

Internet Protocols over Wireless Networks

George C. Polyzos and George Xylomenos
{polyzos, xgeorge}@cs.ucsd.edu

Center for Wireless Communications

and

Computer Systems Laboratory

Department of Computer Science and Engineering

University of California, San Diego

La Jolla, California, 92093-0114, U.S.A.

Abstract

We discuss the problems that arise when standard Internet protocols are used over wireless links, such as degraded Transmission Control Protocol (TCP) performance when wireless errors are interpreted as congestion losses. We use case studies drawn from commercial Wireless Local Area Networks and cellular telephony systems to illustrate these problems. Then, we survey some proposed approaches to mitigating such problems and examine their applicability. Finally, we look at the future of wireless systems and the new challenges that they will create for Internet protocols and state some goals for further protocol enhancement and evolution.

I. INTRODUCTION

The Internet has expanded its reach over new telecommunications systems not long after each has become available, so it is not surprising that existing and emerging wireless systems are no exception. This ubiquity is partly due to the design of the Internet Protocol (IP), which seamlessly interconnects dissimilar networks into a global internetwork, offering a common interface to higher protocol layers. Despite the fact that satellite links have been long used on the Internet, the focus of Internet protocol development has been on wired media, with their decreasing error rates and increasing bandwidth. Supporting the simple services of IP over media with such nice characteristics has been rather straightforward.

Physical and economic factors cause wireless links to lag behind their wired counterparts, generally exhibiting higher error rates and lower data rates. Since these characteristics violate assumptions commonly made for wired media, more sophisticated and complex link layer protocols have been used over wireless links in an attempt to improve performance at the network and higher layers. A typical approach is to reduce error rate at the expense of data rate. As IP can accommodate any kind of link layer protocol, these techniques hide the peculiarities of underlying media without compromising compatibility with higher layers.

Emerging candidates for inclusion in the mainstream of the Internet are wireless communications systems such as Cellular and Cordless Telephony (CT) and Wireless Local Area Networks (WLANs). Our discussion also applies to satellite systems, but particularly in the case of geostationary satellite links, the much larger propagation delays encountered are a major consideration that we do not address here. Wireless systems present new challenges, such as a rapidly changing error behavior due to mobility and terrestrial obstructions and reflections. Cellular systems in addition suffer from communication pauses during *handoffs*, when mobile devices move between adjacent cells. Their performance shortcomings when employed on the Internet can be addressed by a synthesis of techniques for enhancing the performance of both wired and wireless links, modified to take into account their unique characteristics.

This article discusses the specific problems wireless links and mobility present to Internet protocol performance (section II), surveys approaches to enhancing their performance over such links (section III), and looks into the requirements future wireless systems and applications will probably introduce and how to address them (section IV).

II. INTERNET PROTOCOLS AND WIRELESS LINKS

A. Internet Transport Layer Protocols

Transport layer protocols lie between user applications and the network. Although they offer user-oriented services, their design is based on assumptions about network characteristics. One choice offered by the Internet is the User Datagram Protocol (UDP), essentially a thin layer over IP. UDP offers a best-effort message delivery service, without any flow, congestion, or error control. Such facilities may be built on top of it, if needed, by higher-layer protocols or applications. Besides offering nearly direct access to IP, UDP is also useful for applications that communicate over Local Area Networks (LANs). Because wired LANs are typically extremely reliable and have plenty of bandwidth available, their lack of error and congestion control is unimportant.

Even though wired long-haul links have been exhibiting decreasing error rates (due to wide-spread use of optical fiber), the statistical multiplexing of increasing traffic loads over wide-area links has replaced errors with congestion as the dominant loss factor on the Internet. Congestion is caused by temporary overloading of links with traffic that causes transmission queues at network routers to build up, resulting in increased delays and eventually packet loss. When such losses occur, indicating high

| User Payload (Bytes) | Datagram Error Rate (%) | Overhead (%) | Throughput (%) |
|-------------------------|----------------------------|--------------|----------------|
| 200 | 0.097 | 19.355 | 80.567 |
| 500 | 0.194 | 8.759 | 91.064 |
| 800 | 0.388 | 5.660 | 93.974 |
| 1100 | 0.775 | 4.181 | 95.076 |
| 1400 | 1.550 | 3.315 | 95.186 |

TABLE I
WIRELESS LAN PERFORMANCE (UDP)

levels of congestion, the best remedy is to reduce the offered load to empty the queues and restore traffic to its long-term average rate [8].

