To Appear in Proceedings of ACM/IEEE MobiCom 99.

Optimizing the End-to-End Performance of Reliable Flows over Wireless Links

Reiner Ludwig
Ericsson Research
Herzogenrath, Germany

Almudena Konrad, Anthony D. Joseph
Computer Science Division
University of California at Berkeley

Abstract

We present the results of a performance evaluation of link layer error recovery over wireless links. Our analysis is based upon a case study of the circuit-switched data service implemented in GSM digital cellular networks. We collected a large set of block erasure traces in different radio environments and used a measurement-based approach to derive real-world models of the wireless link. We show that the throughput of the GSM circuit-switched data channel can be improved by up to 25 percent by increasing the (fixed) frame size of the reliable link layer protocol. Our results also suggest that adaptive frame length control could further increase the channel throughput. In general, our case study shows that pure end-to-end error recovery fails to optimize throughput when wireless links form parts of the end-to-end path. In many cases, it leads to decreased end-to-end throughput, an unfair load on a best-effort network, such as the Internet, and a waste of valuable radio resources (e.g., spectrum and transmission power). In fact, we show that link layer error recovery over wireless links is essential for reliable flows to avoid these problems.

1. Introduction

The Internet is evolving to become *the* communication medium of the future. It will not be long before the last circuit switch is taken out of service and virtually all people-to-people, people-to-machine, and machine-to-machine communication are carried in IP [25] packets. The tremendous recent growth of the Internet in terms of connected hosts is only matched by the similar tremendous growth 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 every *thing* that communicates will have: (1) an IP protocol stack and (2) a wireless interface.

It is well known that the performance of reliable transport protocols such as TCP [26] may degrade when wireless links span the end-to-end path. However, related work has mostly focused on the problem that wireless links cause for the congestion control scheme used in most implementations of TCP. Employing a link layer error recovery scheme over the wireless link removes this problem. Furthermore, [20] shows that at least in some wireless networks - in fact the one we use for the case study in this paper - the potential problems that may result from competition between end-to-end and link layer error recovery do not exist. Given such a wireless network, we are not aware of any study that evaluates the performance of pure end-to-end error recovery versus adding link layer error recovery. This has been the key motivation for the work presented in this paper.

The key premise for our analysis is that we assume the model of a bulk data transfer based on a reliable end-to-end flow (e.g., a TCP-based flow). This is a valid assumption given the concept of *flow-adaptive* wireless links introduced in [19]. A flow-adaptive implementation of a link layer error recovery scheme can perform the flow type differentiation required to identify reliable flows. This ensures that link layer retransmissions do not interfere with unreliable, possibly real-time, end-to-end flows (e.g., UDP-based flows). The attractiveness of link layer solutions over approaches that require access to the transport layer headers in the network (e.g., [1],[2],[3],[4],[5],[11],[17],[21]), are their independence from transport (or higher) layer protocol semantics and the possibility of co-existence with any form of network layer encryption as proposed in [16].

The analysis presented in this paper is based upon a case study of the circuit-switched data service implemented in GSM (Global System for Mobile communications) digital cellular networks. Our measurement-based approach gave us the unique opportunity to derive models of the wireless link that capture the aggregate of real-world effects like noise, interference, fading, and shadowing. It also provided us with new insights into how the current system can be optimized, and suggested techniques that can be used to design future wireless links. The fact that GSM has been deployed globally and is in widespread use, highlights the relevance of our results. An unrealistic error model of the wireless channel can completely change the results of a performance analysis leading to non-optimal design decision. For wireless systems it is therefore particularly important that prototypes are developed early in the design process so that measurement-based performance studies can be performed.

The rest of this paper is organized as follows: Section 2 provides a background on the circuit-switched data service implemented in GSM; Section 3 describes the measurement platform we developed to collect *block erasure traces*, and explains our analysis goal and the methodology we used for our trace-based analysis; Section 4 presents and discusses our measurement results; and Section 5 closes with our conclusions and plans for future research.

2. Circuit-Switched Data in GSM

GSM implements several error control techniques, including adaptive power control, frequency hopping, Forward Error Correction (FEC), interleaving. In addition, the Circuit-Switched Data (CSD) service provides an optional fully reliable link layer protocol called Radio Link Protocol. We briefly describe the latter three control schemes as implemented for GSM-CSD using Figure 1. More details can be found in [22].

