Impact of Interference and Capture Effects in 802.11 Wireless Networks on TCP

Vijay Subramanian,
ECSE Dept, RPI,
subrav@rpi.edu

K.K. Ramakrishnan,
AT&T Labs Research,
kkrama@research.att.com

Shiv Kalyanaraman,
ECSE Dept, RPI,
kalyas@rpi.edu

Lusheng Ji,
AT&T Labs Research,
lij@research.att.com

***Abstract*—**
As broadband wireless channels become common, the performance of TCP over end-to-end paths containing such links is important. TCP SACK suffers substantially when residual packet error rates increase beyond a value of about 1% - 5 % (especially for longer RTTs). Recently we have proposed improvements to TCP (called LT-TCP) to make TCP loss-tolerant in heavy and bursty erasure environments. However, real world wireless systems do not just present bursty random loss patterns to the transport layer. The PHY, MAC and transport layers all respond to errors, interacting in myriad ways. In this paper we focus on one underlying source of packet erasure, namely interference in 802.11 environments (from Bluetooth and co-channel interferers), and the resulting interaction between the MAC and transport layer mechanisms. We argue that because MAC layer mechanisms cannot fully mitigate the interference problem and tend to interpret interference as noise, they should not aggressively respond with techniques like rate-adaptation. Such aggressive responses lead to poor scheduling performance at the MAC layer (e.g., well-known unfairness and capture effects) and limit mitigation opportunities at the transport layer. While LT-TCP still performs better than TCP SACK under such conditions, reconfiguration of MAC layer mitigation options and increased buffering leads to significantly improved end-to-end performance.

I. Introduction

The rapid deployment of broadband wireless systems such as 802.11 Wireless LANs (WLANs), 802.16 wireless broadband and neighborhood area wireless networks raises expectations of high end-to-end performance.

However, TCP performance is known to degrade on wireless links due to packet corruption being misinterpreted as congestion losses. Figures 1 and 2 show that the degradation is particularly worse with increase in loss rates, burstiness of losses and round trip times (RTTs). We have recently proposed improvements in TCP, called LT-TCP [14], to support high loss tolerance. Figures 1 and 3 show that LT-TCP does not suffer such rapid degradation. The drop in performance is more graceful due to its resilience at higher error rates, and reduced sensitivity to RTT and burstiness effects.

In this paper we dig deeper into the sources of residual erasures in networks with 802.11-based access links or last hop links. In multi-user enterprise/campus LAN environments, a dominant source of erasures is interference (and not channel impairments or noise). Though the purpose of the 802.11 MAC layer is to coordinate multiple user access, it cannot eliminate interference. We therefore consider *residual* interference in ISM open spectrum bands due to nodes operating with a different technology (e.g., Bluetooth) or due to asymmetric co-

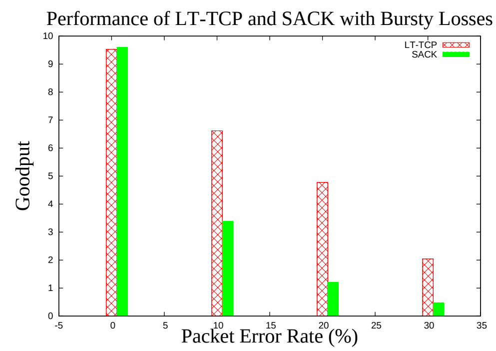

Fig. 1. Comparative Performance of LT-TCP vs TCP-SACK with Bursty Losses (Gilbert Model, 10Mb/s Capacity, 10 flows, 1000s runs, Avg of 6 runs).

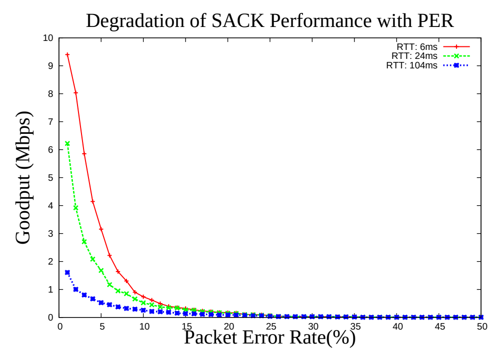

Fig. 2. TCP-SACK Degradation with Increased Erasure Rate and RTT (Uniform Loss Probabilities, 10 Mb/s Capacity, 1 flow)

