Aalto University
School of Electrical Engineering

In the Interest of the Short:

An approach for e2e congestion control

Gautam Raj Moktan

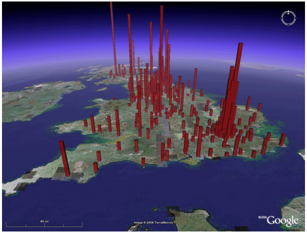
Department of Communications and Networking
Aalto University, School of Electrical Engineering
gautam.moktan@aalto.fi

January 11, 2012

Outline

- Motivations
- Idea ! ? “
- Algorithm Design
- Testbed
- Results
- Furthermore ...

Flow Lengths in the Internet



There are a lot of Short ones !!

- Short flows contribute to more than 95th percentile of Internet traffic [1]
  - Average web-page size on the Internet is 320 KB.
  - The 90th percentile for web-page size is 663KB.
  - The 90th percentile for KB per host is 179 KB.
- TCP flows in one Finnish mobile operator's 3G network :
  - 98th percentile less than 60.1KB
  - 99th percentile was 123.8KB

[1] S. Ramachandran, *"Let's make the web faster"*, May 2010, Website: http://code.google.com/speed/articles/web-metrics.html.

Case for Saving Energy

- Energy cost per bit for wireless transmission $\geq 1000\times$ Energy cost for computing that bit [5]
- Energy consumption while radio is "Idle" $\simeq$ Energy consumtion while receiving [6]
- The wireless modem consumes more power than the display and applications processor combined [6]
- Minimizing the uptime of radio link to reduce energy consumtion
  - Compression strategies at application layer [5]
  - TDMA scheduling at link layer [7]
  - We attempt it at the **Transport Layer !!**

---

[5] Barr and Asanovic, *Energy aware lossless data compression*, ACM Transactions on Computer Systems (TOCS), 2006

[6] S. Goel and T. Imielinski, *"Etiquette protocol for ultra low power operation in sensor networks"*, Technical Report DCS-TR-552, Department of Computer Science, Rutgers University, April 2004.

[6]

[7] Shi, L. and Fapojuwo, A.O., TDMA scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks, IEEE Transactions on Mobile Computing, 2010

Other Motivations ...

- User Experience
  - User Ergonomics: Show that delay and disruptions is more perceptible in shorter length flows[2]
  - User's Frustation Levels: Increase with increase in download delay
  - Utility of Webcontents: Related to the time taken to download at end-device
- Change in Paradigm:
  - Volume Pricing : Spoils User Experience
  - Flat Pricing: Fairness it claims isn't really fair [4] since Customers using less than subscribed BW are paying for others usage
  - Congestion Pricing: Driven by fairness as in traditional goods markets, ECN and Re-ECN apply this new pholisophy

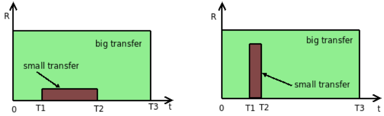
---

2 J.A.Jacko, A. Sears, and M.S. Borella, *"The effect of network delay and media on user perceptions of web resources"*, Behaviour and Information Technology 19 (2000), no. 6, 427-440.

4 B. Briscoe, *"Flow rate fairness: Dismantling a religion"*, ACM SIGCOMM Computer Communication Review,37(2):63-74, april 2007.

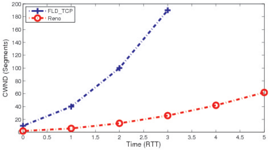
Idea ! ? “

- TCP enhancement with a Flow length (bytes) Dependent Congestion Control Algorithm (FLD_TCP)
- Philosophy:
  - Shorter Flows get priority over longer flows
  - Obtain Kaldor-Hicks improvement
- Supports Go Fast Finish Early types of traffic patterns



- Being aggressive in the beginning to throttle transport and help get the job done early
- Be nice if the flow is longer, hence be fair too !

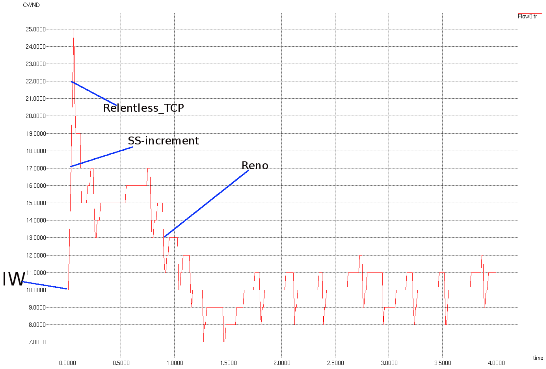
How to Go Fast and Finish Early ?