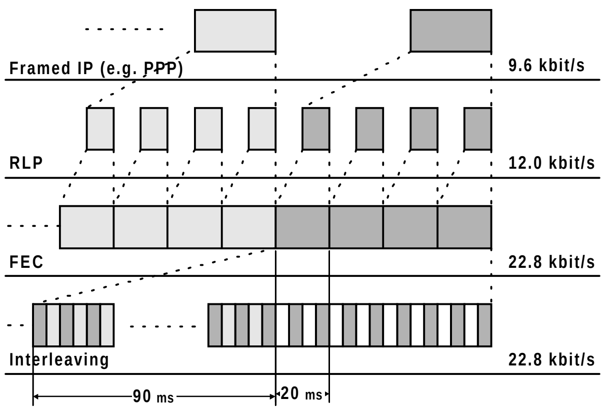

Figure 1: Error Control in GSM Circuit-Switched Data.

GSM is a TDMA-based (Time Division Multiple Access) circuit-switched network. At call-setup time a mobile terminal is assigned a user data channel, defined as the tuple (carrier frequency number, slot number). The slot cycle time is 5 milliseconds on average, allowing 114 bits to be transmitted in each slot and yielding a gross data rate of 22.8 kbit/s. The fundamental transmission unit in GSM is a *data block* (or simply *block*). The size of an FEC encoded data block is 456 bits (the payload of 4 slots). In GSM-CSD the size of an unencoded data block is 240 bits resulting in a data rate of 12 kbit/s (240 bits every 20 ms).

Interleaving is a technique that is used in combination with FEC to combat burst errors. Instead of transmitting a data block in four consecutive slots, it is divided into smaller fragments. Fragments from different data blocks are then interleaved before transmission. The interleaving scheme chosen for GSM-CSD, interleaves a single data block over 22 TDMA slots. The benefit is that a few of these smaller fragments can be completely corrupted, while the corresponding data block can still be reconstructed by the FEC decoder. The disadvantage of this large interleaving depth is that it introduces a significant one-way latency of approximately 90 ms. This high latency can have a significant negative effect on interactive protocols, as discussed in [19].[1]

The Radio Link Protocol (RLP) [7] is a full duplex HDLC-derived logical link layer protocol. RLP uses selective reject and checkpointing for error recovery. The RLP frame size is fixed at 240 bits aligned to the above mentioned FEC coder. RLP introduces an overhead of 48 bits per RLP frame yielding a user data rate of 9.6 kbit/s in the ideal case (no retransmissions)[2]. RLP transports user data as a transparent byte stream (i.e., RLP does not "know" about PPP frames or IP packets). It is important to note that, although RLP usually provides a fully reliable link, data loss can occur if the link is reset. This can have a severe impact on higher layer protocol performance as shown in [20].

3. Analysing Block Erasure Traces

In this section, we describe the measurement platform we developed to collect *block erasure traces*. We then explain our analysis goal and the methodology we used for our trace-based analysis. The measurement platform is basically the same as the one used in [20] to study the interactions between TCP and RLP.

3.1 What is a Block Erasure Trace?

In wireless networks that do not employ FEC, the error characteristics of a given wireless channel over a certain period of time can be captured by a bit error trace. A bit error trace contains information about whether a particular bit was transmitted correctly or not. The average Bit Error Rate (BER) is commonly used to describe a bit error trace. The same approach can be applied to networks that *do* employ FEC, as in GSM, but on block level instead of on bit level. Hence, a block erasure trace contains information about whether a particular data block was transmitted correctly or not. Likewise, the average BLock Erasure Rate (BLER) is commonly used to describe a block erasure trace.

It is important to emphasize that the error characteristics we have measured are only valid for the particular FEC and interleaving scheme implemented in GSM-CSD (see Section 2). Nevertheless, this data service has been deployed globally and is in widespread use. As such, we believe that our results (see Section 4) provide useful insights into how the current system can be optimized, and also suggest techniques that can be used to design future wireless links.

3.2 Measurement Platform

The architecture of the measurement platform we have developed to collect the block erasure traces is depicted in Figure 2. A single hop network running the Point-to-Point Protocol (PPP) connects the mobile to a fixed host which terminates the circuit-switched GSM connection. Various tools can then be used to generate traffic on the link (e.g., `ping` as described in [28]).

In order to collect block erasure traces, we have ported the RLP protocol implementation of a commercially available GSM data PC-Card (Ericsson DC23) to BSDi3.0 UNIX. In addition, we developed a protocol monitor for RLP (RLPDUMP) which logs whether a received block could be correctly recovered by the FEC decoder or not. This was possible because every RLP frame corresponds to an FEC encoded data block (see Section 2). Thus, a received block had suffered an erasure whenever the corresponding RLP frame was received with a frame checksum error.

1. Note, that voice is treated differently in GSM. Unencoded voice data blocks have a size of 260 bits and the interleaving depth is 8 slots.

