Receiver-based Management of Low-bandwidth Access Links

INFOCOM 2000

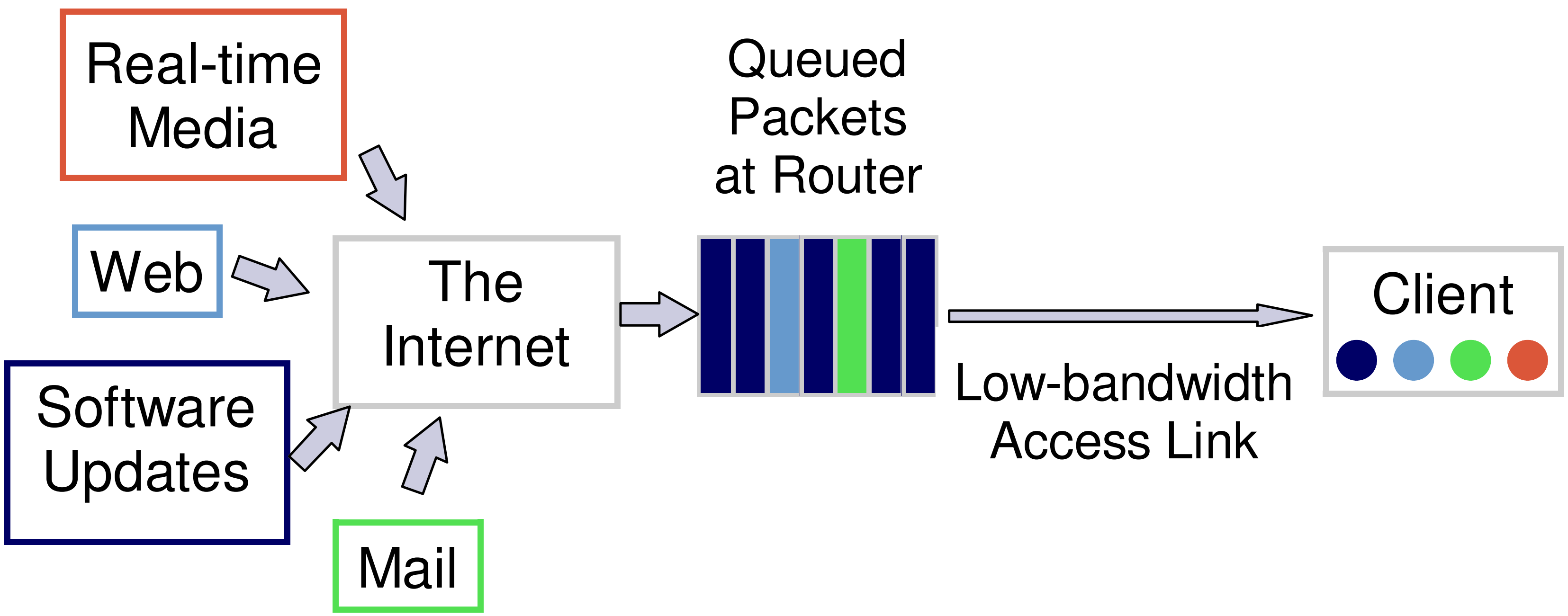
March 28, 2000

Neil Spring, Maureen Chesire, Mark Berryman, Vivek Sahasranaman, Thomas Anderson, and Brian Bershad

