Task 1.1
Analytic and hybrid cores for reducing simulation time

Observations about modeling tools that use analytical or hybrid techniques

1. Task statement

- There are two basic techniques for simulating networks which approach the problem from different directions
  - 1. Analytical techniques examine the network in the steady-state and use mathematical formulas employing assumptions about the network
  - 2. Discrete-event simulation examines the dynamic behavior of the network and incorporates every event in the network

- Two tasks
  - 1. Investigate the strengths and limitations of modeling tools which are based on analytic cores
  - 2. Investigate hybrid simulation techniques in which one focuses the simulation on a portion of the network of special interest, and models the remainder of the network using analytic techniques

1. Task statement (cont.)

- Problem space -- one viewpoint …
  - Network modeling usually requires components operating at the network edge (e.g., traffic generators, protocol stacks), and components operating in the network core (e.g., AQM, routing protocols)
  - Some simulations require very high fidelity in one or both component spaces
  - One can (naïvely) partition the problem space this way →

		Network Fidelity	
		Low	**High**
Edge Fidelity	**Low**	Trivial	Analytical edge, DES core
	High	DES edge, analytical core	Difficult

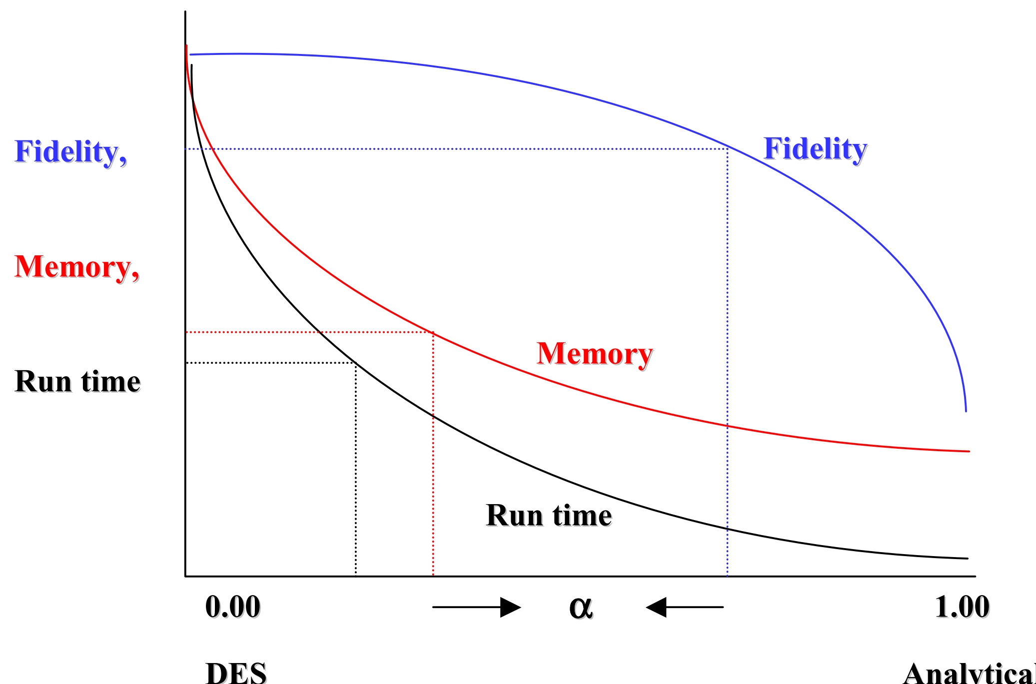
1. Task statement (cont.)

- Examples
  - (Lo, Lo): Some low-end commercial 'predictors'
  - (Lo, Hi): Test a new variant of OSPF -- network behavior is most important
  - (Hi, Lo): Test a new variant of TCP for protocol correctness -- edge behavior is most important
  - (Hi, Hi): Many interesting problems
  - Many techniques have been proposed to reduce (Hi, Hi) difficulty to manageable levels
  - Hybrid techniques may combine analytical methods and DES

		Network Fidelity	
		Low	**High**
Edge Fidelity	**Low**	Trivial	Analytical edge, DES core
	High	DES edge, analytical core	Difficult