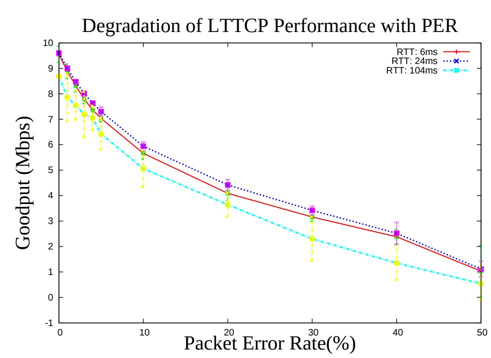

Fig. 3. LT-TCP performance with Increased Erasure Rate and RTT (Uniform Loss Probabilities, 10 Mb/s Capacity, 1 flow)

channel interference in WiFi leading to capture effects (e.g., hidden nodes in WiFi networks). Our focus is on the interaction between mechanisms at the 802.11 MAC layer and the transport layer in response to such interference-induced packet corruption. In particular, we ask: "*Can MAC and transport protocols effectively deliver a significant proportion of the raw bit-rate available at the physical layer to the application in a multi-user environment prone to interference effects?*"

We show that transport-level erasure mitigation opportunities can be *significantly limited* by PHY level mechanisms (like slow-rate preamble), and aggressive MAC layer mechanisms such as rate-adaptation and persistent ARQ. These mechanisms were originally designed with channel impairments and noise in mind. Since interference is indistinguishable from noise, these mechanisms tend to backfire and also severely limit transport layer mitigation mechanisms. In particular, rate-adaptation mechanisms tend to overreact leading to poor channel sharing and *increased* vulnerability to interference.

Even if rate adaptation were turned off, MAC layer reliability mechanisms interact negatively with hidden terminal interference and capture effects. The reasons are as follows:

In the years to come, we envision neighborhood areas that will be serviced by multiple WLAN systems. These coverage areas or cells will be small so that the link quality and capacity is high (from a noise perspective). Moreover, to provide good performance (despite attenuation from walls, floors etc), cells are designed for the worst case and may be as small as 30 m radius. However, in typical environments this means that client nodes can associate with multiple APs on the same frequency. This causes a high incidence of hidden node problems with increased adoption and usage of WiFi. In the worst case, it causes severe capture effects. The impact of capture effects can be mitigated by the use of larger buffers (and ECN thresholds) to absorb the burstiness during capture and use of ARQ persistence (drop fewer packets because ARQ attempts to transmit each packet longer).

The RTS/CTS mechanisms which were designed to mitigate the hidden node problem are rarely turned on in practice. The reason for this lies in the fact that the overhead incurred is high for the amount of data sent (RTS/CTS are sent at 1 Mbps). These factors together contribute to a potentially large *raw* packet error rate. To compensate, WiFi LANs set the number of MAC-level ARQ retransmission attempts to 7 when RTS/CTS is turned off.

Link-layer ARQ is known to be helpful in WiFi LANs: higher ARQ persistence does decrease residual loss rates and increase resilience to capture effects. The link-level transmission (assuming reduced rate-adaptation) and propagation times are small enough in LANs to allow multiple retransmission attempts. However, the utility of persistent ARQ is affected negatively due to delays induced by exponential timer back-off between successive ARQ retries. Lower latency demands by emerging applications like VoIP-over-WiFi (a.k.a cell-Fi) and potential interaction with TCP timers limits the safe number of ARQ retries without triggering negative performance interactions. Longer ARQ retries do not help in really long capture periods (e.g., beyond 0.5 s capture) because spurious timeouts occur at the TCP level.

The persistent ARQ process also leads to increased per-packet MAC-level overheads and increased vulnerability to further interference because the preamble of all packets (24 bytes) and the entire MAC-level acks (48 bytes) for every ARQ attempt are sent at 1 Mbps. Since TCP acks also generate multiple ARQ retries and MAC-level acks, the useful TCP good-put with 1500 byte segments on a fully utilized 11 Mbps link after subtracting out all these per-packet MAC-level overheads is less than 55% if ARQ is done only once (i.e., no interference). The maximum goodput percentage drops rapidly with reduced segment sizes or increased ARQ persistence.

While LT-TCP still performs better than TCP SACK under such interference-induced multi-layer interactions, reconfiguration of a few key MAC layer mitigation options leads to dramatically improved end-to-end performance.

In summary, our recommendations include:

- Reconsider aggressive rate-adaptation in 802.11 and *de facto* rate adaptation for MAC level acknowledgements. Preamble can remain at lower rates for safety even though it costs over 3 Mbps.
- Consider limit 802.11 ARQ retries to 3 (instead of 7) to limit per-packet overheads.
- LT-TCP improvements to TCP-SACK at the transport layer and ECN at bottleneck queues.
- Use larger buffers and set ECN-triggering thresholds higher to survive capture effects.

