# Impact of Network Subsystem on Reliable Transport Protocol Performance over Wireless Links

Zhiguo Xu, Mahesh Marina and Rajive Bagrodia
Mobile Systems Lab
Computer Science Department
University of California, Los Angeles
Email: {zhiguo, mahesh, rajive}@cs.ucla.edu

***Abstract*— We consider the impact of system-related overheads on the performance of two reliable transport protocols (TCP and XCP) over wireless links. Our measurement results indicate that various components providing networking support at a wireless host collectively have significant impact on transport layer performance. Furthermore, the relative performance of protocols is dependent on the host's network subsystem configuration. These results highlight the importance of considering system-dependent behaviors in wireless network protocol evaluations.**

## I. Introduction

Performance of network applications running over different network subsystems can be quite different, even when all systems use identical protocol architectures and specifications. Here network subsystem refers to the combination of system software and hardware used for networking support at a node (e.g., end host, router, access point or AP). Network subsystem software consists of the network API (e.g., sockets), protocol suite (e.g., TCP/IP) and device driver all implemented within the operating system (OS), and firmware on the network interface card (NIC). Hardware components of the network subsystem include CPU, memory, I/O buses and NIC. Throughput achieved by applications depends on how efficiently network subsystem moves data between the application and the network, which in turn rests on the node hardware capabilities, protocol implementations in the OS and their interactions with rest of the OS facilities such as process and memory management, OS-NIC interaction, division of link layer functionality between device driver and NIC firmware, and NIC design and form factor.

Impact of network subsystem on application-level throughput has been well studied in the context of high-speed wired networks [1], [2], [3]. Early work by Clark et al [1] has shown that TCP protocol processing is not the main overhead contrary to popular belief at the time; instead memory copy (user-to-kernel copy and kernel-to-NIC copy) and other OS operations (e.g., interrupt handling) were identified as dominant sources of overhead. Subsequent work has focused on design and evaluation of various optimizations aimed at making the network subsystem more efficient with appropriate NIC support [2], [3]. These optimizations include: zero-copy networking, checksum offloading to NIC, integrated copy/checksum, interrupt coalescing and jumbo frames. Besides network subsystem design, configuration parameters such as send and receive socket buffer size at end hosts also affect throughput [4].

In wireless networks, there has been relatively less work focusing on the impact of network subsystem. Most experimental wireless network studies focus on evaluating the effectiveness of various protocol design aspects (e.g., TCP congestion control, MAC RTS-CTS handshake) when subject to channel effects such as fading, interference and mobility (e.g., [5]). Common to these studies is the implicit assumption that network subsystem is either ideal or has negligible overhead. However, this assumption may not hold in practice across the wide variety of currently used wireless systems (differing in their capabilities, implementation and form factor). Thus, it is important to characterize the impact of system-dependent overheads on wireless network performance. Such an investigation is further warranted by the relatively constrained nature of portable wireless platforms in terms of CPU and memory resources, and the emerging trend toward software-based radios.

Limited work that exists on evaluating system-dependent overheads on wireless network performance primarily looks at this issue from an energy consumption perspective [6], [7]. Feeney and Nilsson [6] have investigated the energy consumption of different 802.11 wireless NICs in an ad-hoc network environment, whereas Wang and Singh [7] have quantified the energy consumption overhead of different functions involved in running TCP on a 802.11 wireless host. However, there are several real-world situations where energy consumption may not be a key concern or networking-related activity is not the dominant energy consumer (e.g., laptop usage in an indoor office environment).

