The UNIVERSITY *of* NORTH CAROLINA
at CHAPEL HILL

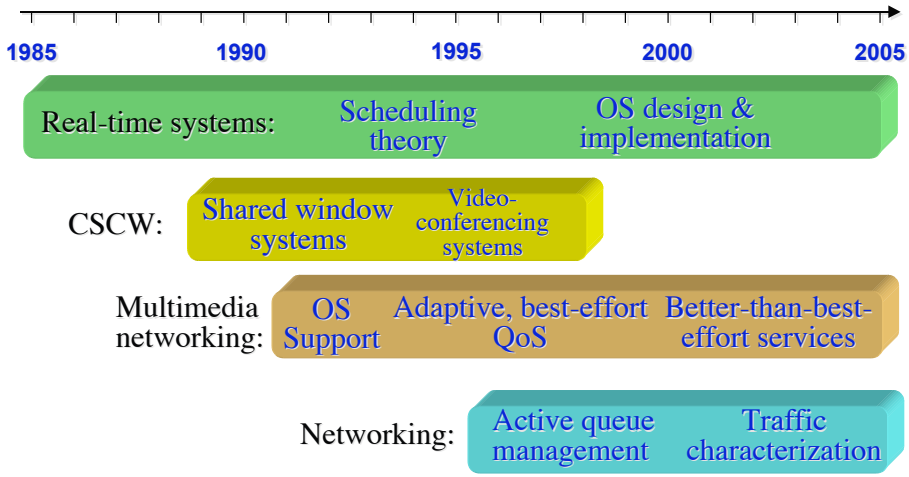
How "Real" Can Synthetic Network Traffic Be?

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

Andrew Nobel
Department of Statistics

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

Summary of Past Work
Research timeline



Summary of Past Work
Current projects

- Practical multiprocessor scheduling
- Rate-based resource allocation for embedded systems
- Adaptive haptics in distributed virtual environments
- Differential congestion notification
- Internet tomography

Real-time systems

Multimedia networking

Networking

Generation of Synthetic Traffic
Outline

- The synthetic traffic generation problem
  - The case for *source-level traffic modeling*
- A signature-based approach to modeling TCP connections
  - The *a-b-t* trace modeling paradigm

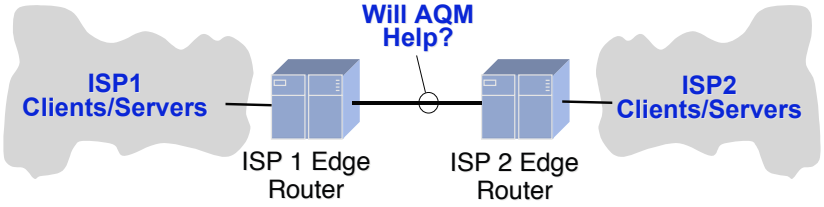
Today

- Synthetic traffic generation
  - The *tmix* traffic generator
- Validation of synthetically generated traffic

"Tomorrow"

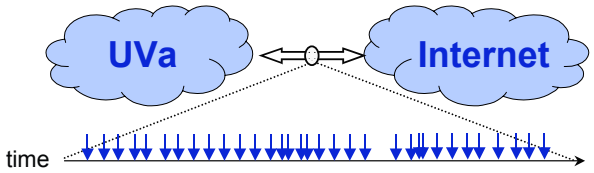
- Traffic analysis and characterization
  - Statistical cluster analysis of connection signatures
- Applications:
  - Workload evolution
  - DDoS detection
  - …

Synthetic Traffic Generation
A simple example



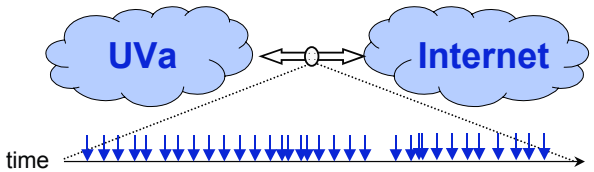
- How does one (empirically) evaluate if a new active queue management (AQM) scheme works?
  - Or new protocol, router architecture, ...
- You simulate it!
  - Simulate the network and the AQM scheme in software, or use a real AQM implementation in a testbed
  - Simulate a set of traffic generation processes

Synthetic Traffic Generation
A simple example



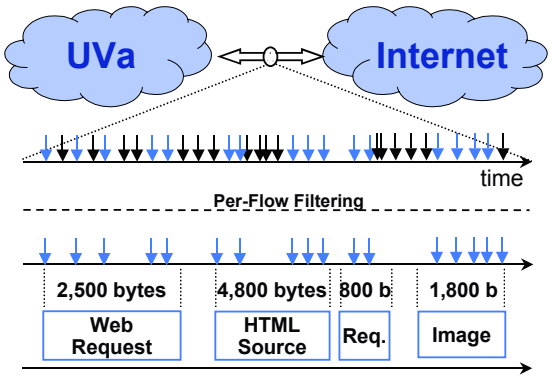
- "Realistic" traffic generation:
  - Collect a packet trace from a link of interest
    - » arrival times, packet sizes, …
  - Replay the trace directly, or
  - Model the trace and use the model to generate statistically similar traces
- Will the resulting traffic be "real" enough?

Synthetic Traffic Generation
Source-level traffic generation

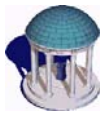


- Since the network shapes the traffic, what about the traffic is invariant of the network?
  - Axiom: The application/user's behavior is invariant of low-level network processes
- The Floyd, Paxson argument: source-level generation of traffic is preferred over packet-level generation
  - We desire *application-dependent*, *network independent* models of traffic

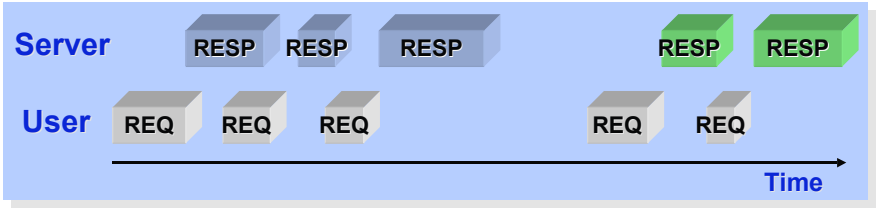
Synthetic Traffic Generation
Source-level traffic generation



