Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks

Keith Winstein

MIT Computer Science and Artificial Intelligence Laboratory

http://alfalfa.mit.edu

March 6, 2014

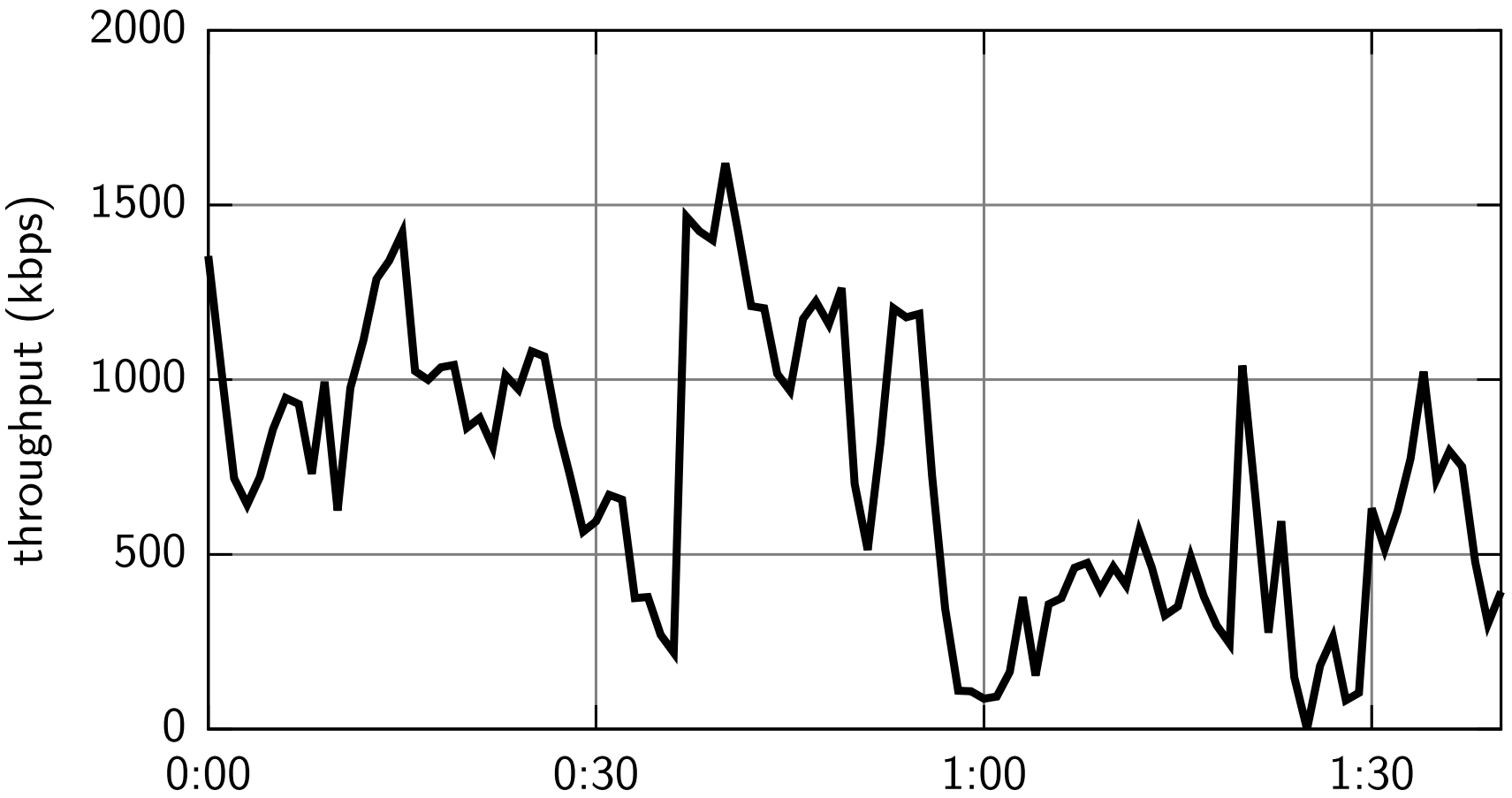
Joint work with Anirudh Sivaraman and Hari Balakrishnan.

Sprout: a transport protocol designed for variability

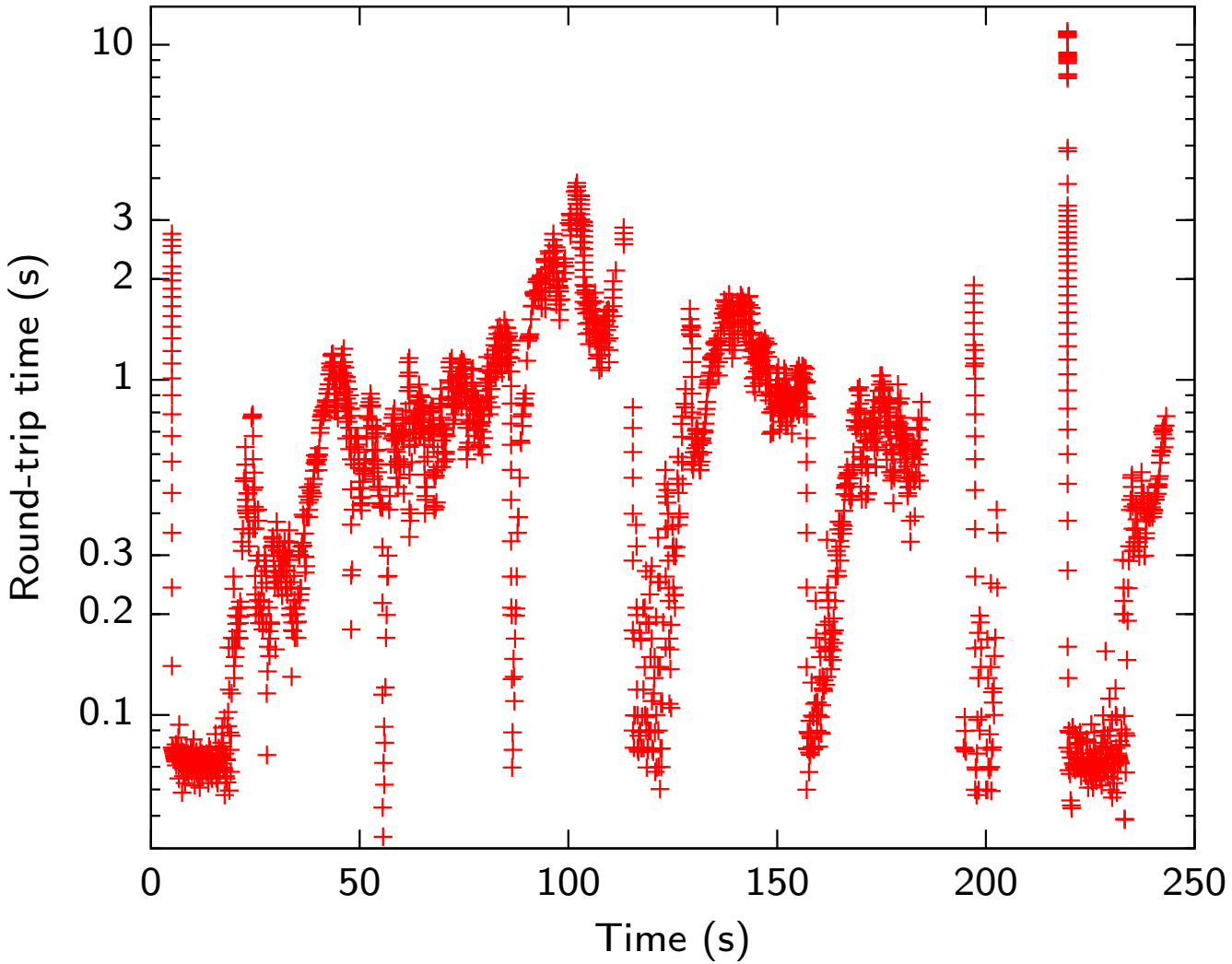
Observation:

Videoconferences perform poorly over cellular networks.

Verizon LTE uplink throughput



Verizon LTE ping delay during one TCP download

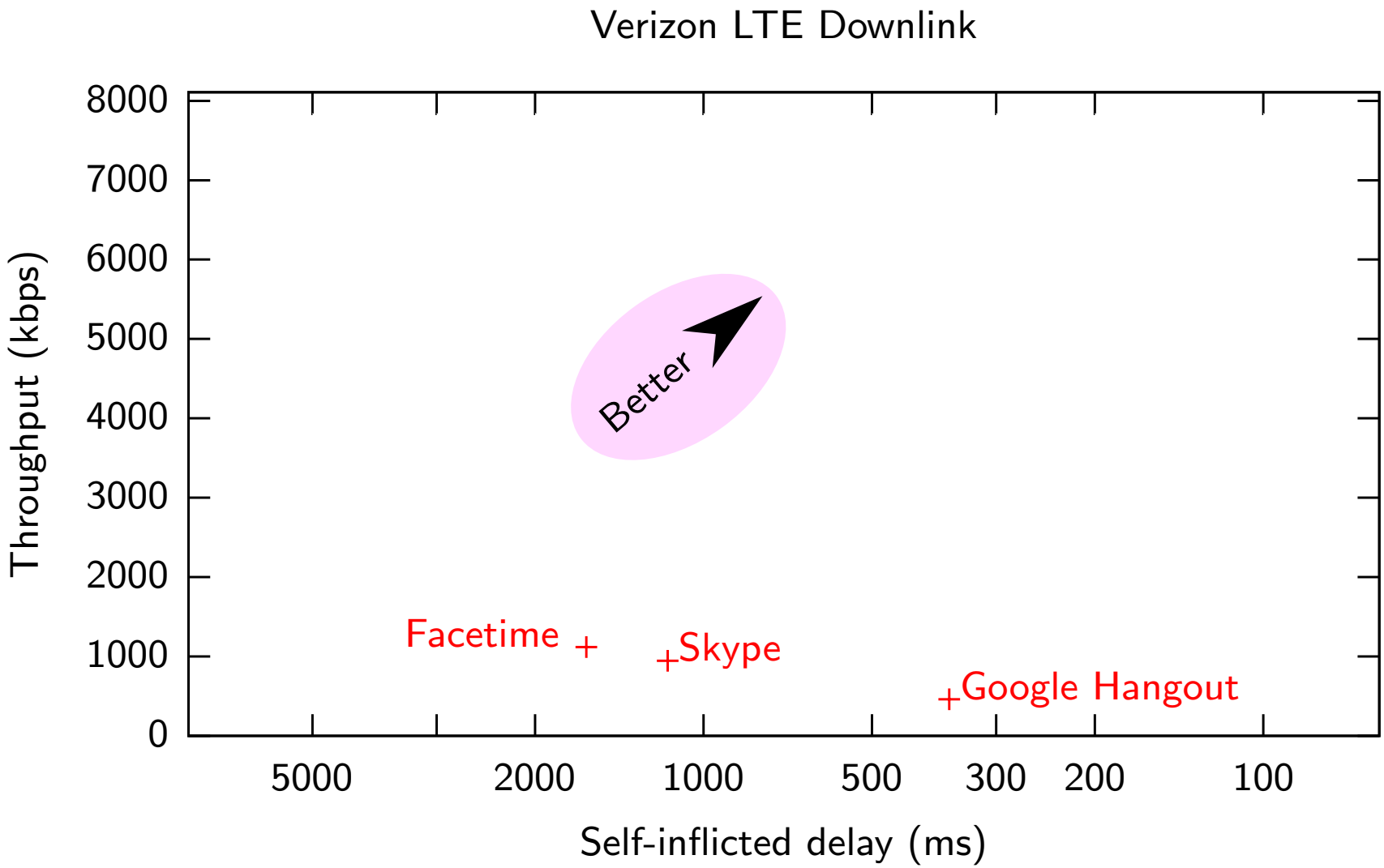


Interactive apps work poorly

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

Skype's performance

Performance summary



What's wrong?

- Existing schemes **react** to congestion signals.
  - Packet loss.
  - Increase in round-trip time.
- Feedback comes too late.
- The killer: **self-inflicted queueing delay**.