In this paper, our goal is to quantify the impact of various components of the network subsystem in wireless LAN environment (based on IEEE 802.11 standard [11]) on real-world throughput of applications running over diverse reliable transport protocols such as TCP and XCP [12]. In particular, we conduct a coarse-level evaluation focused on the impact of following four aspects: (i) wireless NIC and its interaction with OS; (ii) OS; (iii) node hardware; (iv) system configuration parameters (e.g., buffer size). Our study seeks to answer the following two questions: (i) what is the amount and nature

This work has been funded by the NSF under the Network Research Testbed grant "WHYNET: Scalable Testbed for Next Generation Mobile Wireless Networking Technologies" (award number ANI-0335302).

| | |
|---|---|
| Platform | $Platform_1$: Dell Latitude D600 (Processor 1.6GHz Pentium M, Memory 512MB@266MHz) |
| | $Platform_2$: IBM ThinkPad T43 (Processor 1.8GHz Pentium M, Memory 512MB@400MHz) |
| OS | $OS_1$: Linux Fedora Core 3 (2.6.9) |
| | $OS_2$: Linux Red Hat 9.0 (2.4.20) |
| 802.11 NIC | $NIC_1$: Proxim/Orinoco Gold 11b/g ( Atheros 5001x chip set, PCMCIA, MADWiFi driver [8]) |
| | $NIC_2$: Linksys WPC11 v3.0 Wireless-B (Intersil Prism 3.0 chip set, PCMCIA, HostAP driver [9]) |
| | $NIC_3$: Intel PRO/Wireless 2200BG (Intel chip set, miniPCI, IPW2200 driver [10]) |

TABLE I

SUMMARY OF DIFFERENT ALTERNATIVES USED IN EXPERIMENTS.

of impact on the performance of a transport protocol due to the various system-dependent factors; (ii) is the relative performance of different transport protocols sensitive to the composition of the network subsystem? Addressing these questions helps in identifying the key system aspects that need to be represented in wireless network evaluation tools (e.g., simulators, emulators) for improved fidelity. We also point out how various existing tools differ in their support for incorporating system-dependent behaviors, and how tools lacking adequate support may be extended to model such behaviors.

Our main findings are as follows.

- Design and implementation of each of the key components (i.e., wireless NIC, OS, host system hardware) of the network subsystem and its interaction with rest of the system has substantial impact on the performance of transport protocols. As an example, the performance impact due to NIC overhead alone can match that of key protocol parameters such as physical layer preamble. When put together, transport performance with different network subsystems can be significantly different (more than 30%) depending on their individual composition. This observed impact of the network subsystem is comparable to other well known network effects, e.g. wireless channel errors [13].
- Of even greater significance is the fact that the relative performance of different transport protocols is sensitive to the composition of the network subsystem because of differences in their interaction with the system. In particular, we observe reversals in performance between TCP and XCP in some operating regions depending on the choice of wireless NIC. We also observe that the system configuration parameters such as socket buffer size have similar non-uniform impact on different protocols depending on the workload characteristics. By way of representing system-related overheads in common evaluation tools, we show the effectiveness of a simple mechanism for approximately modeling the NIC-dependent overhead in wireless network simulators.

The rest of the paper is organized as follows. Section II elaborates on our evaluation methodology and experiment settings. Section III presents our measurement results and analysis of the impact of various network subsystem components/parameters on transport protocol performance. We conclude in Section IV.

## II. EXPERIMENTAL SETUP

Our evaluations are based on measured throughput performance of reliable transport protocols on laptop platforms running Linux and equipped with a commodity 802.11 wireless NIC. The different platforms, OSs and 802.11 NICs used in our evaluations are summarized in Table I. Since real-world protocol performance is dependent not only on the system aspects but also on the wireless channel, we isolate system effects by limiting attention to an almost ideal radio propagation environment (short distance, negligible wireless losses, no interference).