- We need models of how applications generate traffic
  - Models of application protocols plus models of how applications are used by users
- Approaches:
  - Analytic models
  - Empirical models



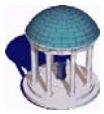
Source-Level Traffic Generation
Example: HTTP traffic generation



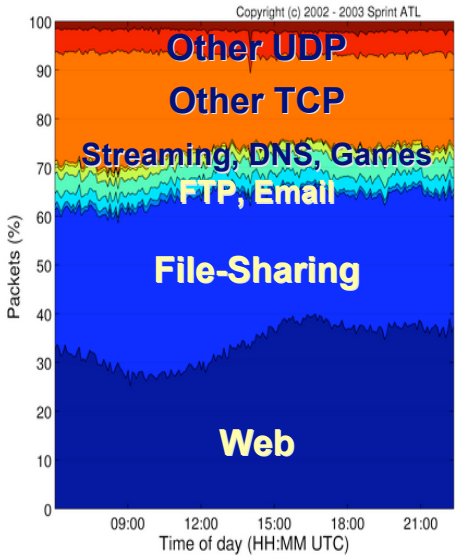
- *thttp* — The UNC synthetic web traffic generator [SIGMETRICS 2001, SIGCOMM 2003, MASCOTS 2003]
- Primary random variables:
  - Request sizes/Reply sizes
  - User think time
  - Persistent connection usage
  - Nbr of objects per persistent connection
  - Number of embedded images/page
  - Number of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

Generation of Synthetic Traffic
Outline

- The synthetic traffic generation problem
  - The case for source-level traffic modeling
- 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
  - Reproduction of *source-level* properties
  - Reproduction of *end-system* properties
  - Reproduction of *path* properties



Source-Level Traffic Generation
The failure of existing approaches



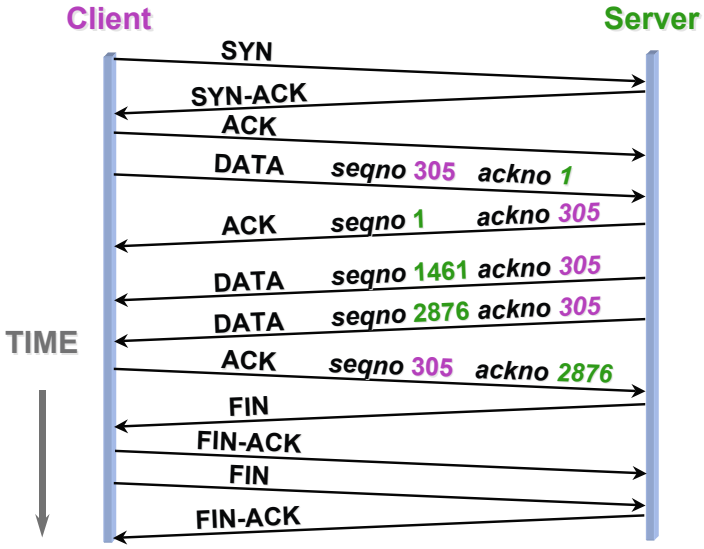
- Dominant approach is to model individual applications
- Wide-area traffic is generated by *many* different applications
- Simulation/testbed experiments should generate "traffic mixes"
- Does the HTTP source-level model construction paradigm scale to other applications?

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

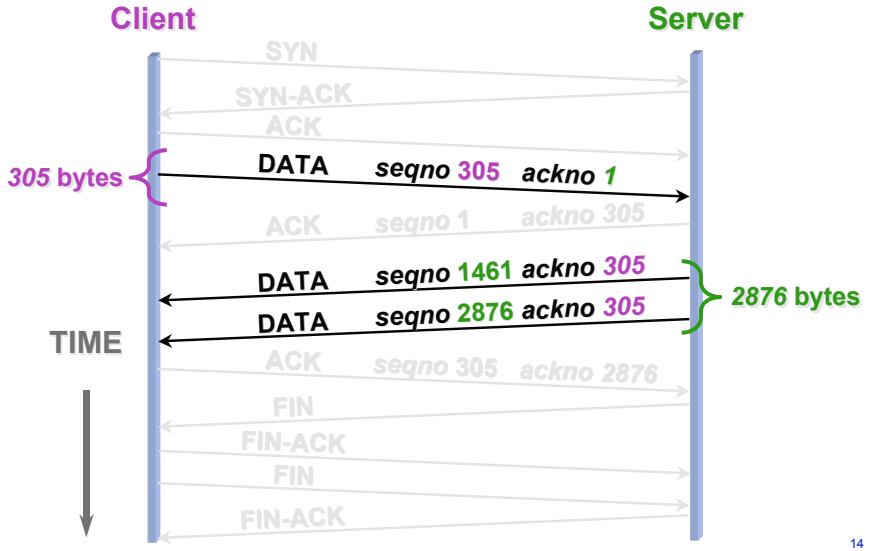
Ex: HTTP Model Construction

HTTP inference from TCP packet headers



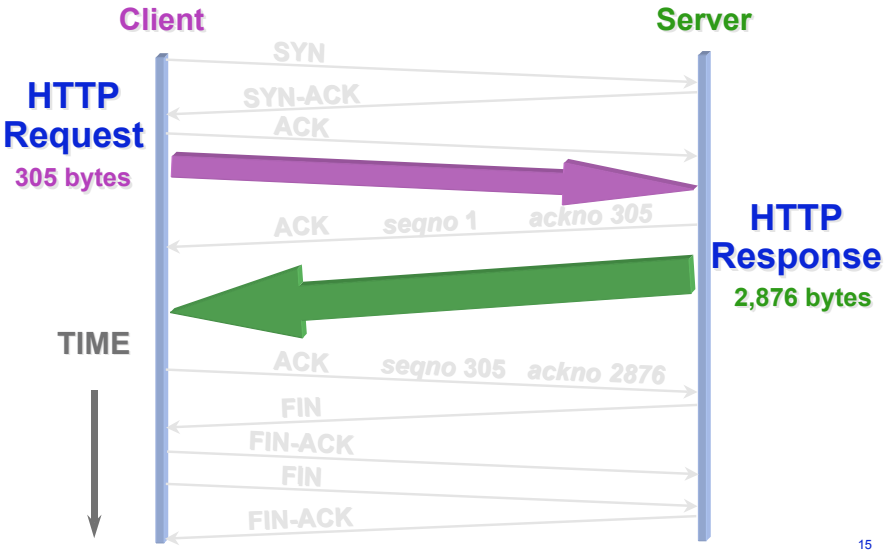
Ex: HTTP Model Construction

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?