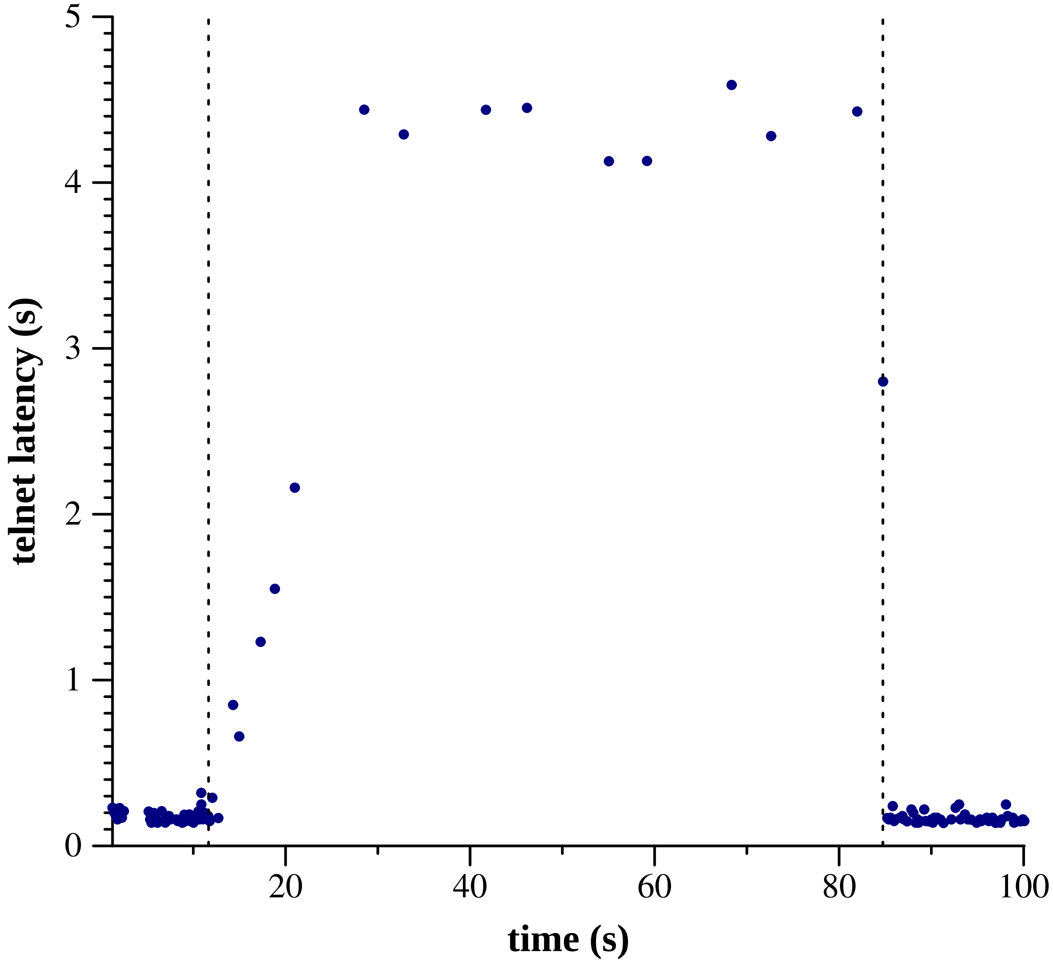
The Problem

- While downloading
  - Large software or email attachment
- Try
  - Checking your mail or surfing the Web
- Interactive performance is *abysmal* during background transfers

The Conceptual Model



Telnet vs. FTP



Why does this happen?

- Queue length increases effective latency of new or interactive connections
- Existing connections dominate throughput consumption, new connections face an entrenched enemy

Our Approach

- Control the queue at the bottleneck link
  - Queue composition implies throughput share
  - Long queue length implies delay
- From the <u>receiver</u>
  - Unique knowledge determines priorities
  - Solution is deployed here