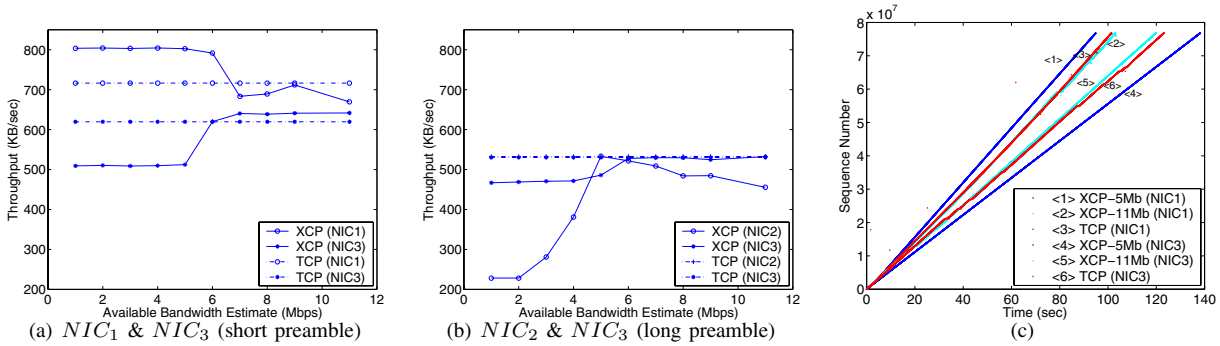
We consider standard TCP and XCP [12] as two representative reliable transport protocols. We choose these two protocols because of their widely different approaches to congestion control. TCP probes for available bandwidth by gradually increasing the sending rate and infers congestion implicitly via packet loss, while XCP uses explicit feedback from the network about the level of congestion and adapts the sending rate accordingly. Certain properties of XCP are well-suited for the wireless environment even though it was designed for high bandwidth-delay product networks (e.g., high-speed optical networks, large delay satellite links). For example, XCP enables identification of non-congestion related wireless losses by decoupling rate control from error control. XCP, however, requires accurate estimation of available bandwidth at each link along the end-to-end path for feedback calculation, which is challenging over lossy and shared wireless channels.

Rather than use an actual bandwidth estimation capability for XCP, we run experiments for a wide range of static values for the "estimated bandwidth" to reflect a wide range of estimation errors (covering underestimation, accurate estimation and overestimation cases). We use a Linux implementation of XCP from Zhang and Henderson [14], which is developed as an extension to the standard TCP implementation.

## III. RESULTS

### A. Wireless NIC

We begin our analysis of the impact of various network subsystem components on transport protocol performance by looking at the impact of wireless NIC and its interaction with rest of the network subsystem. For this purpose, we consider three wireless NICs, listed in Table I, differing in various aspects, including: chip set design, form factor, bus interface, driver implementation and functionality. This set of NICs are selected with careful consideration for wider applicability of our measurements — most available commodity 802.11 NICs




(a) $NIC_1$ & $NIC_3$ (short preamble) (b) $NIC_2$ & $NIC_3$ (long preamble) (c)

Fig. 1. (a,b) Impact of wireless NIC on throughput performance of TCP and XCP. (c) Sequence number versus time plots with $NIC_1$ and $NIC3$ for XCP with two bandwidth estimate values (5Mb and 11Mb) and TCP.

use chip-sets identical to one of the NICs in our set. For brevity, we henceforth refer to the three NICs as $NIC_1$, $NIC_2$, $NIC_3$ respectively (see Table I). All experiments in this subsection are based on Dell Latitude D600 ($Platform_1$ in Table I) and Linux Fedora Core 3 kernel ($OS_1$ in Table I).

For this study, we consider a single wireless link with one sender host transferring a large file (80MB) to a receiver host. As noted already, for isolating system effects, our experiments are conducted in an environment with little or no wireless losses and interference; this was also verified by examining the experiment traces collected using the AiroPeek tool [15]. During the course of experimentation, we found that different NICs use different default values for the physical layer preamble (by default $NIC_1$ and $NIC_3$ use a short preamble, whereas $NIC_2$ uses a long preamble). As there was no easy way to modify this parameter for two of the NICs ($NIC_1$ and $NIC_2$), we separately present the results for short and long preamble cases. All NICs operate in 802.11b mode with RTS/CTS disabled, fixed PHY data rate of 11Mbps, 1500 byte MTU and identical values for other 802.11 MAC parameters. As per transport layer settings, a large socket buffer size of 640KB is used for both sender and receiver, and the delayed ACK mechanism is disabled.