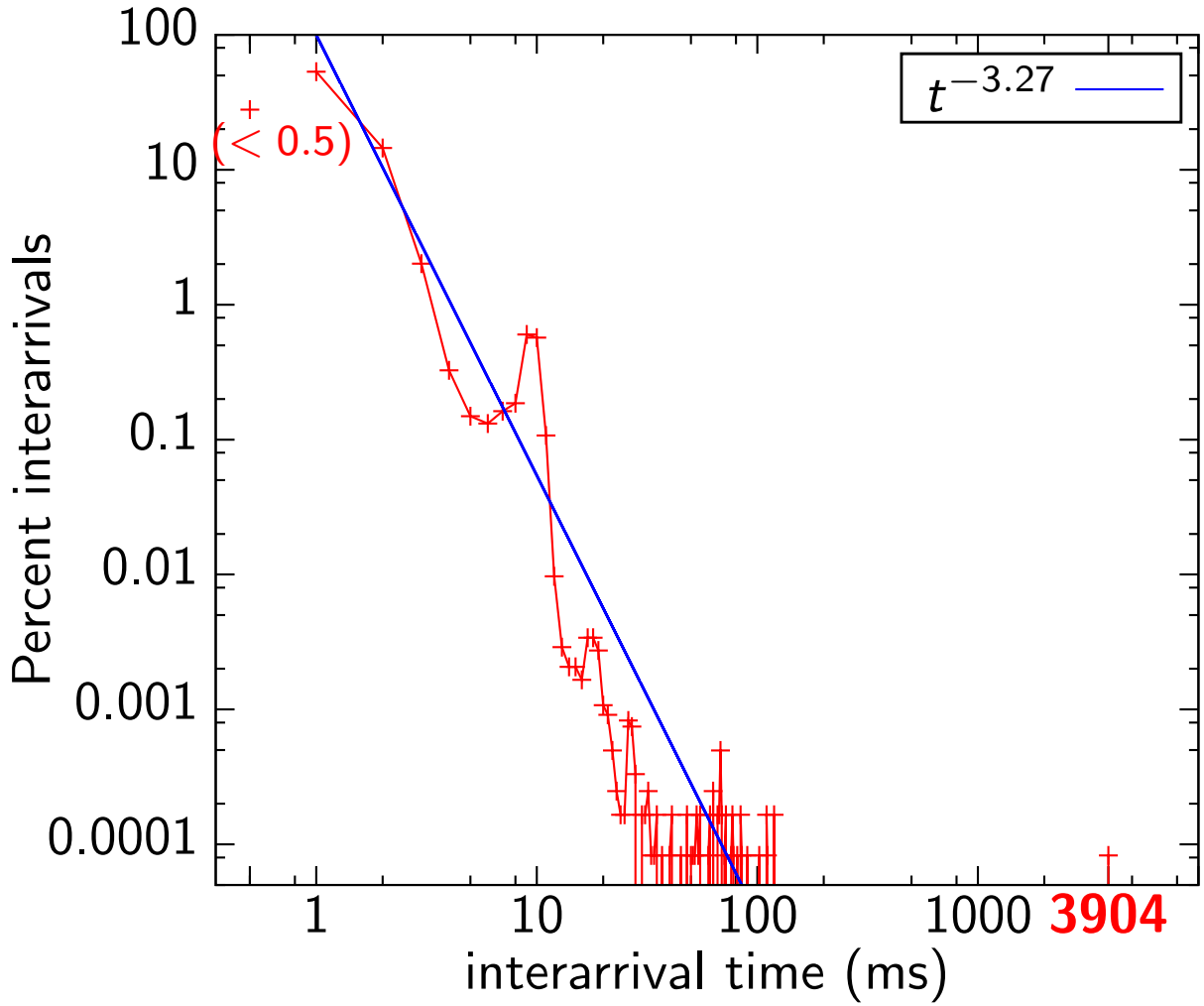
Sprout's **mission**

- Most throughput
- Bounded risk of delay $>$ 100 ms

Bounded risk of delay

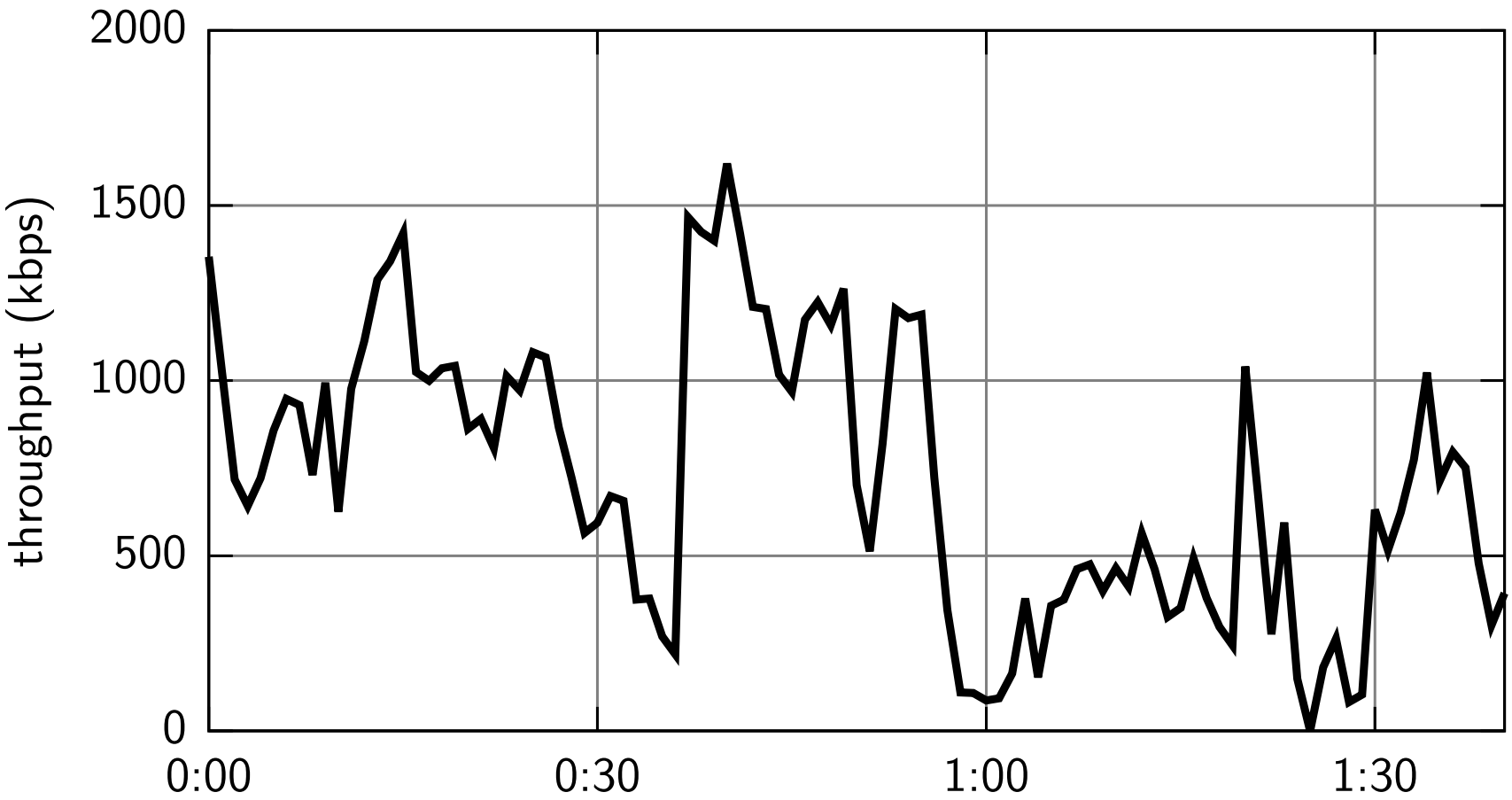
- **Model** variation in link speed
- **Infer** current link speed
- **Predict** future link speed
  - Don't wait for congestion
- **Control:** Send as much as possible, but require:
  - 95% chance all packets arrive within 100 ms

Model: packet deliveries looks like flicker noise

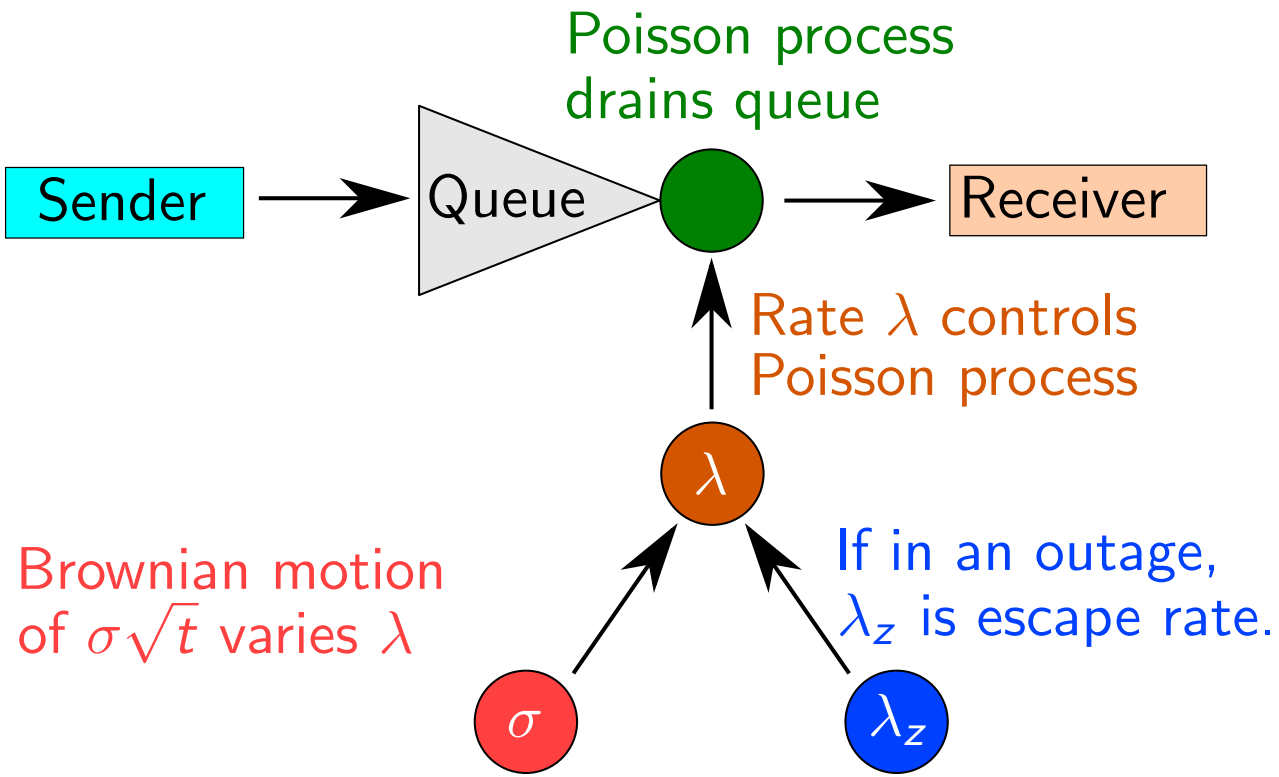


(Verizon LTE, phone stationary.)

Model: average rate looks like random walk



Sprout's **model**



Sprout's model parameters

Volatility σ: fixed @	200 $\frac{\text{pkts}/s}{\sqrt{s}}$
Expected outage time $1/\lambda_z$:	1 *s*
Tick length (τ):	20 *ms*
Forecast length:	160 *ms*
Delay target:	100 *ms*
Risk tolerance:	5%

