From Traffic Measurement to Realistic Workload Generation

Felix Hernandez-Campos

Ph. D. Candidate

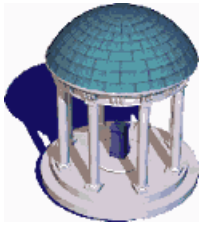
Dept. of Computer Science

Univ. of North Carolina at Chapel Hill

Joint work with

F. Donelson Smith and Kevin Jeffay

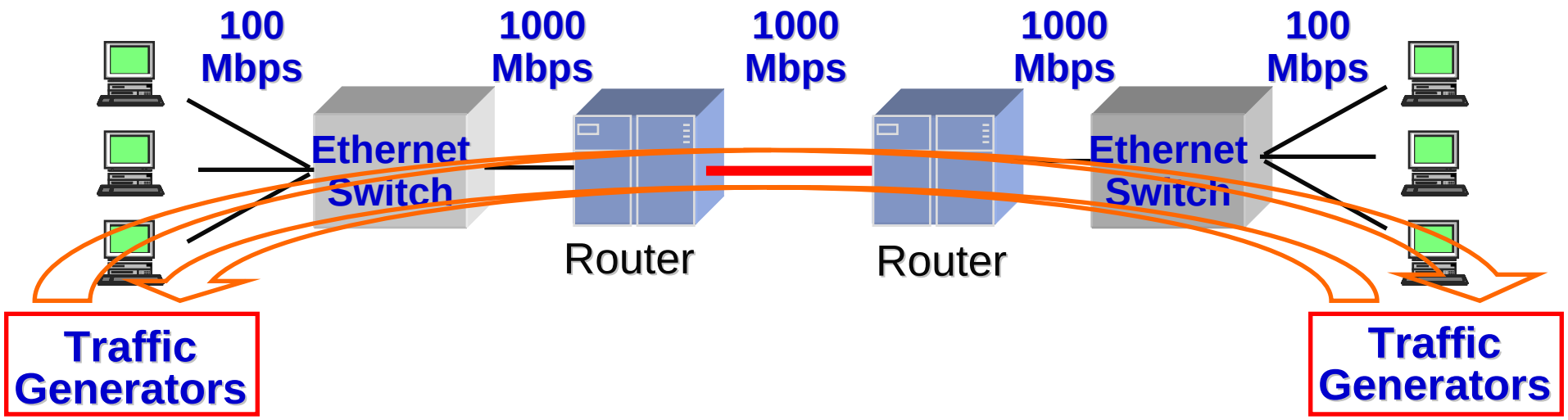
WISP November 12, 2004



Problem Formulation

- Evaluating network protocols and mechanisms requires careful experimentation
- A critical element of these experiments is the traffic workload
  - What is a *realistic* workload?
- Given a packet header trace T, extract a set of features that describes the traffic, and regenerate this traffic accordingly, collecting a new trace T^*
  - What kind of analysis will demonstrate that T and T^* are *close enough*?

Testbed Example



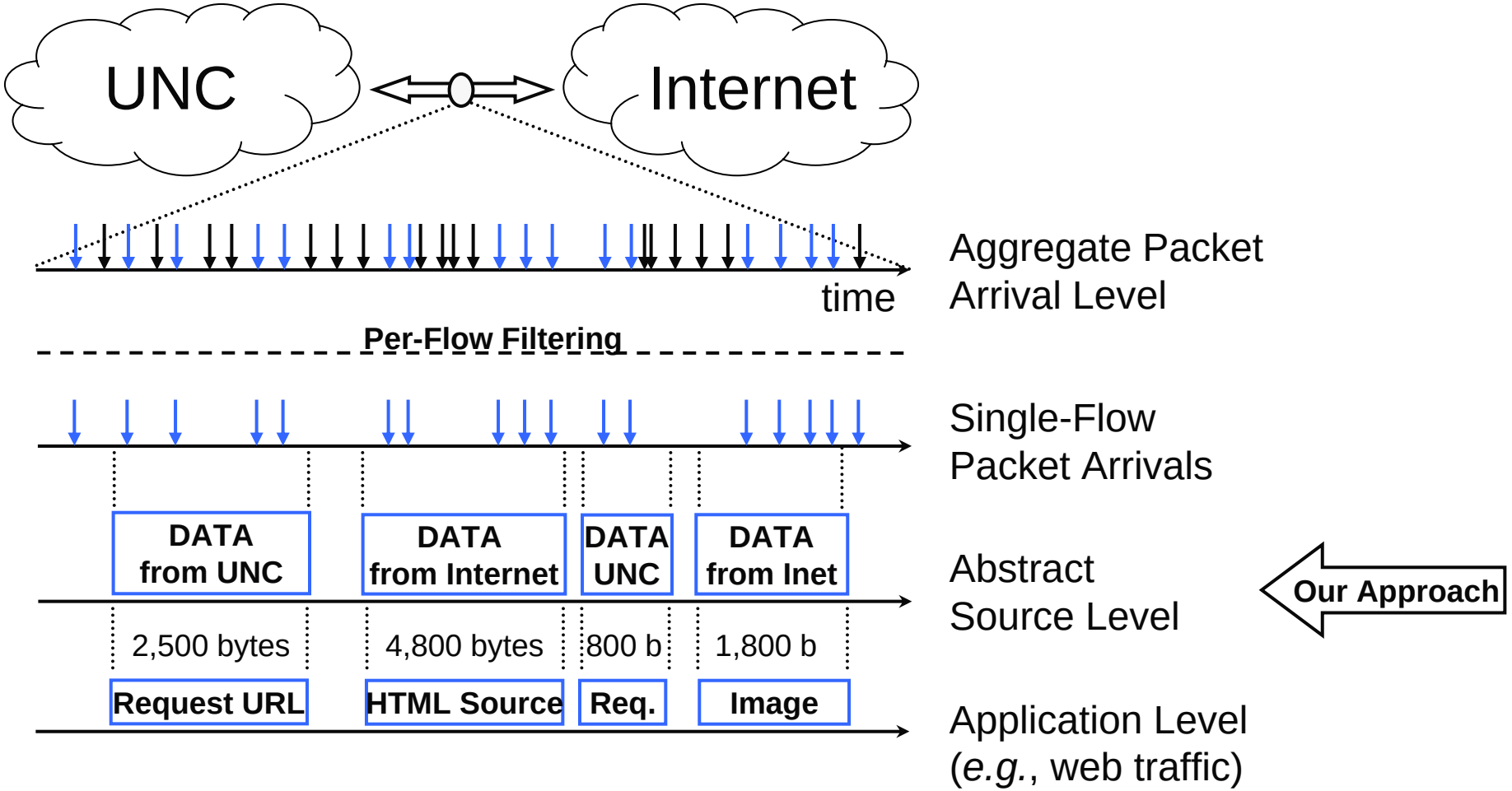
- Make generated traffic look like UNC edge link
  - Evaluate accuracy and impact of available bandwidth estimation techniques
  - Evaluate performance and impact of high-speed TCP flavors