Fig. 1(a,b) presents throughput for TCP and XCP with different NICs as a function of the available bandwidth estimate value (see Section II). Note that TCP throughput remains identical across various bandwidth estimate values as it does not use explicit bandwidth estimation. Fig. 1(a) compares $NIC_1$ and $NIC_3$ using short preamble, whereas Fig. 1(b) compares $NIC_2$ and $NIC_3$ using long preamble. For all experiment results in the paper, each data point represents an average of at least ten runs.

Clearly, the choice of wireless NIC substantially affects transport protocol throughput. Fig. 1(a) shows that TCP performance with $NIC_1$ is about 15% better relative to $NIC_3$. We observed negligible number of packet losses in both cases. So this throughput difference is solely due to differences between the NICs (and the corresponding drivers) since all other settings are identical including transmission and propagation delays. The overhead in moving data between the driver and the wireless medium may account for the throughput differences; this overhead is due to a combination of factors such as NIC implementation and NIC-OS interaction (in terms of interrupt handling and DMA support). We quantify this overhead via round-trip time (RTT) measurements obtained using 512 pings with 64 byte packets sent successively at one second intervals. We find that minimum RTT with $NIC_3$ was larger than $NIC_1$ by more than a factor of 2.25 (1.35ms vs. 0.59ms). These differences in RTTs can help explain TCP throughput differences since TCP throughput has an inverse relationship with RTT [16]. Same reasoning as above applies for the similar TCP throughput seen between $NIC_2$ and $NIC_3$ (Fig. 1(b)) when using a long preamble — $NIC_2$ and $NIC_3$ have similar minimum RTT values (1.77ms and 2.1ms, respectively).

Besides, comparison of TCP throughput with $NIC_3$ between Fig. 1(a) and (b) shows that the performance drops by about 15% with long preamble. This suggests that performance impact due to NIC-related overhead alone (as seen from Fig. 1(a)) can be comparable to that of protocol-specific parameters (preamble length in this case).

As for XCP, throughput differences across NICs vary over a wide range depending on the available bandwidth estimate value. As shown in Fig. 1(a), $NIC_1$ provides better XCP throughput relative to $NIC_3$ with improvements ranging from 60% in underestimation case, 30% near accurate bandwidth estimation case, and about 8% in overestimation case. These results with XCP will be discussed below.

More importantly, the NIC-dependent overhead can alter the relative protocol performance in some cases. As seen from Fig 1(a), XCP performance in the underestimation region is about 15% better relative to TCP with $NIC_1$, whereas it gets about 15% worse with $NIC_3$. As discussed below, different sensitivity of XCP to bandwidth underestimation with different NICs (with different RTTs) explains this reversal in relative performance.

Further, the specifics of wireless NICs can influence protocol behavior through interaction with protocol-related parameters. This is evident from XCP performance trends in Fig. 1, where the effect of bandwidth estimation errors is noticeably different with different NICs. Let us first focus on Fig. 1(a) comparing XCP performance with $NIC_1$ and $NIC_3$. In the

underestimation region (1-5Mbps), the congestion window was seen to stay at one (segment) for both NICs; this implies that throughput performance is fully determined by the RTT — receipt of an ACK every RTT lets the sender transmit a new segment. Hence the throughput performance with $NIC_3$ is poor in this region due to its relatively larger RTT. Note that RTT also determines the optimal congestion window size. Here optimal window is close to 1 for $NIC_1$ while it is between 1 and 2 for $NIC_3$. This explains why $NIC_1$ delivers peak throughput in the underestimation region, whereas $NIC_3$ reaches its peak beyond this region.