Our Design

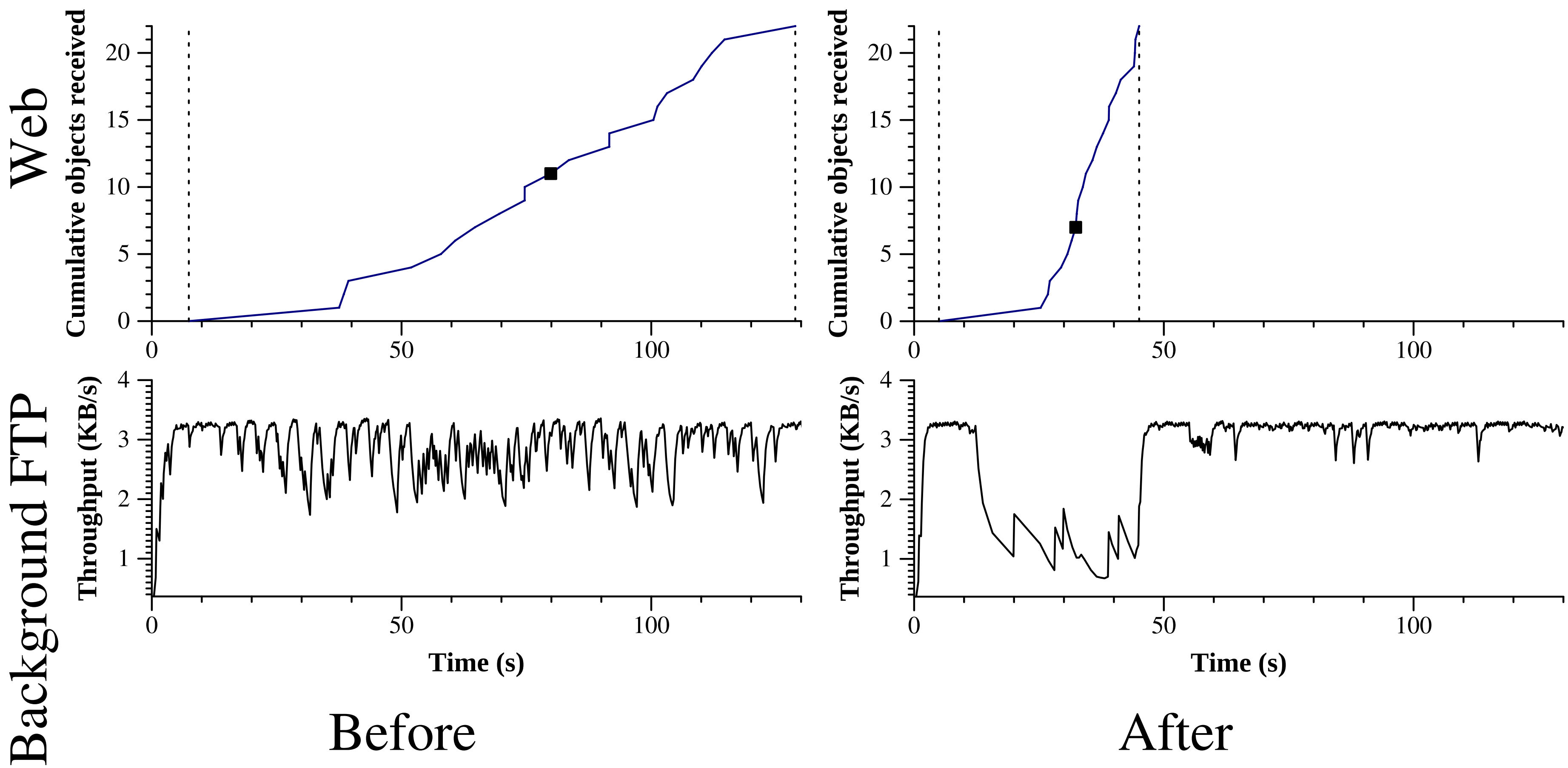
- Categorize flows automatically
  - Idle: long time since last activity
  - Long transfer: many bytes received since last send
  - Short transfer: few bytes received since last send
  - Interactive: single, short packets exchanged
- Determine link parameters
- Relate throughput share to window size
  - Estimate RTT: receiver→sender →receiver
  - Wnd_c = Queued $packets_c$ + $Throughput_c$ * RTT_c
- Constraints:

$$Wnd_c \geq 1 packet \qquad \sum_c Throughput_c \approx LinkThroughput$$

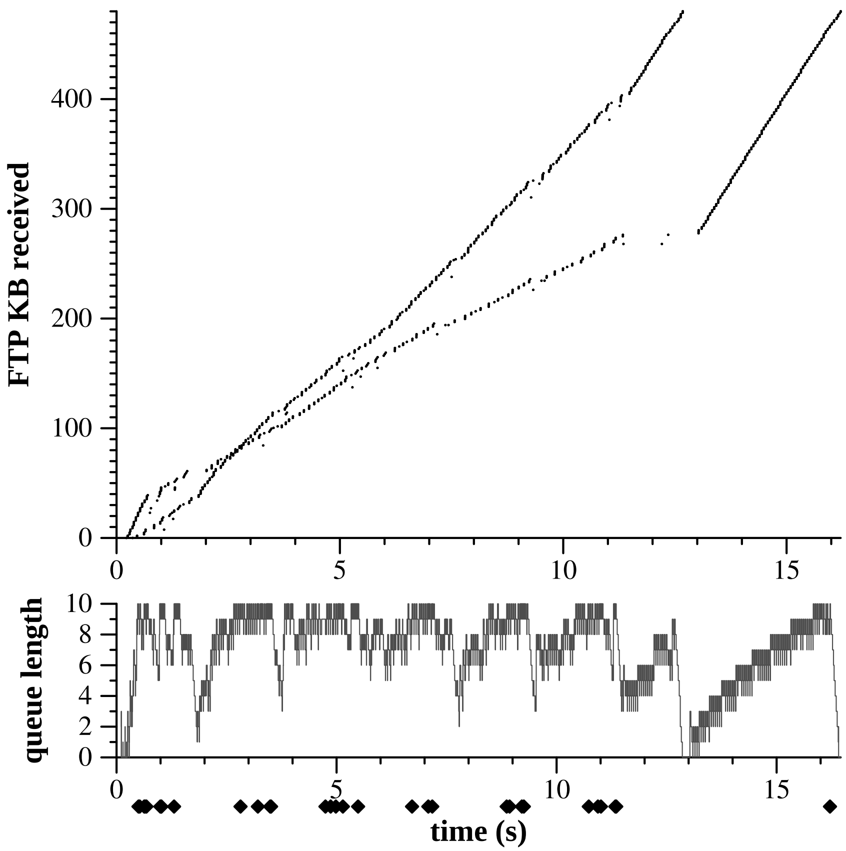
Our Mechanism

- TCP's advertised window
  - Ordinarily used for flow control
  - How much traffic the receiver will accept from the sender
- Linux 2.2 kernel module
  - No application, network, or server changes

