The UNIVERSITY *of* NORTH CAROLINA
at CHAPEL HILL

A Non-Parametric Approach to Generation and Validation of Synthetic Network Traffic

Félix Hernández-Campos
Kevin Jeffay
Don Smith
Department of Computer Science

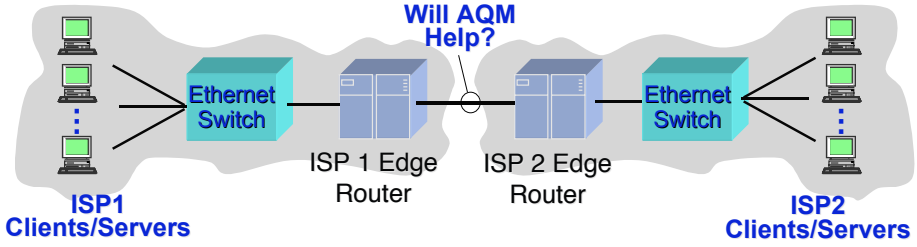
Andrew Nobel
Department of Statistics

http://www.cs.unc.edu/Research/dirt

Generation of Synthetic Traffic
Outline

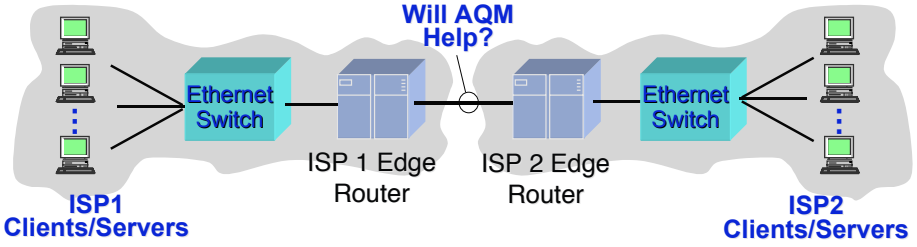
- The synthetic traffic generation problem — what is it and why should you care?
  - A simple case study of active queue management mechanisms
- A signature-based approach to modeling TCP connections
  - The *a-b-t* trace modeling paradigm
- Synthetic traffic generation — from traces to replayed connections
  - The *tmix* traffic generator
- Validation of synthetically generated traffic
  - Validation of intrinsic properties
  - Validation of extrinsic properties

Synthetic Traffic Generation
A simple example



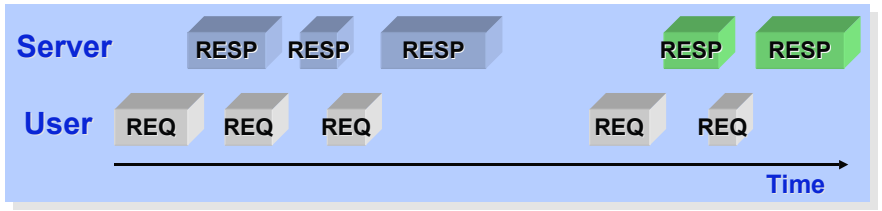
- How does one (empirically) evaluate if a new active queue management (AQM) scheme works?
  - Or new protocol, router architecture, ...
- You simulate it!
  - Simulate the network and the AQM scheme or use a real implementation
  - Simulate the use of the network by a population of users/ applications

Synthetic Traffic Generation
A simple example



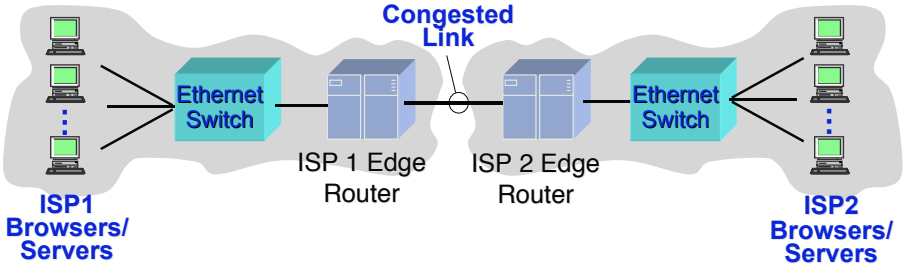
- The synthetic traffic generation problem: Simulating the use of a network by a population of users
- The Floyd, Paxson argument: source-level generation of traffic is preferred over packet-level generation
  - We desire *application-dependent*, *network independent* traffic generators
- Thus we need models of how applications generate traffic *and* a model of how users use applications

Source-Level Traffic Generation
Example: HTTP traffic generation



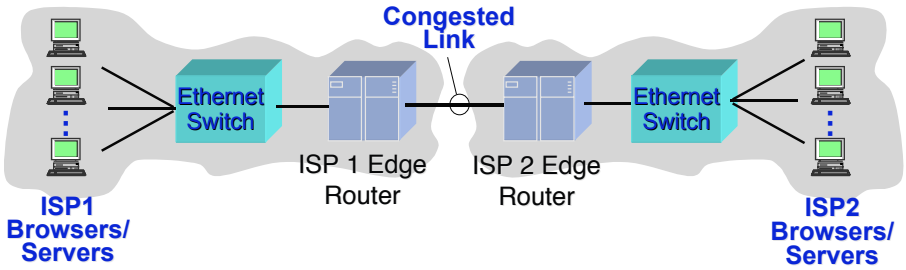
- *thttp* — The UNC synthetic web traffic generator [SIGMETRICS 2001, SIGCOMM 2003, MASCOTS 2003]
- Primary random variables:
  - Request sizes/Reply sizes
  - User think time
  - Persistent connection usage
  - Nbr of objects per persistent connection
  - Number of embedded images/page
  - Number of parallel connections
  - Consecutive documents per server
  - Number of servers per page

The Impact of Traffic Models
Case study: Evaluating AQM policies



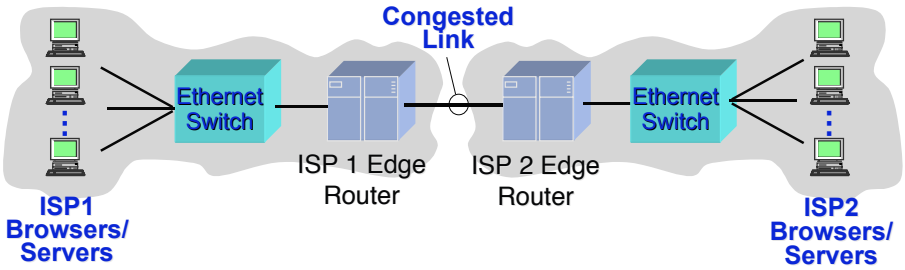
- We previously evaluated a number of prominent AQM schemes on an emulated ISP peering link carrying only web traffic [SIGCOMM03]
  - Construct a physical network emulating a congested peering link between two ISPs
  - Generate synthetic HTTP requests and responses but transmit data over real TCP/IP stacks, network links, and switches