All source code was **frozen before data collection began.**

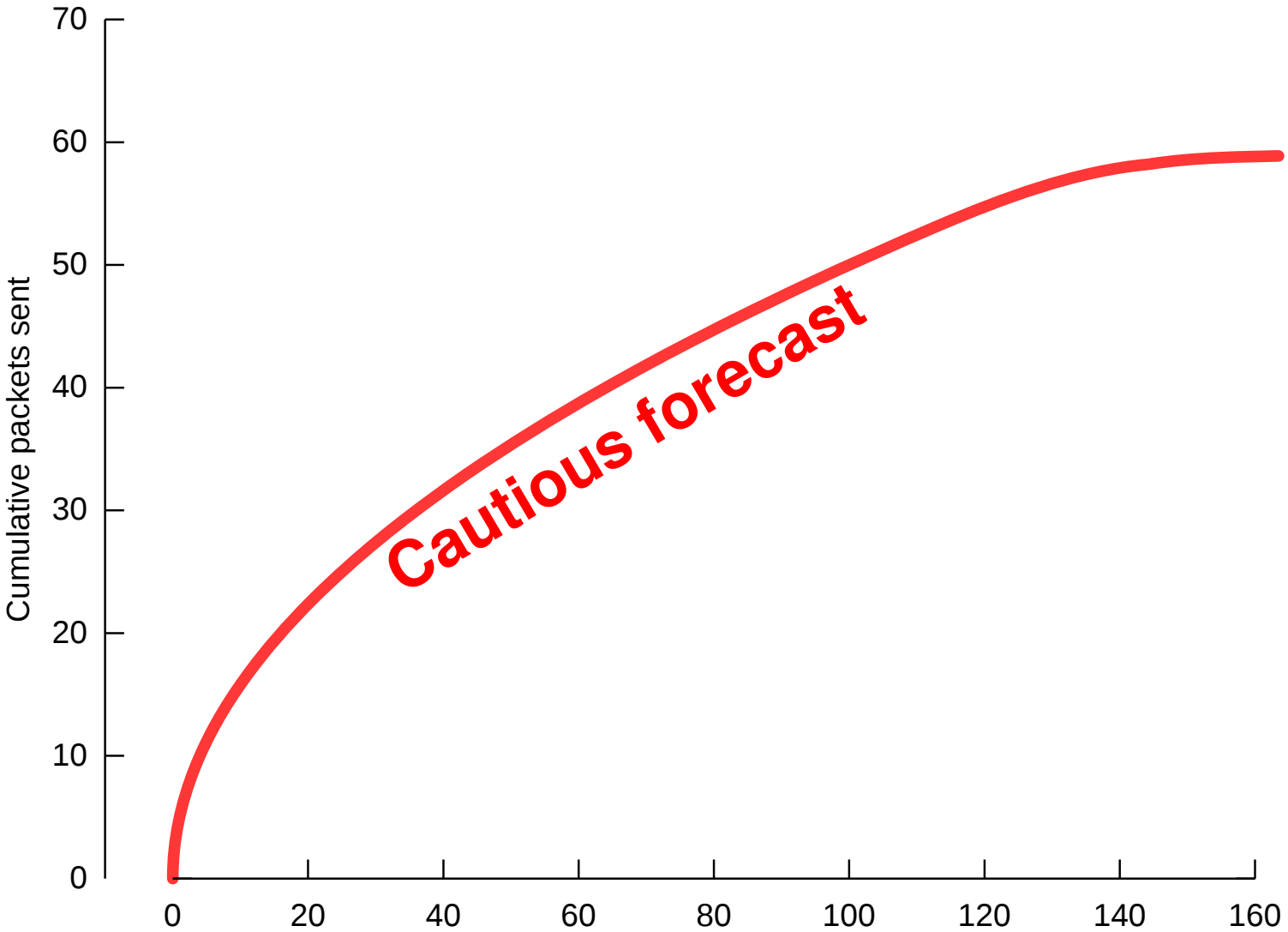
Infer: current link speed

- **Observe** packets received every τ
- **Update** $\mathrm{P}(\lambda)$

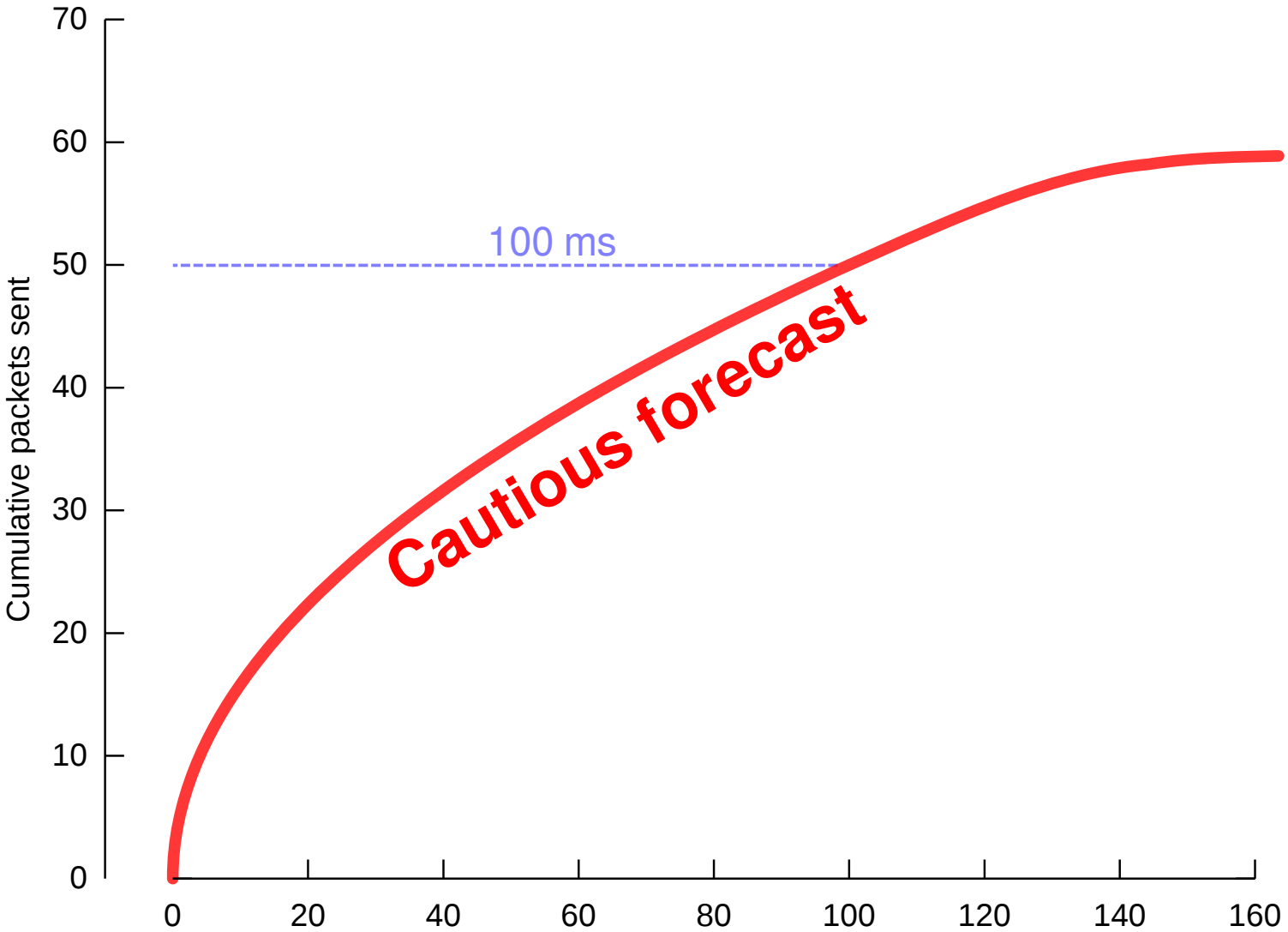
Predict: future link speed

- **Evolve** model forward
- **Forecast** 5th percentile cumulative packets

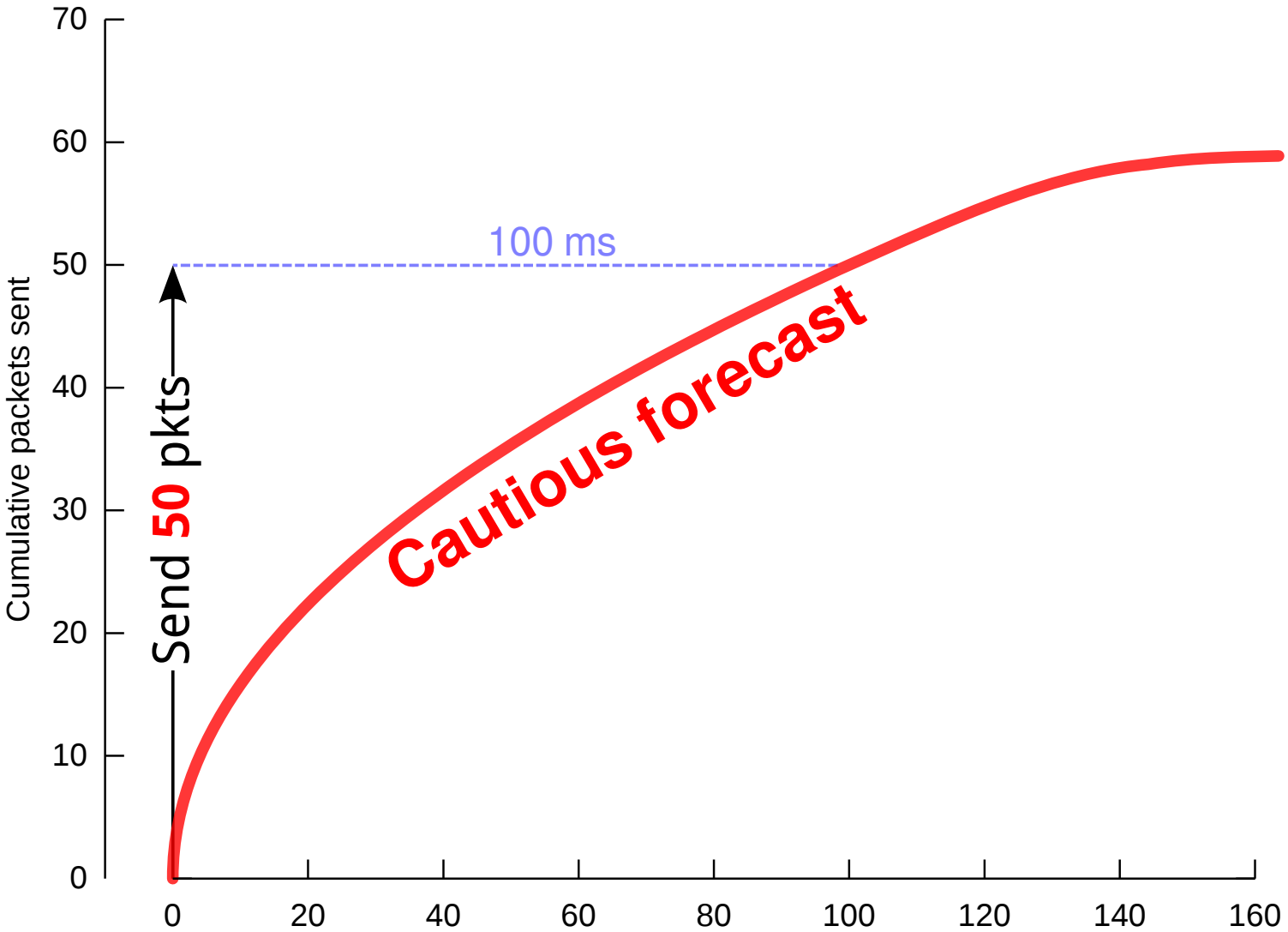
Control: fill up 100 ms forecast window