Relative performance trends of XCP in the overestimation region (beyond 6Mbps) in Fig. 1(a) are interesting — throughput drops below the peak for $NIC_1$, while it stays close to the peak value for $NIC_3$. Note that with bandwidth overestimation, XCP router (same as the sender node in this case) provides inflated feedback (greater than optimal) to the XCP sender resulting in an inflated congestion window. The response from the sender is to inject more data into the network, which results in more data drawn from the application by XCP and moved to the NIC, creating an artificial overload situation. In a general multihop case, this can cause packet loss due to buffer overflow at the bottleneck node on the path. In our single link case, however, there is no loss due to buffer overflow as a host does not overflow its NIC. Nevertheless, the increased queueing due to this artificial overloading can stress the NIC. The sequence number versus time plots in Fig. 1(c) (obtained from complete packet trace collected using AiroPeek) suggest that $NIC_3$ responds better compared to $NIC_1$ in such situations, thus explaining their different performance trends in the overestimation region. Note that the effect of bandwidth overestimation in XCP leads to a congestion window behavior similar to TCP (linear increase with time), hence their similar performance.

XCP performance trends in Fig. 1(b) can be explained using similar reasoning as above with the following two exceptions. The optimal congestion window for both $NIC_2$ and $NIC_3$ is greater than 1 due to the use of long preamble, which affects their behavior in the underestimation region. Also $NIC_2$ appears to respond poorly to load in the overestimation region like $NIC_1$ in Fig. 1(a).

*Modeling wireless NIC overhead:* The preceding discussion highlights the importance of incorporating NIC-dependent overhead in wireless network evaluation tools for accurate performance prediction as it impacts both individual and relative performance of transport protocols. Further, our analysis suggests that observed performance behaviors of TCP and XCP can be largely attributed to the RTT differences between the NICs. We also carried out extensive set of ping tests for different packet sizes (between 64 and 1500 bytes), and find that minimum RTT differences are insensitive to packet size. For instance, the minimum RTT with $NIC_3$ is greater by about 0.75ms compared to $NIC_1$ at 11Mbps PHY rate regardless of the packet size. Based on the above observations, a simple mechanism to model NIC-related overhead is to introduce a fixed delay for each packet immediately before it is handed down to the NIC for transmission; the value of this per-packet delay for a given NIC is set to the observed minimum RTT with that NIC.

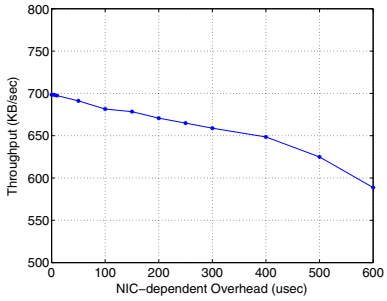



Fig. 2. Impact of varying the delay in the wireless NIC overhead model in QualNet on TCP performance.

We use two common wireless network simulators, namely ns-2 and QualNet, to study the effectiveness of our NIC overhead model in replicating protocol performance behaviors observed in our measurements. While ns-2 seems to have a similar model already, simulations with settings identical to those described earlier in this subsection do not show any noticeable impact on TCP throughput from varying the delay in the NIC overhead model. A closer look revealed that it was because ns-2's modeling of the interaction between network (IP) and link layers differs from real systems. Specifically, ns-2 uses a drop-tail interface queue with a default limit (50 packets) to move packets from IP to the link layer (NIC). The use of a multi-packet interface queue creates kind of pipelining effect that results in multiple packets waiting in the queue to concurrently experience the NIC overhead delay. Also transferring packets to the interface regardless of its queue occupancy causes IP layer to overflow the network device and drop packets; in contrast, a real host does not overflow its own device.

With QualNet, the modeling of the interaction between IP and NIC more closely matches reality, and we have added the simple NIC overhead model. As shown in Fig. 2, introduction of additional delay reflecting difference in RTTs between $NIC_1$ and $NIC_3$ shows a close match with Fig. 1(a) in terms of drop in TCP throughput. This suggests that it is indeed possible to capture system-dependent behaviors resulting from NIC overheads in simulation tools; we can also achieve a similar effect with ns-2 with appropriate modifications to the IP-NIC interaction. However, we should note that the above simple model has some limitations. First, it does not model the effect of physical layer modes (a/b/g) and data rates. Second, it is a static model in that per-packet delay is fixed; in practice, NIC-related overhead may vary with system load.