The Transmission Control Protocol (TCP) is the other common transport layer protocol choice offered on the Internet, and the most popular one, since it supports many additional facilities compared to UDP. It offers a connection-oriented byte stream service that appears to applications similar to writing (reading) to (from) a sequential file. TCP supports reliable operation, flow and congestion control, and segmentation and reassembly of user data. TCP data segments are acknowledged by the receiving side in order. When arriving segments have a gap in their sequence, duplicate acknowledgments are generated for the last segment received in sequence. Losses are detected by the sender either by timing out while waiting for an acknowledgment, or by a series of duplicate acknowledgments implying that the next segment in the sequence was lost in transit. Since IP provides an end-to-end datagram delivery service, TCP resembles a Go-Back-N link layer protocol transmitting datagrams instead of frames. On the other hand, IP can reorder datagrams, so TCP cannot assume that all gaps in the sequence numbers mean loss. This is why TCP waits for multiple duplicate acknowledgments before deciding to assume that a datagram was indeed lost.

During periods of low traffic or when acknowledgments are lost, TCP detects losses by the expiration of timers. Since Internet routing is dynamic, a timeout value for retransmissions is continuously estimated based on the averaged round-trip times of previous data/acknowledgment pairs. A good estimate is very important: Large timeout values delay recovery after losses, while small values may cause premature timeouts and thus retransmissions to occur when acknowledgments are delayed, even in the absence of loss. Recent versions of TCP make the key assumption that the vast majority of perceived losses are due to congestion [8], thus combining loss detection with congestion detection. As a result losses cause, apart from retransmissions, the transmission rate of TCP to be reduced to a minimum and then gradually to increase so as to probe the network for the highest load that can be sustained without causing congestion. Since the link and network layers do not offer any indications as to the cause of a particular loss, this assumption is not always true, but it is sufficiently accurate for the low-error-rate wired links. A conservative reaction to congestion is critical in avoiding congestive collapse on the Internet with its ever-increasing traffic loads [8].

B. Protocol Performance Over a Single Wireless Link

The low-error-rate assumption, while reasonable for wired links, has disastrous results for wireless links. A representative WLAN, the WaveLAN, when transmitting UDP packets with 1400 bytes of user payload at a data rate of 1.5 Mbit/s over an 85-foot distance has an average Frame Error Rate (FER) of 1.55% [11]. The errors are usually clustered as in congestion losses. The main factors influencing FER are distance between hosts and frame size. Distance is generally imposed by the operating environment, but the frame size can be reduced to minimize the error rate. Reducing the frame size by 300 bytes for this distance (85 ft) cuts in half the measured FER [11], as shown in the first two columns of Table I (see [11] for a two-state error model accurately reflecting the behavior of this system). By encapsulating each UDP segment in one IP datagram and then in one WaveLAN frame, we have 48 bytes of header and trailer overhead per segment, so overhead as a percentage of total bandwidth used increases with shorter frames, as shown in the third column of Table I. To find the frame size that maximizes data throughput, we have to combine FER and frame overhead for each frame size to get the percentage of total bandwidth used that consists of user data, shown in the fourth column of Table I. In this case, when reducing frame size the increase in header overhead more than balances the scale in terms of effective data rate. Data throughput at the link layer is maximized with 1400-byte data segments, at roughly 95% of link capacity after discounting losses and overhead. As an example, if we transmit 1400-byte data segments, or 1448-byte Ethernet frames, at a peak bandwidth of 1.6 Mbit/s, FER is 0.0155 and the overhead factor is $48/1448 = 0.03315$. The data throughput then is $(1 - 0.0155) \times (1 - 0.03315) \times 1.6 = 1.523$ Mbit/s, about 95% of the 1.6 Mbit/s of bandwidth used.

TCP is expected to achieve lower data throughput than UDP, not only due to its extra 12 bytes of overhead per data segment (the difference between UDP and TCP header sizes), but also because reverse traffic (acknowledgments) must share the broadcast WLAN medium with forward (data) traffic. A secondary effect of sharing the link is that transmissions can be delayed due to collisions. Our measurements indicate that with the WaveLAN these collisions may sometimes go undetected, thus increasing the error rate visible to higher layers with bidirectional (TCP) traffic [13] (Figure 1). Note also that host mobility increases error rates for this WLAN by about 30%, in the absence of handoffs [11].