Though it is too late to change 802.11b/g standards, we hope this analysis will inform the debate in 802.11n, WiMax, and help WiFi network operators better configure existing equipment in enterprises or hot-spots.

The rest of the paper is organized as follows. Section II discusses the related work. Section III provides an overview of the Loss-Tolerant TCP scheme. Section IV discusses the models for packet corruption considered in this paper and presents the simulation environment and results. Section VI concludes the paper.

II. Related Work

The feasibility of transport-layer FEC was initially explored by Rizzo [12]. Baldatoni et al [10] proposed a version of TCP with FEC (but without adaptivity) that works with small error rates. TCP Westwood [13] uses an output rate estimate for congestion control and survives small error rates (under 5 %). Krishnan et al [9] observe that distinguishing congestion loss from erasure loss is not sufficient to get performance gains. Loss-Tolerant TCP (LT-TCP), a scheme proposed by the authors in [14] is designed to make TCP operate under *high loss regimes* in ECN environments [8] by adding adaptive MSS and FEC mechanisms.

Recent studies have examined the impact of interference in wireless LAN environments. Golmie et al study the performance of Bluetooth Access Control Layer in [6] operating in close proximity to an 802.11 WLAN system. The probability of collision between a Bluetooth transmission and WLAN transmission is derived and is found to be significant. Golmie et al [5] evaluate the effect of mutual interference on the performance of Bluetooth and IEEE 802.11b systems. The authors report significant packet error rates for WLAN transmissions given interference from Bluetooth.

Shellhammer [16] derives the probability of an 802.11 packet error in the presence of interference from Bluetooth. Reference [2] presents a model of the interference that IEEE 802.11 transmissions may experience because of either a Bluetooth call or voice link. The paper also proposes a traffic shaping technique to the Bluetooth flow that can reduce the impact of interference.

Rate adaptation is a technique used by 802.11 a/b/g wireless devices to make use of multi-rate capabilities in response to SNR degradation and packet erasures. Lacage et al [11] and Sadeghi [15] discuss 802.11b's Auto Rate Fallback (ARF) and Receiver Based Auto Rate (RBAR) and propose new rate-adaptation mechanisms. TCP level impact was not considered.

TCP performance is studied by Fu et al [4] in a stationary multi-hop 802.11 wireless network using IEEE 802.11 channels. They show the existence of an optimal TCP window size that is tied to the hop count in the multi-hop path. Moreover, since the link-level drop probability is not enough to keep the TCP window tied down to the optimal window size, a link layer scheme called Link RED is proposed to tune the packet dropping probability to stabilize the TCP window size around the optimal value. Adaptive pacing at the link layer is proposed to coordinate channel access.

ElRakabawy et al [3] also observe that the ideal TCP window is tied to the hop count. However, instead of making changes on the link-layer, a TCP mechanism called Adaptive Pacing is proposed that operates at the TCP layer by estimating 4-hop propagation delay and the coefficient of variation of recent RTT samples.

In contrast to these works, our proposal considers heavy packet erasure rates and multi-layer interactions. We propose revised parameter settings at the MAC layer and a new LT-TCP proposal at the TCP layer.

III. Loss-Tolerant TCP Overview

Loss-Tolerant TCP (LT-TCP) is an enhanced version of TCP-SACK with ECN [14]. It uses the following building blocks.

ECN-Only: Congestion response only to ECN, since it is the definitive signal of congestion in ECN-enabled networks.

Per-Window Loss Rate Estimate (E) Per-window loss fraction samples are averaged using an exponential weighted moving average (EWMA) with parameter 0.5.

$$E = 0.5 \times new_l + 0.5 \times E \qquad (1)$$

Proactive FEC: The number of FEC packets per window (P) used (i.e., Proactive FEC) is a function of the erasure estimate, i.e., $P = f(E)$. The TCP maximum segment size (MSS) is adjusted to allow one or more FEC packets per window (see below).

Adaptive MSS and Granulation: Granulate the congestion window to have at least G packets, subject to limits of a minimum and maximum MSS (MSS_{min} and MSS_{max}). Further, as the window increases (in bytes), the MSS is increased in steps of $MSS\_stepsize$ (200 bytes) provided it does not decrease the window granulation. If necessary, MSS is adjusted to accommodate the proactive FEC $P = f(E)$. MSS is also halved (subject to minimum MSS constraints) when the window is halved after response to ECN.