We will investigate a more realistic dynamic load-dependent model in future drawing from the following approaches. Full system simulators, e.g. M5 [17], avoid the need to consider load-dependent overhead as part of NIC overhead by simulating the host hardware and running unmodified OS (e.g., Linux) and applications within the simulator. However, this approach may severely limit the size of networks that can be

| | $OS_1$ | $OS_2$ |
|---|---|---|
| $NIC_1$ | | |
| *TCP throughput* | 723.96KB/sec | 622.41KB/sec |
| *min RTT* | 0.60msec | 0.56 msec |
| $NIC_2$ | | |
| *TCP throughput* | 559.06KB/sec | 484.97KB/sec |
| *min RTT* | 2.1msec | 2.0 msec |

TABLE II

IMPACT OF OS ON TCP THROUGHPUT PERFORMANCE WITH DIFFERENT WIRELESS NICS.

| | $Platform_1$ | $Platform_2$ |
|---|---|---|
| $NIC_1$ | | |
| *TCP throughput* | 723.96KB/sec | 792.5KB/sec |
| *min RTT* | 0.60msec | 0.65msec |
| $NIC_3$ | | |
| *TCP throughput* | 619.65KB/sec | 725.3KB/sec |
| *min RTT* | 1.55msec | 1.40msec |

TABLE III

IMPACT OF NODE HARDWARE ON TCP THROUGHPUT PERFORMANCE WITH DIFFERENT WIRELESS NICS.

evaluated. Network emulation [18] is another alternative in which applications and OS run on real hardware, whereas the NIC and wireless channel are emulated. Emulation is a middle ground between common network simulators and full system simulators in terms of scalability. So it may be an attractive approach to easily account system-related overheads.

### *B. Operating System and Node Hardware*

*OS:* Here we first evaluate the impact of OS on transport protocol performance by comparing TCP throughput with two versions of Linux, i.e. $OS_1$ and $OS_2$ in Table I. We choose these two specific versions as they are substantially different in terms of their implementation of the network subsystem and hence on its efficiency. Results presented here are based on $Platform_1$ (see Table I). We consider two NICs: $NIC_1$ and $NIC_2$. For $NIC_3$, we could not find a driver compatible with $OS_2$. Rest of the experiment settings are identical to the previous subsection.

Results summarized in Table II indicate that the OS has noticeable impact on TCP performance. In particular, going from $OS_2$ to $OS_1$ can improve TCP throughput by about 15% with both NICs. Unlike before, however, the minimum RTT differences are not prominent enough to explain the differences in throughput. Among the many new features/enhancements included in Linux Fedora Core 3 kernel (i.e., $OS_1$), it appears that the new scheduler, kernel preemption capability and memory management changes most likely explain the observed TCP throughput improvements. Note that network subsystem efficiency is affected by the design of other OS facilities, notably process and memory management.

*Node hardware:* We now evaluate node hardware impact on transport protocol performance using two platforms ($Platform_1$ and $Platform_2$ shown in Table I) with different CPU and memory bus speeds. We used these two platforms as they were readily available at the time of our experimentation. For this experiment, we used $OS_1$. We consider two NICs ($NIC_1$ and $NIC_3$) both using short preamble. Other experiment settings are same as before.

Results in Table III show that the host hardware can have a noticeable impact on transport performance. In particular, $Platform_2$ provides better performance than $Platform_1$ (10% or more) with both NICs. This may be largely due to the much faster (around 50%) memory bus speed with $Platform_2$ as moving each word of data from the application to the NIC incurs three memory accesses (read from application, write to kernel buffer, and read from kernel buffer to NIC). Interestingly, we see different impact of the host hardware with different NICs going from $Platform_1$ to $Platform_2$ (10% improvement with $NIC_1$ vs. 17% improvement with $NIC_3$). With further investigation, we found that the driver for $NIC_1$ additionally includes an optimization to reduce the number of interrupts from the NIC, which has the effect of reduced CPU dependence, which may explain the greater performance improvement with $NIC_3$ from a faster CPU in $Platform_2$. We can also observe that the collective impact of the host hardware and the NIC can result in nearly 30% throughput difference (compare $Platform_1$-$NIC_3$ and $Platform_2$-$NIC_1$).