Traffic Generation
State-of-the-Art

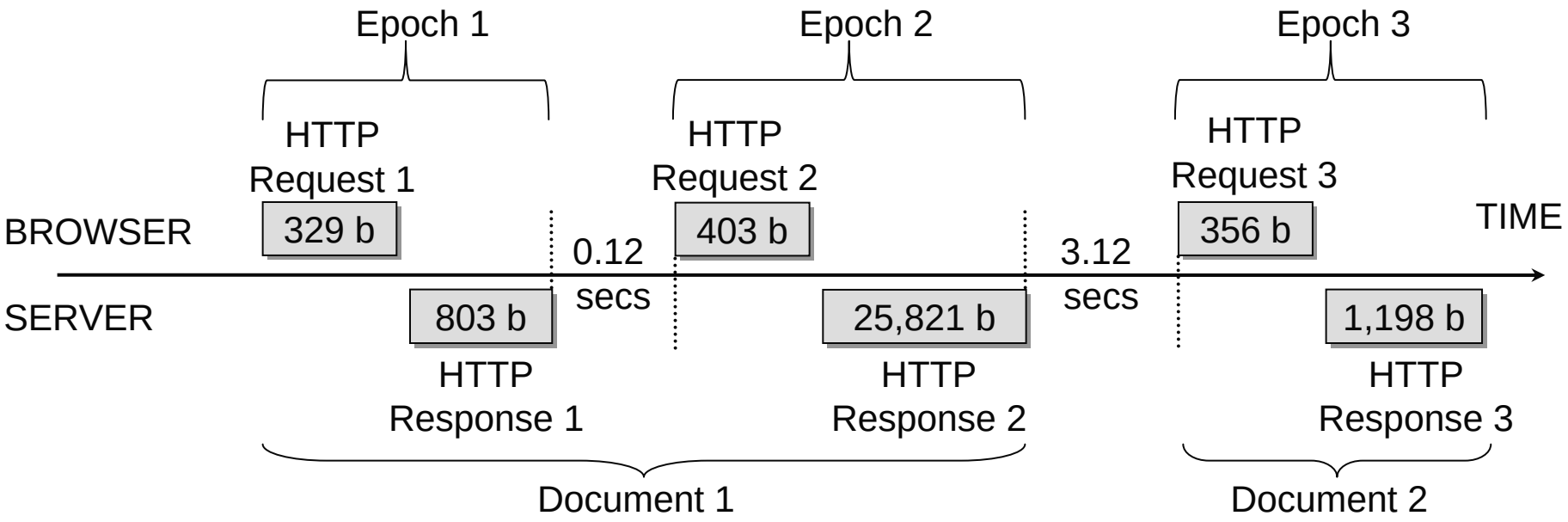
- **Open-loop**
  - Large number of sophisticated models
    - » ***Packet-level modeling***
  - But TCP is a closed-loop protocol
    - » Open-loop traffic generation breaks reliability, flow control, and congestion control
- **Closed-loop**
  - The idea is to simulate the behavior of users/applications
    - » ***Source-level modeling***
    - » $T = f(S)$ and $T^* = g(S)$ => study S, f and g

