CHAPTER 1

Introduction and Outline

The Internet has evolved into *the* communication medium of the future. It will not be long before virtually all people-to-people, people-to-machine, and machine-to-machine communication are carried end-to-end in *Internet Protocol (IP)* [RFC791], [RFC2460] packets. The recent tremendous growth of the Internet in terms of connected hosts is only matched by the similar growth rate of cellular telephone subscribers. While most hosts on today's Internet are still wired, the next *big* wave of hosts has yet to hit the Internet. We believe that the predominant Internet access of the future will be wireless. Not only every cellular phone, but the majority of general communication devices will have: (1) an IP protocol stack and (2) a wireless network interface.

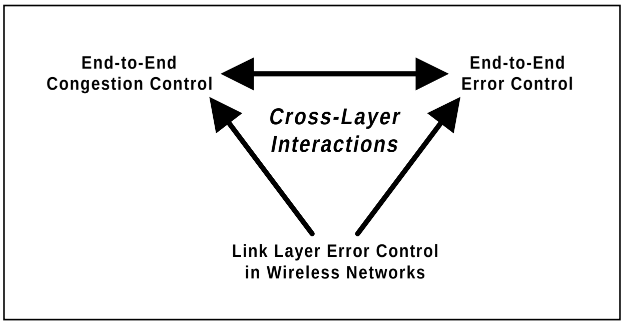

Figure 1-1: Cross-Layer Interactions in Wireless Networking.

Wireless networking and more specifically, the problems related to protocol performance of "IP over Wireless", are high priority research topics in both academia and industry. Yet, a number of difficult problems remain unsolved. The root of these lies in inefficient cross-layer interactions, caused by events occurring at the link layer that affect the performance of end-to-end error and congestion control schemes (see Figure 1-1). Those events are packet losses due to transmission error and increased packet transmission delays caused by link layer error control.

The goal of this dissertation is to study inefficient cross-layer interactions in wireless networking and to develop and evaluate a solution that eliminates these inefficiencies.

Internet traffic today is still largely generated by applications that use the *Transmission Control Protocol (TCP)* [RFC793] as the underlying communications protocol. The most popular Internet applications, namely World Wide Web (WWW), e-mail, and file transfer, all rely on TCP. In the Internet backbone studied in [TMW97], the authors find that TCP averages about 80 percent of the flows. Related traffic analysis studies find similar numbers. This motivates why TCP has been the focus when it comes to studying inefficient cross-layer interactions in wireless networking.

The solution we develop in this dissertation is, however, *not* “TCP-specific”, but is independent from any specific protocol. Nevertheless, to demonstrate the feasibility of our approach, most of our measurements, analysis, and implementation work is based on TCP. Our solution comprises new end-to-end *and* link layer mechanisms that make the protocol implementations on those layers more “intelligent” and robust. For flows that are loss responsive and fully-reliable, in particular those based on TCP, our solution eliminates all known inefficient cross-layer interactions. It provides optimal end-to-end performance over a wide range of conditions of the wireless link while efficiently utilizing radio resources. Beyond solving the problems for such flows, our solution provides a framework that accommodates loss responsive real-time flows, e.g., semi-reliable, or error-resilient flows.

Related work suggests solutions that are either pure end-to-end (e.g., [SF98]), leaving unexploited the potential of link layer error control schemes to optimize end-to-end performance, or only suggest link layer mechanisms (e.g., [BBKT97], [ES98], [Kar93]), none of which is capable of solving all inefficient interactions with end-to-end control schemes. Most prior work (e.g., [BB95], [BK98], [BS97], [DMT96], [HK99], [KRLKA97]), however, suggests *Performance Enhancing Proxies (PEPs)* that couple link layer and end-to-end control schemes to solve inefficient cross-layer interactions. These solutions violate *the* fundamental design principle in data communications, *protocol layering*, by requiring access to transport layer headers by the PEPs. Our approach is fundamentally different in that we do not require - in fact, we argue against - such cross-layer couplings. The key advantages of our solution over PEP-based approaches are (A) its independence from transport (or higher) layer protocol semantics making it a “non-TCP-specific” solution, (B) the possibility of co-existence with network layer encryption, e.g., *IPsec* [RFC2401], and (C) that no per-flow state needs to be maintained in the network making it more scalable. The main contributions of this dissertation are the following:

1. The new concept of *flow-adaptive wireless links* and its application for fully-reliable flows. This work has been published in [LR99], [LRKOJ99], and [LKJK00].
2. Two new mechanisms for reliable end-to-end protocols, the *Eifel algorithm* and the *Eifel retransmission timer*. The former has been published in [LK00] while the latter is described in [LS99]. We have implemented both mechanisms for TCP, and refer to that implementation as *TCP-Eifel* that we have made publicly available [Lud99c].
3. New measurement methods and tools we developed throughout our work.