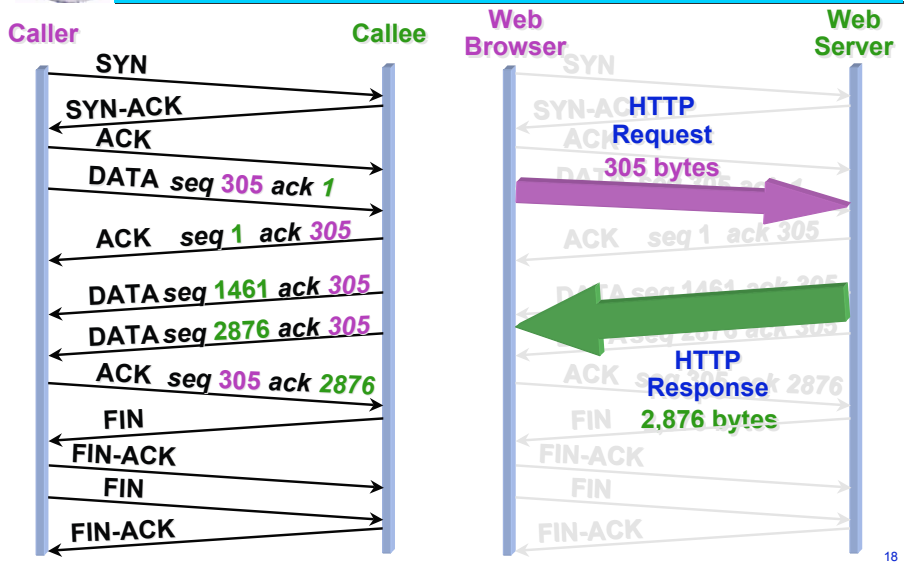
- Implicit assumptions behind application modeling techniques:
  - We can identify the application corresponding to a given flow recorded during a measurement period
  - We can identify traffic generated by (instances) of the same application
  - We know the operation of the application-level protocol

- Ex: The HTTP success story:
  - 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

Source-Level Traffic Generation
Do current model generation methods scale?

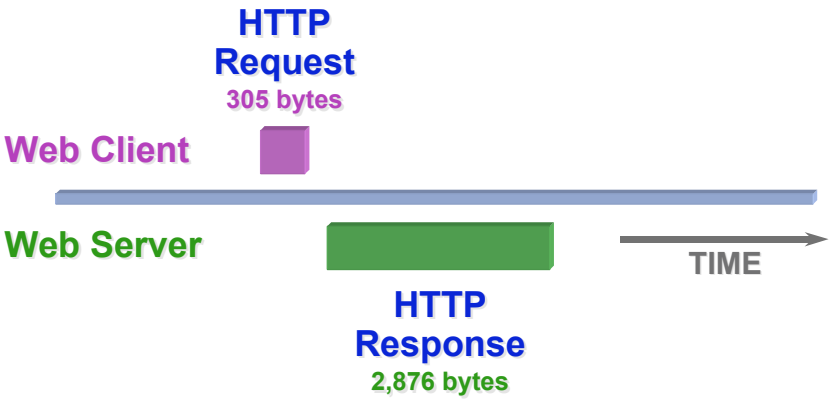
- Implicit assumptions behind application modeling techniques:
  - We can identify the application corresponding to a given flow recorded during a measurement period
  - We can identify traffic generated by (instances) of the same application
  - We know the operation of the application-level protocol
- 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)

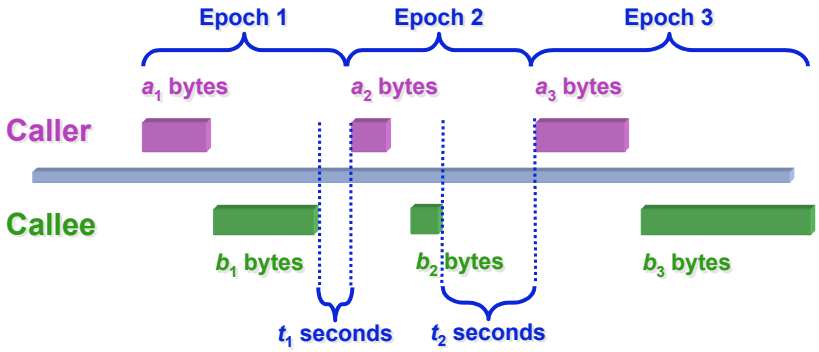


TCP Connection Signatures
The *a-b-t* trace model

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

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

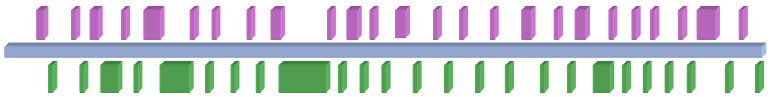
where e is the number of epochs



The *a-b-t* Trace Model
Typical Communication Patterns

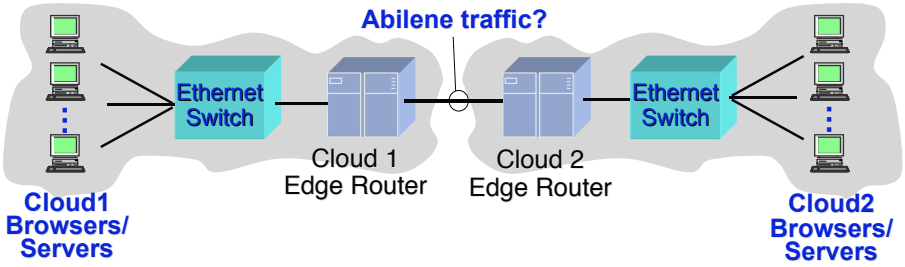
- SMTP (send email)