Reactive FEC: Since proactive FEC may be underestimated, the sender transmits R reactive FEC packets where R depends on the currently estimated Loss Rate, the number of Proactive FEC packets sent for this block and the number of *holes* left to be filled to completely decode this block. The precise reactive policy is specified in the technical report [17].

IV. IEEE 802.11b Simulation Model

Among all the flavors of the IEEE 802.11, we choose 802.11b DSSS (2.4- 2.475 GHz using 22 MHz bandwidth). The RTS/CTS contention avoidance mechanism is turned off. MAC level ARQ has a persistence of either 3 or 7 (i.e. 2 or 6 retries). With the RTS/CTS mechanism turned off, the number of attempts per packet will be *ShortRetryLimit* which has a default value of 7. Random exponential back-off is used for each retry.

The IEEE 802.11b supports four data rates: 1, 2, 5.5, and 11 Mbps and multi-rate operation to combat slow fading. Every packet, ack or MAC level ack (MAC-ack) has a preamble of 24 bytes sent at the basic rate 1 Mbps. The implementation and decision basis to change the rate are usually proprietary though some general heuristics are known [11].

However, the implicit assumption is that lowering the rate will decrease the probability of packet error. This is true if the causes of packet corruption involve link impairments alone. However, if the cause of packet corruption is *interference*, rate adaptation will not help if the signal strength is high enough. In fact, lowering the rate will expose the packet to higher probability of error since the packet is "in the air" for a longer time. In other words, rate adaptation is effective in dealing with propagation losses and not with interference losses. We demonstrate this effect in the next section.

A. Cross-System Interference Model: 802.11b with Bluetooth Interference

Among various wireless technologies which may produce cross-system interference for 802.11 systems, we choose Bluetooth. Bluetooth headsets for devices such as cellphones are popular and concurrent Bluetooth and WLAN sessions are likely.

Bluetooth wireless links are short range (0-10 m), medium data rate (1 Mbps) operating in the 2.4 GHz ISM spectrum [7]. HV1, HV2 and HV3 are three packet formats that are used to transmit 64 Kbps voice over Synchronous Connection-Oriented (SCO) links. Typically, Bluetooth headsets operate in the Class 2 mode which is designed for communication up to 10 m with transmission power of 2.5 mW. If the Bluetooth transmitter is close to the WLAN receiver, it can cause WLAN reception bit errors. We adopt a simple Bluetooth interference model: within the duration of a WLAN packet reception, if a Bluetooth hop falls into the WLAN channel frequency range, the WLAN packet is corrupted.

The effect of Bluetooth is modelled through the probability of WLAN packets being corrupted by Bluetooth transmissions occurring near the WLAN receiver. We now begin to derive the

WLAN packet corruption probability with a Bluetooth interference source close by (similar to the approach in Shellhammer [16] and Golmie [5]).

A typical Bluetooth voice call uses a full-duplex 64 kbps channel. The probability of a Bluetooth packet being on a WLAN channel is dependent on the Bluetooth frame format used. We assume that pure SCO packets (as opposed to hybrid DV (Data-Voice) packets) are used to carry the packetized voice data. For two-way traffic, the three formats HV1, HV2, and HV3 occupy 12, 6 and 4 out of every 12 slots respectively giving slot utilization factors λ of thus 1.0, 0.5 or 0.3 respectively.

The following notations are employed.

Parameter	Meaning
λ	Bluetooth Utilization Factor
PCR	Packet Corruption Rate for WLAN
PER	Residual Error rate on the link.
p	Probability of collision in a specific Bluetooth slot.
N	Minimum number of Bluetooth slots that overlap the WLAN transmission.
K	ARQ Persistence.

TABLE I
BLUETOOTH PARAMETERS

We consider an 802.11b channel which occupies 22 MHz of the 79 MHz Bluetooth band. Thus the probability that a Bluetooth packet hops into a WLAN channel is $\frac{22}{79} = 0.27$. We determine the length of each WLAN transmission and the number of Bluetooth slots that the duration of transmission covers. The transmission time of the WLAN packet can be expressed as: $transmission\ time = \frac{packet\ size}{Data\ rate}$. Bluetooth has a dwell time of 625 microseconds. Thus the minimum number of *complete* slots that the WLAN packet covers is $N = \lfloor \frac{transmission\ time}{dwell\ time\ of\ Bluetooth} \rfloor$. The actual number may be one more than this depending on the relative positioning of Bluetooth slots and the WLAN packets. Conservatively, we assume that it is N. For simplicity, we disregard partially overlapping slots. The probability of packet corruption is then given by $PCR = 1.0 - (1.0 - p)^N$ where $p = 0.27 \times \lambda$.