1. Task statement (cont.)

- One approach for the (Hi, Hi) situation
  - Edge -- use DES for high fidelity; models may be simple or complex
  - Core
    - Use analytical methods (AM) for background and aggregate traffic
    - Run a simple routing algorithm (e.g., SPF) to adjust after link failures
    - Use some proportion of explicit, DES-based traffic to introduce variability and stochastic effects
  - Core = α x AM + (1- α) x DES
  - Choice of α depends on constraints
  - Choice is subjective, and perhaps adaptive



2. Analytical methods

- Rationale for using analytical methods
  - Potential for greatly reduced run times and memory requirements, higher scalability
  - Can provide adequate, first-order estimates of network behavior if carefully crafted; however, statistics may be point estimates
  - Some commercially successful tools are based on analytical cores, or on hybrid cores

- Task #1: "Investigate the strengths and limitations of modeling tools which are based on analytic cores"

- How are analytical methods implemented in commercial tools?
  - Implementations may provide insight about strengths and limitations of these methods
  - Details are usually a trade secret, but there are hints in white papers and technical reports
  - Some educated guesses follow, and we can certainly drill deeper

2. Analytical methods (cont.)

- Method 1: Jackson networks (Jackson, Kleinrock)
  - Independent servers, independent M/M/1 queues, independent queue occupancies and delays, steady state assumptions
  - External (new) arrivals are Poisson, network has random routing, packet lengths are approximately exponentially distributed
  - Merging traffic streams restores the independence of interarrival times and packet lengths (Kleinrock Independence Approximation)
  - One treats the network of queues as a single queuing system, and applies Little's Law for occupancies and delays
  - Provides good first-order approximations for moderate traffic loads
- Method 2: BCMP networks (Baskett, Chandy, Muntz, Palacios)
  - N classes of customers, class-based capacity limits
  - Random class-based routing, class-based service rates
  - Servers can be
    - Single, FIFO, exponential service time distribution
    - Single, un-weighted round robin, arbitrary service time distribution
    - Single server, preemptive LIFO, arbitrary service time distribution
    - Infinite servers, arbitrary service time distribution

2. Analytical methods (cont.)

- Method 3: Operational analysis, operational laws (Buzen, Denning)
  - A simplified form of queuing theory that is generally free of assumptions about the distributions of interarrival and service times
  - E.g., Little's Law is an operational law
  - "Operational" ⇒ "directly measured"; i.e., the method is used to derive equations from measurements (or estimates) that characterize network performance during a given period
  - Equations based on operational laws are used to estimate throughput, service time, utilization, waiting time, response time, occupancy, …
- Method 4: Generic performance models
  - Models objects -- message handlers, latency and load on clients, servers, links, and infrastructure
  - Delay - round trip, send, process, or reply - characterized by mean, mode, median, standard deviation, percentile
  - Utilization by link or server; worst case analysis based on loading levels
  - Link failure and link errors, and transport layer error rates
  - Sensitivity analysis for variations in key parameters

2. Analytical methods (cont.)

- Method 5: Kalashnikov's method
  - Self-adapting method that splits a model into analytical and DES components
  - Uses Whitt's Queuing Network Analyzer (QNA) for analytical modeling component
    - http://www.columbia.edu/~ww2040/A1b.html
  - Uses Transform Expand Sample (TES) processes as traffic generators
    - Jelenkovic and Melamed, "Automated TES modeling of compressed video", *IEEE INFOCOM'95*, pp. 746-752
  - Uses LRM (Likelihood Ratio Method) sensitivity estimates for transformation and abstraction of network models
    - http://www.jip.ru/2002/29.pdf
  - Uses piecewise linear aggregates as the formalism for DES component
    - Dzemydiene and Pranevicius, "Integration of Aggregate Approach in Knowledge Representation of the Multi-modal Transport Evaluation System", *Proc. Third Intl. Workshop on Databases and Information Systems*, 1998

