Modern Application Layer Transmission Patterns from a Transport Perspective

Passive and Active Measurements Conference 2014

March 11, 2014

Matthew Sargent, Ethan Blanton, Mark Allman

CASE SCHOOL
OF ENGINEERING
CASE WESTERN RESERVE
UNIVERSITY

TCP Modeling - SIGCOMM '13

TCP ex Machina: Computer-Generated Congestion Control

Keith Winstein and Hari Balakrishnan
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, Mass.
{keithw, hari}@mit.edu

ABSTRACT

This paper describes a new approach to end-to-end congestion control on a multi-user network. Rather than manually formulate each endpoint's reaction to congestion signals, as in traditional protocols, we developed a program called Remy that generates congestion-control algorithms to run at the endpoints.

In this approach, the protocol designer specifies their prior knowledge or assumptions about the network and an objective that the algorithm will try to achieve, e.g., high throughput and low queueing delay. Remy then produces a distributed algorithm—the control rules for the independent endpoints—that tries to achieve this objective.

In simulations with ns-2, Remy-generated algorithms outperformed human-designed end-to-end techniques, including TCP Cubic, Compound, and Vegas. In many cases, Remy's algorithms also

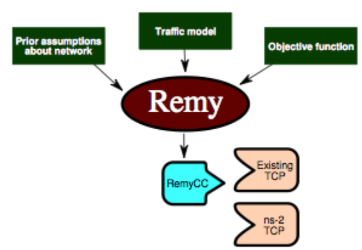
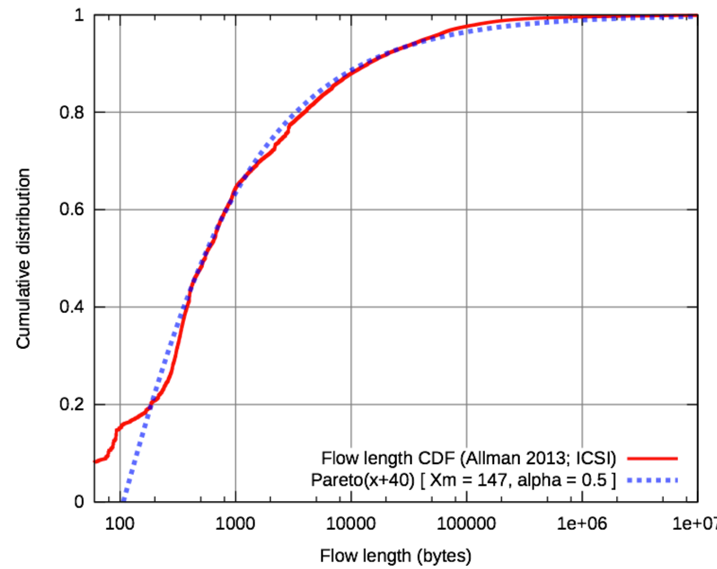


Figure 1: Remy designs congestion-control schemes automati-



TCP Modeling - Hotnets ‘13

How to Improve your Network Performance by Asking your Provider for Worse Service

Radhika Mittal
UC Berkeley
radhika@eecs.berkeley.edu

Justine Sherry
UC Berkeley
justine@eecs.berkeley.edu

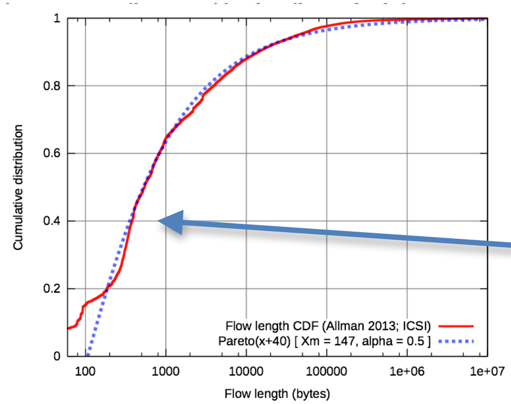
Sylvia Ratnasamy
UC Berkeley
sylvia@eecs.berkeley.edu

Scott Shenker
UC Berkeley and ICSI
shenker@icsi.berkeley.edu

ABSTRACT

TCP’s congestion control is deliberately “cautious”, avoiding overloads by starting with a small initial window and then iteratively ramping up. As a result, it often takes flows several round-trip times to fully utilize the available bandwidth. In this paper we propose using several levels of lower priority service and a modified TCP behavior to achieve significantly improved flow completion times while preserving fairness.

than aggression, is the better choice for a uniform behavior because … networl … could e … sistentl … Thes … congest … act cou … flows w … the caut …



“The traffic matrix is generated as follows: each end host generates flows with Poisson arrivals. Flow sizes are drawn from an empirical traffic distribution [3].”