Different Views of Internet Traffic

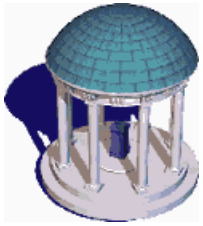
Abstract Source-level Modeling



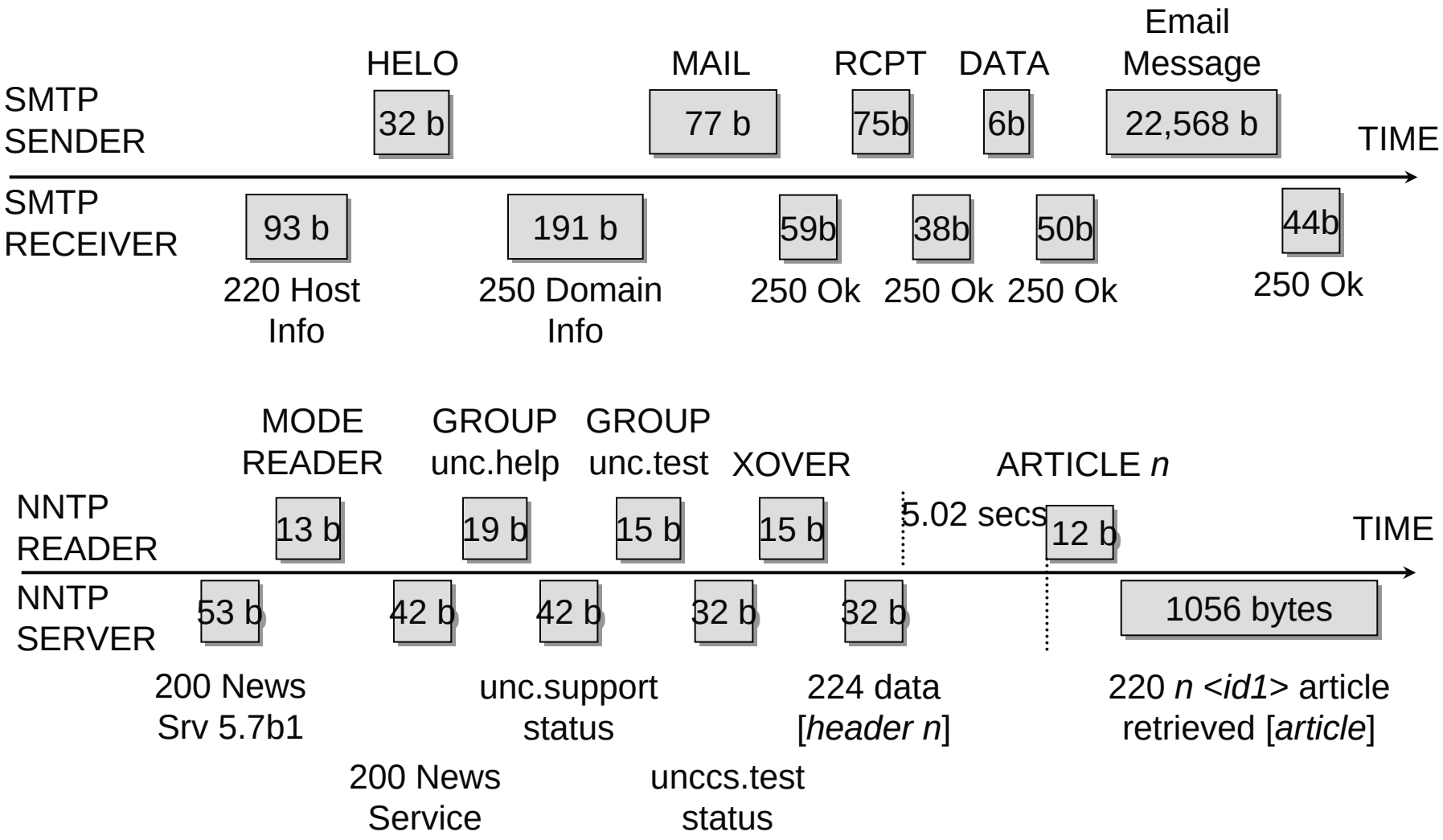
Client-Server Applications
Persistent HTTP Example

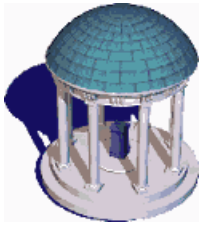


- We call pairs of ADUs that carry a request/response exchange an *epoch*
- *Quiet times* are also part of the workload of TCP



