Transport Architectures for an Evolving Internet

Keith Winstein

MIT Computer Science and Artificial Intelligence Laboratory

March 5, 2014

Joint work with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan.
The Internet evolves

In 20 years, computer networks have seen dramatic change:

- Wi-Fi
- Cellular networks
- Datacenters
- 10 GigE
- Transoceanic links
- Ubiquitous mobility
- Huge amounts of streaming video
Coping with change

How should users deal with an evolving network?

One approach: design new protocols.
The march of congestion-control protocols

- DECbit
- Tahoe
- CARD
- DUAL

1980s

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
The march of congestion-control protocols

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
The march of congestion-control protocols

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
The march of congestion-control protocols
The march of congestion-control protocols

1980s
- DUAL
- CARD
- DECbit
- Tahoe

1990s
- Reno
- NewReno
- SACK
- Westwood
- FAST
- Skype
- GentleAggression
- Vegas
- Eifel
- Veno
- Compound
- LEBAT

2000s
- BIC
- H-TCP
- Cubic
- PRR

2010s
- EBCC
- Binomial
- Sprout
- Remy

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
The march of congestion-control protocols

End-to-end

DUAL
CARD
DECbit
Tahoe
Reno
NewReno
Vegas
Eifel
Veno
Compound
LEDBAT
Skype
FAST
GentleAggression

In-net

EBCC
Binomial
SACK
Westwood
H-TCP
Cubic
PRR

1980s 1990s 2000s 2010s

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
The march of congestion-control protocols

End-to-end
- DUAL
- CARD
- DECbit
- Tahoe
- Reno
- NewReno
- BIC
- H-TCP
- Cubic
- PRR

In-net
- GPS
- WFQ

1980s 1990s 2000s 2010s

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
The march of congestion-control protocols

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
The march of congestion-control protocols

End-to-end
- DUAL
- CARD
- DECbit
- Tahoe
- Reno
- NewReno
- SACK
- Westwood
- FAST
- Skype
- GentleAggression
- EBCC
- Binomial
- Vegas
- Eifel
- Veno
- Compound
- LEDBAT
- H-TCP
- Cubic
- PRR

In-net
- GPS
- RED
- ECN
- CHOKe
- XCP
- RCP
- WFQ
- BLUE
- AVQ
- VCP

1980s 1990s 2000s 2010s
The march of congestion-control protocols
Declarative design

Systems with a model and a mission.
Explicitness in systems design

**Model**: explicit statement of assumptions about the problem

**Mission**: objective that the application wants

Explicit design considerations $\rightarrow$ **freedom to make changes**
**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

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
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.)

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

*Transport Architectures for an Evolving Internet*
**Model**: average rate looks like random walk
Sprout’s model

A Poisson process drains the queue at a rate of \( \lambda \). Brownian motion of \( \sigma \sqrt{t} \) varies \( \lambda \). If in an outage, \( \lambda_z \) is the escape rate.
## Sprout’s model parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volatility $\sigma$: fixed @</td>
<td>$200 \frac{\text{pkts/s}}{\sqrt{s}}$</td>
</tr>
<tr>
<td>Expected outage time $1/\lambda_z$:</td>
<td>1 s</td>
</tr>
<tr>
<td>Tick length ($\tau$):</td>
<td>20 ms</td>
</tr>
<tr>
<td>Forecast length:</td>
<td>160 ms</td>
</tr>
<tr>
<td>Delay target:</td>
<td>100 ms</td>
</tr>
<tr>
<td>Risk tolerance:</td>
<td>5%</td>
</tr>
</tbody>
</table>

All source code was **frozen before data collection began.**
**Infer**: current link speed

- **Observe** packets received every $\tau$
- **Update** $P(\lambda)$
**Predict**: future link speed

- **Evolve** model forward
- **Forecast** 5th percentile cumulative packets
**Control**: fill up 100 ms forecast window

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
**Control**: fill up 100 ms forecast window
**Control**: fill up 100 ms forecast window