2. Note, that the transparent (not running RLP) GSM-CSD service introduces a wasteful overhead of modem control information that also reduces the user data rate to 9.6 kbit/s.

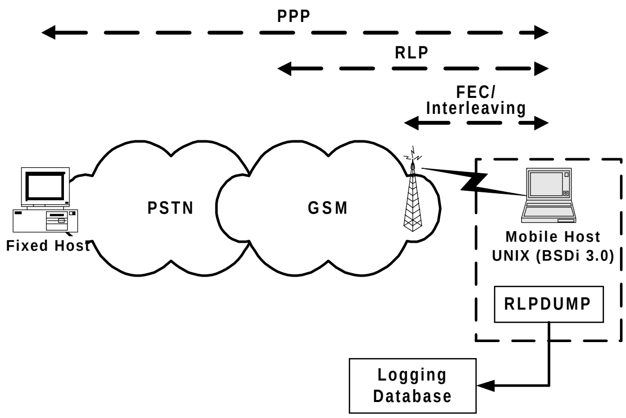

Figure 2: Current Measurement Platform.

At this stage of our work we have only performed measurements in commercially deployed GSM networks where the network-side of RLP was not accessible. This means that we could only collect downlink block erasure traces. Nevertheless, this allowed us to understand the GSM-CSD channel characteristics to a degree that was sufficient enough for our analysis. We do not believe that additional uplink block erasure traces would have changed our conclusions.

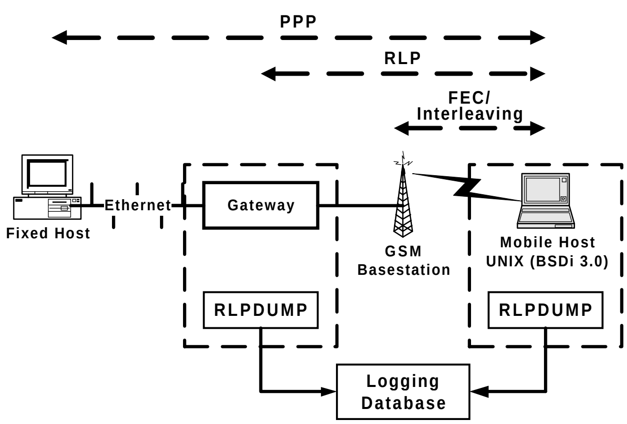

Figure 3: Future Measurement Platform.

Ultimately, we will use a stand-alone GSM basestation with a dedicated gateway that is being developed as a part of the ICEBERG project [29]. The gateway "translates" between circuit-switched (voice and data) and IP-based packet-switched traffic. As shown in Figure 3 we are currently enhancing this gateway to also terminate RLP and run RLPDUMP. With this platform we will then be able to also collect detailed uplink information and, e.g., trigger cell handovers in a controlled fashion.

3.3 Analysis Goal and Methodology

We focus on evaluating the performance of RLP as an example of a link layer error recovery scheme. The key premise for our analysis is a model of bulk data transfer based on a *reliable* end-to-end flow (e.g., a TCP-based flow). This is a valid assumption given the concept of *flow-adaptive* wireless links introduced in [19]. A flow-adaptive implementation of RLP can perform the flow type differentiation required to identify reliable flows. This ensures that link layer error recovery does not interfere with unreliable, possibly real-time, end-to-end flows (e.g., UDP-based flows). The requirements of applications that use reliable flows are simple: the application layer data object should be transferred as fast as possible *but* reliable, i.e. the transfer fails if the data object is corrupted when received by the destination application. This translates into similarly simple quality of service requirements for reliable flows: maximize throughput while the per packet delay is (almost) irrelevant[3]. For this study, we ignore interactions with end-to-end congestion control schemes. Our previous work [20] shows that this is not a problem for TCP over RLP in GSM-CSD.

Given this goal, we were not interested in identifying those factors (e.g., noise, fading, interference, or shadowing) that caused measured block erasures. Rather, we were interested in the aggregate result (similar to the approach suggested in [23]). That is, we were interested in the characteristics of block erasure traces so that we could evaluate the performance of RLP. In particular we wanted to find answers to the following questions:

- Considering the non-adaptive FEC scheme implemented for GSM-CSD: Is the fixed frame size chosen for RLP optimal or would a larger frame size yield higher channel throughput?

- Considering adaptive frame length control schemes: How "fast" do channel characteristics change in GSM-CSD? What is the margin of potential throughput improvement that adaptive frame length control could possibly yield?