- Telnet (remote terminal)



- FTP-DATA (file download)

Source-Level Trace Replay
Traffic generation in a laboratory testbed

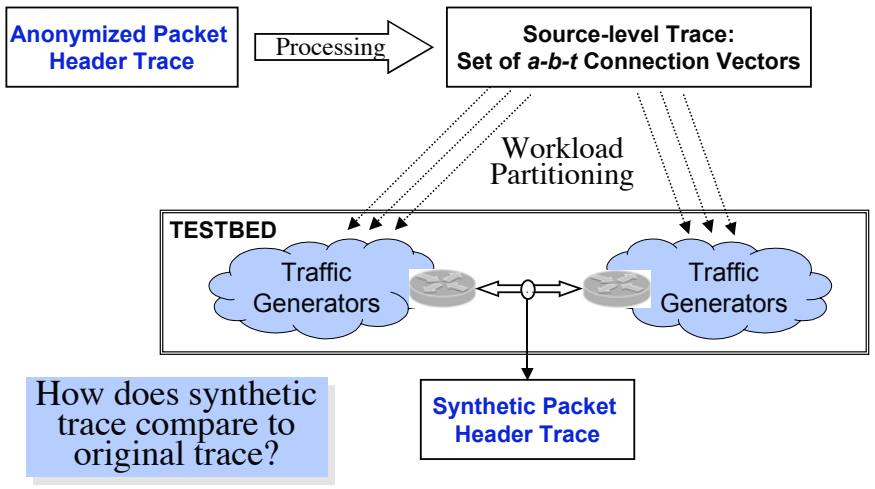


- Given a testbed or simulator, can we effectively simulate Abilene?
  - Can we simulate "the Internet" in a lab or inside a modest computer using a simple dumbbell topology?
  - Can we get away from having to make arbitrary decisions about how we generate synthetic traffic?

Source-Level Trace Replay
Traffic generation in a laboratory testbed

- Testbed:
  - 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

Source-Level Trace Replay
Traffic generation in a laboratory testbed

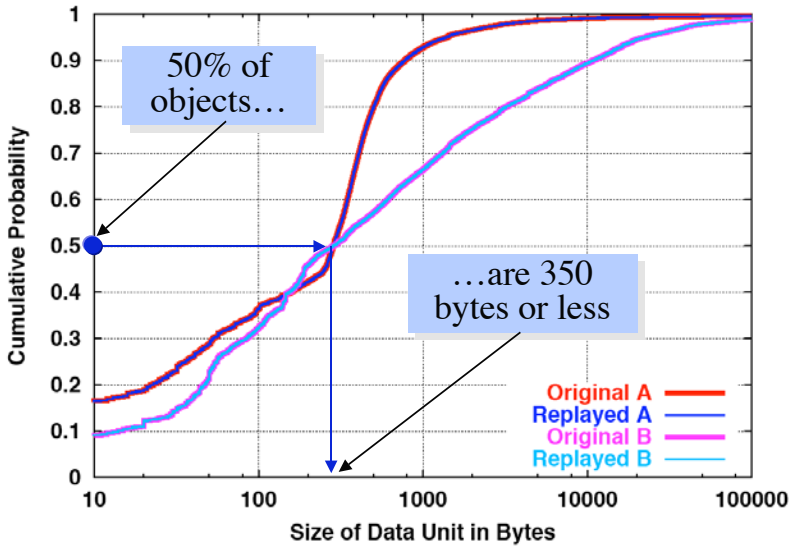


How does synthetic trace compare to original trace?

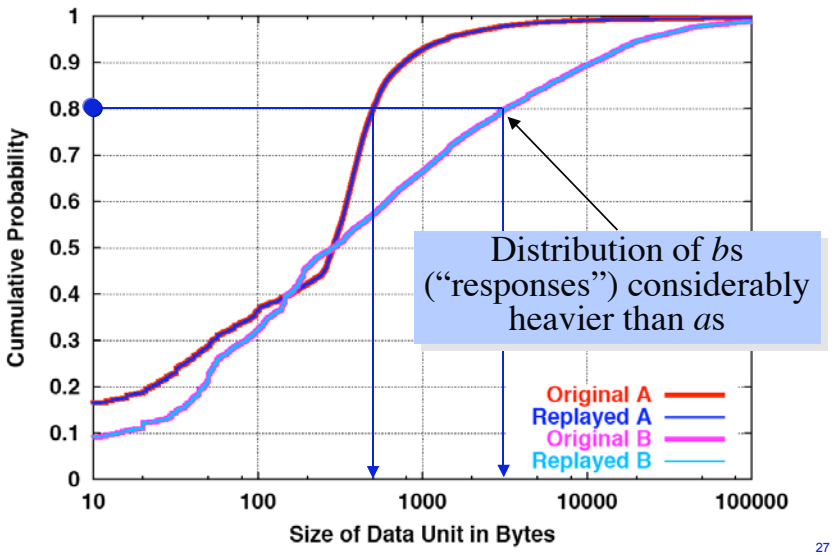
Validation of Generated Traffic
Questions

- Can we reproduce source-level properties of the original traffic?
- Can we reproduce interesting measures of the original trace?
  - Throughput per unit time
  - Number of active connections per unit time
  - Connection transmission rates
  - Long range dependence in packet and byte arrivals
  - …
- Can we see interesting differences between UNC traffic and Abilene traffic?