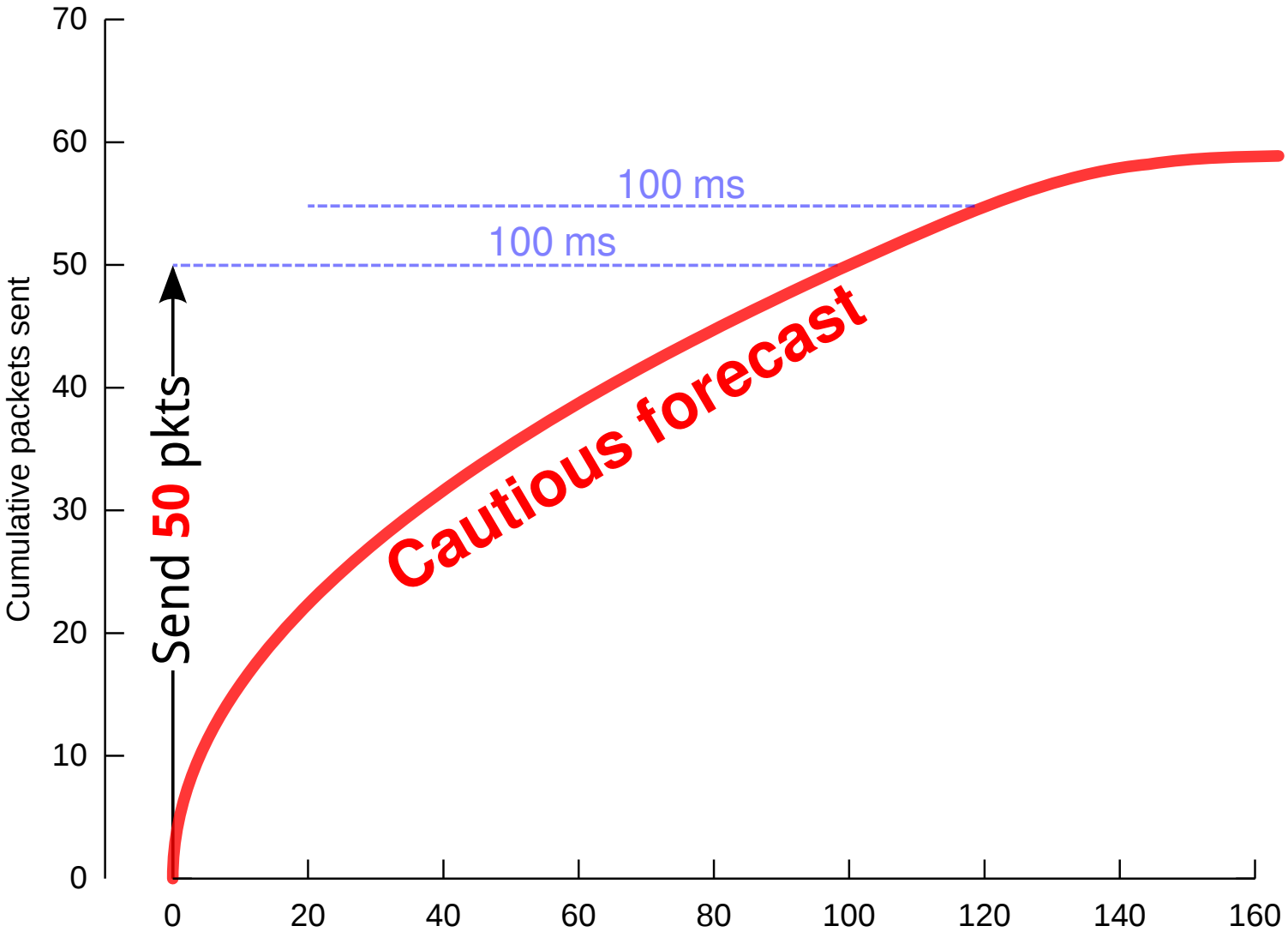
Control: fill up 100 ms forecast window



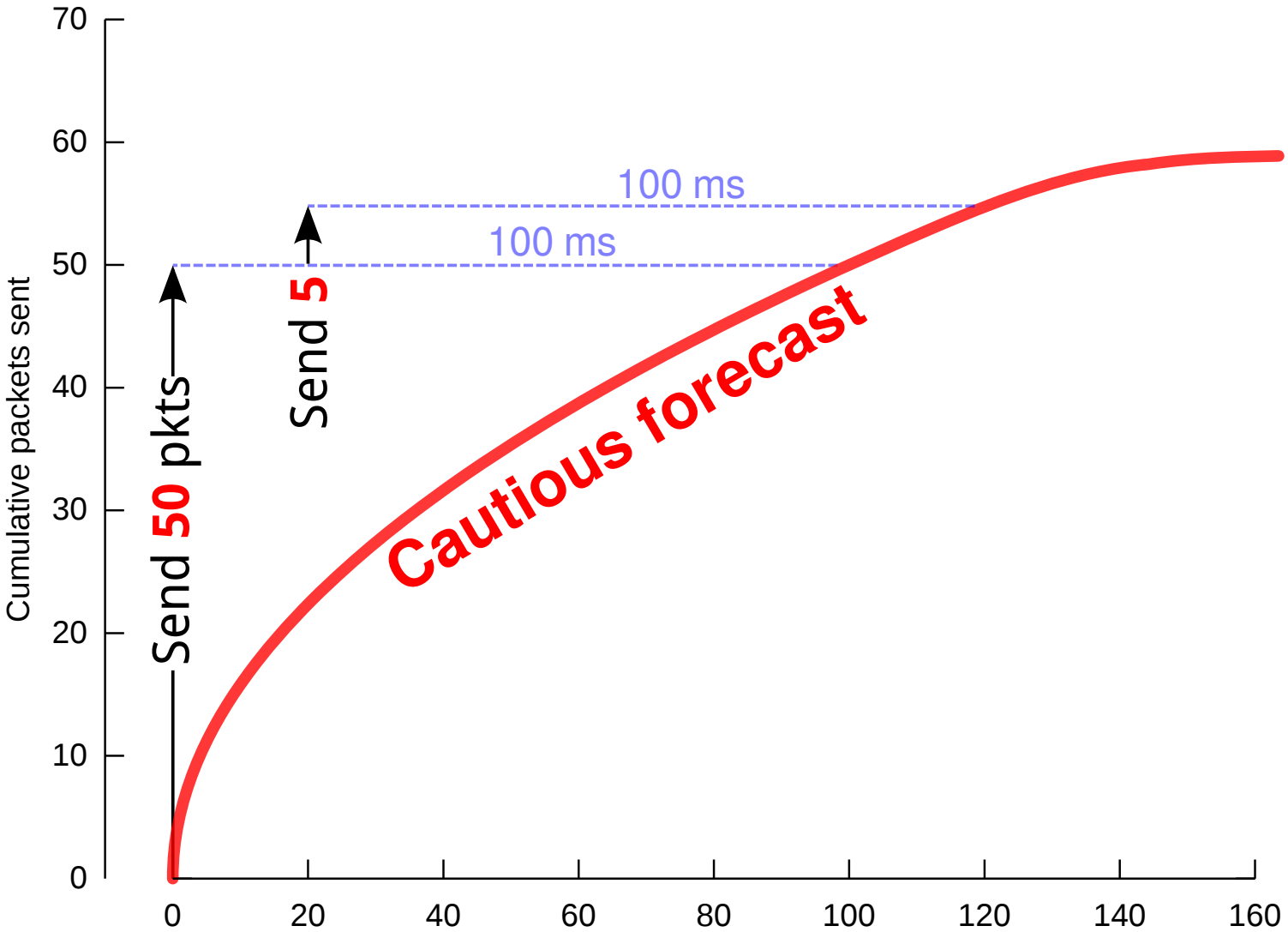
Control: fill up 100 ms forecast window



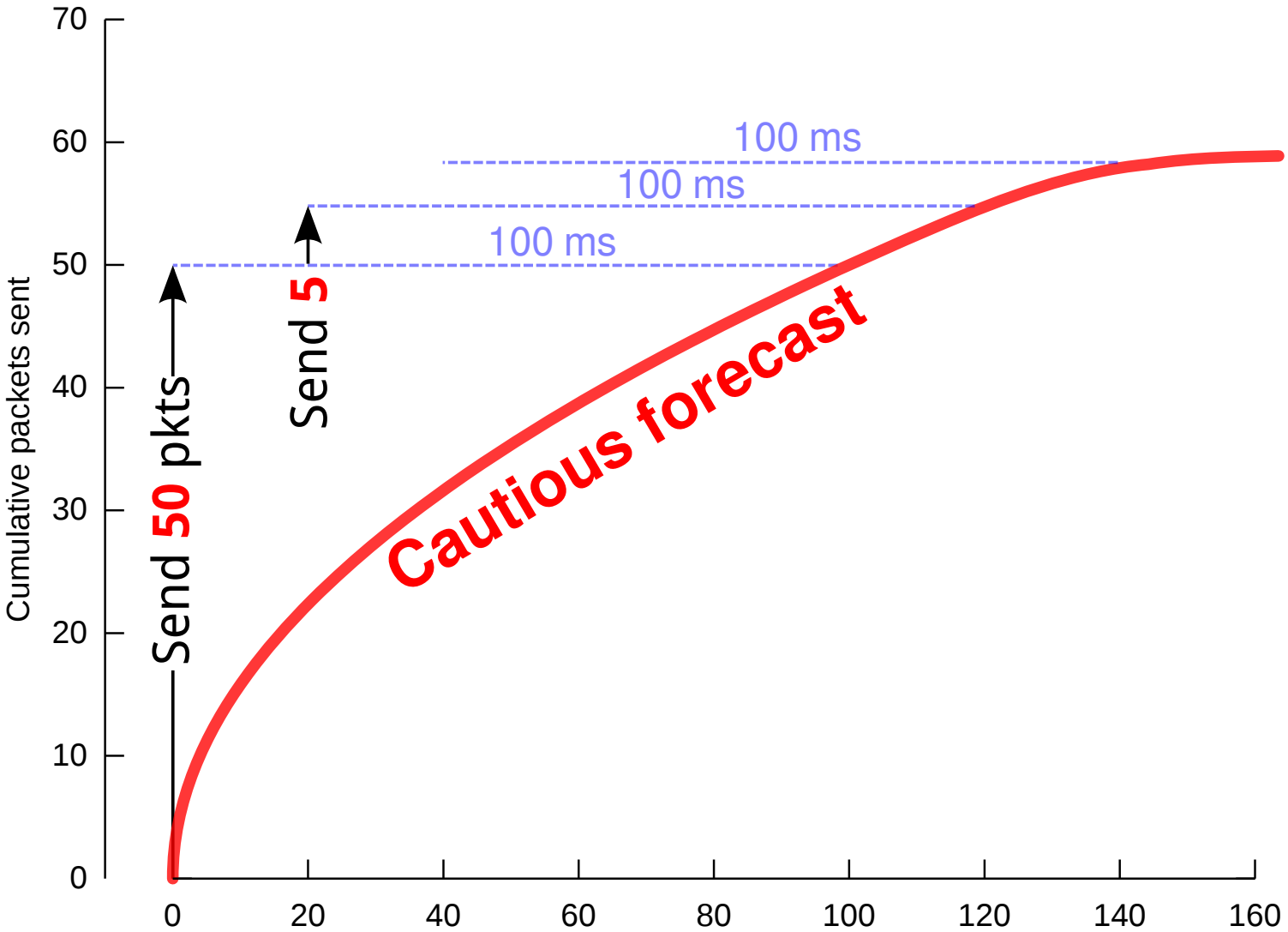
Control: fill up 100 ms forecast window



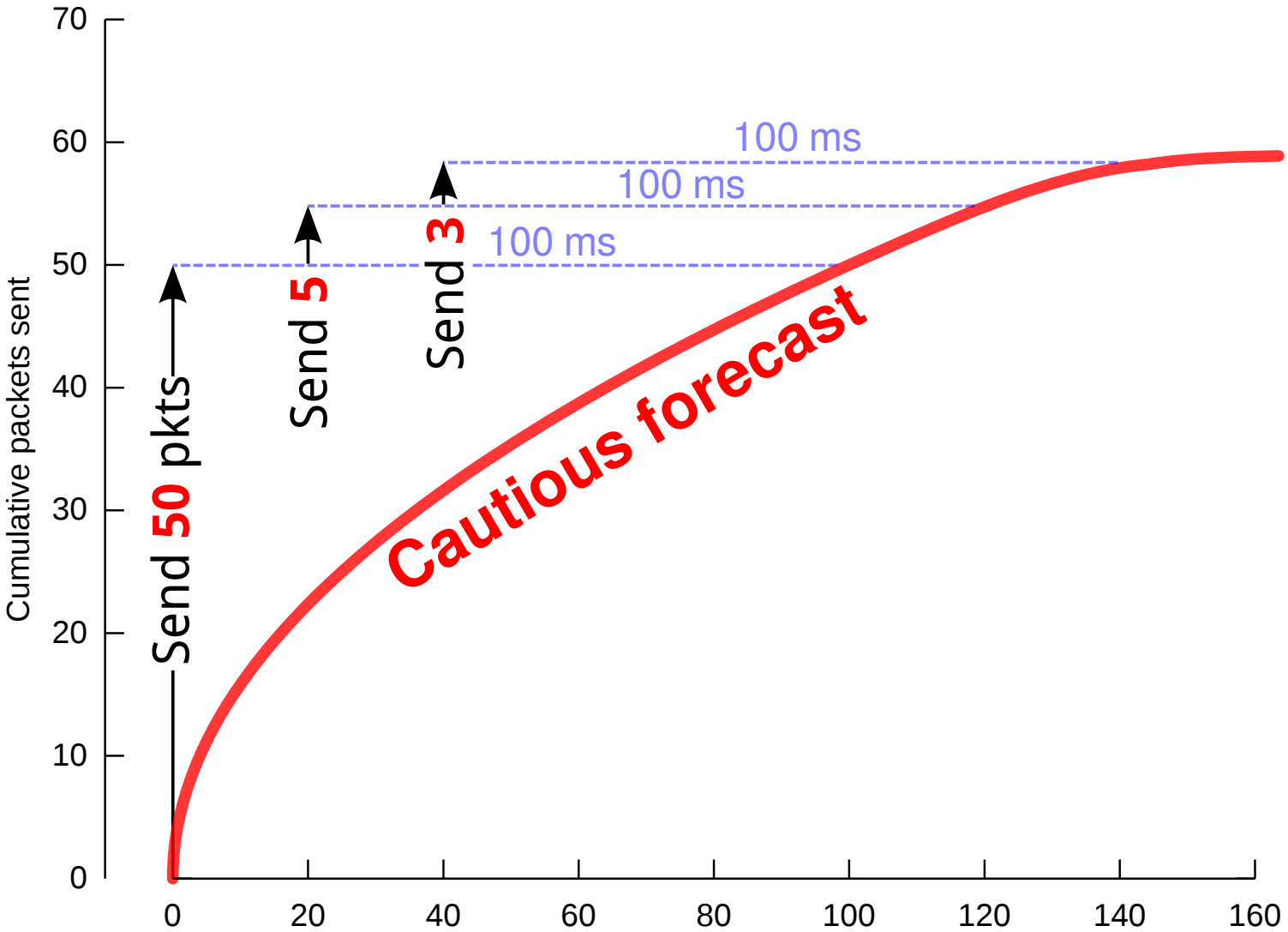
Control: fill up 100 ms forecast window



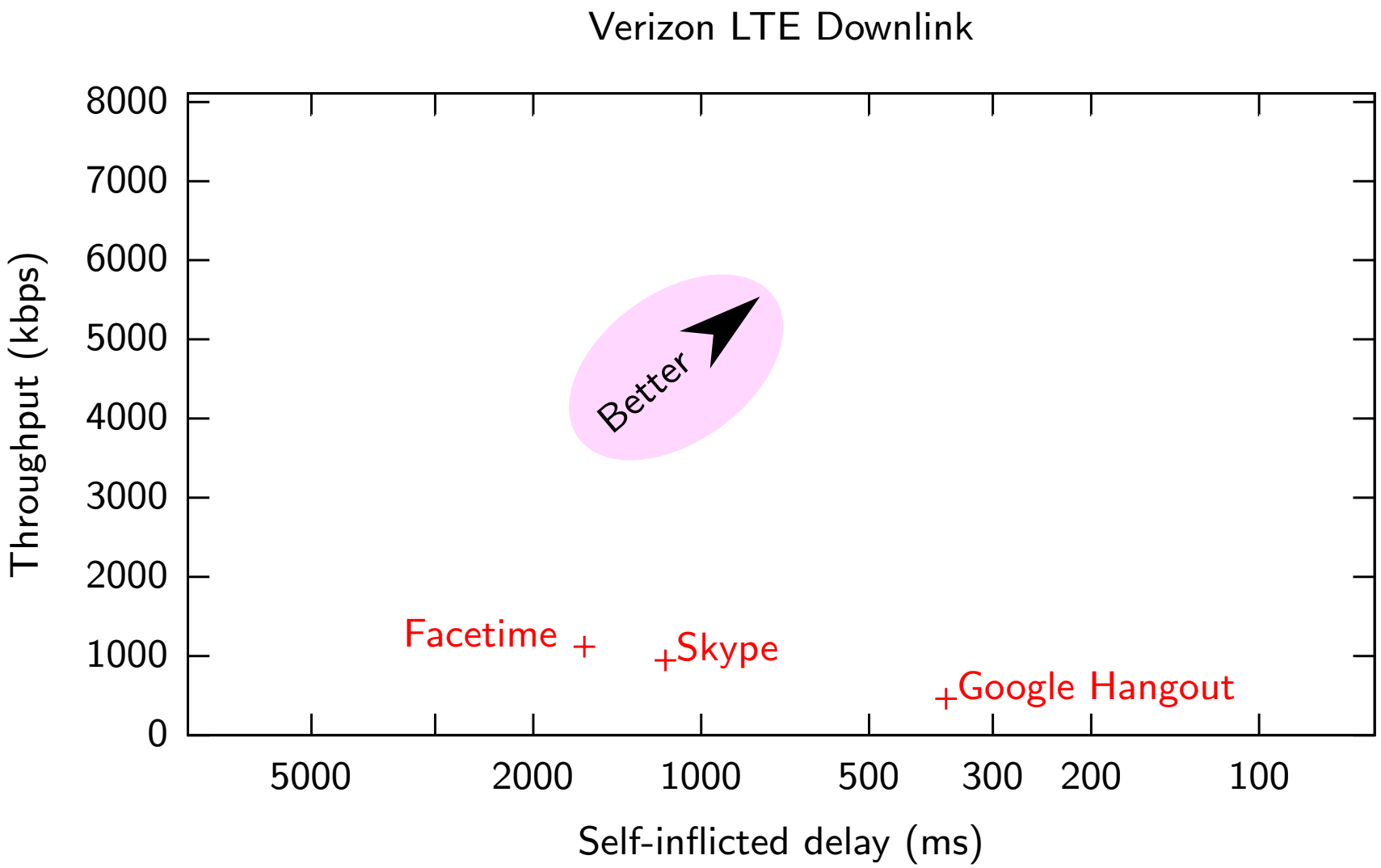
Control: fill up 100 ms forecast window

