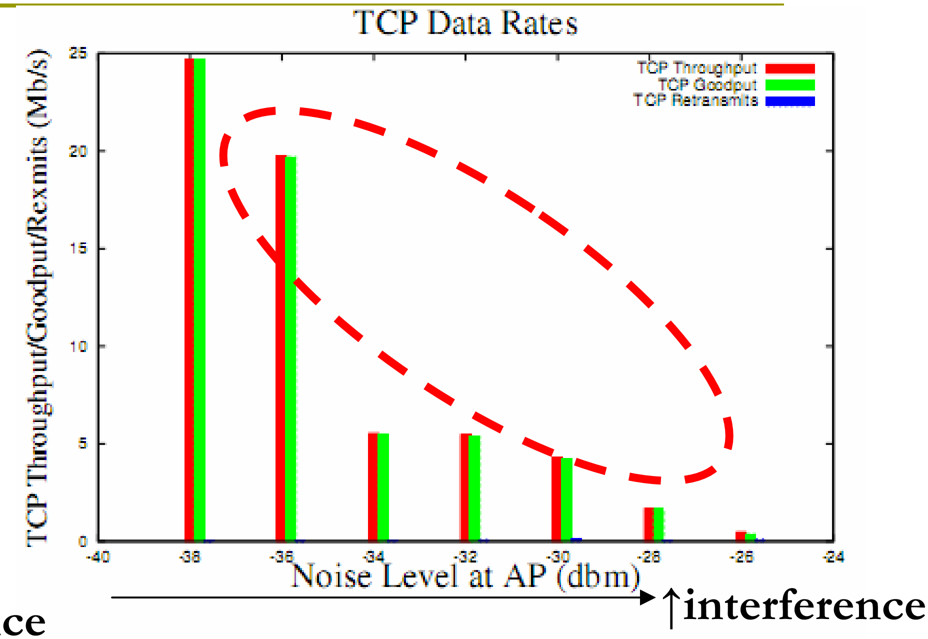
(b) Residual Loss Rates

- As the ARQ persistence is varied from 7 to 15 to 25, the residual loss rate decreases with increasing ARQ.
- Even though MAC throughput improves with higher ARQ, the penalty paid is an increase in latency and lowered goodput.
  - MAC throughput is not reflective of goodput

Link Loss Rates and TCP performance with ↑interference



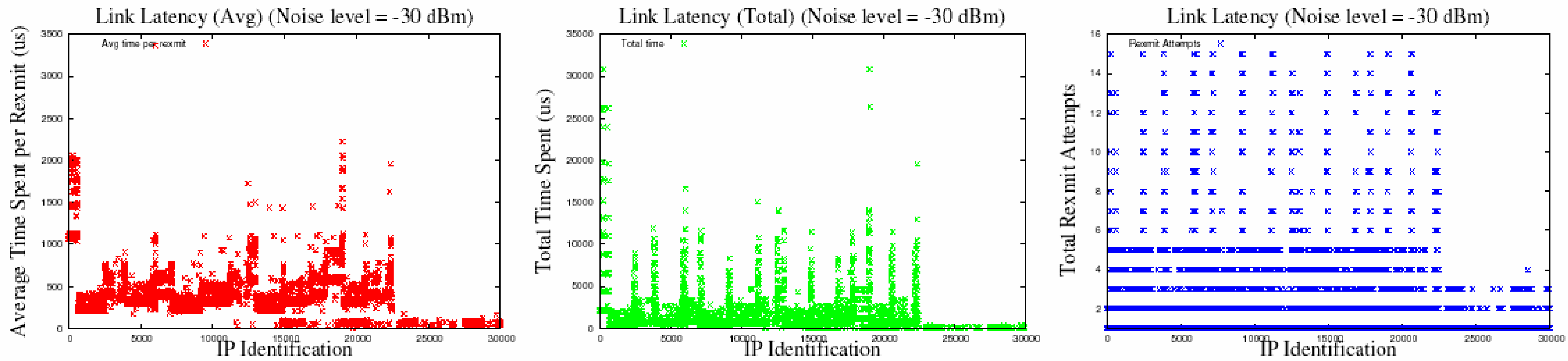
(a) Loss Rates



(c) Statistics of TCP performance

- The figure on the left shows the raw and the residual loss rates (after 15 attempts) exported to the transport layer.
- The raw loss rate can be quite high (>50% after -34 dBm interference) which leads to a significant residual loss rate seen by TCP *even with high ARQ persistence*.
- For TCP, as the residual loss rate goes above 5% (-29 dBm), the performance drops rapidly.

Link layer latencies:@ -30dBm (high) interference



(a) Avg Time per ARQ attempt

(b) Total Latency

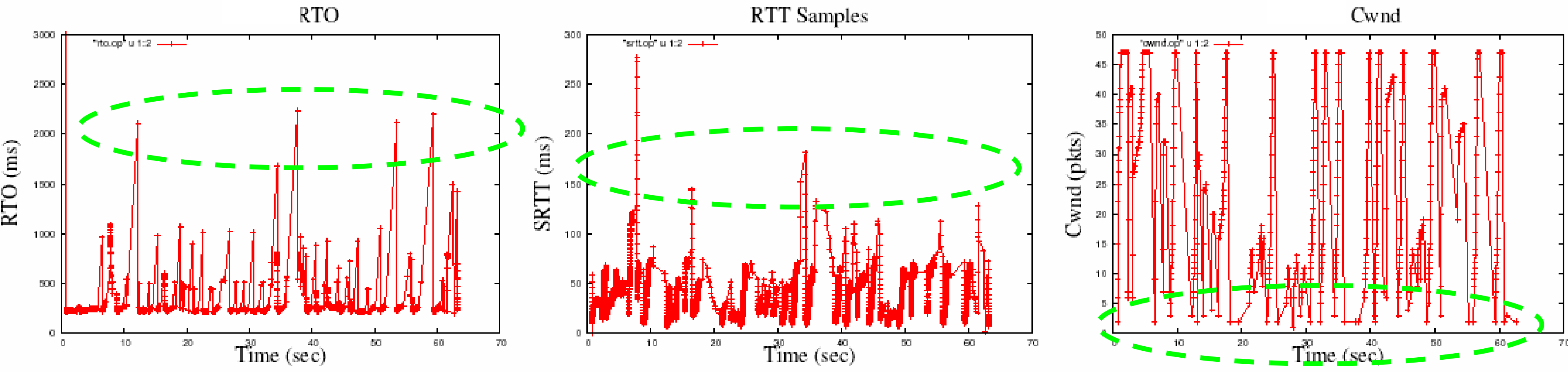
(c) Number of ARQ attempts taken

Link layer latencies for noise level -30 dBm: high noise (interference) case.
Figure (a) shows the average link latency per retransmission attempt in microseconds.

- Note: each packet takes several attempts, and each attempt is a random time chosen between a min/max value.
- Note that there is no back-off between successive retransmissions.
- With exponential backoff, the latency penalty will be even worse.

Figure (b) shows the total time spent for a given packet.
Figure (c) shows the number of retransmission attempts made for different packets.