Goals

- Create a TCP connection classification scheme
- Empirically inform our mental model of TCP usage

Data

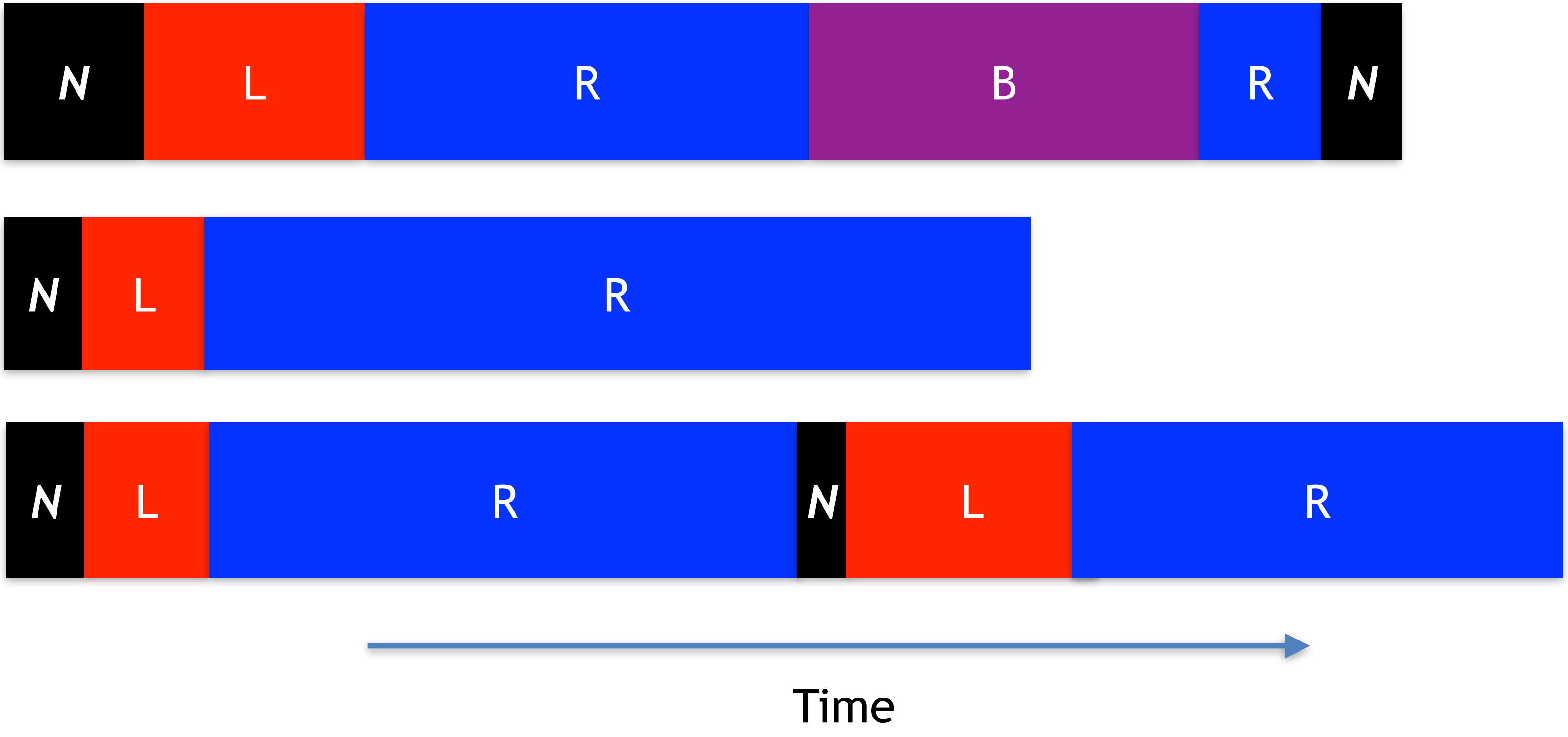
	CCZ Cleveland, OH	ICSI Berkeley, CA
Type	Residential	Research Lab
# Users	200-300	~100
# Months	15	7
Conns. Considered	3.9M	27.6M

