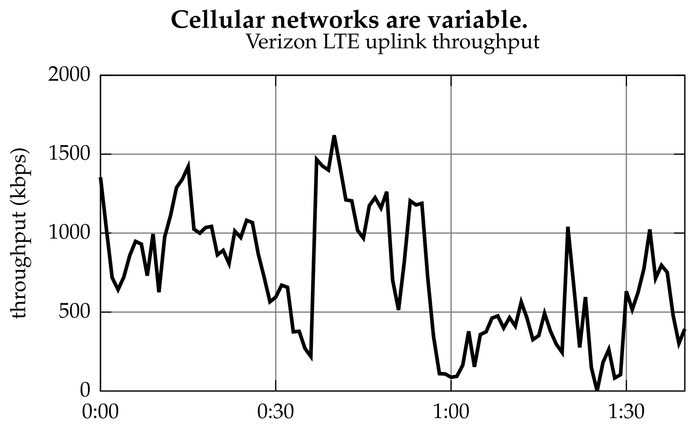
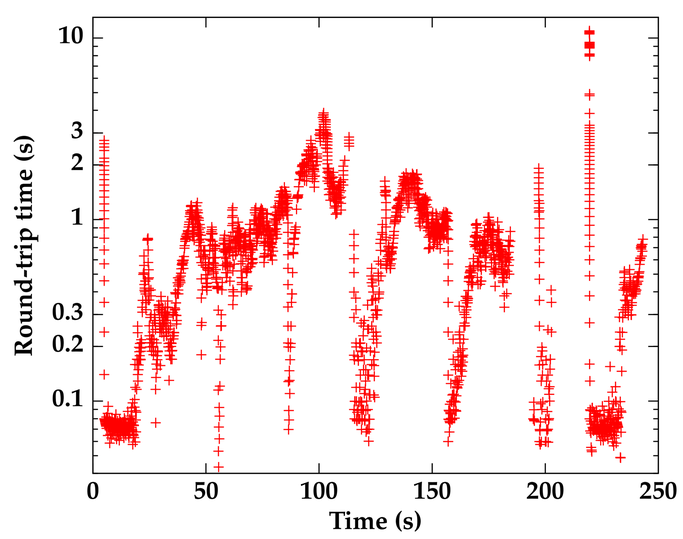
Sprout: Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks

Keith Winstein, Anirudh Sivaraman, and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, Mass.
{keithw,anirudh,hari}@mit.edu

Problem setting





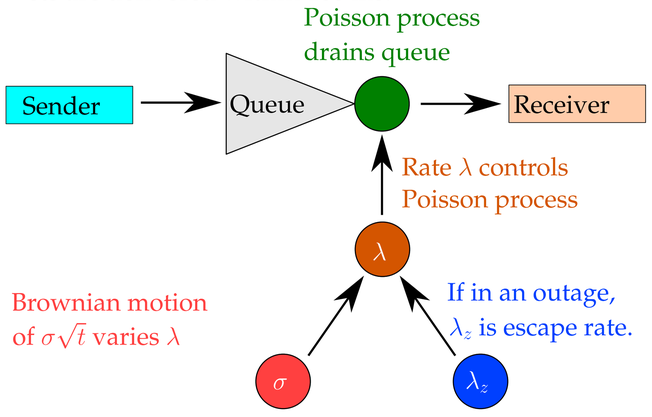
Interactive apps perform poorly because they are reactive.

Sprout's goal

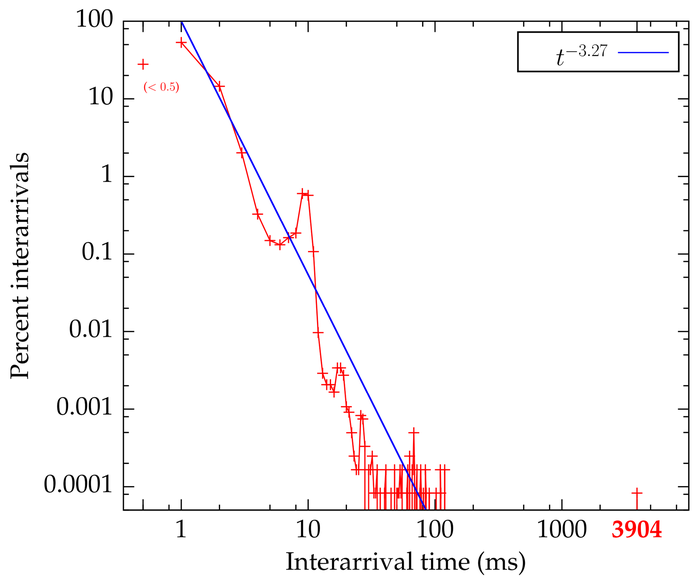
Maximize throughput subject to a bounded risk of delay exceeding 100 ms.