TCP dynamics: RTO, SRTT, CWND: @ -30 dBm interference



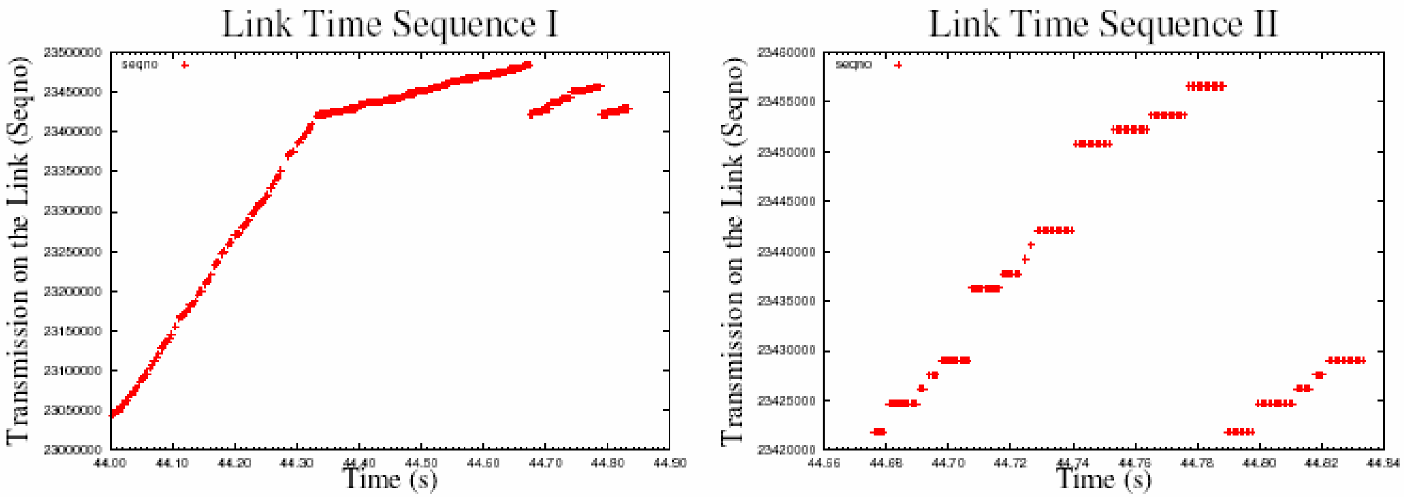
(a) Retransmission Timeout (b) Smoothed RTT (c) Congestion Window

Window and RTT/RTO plots (noise level -30dBm):
These three figures show the behavior of the RTO, smoothed RTT (SRTT) and the congestion window in packets for high interference level -30dBm.

Note the ***__RTO spikes__*** are ***__caused by the variation in link latency caused by high ARQ persistence__***.

Leads to spurious timeouts and CWND dropping to 1.

Eg: Spurious Timeout: @ -30 dBm interference



(a) Time Interval: 44.0-44.9 s

(b) Time Interval: 44.66-44.84 s

For noise level -30 dBm, we see the occurrence of a spurious retransmission and timeout.

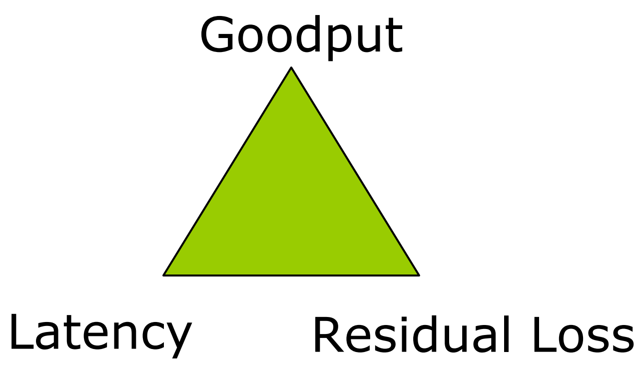
The first figure shows the transmission of packets around the period of interest and we then zoom into this further.

Packet with seq no 23421817 is sent thrice.

While attempt 2 is successful, the high latency (due to large link ARQ delays) causes a timeout resulting in a third transmission of the packet.

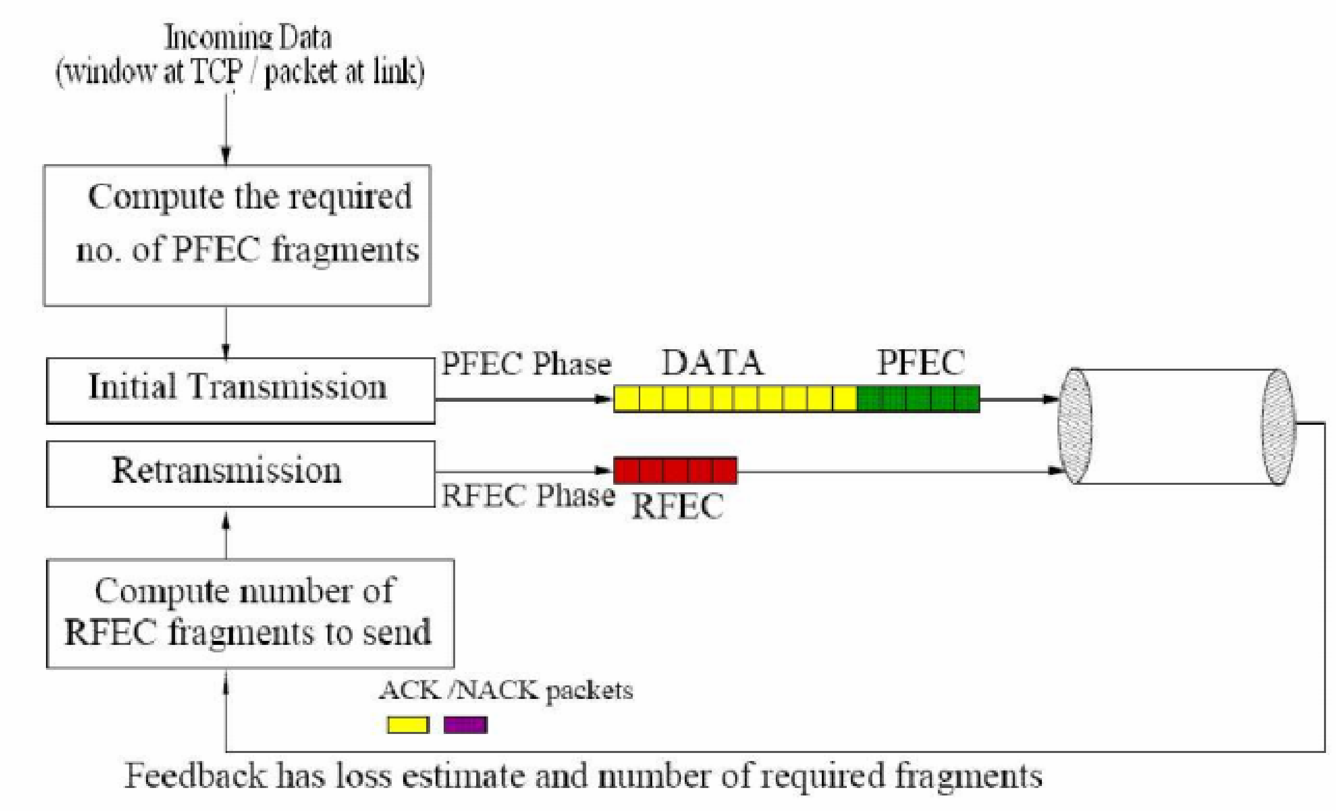
Take away: High ARQ persistence causes negative interactions with TCP.

