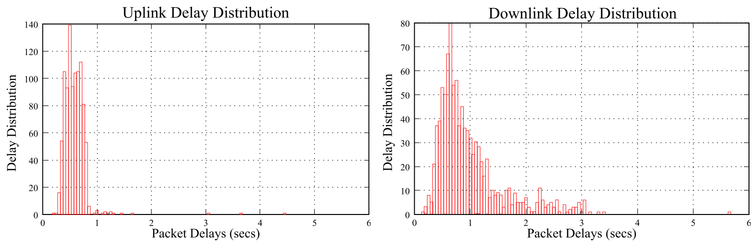
Practical Experience with HTTP and TCP over GPRS

Rajiv Chakravorty and Ian Pratt, Cambridge University, U.K.

GPRS Link Characterization

+ High RTTs: 1000ms or more
+ Links Outages: Typically, observed for duration of 4-40s. Link "stalls"can occur when mobile host is stationary.

Single Packet flight time distributions for 1000 packets; random delay higher than 4s between packets



+ Bandwidth: Variable and Fluctuating, Max: **4.15 KB/s** downlink and **1.5KB/s** uplink using a 3+1 channel GPRS phone
+ ACK Compression: Link Layer (RLC) effect, both on uplink as well as downlink
+ Losses: Relatively rare during stationary conditions

TCP Problems over GPRS

+ Sluggish slow-start phase in TCP due to very high RTTs (so, short TCP flows have higher transfer times)
+ Excess queuing at wireless gateway during long flows, leading to RTTs of over 30s. This leads to gross unfairness to other existing flows and a high probability of timeouts on initial connection request for new flows
+ Slow recovery after timeouts, due to excess queuing.
+ Spurious TCP timeouts due to occasional link "stalls"

Browser Behavior

+ Most Web browsers aggressive, open many concurrent TCP connections
+ Good for Wired-Internet (reduces download times), but has a high **cost** over GPRS
+ Cost: Signaling and Connection setup overhead
+ Also, may lead to saturation of downlink buffers
+ HTTP/1.0 very inefficient, HTTP/1.1 is better, however, what about ***pipelined*** connections?
+ Pipelining HTTP requests over GPRS can yield higher downlink utilization

Improving Performance: How?

+ Use an **interposed Mobile Proxy**, located close to wireline-wireless border near GPRS CGSN Node
+ Proxy **performance enhanced** at Transport Layer (TCP) and the Application Layer, for benefit over the GPRS downlink
+ Aggressive Web Browser **Pipelining** over GPRS

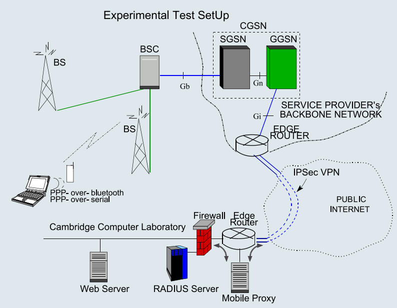
Transport Layer Enhancement (TL-E)

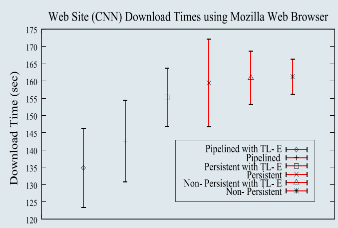
+ Splits TCP connection transparently into two halves
+ On GPRS side, it avoids TCP's slow start phase and instead, uses a fixed value of the congestion window (*cwnd*) and clamps it for the full connection duration
+ TCP clamped *cwnd* value = optimistic value of the bandwidth delay product of the GPRS downlink
+ Share this window amongst all flows to the same mobile host
+ TCP *cwnd* clamping leads to: (i) Minimization of excess queuing, (ii) Faster startup for short flows and, (iii) Quicker recovery from losses

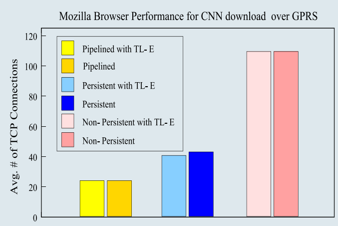
Application Layer Enhancement (AL-E)

+ Allows pipelined requests from pipeline capable browsers, even if there is no such support from servers

Experimental Setup and Results







Experimental Web download tests over GPRS show that:

- *Use of a performance enhanced mobile proxy combined with a moderate support from a pipeline Web browser,* ***reduces mean web page download times by about 15-20%.***
- *A browser making few concurrent connections but aggressively using pipelining on them can substantially improve response times over GPRS.*

Open Issues and Future Work

1. How can web browsers be made to dynamically adapt to the underlying network heterogeneity?
2. How to 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?

Read On! http://www.cl.cam.ac.uk/users/rc277/gprs.html