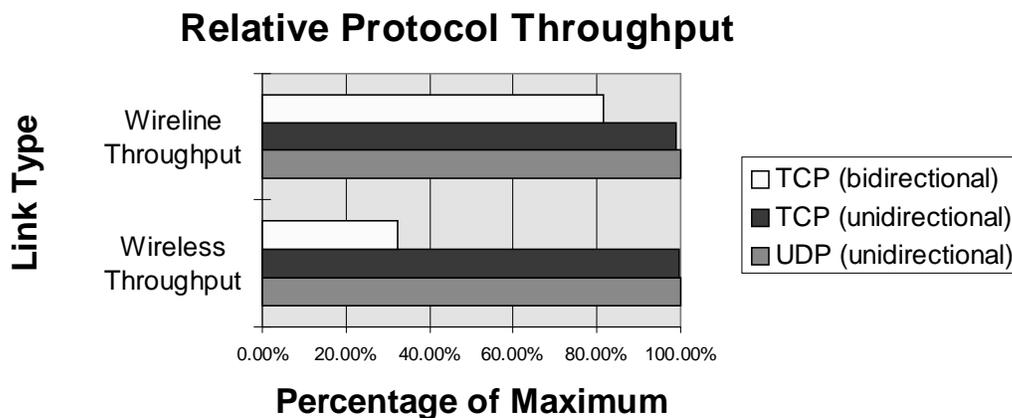


Fig. 1. Wireless LAN performance: TCP traffic has to contend with ACKs on the single wireless channel. The inefficiency of Collision Avoidance versus Collision Detection is shown.

| User Payload (Bytes) | Datagram Error | | |
|-------------------------|----------------|--------------|----------------|
| | Rate (%) | Overhead (%) | Throughput (%) |
| 81 | 7.763 | 5.814 | 86.874 |
| 167 | 14.924 | 2.907 | 82.603 |
| 339 | 27.620 | 1.453 | 71.328 |
| 683 | 47.612 | 0.727 | 52.008 |
| 1371 | 72.555 | 0.363 | 27.346 |

TABLE II
CELLULAR LINK PERFORMANCE (TCP)

Measurements of a TCP file transfer over a single WaveLAN link using 1400-byte data segments revealed that the throughput achieved is only 1.25 Mbit/s out of the 1.6 Mbit/s available on the link [11]; thus, a throughput reduction of 22% is caused by only a 1.55% frame error rate. This is due to TCP initiating congestion avoidance mechanisms that reduce its transmission rate, even though the loss was not due to congestion. Note that if errors were uniformly distributed rather than clustered, throughput would be 1.51 Mbit/s [11], only a 5.5% reduction from the nominal rate, since TCP performs worse with losses clustered within one transmission window.

Cellular links make even less of their nominal bandwidth available to the users, due to their 1–2% FER [9], quite large considering the short frames used. An IS-95 (CDMA cellular) link transmitting at full rate uses fixed-size 172-bit data frames, excluding link layer overhead. Since this frame size is insufficient even for TCP and IP headers, the link layer segments IP datagrams into multiple frames. For comparison with the WLAN case above, consider a UDP packet with 1400 bytes of user payload and 28 bytes of UDP/IP overhead. This datagram would be segmented into 68 frames, so at a 1% FER the probability that the datagram will make it across the link is 50.49%, while at a 2% FER this probability drops to 25.31%, assuming independent frame errors. Even if frame errors are heavily correlated (bursty), high average FER results in high datagram error rates. Frame errors are less bursty than bit errors in CT systems because data bits from multiple frames are interleaved before transmission. After deinterleaving at the receiver, error bursts are spread uniformly over multiple frames so that usually the embedded error correction code of each frame can recover its contents. This technique reduces FER and randomizes frame errors so as to avoid audible speech degradation, but it adds considerable delay, as multiple frames need to be received before deinterleaving and decoding can take place. For example, the typical frame delivery delay on IS-95 is around 100 ms [9].

Reducing the datagram size reduces FER at the expense of increasing header overhead. However, TCP overhead can be reduced to 3–5 bytes per datagram by employing header compression, a technique customized for low-bandwidth serial links. This optimization is feasible *only* for the TCP/IP combination. Since CT systems use separate uplink and downlink channels, forward (data) and reverse (acknowledgment) traffic do not interfere as in the WLAN case. Rather surprisingly then, TCP offers potentially more bandwidth to the user than UDP in CT links, due to TCP header compression.

Assuming full frame utilization, 5-byte compressed TCP/IP headers, and a 2% frame error rate, Table II shows the percentage of the nominal link layer bandwidth that is available for user data. The datagram size varies from 4 to 64 link layer frames (i.e., 81–1371 data bytes), as shown in the first column, and independent errors are assumed for simplicity, with the datagram error rate shown in the second column. Independent errors are a pessimistic assumption, since clustered errors would affect fewer datagrams, but it is reasonable in view of the physical layer interleaving. While error rate increases with longer datagrams, the constant TCP/IP overhead becomes a smaller factor of the total bandwidth, as shown in the third column. Throughput is maximized

TCP Throughput (1.9e-6 Bit Error Rate)