Client-Server Applications
SMTP and NNTP Examples

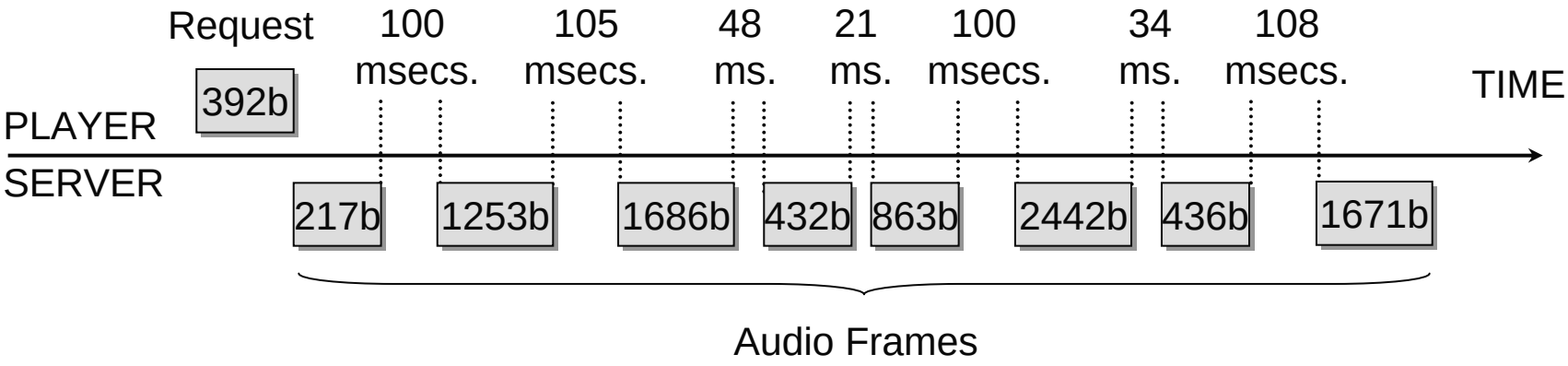




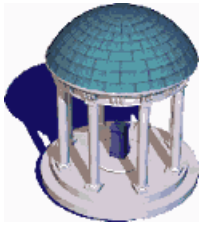
Sequential A-b-t Model

- Abstract source-level model for describing the workload of TCP connections
- Each connection is summarized using a ***connection vector*** of the form $C_i = (e_1, e_2, \ldots, e_n)$ with $n \geq 1$ epochs
  - Each epoch has the form $e_j = (a_j, ta_j, b_j, tb_j)$
- Connection vectors can be extracted from TCP segment header traces
  - Sequence number directionality, timing analysis, write size and packet size interactions
  - $O(n \log n) + O(n*W)$