3-way reliability tradeoff



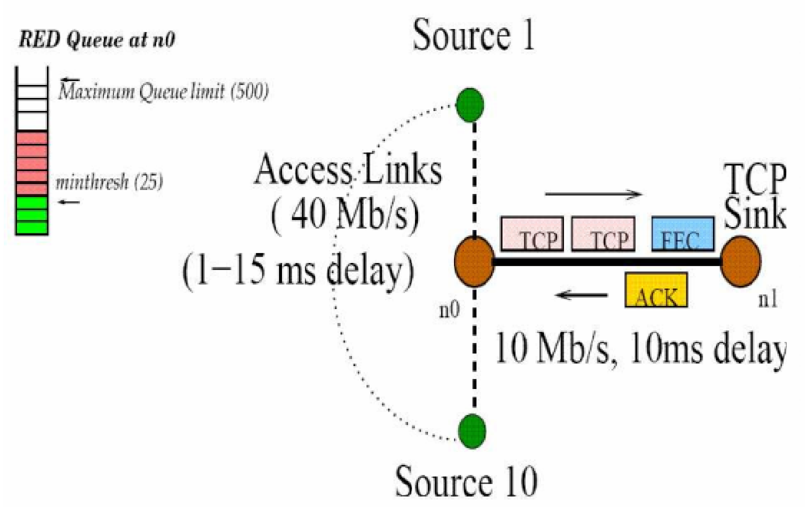
- Reliability scheme (ARQ or HARQ) faces the three way tradeoff at each layer (link or transport).
- 802.11 MAC choice: focus on zero-residual loss, at the expense of goodput and latency.
- **Can we achieve a better tradeoff?**
- If link layer aims for high goodput, subject to low latency, we could incur a (small) residual loss rate under really stressful conditions.
- In particular, it is possible to design a link-layer HARQ scheme which uses only 1 ARQ attempt (bounding latency)
  - … with minimal residual loss only under very high and very bursty raw loss conditions.
- Transport layer (TCP) has to then compensate for the residual loss rate.
  - TCP SACK is brittle beyond 5% residual loss rate exposure.
  - We have designed a loss-tolerant TCP (LT-TCP) to handle higher residual loss rates; no overhead at lower loss rates; fair with TCP at low loss rates.

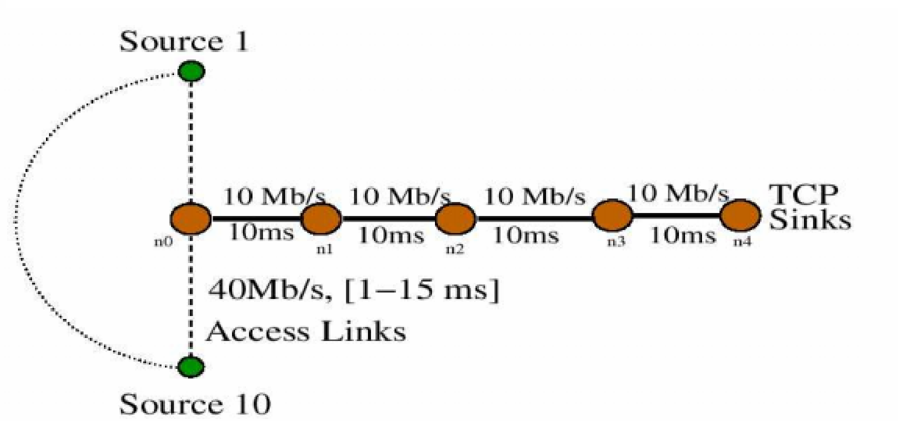
LT-TCP and LL-HARQ overview



Details of LT-TCP and LL-HARQ are omitted in this talk. Please refer to paper and references therein.

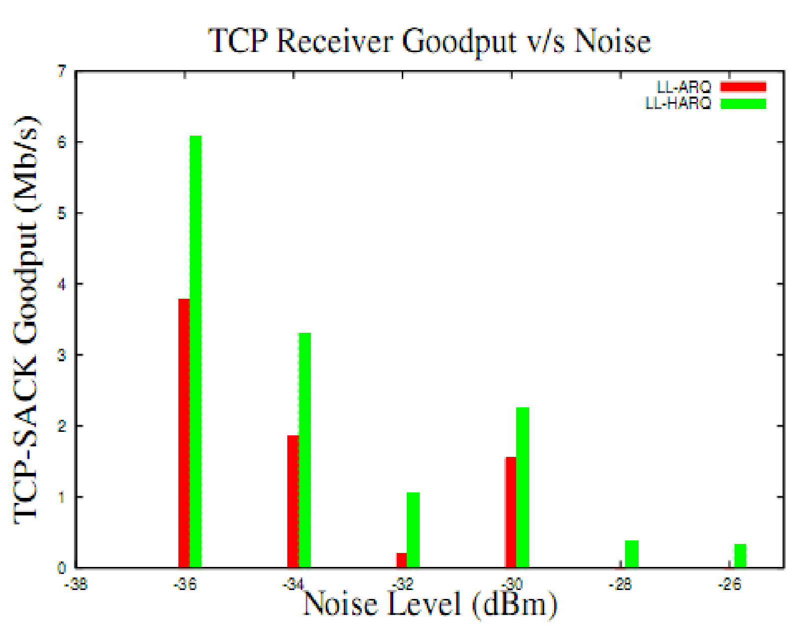
Simulation Setup

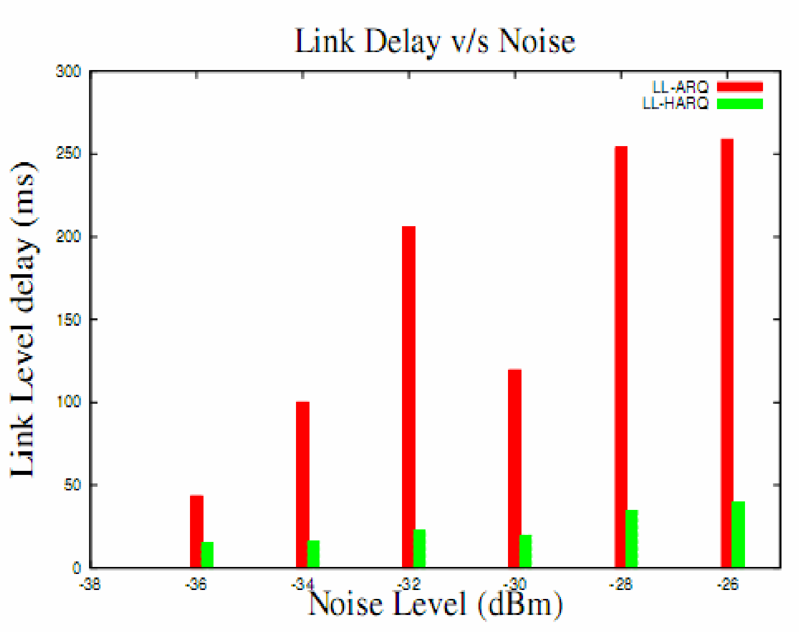




- We consider both single hop and 4-hop topologies as shown.
- We test our protocols (LT-TCP + LL-HARQ) and compare with TCP-SACK and LL-ARQ (LL-ARQ has large ARQ but no FEC).
- We use both *trace-driven* simulations using the model generated from experiments and also *synthetic-models* to stress-test the protocols.
- Simulations were run for 100 seconds and confidence intervals were measured.
- Synthetic link model has a uniform loss model with the loss rate going from 0-50%

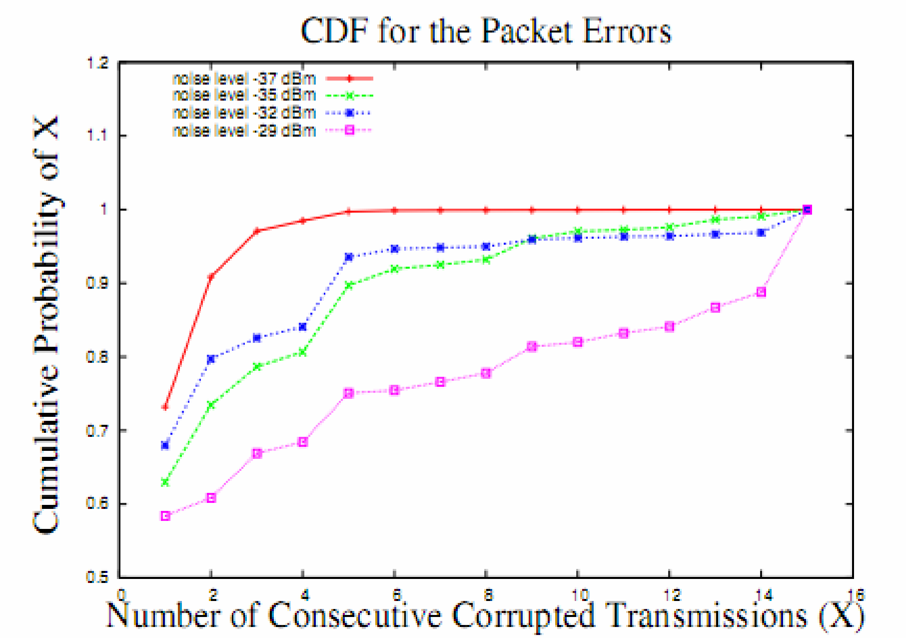
Trace-Driven Simulations (1 hop)