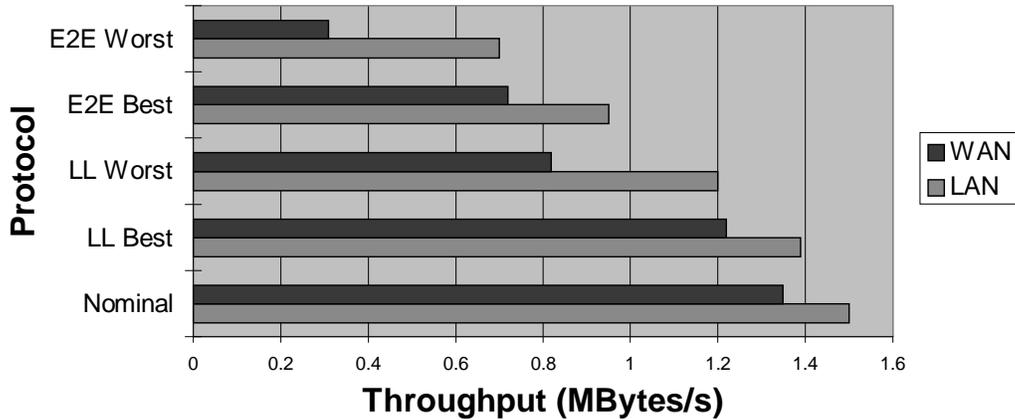


Fig. 2. TCP Performance in multi-hop versus single hop environments. (LL = link layer enhancements.)

| | Nominal | Simple TCP | | Improved TCP | |
|-----|-------------|-------------|--------|--------------|--------|
| LAN | 1.5 Mbit/s | 0.70 Mbit/s | 46.66% | 0.89 Mbit/s | 59.33% |
| WAN | 1.35 Mbit/s | 0.31 Mbit/s | 22.96% | 0.76 Mbit/s | 56.29% |

TABLE III
THROUGHPUT OF LAN AND WAN CONNECTIONS (TCP)

here with 81-byte datagrams, at about 87% of link capacity, as error rate dominates overhead in the throughput calculation. As an example, when transmitting 81-byte data segments, that is, 86-byte datagrams, the error rate is $(1 - 0.98^4) = 0.07763$, since four link layer frames are used, and the overhead factor is $5/86 = 0.05814$. Since the total bandwidth of this system is 8.6 Kbit/s (each 171-bit frame takes 20 ms to transmit), the data throughput is $(1 - 0.07763) \times (1 - 0.05814) \times 8.6 = 7.47$ Kbit/s, which is about 87% of the bandwidth used. Note, however, that this throughput assumes perfect error recovery (i.e., TCP never waits for a timeout and never retransmits correctly received data), which is not the case in practice.

C. Protocol Performance Over Multiple Links

The preceding discussion focused on paths composed of a single wireless link. When multiple wireless links are traversed, errors accumulate accordingly. This is the case when users of separate cellular or WLAN systems communicate via the wired infrastructure. Making the assumption that the behavior of the two wireless links is uncorrelated, the cumulative datagram error rate for two WaveLAN links is 3.08%, while for two cellular links it is 14.92%, for the optimum cases in the preceding tables (1400-byte UDP and 85-byte TCP payloads, respectively). Increased losses mean more frequent invocations of TCP congestion avoidance algorithms. Reducing transmission rates due to mistaking errors for congestion has the side effect of underutilizing the wireless link. Since cellular links operating at bandwidths of around 10 Kbit/s are most likely the bottlenecks of an end-to-end path, underutilizing them means reduced end-to-end performance. In addition, cellular links usually serve one user with few transport connections, so their bandwidth will most likely be wasted, even though the user is billed based on (physical) connection time. WLANs, despite their higher bandwidths, are also likely to be the bottlenecks of end-to-end paths, but given enough users on the shared medium, the bandwidth released by one TCP connection may be absorbed by others.

Another problem with multihop paths is that TCP retransmissions traverse the path from the sender to the wireless link again, even if they already arrived successfully there previously. If a single wireless link is located on the receiver's end of the path, these retransmissions waste wired link bandwidth and delay recovery. If more than one link is wireless, datagrams are retransmitted over wireless links that were already crossed successfully, reducing their precious throughput (goodput). The combined effect of mistaking wireless loss for congestion and end-to-end recovery is more pronounced on longer paths that employ large TCP windows to keep data flowing through the network until acknowledgments return. This is because the TCP transmission window is reduced after losses are detected, leaving the end-to-end path underutilized until the window grows back to its appropriate size, which takes more time for long paths. Whenever multiple losses occur during a single window (more likely with larger windows), TCP performance is further reduced.