(Additional details in the paper)

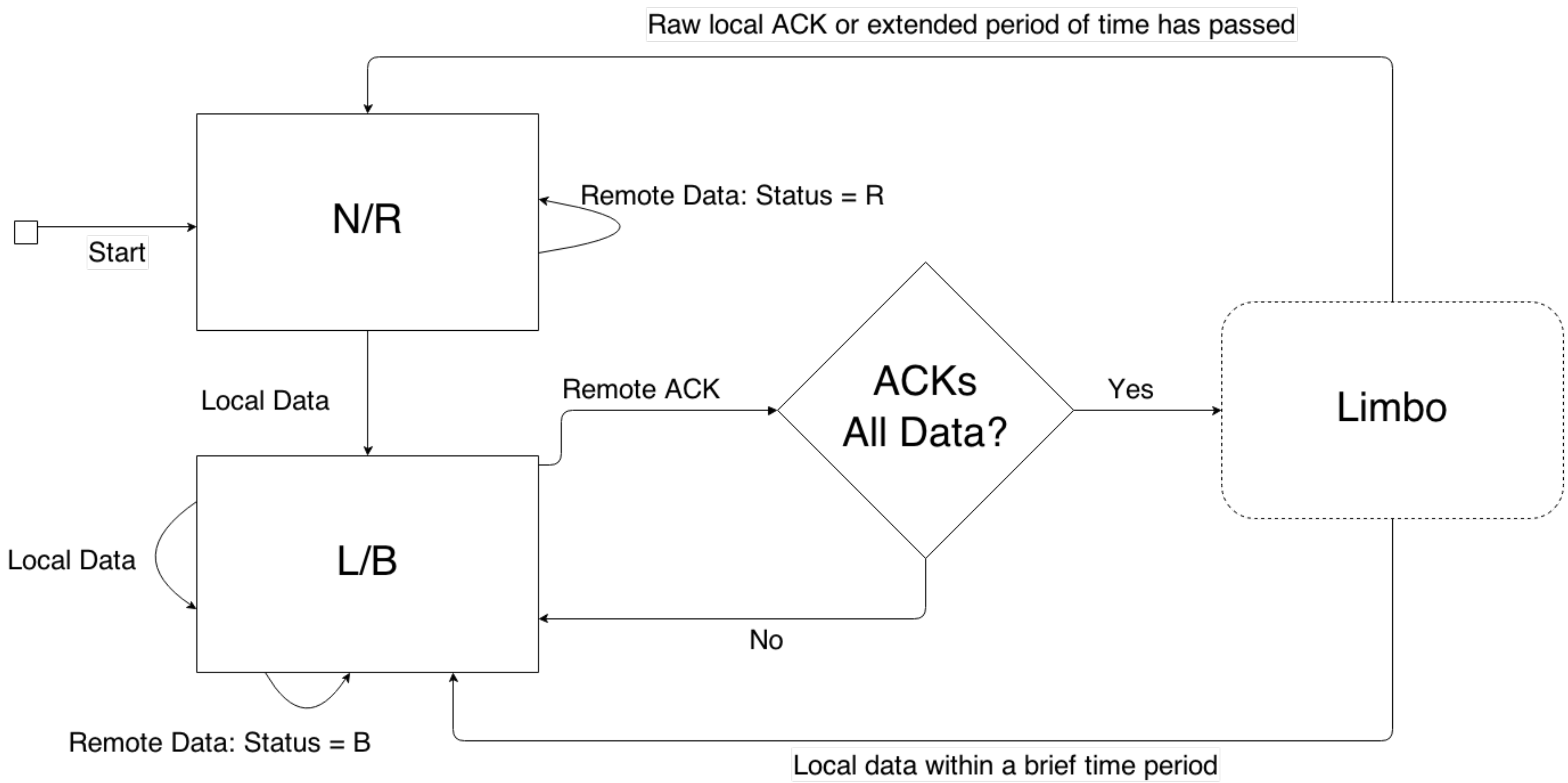
Methodology

- Split connections into 4 types of time periods:
  - ***L***ocal - Only local host transmits data
  - ***R***emote - Only remote host transmits data
  - ***B***oth - Both hosts transmit data
  - ***N***one - Neither host transmits data