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
**Control**: fill up 100 ms forecast window
**Control**: fill up 100 ms forecast window

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
**Control**: fill up 100 ms forecast window

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

*Transport Architectures for an Evolving Internet*
**Control:** fill up 100 ms forecast window

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Sprout’s results
Introduction

Sprout Remy

Throughput (kbps)

Self-inflicted delay (ms)

Verizon LTE Downlink

Facetime

Skype

Google Hangout

Better

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Verizon LTE Downlink

Throughput (kbps) vs. Self-inflicted delay (ms)

- Better
- LEDBAT
- Vegas
- Compound TCP
- Facetime
- Skype
- Google Hangout

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Verizon LTE Downlink

Throughput (kbps) vs. Self-inflicted delay (ms)

- Sprout-EWMA
- Sprout
- LEDBAT
- Vegas
- Compound TCP
- Better
- Cubic
- Facetime
- Skype
- Google Hangout

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Throughput (kbps) vs. Self-inflicted delay (ms) for various TCP variants on Verizon LTE Uplink.

- Sprout-EWMA
- Sprout
- LEDBAT
- Vegas
- Compound TCP
- Cubic
- Skype
- Facetime
- Google Hangout

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Overall results on 8 links

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

<table>
<thead>
<tr>
<th>Sprout vs.</th>
<th>Avg. speedup</th>
<th>Delay reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skype</td>
<td>2.2×</td>
<td>7.9×</td>
</tr>
<tr>
<td>Hangout</td>
<td>4.4×</td>
<td>7.2×</td>
</tr>
<tr>
<td>Facetime</td>
<td>1.9×</td>
<td>8.7×</td>
</tr>
<tr>
<td>Compound</td>
<td>1.3×</td>
<td>4.8×</td>
</tr>
<tr>
<td>TCP Vegas</td>
<td>1.1×</td>
<td>2.1×</td>
</tr>
<tr>
<td>LEDBAT</td>
<td>Same</td>
<td>2.8×</td>
</tr>
<tr>
<td>Cubic</td>
<td>0.91×</td>
<td>79×</td>
</tr>
</tbody>
</table>

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
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:

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
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*
Introduction Sprout Remy

Baseline

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Land of 3,000 student protocols

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Sprout was on the frontier

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
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
Introduction Sprout Remy

Sprout’s mark

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Sprout’s mark

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Now that we have 40+ algorithms...

- Sprout for cellular networks?
- Wireless-TCP for Wi-Fi?
- High-BDP-TCP for transoceanic links?
- Datacenter-TCP for datacenters?
- CoDel for cable modems?
- **TBA-TCP for tomorrow’s networks?**
Rational choice of scheme is challenging

- Different missions?
- Different assumptions about network?
- One scheme just plain better?
Networks constrained by a fuzzy idea of TCP’s assumptions

- Mask stochastic loss
- Bufferbloat
- Mask out-of-order delivery
- No parallel/multipath routing

*Advice for Internet Subnetwork Designers* (RFC 3819) is 21,000 words!
Apps hack around TCP

- Open lots of flows
- Goose slow start
- Add pacing
- Give up and do it yourself
Apps hack around TCP

- Open lots of flows
- Goose slow start
- Add pacing
- Give up and do it yourself
Apps hack around TCP

- Open lots of flows
- Goose slow start
- Add pacing
- Give up and do it yourself
Apps hack around TCP

- Open lots of flows
- Goose slow start
- Add pacing
- Give up and do it yourself
Apps hack around TCP

- Open lots of flows
- Goose slow start
- Add pacing
- Give up and do it yourself

Chrome (QUIC)
BitTorrent ($\mu$TP)
Mosh (SSP)
IBM Aspera (fasp)
**Idea: computer-generated protocols**

Transport layer should adapt to *whatever*:

- network does
- application wants
Idea: computer-generated protocols

Transport layer should adapt to *whatever*:

- network does *(model)*
- application wants *(mission)*
What we built

**Remy**: a program that generates congestion-control schemes offline

**Input:**
- Assumptions about network and workload (*model*)
- Application’s objective (*mission*)

**Output**: CC algorithm for a TCP sender (*RemyCC*)

**Time**: hours to days

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
The basic question of congestion control

At this moment, do I:

▶ send a packet
▶ not send a packet?
Missions of congestion control

Maximize

\[ \sum_i \log [\text{throughput}_i] \quad \text{(proportionally fair throughput)} \]
Missions of congestion control

Maximize

- $\sum_i \log [\text{throughput}_i]$ (proportionally fair throughput)
- $\sum_i \log \left[ \frac{\text{throughput}_i}{\text{delay}_i} \right]$ (proportionally fair throughput/delay)

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Missions of congestion control

Maximize

\[ \sum_i \log[\text{throughput}_i] \] (proportionally fair throughput)

\[ \sum_i \log \left[ \frac{\text{throughput}_i}{(\text{delay}_i)^\delta} \right] \] (proportionally fair throughput/delay)
Missions of congestion control

Maximize

- \( \sum_i \log [\text{throughput}_i] \) (proportionally fair throughput)
- \( \sum_i \log \left[ \frac{\text{throughput}_i}{(\text{delay}_i)^\delta} \right] \) (proportionally fair throughput/delay)
- \( \min_i \text{throughput}_i \) (max-min throughput)

Minimize

- mean flow completion time
- page load time

Prevent

- pathological behavior
- congestion collapse
Encoding the designer’s prior assumptions

- **Model** of network uncertainty
  - Link speed distribution
  - Delay distribution
  - Topology distribution

- **Model** of workload
  - Web browsing
  - MapReduce
  - Videoconferencing
  - Streaming video (YouTube/Netflix)
Dumbbell network

Sender

Queue

Link

Receiver

Round-trip time

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Dumbbell network

Sender

Queue

Receiver

Sender 2

Link

Round-trip time

Receiver 2

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Dumbbell network

Sender
Sender 2
Sender n
Queue
Link
Receiver
Receiver 2
Receiver n

Round-trip time

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Superrational congestion control

At this moment, *do I:

- send a packet
- not send a packet?
Superrational congestion control

At this moment,* do I:

- send a packet
- not send a packet?

* Assuming every node is running the same algorithm.
Remy: tractable search for best policy

- Best decision given all history: not tractable
- Instead, summarize the history
A RemyCC tracks four congestion signals

\( r_{\text{ewma}_\alpha} \): short-term moving average of interval between acks
  “How fast are packets arriving (now)?”

\( r_{\text{ewma}_\beta} \): long-term moving average of same
  “How fast are packets arriving (smoothed)?”

\( s_{\text{ewma}} \): moving average of interval between acked timestamps
  “How fast was I sending?”

\( rtt\_ratio \): ratio of last RTT to smallest RTT so far
  “How long is the queue?”
Why these four features?

- We can measure the benefit of each!
- Removing any one hurts
  - losing $r_{\text{ewma}_\alpha}$ hurts the most
- More signals increase search time
- ...but others might help on some networks
A RemyCC maps each state to an action

$$\text{REMYCC}(r_{\text{ewma}}_{\alpha\beta}, s_{\text{ewma}}, rtt_{\text{ratio}}) \rightarrow \langle m, b, \tau \rangle$$

- $m$ Multiple to congestion window
- $b$ Increment to congestion window
- $\tau$ Minimum interval between two outgoing packets
Runtime for a RemyCC

On ack:

- \( \langle m, b, \tau \rangle \leftarrow \text{REMYCC}(r_{\text{ewma}}_{\alpha\beta}, s_{\text{ewma}}, \text{rtt\_ratio}) \)
- \( \text{cwnd} \leftarrow m \cdot \text{cwnd} + b \)

Send packet if:

- \( \text{cwnd} > \text{FlightSize}, \text{and} \)
- \( \text{last packet sent} > \tau \text{ ago} \)
Remy’s job

Find piecewise-continuous $\text{REMYCC}()$ that optimizes expected value of objective function
Remy example: 2D state space

