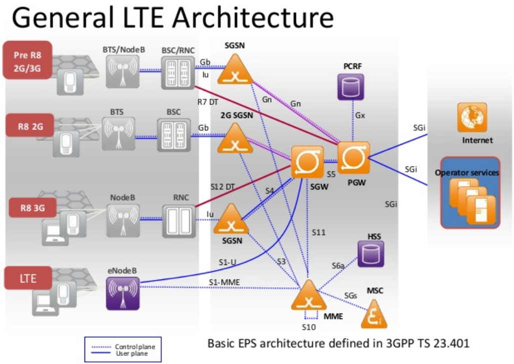
Sprout: Stochastic Forecasts Improve Performance in Cellular Networks

Keith Winstein, Anirudh Sivaraman, Hari Balakrishnan

6.829 Fall 2018

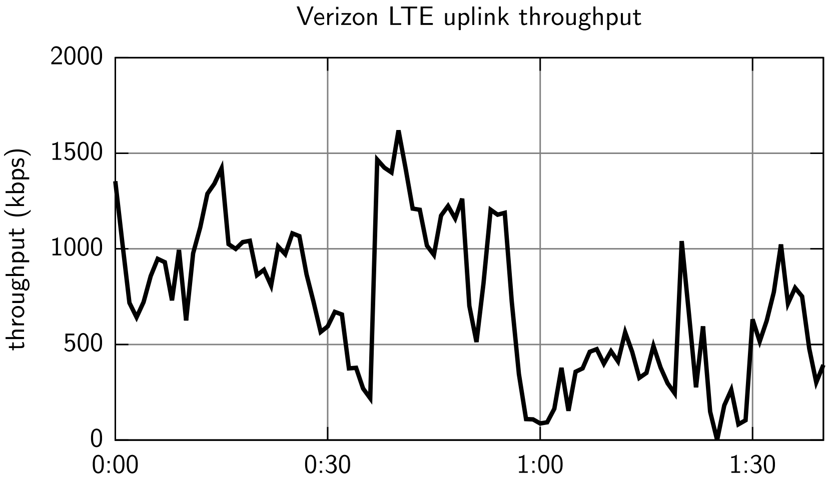
October 25, 2018

Cellular Network Architecture

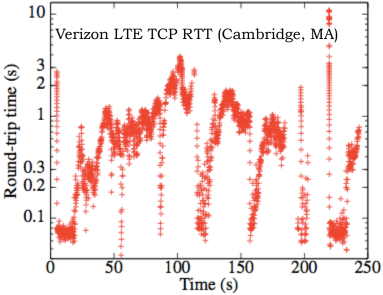


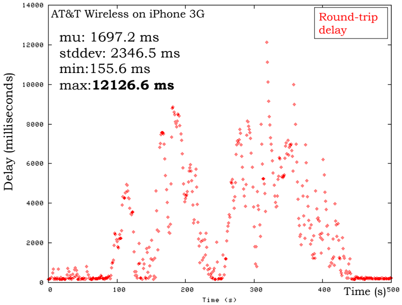
http://image.slidesharecdn.com/introductiontomobilecorenetworkpublic-130325020435-phpapp01/95/introduction-to-mobile-core-network-13-638.jpg

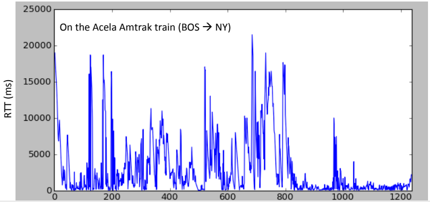
Wireless: Highly Variable Rates



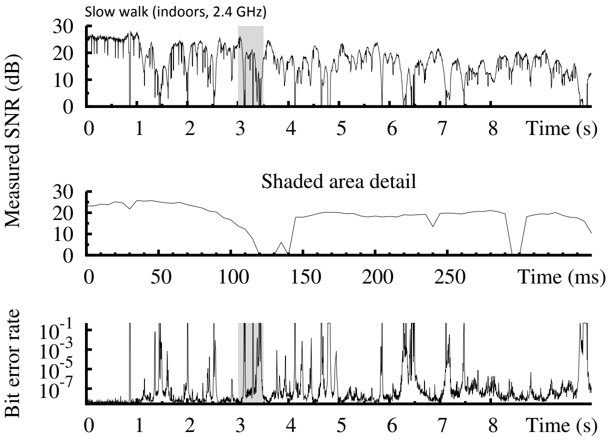
Awful Delays: The *Too-Reliable* Network







Highly Variable Signal-to-Noise Ratio (SNR)



M. Vutukuru, K. Jamieson, HB, "Cross-Layer Wireless Bit Rate Adaptation", SIGCOMM 2009.

Packet Scheduling

- What is the "fair" way to schedule wireless users?
- Case 1: User i has a rate r_i packets/s.
- Case 2: $r_i(t)$ is a function of time.