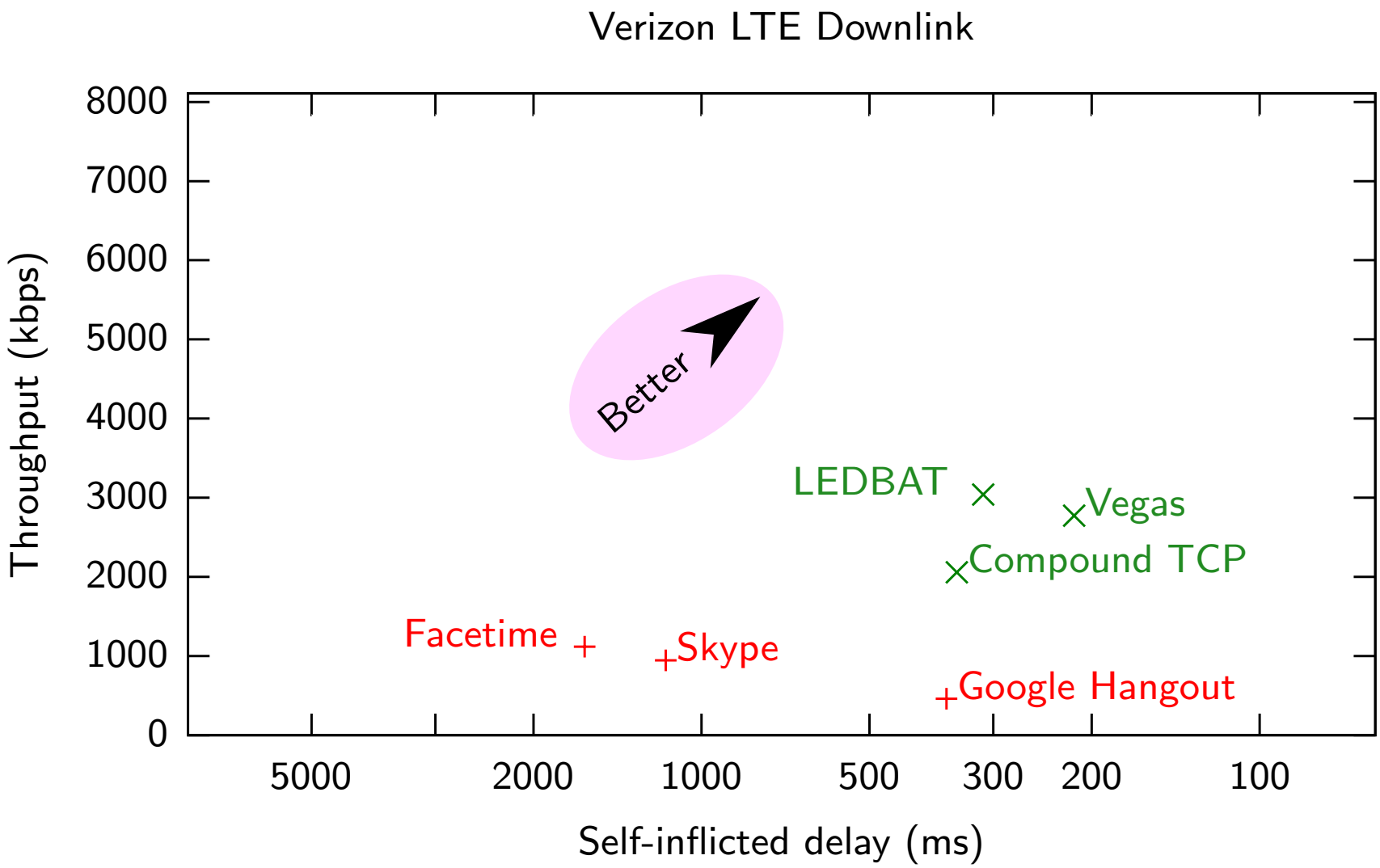


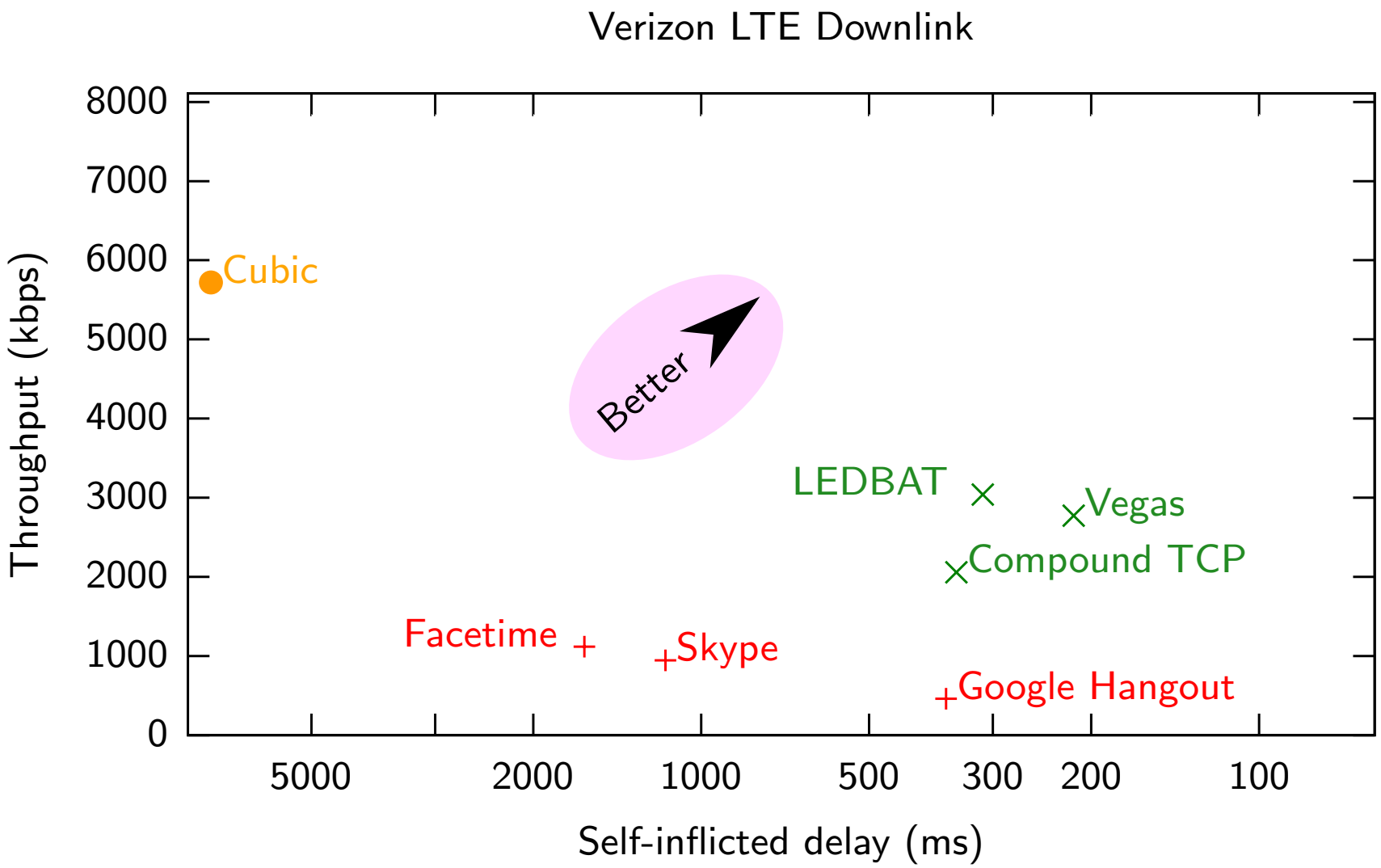
Control: fill up 100 ms forecast window

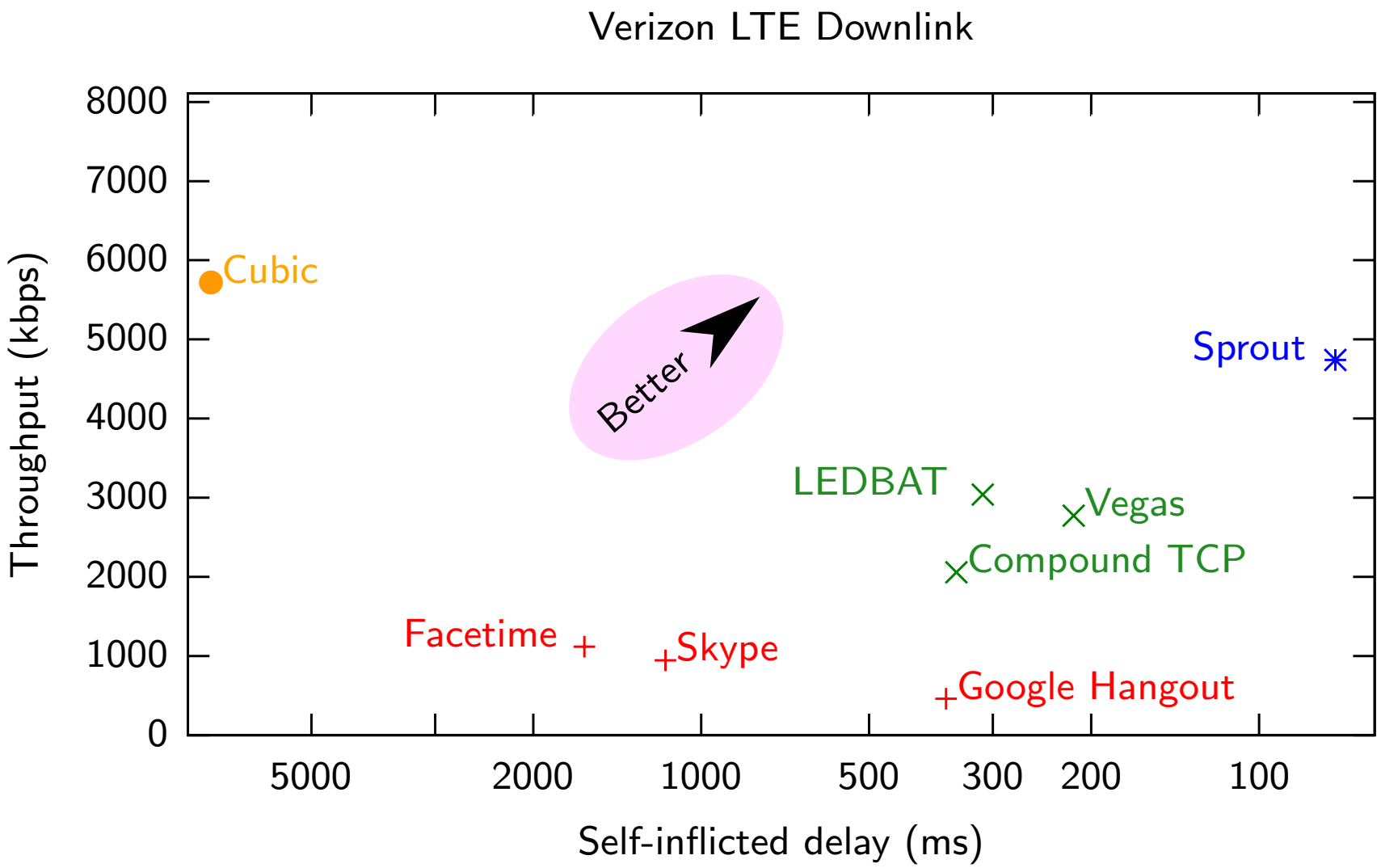


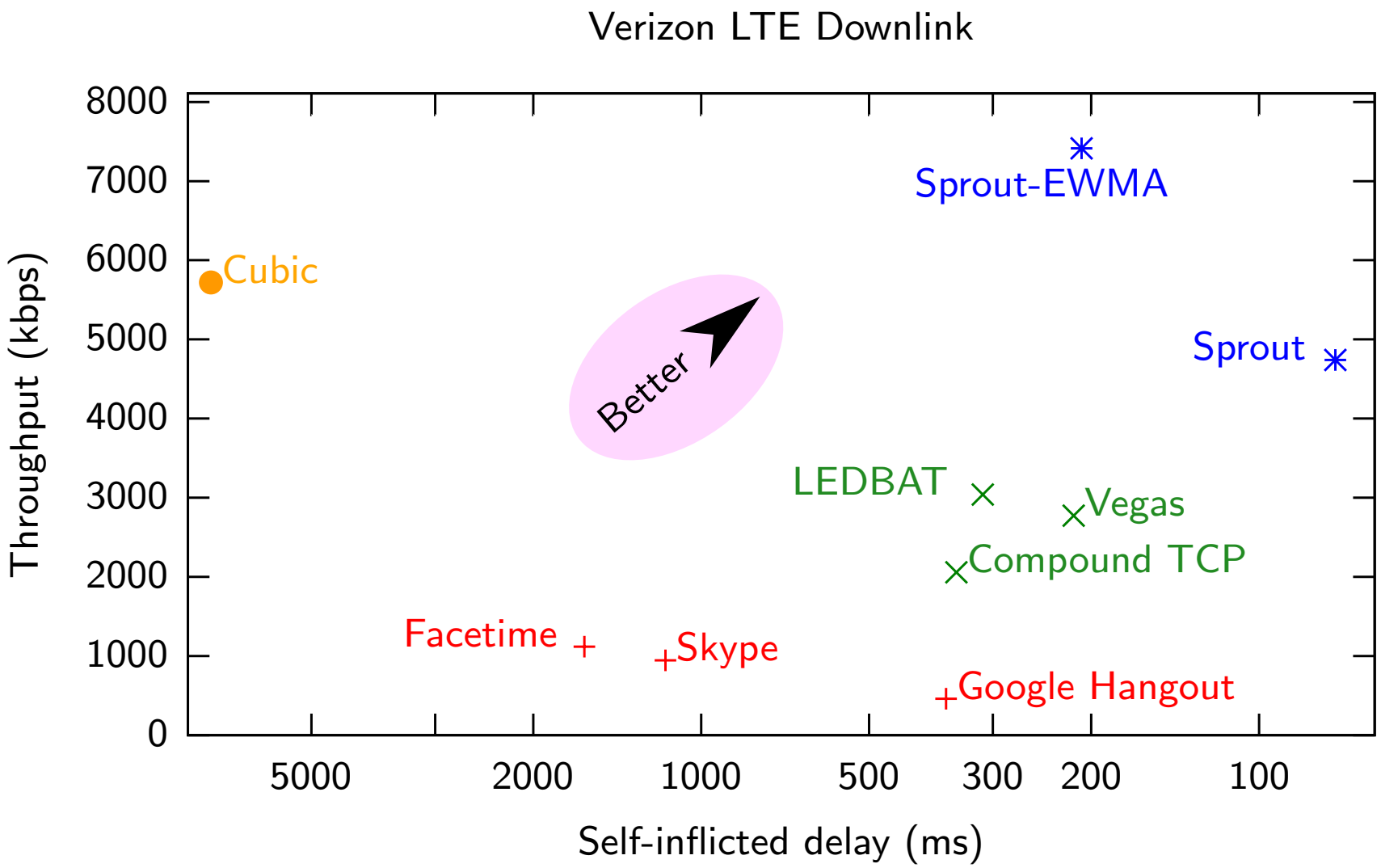
Sprout's results

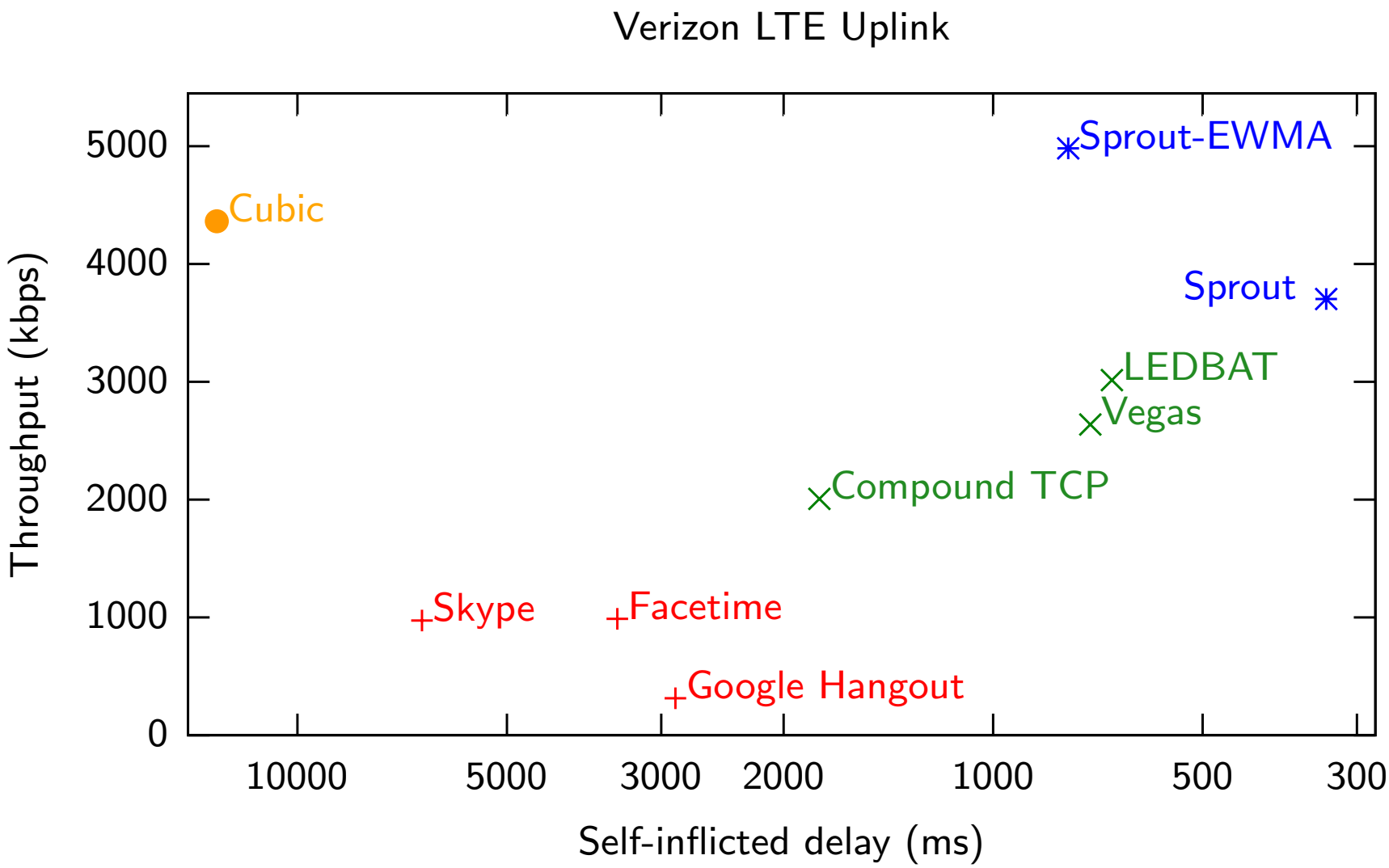

Verizon LTE Downlink
Throughput (kbps)
8000
7000
6000
5000
4000
3000
2000
1000
0
Better
Facetime
Skype
Google Hangout
5000
2000
1000
500
300
200
100
Self-inflicted delay (ms)