- Three parameters:
  1. Increase Initial Congestion Window (IW):
     - Traditional TCP starts with 1 or 2 segments
     - IETF proposed in 1998/2002 to increase the inital window to 4kB (3 or 4 segments).
     - Google proposed in 2010 to increase it to 15kB (10 segments).
  2. Increase Slow Start Increment Rate
     - Original method: for each ACK obtained, double 2^x the window
     - Instead: Triple or x-tuple the increase for each ACK. eg. CWND increase becomes 3^x , 5^x,...
  3. Choice of Congestion Avoidance Algorithm depending on flow length:
     - Aggressive CA algorithms: Relentless TCP, FAST TCP,...
     - Fair algorithms : Cubic, Compound,Reno,...
     - Laid back algorithms : TCP Nice, TCP-LP(low priority), LEDBAT,..
- Tailoring of these properties to redesign a flow-length dependent CC algorithm that supports Go Fast Finish Early packet transactions

Algorithm Design

- Initial Window of 10 segments
- SS-increment factor 3^x
- Relentless TCP (with tweaks) as the Aggressive Algorithm choice
- Reno after threshold is crossed



Algorithm Design (cont.)

- Threshold Selection
  - Webpage size, Average: 320KB, 90th percentile [1]: 663.19 KB
  - Average GET size on the web [1] : 7.3KB (Http Pipelining causes size of a TCP flow to increase)
  - KB per host, 90th percentile [1] :179.08 KB
  - Mobile 3G network 99th percentile: 123.8 KB [2]
  - Wireshark Analysis of few web sessions gave similar projections
- Our threshold value for experiments = 2MB [3]
- Objective: Obtain a throttled transport for websurfers by shifting the pain to flows longer than 2MB
- Projections:Takes care of more than 95th percentile of the tcp flows

---

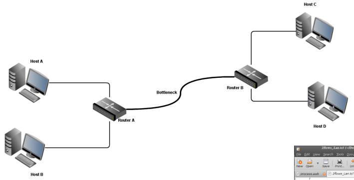
1 Google: http://code.google.com/speed/articles/web-metrics.html

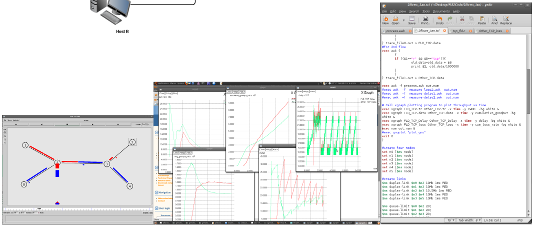
2 Analysis of data from a finnish telecom operator

3 Can be changed easily depending on the usage scenario.

Testbed

- A Linux TCP implementation for NS2
  - helps develop new TCP algorithms that can be ported into pluggable CC implementation of Linux TCP





-
  - Boundary Conditions: Links with different BW (2MB - 200MB)and delays (5ms - 100ms)
  - Packet size 1500 Bytes, Initial Window 10 Segments: 1st RTT sends 14KB approx...
  - queue length = 20 pkts and queue type = RED

Results

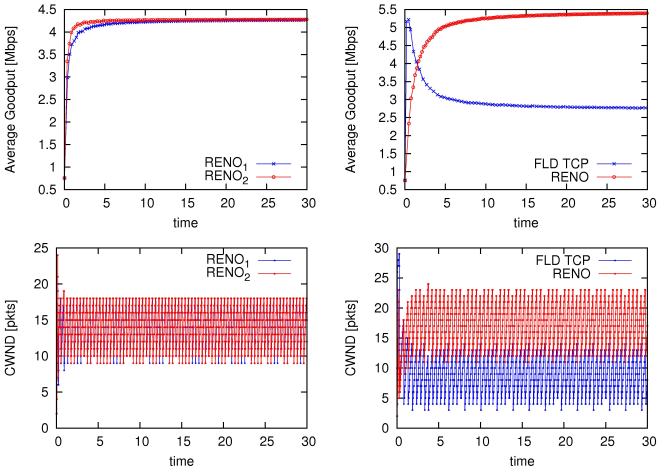


Figure: Simulation results in Wan Case:Bottleneck BW=10Mbps, Delay=15ms

Results (contd..)