Altogether we have collected block erasure traces for over 500 minutes of "air-time". We distinguish between measurements where the mobile host was stationary versus mobile when driving in a car. All stationary measurements were taken in the exact same location. The following three categories of radio environments were chosen:

A. Stationary in an area with good receiver signal strength (3-4): 258 minutes.

B. Stationary in an area with poor receiver signal strength (1-2): 215 minutes.

C. Mobile in an area with mediocre receiver signal strength (2-4): 44 minutes.

The method we used to determine the receiver signal strength is rather primitive. We simply read the mobile phone's visual signal level indicator which has a range from 1-5. In the future, we will continuously log internal signal strength measurements from the mobile phone. That way we will then be able to correlate changing receiver signal strength with the block erasure traces.

Clearly, the size of an RLP frame does not need to match the size of an unencoded data block. Any multiple of the size of an unencoded data block would have been a valid design choice. In fact a multiple of 2 has been chosen for new RLP [8] in the next generation of the GSM-CSD service which uses a weaker FEC scheme [8]. The trade-off here is that larger frames introduce less overhead per frame, but an entire frame has to be retransmitted even if only a single data block incurs an erasure. Applying a technique, we call *retrace analysis*, we studied this trade-off using the large amount of block erasure traces we had collected. Based on a given block eras-

3. In theory it would not matter in a file transfer if the first packet reached the destination last. What matters is that the file transfer is completed in the shortest amount of time. In practice this is difficult, since transport layer receiver buffers required for packet re-sequencing place a limit on the maximum per packet delay that is tolerable without affecting performance. This limit is nevertheless low.

ure trace and a given bulk data transfer size, retrace analysis is a way to reverse-engineer the value of target metrics (e.g., channel throughput or number of retransmissions) assuming a particular fixed frame size. We then iterated the retrace analysis over a range of RLP frame sizes defined in terms multiples of the data block size. That way we could for example find the frame size that would have maximized the bulk data throughput for a particular block erasure trace.

We used three different block erasure traces for our analysis. One which we call *trace_A* is a concatenation of all block erasure traces we collected in environment *A* (see above). Likewise, *trace_B* and *trace_C* are the concatenations of all block erasure traces we collected in environment *B* and *C*, respectively. We then chose an appropriate bulk data size to cover the entire trace (e.g., for *trace_B* a size corresponding to a transmission time of 215 minutes was chosen). Once the retrace analysis had reached the end of a trace it wrapped around to its beginning. In addition, we wanted to understand the impact of error burstiness, i.e., the extent to which the distribution of block erasures within a trace influenced our results. For that purpose, we artificially generated three more "non-bursty" block erasure traces, *trace_A_even*, *trace_B_even* and *trace_C_even*, which had the same BLER as the corresponding real traces, but with an even block erasure distribution.

4. Measurement Results

In this section we provide the answers to the questions we put forward in Section 3.3. We show that the throughput of the GSM-CSD channel can be improved by up to 25 percent by increasing the (fixed) RLP frame size. Our results also suggest that techniques like adaptive frame length control and adaptive FEC are worth further exploration for additional increases in channel throughput. Furthermore, we argue why in systems like GSM-CSD, pure end-to-end error recovery fails to optimize end-to-end performance.

4.1 Block Erasure Rates and Burstiness

Deriving the overall BLERs for *trace_A*, *trace_B* and *trace_C* (see Section 3.3) would have delivered little useful information. Instead, we also wanted to capture how "fast" the BLER can change over time in a given radio environment. We therefore divided each trace into one minute long *sub-traces* and treated each of those independently.

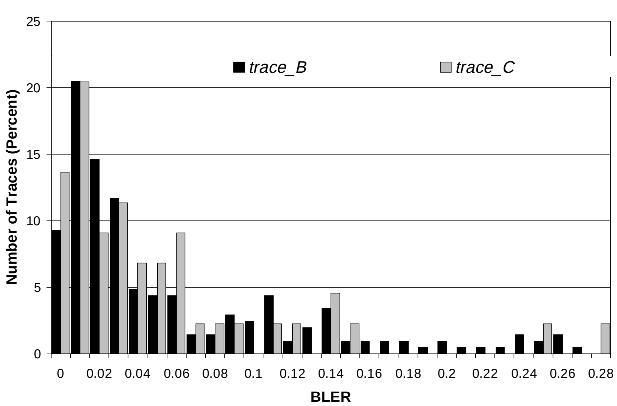

Figure 4: Measured BLERs.