Because of the retransmission mechanism of WLAN, not every "raw" packet loss is visible to higher layers. What is exposed to higher layers is what we call "residual loss rate", which is defined as the packet loss rate after link-level retransmissions have taken place. If the link-layer ARQ persistence is K, then the residual error rate is given by $PER = PCR^K$ which is the probability that the packet was corrupted on all K attempts.

B. Simulation Results: Cross-System Interference with Bluetooth

We compare the performance of LT-TCP and SACK over WLAN with and without Rate Adaptation and affected by Bluetooth interference. Since rate adaptation algorithms used in real systems vary from device to device, we used a simple algorithm wherein the rate is reduced when the sender suffers from successive transmission failures and vice-versa. Tables II and III show the performance of the transport protocols under these conditions. It is clear that operating at the highest data rate is optimal *even in the presence of large error rates* since the packet is exposed to interference for a shorter duration. The results show that operating at 11Mbps enables us to obtain a MAC-level throughput that is close to the maximum obtainable. At lower data rates, repeated packet losses lead to residual losses that lead to timeouts at the TCP level. This limits the flow of data and performance drops drastically. This effect is more pronounced for HV1 interference.

Single-source (HV3)	LT-TCP		SACK	
PARAMETER	Without RA	With RA	Without RA	With RA
TCP Good-put (Mbps)	3.74	0.06	2.32	0.005
95% CI for Good-put	[3.59,3.88]	[0.05,0.07]	[2.24,2.41]	[0.0, 0.01]
Number of Timeouts	0	42	0	12.6
MAC Throughput (Mbps)	5.22	0.54	3.09	0.01

TABLE II
PERFORMANCE WITH AND WITHOUT RATE ADAPTATION IN THE PRESENCE OF HV3-ENCODED BLUETOOTH VOICE CALLS.

Single-source (HV1)	LT-TCP		SACK	
PARAMETER	Without RA	With RA	Without RA	With RA
TCP Good-put(Mbps)	2.83	0.006	0.40	0.0002
95% CI for Good-put	[2.61,3.04]	[0.002,0.01]	[0.37,0.43]	[0,0.0003]
Number of Timeouts	0	51.3	13.6	13.8
MAC Throughput(Mbps)	5.25	0.08	0.65	0.004

TABLE III
PERFORMANCE WITH AND WITHOUT RATE ADAPTATION IN THE PRESENCE OF HV1-ENCODED BLUETOOTH VOICE CALLS.

Rate adaptation was designed to counter *weak signal strength* and provide improved *spatial coverage* for WLAN networks and not to mitigate interference. However, when the source of error is strong interference which affects all data rates equally, rate adaptation is counter-productive. Since we expect future wireless cells to be compact with good-coverage, we need link-layer mechanisms to be robust against interference and not just propagation errors. Our suggestion is to moderate rate adaptation off and let higher layers tackle residual errors. The rest of the paper assumes that rate adaptation has been turned off.

C. Co-channel Interference Model: Hidden Nodes in Remote Cells

In this section, we assume rate adaptation is turned off and cells operate at 11 Mbps and examine issues with co-channel interference.

Consider the effect of operating different WiFi cells in close proximity in the same frequency channel. Cells more than one cell-hop away typically reuse the spectrum. As mentioned earlier, due to worst-case design constraints, cells could have radii of as low as 30 m. While this design improves SNR when there is no interference, it backfires when there is significant amount of co-channel interference.

The packet corruption due to interference is modeled as follows. While a receiver is receiving a frame, another transmission occurs in its vicinity and at the receiver location the new

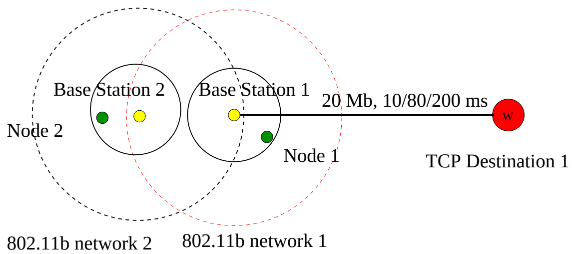

Fig. 4. Simulation Setup for Co-Channel Hidden Node Interference.

transmission's observed signal strength exceeds a threshold, the new transmission corrupts the frame currently being received by this receiver. Interference from multiple sources can also aggregate. It is enough to corrupt a few bits of a packet to render the whole packet useless. However, at high bit rates (11 Mbps), even 1500 byte packets are short. Further, MAC overheads increase with the number of packets (irrespective of the number of bytes in packets). Therefore it is better to send larger packets if the bit rate is high (and rate adaptation is turned off).