Packet Scheduling

- Cellular networks maintain per-device queues because it allows the base station to trade-off between efficiency and fairness.
- Scheduling depends on the state of the channel to a user.

Proportional Fair Wireless Scheduler

- Let $r_i(t)$ be the current ("instantaneous") rate and let $R_i(t)$ be the value of time t of an EWMA-filtered average:

$$R_i(t+1) = (1-\alpha)R_i(t) + \alpha r_i(t) \text{ if } i = j,$$

and

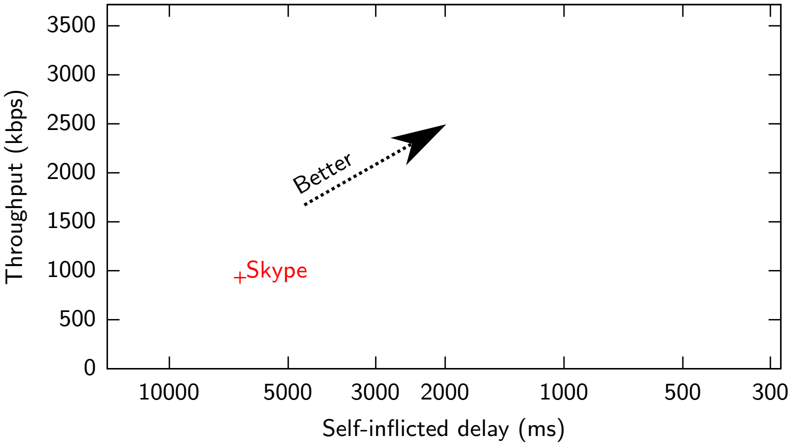
$$R_i(t+1) = (1-\alpha)R_i(t) \text{ otherwise}$$

- Select j that maximizes $\frac{r_i(t)}{R_i(t)}$.

Video & Conferencing over Wireless

- We measured cellular networks while driving:
  - **Verizon LTE**
  - Verizon 3G (1xEV-DO)
  - AT&T LTE
  - T-Mobile 3G (UMTS)
- Then ran apps across emulated network:
  - **Skype** (Windows 7)
  - Google Hangout (Chrome on Windows 7)
  - Apple Facetime (OS X)

Characterizing Performance



Why is Wireless Videoconferencing So Bad?

- Today's protocols **react** to congestion signals
  - Packet loss
  - Increase in round-trip time
- Feedback comes too late to help
- The killer: **self-inflicted queueing delay**
- Any overshoot means a queue filling up with packets

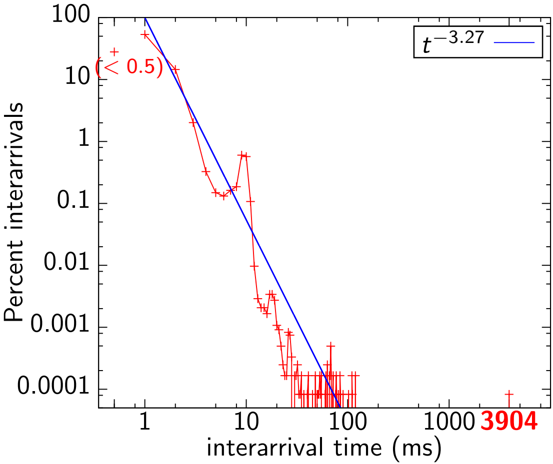
Sprout's Goal

- Maximize throughput, but
- **Bounded risk of delay** $> D$ **(e.g.,** $D = 100$ **ms)**.

Bounded Risk of Delay

- **Infer** rate from *interarrival distribution*
- **Predict** future link rate and convey prediction to sender
  - Don't wait for congestion
- **Control:** Send as fast as possible, but require:
  - 95% probability all packets will arrive within 100 ms

Infer Rate from Interarrival Process

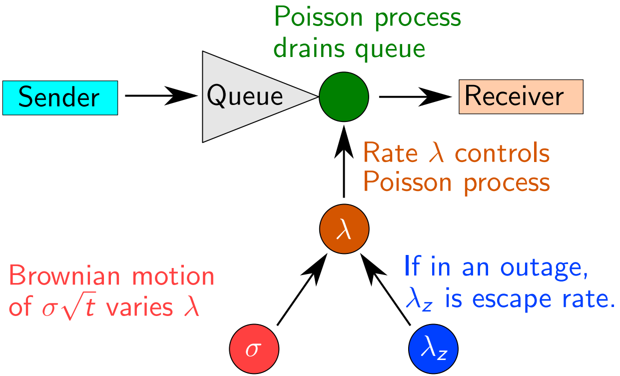


Verizon LTE. Stationary phone. 3 am... "Flicker noise" process.