Figure 4 summarizes the BLERs that we have determined in this manner. The BLERs for the sub-traces of *trace_A* are not shown because we found *trace_A* to be almost free of block erasures: over 96 percent of all sub-traces did not have a single block erasure (!) and the remaining ones had a BLER of less then 0.0025. This result shows how strongly the GSM-CSD channel is protected by FEC and interleaving, leaving little error recovery work for RLP. This is especially striking because radio environment *A* was not even ideal as it only provided a receiver signal strength of 3-4. Many radio environments often provide a maximum receiver signal strength of 5. This indicates that a weaker FEC scheme and/or a larger RLP frame size would increase the channel throughput in such radio environments. The results for *trace_B* and *trace_C* are similar but very different from the results for *trace_A*. In both of these environments, over 30 percent of all sub-traces had no single block erasure or a BLER of less then 0.01. But overall the BLERs vary considerably and can be as high as 0.28 (!). These large variations take place over time scales of one minute. This seems "slow" enough to make adaptive error control schemes applicable even within the same radio environment (e.g., environment *B*). This is an important result because otherwise such schemes would only be effective if the mobile user changed location to a different radio environment. The reason is that adaptive error control schemes can only adapt with a certain latency, which depends on the delay required to feedback channel state information. In our future work, we will study the potential of adaptive frame length control (e.g., proposed in [6] and [18]) as a technique to increase channel throughput. This decision is partly driven by our measurement-based analysis approach and the fact that we are not able to implement schemes like adaptive FEC in our testbed (see Section 3.2). Evaluating adaptive FEC schemes usually requires simulation-based analysis as, e.g., studied in [24] for the case of the GSM voice service.

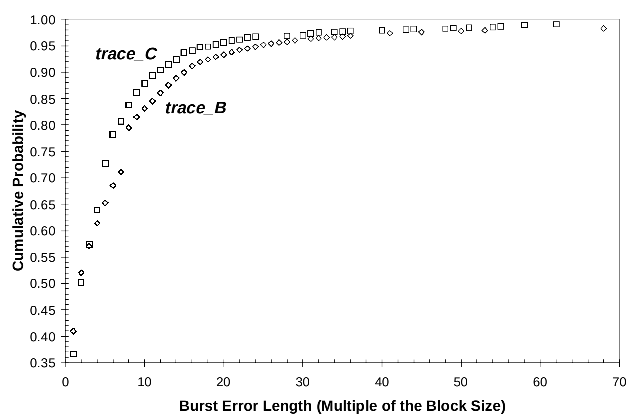

Figure 5: Burst Error Length Distribution.

In an initial attempt to quantify the error burstiness of the GSM-CSD channel, we determined the cumulative distribution function for the burst error lengths, i.e., the number of consecutive blocks that suffered an erasure, for *trace_B* and *trace_C* (see Figure 5). There was no point in showing the distribution for *trace_A* as it was basically error-free. The problem with the model shown in Figure 5, though, is that it does not sufficiently describe error burstiness. As seen in both graphs, over 50 percent of all burst errors are only 1 or 2 blocks long. It can also be seen that longer error bursts are more common when the mobile host is stationary, e.g., in *trace_C* less than 5 percent of all error bursts are larger than 18 blocks whereas in *trace_B* this number goes up to 26. However, the model does not show whether the block erasures occurred in clusters or were isolated, i.e., the correlation between error bursts is not captured by this simple model. In the following section we show how the (fixed) frame size, which maximizes channel throughput, can be used as a metric to quantify error burstiness.

4.2 Error Burstiness Allows Larger Frames

The results from the preceding section show that in many GSM radio environments, a higher channel throughput could be achieved by increasing the RLP frame size. Those results also indicate that - given a non-adaptive FEC scheme - an optimal “on size fits all” RLP frame size does not exist. Nevertheless, we wanted to determine the fixed RLP frame size that maximized channel throughput in the three radio environments *A*, *B*, and *C*. This is relevant because it indicates the margin of potential throughput improvement that adaptive frame length control could possibly yield. The implementation complexity of such techniques must be justified with substantial performance improvements. Thus, if the margin was too small, it would not be worthwhile to continue studying algorithms for adaptive frame length control in the current GSM-CSD. For that purpose, we performed the retrace analysis described in Section 3.3. Figure 6 shows that an optimal frame size of 1410 bytes would have yielded a throughput of 1420 bytes/s for *trace_A* and a frame size of 210 bytes would have maximized throughput to 1290 bytes/s for *trace_C*. The results for *trace_B* are so close to those of *trace_C* that we do not show them here. However, the gradual performance improvements in the case of *trace_A* rapidly decrease above a frame size of 210 bytes (it still would have yielded a throughput of 1390 bytes/s). This is important for our future work as it indicates that for an adaptive frame length control algorithm it would probably be sufficient to adapt the frame size in a range of about 30-210 bytes.