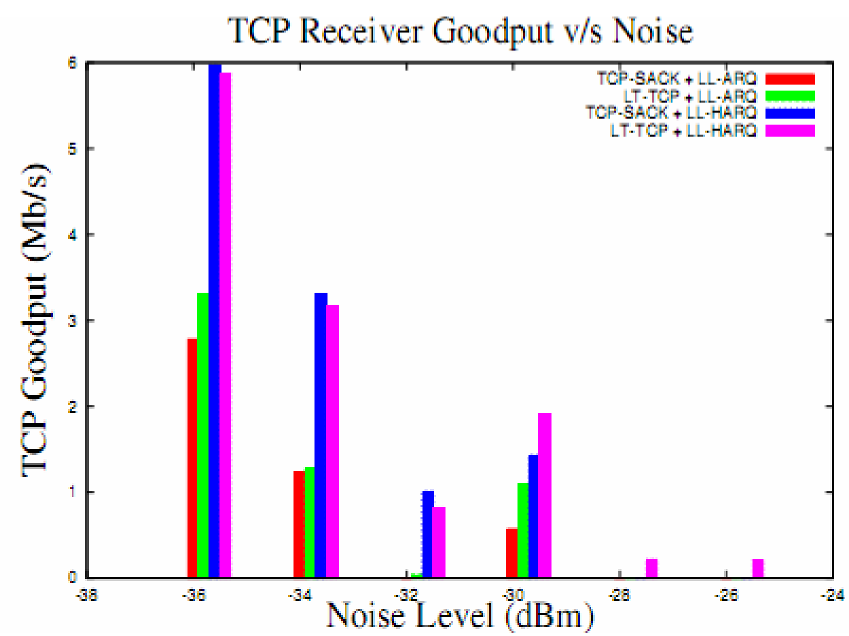




- LL-HARQ gives much better performance over LL-ARQ in terms of link good-put (leading to better transport good-put), link latency and residual loss rate.

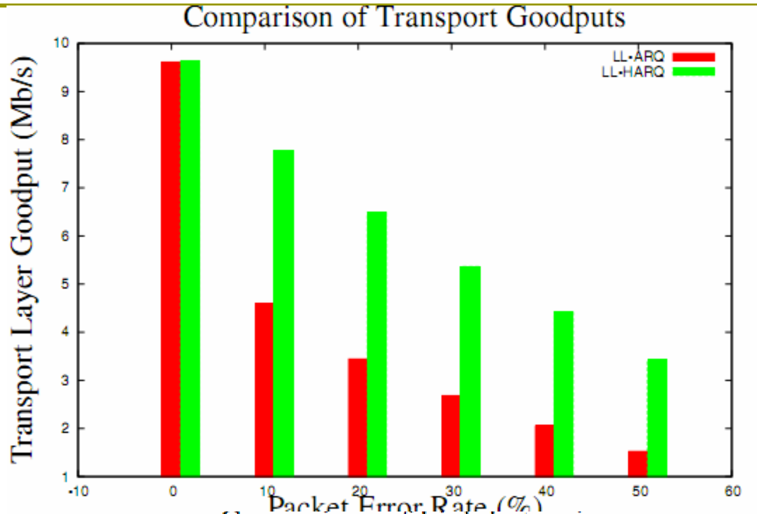
Trace-Driven Simulation (4hops)

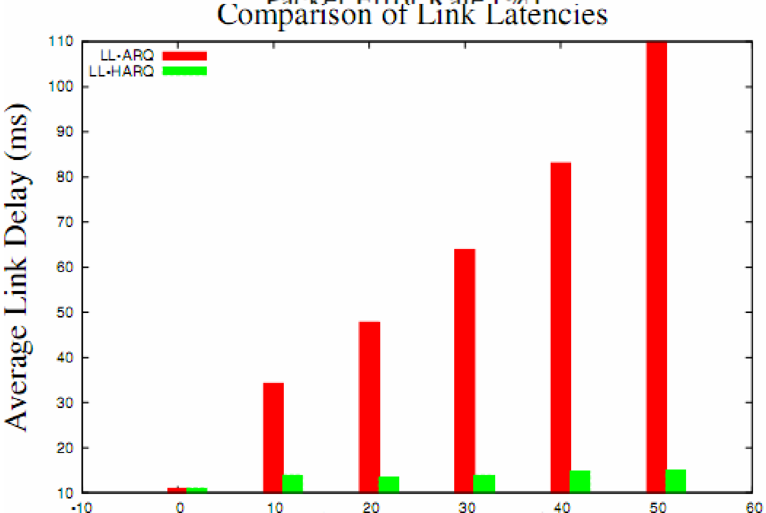




- The figure on the left shows the cumulative distribution function (CDF) for the number of consecutive packet errors for different noise levels from our gathered traces.
- For the four hop topology, LL-HARQ exports a very low residual loss rate.
- As the noise level and loss rate increase, LT-TCP mechanisms kick in to provide better performance compared to TCP-SACK.

Synthetic-Error Process (1 hop)



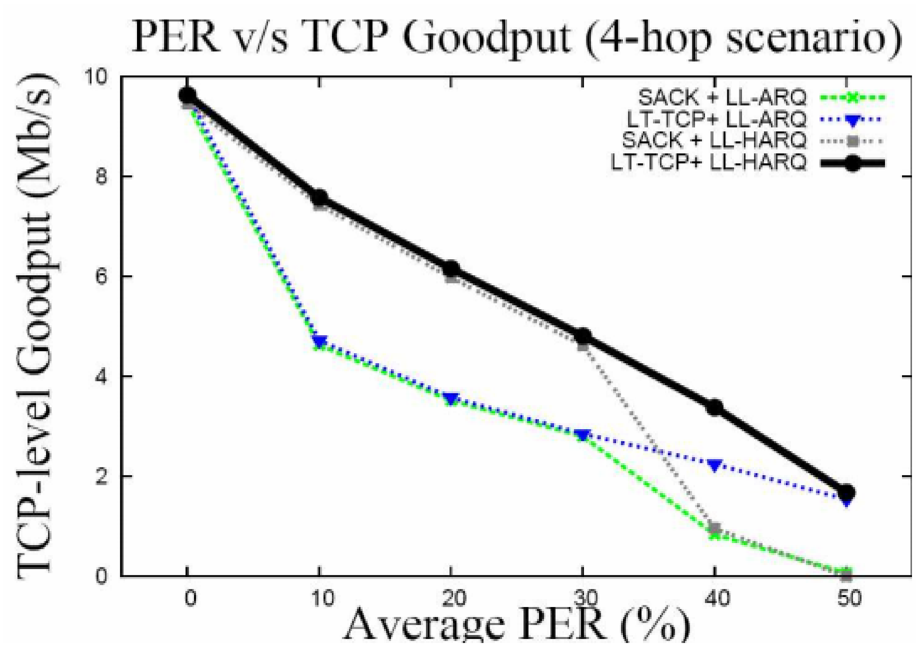


- We compare the transport layer goodput and the per-packet average link latency for the LL-ARQ and the LL-HARQ link protocols with LT-TCP as the transport protocol.
- It can be seen that the transport goodput and the latency are much better with LL-HARQ.
- Both variants export zero residual loss rate to TCP but because LL-HARQ uses only 2 transmission attempts, the obtained trade-off is much better than with LL-ARQ.