Methodology



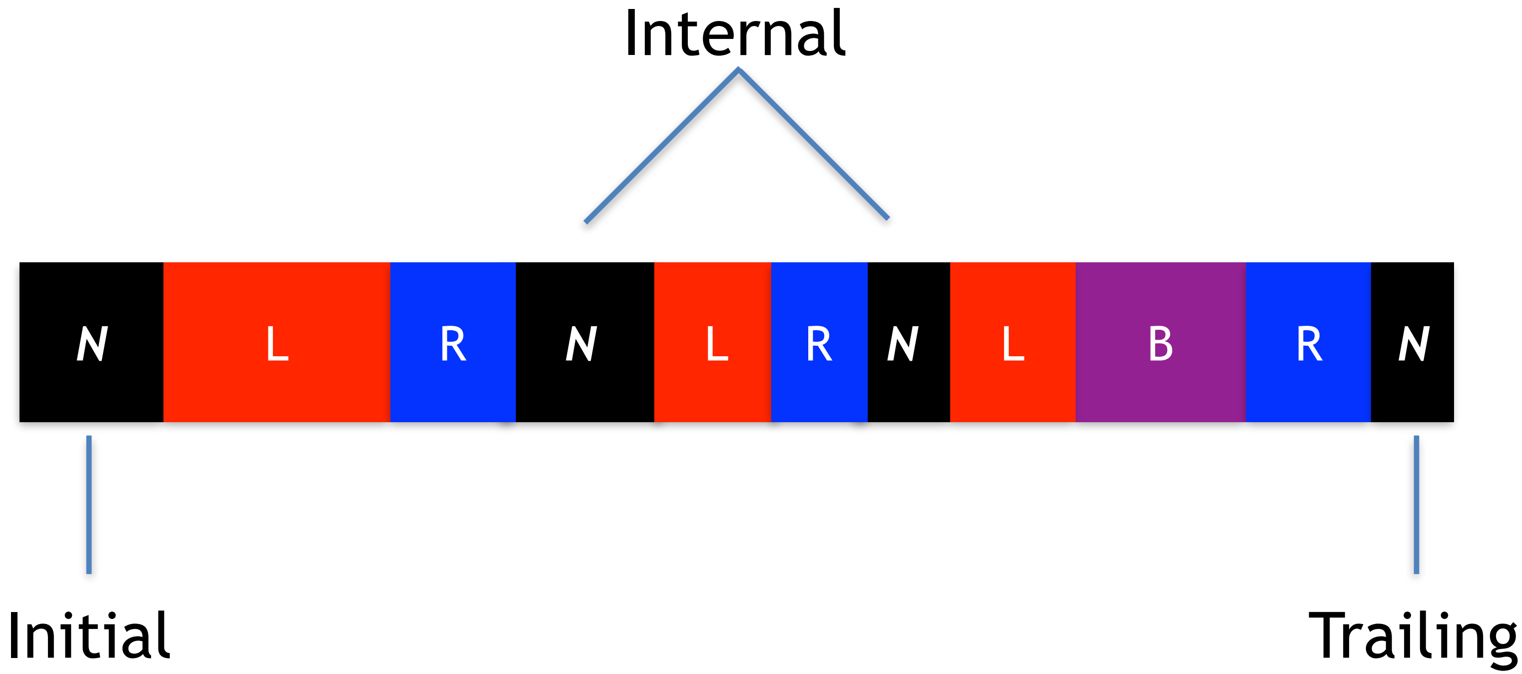
Methodology



(Additional details in the paper)