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)
Better
LEDBAT
Vegas
Compound TCP
Facetime
Skype
Google Hangout


Verizon LTE Downlink
Throughput (kbps)
8000
7000
6000
5000
4000
3000
2000
1000
0
Cubic
Better
LEDBAT
Vegas
Compound TCP
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
Cubic
Better
Sprout
LEDBAT
Vegas
Compound TCP
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)
Sprout-EWMA
Cubic
Better
Sprout
LEDBAT
Vegas
Compound TCP
Facetime
Skype
Google Hangout

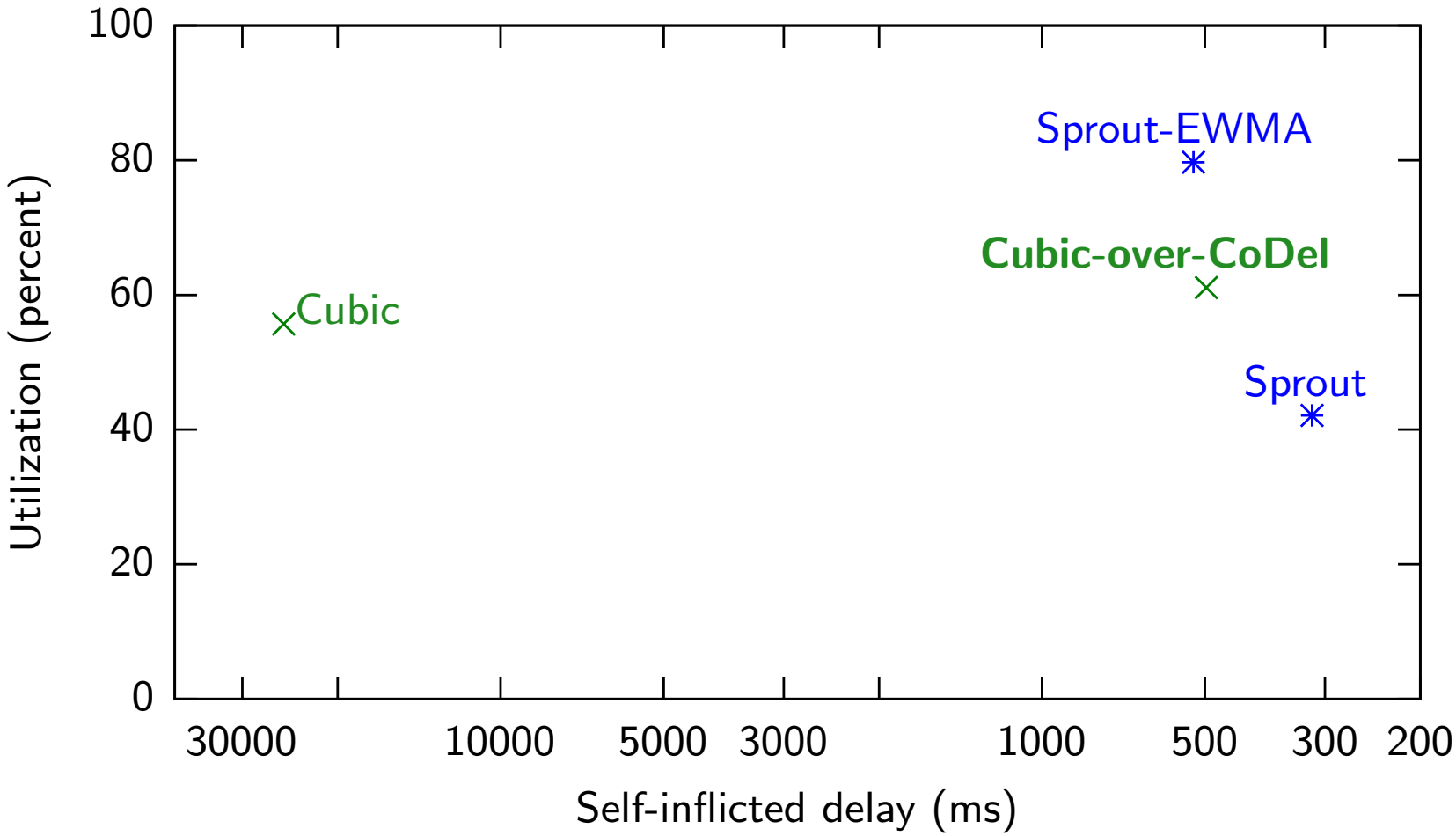

Verizon LTE Uplink
Throughput (kbps)
5000
4000
3000
2000
1000
0
Sprout-EWMA
Cubic
Sprout
LEDBAT
Vegas
Compound TCP
Skype
Facetime
Google Hangout
10000
5000
3000
2000
1000
500
300
Self-inflicted delay (ms)

Overall results on 8 links

Verizon 3G/LTE, AT&T LTE, T-Mobile 3G uplink and downlink:

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×

Sprout is end-to-end, but comparable to in-net control



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

- Turnkey network emulator, evaluation
- Sender, receiver run in Linux containers
- **Mission**: maximize throughput/delay
- 4th prize: $20
- 3rd prize: $30
- 2nd prize: $40
- (If beat Sprout) 1st prize:

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

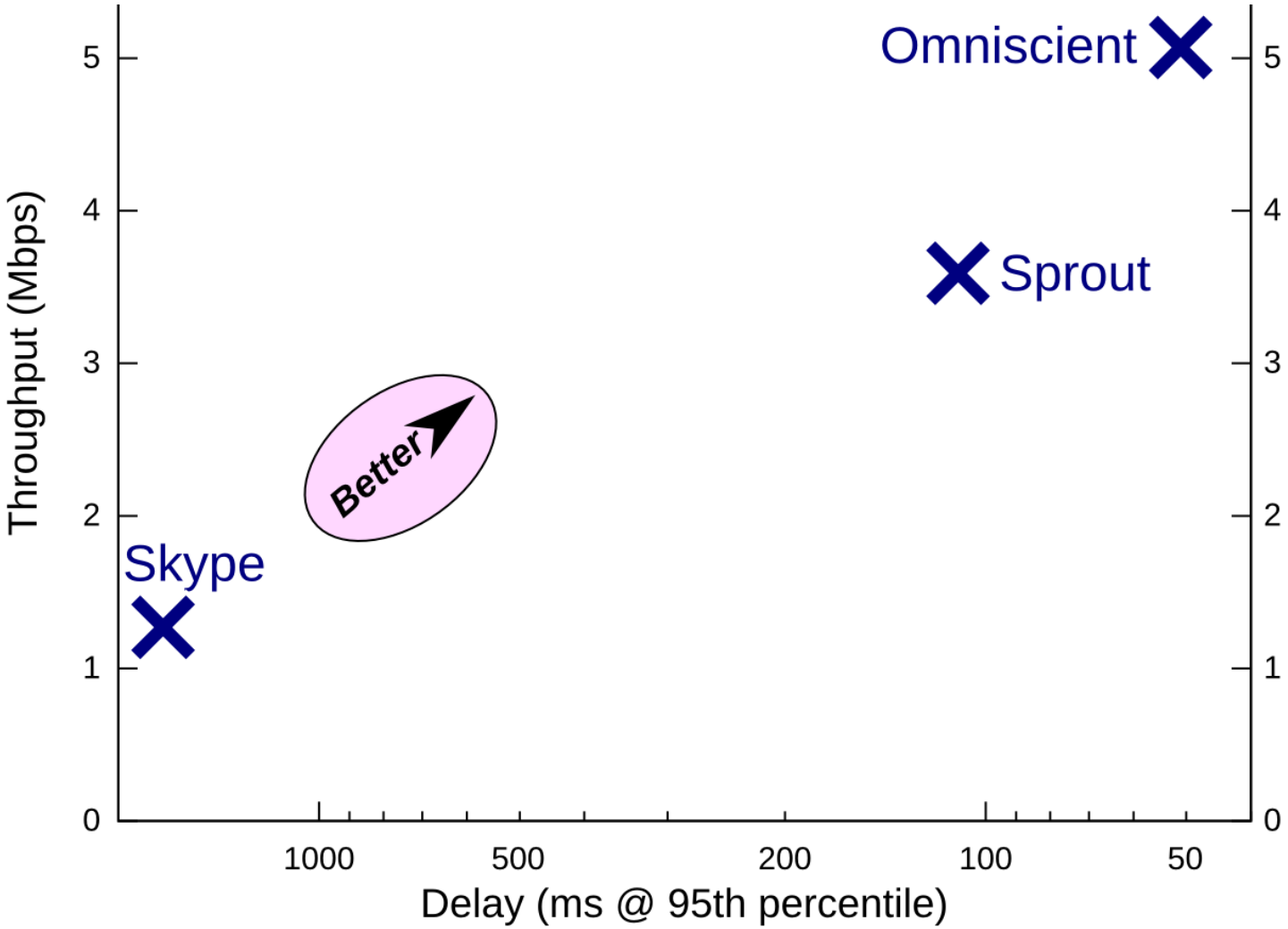
- Turnkey network emulator, evaluation
- Sender, receiver run in Linux containers
- **Mission**: maximize throughput/delay
- 4th prize: $20
- 3rd prize: $30
- 2nd prize: $40
- (If beat Sprout) 1st prize: **Co-authorship on future paper**

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

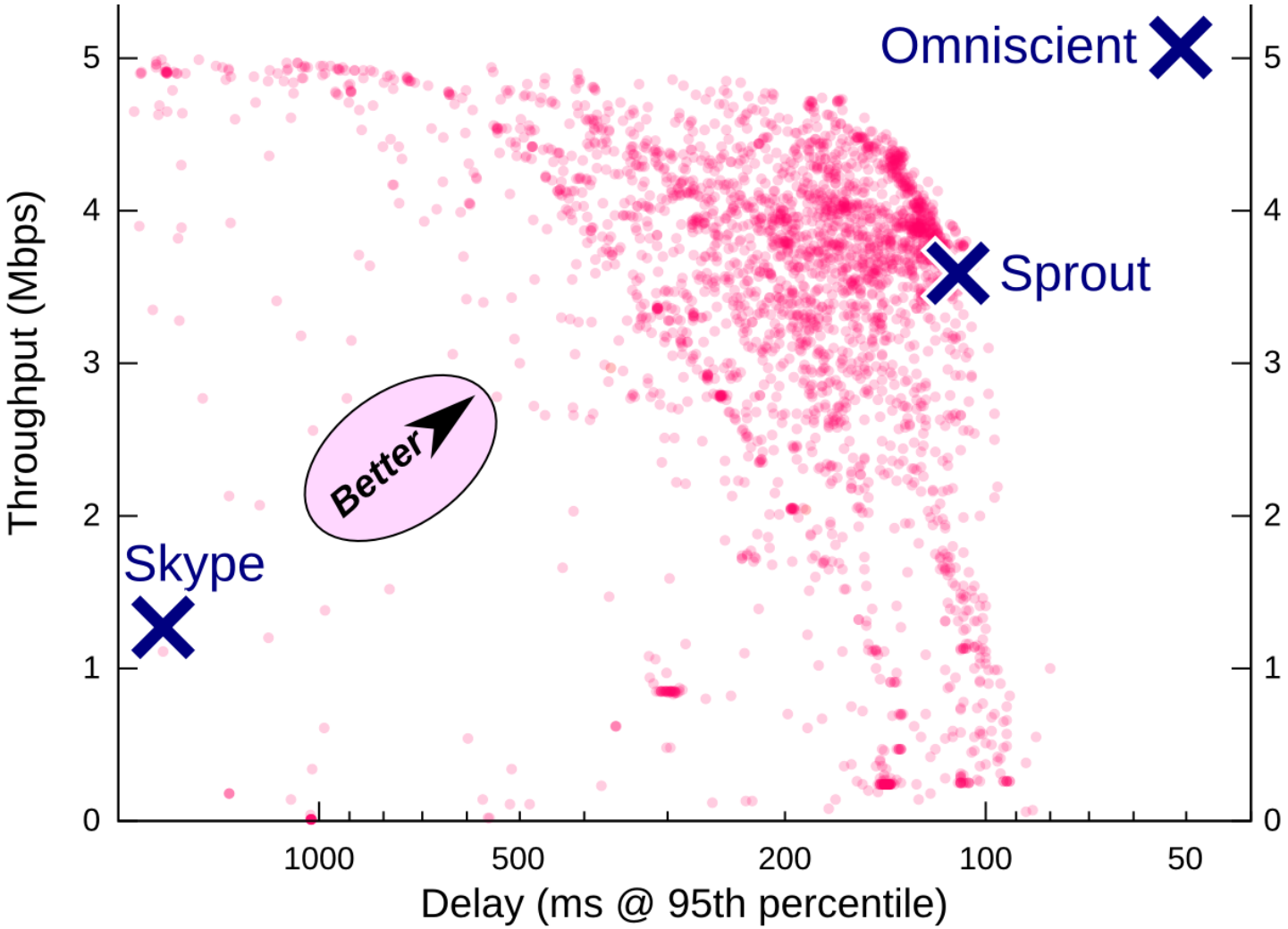
- Turnkey network emulator, evaluation
- Sender, receiver run in Linux containers
- **Mission**: maximize throughput/delay
- 4th prize: $20
- 3rd prize: $30
- 2nd prize: $40
- (If beat Sprout) 1st prize: **Co-authorship on future paper**