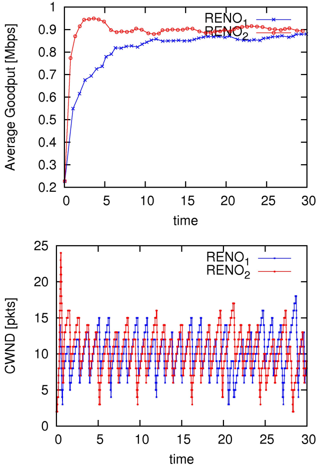


Figure: Simulation results in Mobile Case:Bottleneck BW= 2 Mbps, delay=50ms

Results (contd..)

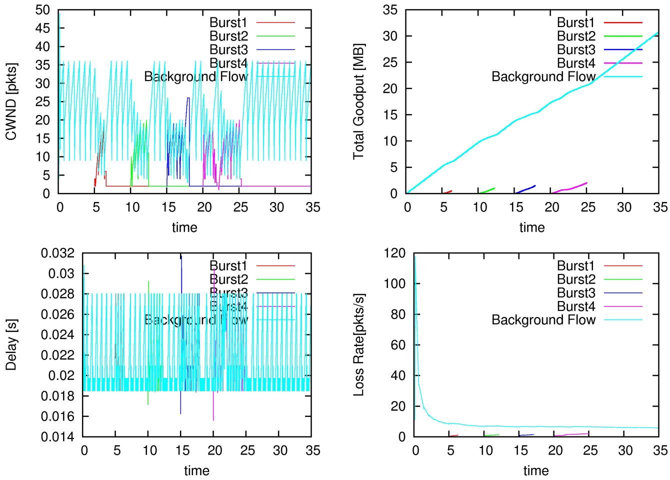


Figure: Reno bursts in single background flow

Results (contd..)

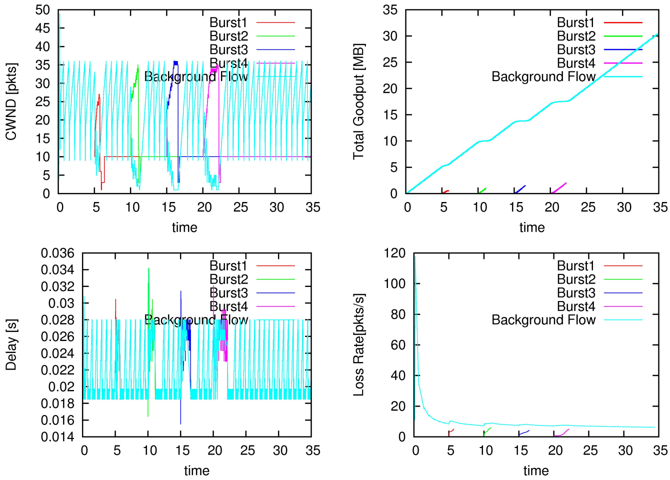


Figure: FLD_TCP bursts in single background flow

Results (contd..)

Burst Transfer Time :

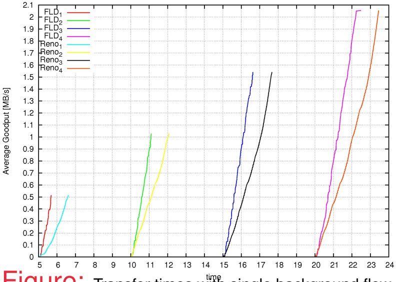


Figure: Transfer times with single background flow

	FLD_TCP Transfer Time	RENO Transfer Time	RENO/FLD_TCP
Burst1(512KB)	0.75 sec	1.4 sec	2
Burst2(1MB)	1.2 sec	1.83 sec	1.6
Burst3(1MB)	1.7 sec	3 sec	1.76
Burst4(1MB)	2.6 sec	5.2 sec	2

Table: Transfer times of FLD_TCP and RENO bursts

FLD_TCP finishes almost twice as fast than Reno.

Results (contd..)

Background Flow Transfer Time :

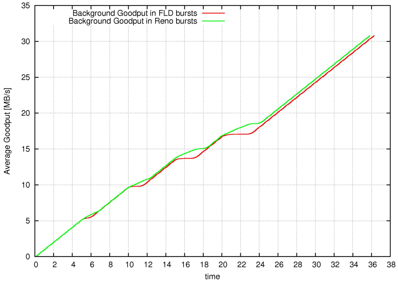


Figure: Transfer time of background flow

With FLD_TCP, background flow is delayed 0.011 %.

Results (contd..)