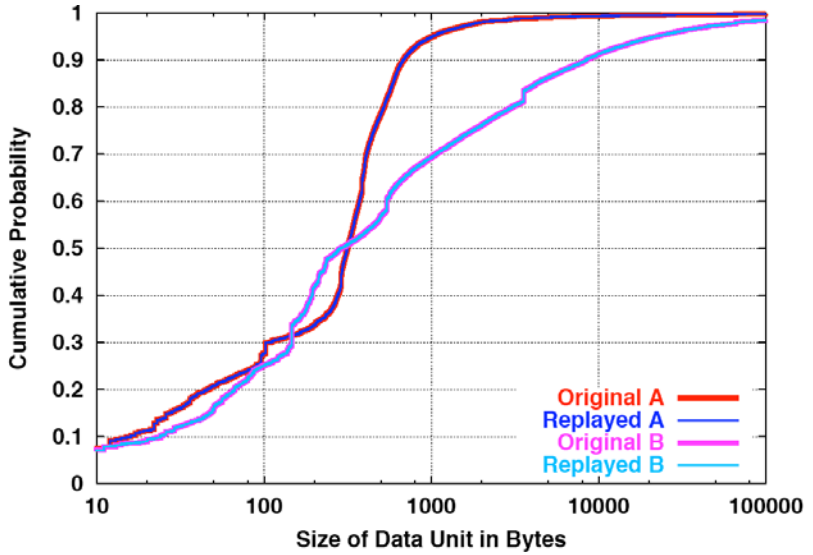
Verification of Source-Level Properties
Distribution of *a* and *b* sizes (Abilene)



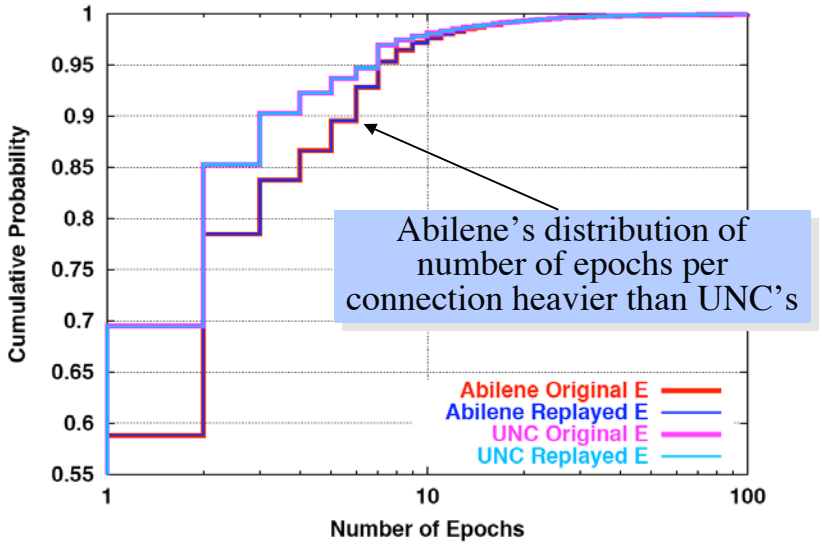
Verification of Source-Level Properties
Distribution of *a* and *b* sizes (Abilene)



Verification of Source-Level Properties
Distribution of *a* and *b* sizes (UNC)

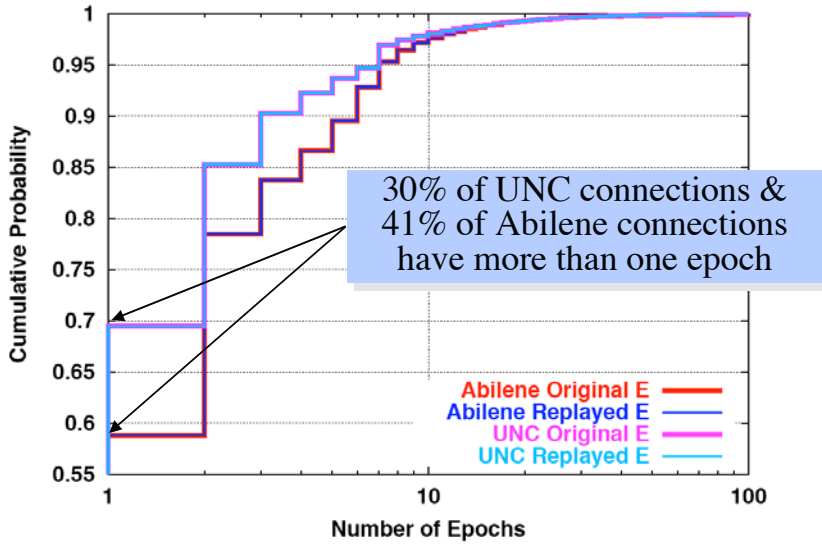


Verification of Source-Level Properties
Distribution of epochs per connection



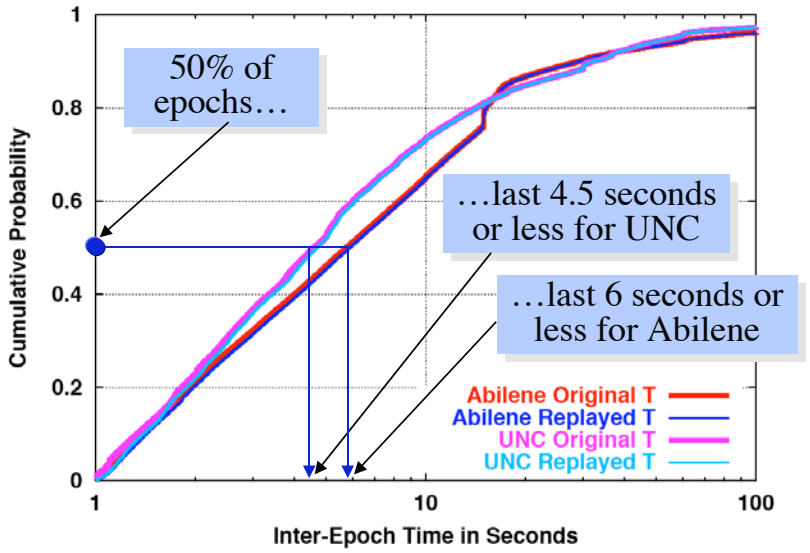
Abilene's distribution of number of epochs per connection heavier than UNC's