On ack:

\[ \langle m, b, \tau \rangle \leftarrow \text{REMYCC}(s_{\text{ewma}}, r_{\text{ewma}}_\alpha, r_{\text{ewma}}_\beta, \text{rtt\_ratio}) \]
Remy example: 2D state space

On ack:

\[ \langle m, b, \tau \rangle \leftarrow \text{REMYCC}(s_{ewma}, r_{ewma_{\alpha}}, \text{rtt ratio}) \]
### Remy example: model

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Distribution</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link speed</td>
<td>Uniform(10, 20)</td>
<td>Mbps</td>
</tr>
<tr>
<td>RTT</td>
<td>Uniform(100, 200)</td>
<td>ms</td>
</tr>
<tr>
<td>$n$</td>
<td>Uniform(1, 16)</td>
<td></td>
</tr>
<tr>
<td>“On” process</td>
<td>$\text{exp}[\mu = 5]$</td>
<td>seconds</td>
</tr>
<tr>
<td>“Off” process</td>
<td>same</td>
<td></td>
</tr>
</tbody>
</table>

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Remy example: \textbf{mission}

\[
\sum_i \log \left[ \frac{\text{throughput}_i}{\text{delay}_i} \right]
\]
One action for all states. Find the best value.
The best (single) action. Now split it on median.

\[ <0.90, 4, 3.3> \]
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize each of the new actions

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Now split the most-used rule
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Simulate
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Split

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate
Optimize
Split

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Split

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Optimize

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Simulate

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Introduction

Sprout Remy

RemyCC

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC
RemyCC

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Evaluation in ns-2

- End-to-end comparators: NewReno, Cubic, Compound, Vegas
- In-net comparators: Cubic-over-sfqCoDel, XCP
- Simulation setup published for replication

TCP ex Machina: Computer-Generated Congestion Control

Remy is a computer program that figures out how computers can best cooperate to share a network.

- Read the Paper
- Reproduce the Results
- Get the Code

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Scenario 1: fixed-rate network, homogenous senders
Scenario 1: details

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Simulation parameter</th>
<th>Remy assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link speed</td>
<td>15 Mbps</td>
<td>Uniform(10, 20) Mbps</td>
</tr>
<tr>
<td>RTT</td>
<td>150 ms</td>
<td>Uniform(100, 200) ms</td>
</tr>
<tr>
<td>n</td>
<td>8</td>
<td>Uniform(1, 16)</td>
</tr>
<tr>
<td>“On” process</td>
<td>exp[µ = 100] kB</td>
<td>exp[µ = 5] s</td>
</tr>
<tr>
<td>“Off” process</td>
<td>exp[µ = 1/2] s</td>
<td>exp[µ = 5] s</td>
</tr>
</tbody>
</table>

Remy objective:

$$\sum_i \log \left( \frac{\text{throughput}_i}{(\text{delay}_i)^\delta} \right)$$

$$\delta \in \{\frac{1}{10}, 1, 10\}$$
Scenario 1: throughput-delay plot

Throughput (Mbps) vs Queueing delay (ms)
Scenario 1: throughput-delay plot

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Scenario 1: throughput-delay plot

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Scenario 1: throughput-delay plot
Scenario 1: throughput-delay plot
**Scenario 1: throughput-delay plot**

Throughput variability

Delay variability

Median outcome

Better
Scenario 1: throughput-delay plot
Scenario 1: throughput-delay plot
Scenario 1: throughput-delay plot

Throughput (Mbps) vs. Queueing delay (ms)

- Better
- NewReno
- Vegas
- Cubic
- Compound

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Scenario 1: throughput-delay plot

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Scenario 1: throughput-delay plot
Scenario 1: throughput-delay plot

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
Scenario 2: Verizon LTE, $n = 8$

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Remy as an instrument to study network science

From the perspective of an endpoint, what does it help to know about the network?