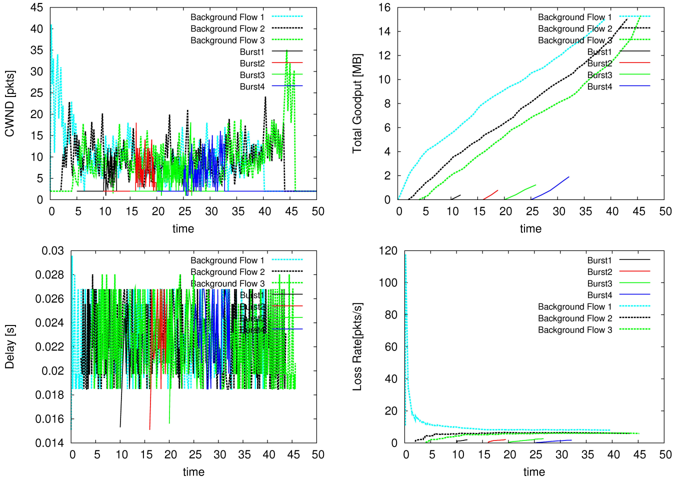


Figure: Reno bursts in 3 background flows

Results (contd..)

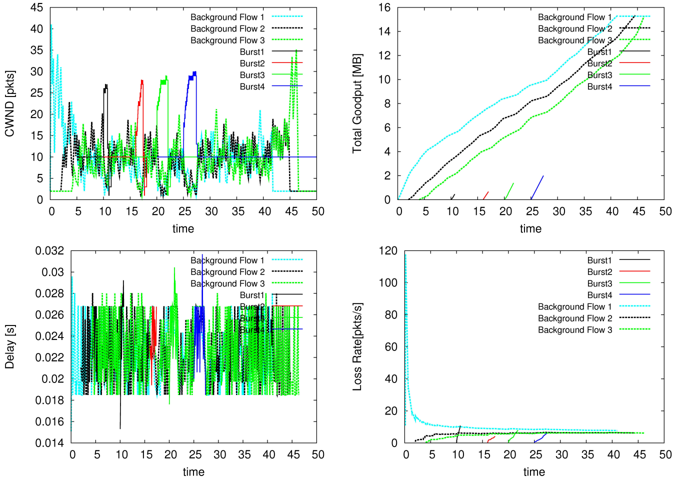


Figure: FLD_TCP bursts in 3 background flows

Results (contd..)

Burst Transfer Time :

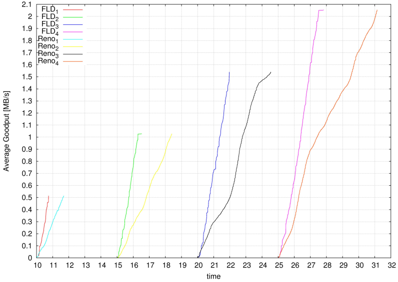


Figure: Transfer times with 3 background flows

	FLD_TCP Transfer Time	RENO Transfer Time	RENO/FLD_TCP
Burst1(512KB)	0.6 sec	1.7 sec	2.833
Burst2(1MB)	1.5 sec	2.4 sec	1.6
Burst3(1MB)	2 sec	4.6 sec	2.3
Burst4(1MB)	2.8 sec	6.2 sec	2.2

Table: Transfer Times of FLD_TCP and RENO bursts

FLD_TCP finishes almost twice as fast than Reno.

Results (contd..)

Background Flow Transfer Time :

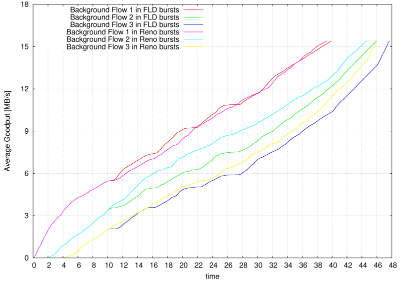


Figure: Transfer time of background flow

	With FLD_TCP bursts	With Reno bursts	FLD_TCP/RENO
Background Flow 1	39.9 sec	39.3 sec	1.0155
Background Flow 2	43.9 sec	41.6 sec	1.0555
Background Flow 3	43.7 sec	42 sec	1.0405

Table: Transfer Times of Background Flows

With FLD_TCP, background flow is delayed upto 0.05 %.

Hence,

- Flows finish faster with FLD_TCP
- Becomes flow rate friendly after crossing threshold
- Go Fast Finish Early
- Good for saving energy & Enhancing user experience
- and furthermore...
  - Analysis of the behavior in other conditions still to be made
    - Flows with distribution of web traffic
    - Tests under other evaluation criteria
    - Scalability / Deployability/ Stability analysis
  - Mathematical Modelling
  - A real Linux implementation
  - Quantify the savings in terms of energy (Power/Battery life)
  - Equation based approach !
    - removing the arbitrary threshold.

- Thank You!!