CASE SCHOOL
OF ENGINEERING
CASE WESTERN RESERVE
UNIVERSITY

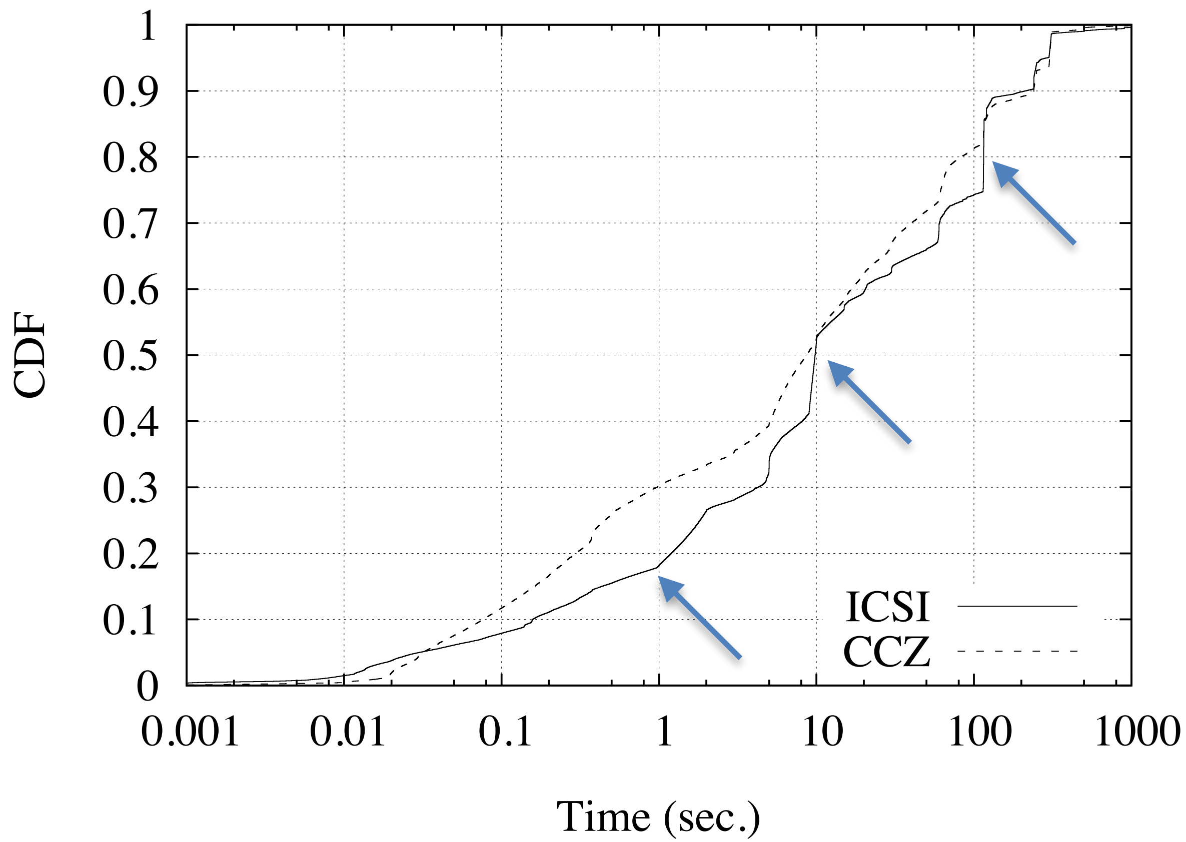
Types of Silence



Results - Silence in Connections

	CCZ	ICSI
No Silence/ Initial Only	31%	51%
Trailing Only	32%	21%
Internal Only	15%	18%
Both	22%	10%