Beyond the Client-Server Model
Icecast – Internet Radio

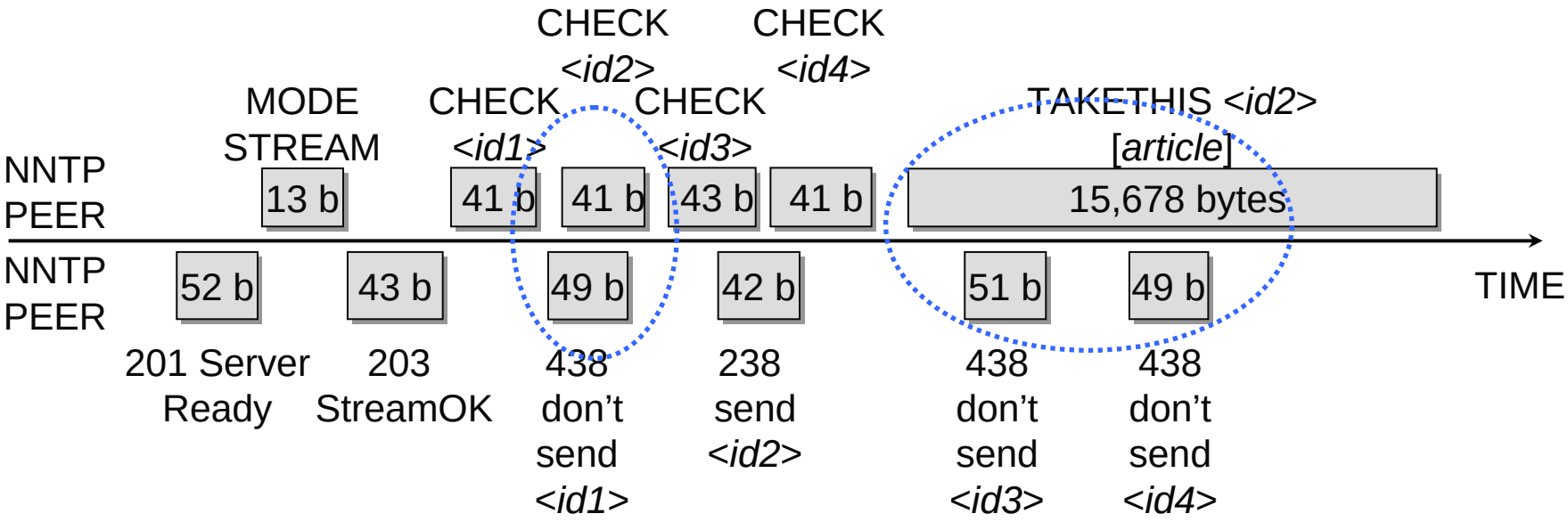


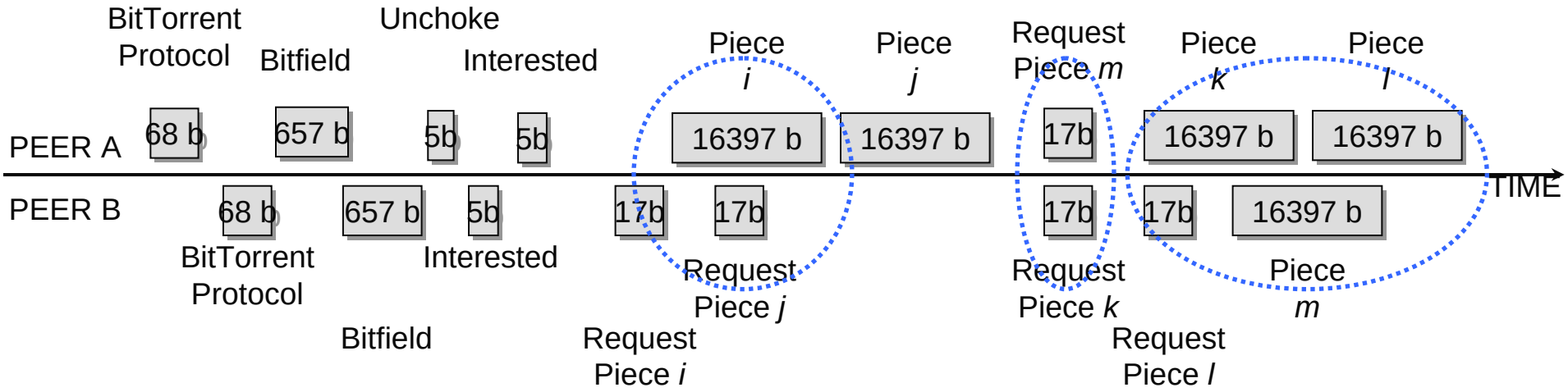
- Server PUSH applications do not follow the traditional client-server model
- The sequential a-b-t model is still applicable
  - Make a_i and tb_i zero