Predict Future Link Rate

- Model rate evolution as **random walk** (Brownian motion)
- Count packets in every 20 ms *tick*
- Use Bayesian updating to make cautious forecast (5th percentile cumulative packets)
- Receiver makes forecast; tells sender in ACK

Network Model



Bayesian Update

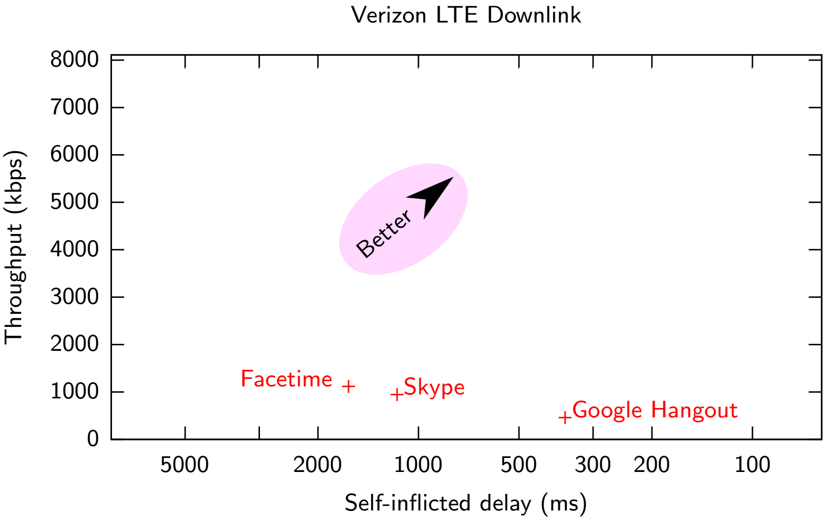
- Discrete set of possible rates, λ (e.g., 0 to 1000 packets/s)
- Initially, each λ is equi-probable
- Each tick (τ seconds), if we receive k bytes, run update step:

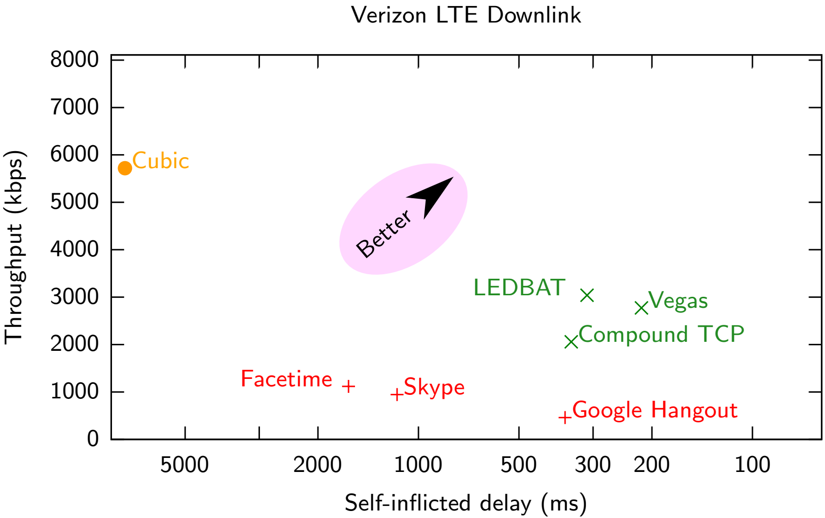
$$\mathbb{P}_{\text{new}}(\lambda = x) \leftarrow \frac{\mathbb{P}_{\text{old}}(\lambda = x) \cdot \frac{(x\tau)^k}{k!} e^{-(x\tau)}}{Z},$$

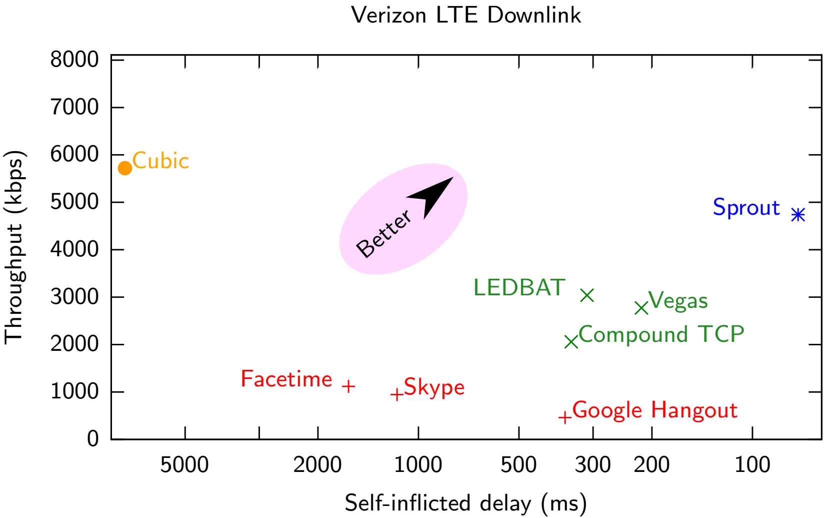
  where Z ensures that the probabilities sum to 1
- Pre-compute most of the math