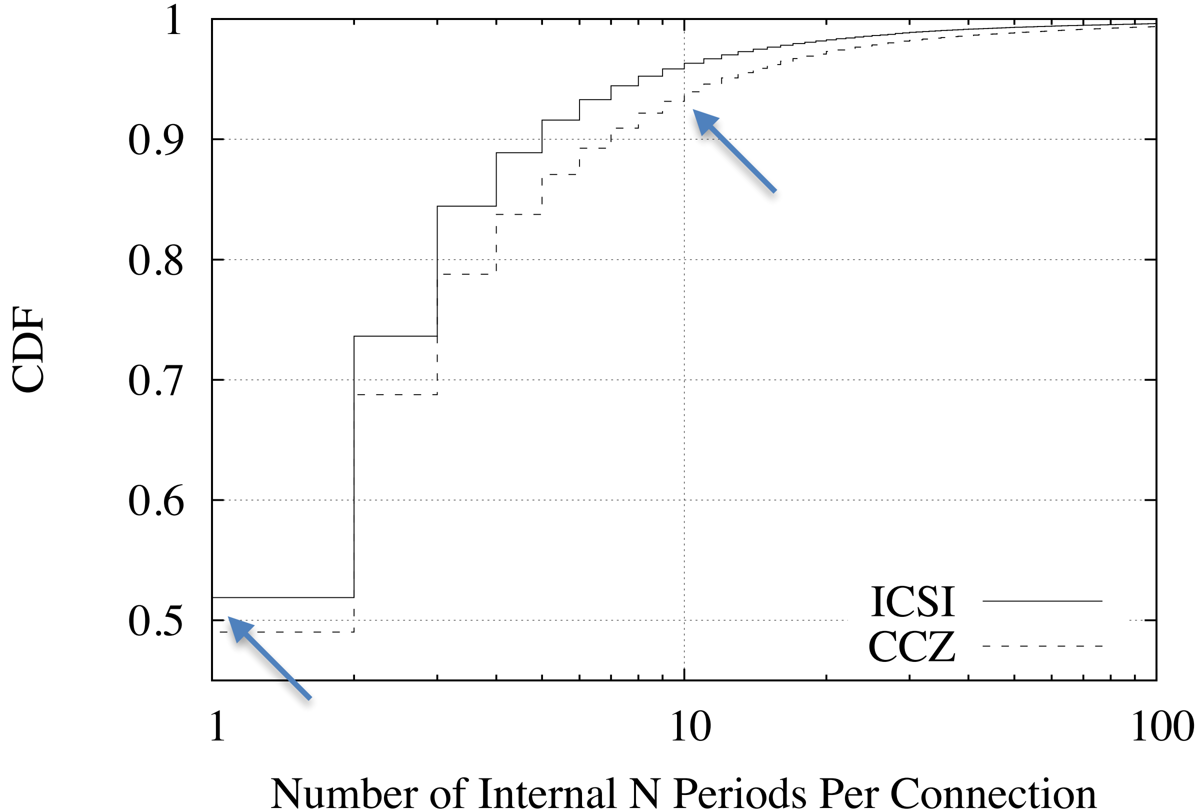
Results - Trailing Silence



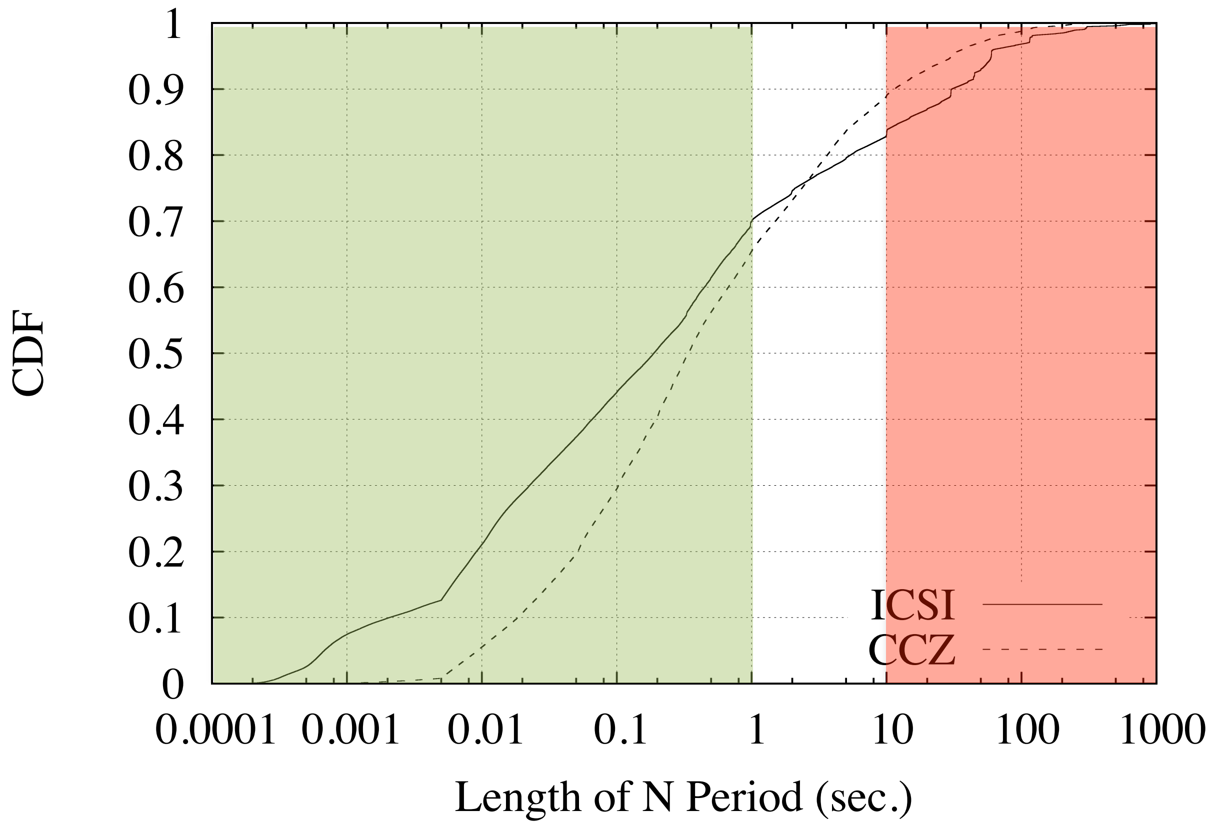
Results - Trailing Silence

- > 75% of connections with trailing silence are HTTP(S)
- Other heavy-hitter applications vary by location
  - CCZ: P2P
  - ICSI: SMTP, DNS