Verification of Source-Level Properties
Distribution of epochs per connection



30% of UNC connections & 41% of Abilene connections have more than one epoch

Verification of Source-Level Properties
Distribution of inter-epochs times

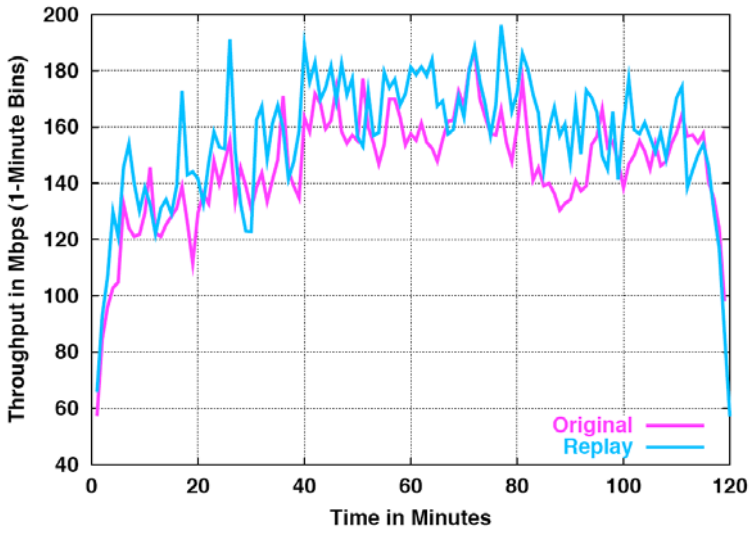


50% of epochs…

…last 4.5 seconds or less for UNC

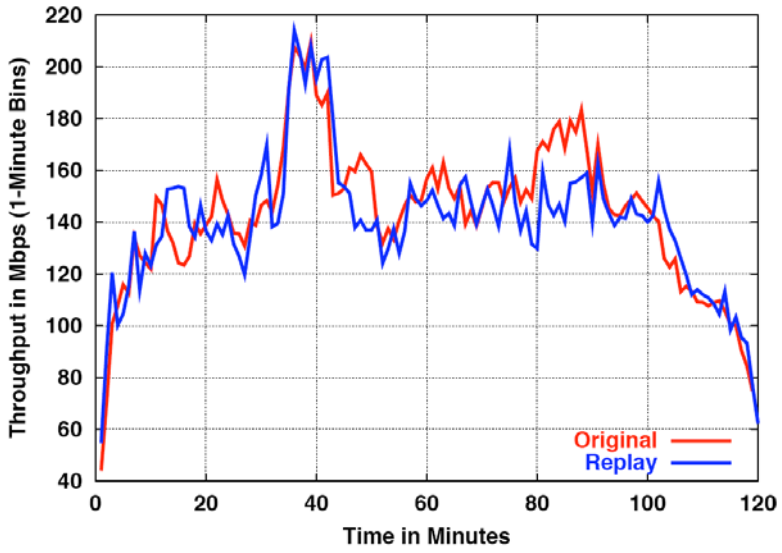
…last 6 seconds or less for Abilene

Reproduction of Throughput
Abilene tput — Cleveland to Indianapolis



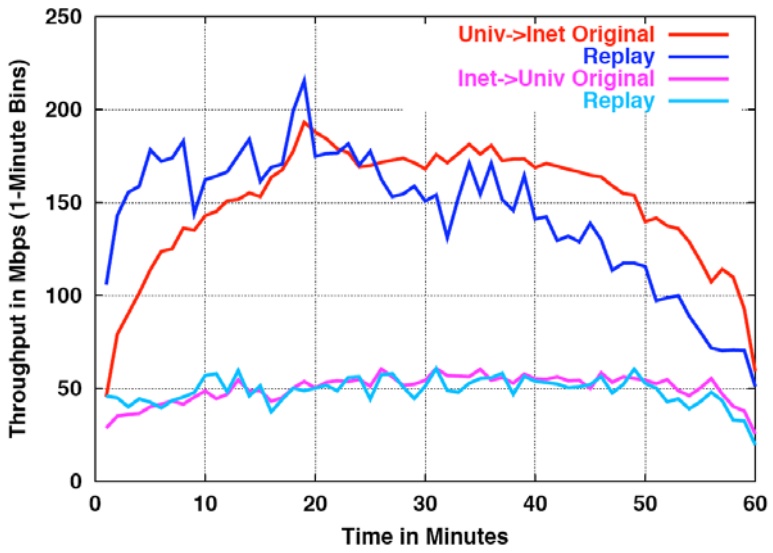
Reproduction of Throughput

Abilene tput —Indianapolis to Cleveland



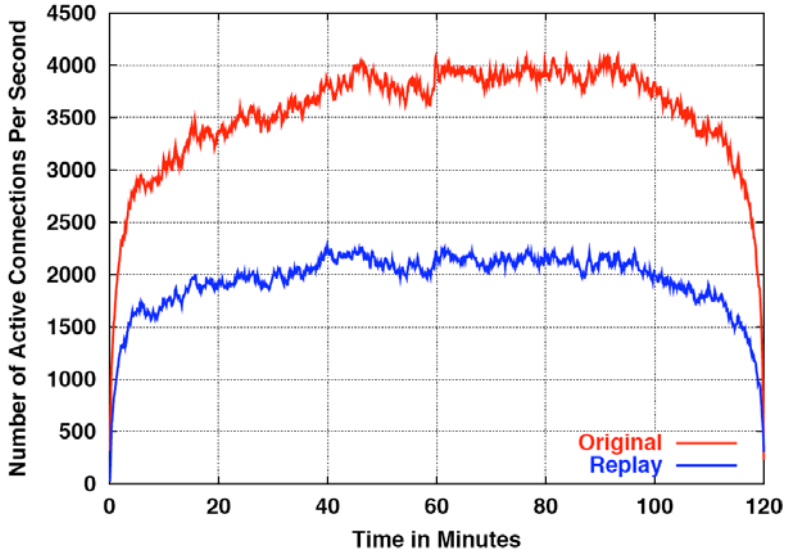
Reproduction of Throughput

UNC throughput —Inbound & outbound



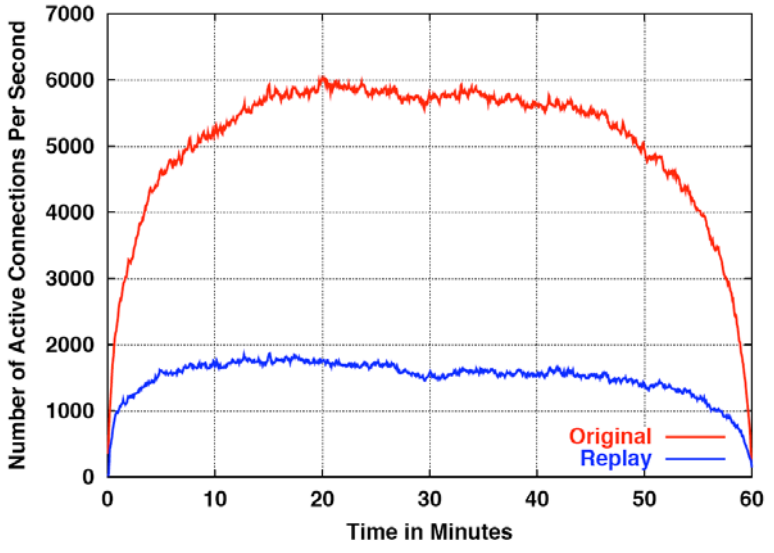
Reproduction of Active Connections

Abilene replay *v.* original



Reproduction of Active Connections

UNC replay *v.* original