Compared to LL-ARQ, LL-HARQ provides a much better trade-off among goodput, latency and loss rate since it is able to operate well with just 2 ARQ attempts. This verifies our earlier results with the traces.

Synthetic Error Process (4 Hops)

- The transport-layer good-put for the 4-hop scenario is shown in this graph.
- With TCP-SACK as the transport protocol, the performance collapses beyond 30%.
- With LT-TCP however, the degradation in performance is linear, especially with LL-HARQ as the link protocol.
- LL-HARQ also leads to lower link latencies compared to LL-ARQ.



With increasing loss rates,
the combination of LT-TCP + LL-HARQ performs best

Conclusions

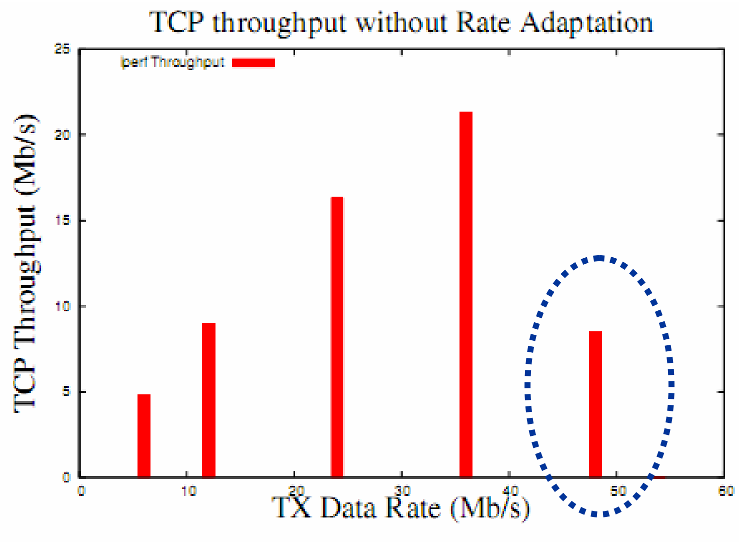
- In this paper, we studied in detail the behavior of the link and transport protocols of real wireless links (802.11a) in the presence of interference.
- We used the ORBIT test-bed to experiment with synthetic interference causing high loss rates, which in turn leads to variable link delays due to high ARQ.
- We showed how this affects the link performance and looked at how this can lead to transport layer timeouts and spurious retransmissions.
- Using the loss model developed from the experiments and a more stressful synthetic model, we compared our proposed protocols (LT-TCP + LL-HARQ) against TCP-SACK and LL-ARQ and showed performance gains & better tradeoffs.
- In conclusion, the best 3-way trade-off between loss rate/goodput and latency is obtained with LT-TCP + LL-HARQ: high performance even with very low ARQ limits.

Questions ?

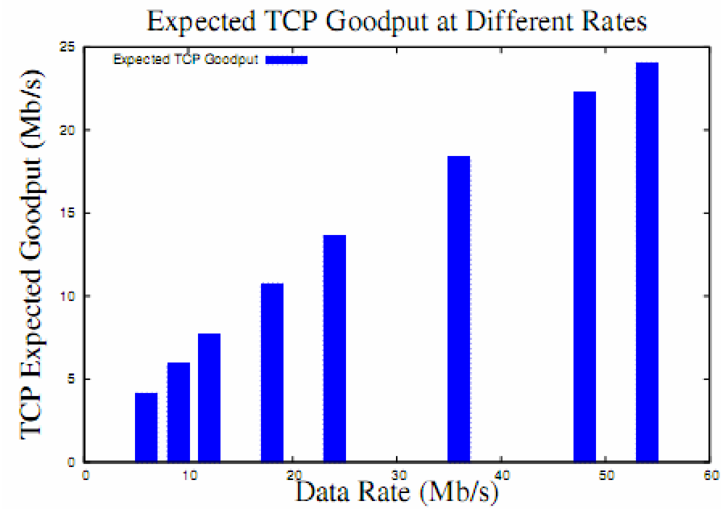
- Vijay Subramanian: subramanian.vijay@gmail.com
- K.K.Ramakrishnan: kkrama@research.att.com
- Shivkumar Kalyanaraman: shivkuma@gmail.com

- The authors would like to thank Prof. Dipankar Raychaudhuri and Prof. Ivan Seskar for use of the ORBIT facility and technical discussions.

Performance without Rate adaptation: Noise at high Tx Rates is an issue



(a) Baseline: No Rate adaptation

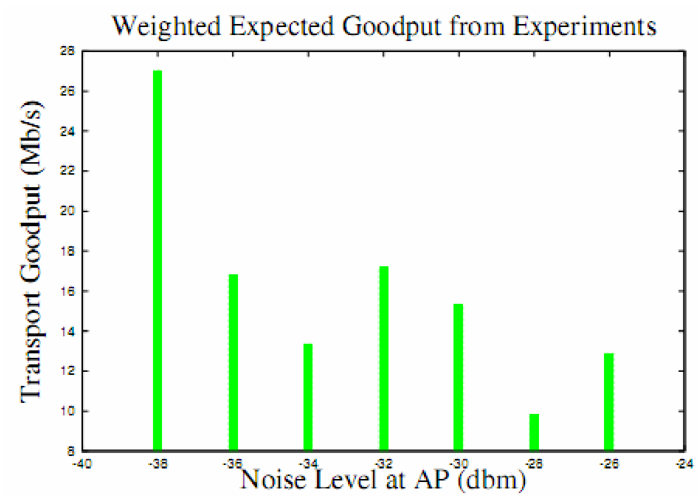


(b) Expected Goodput (no loss)

In Figure (a), without Rate Adaptation, the throughput obtained is maximum at 36 Mb/s setting at the baseline setting (no noise). Ambient noise at higher rates affects TCP throughput.

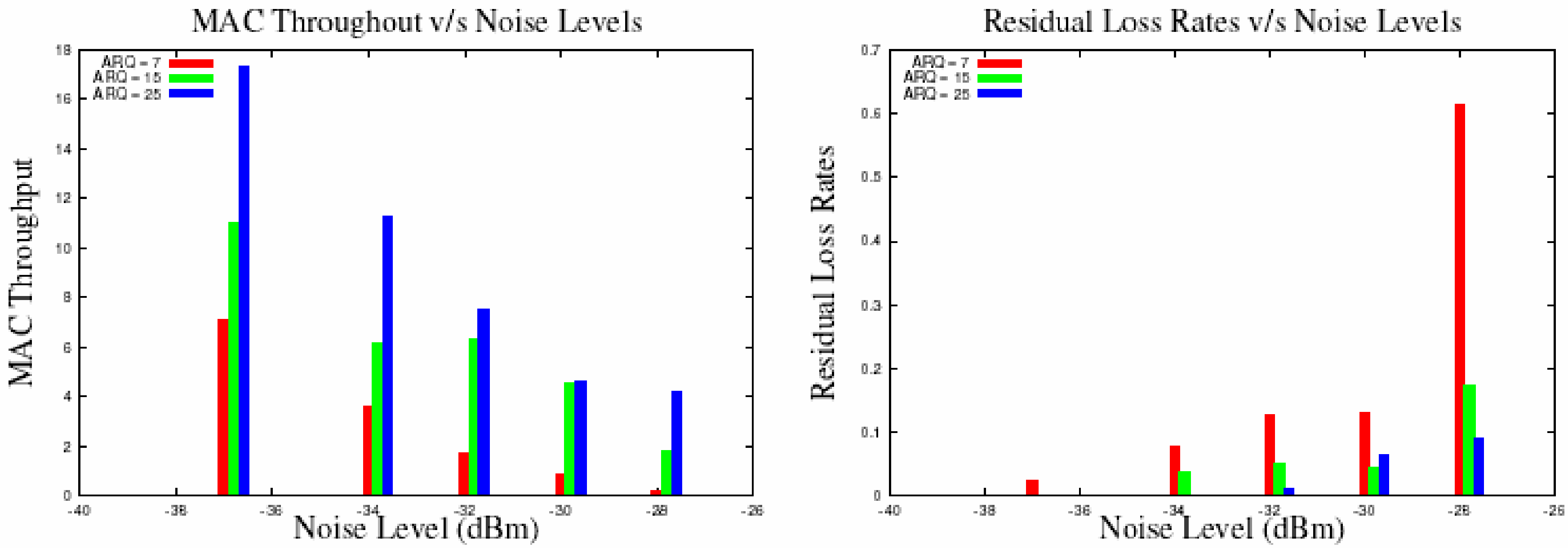
Figure (b) shows the maximum expected goodput at the transport layer for different transmission data rates. TCP data packets and acks are sent at the MAC data rate (X axis). The MAC-level acks are sent at half the data rate.