Results - Internal Silence



Results - Internal Silence



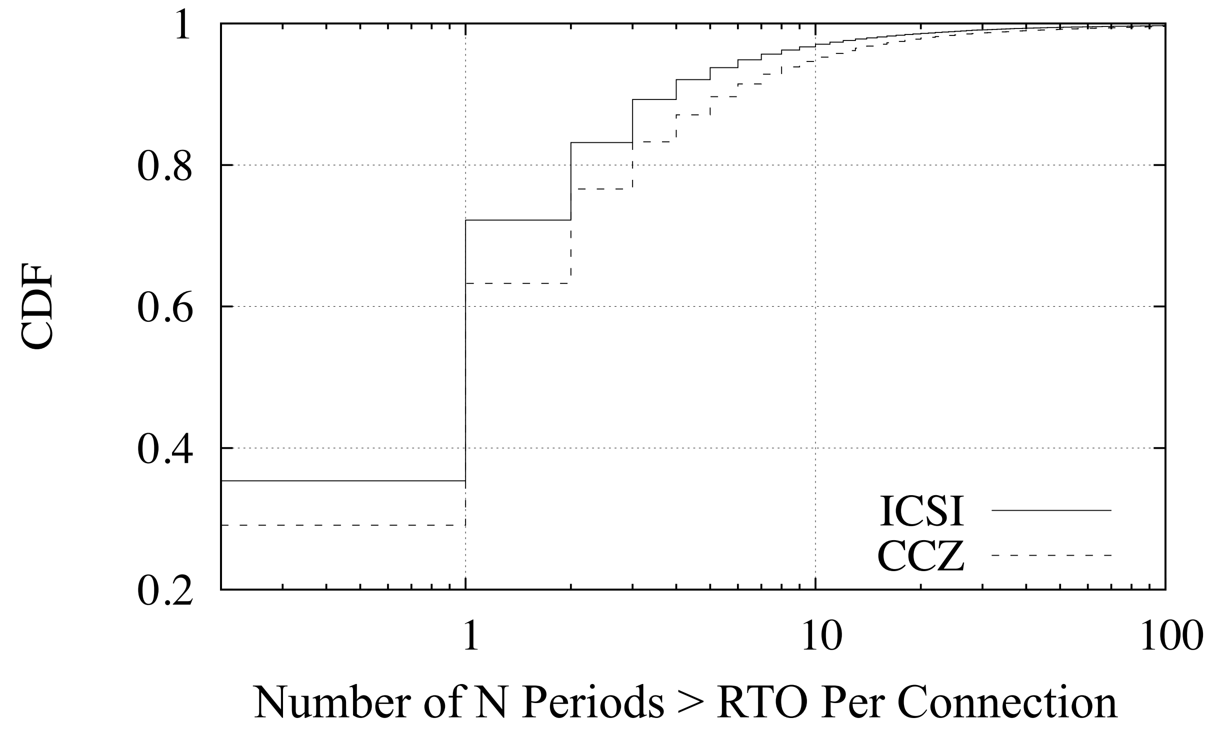
Results - Internal Silence

- > 65% of connections are HTTP(S).
- Individual heavy-hitter applications match the overall distribution for number of internal silence periods.