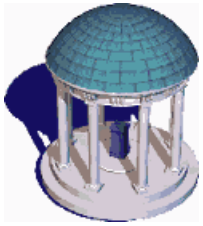


Beyond the Client-Server Model

NNTP in Stream-Mode and BitTorrent



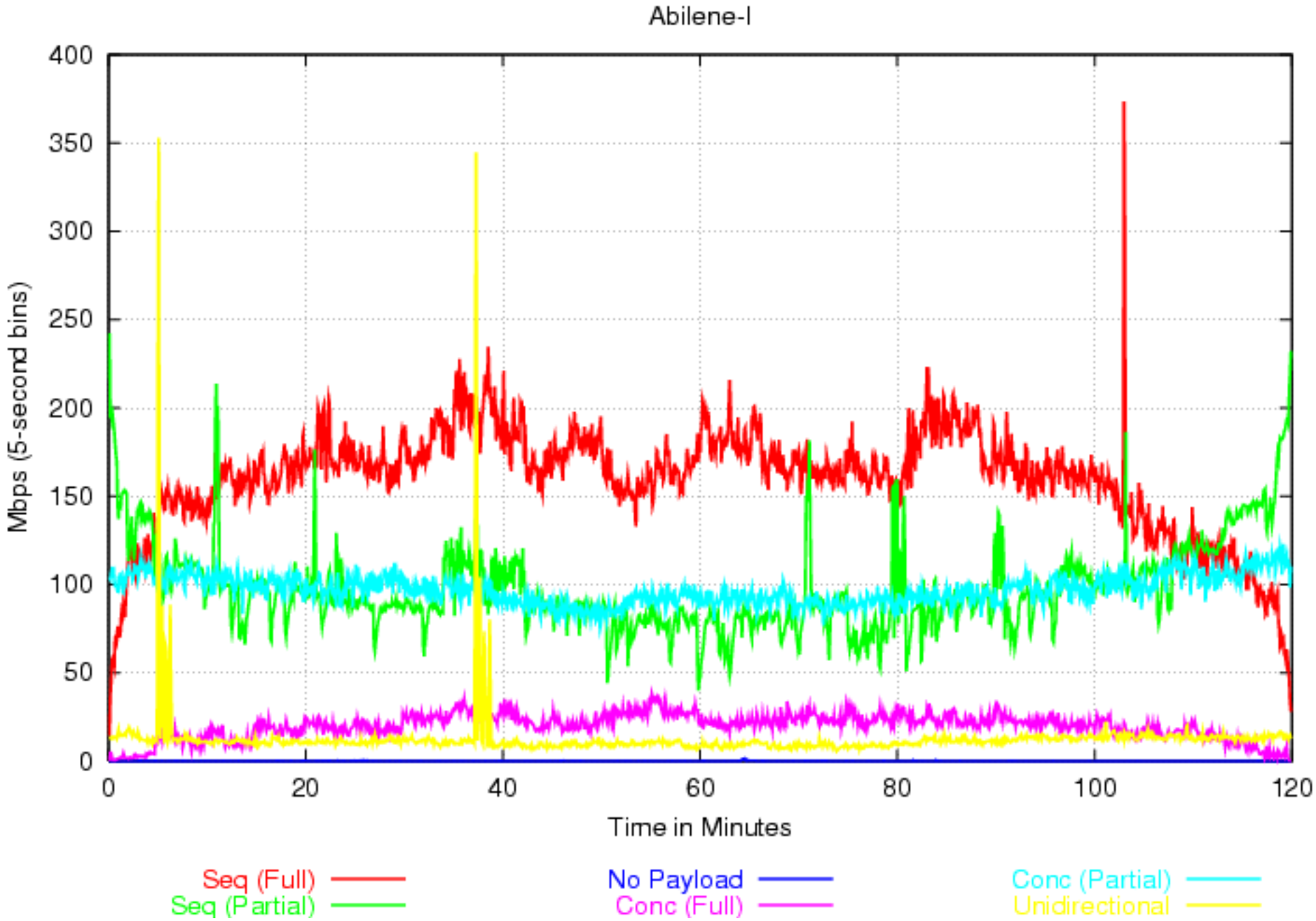


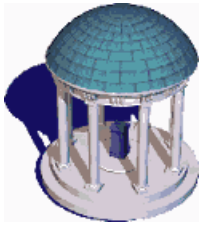


Concurrent A-b-t Model

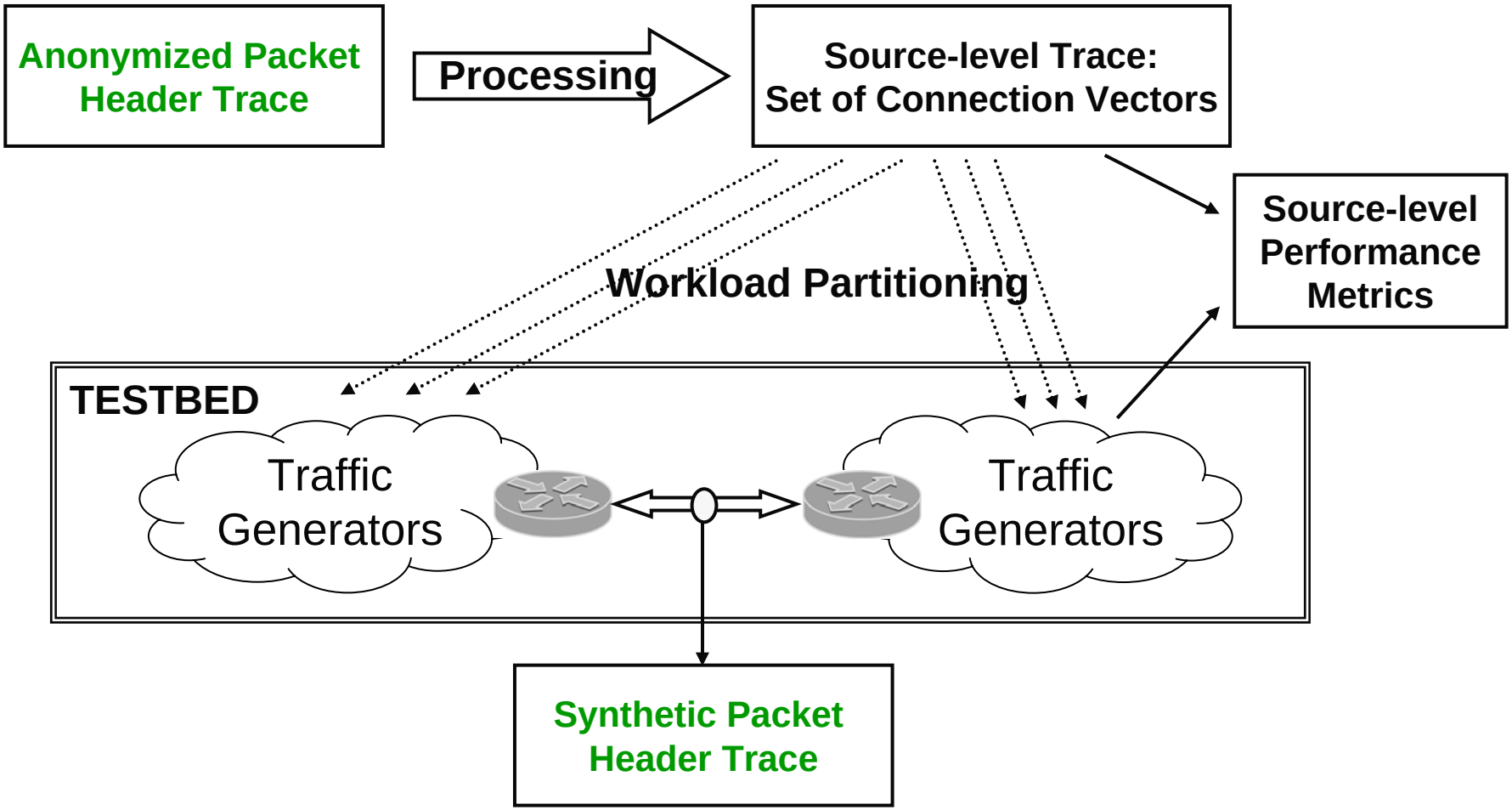
- Some connections are said to exhibit ***data exchange concurrency***
- Two reasons:
  - Increasing performance
  - Enabling natural concurrency
- Concurrent a-b-t model describes each side of the connection separately