No correlation is observed between minimum RTT differences and throughput differences across the two platforms, which may be because minimum RTT does not include load-dependent overhead. As noted earlier, network emulation approach may make it easier to capture overheads related to OS and host hardware that affect transport performance.

### *C. Sensitivity to System Configuration Parameters*

Here we present how system configuration parameters affect performance of TCP and XCP in a heterogeneous wired/wireless setting. In particular, we consider the *socket buffer size* at end hosts, an important system configuration parameter in TCP/IP networks. Semke et al [4] showed that this parameter value can significantly affect TCP throughput in wired networks. Recently XCP was also shown to be sensitive to this parameter setting in wired networks using a single bottleneck network topology [14], albeit for somewhat different reasons than TCP. In contrast to these prior efforts, our goal is to characterize its impact in a wired/wireless network on the *relative* performance of transport protocols, namely TCP and XCP.

The experiment setup consists of a simple wireless LAN with one 802.11 wireless host associated with an access point (AP) and the AP additionally connected to a wired host over a gigabit ethernet link. Fixed data rate of 11Mbps is set for the wireless link. We use $Platform_1$ and $OS_1$ for all nodes and $NIC_1$ for the wireless link. We consider two buffer sizes: small — 16KB (default in $OS_1$) and large — 640KB. For workload, we consider two types of bulk transfer flows: (i) from wireless host to the wired host via the AP ("upload"); (ii) from wired host to the wireless host ("download"). Identical buffer settings are used at both sender and receiver hosts. All other settings are as in Section III.A.

Fig. 3(a,b) shows that TCP and XCP react differently to the change in socket buffer size depending on the flow direction.

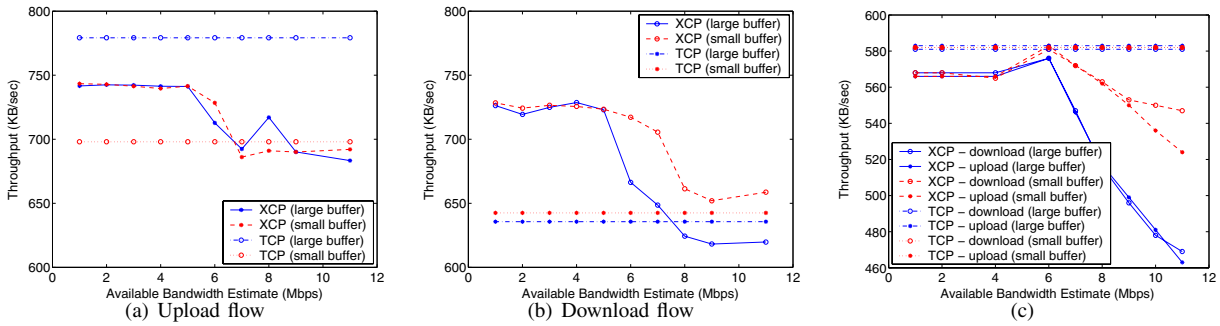



Fig. 3. (a,b) Impact of socket buffer size on relative performance of TCP and XCP using real experimentation; (c) using ns-2 simulations.

For the upload case (Fig. 3(a)), TCP with large buffer performs better (around 10%) than with small buffer; in contrast, there is no noticeable difference for XCP between large and small buffers both in absolute throughput and sensitivity to the bandwidth estimate value. Buffer size has a reverse effect in the download case (Fig. 3(b)) — TCP performance remains unaffected, whereas XCP gains with use of small buffer by about 10% in the overestimation region. In summary, the buffer size and the flow direction together contribute to the different responses of XCP and TCP.