The remainder of this chapter describes and motivates the outline of this dissertation.

Chapter 2 provides required background. After we introduce related terminology in Section 2.1, the following three sections review those functions of the control schemes shown in Figure 1-1 that are relevant for our studies. Since all our “real-world” measurements in wireless networks were carried out using *GSM-CSD*, the *Circuit-Switched Data (CSD)* service implemented in the *Global System for Mobile communications (GSM)*, we explain that network in more detail. In Section 2.5 we describe all inefficient cross-layer interactions in wireless networking that are known in literature. Related work is reviewed and evaluated in Section 2.6. We present a brief motivation and outline of the approach taken in this dissertation in Section 2.7.

Chapter 3 explains the analysis methodology we applied and the tools we developed to obtain the results presented in Chapter 4 and Chapter 5. We first motivate why we have mostly chosen a measurement-based analysis approach. In Section 3.1, we then explain the methodology we use in Section 4.3 to evaluate the benefit of link layer error recovery for reliable flows. We explain how we capture and analyze the error characteristics of the GSM-CSD wireless link. Our ReTracer tool is described, which we developed to reverse-engineer target metrics such as throughput given certain parameters like the link layer frame size. In Section 3.2, we explain the methodology we use in Section 4.2 to detect inefficient cross-layer interactions between TCP and the link layer error control implemented in GSM-CSD. We describe the tools we developed for that purpose: `rlpdump`, an event logging tool for the reliable link layer protocol implemented in GSM-CSD, and MultiTracer, used to correlated events on different protocol layers. There we also explain how to interpret TCP trace plots, which we often use to illustrate certain effects, problems, or solutions. In Section 3.3, we explain the methodology we use in Section 5.1 to study the problem of competing error recovery for the case of TCP, and to develop the *Eifel algorithm* in Section 5.2 that eliminates this problem. We explain how we used the `hiccup` tool that we developed to reproduce inefficient cross-layer interactions in a “non-wireless” but controllable network environment. In Section 3.4, we explain the methodology we use in Section 5.3 to study and reveal the problems of TCP-Lite’s retransmission timer. We use the same model in Section 5.4 to develop the *Eifel retransmission timer* that eliminates those problems. We explain the model that we developed to analyze those end-to-end retransmission timers, and the measurement setup we used to validate the correctness of the model. In Section 3.5, we provide a summary of the chapter.

Chapter 4 introduces the concept of flow-adaptive wireless links, and validates it for fully-reliable flows. In Section 4.1, we explain that concept and discuss its deployment concerns, and possible implementation alternatives. The key idea is that network end-points use the IP layer as a level of indirection through which their QoS requirements are signalled to each link layer along the path, on a per packet basis. This allows for a (wireless) link layer to adapt its error control schemes to meet those requirements while minimizing radio resource consumption. We argue against and illustrate the problems of running low link layer error recovery persistency for fully-reliable flows. In Section 4.2, we verify through measurements, that our solution of flow-adaptive wireless links eliminates all known inefficient cross-layer interactions except for the problem of competing error recovery. This study also delivered early indications that the retransmission timer implemented in TCP-Lite is too conservative. In Section 4.3, we show how the GSM-CSD wireless link can be adapted to optimize the end-to-end performance of bulk data flows. We use this case study to demonstrate that link layer error recovery over wireless links is essential for reliable flows to optimize their end-to-end performance. In Section 4.4, we provide a summary of the chapter.

Chapter 5 provides solutions to the remaining problems we identified in Chapter 4. In Section 5.1, we identify the retransmission ambiguity in TCP as the root of the problems caused by competing error recovery. In Section 5.2, we develop the *Eifel algorithm* that uses extra information in the TCP acknowledgements to resolve the retransmission ambiguity, and show how this is used to eliminate those problems. The Eifel algorithm only requires changes to the TCP sender implementation, not to the protocol itself. Thus, given this backwards compatibility and the fact that it does not change TCP’s congestion control semantics, the new algorithm can be incrementally deployed. In Section 5.3, we analyze TCP-Lite’s retransmission timer, and reveal a number of problems related to its definition and implementation. This explains why we had suspected that this timer is too conservative. In Section 5.4, we then propose an alternative retransmission timer, which we call the *Eifel retransmission timer*, that eliminates those problems. We demonstrate that the Eifel retransmission timer is a more precise predictor of an upper bound for the path’s RTT. Combining both, the Eifel algorithm and the Eifel retransmission timer, we propose a new approach to designing retransmission timers. The idea is to let the timer become increasingly aggressive while adapting it to the measured fraction of spurious timeouts. We validated the correctness of our analysis by showing that the model- and the measurement-based approach leads to the same results. In Section 5.5, we provide a summary of the chapter.

Chapter 6 concludes this dissertation by summarizing our main results and outlining related but unsolved research problems that merit further exploration.