The Impact of Traffic Models
Case study: Evaluating AQM policies



- We previously evaluated a number of prominent AQM schemes on an emulated ISP peering link carrying only web traffic [SIGCOMM03]
  - Compared drop-tail FIFO, PI, REM, ARED
  - Distribution of request-response response-times was the primary measure of performance
- Results: Control theoretic AQM good, ARED bad

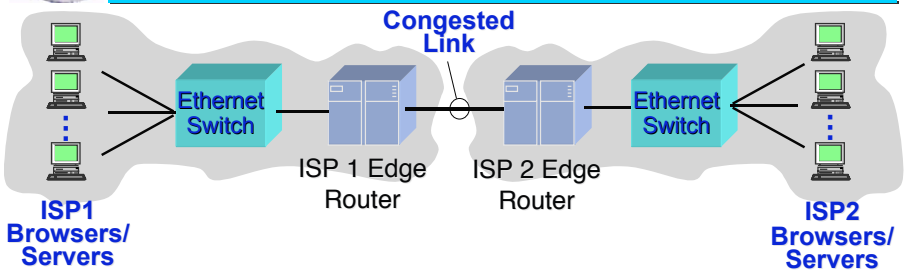
The Impact of Traffic Models
Case study: Evaluating AQM policies



- We previously evaluated a number of prominent AQM schemes on an emulated ISP peering link carrying only web traffic [SIGCOMM03]
- What's the impact of performing the experiments with a synthetic traffic *mix*?

The Impact of Traffic Models

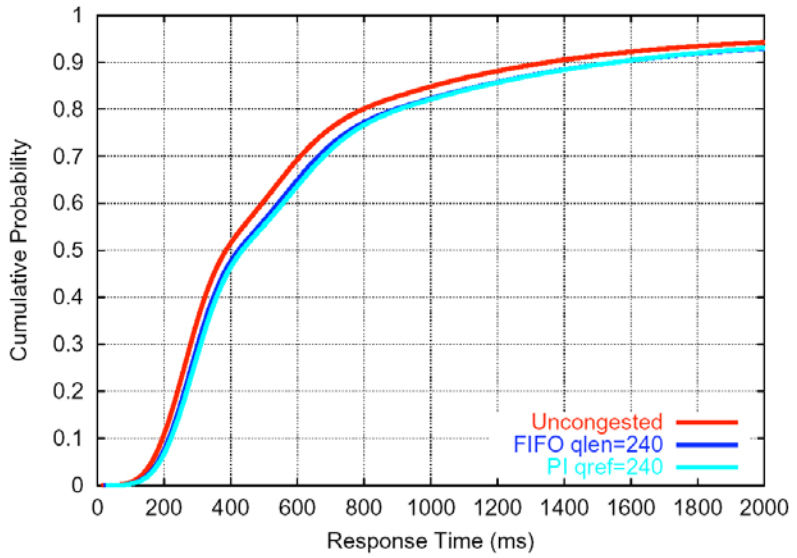
Case study: Evaluating AQM policies



- Experiment: Rerun experiments with a mix of TCP connections from an Abilene backbone
- Then rerun experiments using connections from the same trace but filtered for only HTTP (port 80) connections and scaled to achieve the same load as the "full" Abilene trace