Figure 2 and Table III show the performance of TCP over a single hop (LAN) versus a multihop Wide Area Network (WAN) path (the results are taken from [2]), in absolute terms and as a percentage of the nominal bandwidth. One end of the path is on a wireless LAN transmitting 1400 byte frames with an FER of about 2.3%, a situation slightly worse than the one described in the preceding discussion. The nominal bandwidths hold in the absence of any congestion or wireless link losses, while the TCP

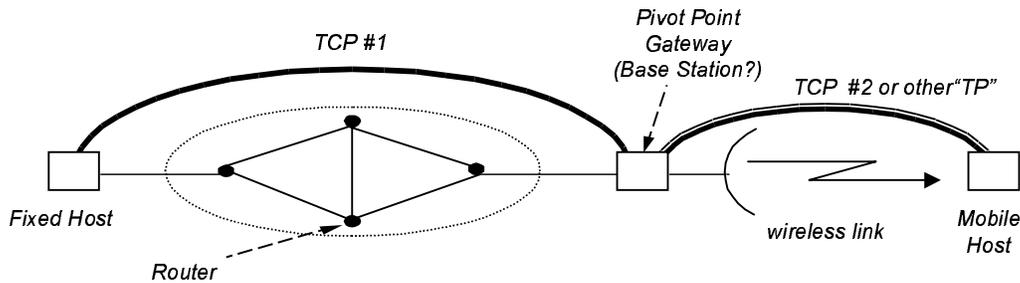


Fig. 3. Illustration of a simple Indirect TCP configuration.

throughput numbers are based on simulating errors using a simple independent error model. The difference between the two TCP variants is that the improved protocol can recover from more than one error during a single-error recovery period. As a result, in the high-error environment of the wireless link, the improved variant achieves higher throughput and depicts smaller differences between the LAN and WAN cases. In all cases, the throughput degradation is at least 10 times the FER.

III. PERFORMANCE ENHANCEMENTS FOR INTERNET PROTOCOLS

A. Approaches at the Transport Layer

Most of the work on Internet protocol performance over wireless links has focused on TCP, since it is the most popular Internet transport protocol. The root cause of the reported problems is the TCP assumption that all losses are due to congestion, so that loss detection triggers congestion avoidance procedures. Thus, the frequent losses seen on wireless systems, whether due to communications errors or pauses during handoffs, cause TCP to underutilize the bandwidth-starved wireless links, dramatically reducing end-to-end performance. Longer paths further delay end-to-end recovery, aggravating these performance problems.

A direct approach to avoid TCP performance problems is to modify TCP itself, since it is TCP assumptions that cause the problems. In addition, TCP, being an end-to-end protocol, requires only the two communicating peers to upgrade their software in order to take advantage of improvements. Solutions depend on the cause of losses: handoffs or errors. During handoffs connectivity is temporarily lost, and many times a timer has to expire before recovery can be initiated. To avoid long communication pauses after handoffs, one approach is to invoke fast retransmission right after a handoff completes, instead of after a timeout. This requires signaling to notify the transport layer about handoff completion [6]. Invoking full congestion recovery procedures after every handoff still reduces throughput, so an alternative scheme attempts to detect whether loss is due to mobility or congestion by exploiting mobility hints from lower layers [10]. If the loss is classified as due to congestion, both slow start and congestion avoidance phases [8] are invoked. If the loss is classified as due to mobility, only slow start is invoked, reducing recovery time.

Unlike handoffs, where some congestion avoidance procedure is needed to probe the state of the new link, with losses due to errors we should skip congestion avoidance completely. Since these losses are local, end-to-end retransmissions unnecessarily delay recovery. One approach to improved error recovery is to split TCP connections at *pivot points*, that is, those routers on the path that are connected to both wireless and wired links (indirect TCP; see Figure 3). One instance of TCP executes over each wired part, while either another instance of TCP or a special-purpose transport layer protocol executes over each wireless part [1], [14]. These segments are bridged by a software agent at each pivot point that also translates between the different protocols, if required. As a result, losses at wireless segments do not trigger end-to-end recovery. When TCP is used over the wireless segments, it can recover fast due to the short paths involved. Alternatively, a transport protocol with more appropriate mechanisms (such as selective repeat) may improve performance [14]. The ideal scenario for this approach is a path with wireless endpoints, where the pivot points are the routers connecting the wireless network to the Internet. Only the pivot points and the wireless hosts need software upgrades. Handoffs in these schemes change the endpoint but not the pivot point, so that the agent at the pivot can establish new connections as needed and speed up recovery after a handoff. For UDP, datagrams lost during communications pauses can be retransmitted when connectivity is reestablished, as in M-UDP [4]. For TCP, the pivot agent can choke the remote sender by reducing the advertised window to zero, as in M-TCP [5]. This causes the sender to go into *persist* mode, during which it periodically probes the receiver's window while all pending timers are frozen. However, shrinking the advertised window violates TCP guidelines.