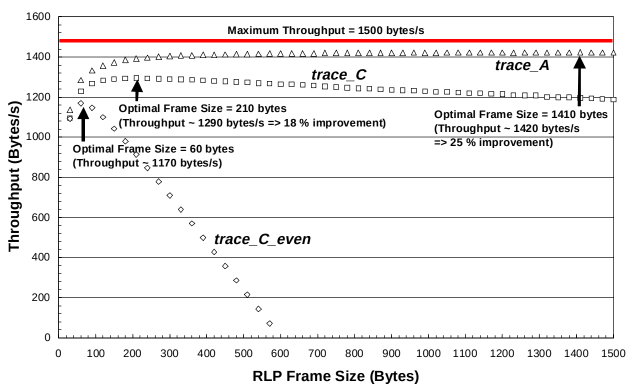

Figure 6: Throughput versus Frame Size.

Our main result is that the frame size chosen for RLP in the current GSM-CSD was an overly conservative design decision. Increasing the (fixed) frame size to 210 bytes would increase the channel throughput by 18-23 percent (!) depending on the radio environment[4]. This still leaves a (theoretical) margin of potential throughput improvement of 8-16 percent for adaptive frame length control, depending on the radio environment.

At first sight, this result might mislead to the conclusion that RLP could be disabled providing a user data rate of 12 kbit/s was offered to PPP (which is not the case in commercially deployed GSM networks as noted in Section 2). If the GSM-CSD channel was the bottleneck link, it appears that pure end-to-end error recovery with a Maximum Transmission Unit (MTU) [28] of 296 bytes (a size commonly used for dial-up lines) would yield almost the same end-to-end throughput. For this MTU size, the widely deployed TCP/IP header compression scheme proposed in [12] would increase application layer throughput by more than 10 percent. The problem, though, is that pure end-to-end error recovery cannot benefit from this advantage. The reason is that the mentioned header compression scheme causes poor TCP performance when employed over unreliable links as, e.g., shown in [20].

Another drawback of pure end-to-end error recovery in this case is that potential advantages of adaptive frame length control schemes could not be realized. The reason is that the path’s MTU cannot be re-negotiated during a connection in current transport protocols. Implementing such a mechanism would also be a poor design choice as optimizing a link’s frame length is not an end-to-end issue. This highlights the weakness of pure end-to-end error recovery as, e.g., studied in [27] for paths in which the bottleneck link is wireless. However, we believe that wide-area wireless links, such as GSM-CSD, will often be the bottleneck in a future Internet.

Another less obvious, but nonetheless plausible result is that the error burstiness on the GSM-CSD channel allows for larger frame sizes than if block erasures were evenly distributed, i.e., not bursty at all. This effect can seen by comparing the graphs *trace_C* and *trace_C_even* (see definition in Section 3.3) in Figure 6. The retrace analysis for *trace_C_even* yields an optimal frame size of 60 bytes (comparing *trace_B* and *trace_B_even* gives the same result). In fact one could view the quotient of the optimal frame sizes for an error trace (bit error trace or block erasure trace) and the corresponding “*_even*” trace as the *burst error factor*. The closer a trace’s burst error factor is to 1 the less the corresponding channel exhibited error burstiness.

4.3 Unfairness of Pure End-to-End Recovery

The preceding section outlined the weakness of pure end-to-end error recovery in situations where the wireless link is the bottleneck of the end-to-end path. In this section, we briefly point out another shortcoming of pure end-to-end error recovery that is independent of whether the wireless link is the path’s bottleneck or not.

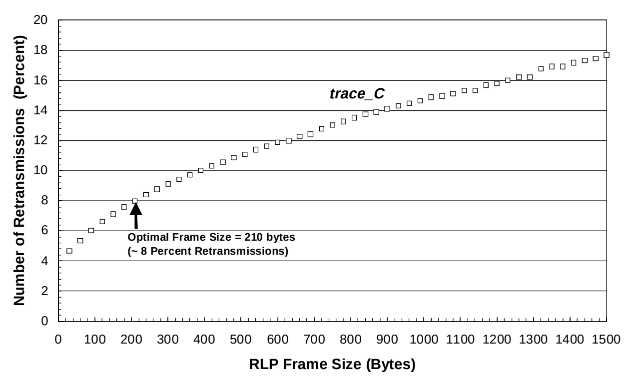

Figure 7: Number of Retransmissions versus Frame Size.