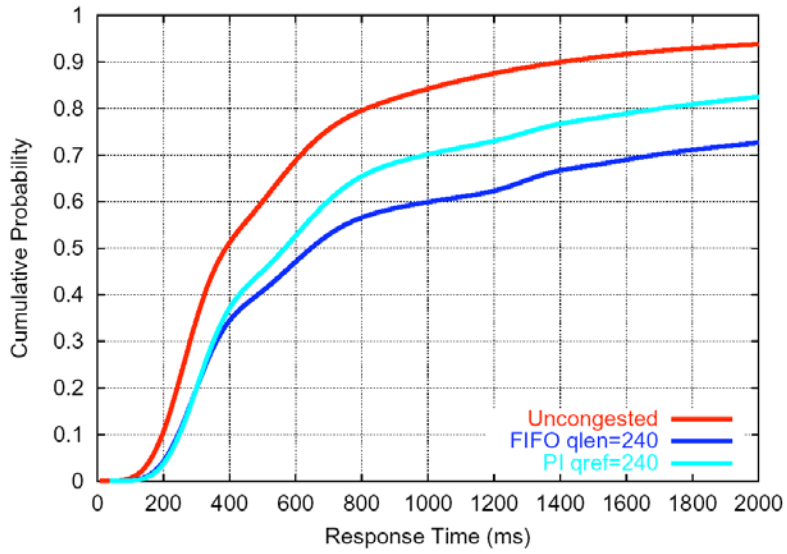
Impact of Using Traffic Mixes

Abilene HTTP only traffic, heavy load



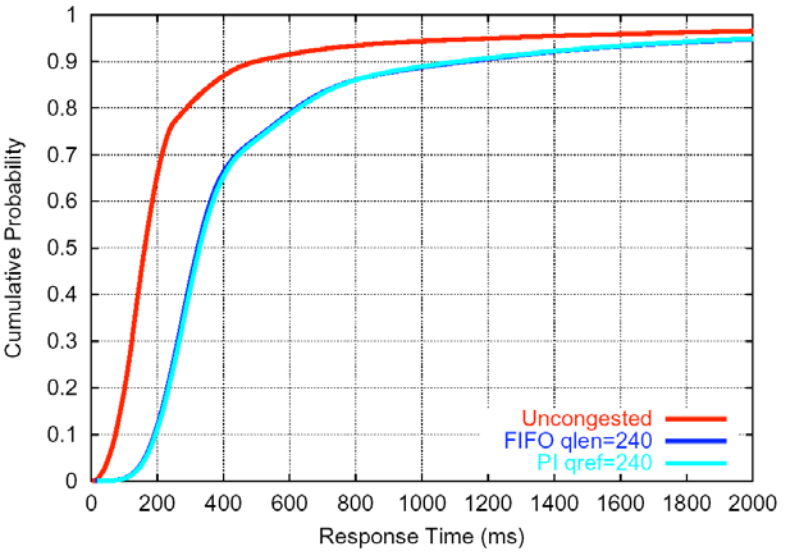
Impact of Traffic Models

Abilene HTTP only traffic, saturation load

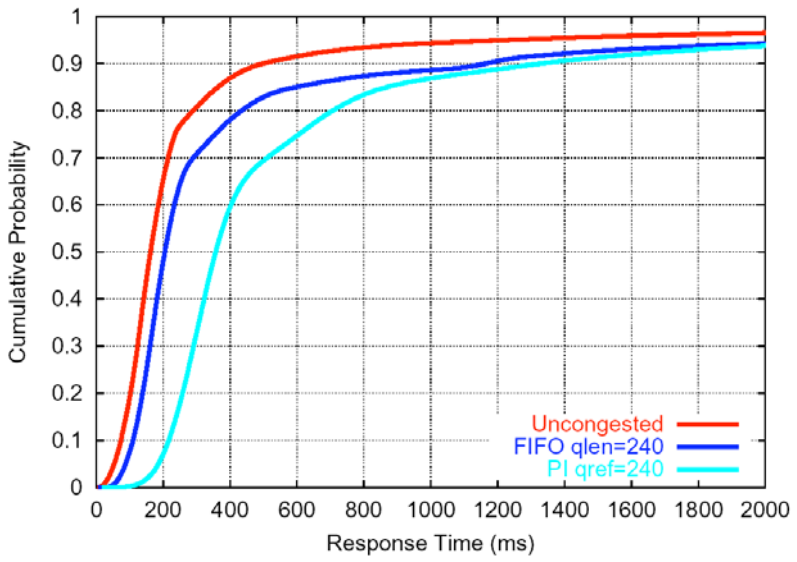


Impact of Traffic Models

Abilene all TCP connections, heavy load



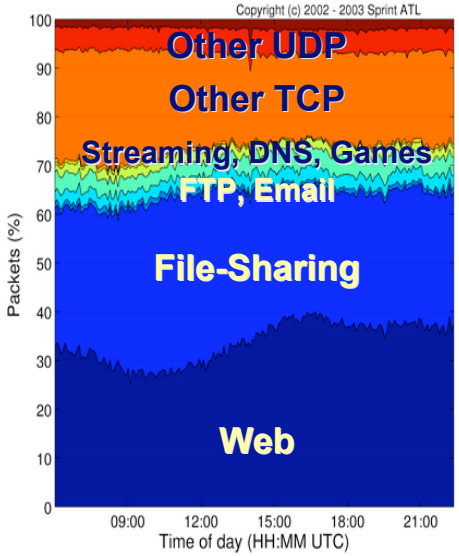
Impact of Traffic Models
Abilene all TCP connections, saturation load



Generation of Synthetic Traffic
Outline

- The synthetic traffic generation problem — what is it and why should you care?
  - A simple case study of active queue management mechanisms
- A signature-based approach to modeling TCP connections
  - The *a-b-t* trace modeling paradigm
- Synthetic traffic generation — from traces to replayed connections
  - The *tmix* traffic generator
- Validation of synthetically generated traffic
  - Validation of intrinsic properties
  - Validation of extrinsic properties

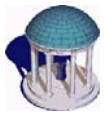
Source-Level Traffic Generation
Models for other common applications?



- Wide-area traffic is generated by *many* different applications
- Simulation/testbed experiments should generate "traffic mixes"
- Does the HTTP source-level model construction paradigm scale to other applications?

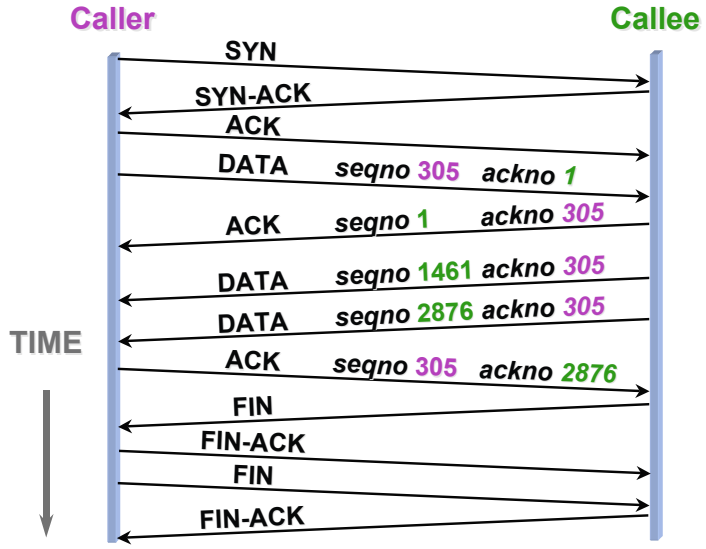
Constructing Source-Level Models
Steps for simple request/response protocols

- Obtain a trace of TCP/IP headers from a network link
  - (Current ethics dictate that tracing beyond TCP header is inappropriate without users' permission)
- Use changes in TCP sequence numbers (and knowledge of HTTP) to infer application data unit (ADU) boundaries
- Compute empirical distributions of the ADUs (and higher-level objects) of interest