Anirudh Sivaraman, KW, Pauline Varley, Somak Das, Joshua Ma, Ameesh Goyal, João Batalha, and Hari Balakrishnan, **Protocol Design Contests**, *in submission*

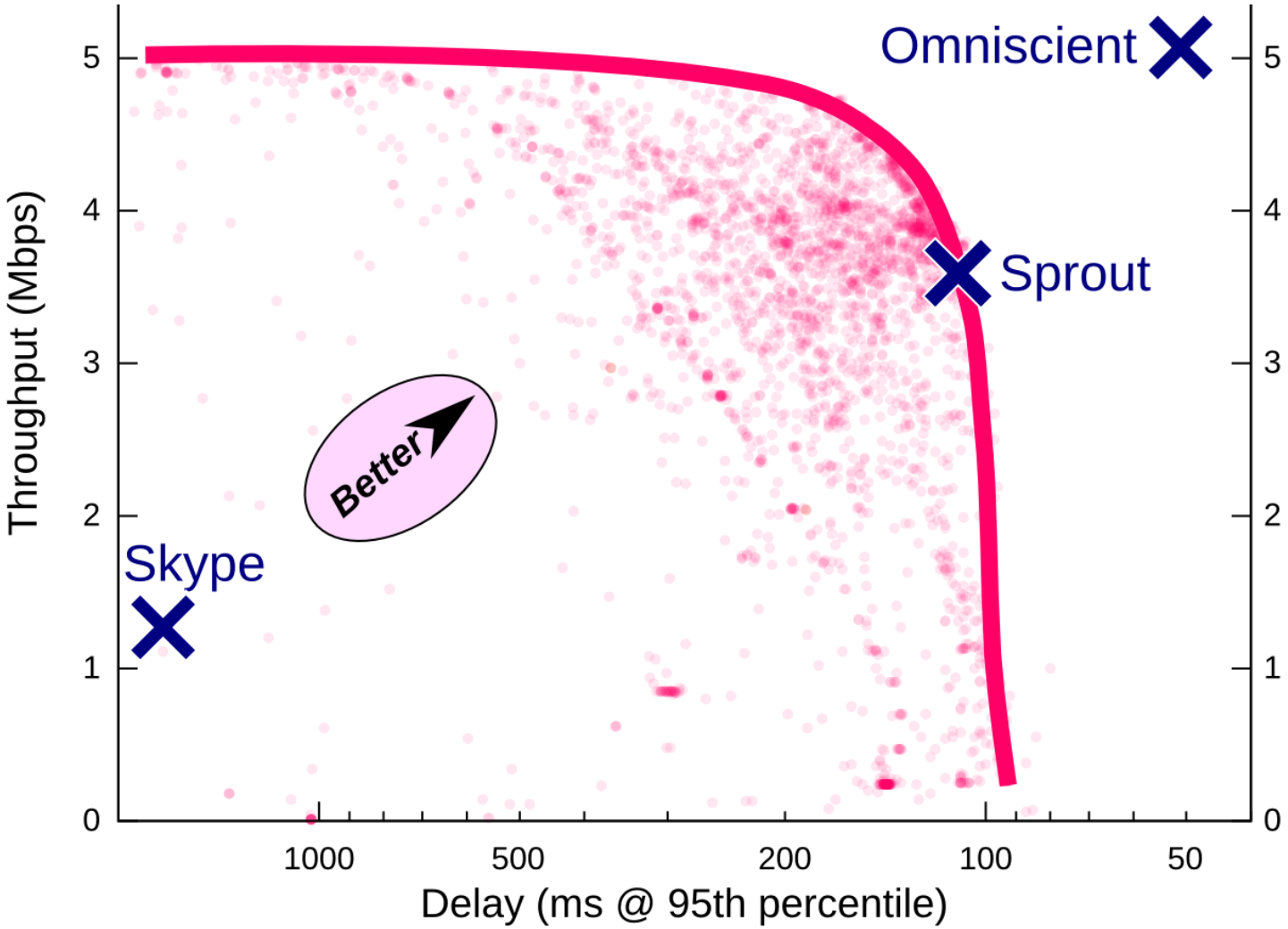
Baseline


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

Land of 3,000 student protocols



Sprout was on the frontier



Limitations

- Sprout wants to control all of the traffic on a queue.
  - Cells generally have **per-user** queues...
  - ...but Wi-Fi and wired networks usually don't.
- What if cell link *isn't* the bottleneck?
- Assumption: application always has data to send

Our approach

- Pick a model, any model.
- All models are wrong, but they help anyway!
- See if it lands on the frontier.*

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

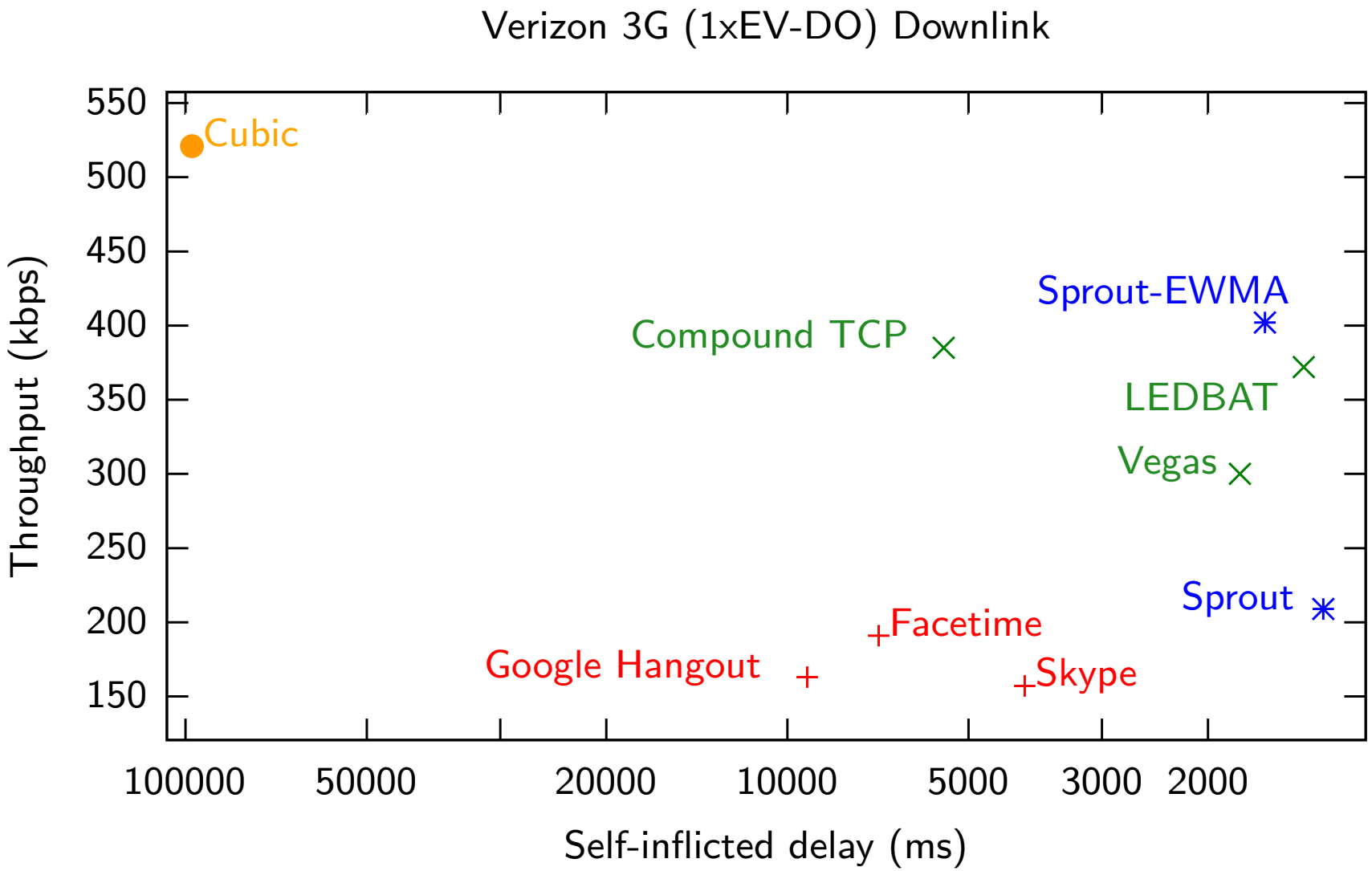
- Kaizen for congestion

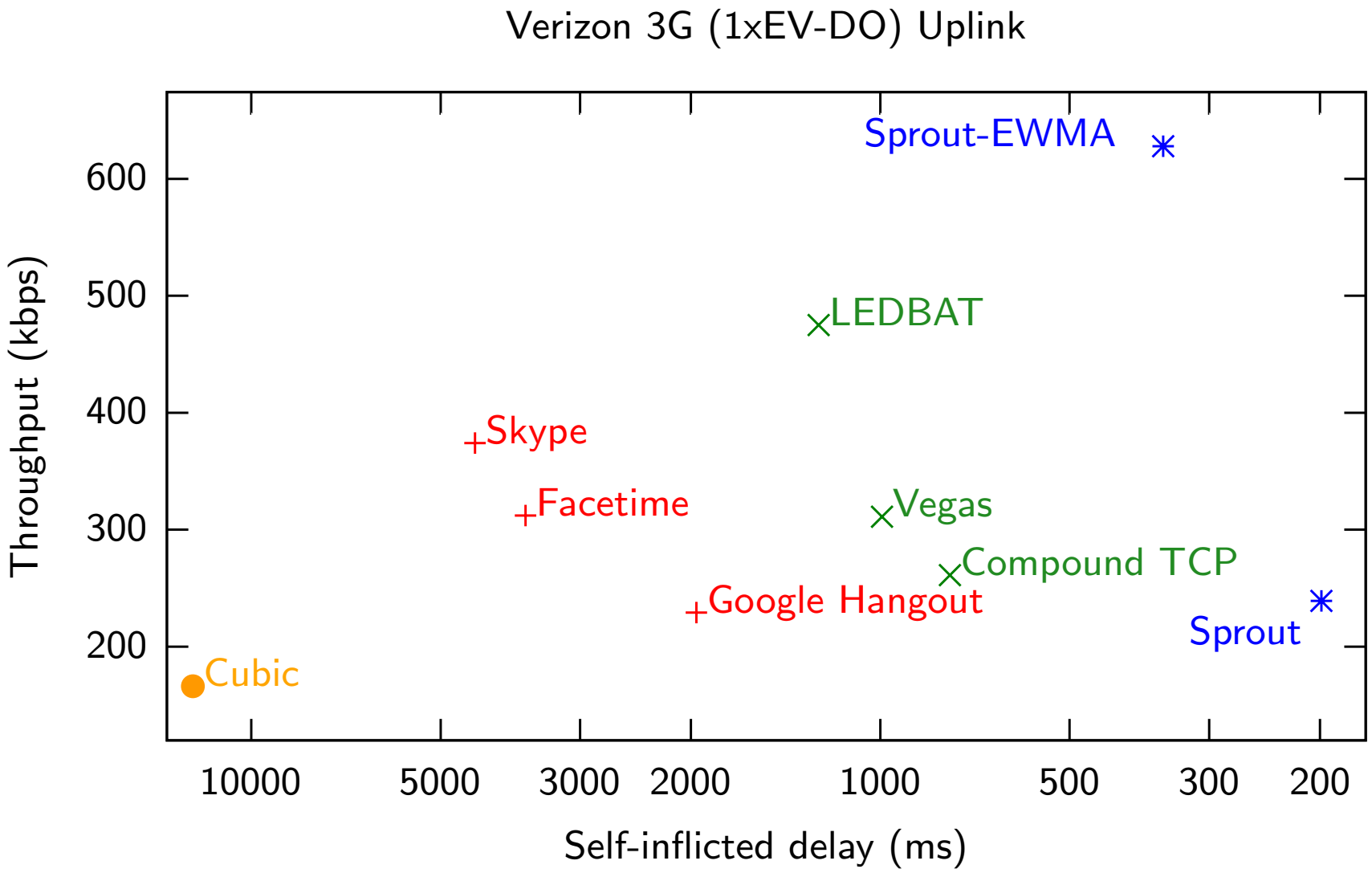
Thank you

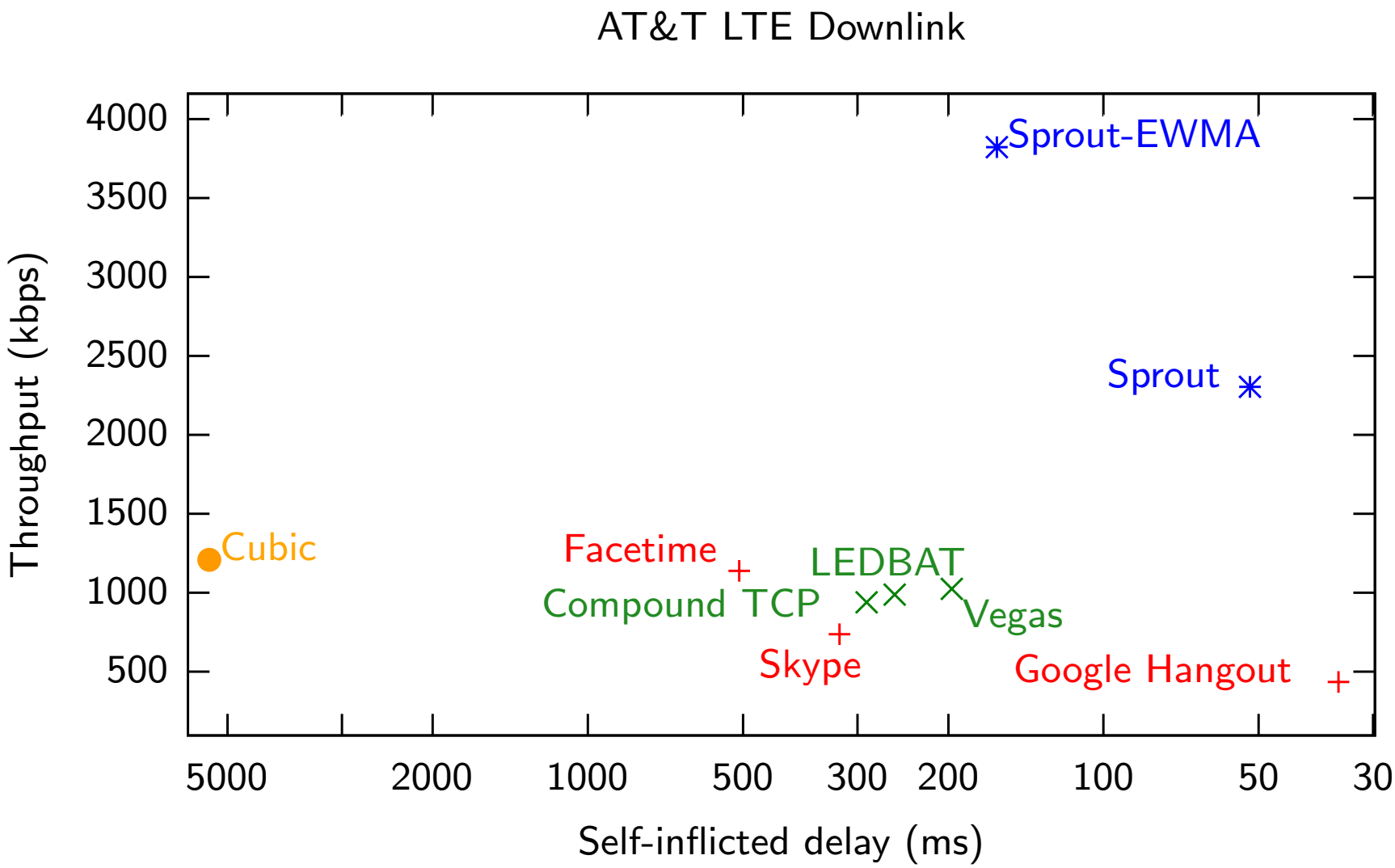
- Lakshminarayanan Subramanian
- Shuo Deng
- Jonathan Perry
- Katrina LaCurts
- Andrew McGregor
- Tim Shepard
- Dave Täht
- Michael Welzl
- Hannes Tschofenig
- Wireless@MIT members (http://wireless.csail.mit.edu)
- NSF & Shannon family (fellowship)