2. Analytical methods (cont.)

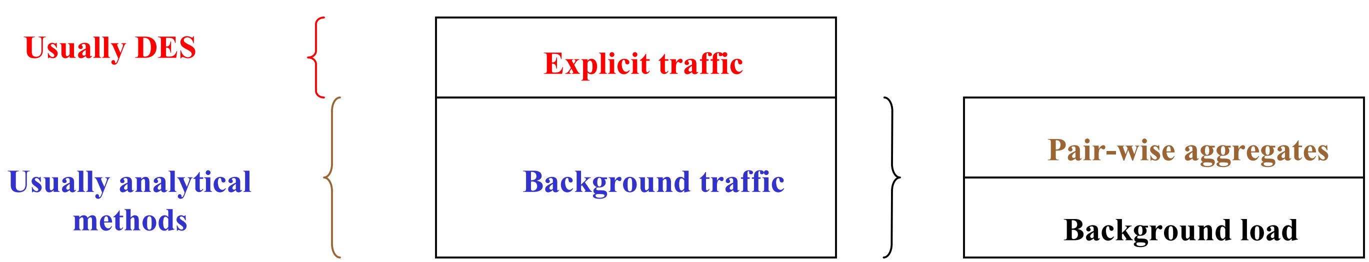
- Method 6: Trace-driven background traffic models
  - Uses packet traces collected at network exchange points
  - Partitions the aggregate traffic into sub-streams, one for each destination in the backbone network
  - Result is a background traffic load model for wide-area network simulations
    - E.g., Lucas et al., "(M, P, S) - An efficient background traffic model for wide area network simulation," *1997 IEEE Global Telecom. Conf.*

- Method 7: Traffic matrices
  - Matrices for source x destination pairs, with per-pair traffic loads in discrete time slots (e.g., 15 minute slots)
  - Often based on measured traffic traces; estimates are also used

- Method 8: Neuro-dynamic programming (Dimitri Bertsekas)
  - "Appropriate for systems that are difficult to model but easy to simulate, e.g., large data networks" (D.B.)

2. Analytical methods (cont.)

- Method 9: Hybrid methods for modeling the core network
  - 1. Background traffic
    - 1a. Background load, usually static per link
    - 1b. Aggregate flows, may be per end-to-end pairs, may change with time
    - Both may be based on imported traffic traces (e.g., RMON-2)
    - Both use counts -- packets, bits per unit time -- not individual packets
    - Both can be used to build traffic matrices
    - Both provide coarse point estimates of mean utilization and latency
  - 2. Explicit traffic
    - Usually based on DES -- packet-level detail
    - Captures (some) important protocol effects on a per-packet basis: flow control, loss, retransmissions, prioritization, window sizes, blocking, …
    - Used to introduce fine-grained variability and stochastic effects
    - Used to estimate response time, jitter, empirical distributions, min/max



3. Network emulation

- Task #2: "Investigate hybrid simulation techniques in which one focuses the simulation on a portion of the network…, and models the remainder … using analytic techniques"
- Another technique is worth investigating -- network emulation
  - 1. Core -- virtual links, arbitrary topologies, and arbitrary bandwidth, delay and error characteristics
  - 2. Edge -- Artificial traffic sources and/or realistic protocol stacks
  - References
    - "Report of NSF Workshop on Network Research Testbeds", http://www-net.cs.umass.edu/testbed_workshop/testbed_workshop_report_final.pdf, November 2002
    - Rizzo, "Dummynet: a simple approach to the evaluation of network protocols", ACM CCR, January 1997
    - White et al., "An integrated experimental environment for distributed systems and networks", *Proc. 5th Symp. Operating Systems Design and Implementation,* 2002
    - Vahdat et al., "Scalability and accuracy in a large-scale network emulator", *Proc. 5th Symp. Operating Systems Design and Implementation,* 2002

Task 1.2
Statistical Analysis Tools

Thoughts about packaging simulation output

1. Task statement

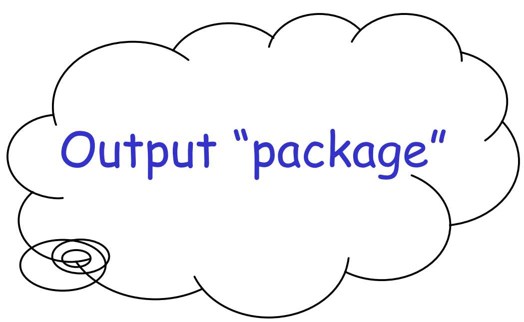
- The quality and depth of statistical analysis tools embedded in some commercial simulation products are disappointing

- This may require that simulation output be analyzed using an external tool like S-Plus or SAS or Matlab

- *An elegant solution would be to have simulation tools “package” their output for third-party tools in a standard format*

OPNET
ns-2
NetRule
...

==>



==>

S-Plus
SAS
Matlab
...