Figure (c) shows the weighted expected goodput we could have gotten for the given experiment.



(c) Expected Goodput (expt.)

MAC Throughput vs Residual Loss Rates: effect of ARQ persistence (covered in slide 6)



(a) MAC Throughput

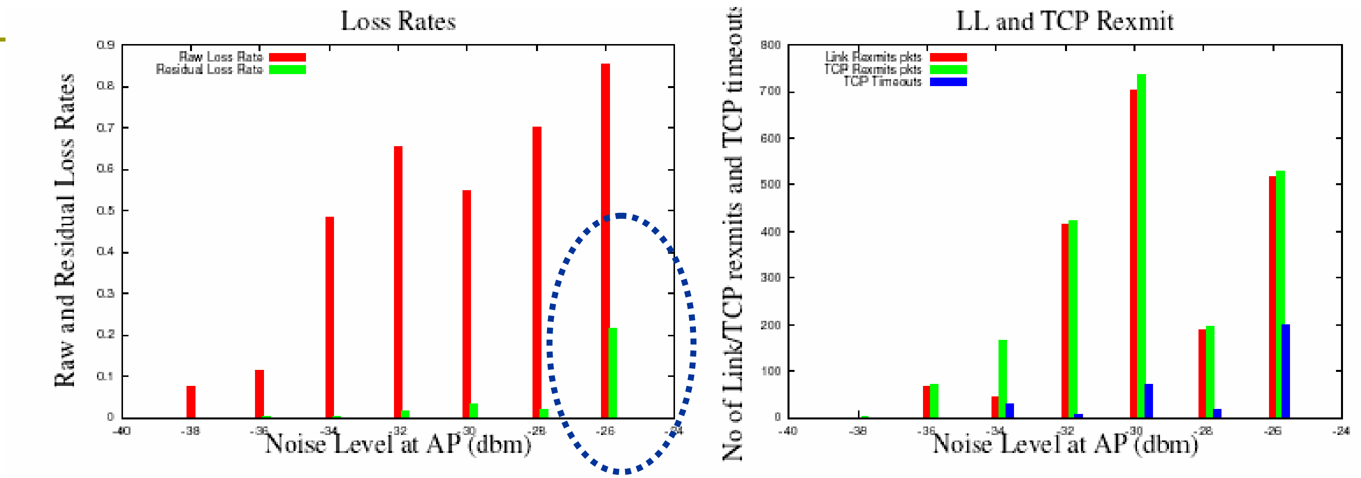
(b) Residual Loss Rates

Fig. 1. MAC-level throughput and the residual loss rates for three different ARQ persistence levels for a number of interference-scenarios.

The residual loss rate is lowered as the ARQ persistence increases.

The penalty paid is an increase in latency and lowered goodput.

Interference ↑ impacts TCP throughput <u>even w/ ARQ persistence (covered in slide 7)</u>



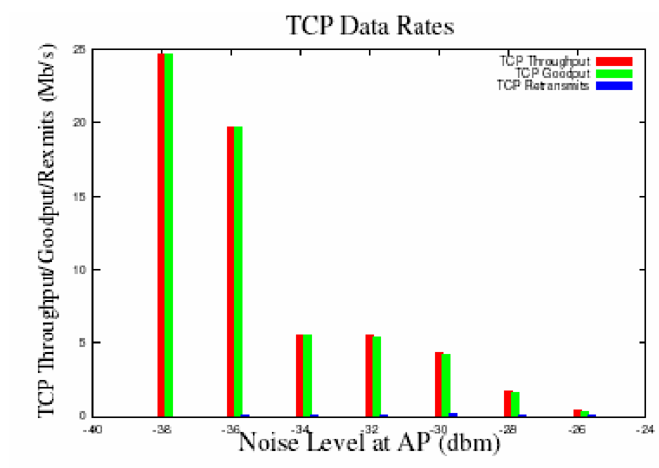
(a) Loss Rates (b) Retransmissions and Timeouts

For interference level -30dBm, **<u>Figure 4(a</u>**) shows the raw loss rate at the link layer and the residual loss rate exported to the upper layer. The raw loss rate can be quite high which leads to a significant residual loss rate seen by TCP even with a high ARQ limit.

<u>Figure 4(b)</u> shows the number of retransmitted packets at the two layers and the number of TCP timeouts.

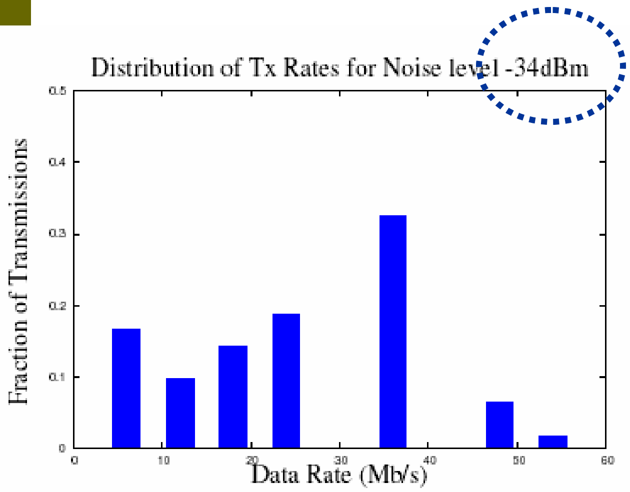
<u>Figure 4(c)</u> shows the transport-layer throughput, goodput and retransmitted bytes.

It should be noted that as the noise level increases, the performance falls sharply.