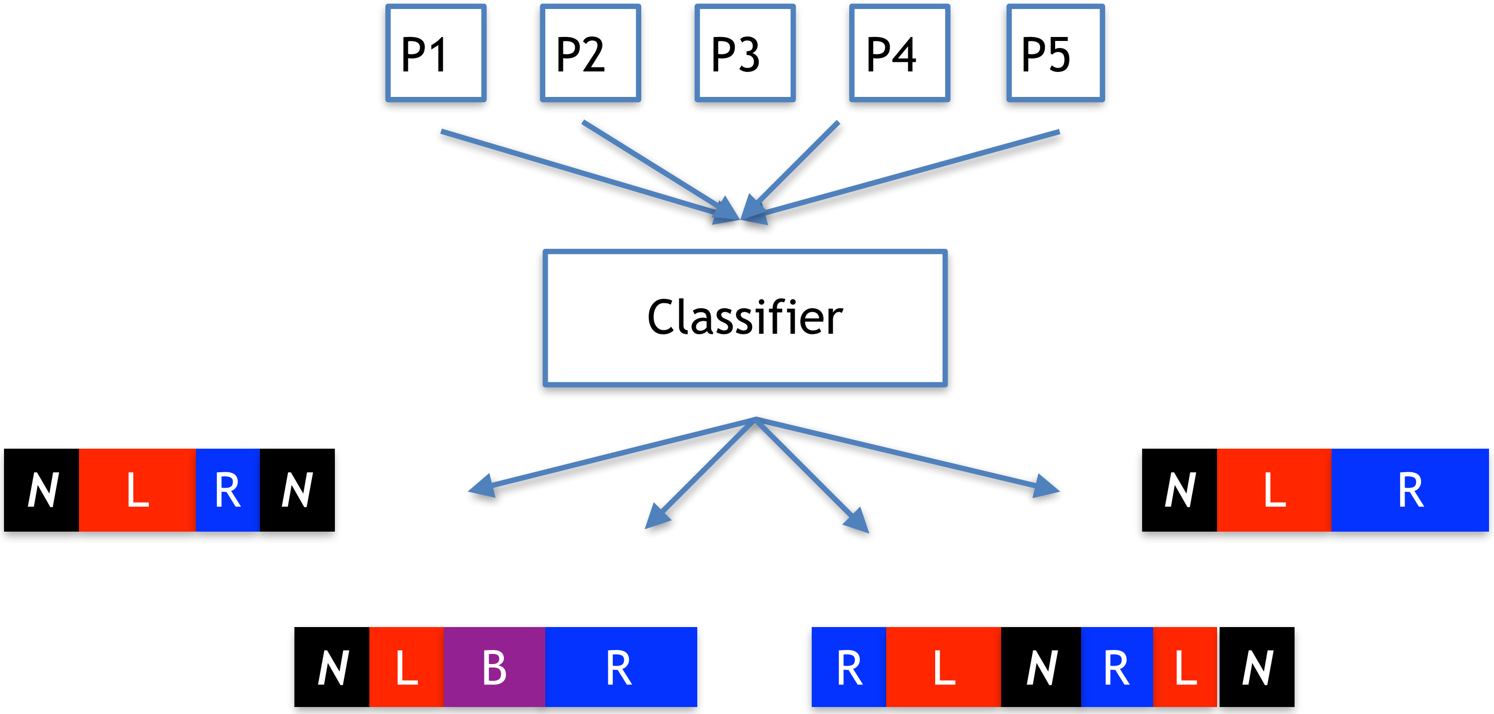
Results - Internal Silence

- Internal silence length varies across heavy-hitter applications.
  - Example:
    - HTTP(S) median ranges from 0.5 sec. - 3 sec.
    - SMTP has only internal silence < 3 sec.
- For both datasets, HTTPS has longer internal N periods than HTTP.

Results - Long Internal Silence

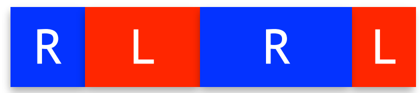


Results - Application Complexity

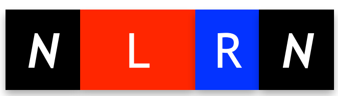


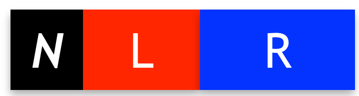
Results - Application Complexity

- Active:



- Simple:





- Complex:

Results - Application Complexity

Class	*Med. # Time Periods*	*Mean # Time Periods*	*Std. Dev.*	*% Conns.*
CCZ Active	*2*	*2.8*	*1.13*	*3%*
CCZ Simple	*3*	*3.45*	*1.34*	*62%*
CCZ Complex	*8*	*20.0*	*199*	*35%*

Conclusions and Contributions

- Provide a methodology for studying application patterns from the transport layer.
- Confirm TCP is used for other than bulk transfers.
- Show that silence in connections is largely short.
- Show complexity in TCP traffic not captured by the current model.

Future Work

- Track usage of TCP by various applications over time.
- Empirically develop a more detailed TCP model.
- Examine what insight a new model provides.

Questions?