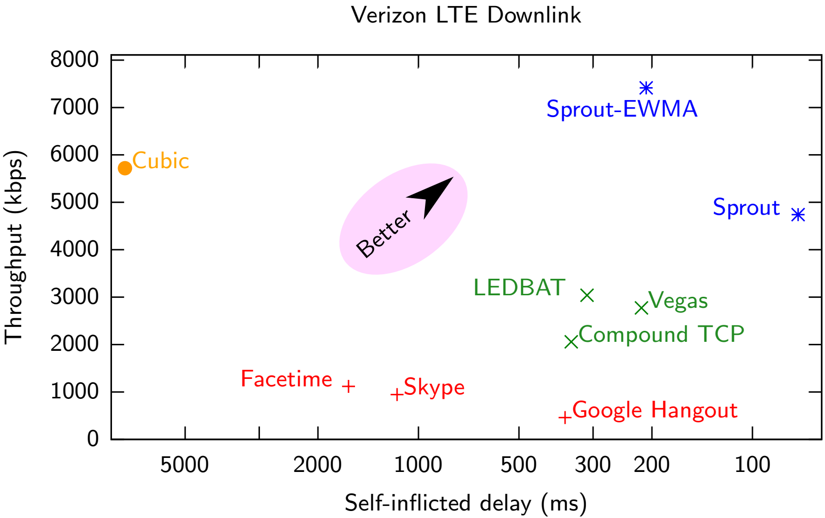
The Cautious Forecast

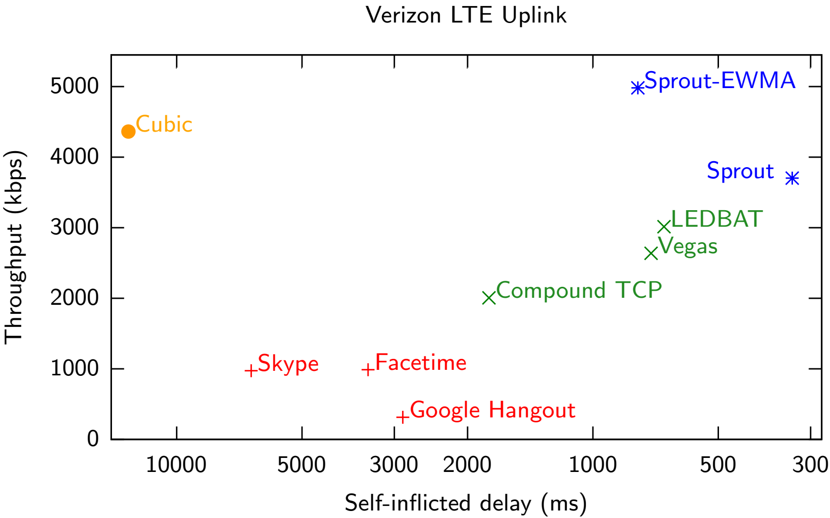
- Receiver has "cloud" of current link speeds
- For eight ticks in the future:
  - Predict future link rate by simulating Brownian motion of rates
  - Find 5th percentile of cumulative packets
- Send forecast to sender on ACKs
- Most of the math is pre-computed

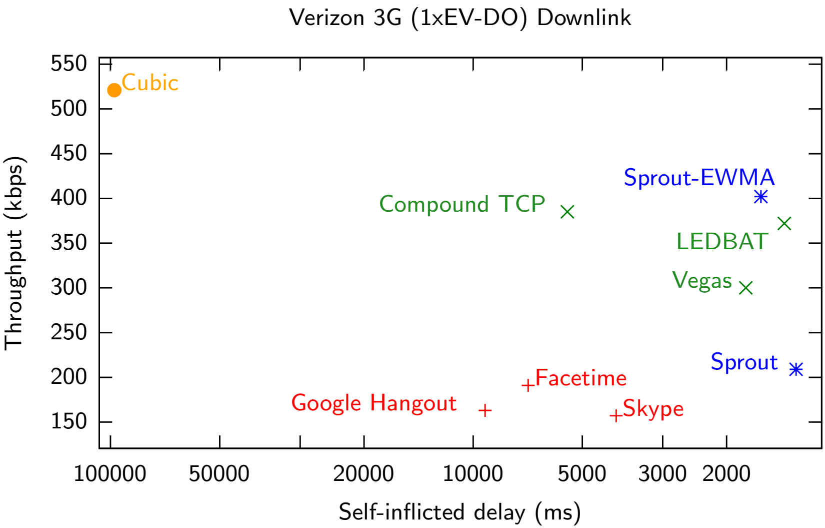

Verizon LTE Downlink
Throughput (kbps)
8000
7000
6000
5000
4000
3000
2000
1000
0
Better
Facetime
Skype
Google Hangout
5000
2000
1000
500
300
200
100
Self-inflicted delay (ms)