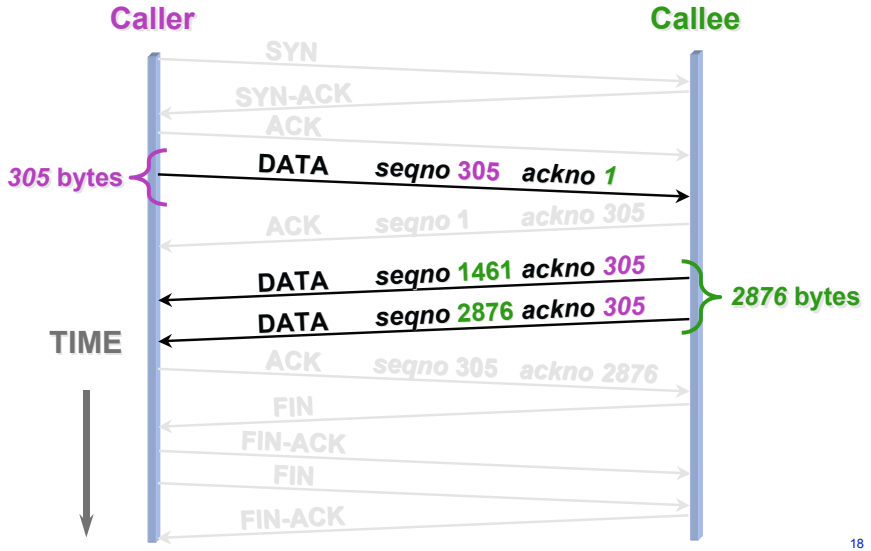
Ex: HTTP Model Construction

HTTP inference from TCP packet headers



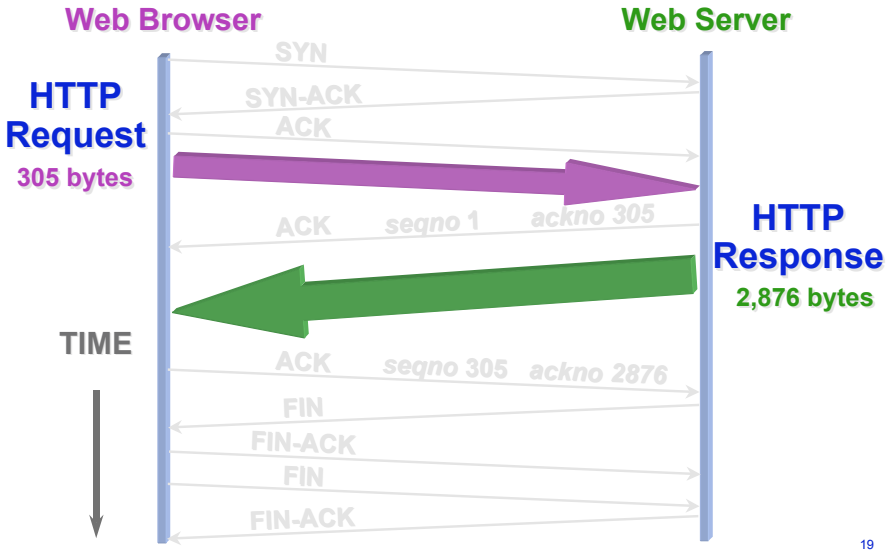
Ex: HTTP Model Construction

HTTP inference from TCP packet headers



Ex: HTTP Model Construction

HTTP inference from TCP packet headers



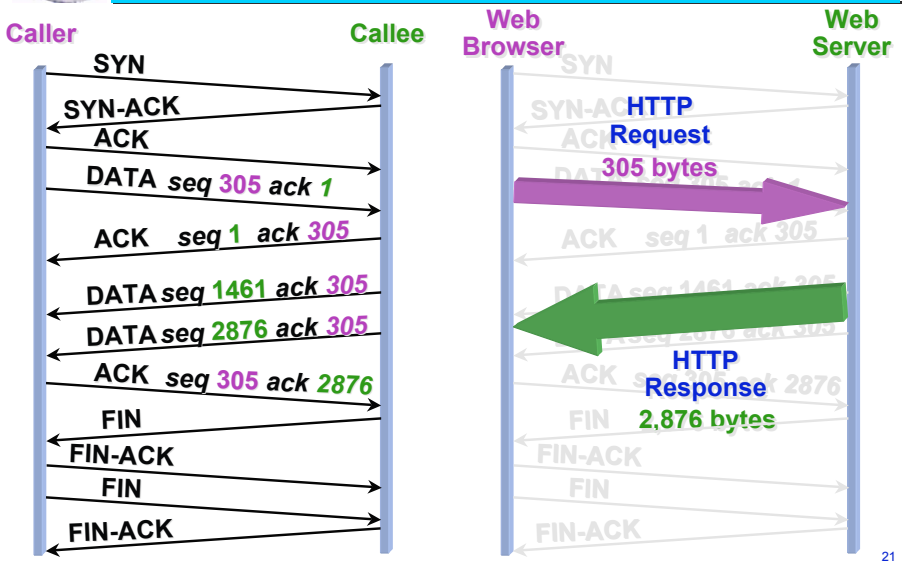
Source-Level Traffic Generation

Do current model generation methods scale?

- Implicit assumptions behind application modeling techniques:
  - We can identify the application corresponding to a given flow recorded during a measurement period
  - We can identify traffic generated by (instances) of the same application
  - We know the operation of the application-level protocol
- What's needed is an application-independent method of constructing source-level traffic models
  - We need to be able to construct application-level models of traffic without knowing what applications are being used or how the applications work
  - We need to construct source-level models of *application mixes* seen in real networks

TCP Connection Signatures

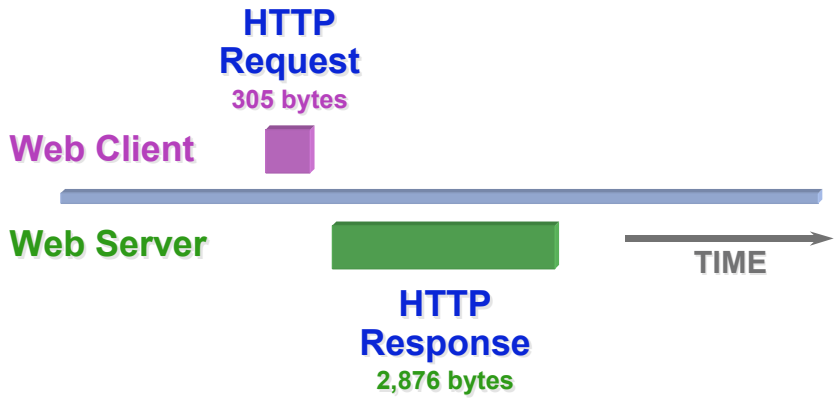
Recording communication "patterns"



TCP Connection Signatures

Recording communication "patterns"

- Communication pattern was (a_1, b_1)
  - *E.g.*, (305 bytes, 2,876 bytes)



