Practical Experience with HTTP and TCP over GPRS

Rajiv Chakravorty and Ian Pratt
Email: {rajiv.chakravorty, ian.pratt}@cl.cam.ac.uk
University of Cambridge Computer Laboratory, Cambridge CB3 0FD, U.K.

1. INTRODUCTION
Performance of protocols like HTTP and TCP are known to degrade over wireless links. This abstract discusses experimental results of TCP and HTTP performance over General Packet Radio Service (GPRS), an extension to the GSM standard which is being deployed rapidly throughout much of the world. The research is aimed at:

1. understanding GPRS link characteristics
2. identifying protocol performance problems experienced by TCP and HTTP over GPRS,
3. devising solutions that can overcome or even optimise such bottlenecks and,
4. experimentally demonstrating the overall efficacy of our solution.

2. TEST ENVIRONMENT AND TOOLS
Our experimental setup for characterizing GPRS links is shown in figure 1. The measurements reported were all performed over Vodafone UK’s national GPRS network, though similar results were measured using a number of other European GPRS networks. A number of different handsets from different vendors were tried; we found no significant performance variation between handsets of the same GPRS device class. Measurements were repeated at many different locations resulting in a wide range of radio and network conditions to try and gain a view of ‘typical’ performance experienced by a user.

In an example test set-up (figure 1), a laptop connects to a Motorola T200 GPRS (3–1–1) (3 downlink, 1 uplink channels) phone through a serial PPP (point-to-point) link to act as a GPRS mobile terminal. Complying to the usual GPRS architecture, the base stations (BSs) are linked to the SGSN which is then connected to a GGSN. In the current Vodafone configuration, both SGSN and GGSN node is co-located in a CGSN (Combined GPRS Support Node).

Since we were unable to install equipment next to the CGSN we made use of a well provisioned IPSec (IP Security) VPN tunnel to route all traffic via the Computer Laboratory. The measurement terminal (see figure 1) was then located at the end of the tunnel, with routing configured so that all packets flowing to and from the mobile host are passed to it for processing. A RADIUS server was used to authenticate mobile terminals and IP addresses.

All the characterization tests were performed using a version of the ttcp program modified (ttcp [4]) to enable traffic streams to be generated at specified rates and with particular burst characteristics. We also inserted time stamps and sequence numbers in packets to track time-in-flight between sender and receiver (using NTP synchronized clocks) and to detect packet loss and reordering.

3. EXPERIMENTAL EVALUATION
We conducted a series of link characterization measurements over GPRS [5]. Results from the measurements reveal that GPRS links have very high RTTs (>1000ms) and fluctuating available bandwidth. Packet losses are relatively rare when the mobile host is stationary. However, link blackouts can still occur, resulting in stalls of typically 5–30 seconds. Longer outages are more likely to cause packet loss, ACK compression is fairly common over both uplink and downlink, caused by the link-layer protocol.

Due to such link inconsistencies, experimental evaluation of TCP over GPRS uncovered a number of performance problems (see [1] for more information):

- a sluggish slow-start that can take many seconds (due to high RTTs) for the window to ramp-up and achieve full downlink utilization,
- excess queue build up over the downlink with long-lived TCP sessions - this can cause gross unfairness to other TCP flows and a high probability of timeouts during initial connection request,
- spurious TCP timeouts due to occasional link outages and,
- a high drain time to recover after TCP timeouts.

We also investigated ways to improve web (HTTP) performance over GPRS [2]. Our approach towards this problem was not only from the perspective of improving TCP, but also from the point of optimizing web browser performance. Web browser
4. MEASUREMENT RESULTS

We have shown that using an interposed mobile proxy located close to the wired-wireless boundary can give substantial benefits to both TCP and Web [HTTP] performance over GPRS. We offer performance enhancements to the mobile proxy at two different layers: transport layer and also at the application layer.

The transport layer enhancement (TL-E) offers a transparent split TCP approach and uses a fixed size congestion window [cwnd]. This approach eliminates the slow start phase and clamps TCP’s cwnd to the value of GPRS downlink bandwidth-delay product. The approach has shown to clearly improve TCP performance over GPRS without requiring modification to either client or the server. The scheme has shown to offer (see [1]) the following benefits:

1. **Minimization of Queuing Delays** – excessive queuing over the downlink is minimized by limiting data of long TCP flows over the link. As a consequence, RTT inflation and its impact on retransmit timer values are also minimized,

2. **Faster Startup** – slow-start is avoided, which improves start-up performance and transfer times (especially for short web transfers),

3. **Quick Recovery from Losses** – reduces drain time after losses leading to quicker TCP recovery. By limiting data over the link, spurious retransmission cycles due to sudden delay fluctuations can be avoided. This also reconciles with other negative effects such as stale (or leftover) TCP data due to abnormal disconnections.

Our application level enhancement (AL-E) offers data caching at the mobile proxy and can service pipelined requests from pipelined capable web browsers. For test purposes, we created a mock-up of a proxy server, CNN.micromedia.com, on all the base provisioned web server. This was done to avoid the effects of fast changing web-content and to eliminate measurement ‘noise’ that could be introduced by changes in Internet performance. We refer to our dynamic web-site as local CNN, or in short LCNN. To compare the download performance of our scheme, we used the mozilla web browser. The latest version from mozilla supports pipelining. However, after analysing browser traces we found few instances where the browser actually pipelined requests on persistent connections, even when doing so was clearly possible (for detailed discussion on this, see [2]).

5. FUTURE RESEARCH

There are many interesting artifacts resulting from this piece of research. Specifically, our research entails further questions:

1. How can web-clients be made to dynamically adapt to the underlying network heterogeneity?
2. How do we minimize the possibility of Head of Line (HOL) blocking effects with pipelining?
3. How to maximize pipelining efficiency in presence of resource inter-dependencies with dynamic web-content?

In the future, we also plan to investigate a number of optimization techniques such as application data compression, delta encoding and deterministic prefetching to improve protocol performance over low bandwidth GPRS. Please visit our web-page, http://www.cl.cam.ac.uk/users/rc277/gprs.html for more information on our research and related published papers.

Acknowledgements

We thank Vodafone Group R&D, Sun Microsystems Inc. and Benchmark Capital for supporting this work.

6. REFERENCES