The problem is that the path’s MTU is often not (or cannot be) chosen to “match” the error characteristics of the wireless link, i.e., the MTU is often larger than the wireless link’s throughput optimal frame size. For example, a GSM-CSD link is actually just a seg-

4. For example, for *trace_A* the retrace analysis yields a throughput of 1392 bytes/s for a frame size of 210 bytes and a throughput of 1138 bytes/s for a frame size of 30 bytes/s. For *trace_B* and *trace_C* these frame sizes yield a throughput of 1295 bytes/s and 1096 bytes/s, respectively.

ment of the actual link (IP-hop) provided by serial line protocols such as PPP (see Figure 2). The end points of these serial line protocols determine the MTU. However, since these end points are outside of a GSM network, the MTU cannot be influenced from within GSM. The consequences for pure end-to-end error recovery are that the greater the “mismatches” between the MTU and the wireless link, the more end-to-end retransmissions are required. This result is depicted in Figure 7 for *trace_C* showing the number of retransmissions (as fraction of the overall number of transmissions) that are required for a range of different frame (packet) sizes. The MTU that “matches” the wireless link (210 bytes) would cause 8 percent retransmissions. The more commonly used MTU size of 1500 bytes would instead cause almost 18 percent retransmissions, and the default MTU size of 576 bytes would still require 12 percent retransmissions. Thus, such flows impose an unfair load on a best-effort network, such as the Internet.

Apart from fairness concerns, this inefficiency also decreases end-to-end throughput regardless of where the bottleneck link is located in the path. It also leads to inefficient utilization of the wireless link wasting valuable radio resources (e.g., spectrum and transmission power). We expect these numbers to get worse as weaker FEC schemes are deployed[5]. The reason is that these schemes will most likely further decrease the throughput optimal frame size on those wireless links.

5. Conclusion and Future Work

In this paper, we presented the results of a performance evaluation of link layer error recovery over wireless links. Our analysis is based upon a case study of RLP as an example of a reliable link layer protocol implemented in GSM digital cellular networks. The study leverages of the large set of block erasure traces we collected in different radio environments, with both stationary and mobile end hosts. Our measurement-based approach gave us the unique opportunity to derive models of the wireless link that capture the aggregate of real-world effects like noise, interference, fading, and shadowing. The key premise for our analysis was a model of bulk data transfer based on a reliable end-to-end flow (e.g., a TCP-based flow).

For the case of GSM, we show that the throughput of the circuit-switched data channel can be improved by up to 25 percent by increasing the (fixed) RLP frame size. Larger frame sizes are made possible due to the channel’s error burstiness, a quantity we define as the *burst error factor*. Our results also suggest that techniques such as adaptive frame length control and/or adaptive FEC are worth further exploration as a basis for increasing channel throughput in GSM. This is a topic for our future research, as we plan to implement a measurement-based adaptive frame length control scheme in our testbed.

In general, our case study shows that pure end-to-end error recovery fails to optimize throughput when wireless links form parts of the end-to-end path. The fundamental problem is that the path’s end points are not capable of dynamically adapting their MTU to changing local error characteristics on (possibly multiple) wireless links. Iin many cases, this will lead to decreased end-to-end throughput, an unfair load on a best-effort network, such as the Internet, and a waste of valuable radio resources (e.g., spectrum and transmission power). In fact, we show that link layer error recovery over wireless links is essential for reliable flows to avoid these problems.

In the study presented in this paper, we have not discussed the potential interactions with end-to-end congestion control schemes. However, for TCP our in-progress work shows that this is an unlikely event due to the “conservativeness” of the retransmission timer used in common TCP implementations. Nevertheless, we are working on a new error recovery scheme for TCP. It uses the timestamp option [13] to eliminate the retransmission ambiguity problem [14]. The scheme thereby avoids excessive duplicate retransmissions and unnecessary reductions of the congestion window caused by spurious timeouts and/or packet re-orderings. The results presented in this paper and the initial results from our in-progress work lead us to the conclusion that a fully-reliable mode should be chosen for link layer error recovery when carrying reliable flows. That is, the link layer sender should not lose any packets even over long link outages, up to some conservative termination condition (possibly after minutes, hours, or even days; as in TCP). That way the end-to-end flow of data can be re-started immediately after the link has become available again. Alternative approaches, such as [15], leave this responsibility to the transport layer protocol. However, this will often cause an idle wait for a possibly backed-off retransmission timer to expire.

5. Weaker FEC schemes will be used in the upcoming GSM packet-switched data service [10] and the new GSM-CSD service [9].

Acknowledgments

We would like to thank Prof. Randy Katz for his advice. We also thank Keith Sklower for helping us port the RLP code to UNIX and for lots of fruitful discussions. We thank Bela Rathonyi and the anonymous reviewers for comments on earlier versions of this paper.

References

[1] Bakre A., Badrinath B. R., *I-TCP: Indirect TCP for Mobile Hosts*, In Proceedings of ICDCS 95, May 1995.

[2] Balakrishnan H., Seshan S., Katz R. H., Improving reliable transport and handoff performance in cellular wireless networks, Wireless Networks, Vol. 1, No. 4, pp. 469-481, December 1995.