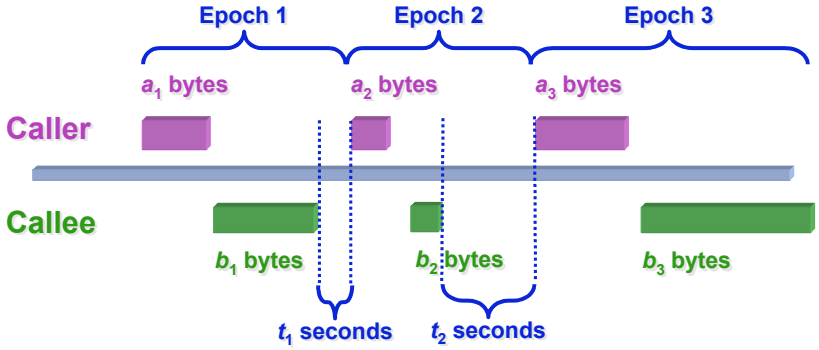
TCP Connection Signatures

The *a-b-t* trace model

- We model a TCP connection as *a-b-t* vector:

$$((a_1, b_1, t_1), (a_2, b_2, t_2), \ldots, (a_e, b_e, \perp))$$

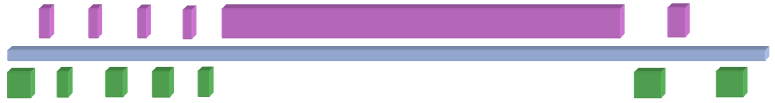
where e is the number of epochs



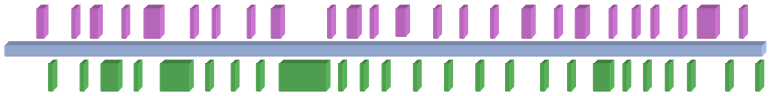
The *a-b-t* Trace Model

Typical Communication Patterns

- SMTP (send email)

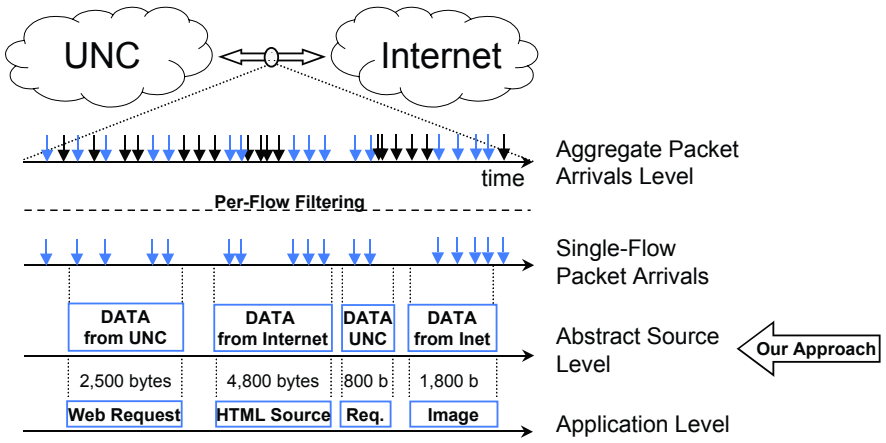


- Telnet (remote terminal)

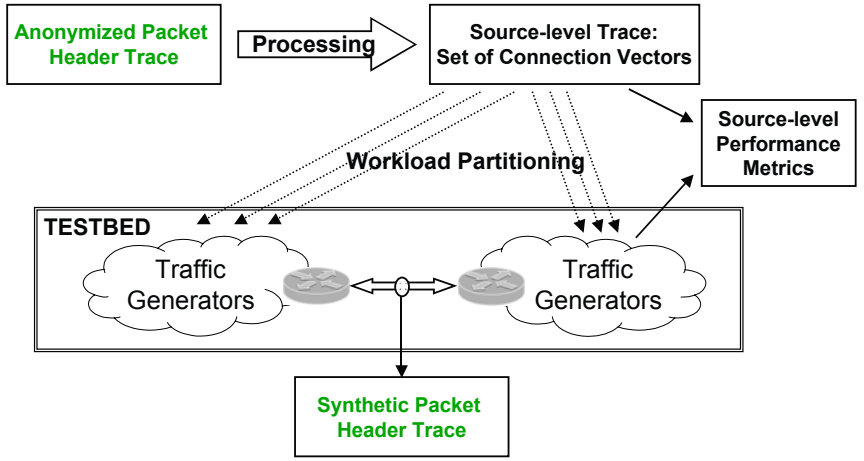


- FTP-DATA (file download)

The *a-b-t* Trace Model
Abstract Source-level Modeling

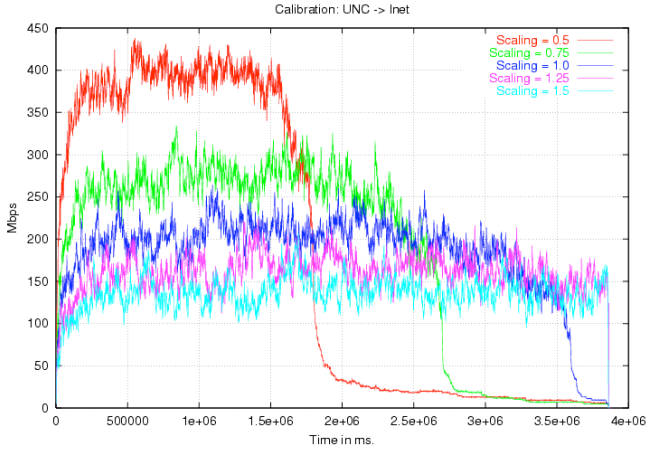


Source-Level Trace Replay
Traffic generation in a laboratory testbed



Source-Level Trace Replay
Traffic generation in a laboratory testbed

- Load can be scaled up/down by compressing TCP connection start times



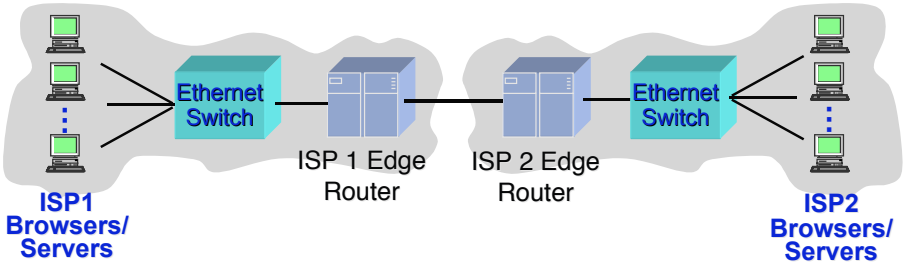
Generation of Synthetic Traffic
Outline

- The synthetic traffic generation problem — what is it and why should you care?
- A signature-based approach to modeling TCP connections
  - The *a-b-t* trace modeling paradigm
- Synthetic traffic generation — from traces to replayed connections
  - The *tmix* traffic generator
- Validation of synthetically generated traffic
  - Validation of intrinsic properties
  - Validation of extrinsic properties