How it works

- **Inference: Infer probability distribution of link speeds from packet arrivals.**
- **Prediction: Predict future link speed.**
- **Control: Send as much as possible, but ensuring a 95 % chance that all packets are delivered within 100 ms.**



Inference



At the receiver:

- **Maintain a probability distribution over link rates.**
- **Model receiver's arrivals as a Poisson process with a time-varying rate λ.**
- **Observe number of bytes k that arrived in a tick of duration τ.**
- **Compute the likelihood $L(x)$ that a Poisson process with rate λ generated k bytes.**

$$L(x) = \frac{(x.\tau)^k}{k!} exp(-x.\tau) \quad (1)$$

- **Update current probability distribution using $L(x)$**

$$F(x) = Pr_{old}(\lambda = x) * L(x) \quad (2)$$

- **Normalize the new probabilities $F(x)$, so that they sum to unity.**

Prediction

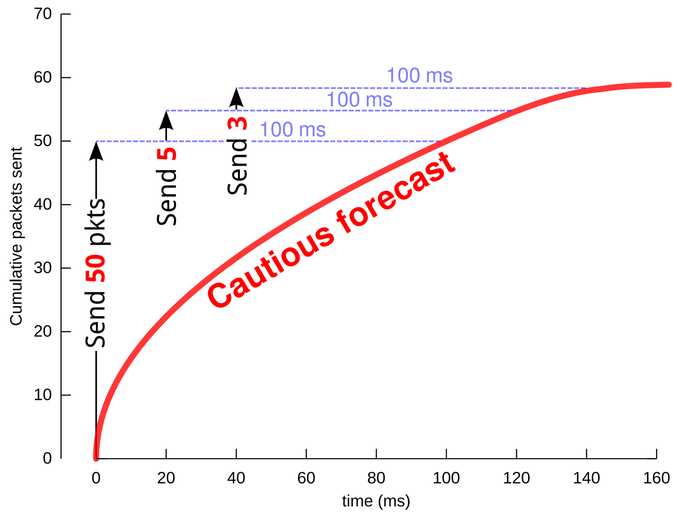
At the receiver:

- **Model evolution of link rates as Brownian motion with constant volatility σ.**
- **Explicitly model outages, using outage escape rate λ_z.**
- **Cautious forecast: Find the 5th percentile of the cumulative number of packet deliveries for a certain number of ticks in the future.**
- **Send back forecast to sender.**
- **Almost all steps are precalculated.**

Control

At the sender:

- **Use the forecast to determine the cumulative number of packets delivered 100 ms into the future.**
- **Subtract number of packets already in queue.**
- **Remaining number is safe to send.**



Evaluation

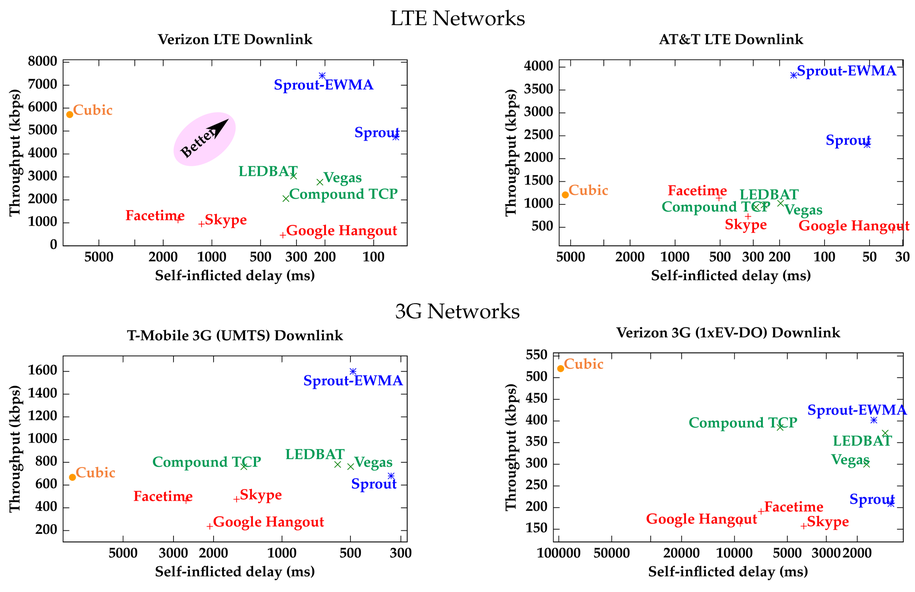
- **Saturate a cellular link in both the uplink and downlink directions.**
- **Playback trace in a trace-driven link emulator.**
- **Measure total throughput and end-to-end delay for reconstructing 95 % of the signal.**
- **Also evaluated Sprout-EWMA, a simplified variant of Sprout:**
  - **At the receiver, estimate link rate using a moving average filter.**
  - **Send link rate estimate to the sender, with no forecast.**
- **All source code was frozen before data collection.**

Sprout parameters used in evaluation

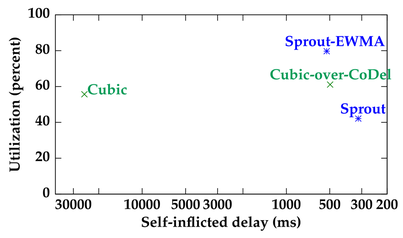
Volatility σ: fixed @	200 $\frac{\text{pkts}/s}{\sqrt{s}}$
Expected outage time $1/\lambda_z$:	1 s
Tick length:	20 ms
Forecast length:	160 ms
Delay target:	100 ms
Risk tolerance:	5%

Results

Sprout outperforms other end-to-end protocols



Sprout competes with AQM even though it is end-to-end



Sprout, used as a tunnel, successfully isolates competing traffic

	Direct	via Sprout	Benefit
Cubic throughput	8336 kbps	3776 kbps	0.5x (= worse)
Skype throughput	78 kbps	490 kbps	6x
Skype 95% delay	6.0 s	0.17 s	35x