[3] Brown K., Singh S., *M-TCP: TCP for Mobile Cellular Networks*, Computer Communications Review, 27(5), October 1997.

[4] Cáceres R., Iftode L., *Imroving the Performance of Reliable Transport Protocols in Mobile Computing Environments*, IEEE JSAC, Vol. 13, No. 5, pp. 850-857, June 1995.

[5] Durst R. C., Miller G. J., Travis E. J., *TCP Extensions for Space Communications*, In Proceedings of ACM MOBICOM 96.

[6] Eckhardt D. A., Steenkiste P., *Improving Wireless LAN Performance via Adaptive Local Error Control*, In Proceedings of IEEE ICNP 98.

[7] ETSI, *Radio Link Protocol for data and telematic services on the Mobile Station - Base Station System (MS-BSS) interface and the Base Station System - Mobile Switching Center (BSS-MSC) interface*, GSM Specification 04.22, Version 5.0.0, December 1995.

[8] ETSI, *Digital cellular communications system (Phase 2+); Radio Link Protocol for data and telematic services on the Mobile Station - Base Station System (MS-BSS) interface and the Base Station System - Mobile Switching Center (BSS-*

MSC) interface, GSM Specification 04.22, Version 6.1.0, November 1998.

[9] ETSI, *Digital cellular communications system (Phase 2+); Rate adaption on the Mobile Station - Base Station System (MS - BSS) Interface*, GSM Specification 04.21, Version 7.0.0, October 1998.

[10] ETSI, *Digital cellular communications system (Phase 2+); General Packet Radio Service (GPRS); Mobile Station (MS) Base Station System (BSS) interface; Radio Link Control / Medium Access Control (RLC/MAC) protocol*, GSM Specification 04.60, Version 6.1.0, August 1998.

[11] Henderson T. R., Katz R. H., *Transport Protocols for Internet-Compatible Satellite Networks*, To appear in IEEE JSAC in 1999.

[12] Jacobson V., *Compressing TCP/IP Headers for Low-Speed Serial Links*, RFC 1144, February 1990.

[13] Jacobson V., Braden R., Borman D., *TCP Extensions for High Performance*, RFC 1323, May 1992.

[14] Karn P., Partridge C., *Improving Round-Trip Time Estimates in Reliable Transport Protocols*, In Proceedings of ACM SIGCOMM 87.

[15] Karn P., *The Qualcomm CDMA Digital Cellular System*, In Proceedings of the USENIX Mobile and Location-Independent Computing Symposium, USENIX Association, August 1993.

[16] Kent S., Atkinson R., *Security Architecture for the Internet Protocol*, RFC 2401, November 1998.

[17] Kojo M., Raatikainen K., Liljeberg M., Kiiskinen J., Alanko T., *An Efficient Transport Service for Slow Wireless Telephone Links*, IEEE JSAC, Vol. 15, No. 7, pp. 1337-1348, September1997.

[18] Lettieri P., Srivastava M. B., *Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency*, In Proceedings of IEEE INFOCOM 98.

[19] Ludwig R., Rathonyi B., *Link Layer Enhancements for TCP/IP over GSM*, In Proceedings of IEEE INFOCOM 99.

[20] Ludwig R., Rathonyi B., Konrad A., Oden K., Joseph A., *Multi-Layer Tracing of TCP over a Reliable Wireless Link*, In Proceedings of ACM SIGMETRICS 99.

[21] Manzoni P., Ghosal D., Serazzi G., *Impact of Mobility on TCP/IP: An Integrated Performance Study*, IEEE JSAC, Vol. 13, No. 5, pp. 858-867, June 1995.

[22] Mouly M., Pautet M.-B., *The GSM System for Mobile Communications*, Cell & Sys, France 1992.

[23] Noble B. D., Satyanarayanan M., Nguyen G. T., Katz R. H., *Trace-Based Mobile Network Emulation*, In Proceedings of ACM SIGCOMM 97.

[24] Pons J., Dunlop J., *Bit error Rate Based Link Adaptation for GSM*, In Proceedings of IEEE PIMRC 98.

[25] Postel, J., Internet Protocol, RFC 791, September 1981.

[26] Postel, J., Transmission Control Protocol, RFC793, September 1981.

[27] Samaraweera N. K. G., Fairhurst G., *Reinforcement of TCP Error Recovery for Wireless Communication*, Computer Communications Review, 28(2), April 1998.

[28] Stevens W.R., *TCP/IP Illustrated, Volume 1 (The Protocols)*, Addison Wesley, November 1994.

[29] The ICEBERG Project, CS Division, EECS Department, University of California at Berkeley, http://iceberg.cs.berkeley.edu/.