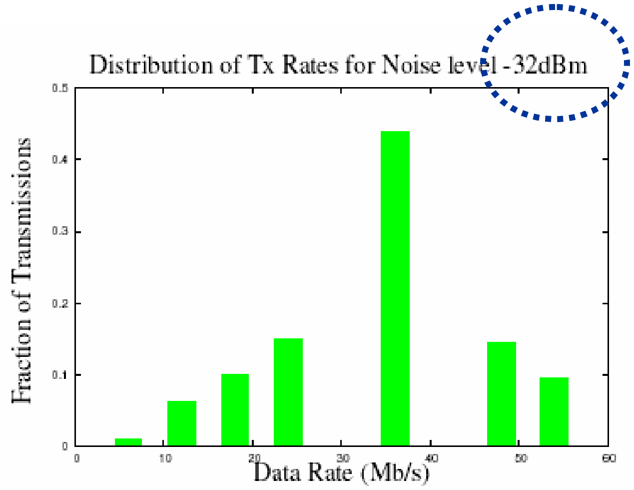


(c) Statistics of TCP performance

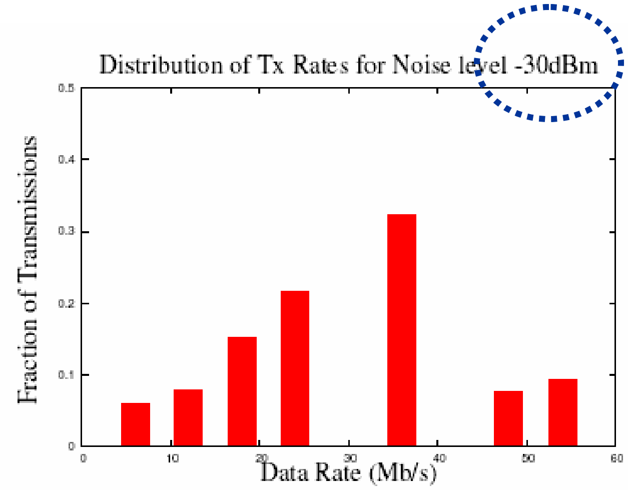
Distribution of Tx rates with different interference levels



(a) Noise level = -34 dBm



(b) Noise level = -32 dBm



(c) Noise level = -30 dBm

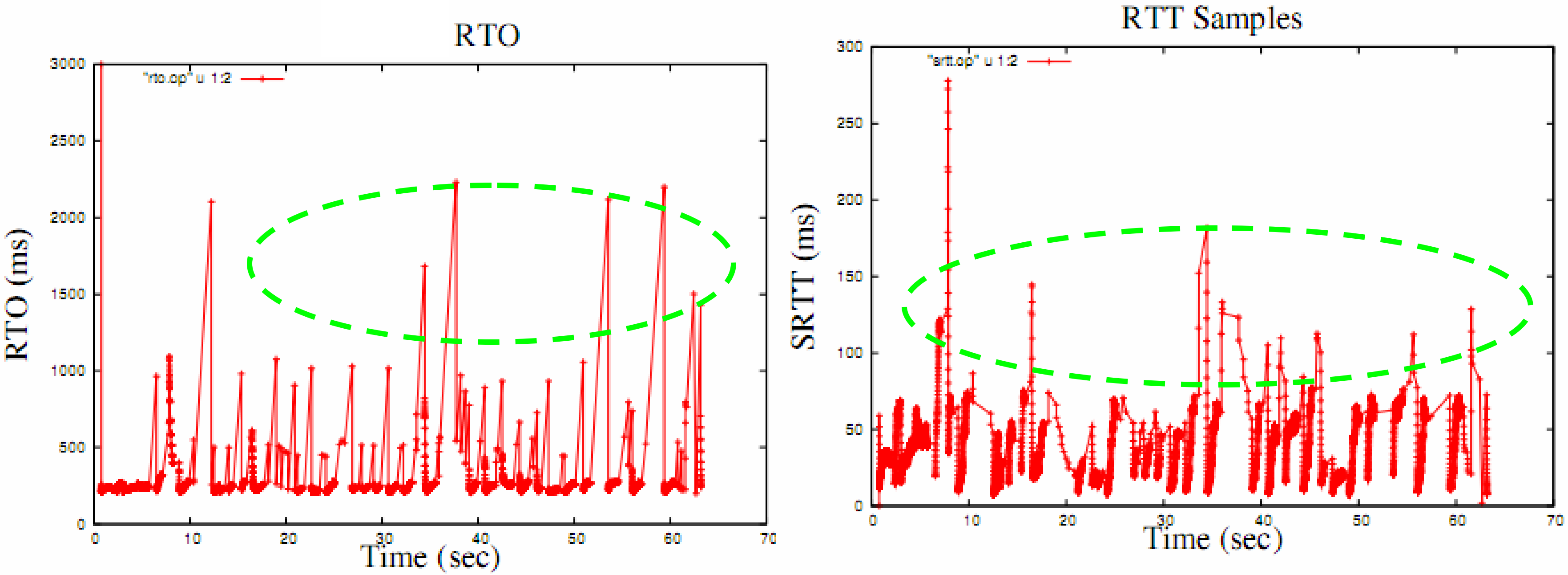
→ ↑**interference**

Distribution of transmission rates for three different interference-levels.

From this distribution and from the expected TCP goodput at each data rate, …

… we can compute a weighted average of the expected TCP goodput for each experiment scenario.

TCP RTTs for -30 dBm interference (covered in slide 9)

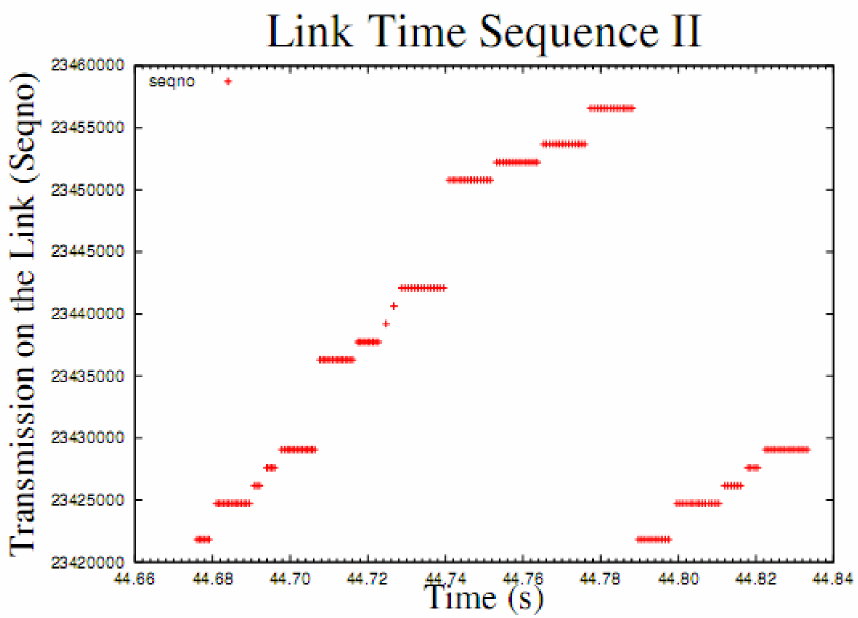


(a) Retransmission Timeout

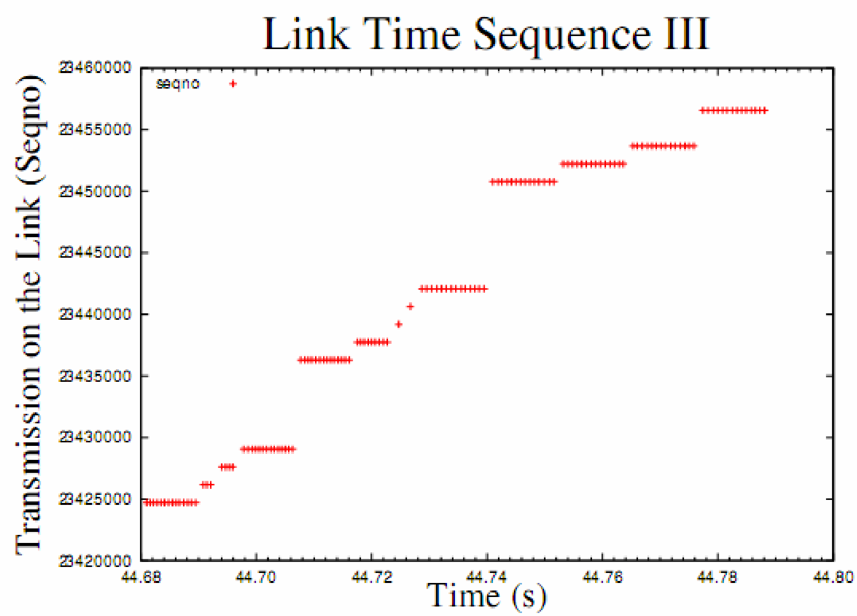
(b) Smoothed RTT

- For noise level of -30 dBm, we look at the behavior of the RTO and RTT samples.
- Note the spikes in the RTO that are caused due to the variation in the link latency caused by high ARQ.

