CHAPTER 6

Conclusion

A commonly used link layer design philosophy says “leave the link layer dumb but simple”. The term “dumb” in this context stands for independence from any kind of higher layer information about a flow’s nature[1]. This approach has been explicitly applied to the design of many wireless networks that have been deployed in recent years, e.g., IS-95 [Kar93] and WLAN [IEEE802.11]. The “end-to-end argument” [SRC84] is usually quoted to advocate this design philosophy. In a nutshell, the reasoning is that appropriate error control can *only* be implemented on an end-to-end basis, since only the network end-points have sufficient information to perform this task. Despite its attractiveness, we showed that the simplicity of such link layers comes at a cost: reduced end-to-end performance due to inefficient cross-layer interactions and a waste of radio resources. Earlier work addressing these problems either fails to eliminate all inefficiencies, or resorts to Performance Enhancing Proxies (PEPs) that are dependent on transport (or higher) layer protocol semantics and cannot interoperate with network layer encryption as long as they are untrusted.

The concept of *flow-adaptive wireless links* is a new link layer design philosophy we have proposed. It eliminates all known inefficient cross-layer interactions, with the exception of the problem of competing error recovery which we solved with the *Eifel algorithm*. We argued why carrying a network end-point’s QoS requirements as part of the flow’s packet headers and accordingly adapting link layer error control, is orthogonal to the “end-to-end argument”. Moreover, our solution has the key advantage that it avoids PEPs and their drawbacks. We showed the feasibility of our solution by applying it for the class of fully-reliable flows. We demonstrated that highly persistent link layer error recovery is required to optimize the end-to-end performance provided by such flows while efficiently utilizing radio resources.

1. TCP/IP header compression is the exception.

Our measurement results revealed that the GSM-CSD wireless link is over-protected with forward error correction, and that the default value standardized for the link layer error recovery persistency is too low. We also demonstrated that the throughput of the GSM-CSD wireless link can be improved by up to 25 percent by increasing the (fixed) link layer frame size to reduce the relative per packet overhead. These results highlight the importance of measurement-based analysis in wireless networking. It would have been difficult to obtain those results through simulations. The reason is that such an analysis is highly dependent on the error characteristics of the wireless link that are difficult to model with sufficient accuracy. We therefore believe that for wireless systems it is particularly important that prototypes are developed early in the design process so that measurement-based performance studies can be carried out.

We proposed two new mechanisms, the Eifel algorithm and the Eifel retransmission timer, that improve the performance of end-to-end error recovery protocols independent of whether they are run over wireless links. Both mechanisms have been implemented and evaluated on the basis of TCP. The implementation, TCP-Eifel, is publicly available [Lud99c], as well as the spreadsheet-based model we developed for the timer analysis [Lud99a].

The Eifel algorithm uses extra information in the TCP header to eliminate the problems caused by competing error recovery. It only requires changes to the TCP sender implementation, and can be incrementally deployed. In situations where competing error recovery is likely to occur, e.g., in packet-radio networks where the wireless connectivity is often intermittent, the algorithm can improve the end-to-end throughput up to several tens of percent. Another key novelty is that the Eifel algorithm provides for the implementation of optimistic retransmission timers, because it reduces the penalty of underestimating the round-trip time to a single spurious retransmission (in the common case).

The Eifel retransmission timer takes advantage of this feature by becoming increasingly optimistic while adapting to the measured fraction of spurious timeouts. In addition, it eliminates four major problems of the current de facto implementation of TCP's retransmission timer which we have revealed in our work. We demonstrated that the Eifel retransmission timer is a more precise predictor of an upper bound for a path's round-trip time while reacting quicker to packet losses. For network-limited bulk data transfers that do not have enough packets in flight to trigger TCP's fast retransmit algorithm, the Eifel retransmission timer increased the end-to-end throughput by 30 percent in our measurement setup. We validated the correctness of our analysis by showing that the measurements based on the timer implementations yielded the same results that we had predicted based on our timer models.

Optimizing the end-to-end performance of loss responsive *real-time* flows over paths that include wireless links remains an open research problem. Although, we argued that the concept of flow-adaptive wireless links provides an appropriate framework that accommodates such flows, we have not further studied this problem. We are also not aware of related work that addresses it. Below, we outline two issues that need to be resolved in this context.

The performance of loss responsive real-time flows can suffer from the same, and potentially more, inefficient cross-layer interactions that we described in our work. Thus, it needs to be studied how to appropriately adapt link layer error control schemes to minimize such inefficiencies. For example, "how much delay may link layer error control introduce before end-to-end delay bounds are exceeded?", or "which error loss rates can the network end-point tolerate before it starts underestimating the available bandwidth?".

Another fundamental problem is the lack of support for bit-error-resilient audio/video codecs. While these are state-of-the-art in traditional circuit-switched wireless networks like GSM, they cannot be used in today's Internet. The problem is that both transport *and* link layer protocols implemented in the Internet perform error detection, i.e., packets received by an Internet application is always bit-error-free. This effectively disables the key feature of a bit-error-resilient codec, and can greatly reduce user-perceived (audio/video) quality. Solving this problem requires that a transparent service is provided both end-to-end and at the link layer. Yet, transport protocols that use weak checksums, e.g., UDP, rely on strong link layer error detection. Thus, network end-points need to be able to request or disable such a transparent service.

We believe these problems merit further exploration because finding feasible solutions is crucial for true "IP over wireless" to become a reality.