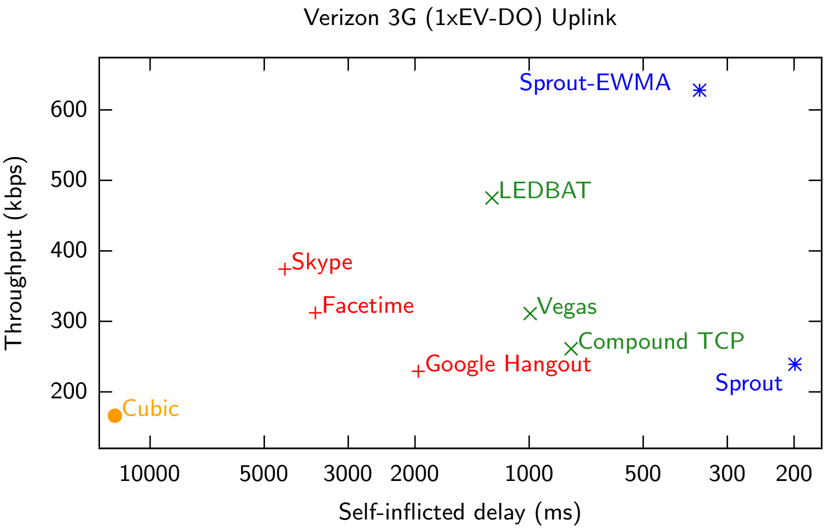

Verizon LTE Downlink
Throughput (kbps)
8000
7000
6000
5000
4000
3000
2000
1000
0
5000
2000
1000
500
300
200
100
Self-inflicted delay (ms)
Cubic
Better
LEDBAT
Vegas
Compound TCP
Facetime
Skype
Google Hangout


Verizon LTE Downlink
Throughput (kbps)
8000
7000
6000
5000
4000
3000
2000
1000
0
Self-inflicted delay (ms)
5000
2000
1000
500
300
200
100
Cubic
Better
Sprout
LEDBAT
Vegas
Compound TCP
Facetime
Skype
Google Hangout

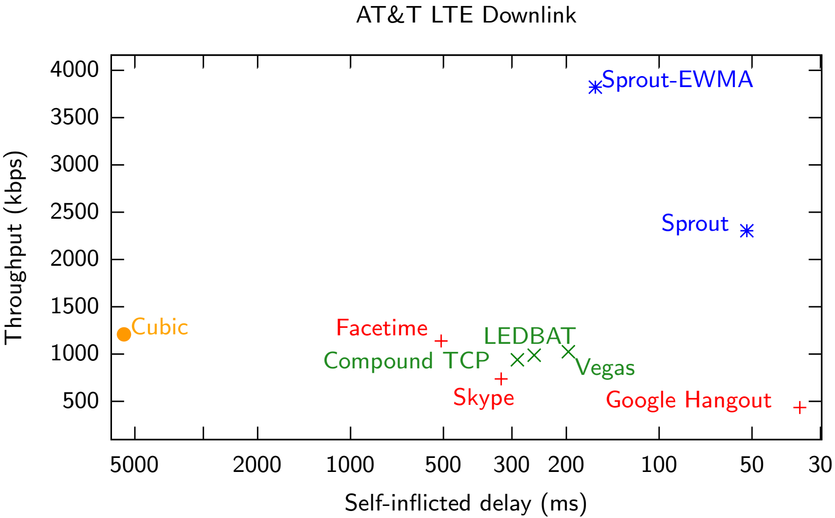

Verizon LTE Downlink
Throughput (kbps)
8000
7000
6000
5000
4000
3000
2000
1000
0
Self-inflicted delay (ms)
5000
2000
1000
500
300
200
100
Sprout-EWMA
Cubic
Better
Sprout
LEDBAT
Vegas
Compound TCP
Facetime
Skype
Google Hangout