We repeated the above experiment using the ns-2 simulator. Note that XCP simulation model is currently available only in ns-2. As we see from the ns-2 results in Fig. 3(c), unlike the physical experiments, TCP performance remains similar in all cases and XCP performance is insensitive to the flow direction. These results with ns-2 highlight the current lack of support in commonly used wireless network simulators for capturing subtle interactions between system parameters (e.g., buffer size) and workload characteristics that affect protocol performance behaviors.

## IV. Conclusions

In this paper, we study the effect of the network subsystem on the transport protocol performance in wireless LAN environment. In particular, we evaluate performance impact due to each of the main network subsystem components (wireless NIC, host system hardware and the OS) on two reliable transport protocols, i.e. TCP and XCP. We also present the impact of network subsystem parameters using the socket buffer size as a representative parameter. Our results show that the composition of the network subsystem can have significant impact (around 30%) on transport performance, comparable to that of channel condition or protocol performance optimizations. More importantly, we find that the configuration of the network subsystem (through interaction with protocol-specific parameters) can have *non-uniform* impact on different protocols resulting in reversals in their relative performance. Thus, it is important to incorporate system-dependent behaviors in evaluation tools for realistic protocol performance studies. As an initial step in this direction, we showed the utility of a simple NIC-overhead model in wireless network simulators.

In the future, we plan to further investigate how to accurately represent system overhead in wireless network evaluation tools, including modeling techniques for simulators and evaluating the effectiveness of emulation methodology. We also plan to extend our study to wider settings (e.g., multi-flow and multi-hop scenarios)

## References


[1] D. Clark, V. Jacobson, J. Romkey, and H. Salwen, "An Analysis of TCP Processing Overhead," *IEEE Communications Magazine*, vol. 27, no. 6, 1989.

[2] J. Chase, A. Gallatin, and K. Yocum, "End System Optimizations for High-Speed TCP," *IEEE Communications Magazine*, vol. 39, no. 4, 2001.

[3] S. Zeadally and L. Zhang, "Enabling Gigabit Network Access to End Users," *Proceedings of the IEEE*, vol. 92, no. 2, 2004.

[4] J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning," in *Proc. ACM Sigcomm*, 1998.

[5] S. Choi, K. Park, and C. Kim, "On the Performance Characteristics of WLANs: Revisited," in *Proc. ACM Sigmetrics*, 2005.

[6] L. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," in *Proc. IEEE Infocom*, 2001.

[7] B. Wang and S. Singh, "Computational Energy Cost of TCP," in *Proc. IEEE Infocom*, 2004.

[8] "Madwifi: Multiband Atheros Driver for WiFi," http://madwifi.sourceforge.net/.

[9] "HostAP: Host AP Driver for Intersil Prism2/2.5/3," http://hostap.epitest.fi/.

[10] "IPW2200: Intel PRO/Wireless 2200BG Driver for Linux," http://ipw2200.sourceforge.net/.

[11] IEEE Std. 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.

[12] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in *Proc. ACM Sigcomm*, 2002.

[13] C. Casetti *et al.*, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," *ACM/Kluwer Wireless Networks Journal*, vol. 8, no. 9, 2002.

[14] Y. Zhang and T. Henderson, "An Implementation and Experimental Study of the eXplicit Control Protocol (XCP)," in *Proc. IEEE Infocom*, 2005.

[15] WildPackets Inc., "AiroPeek Wireless LAN Analyzer," http://www.wildpackets.com/.

[16] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," in *Proc. ACM Sigcomm*, 1998.

[17] N. Binkert *et al.*, "Analyzing NIC Overheads in Network-Intensive Workloads," in *Proc. Eighth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW)*, 2005.

[18] J. Zhou, Z. Ji, and R. Bagrodia, "TWINE: A Hybrid Emulation Testbed for Wireless Networks and Applications," in *Proc. IEEE Infocom*, 2006, to appear.