Validation of Generated Traffic
Approach

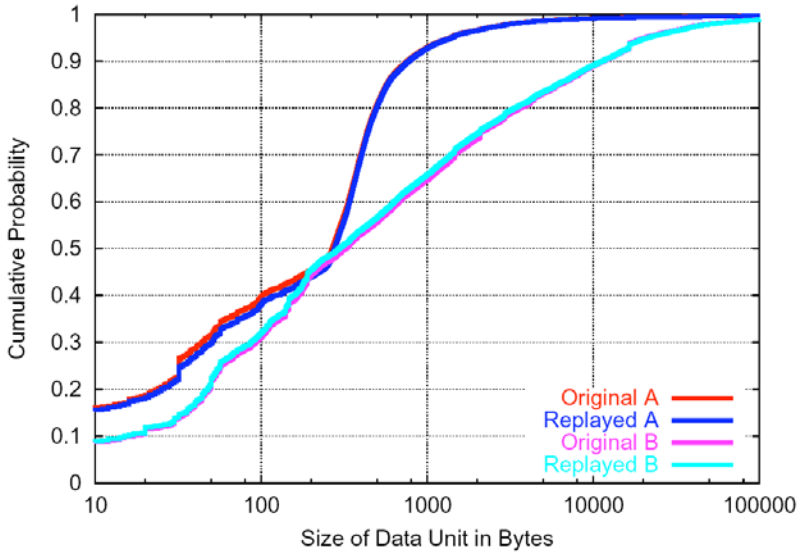
- Acquire a packet header trace of TCP connections from an Internet link
- Derive a new trace $\mathcal{T}$ of *a-b-t* connection vectors from the Internet trace
- Use $\mathcal{T}$ to generate synthetic traffic in a laboratory testbed using the *tmix* traffic generator
- Record a packet header trace of the generated traffic on the testbed link
- Compare various properties of the traffic in the testbed trace with the corresponding traffic from the Internet link

Validation of Generated Traffic
Validation of synthetic Abilene traffic

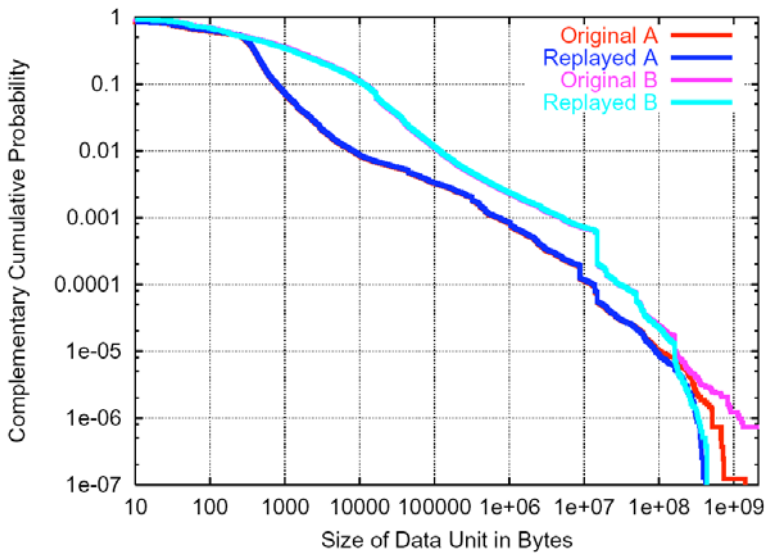


- Testbed: An Internet emulation facility
  - 150+ end-systems, 10/100/1,000 Mbps connectivity, dozens of switches routers
- Input trace: A 2-hour Abilene trace from the NLANR repository
  - 334 billion bytes, 404 million packets, 5 million TCP connections

Comparison of Intrinsic Properties
Distribution of *a* and *b* sizes (body)

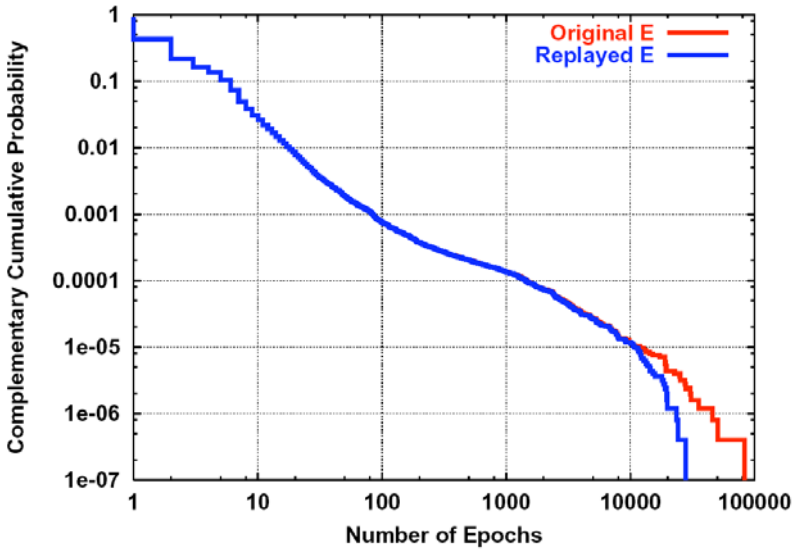


Comparison of Intrinsic Properties
Distribution of *a* and *b* sizes (tail)