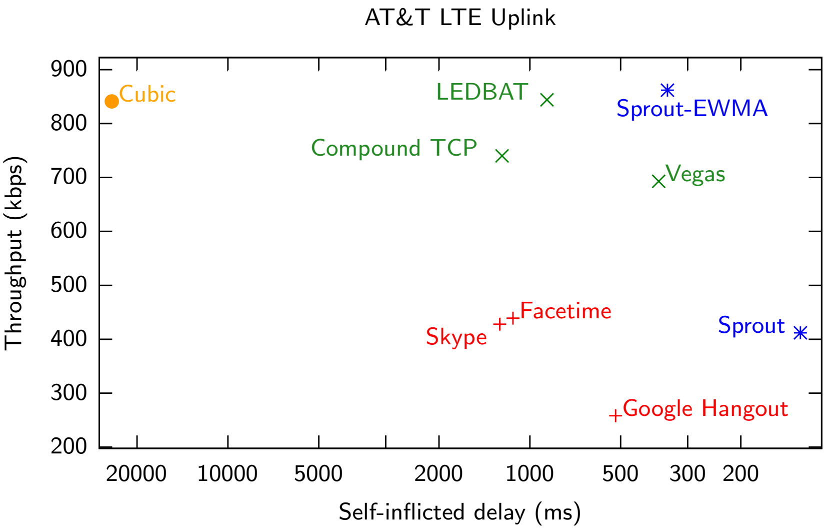

Verizon LTE Uplink
Throughput (kbps)
5000
4000
3000
2000
1000
0
10000
5000
3000
2000
1000
500
300
Self-inflicted delay (ms)
Cubic
Sprout-EWMA
Sprout
LEDBAT
Vegas
Compound TCP
Skype
Facetime
Google Hangout

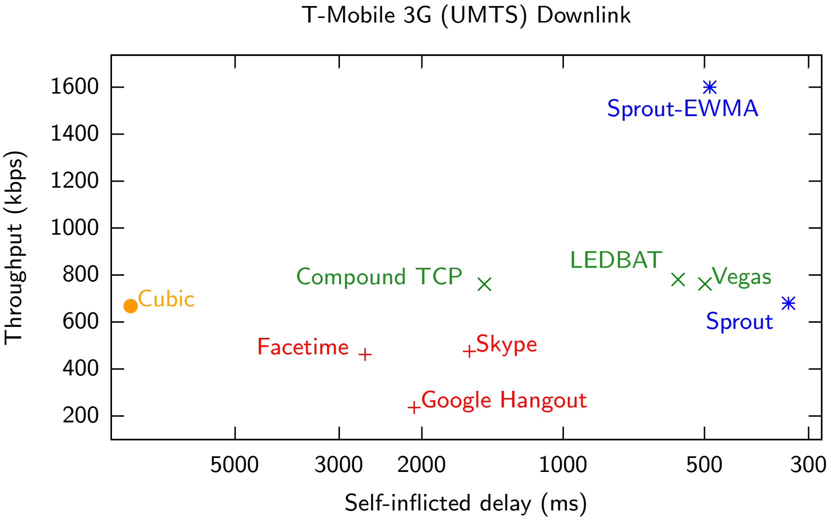

Verizon 3G (1xEV-DO) Downlink
Throughput (kbps)
550
500
450
400
350
300
250
200
150
Self-inflicted delay (ms)
100000
50000
20000
10000
5000
3000
2000
Cubic
Sprout-EWMA
Compound TCP
LEDBAT
Vegas
Sprout
Facetime
Google Hangout
Skype

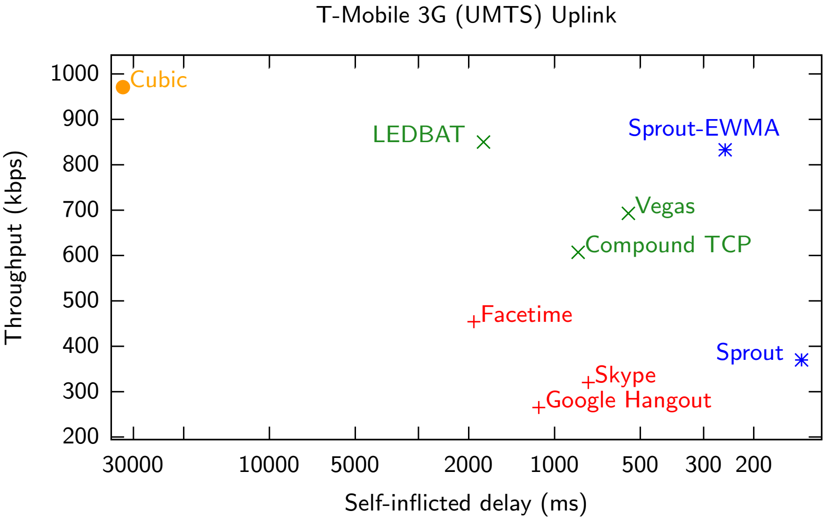


Verizon 3G (1xEV-DO) Uplink
Throughput (kbps)
Self-inflicted delay (ms)
600
500
400
300
200
10000
5000
3000
2000
1000
500
300
200
Sprout-EWMA
LEDBAT
Skype
Facetime
Vegas
Compound TCP
Google Hangout
Sprout
Cubic