TCP modifications are not perfect solutions. Schemes that deal with handoff problems modify the transport layer only at the endpoints, but they still face performance degradation due to wireless errors. To detect handoffs, coupling is introduced between layers to solve an isolated and localized problem, with improvements that are applicable only to TCP. Split (or indirect) TCP approaches [1], [14] deal with both mobility and link errors, but they require modifications both at the endpoints and at the pivot points, the latter generally being beyond user control. New transport protocols compatible with TCP are needed to maximize performance over wireless segments, although the split connection idea is applicable to any protocol. The agents at the pivot points are complex: They must translate semantics and synchronize connections despite communications errors and pauses. Performance is questionable for wireless segments not at path ends or for multiple-link wireless segments. Finally, the end-to-end

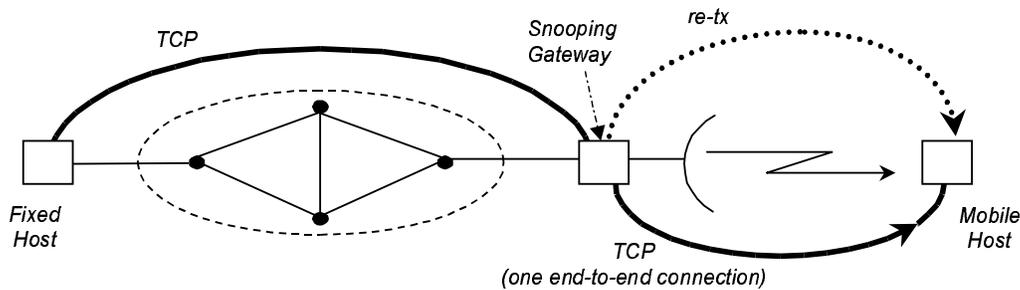


Fig. 4. Illustration of snoop TCP.

semantics of the transport layer may be violated, so applications that need end-to-end reliability must use additional protocols above TCP. Even worse, applications do not know that the end-to-end semantics are violated.

B. Approaches below the Transport Layer

The main alternative to modifying TCP or other end-to-end protocols is to modify the link layer protocol that operates over the wireless link so as to hide losses using local recovery mechanisms. CT systems offer nontransparent Radio Link Protocols (RLPs) that enhance link reliability with this exact goal in mind. Another approach, applicable to both WLANs and CT systems, is adding some link layer functionality to IP to take care of local recovery [3]. IP datagrams carrying TCP data are buffered at the hosts that transmit them over wireless links, and are retransmitted if they are not acknowledged within a short period of time or if multiple acknowledgments for previous data are received. The local error-recovery module snoops on all IP datagrams to gather TCP data and acknowledgment information. Buffered TCP data that need to be retransmitted are inserted into the data stream transparently to the receivers. By leveraging existing TCP messages, snoop TCP (see Figure 4) avoids additional control exchanges and simplifies integration with TCP. Conceptually, this is a link layer mechanism, as it involves single-link error recovery.

Interestingly, employing TCP acknowledgments for feedback, besides violating protocol layering, causes this approach to work only in the direction towards the wireless host. In this direction, TCP acknowledgments are received at the wired endpoint after a one-hop delay only, and retransmissions may be made before timers expire at the other end of the end-to-end path. In the reverse direction, TCP acknowledgments are returned after the round-trip delay for the whole path has nearly elapsed, so the wireless host cannot retransmit lost segments soon enough. To make retransmission effective in that direction, local control exchanges are needed (as required by protocol layering). This is also the case for wireless links that are not at the edges of end-to-end paths, where both sides need link layer control exchanges to initiate retransmissions on time. Overall, this local recovery scheme performs better than split transport layer schemes under wireless link errors [2]. Avoiding the violation of transport layer semantics comes at the cost of violating protocol layering, however. The advantages of TCP and link layer coupling is reduced link layer overhead and avoidance of conflicts between local and TCP retransmissions [7], but the scheme may require modifications whenever TCP is modified, and does not work for any other protocols.

CT system RLPs on the other hand, avoid layering violations, but run the risk of retransmitting data that the transport layer will retransmit anyway, hence the approach of the IS-95 RLP of limited recovery [9]. In addition, they may do more than what is required for applications that do not require complete reliability. Link layer schemes in general have the advantage over end-to-end schemes of working at the local level, with intimate knowledge of the underlying media and low round-trip delays that allow fast recovery, although they cannot deal with handoffs where multiple links are involved. The problem is how much to enhance the underlying link without getting in the way of higher layers. The goal is to offer adequate recovery to ease the task of reliable transports and to allow the realistic operation of unreliable transports, which assume only rare losses. Even if vendors supply fine-tuned link layer protocols with their devices, it is hard to design a single protocol that can cater to the needs of multiple existing and future transport layers and applications.