We assume the transmission range to be 250 m and an interference range to be 500 m. Note that if nodes are separated more than 250 m, the RTS/CTS mechanism may not be enough to prevent hidden node interference. The actual patterns of corruption depend upon relative location of nodes in cells and patterns of traffic from the interferer and whether the interferer sees reciprocal interference. Also the impact of losing TCP packets vs losing TCP acks is different at the transport layer (acks are cumulative; packets need retransmission).

D. Simulations: Co-Channel Interference (Hidden Node)

We use the scenario shown in Figure 4. There are two cells: Cell 1 and Cell 2, served by base station 1 (BS-1) and base-station 2 (BS-2). Node 2 is *downloading* a file from a server adjacent to base-station 2 (BS-2). This leads to packet transmissions by BS-2 that interfere with BS-1. BS-1 is receiving a large *file upload* from node 1 and relaying it to a remote server (which could be 5ms, 40 ms or 100 ms away). Therefore, BS-1's receptions suffer from corruption due to interference. Since BS-1's transmission of TCP acks or MAC acks are short, and it only interferes with BS-2's reception of short TCP ACKs or short MAC acks (which can be recovered with MAC level ARQ), there is little effect on the download performance seen by node 2. Further, since node 2 sees a short RTT, it ramps up its window faster and essentially "captures" the channel for a period of 250 ms.

Node-1's upload session is effectively shut out for 250 ms every 2 seconds. During this period, each packet at node 1's queue is given to the MAC layer which attempts back-off and retransmission for 3 or 7 times (roughly 6.5 ms or 60 ms per packet) before dropping the packet. The TCP layer will see a pattern of no residual loss during periods of no-interference and a huge burst loss during the capture period. In addition, a queue builds up at node 1's IP layer since the MAC layer takes longer to transmit each packet during capture. We therefore recommend buffer size settings of 100 full-sized packets and higher RED thresholds (25 packets and 90 packets) to absorb this sudden burstiness and accommodate a larger window to tolerate capture. We will see that LT-TCP's adaptive MSS method will granulate the window to reduce the likelihood that an entire window is lost during capture and that reactive recovery mechanisms work.

Our first set of results (Table IV) compare SACK and LT-TCP performance when there is no interference (i.e. Cell 2 is quiet). We vary RTTs to be 10ms, 80ms and 200 ms. These numbers are representative of modes in observed RTT distributions reported by CAIDA's Skitter measurement project [1]. The short RTT (10 ms) represents intra-metro or intra-regional RTT (e.g., within the Bay area); medium RTTs (80 ms) represents US east-west coast RTTs; and 200ms (and higher) RTTs are observed in transcontinental links (between US, Europe or Asia). The reason we examine multiple RTTs is because even though the WiFi link itself is a LAN link, the end-to-end RTT matters for TCP-SACK when there is even a small residual erasure rate (see Figure 2).

As expected, the good-puts seen by SACK and LT-TCP are comparable (4.4-4.6 Mbps) and are close to the maximum possible on 802.11b links with no rate adaptation, and MAC-acks sent at 11 Mbps regardless of RTT. Only the preamble for any MAC transmission is sent at 1 Mbps.

In the second set (Table V), we use ARQ = 3 (i.e. two retransmissions at the MAC layer at 11 Mbps) but with 250 ms interference/capture every 2 seconds. SACK goodput is comparable to LT-TCP for LAN RTTs (10 ms), but falls precipitously for higher RTTs. The number of timeouts have increased from 141 to 146 and 200. However, these timeouts are longer on average leading to idle time at the MAC layer (lower MAC throughputs for SACK). In contrast LT-TCP's throughput declines much less (3.3 Mbps to 2.35 Mbps). It has fewer timeouts as the RTTs increase because longer RTTs allow larger windows. Further, LT-TCP strikes a balance between window granulation (to avoid capture) and goodput (reduced per-packet overheads). The decline in goodput for longer RTTs is largely explained through MAC level idle times due to zero queues (reduction in MAC throughput from 5.3 to 4.39 Mbps).