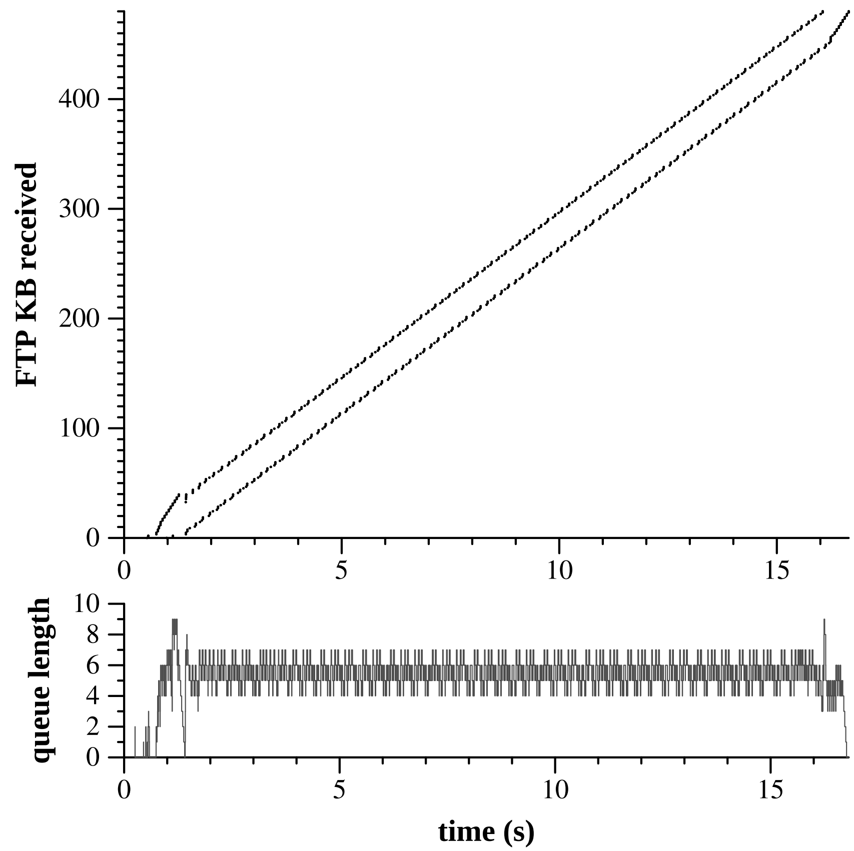
Faster Web Downloads



Fair division between FTP flows



Before

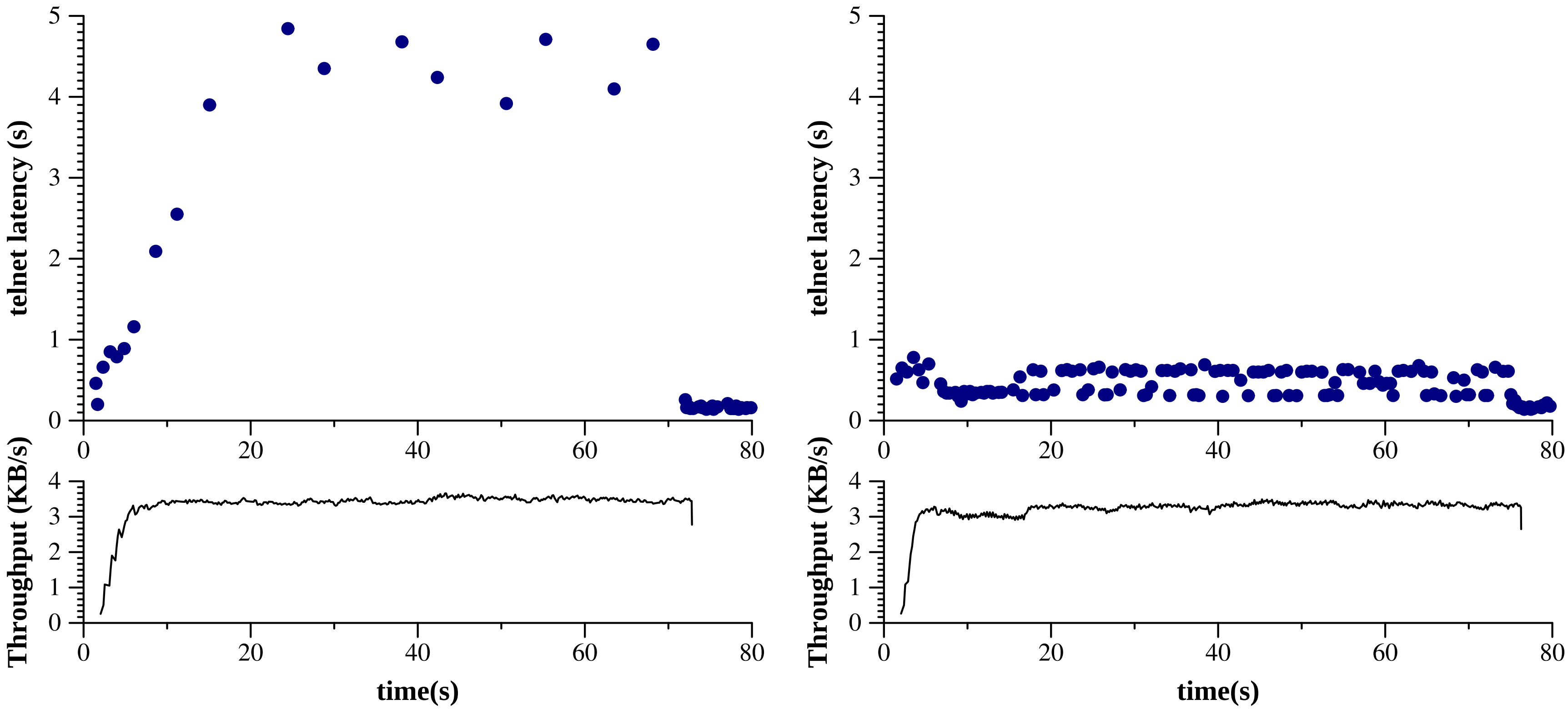


After

Better telnet performance

Telnet

Background FTP



Before

After

Conclusions

- Significant performance advantage
- Future challenges:
  - Shared links (e.g. cable modems)
  - Non-TCP flows (e.g. Real Audio)