IV. THE FUTURE: CHALLENGES AND OPPORTUNITIES

A. Wireless System Evolution

One of the most attractive characteristics of wireless systems is that they enable mobility. Cellular systems allow efficient sharing of the frequency spectrum via reuse and offer wide area mobility with reasonable power requirements. Since each cell is connected to other networks via its own base station, mobility between cells implies a need for handoffs of mobile devices between base stations. These handoffs cause pauses in communication while the mobile completes base station changes. When data are lost during handoffs, reliable Internet protocols such as TCP may be tricked into mistaking them for congestion [6]; this is the case even if they are not really lost but only delayed. Some future cellular systems are expected to employ smaller cells (*picocells*) to offer higher data rates and support more users at the same time. Picocells, due to their smaller area, will require

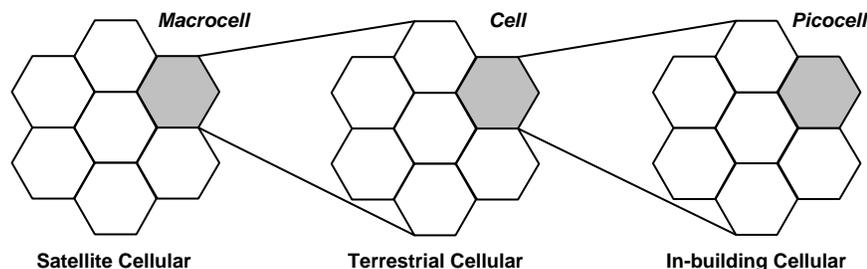


Fig. 5. Overlay networks: hierarchical cellular system.

a dense mesh of base stations, which will only be justifiable within buildings or other densely populated areas. Thus, current cellular systems with standard-sized cells will still provide coverage in areas with fewer users, while sparsely populated areas that do not warrant the cost of a terrestrial cellular infrastructure will be covered by satellite systems. The large area under a satellite beam will then form a *macrocell*. In low-orbit satellite systems handoffs will still occur, but mostly due to satellite rather than user movement. This will give rise to a hierarchical cell structure, as depicted in Figure 5. Higher-level cells are overlaid in areas with (presumably) more users by multiple lower-level cells. Users can use the highest bandwidth system available in each location and move from cell to cell within the same system (performing horizontal handoffs, as in existing cellular systems) or from one system to another (performing new-style vertical handoffs), depending on coverage.

Hierarchical systems will challenge us with additional handoff-induced problems. In the picocells, handoff frequency will increase, and since the per-user bandwidth will be higher, more data will potentially be lost during handoffs, urging for faster handoffs and fast, localized, recovery. In parallel, since handoffs will be possible between different technologies, connections will face two levels of link performance variability. First, short-term performance variations will be caused by environmental changes such as fading, in the same manner as today, with details depending on the technology used at each link. Adapting to these link-specific variations is much easier locally, at the link layer, where the peculiarities of each medium are known. Second, medium-term performance variations will be caused by handoffs between different technologies (picocellular, cellular and macrocellular). These handoffs will dramatically change the performance parameters of the wireless part of the end-to-end path, as each type of link will have its own characteristics. Adapting to such variations is probably only feasible at an end-to-end layer. Since handoffs take place between two separate links, dealing with handoff outages is not a local task, although the link layer could help by providing information to higher layers, for example, notifications of handoffs and disconnections or current link properties after a vertical handoff. These systems illustrate how similar problems (horizontal and vertical handoffs) with different parameters may be best treated at distinct layers.

B. Goals for Protocol Evolution

If we examine the performance problems of existing protocols and the shortcomings of proposed enhancements for wireless links, a few goals for further research emerge.

- Masking wireless errors with nontransparent RLPs is not adequate for all types of transport protocols and applications. For example, UDP-based real-time applications typically prefer sending new packets to retransmitting old ones. Similarly, transport layer modifications, despite using the most appropriate mechanism for any given transport protocol, are protocol specific and not easily extensible to hierarchical cellular systems. We thus need a multiprotocol approach to wireless link enhancements.
- While mobility problems cannot be dealt with at the link layer (whose scope is isolated links), wireless link losses can be efficiently recovered from using local mechanisms [2]. Besides handling handoff pauses, higher layers will have to adapt to the medium-term link variabilities caused by (vertical) handoffs to different networks in hierarchical cellular systems. Thus we need to deal with each wireless link problem at the appropriate layer.
- Since applications have varying requirements, multiple transport layers should be supported on the Internet without having to worry about the details of each specific link. Isolating transport layer protocols from link peculiarities can be achieved by link layer protocols that support flexible and general services rather than specific transport protocols. Thus protocol layering and isolation should be adhered to during design.
- The proliferation of wireless links will eventually force many existing protocols to review their design assumptions and modify their mechanisms. New protocols may also emerge that are more flexible in dealing with wireless links and mobility, by exploiting adaptive mechanisms. Such protocols would be able to exploit advanced link layer services in order to offer enhanced functionality to their own users. Thus lower-layer protocols should consider future higher-layer needs.