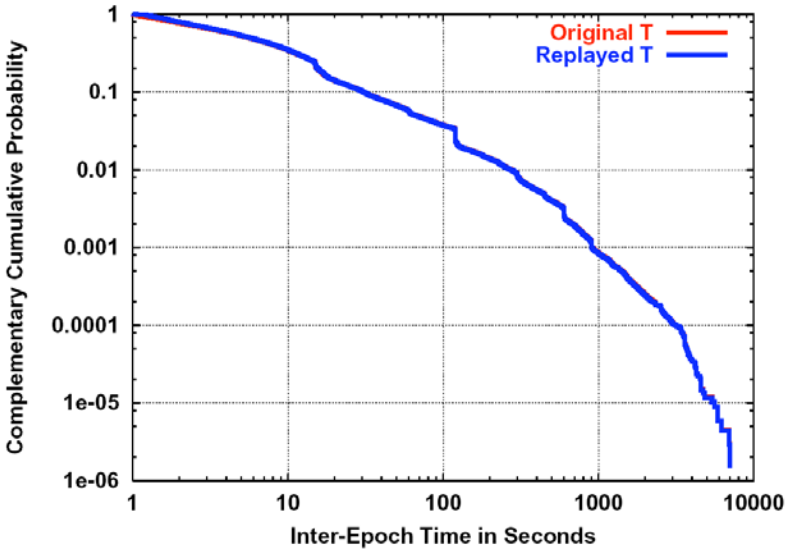
Comparison of Intrinsic Properties

Distribution of number of epochs/connection



Comparison of Intrinsic Properties

Distribution of inter-epoch times



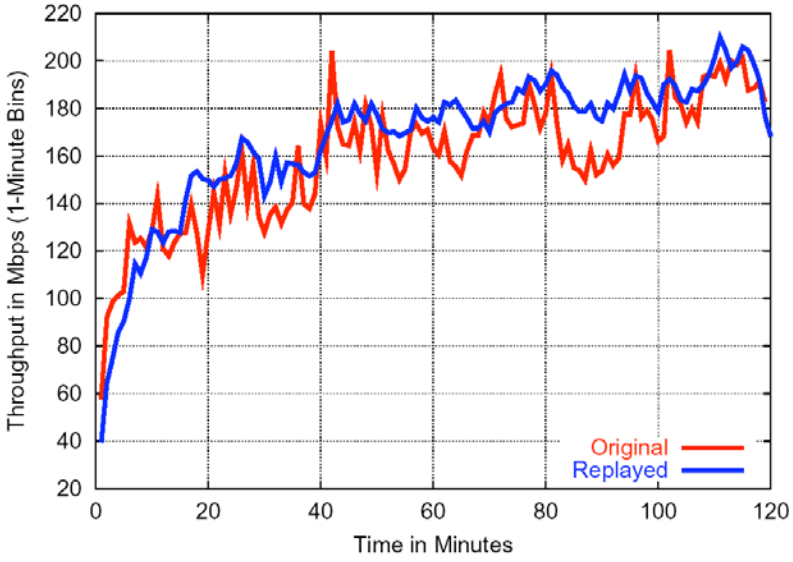
Validation of Generated Traffic

Intrinsic v. extrinsic properties

UNIVARIATE				MULTIVARIATE		
a_{tot}	b_{tot}	t_{tot}	Total bytes/time	***cor.a.b***	***cor.a.t***	***cor.b.t***
a_{max}	b_{max}	t_{max}	Max bytes/time	Correlations		
a_{min}	b_{min}	t_{min}	Min bytes/time	***cor.a.b.x***	***cor.a.t.x***	***cor.b.t.x***
a_{mean}	b_{mean}	t_{mean}	Mean bytes/time	Lagged Correlations		
a_{xq}	b_{xq}	t_{xq}	1st 2nd 3rd Quartiles	***crc.a.b***	***crc.a.t***	***crc.b.t***
a_{stdev}	b_{stdev}	t_{stdev}	Standard Deviation	Cross-correlations		
$a_{cor.x}$	$b_{cor.x}$	$t_{cor.x}$	Autocorrelations	***dir1.a.b***	***dir2.a.b***	
a_{hx}	b_{hx}	t_{hx}	Homogeneity	Directionality		
a_{vs}	b_{vs}	t_{vs}	Total Variation	**UNIVARIATE**		
a_{vm}	b_{vm}	t_{vm}	Max First Diff.	***e***	No. of Epochs	