How difficult is it to learn a good protocol, given an imperfect model of the network?
RemyCC competing against itself

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
**RemyCC competing against itself**

![Graph showing throughput and queueing delay for different protocols](image)

**Throughput (Mbps) vs. Queueing delay (ms)**

- NewReno
- RemyCC [TCP-naive]
- RemyCC competing against itself

**Cost of TCP-awareness**

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

*Transport Architectures for an Evolving Internet*
RemyCC competing against itself

Throughput (Mbps) vs. Queueing delay (ms) graph showing:
- NewReno
- RemyCC [TCP-aware]
- RemyCC [TCP-naive]

Cost of TCP-awareness

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC competing against TCP NewReno

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC competing against TCP NewReno

- Throughput (Mbps)
- Queueing delay (ms)

**Effect of TCP-aware adversary**

**Benefit of TCP-awareness**

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
RemyCC competing against TCP NewReno

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
The cost of generality

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
The cost of generality

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
The cost of generality

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
The cost of generality

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
The cost of generality

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
The cost of generality

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
The cost of generality
Systems ex Machina

- Explicit design considerations → freedom to make changes
- “If this system is the answer, what’s the question?”

**Sprout** 2–4× the throughput and 7–9× less delay than Skype, etc.

**Remy** computer-generated protocol design

keithw@mit.edu

http://mit.edu/keithw
When the model is wrong about the topology

One bottleneck

Flow 1

Flow 2

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
When the model is wrong about the topology

Two bottlenecks

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
When the model is wrong about the topology

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
When the model is wrong about the topology

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
When the model is wrong about the topology
When the model is wrong about the topology

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Verizon 3G (1xEV-DO) Downlink

- **Cubic**
- **Sprout-EWMA**
- **Compound TCP**
- **LEDBAT**
- **Vegas**
- **Sprout**
- **Google Hangout**
- **Facetime**
- **Skype**

Keith Winston (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

*Transport Architectures for an Evolving Internet*
Verizon 3G (1xEV-DO) Uplink

Throughput (kbps) vs. Self-inflicted delay (ms)

- Sprout-EWMA
- LEDBAT
- Skype
- Facetime
- Vegas
- Compound TCP
- Google Hangout
- Sprout
- Cubic

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
Sprout Remy

Throughput (kbps) vs. Self-inflicted delay (ms)

AT&T LTE Downlink

Throughput (kbps)

Sprout-EWMA

Sprout

Cubic

Facetime

Compound TCP

LEDBAT

Vegas

Skype

Google Hangout

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
AT&T LTE Uplink

Throughput (kbps) vs. Self-inflicted delay (ms)

- Cubic
- LEDBAT
- Compound TCP
- Sprout-EWMA
- Vegas
- Skype
- Facetime
- Sprout
- Google Hangout

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet
T-Mobile 3G (UMTS) Downlink

<table>
<thead>
<tr>
<th>Throughput (kbps)</th>
<th>Self-inflicted delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1600</td>
<td>3000</td>
</tr>
<tr>
<td>1400</td>
<td>3000</td>
</tr>
<tr>
<td>1200</td>
<td>3000</td>
</tr>
<tr>
<td>1000</td>
<td>3000</td>
</tr>
<tr>
<td>800</td>
<td>3000</td>
</tr>
<tr>
<td>600</td>
<td>3000</td>
</tr>
<tr>
<td>400</td>
<td>3000</td>
</tr>
<tr>
<td>200</td>
<td>3000</td>
</tr>
</tbody>
</table>

- Sprout-EWMA
- LEDBAT
- Vegas
- Sprout
- Compound TCP
- Facetime
- Skype
- Google Hangout

Keith Winstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)
Transport Architectures for an Evolving Internet
T-Mobile 3G (UMTS) Uplink

Throughput (kbps) vs. Self-inflicted delay (ms)

- Cubic
- LEDBAT
- Sprout-EWMA
- Vegas
- Compound TCP
- Facetime
- Skype
- Google Hangout
- Sprout

Keith Weinstein (with Anirudh Sivaraman, Pratiksha Thaker, and Hari Balakrishnan)

Transport Architectures for an Evolving Internet