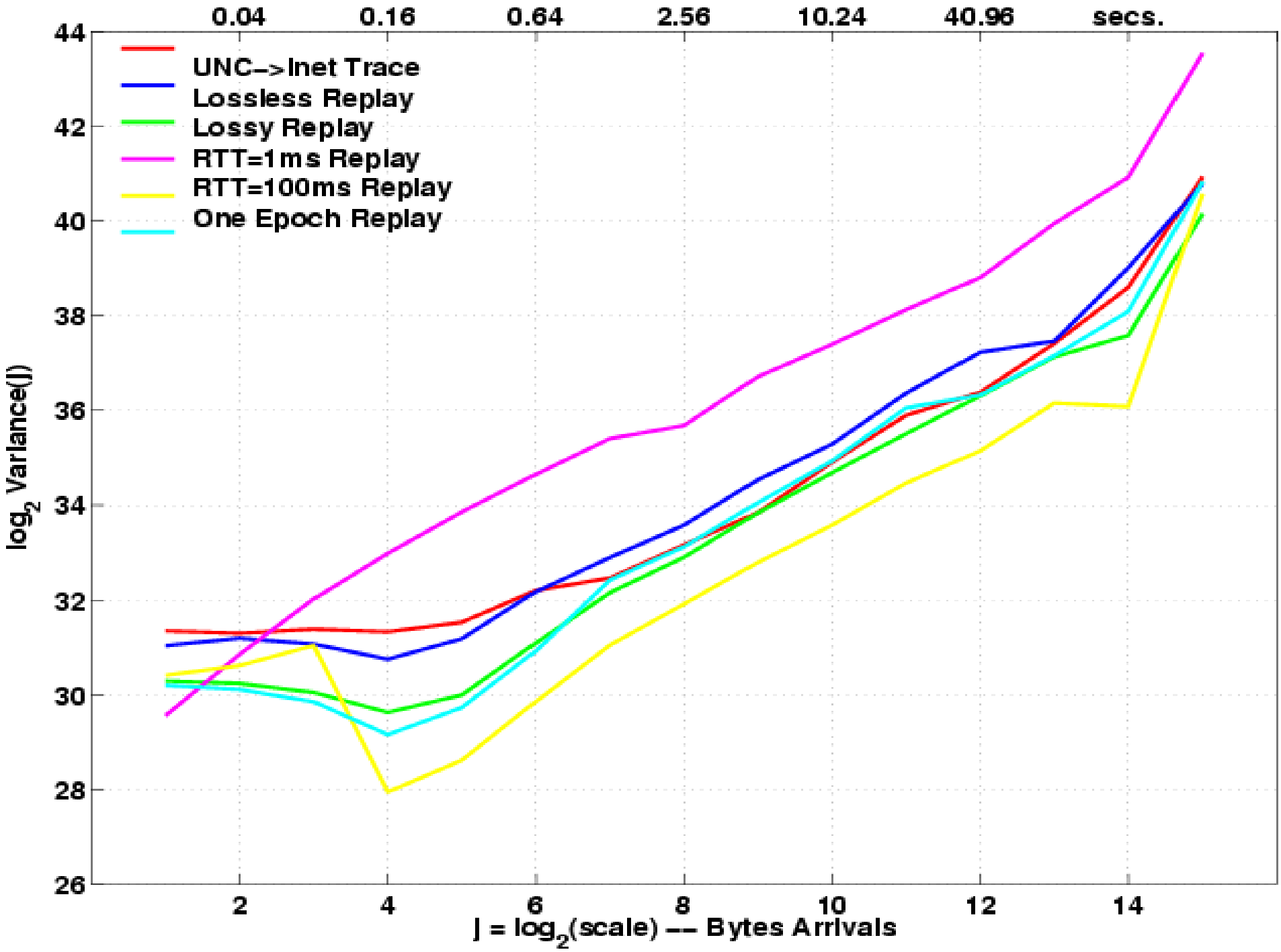
  $((a_1, ta_1), (a_2, ta_2), \ldots, (a_n, ta_n))$
  $((b_1, tb_1), (b_2, tb_2), \ldots, (b_m, tb_m))$
- Concurrency can be detected with high probability
  - *p.seqno* > *q.ackno* and *q.seqno* > *p.ackno*
  - O(*n***W*)