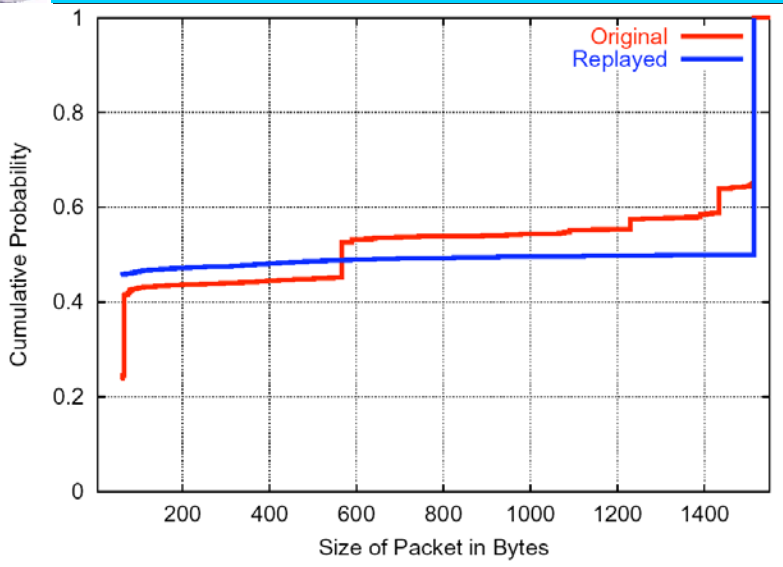
Comparison of Extrinsic Properties

Throughput — Abilene westbound



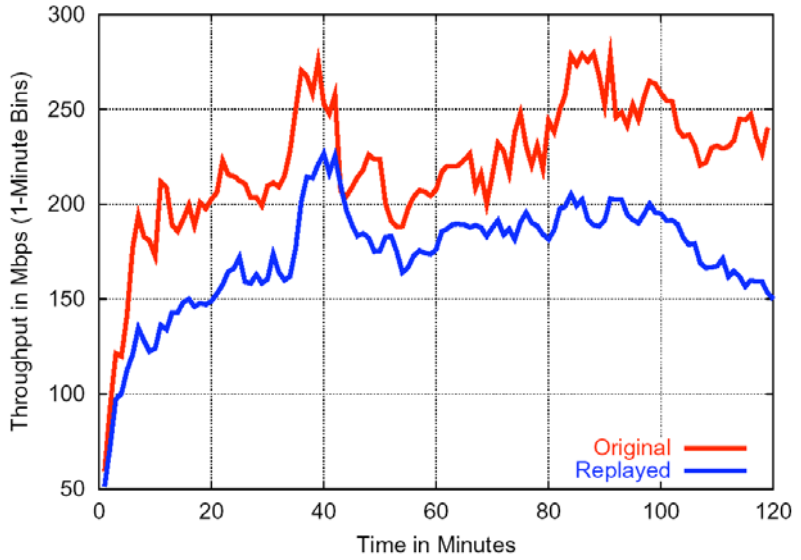
Comparison of Extrinsic Properties

Distribution of pkt sizes — Abilene westbound



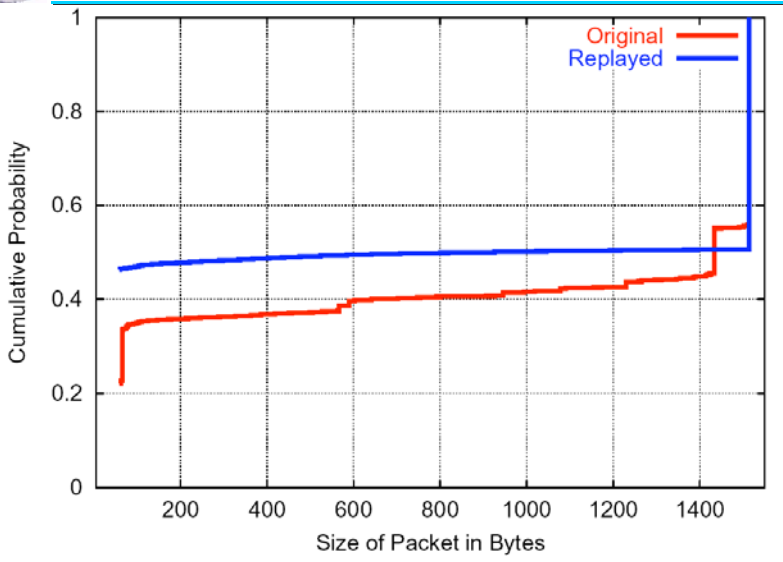
Comparison of Extrinsic Properties

Throughput — Abilene eastbound



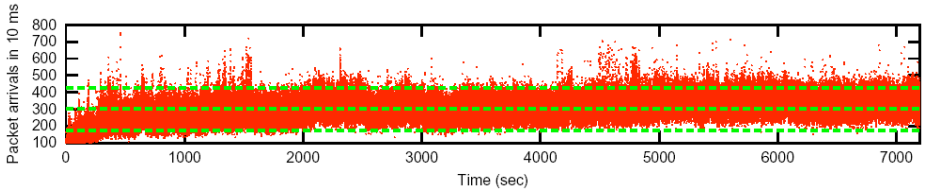
Comparison of Extrinsic Properties

Distribution of pkt sizes — Abilene eastbound

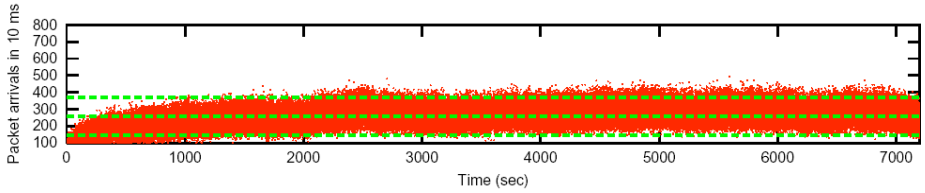


Comparison of Extrinsic Properties

Packet arrivals/10 *ms* — Abilene eastbound

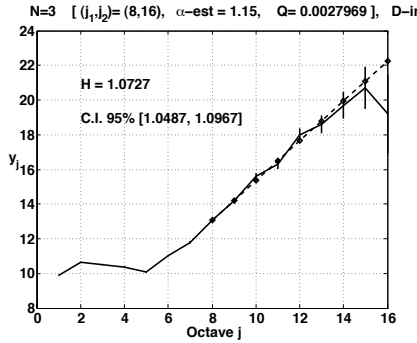


- Data from original Abilene trace

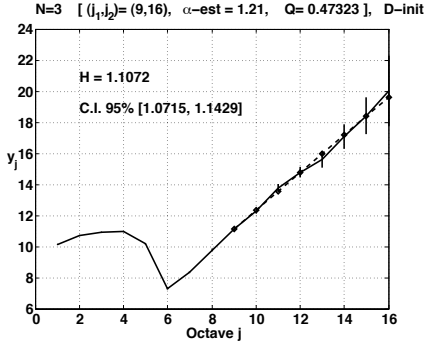


- Data from testbed replay of Abilene trace

Comparison of Extrinsic Properties
Wavelet spectrum — Abilene westbound

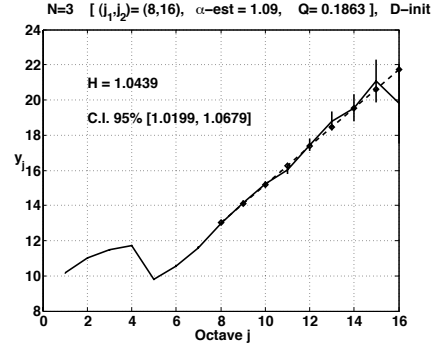


- Data from original Abilene trace

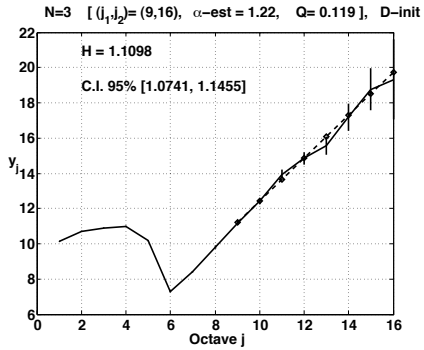


- Data from testbed replay of Abilene trace

Comparison of Extrinsic Properties
Wavelet spectrum — Abilene eastbound



- Data from original Abilene trace



- Data from testbed replay of Abilene trace

Synthetic Traffic Generation
Summary

- Simulation is the backbone of networking research
- Too little attention is paid to realistic traffic generation
  - How can we derive fundamental truths from today's simulation results?
- We advocate modeling traffic as patterns of data exchange patterns within TCP connections
  - Application-independent, network-independent
- Development of new, flexible traffic generators
  - Cluster-based synthetic traffic generation
- Validation — Attempting to understand and articulate which properties of traffic matter most and how they can be controlled

Future Work
Lots!

- Plenty more variables to understand:
  - Alternate scaling paradigms (*e.g.*, sampling)
  - Effect of tracing duration (minutes or hours?)
  - Effects of end-system parameters on extrinsic properties
- Still have yet to experiment with UDP modeling UDP connections