Example: Spurious Timeout (covered in slide 10)



(b) Time Interval: 44.66-44.84 s



(c) Time Interval: 44.68-44.80 s

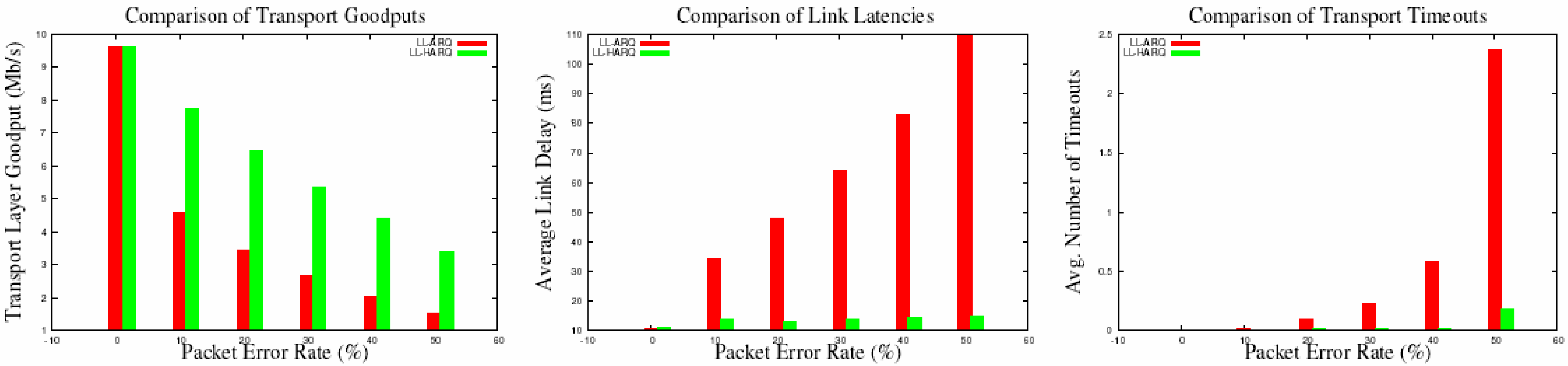
- We see that the packet with sequence no *23421817* is sent thrice (attempts 2 and 3 are shown).
- Attempt 2 is a genuine retransmission but since this packet is stuck waiting to be sent behind other packets due to large ARQ at link, TCP times out and sends it again (third attempt).
- Both attempts 2 and 3 are successful though attempt 3 is a redundant attempt.

- High ARQ persistence causes negative interactions with TCP.

Table 1: LL-HARQ performance

10 Flows, Single-link	ERROR RATE					
PARAMETER	0 %	10 %	20 %	30 %	40 %	50 %
Goodput (Mb/s)	9.96	8.05	6.71	5.61	4.58	3.59
Throughput (Mb/s)	9.99	9.98	9.99	9.99	9.98	9.99
Residual Loss Rate (%)	0.00	0.00	0.00	0.00	0.00	0.00
Avg. Latency (ms)	10.97	13.83	13.26	13.80	14.67	15.05
PFEC Sent (Mb/s)	0.02	1.48	2.90	3.77	4.60	5.40
RFEC Sent (Mb/s)	0.00	0.39	0.36	0.59	0.80	0.98
PFEC Wasted (Mb/s)	0.02	0.61	1.04	1.03	1.02	1.00
RFEC Wasted (Mb/s)	0.00	0.26	0.23	0.33	0.38	0.39

LL-ARQ vs LL-HARQ: performance (slide 16)



(a) Transport Goodput (b) Link Delay (c) Timeouts

Synthetic Error Process (1 hop): We compare the transport layer goodput and the per-packet average link latency for the LL-ARQ and the LL-HARQ link protocols with LT-TCP as the transport protocol.

It can be seen that the transport goodput and the latency are much better with LL-HARQ.

Both variants export zero residual loss rate to TCP but because LL-HARQ uses only 2 transmission attempts, the obtained trade-off is much better than with LL-ARQ.

TCP SACK vs LT-TCP

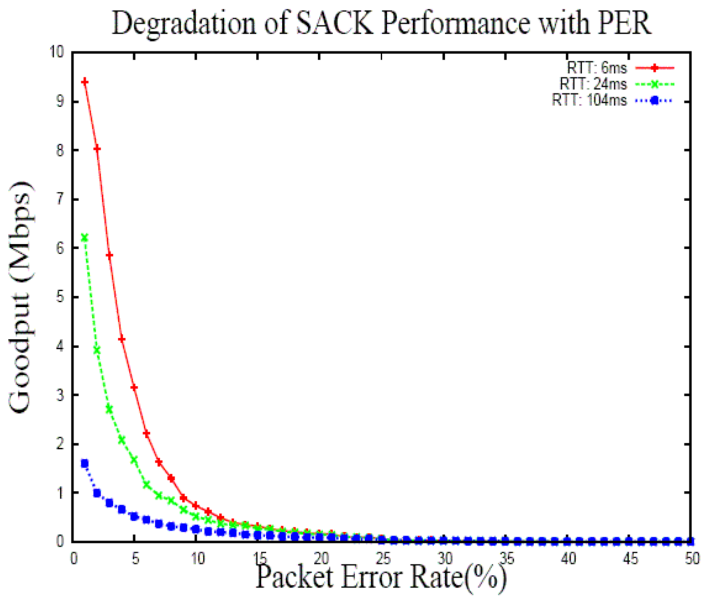


Fig. 1. TCP-SACK Degradation with Increased Erasure Rate and RTT (Uniform Loss Probabilities, 10 Mb/s Capacity, 1 flow)

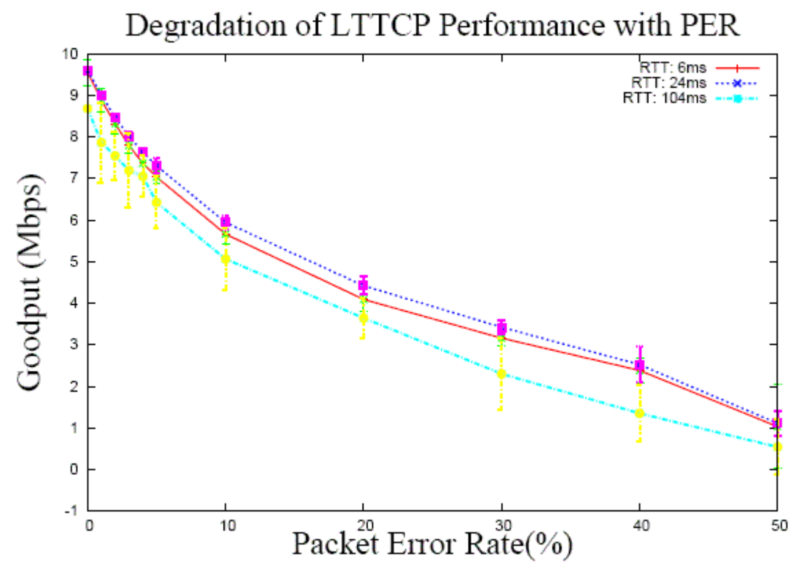


Fig. 4. LT-TCP performance with Increased Erasure Rate and RTT (Uniform Loss Probabilities, 10 Mb/s Capacity, 1 flow)

- SACK: Exponential falloff in performance with PER
- 5%+ PER
- 100 ms+ RTT

- LT-TCP: Linear falloff in performance with PER
- Scales well for up to 50% PER
- Scales well with RTT (100ms)