In the third set of results (Table VI) , we use ARQ = 7 (i.e. six retransmissions at the MAC layer at 11 Mbps) with 250 ms interference/capture every 2 seconds. Due to exponential back-off, these six retransmissions take upto 60-75ms before a packet is dropped during the capture phase. SACK goodput improves for both the LAN (10 ms RTT) and USA continental WAN (80 ms RTT) case, though it still collapses for longer RTTs due to high sensitivity to residual error rates. LT-TCP's performance is competitive with SACK for LANs, and is clearly superior for longer RTTs. This set of results suggests that link level ARQ is not a panacea even with LAN links because the end-to-end RTT still matters. Moreover, such high degrees of ARQ persistence are not possible for longer delay links like MAN or satellite which backs the case for end-to-end mechanisms like LT-TCP.

V. Acknowledgements

The authors would like to thank Bob Miller and N.K. Shankaranarayanan of AT&T Labs Research for valuable discussions and insights. We would also like to thank Omesh Tickoo and Prakash Iyer of Intel Corp. This work was supported by AT&T Labs, Intel Corp and NSF-ITR 0313095.

PARAMETER	LT-TCP			SACK		
RTT	10ms	80ms	200ms	10ms	80ms	200ms
Goodput(Mbps)	4.43	4.40	4.39	4.64	4.63	4.52
95% CI for Good-put	[4.36,4.49]	[4.34,4.46]	[4.34,4.43]	[4.61,4.62]	[4.63,4.65]	[4.45,4.61]
Number of Timeouts	0	0	0	0	0	0
MAC Throughput(Mbps)	5.70	5.68	5.64	5.89	5.88	5.72

TABLE IV

ARQ = 3, NO INTERFERENCE: LT-TCP AND SACK PERFORMANCE WITHOUT INTERFERENCE UNDER CONDITIONS OF VARYING END-END DELAY. THE NUMBER OF LINK-LEVEL RETRANSMISSIONS WAS SET TO 3 FOR ALL PACKETS.

PARAMETER	LT-TCP			SACK		
RTT	10ms	80ms	200ms	10ms	80ms	200ms
Goodput(Mbps)	3.30	2.97	2.35	3.22	1.56	0.24
95% CI for Good-put	[3.21,3.39]	[2.63,3.30]	[2.01,2.68]	[3.10,3.34]	[1.50,1.61]	[0.23,0.25]
Number of Timeouts	55	21	1.16	141	146	200
MAC Throughput(Mbps)	5.3	5.01	4.39	4.71	2.36	0.46

TABLE V

ARQ = 3, 250 MS/2 S INTERFERENCE: LT-TCP AND SACK PERFORMANCE WITH INTERFERENCE OF 0.25 SECONDS OUT OF 2 SECONDS UNDER CONDITIONS OF VARYING END-END DELAY. THE NUMBER OF LINK-LEVEL RETRANSMISSIONS WAS SET TO 3 FOR ALL PACKETS.

PARAMETER	LT-TCP			SACK		
RTT	10ms	80ms	200ms	10ms	80ms	200ms
Goodput(Mbps)	3.72	3.76	2.54	4.08	3.07	0.37
95% CI for Good-put	[3.70,3.74]	[3.69,3.83]	[2.43,2.64]	[4.07,4.09]	[2.98,3.15]	[0.3,0.44]
Number of Timeouts	0	0	0	0	0	25.8
MAC Throughput(Mbps)	5.24	5.26	3.56	5.44	4.00	0.62

TABLE VI

ARQ = 7, NO INTERFERENCE :LT-TCP AND SACK PERFORMANCE WITH INTERFERENCE OF 0.25 SECONDS OUT OF 2 SECONDS UNDER CONDITIONS OF VARYING END-END DELAY. THE NUMBER OF LINK-LEVEL RETRANSMISSIONS WAS SET TO 7 FOR ALL PACKETS.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we first made a crucial distinction between "uniform" or "burst" erasure losses on a simulated full-duplex link versus building models of real wireless channels (that tend to be half-duplex and channel scheduling controlled by a MAC layer). The sources of erasures in real wireless networks include both channel impairments (path loss, shadowing, fading) and interference from co-channel and cross-system interferers. In open-spectrum deployments like WiFi, even planned deployments leave adequate scope for co-channel interference precisely due to the conservative nature of cell sizing for coverage.