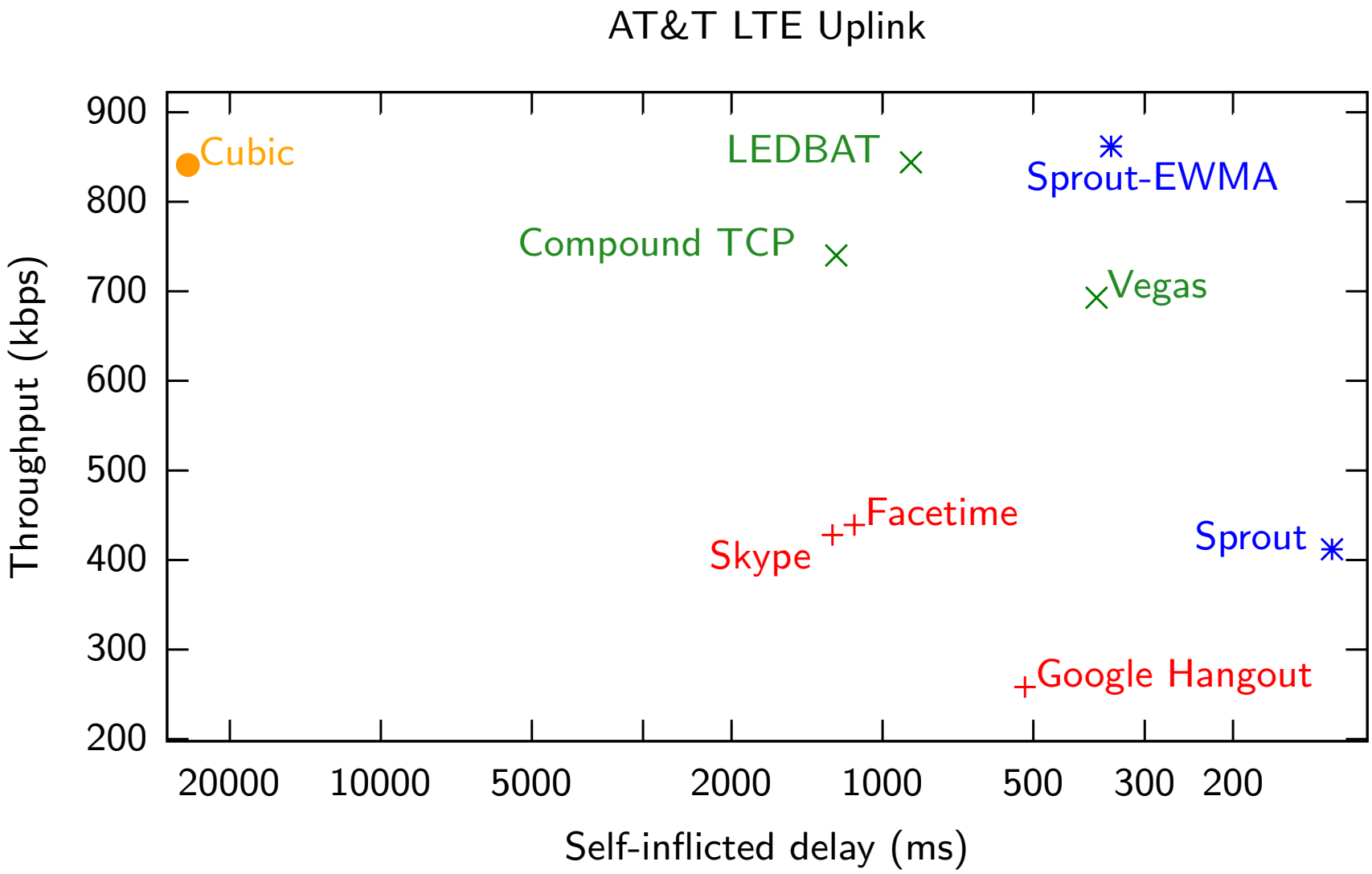
Sprout for controlled delay over cellular networks

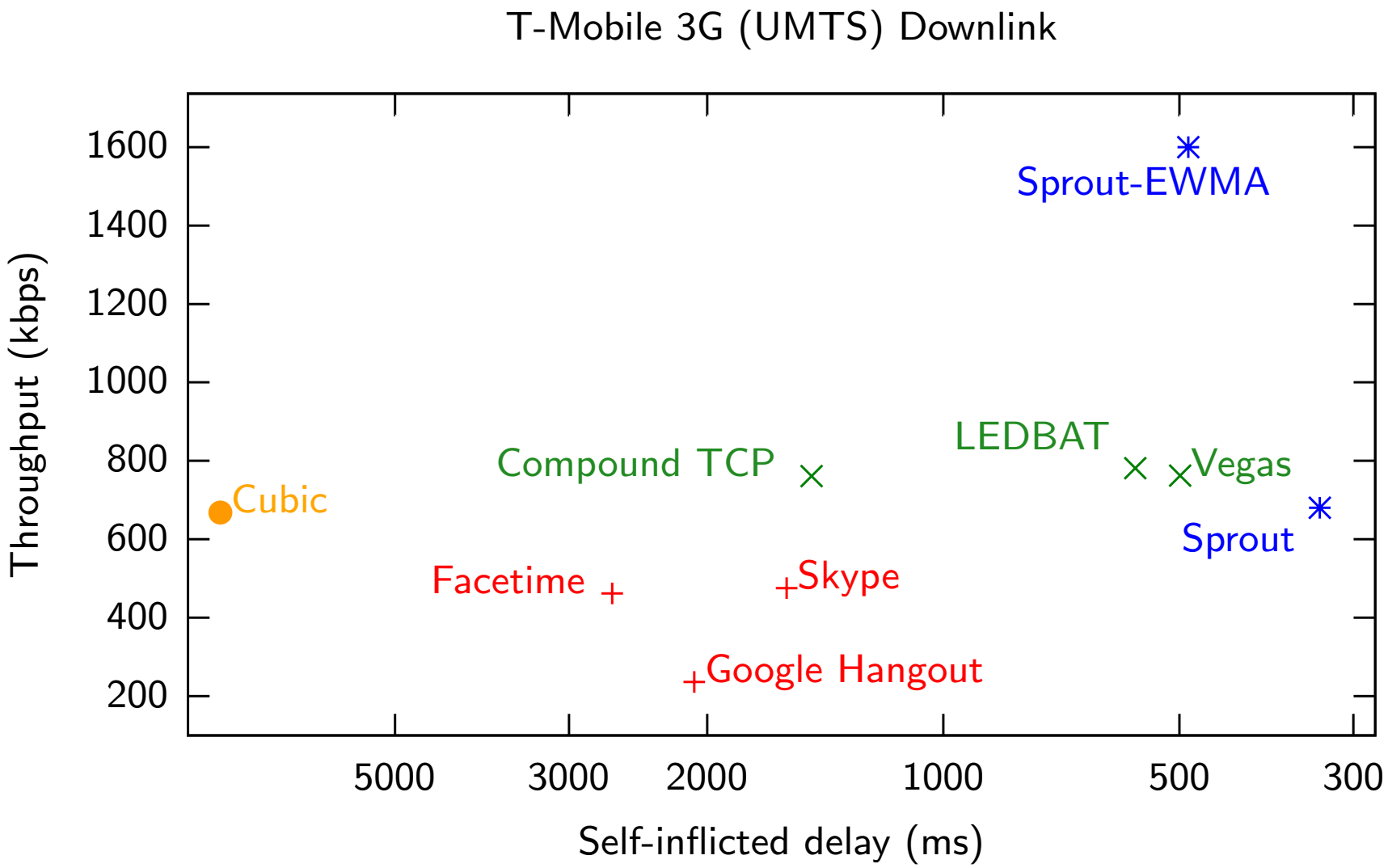
- **Model** varying link speed
- **Infer** current link speed
- **Predict** future link speed
- **Control** risk of large delay with cautious forecast
- Yields 2–4× throughput of Skype, Facetime, Hangout
- Achieves 7–9× reduction in self-inflicted delay
- Matches some active queue management purely end-to-end
- Code and directions at `http://alfalfa.mit.edu`
- `alfalfa@mit.edu`

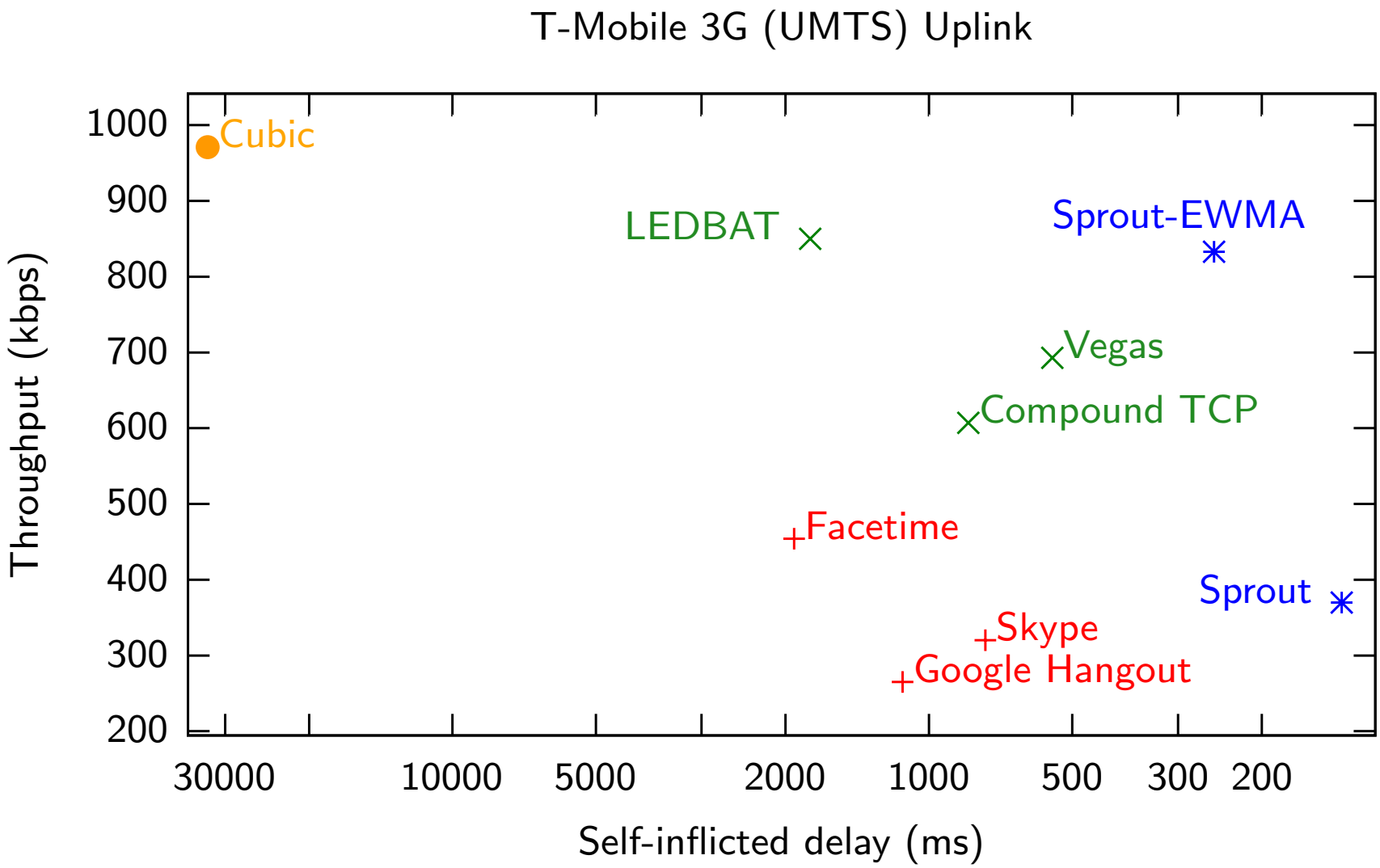

Verizon 3G (1xEV-DO) Downlink
Throughput (kbps)
550
500
450
400
350
300
250
200
150
Cubic
Sprout-EWMA
Compound TCP
LEDBAT
Vegas
Sprout
Facetime
Google Hangout
Skype
100000
50000
20000
10000
5000
3000
2000
Self-inflicted delay (ms)


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


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


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
5000
3000
2000
1000
500
300
Self-inflicted delay (ms)
Sprout-EWMA
LEDBAT
Compound TCP
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)