In our view, neither link layer nor transport layer approaches are sufficient by themselves to solve the problems presented here. While transport protocols are more aware of end-to-end application requirements, link protocols are better positioned to handle

local issues in a fast and transparent manner. A link layer handling link-specific problems, such as wireless errors, isolates higher layers from physical layer peculiarities and evolution, adapting its mechanisms to prevailing conditions. Intimate knowledge of link-specific attributes, including actual hardware status, also allows the link layer to inform higher layers of interesting events (e.g., handoffs) and performance metrics of a link, so that they can in turn adapt their mechanisms accordingly. For example, in the Linux OS we could use pseudodevice files to export performance metrics and software signals for event notifications. Upward propagation of information through the protocol stack could thus ease the introduction of more adaptive and versatile protocols at each layer. On the other hand, each higher-layer protocol would like to see different enhancements to the raw link offered by the link layer. Instead of offering a different protocol for each type of service, a single link layer could support multiple services that cater to various requirements and needs, all sharing the link simultaneously [12]. Choosing between multiple services requires propagation of additional information downwards through the protocol stack. We thus advocate a more synergistic approach between layers, where generic end-to-end requirements are supported by customized local mechanisms. We believe that making the interfaces between protocol layers richer in information content could pave the way for smarter and better-performing future Internet protocols.

V. SUMMARY

We presented the performance problems faced by Internet protocols when deployed over wireless networks. We also discussed existing performance enhancements at various protocol layers. A scenario for the future evolution of wireless communications was then presented, and we concluded by presenting some design goals for further protocol enhancement to address the shortcomings of existing approaches and the new requirements imposed by emerging systems. Overall, we believe that better cooperation and communication between protocol layers are key requirements for improving Internet protocol performance in the future.

REFERENCES

- [1] A.V. Bakre and B.R. Badrinath. Implementation and performance evaluation of Indirect TCP. *IEEE Transactions on Computers*, 46(3):260–278, March 1997.
- [2] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6): 756–769, December 1997.
- [3] H. Balakrishnan, S. Seshan, and R.H. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks*, 1(4):469–481, 1995.
- [4] K. Brown and S. Singh. M-UDP: UDP for mobile cellular networks. *Computer Communications Review*, 26(5):60–78, October 1996.
- [5] K. Brown and S. Singh. M-TCP: TCP for mobile cellular networks. *Computer Communications Review*, 27(5):19–43, October 1997.
- [6] R. Caceres and L. Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE Journal on Selected Areas in Communications*, 13(5):850–857, June 1995.
- [7] A. DeSimone, M.C. Chuah, and O.C. Yue. Throughput performance of transport-layer protocols over wireless LANs. In *Proceedings of the IEEE GLOBECOM '93*, pages 542–549, December 1993.
- [8] V. Jacobson. Congestion avoidance and control. *Computer Communications Review*, 18(4): 314–329, August 1988. (SIGCOMM '88 Symposium: Communications Architectures and Protocols, Stanford, CA, August 1988.)
- [9] P. Karn. The Qualcomm CDMA digital cellular system. In *Proceedings of the USENIX Mobile and Location-Independent Computing Symposium*, pages 35–39, August 1993.
- [10] P. Manzoni, D. Ghosal, and G. Serazzi. Impact of mobility on TCP/IP: An integrated performance study. *IEEE Journal on Selected Areas in Communications*, 13(5):858–867, June 1995.
- [11] B.D. Noble, M. Satyanarayanan, G.T. Nguyen, and R.H. Katz. Trace-based mobile network emulation. *Computer Communication Review*, 27(4):51–61, October 1997. (Proceedings ACM SIGCOMM '97 Conference, Cannes, France, September 1997.)
- [12] G.C. Polyzos and G. Xylomenos. Enhancing Wireless Internet Links for Multimedia Traffic. In *Proceedings International Workshop on Mobile Multimedia Communications (MoMuC'98)*, Berlin, Germany, October 1998.
- [13] G. Xylomenos and G.C. Polyzos. TCP and UDP performance over a wireless LAN. In *Proceedings IEEE Conference on Computer Communications (INFOCOM '99)*, New York, March 1999, pages 439–446.
- [14] R. Yavatkar and N. Bhagawat. Improving end-to-end performance of TCP over mobile internetworks. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 146–152, December 1994.