PHY and MAC layer mechanisms have adaptation techniques designed primarily to handle channel impairments (e.g., rate adaptation, low rate preamble, low-rate control packets like MAC-acks) and export a relatively "clean" virtual link to higher layers. However, these techniques backfire when the primary source of corruption is interference. Such techniques confuse interference as noise, just like the transport layer confuses packet erasure as congestion. Aggressive response in such situations is counter-productive and even eliminates possibilities of mitigation at higher layers. We demonstrated this effect in the context of Bluetooth interference (i.e. cross-system interference).

We suggest moderation in terms of lower-layer adaptation, and availability of hooks for network administrators to turn them off if interference is dominant. Larger buffers, ECN marking and LT-TCP upgrades for TCP make a big difference: large dynamic range of performance (for small and large RTTs, and capture tolerance of at least 250 ms), with small tradeoffs in steady state good-puts.

In future work, we will extend our work to include multi-hop wireless networks, and networks with mixture of wireless links of different error/interference/capacity/RTT characteristics.

REFERENCES

[1] B. Huffake and M. Fomenkov and D. Moore and K. Claffy. Macroscopic analyses of the infrastructure: Measurement and visualization of Internet connectivity and performance.

[2] R. Rao C. Chiasserini. Performance of IEEE 802.11 WLANs in a Bluetooth Environment. In *IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, USA, September 2000.

[3] S. ElRakabawy, A. Klemm, and C. Lindemann. TCP with Adaptive Pacing for Multihop Wireless Networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 288–299, New York, NY, USA, 2005. ACM Press.

[4] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla. The Impact of Multihop Wireless Channel on TCP Throughput and Loss. In *Mobile Computing, IEEE Transactions on*, volume 4, pages 209–221. INFOCOM, March-April 2005.

[5] N. Golmie, R.E.V. Dyck, A. Soltanin, A. Tonnerrre, and O. Rebala. Interference Evaluation of Bluetooth and IEEE 802.11b Systems. *Wireless Networks*, 9(3):201–211, 2003.

[6] N. Golmie and F. Mouveaux. Interference in the 2.4 GHz ISM Band: Impact on the Bluetooth Access Control Performance. In *ICC*, Helsinki, June 2001.

[7] N. Golmie, R.E.V.Dyck, and A. Soltanin. Interference of bluetooth and ieee802.11: Simulation modeling and performance evaluation. In *MSWIM 2001: Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless Systems*, pages 11–18, Rome, Italy, 2001. ACM Press.

[8] K.K. Ramakrishnan, S. Floyd, D. Black. RFC 3168 - The Addition of Explicit Congestion Notification (ECN) to IP, Sep 2001.

[9] R. Krishnan, J. P. G. Sterbenz, W. M. Eddy, C. Partridge, and M. Allman. Explicit Transport Error Notification (ETEN) for Error-prone Wireless and Satellite Networks. *Comput. Networks*, 46(3):343–362, 2004.

[10] L. Baldantoni and H. Lundqvist and G. Karlsson. Adaptive End-to-End FEC for Improving TCP Performance over Wireless Links. In *ICC*, June 2004.

[11] M. Lacage, M.H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *MSWIM 2004: Proceedings of the 7th ACM International Workshop on Modeling, Analysis and Simulation of Wireless Systems*, pages 126–134, Venice, Italy, 2004. ACM Press.

[12] L.Rizzo. On the Feasibility of Software FEC. DEIT Technical Report LR-970131. URL: http://www.iet.unipi.it/ luigi/softfec.ps.

[13] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang. TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. In *Mobile Computing and Networking*, pages 287–297, 2001.

[14] O. Tickoo and V. Subramanian and S. Kalyanaraman and K.K.Ramakrishnan. LT-TCP: End-to-End Framework to Improve TCP Performance over Networks with Lossy Channels. In *Thirteenth International Workshop on Quality of Service (IWQoS 2005)*, Passau, Germany, June 2005.

[15] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. OAR: An Opportunistic Autorate Media Access Protocol for Ad Hoc Networks. In *ACM MOBICOM*, Atlanta , GA, USA, September 2002.

[16] S. Shellhammer. Packet Error Rate of an IEEE 802.11 WLAN in the Presence of Bluetooth. IEEE P802.11 Working Group Contribution, IEEE 802.15-00/133r0, May 2000.

[17] V. Subramanian. Technical Report: LT-TCP: End-to-End Framework to Improve TCP Performance over Networks with Lossy Channels. Technical report, RPI, 2005. URL:http://www.ecse.rpi.edu/Homepages/shivkuma/research/papers-rpi.html#wireless.