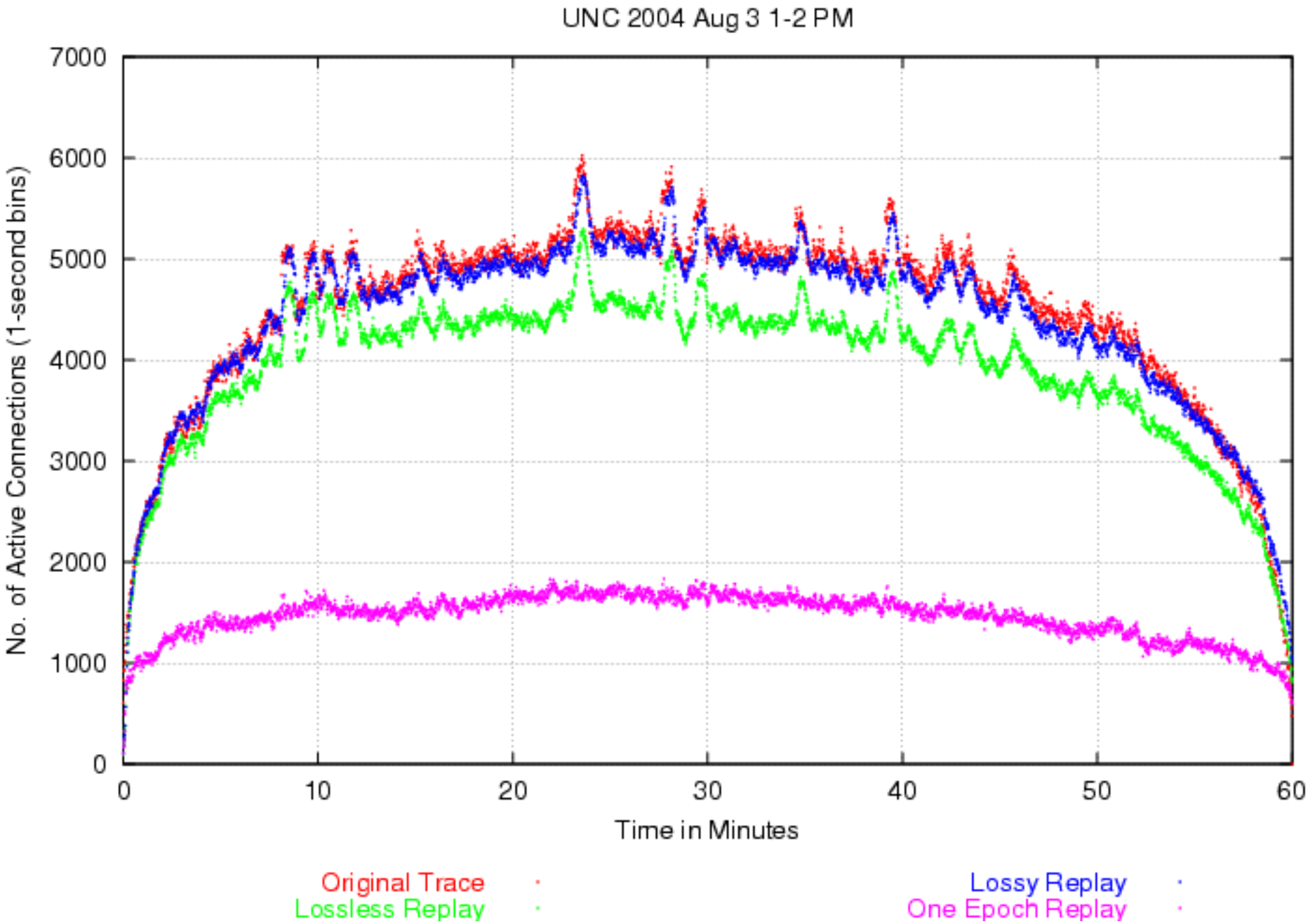

Abilene-I
Mbps (5-second bins)
400
350
300
250
200
150
100
50
0
0
20
40
60
80
100
120
Time in Minutes
Seq (Full)
Seq (Partial)
No Payload
Conc (Full)
Conc (Partial)
Unidirectional

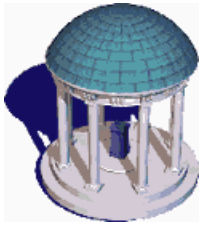


Source-Level Trace Replay
Traffic Generation in Lab Testbed




0.04
0.16
0.64
2.56
10.24
40.96
secs.
UNC->Inet Trace
Lossless Replay
Lossy Replay
RTT=1ms Replay
RTT=100ms Replay
One Epoch Replay
log2 Variance(j)
44
42
40
38
36
34
32
30
28
26
2
4
6
8
10
12
14
j = log2(scale) -- Bytes Arrivals


UNC 2004 Aug 3 1-2 PM
No. of Active Connections (1-second bins)
Time in Minutes
0
1000
2000
3000
4000
5000
6000
7000
0
10
20
30
40
50
60
Original Trace
Lossless Replay
Lossy Replay
One Epoch Replay



Conclusion and Next Steps

- New method for modeling traffic mixes
  - Empirically-derived connection vectors
  - Studied sequential vs. concurrent dichotomy
  - Fully automated, efficient analysis
- New traffic generation approach
  - Enables comparison of real and synthetic traffic
  - Implemented a distributed traffic generator
  - Techniques for scaling traffic load
- Deconstructing traffic and causality
- Traffic classification