AT&T LTE Downlink
Throughput (kbps)
4000
3500
3000
2500
2000
1500
1000
500
Self-inflicted delay (ms)
5000
2000
1000
500
300
200
100
50
30
Sprout-EWMA
Sprout
Cubic
Facetime
LEDBAT
Compound TCP
Vegas
Skype
Google Hangout


AT&T LTE Uplink
Throughput (kbps)
900
800
700
600
500
400
300
200
20000
10000
5000
2000
1000
500
300
200
Self-inflicted delay (ms)
Cubic
LEDBAT
Sprout-EWMA
Compound TCP
Vegas
Facetime
Skype
Sprout
Google Hangout

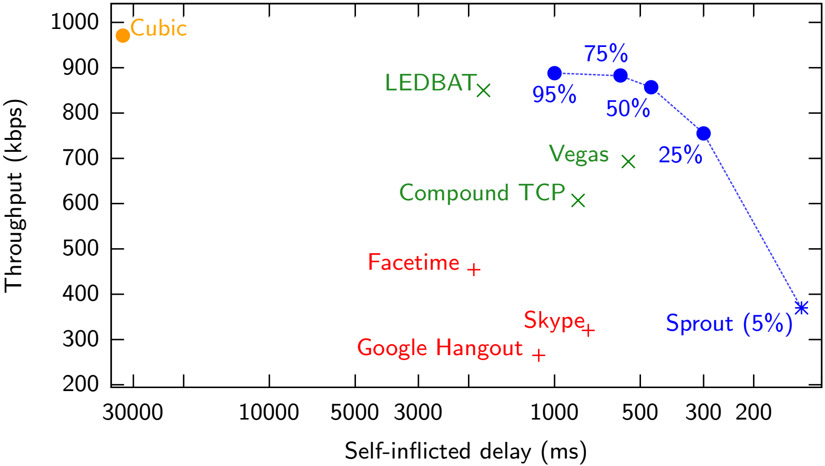

T-Mobile 3G (UMTS) Downlink
Throughput (kbps)
1600
1400
1200
1000
800
600
400
200
Self-inflicted delay (ms)
5000
3000
2000
1000
500
300
Sprout-EWMA
Compound TCP
LEDBAT
Vegas
Cubic
Sprout
Facetime
Skype
Google Hangout


T-Mobile 3G (UMTS) Uplink
Throughput (kbps)
1000
900
800
700
600
500
400
300
200
Cubic
LEDBAT
Sprout-EWMA
Vegas
Compound TCP
Facetime
Sprout
Skype
Google Hangout
30000
10000
5000
2000
1000
500
300
200
Self-inflicted delay (ms)

Overall results

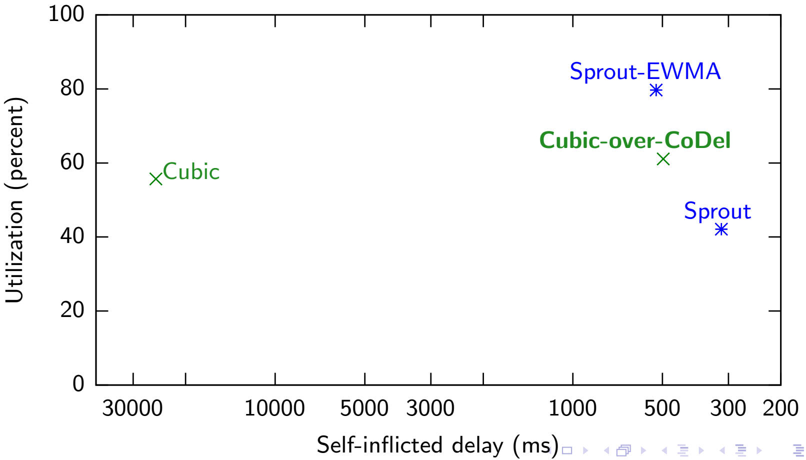
Sprout vs.	Avg. speedup	Delay reduction
Skype	2.2×	7.9×
Hangout	4.4×	7.2×
Facetime	1.9×	8.7×
Compound	1.3×	4.8×
TCP Vegas	1.1×	2.1×
LEDBAT	Same	2.8×
Cubic	0.91×	79×