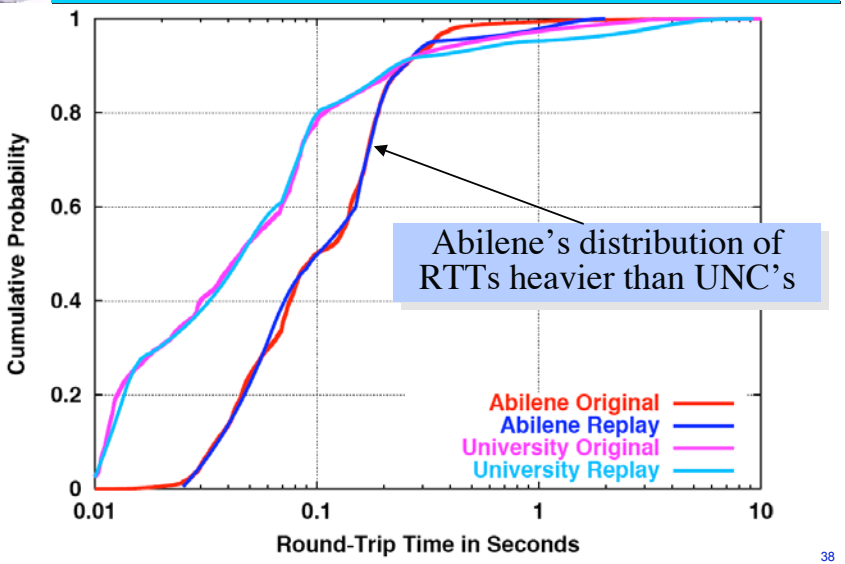
Validation of Synthetic traffic
Summary

- We accurately reproduce source-level properties
- This is sufficient for realistic reproduction of some interesting performance measures (throughput)
- Overall, we're replaying connections too fast
- This argues for modeling of end-system and path properties
  - TCP window size distributions
  - Round-trip time distributions
  - Bottleneck transmission rate distributions
  - Loss rates, …

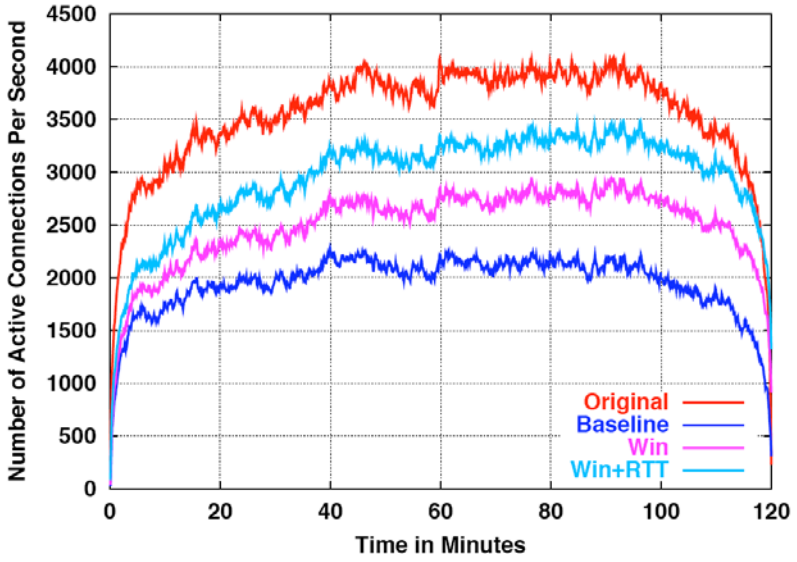
Testbed endsystems too homogenous!

- Fundamental question: What is the minimal level of modeling necessary for an acceptable level of realism?
  - Can the necessary parameters be derived from a header trace?
  - Can we still model the Internet with a dumbbell network?

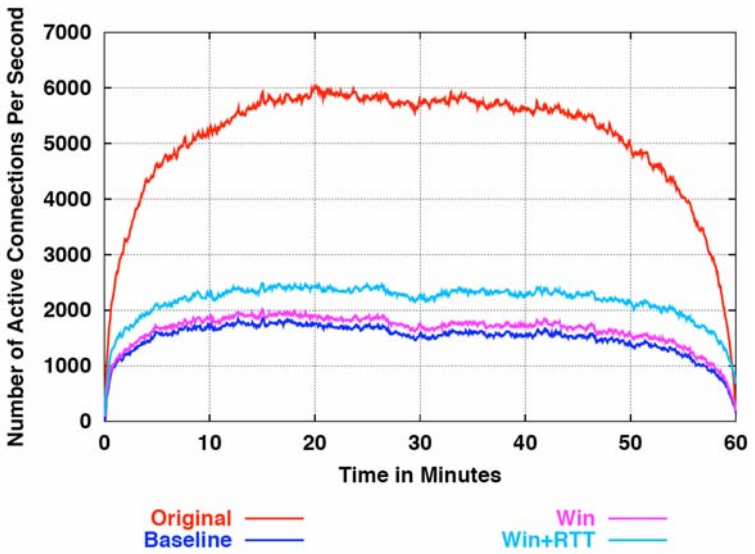
Reproduction of Round-Trip Times
Abilene/UNC replay *v.* original