Varying risk tolerance



T-Mobile 3G (UMTS) Uplink

Competes with Active Queue Mgmt (AQM) even though end-to-end



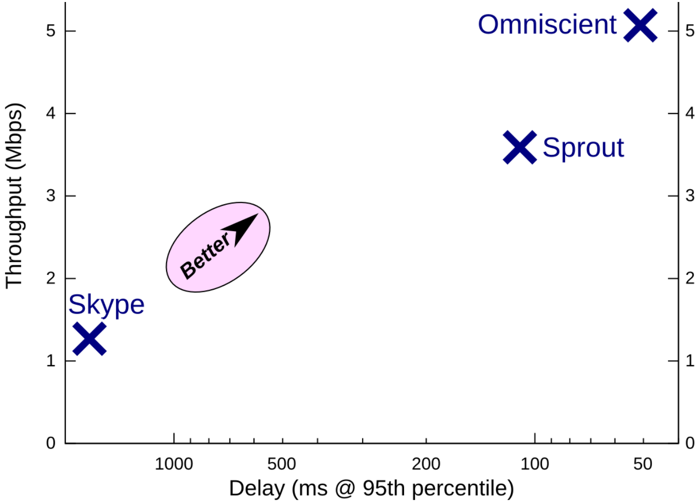
Replication by Stanford students (February–March 2013)

- Alterman & Quach reproduced a few of our measurements
- http://ReproducingNetworkResearch.wordpress.com/2013/03/12/1216/
- Won best project award in Stanford networking class!

M.I.T. 6.829 contest (March–April 2013)

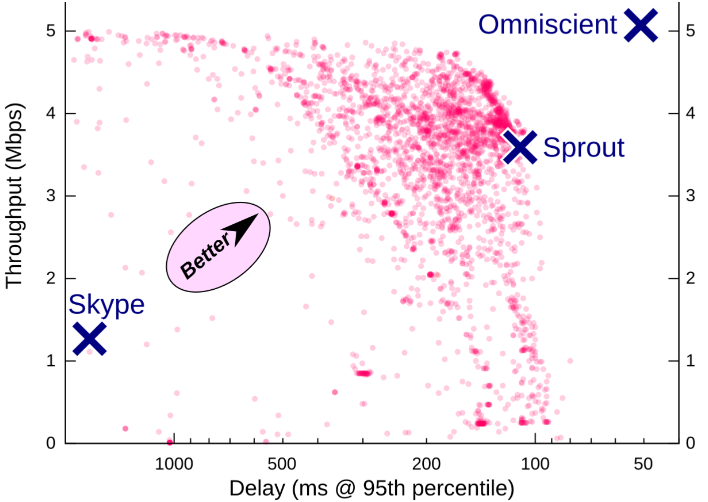
- Turnkey network emulator, evaluation
- Sender, receiver run in Linux containers
- Leaderboards

Baseline

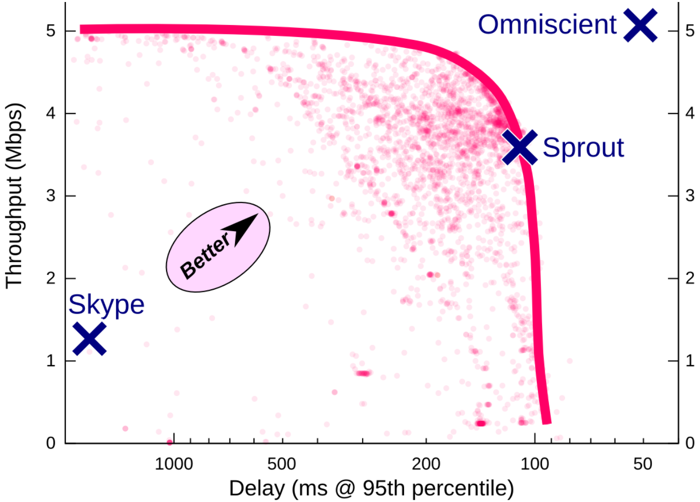


Omniscient
Sprout
Skype
Better
Throughput (Mbps)
Delay (ms @ 95th percentile)
0
1
2
3
4
5
1000
500
200
100
50

Land of 3,000 student protocols



Sprout is on the frontier



Limitations

- Stochastic model has not been tuned
- Only evaluated long-running flows.
- All testing data from Boston.
- User should wrap competing flows inside Sprout.
- Designed for cellular link with per-user queues
  - Fortunately, cells have per-device queues…
  - …but Wi-Fi generally doesn't.
- What about when the cell link *isn't* the bottleneck?

Thoughts on Methods

- Pick a model, any model.
- All models are (at some level) wrong, but they help anyway!
- See if it lands on the throughput-delay frontier.*

* (On a large set of real network paths or newly-collected traces.)

Conclusion: High Throughput + Controlled Delay Achievable over Variable Wireless Networks

- **Infer** link speed from interarrival distribution
- **Predict** future link speed
- **Control** risk of large delay with cautious forecast
- Yields 2–4$\times$ throughput of Skype, Facetime, Hangout
- Achieves 7–9$\times$ reduction in self-inflicted delay
- Matches active queue management **without router changes**
- Code and directions at `http://alfalfa.mit.edu`