Reproduction of Active Connections
Abilene replays *v.* original

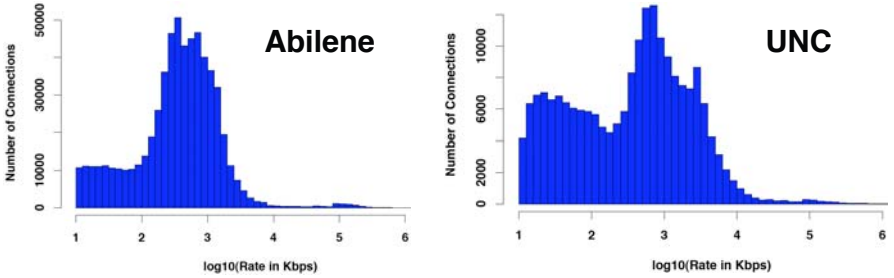


Reproduction of Active Connections
UNC replays *v.* original



Connection Transmission Rates

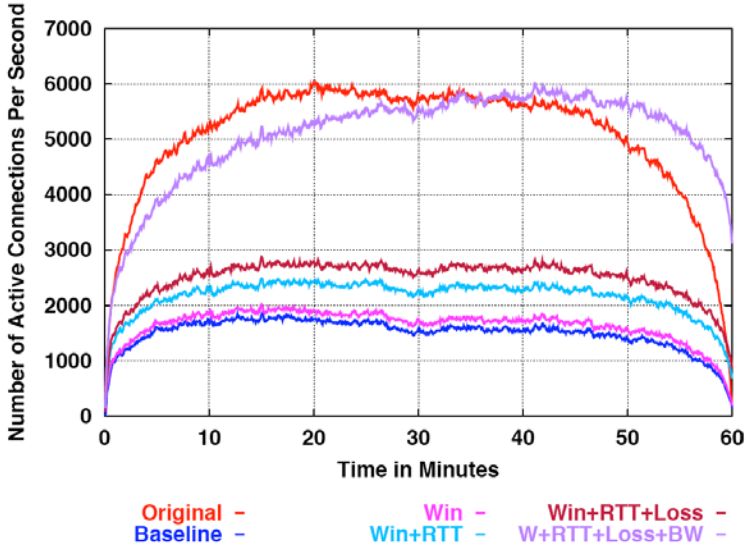
Abilene & UNC rates



- UNC connections have a larger concentration of mass in the lower transmission rates
  - UNC has a higher percentage of bandwidth limited flows than Abilene
  - This suggestions introduces some bandwidth limitations into the testbed

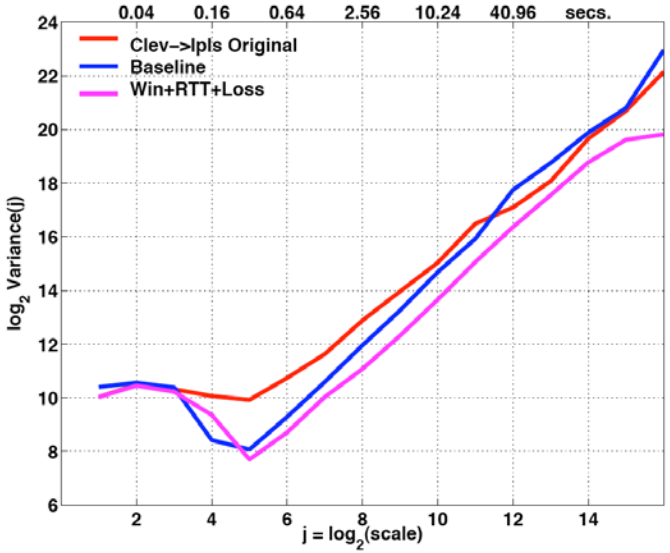
Reproduction of Active Connections

UNC replays *v.* original



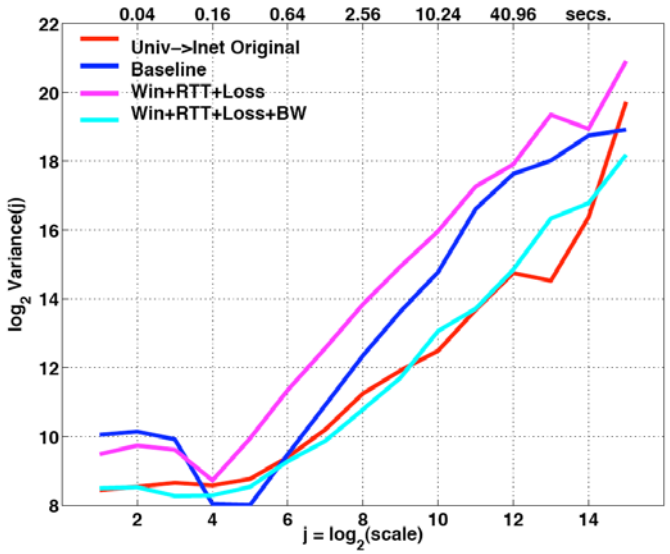
Self-Similarity & Long-Range Dependence

Wavelet spectrum — Abilene westbound



Self-Similarity & Long-Range Dependence

Wavelet spectrum — UNC outbound

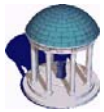


Synthetic Traffic Generation
Summary

- 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
  - With tunable degrees or realism
- Demonstrated that you can simulate the Internet in a lab
  - Realistic network experiments are possible without arbitrary traffic generation choices!

Future Work
Lots!

- Plenty more variables to understand:
  - Scaling and re-sampling paradigms
    - » How do we generate 2x Abilene traffic, or 1.125 Abilene traffic?
  - Effect of tracing duration
    - » Minutes, hours, or days?)
  - Dealing with concurrent connections
- Cluster analysis of *a-b-t* connection vectors on-going
- Still have yet to experiment with modeling UDP connections



The UNIVERSITY *of* NORTH CAROLINA
at CHAPEL HILL

How "Real" Can Synthetic Network Traffic Be?

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

Andrew Nobel
Department of Statistics

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