1. Task statement (cont.)

- Not an **O**(MN) problem today, but certainly an **O**(M+N) problem for many practitioners
- This is a chronic problem with no widely accepted solution
  - *"A feature that is currently of considerable interest is the ability to … export data to … other applications (e.g., an Excel spreadsheet or a database)." **
  - *"It should be possible to export individual model output observations … to other software packages such as spreadsheets, databases, statistics packages, and graphical packages for further analysis and display." **

*Law and Kelton, 3d Edition (2000)

- 'Packaging' goes beyond the (statistical) analysis of output
  - Storage, display, data management, sampling/re-sampling, pseudo-trace construction, documentation, etc.

2. Present approach

- Practitioners use 'external' tools due to
  - Poor, incomplete, or inflexible features embedded in the simulation tool
  - Personal preference; e.g., S-Plus for analysis, AutoBox for time series
  - Quality and availability of external statistical and analytical tools
    - Open source tools (e.g., R, Octave, GnuPlot)
    - Powerful commercial tools with attractive academic discounts (e.g., SAS)

- Capturing output can be unwieldy
  - Some simulation tools use proprietary formats for exporting data
  - Some require a commercial product (e.g., Excel) or conversion
  - Some may be constrained (e.g., spreadsheet format with size limits)
  - Extracting data from simulation plots and displays can require significant manual effort and great care
  - Output/export usually lacks metadata descriptions and rich data formats
    - E.g., comma-delimited ASCII with no description of fields
    - Limited data types, no way to indicate missing values, etc.

3. Possible approach

- Define a format for simulation output
  - Run simulation, generate output “package”, pass to external tool
  - Format should be standard, open, configurable, and limited only by the capacity of the storage device
  - Want to optionally capture raw output, data in plots and displays, event queue snapshots (perhaps), etc.

- Must be amenable to processing by a scripting language or tool

- Format consists of two components
  - Schema (metadata, descriptor, …)
    - Editable; could be something like an SQL CREATE TABLE
    - Could be based on open source variant like MySQL or PostgreSQL, or XML constructs, etc.
  - Data
    - Standard, importable format consistent with relational (tabular) model
    - Unlimited number of rows, very large number of columns

3. Possible approach (cont.)

- The simulation tool generates a schema and data; e.g.,
  - Schema

```
CREATE TABLE QueueSize    (
        ID type(n),
        Timestamp type(n),
        QueueName type(n),
        QueueLength type(n),
        . . .             ) ;
```

    - Editable
    - Amenable to processing by a scripting language or tool

  - Data
    - Standard importable format (e.g., .csv) for wide compatibility
    - Relational model with rich data types
    - Support for missing values, -∞, +∞, and other special indications that are important to statistical tools; there may be standards for these
    - Optional sequence numbers to guarantee order and completeness, and checksums or other 'signatures' to guarantee record and file integrity
    - Amenable to processing by a scripting language or tool

4. Issues

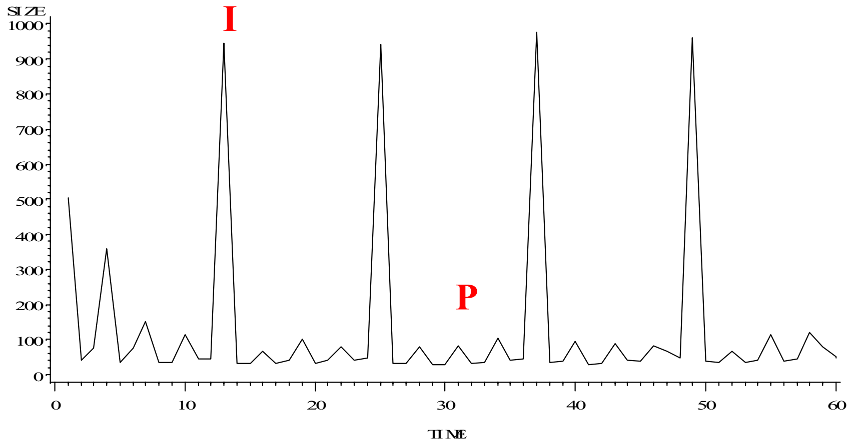
- Has this been done?
  - A cursory search found nothing; this has almost certainly been proposed but has not been (widely) accepted
- Will it succeed?
  - Endorsement by an EU COST Action carries considerable weight
  - Vendors may resist, as proprietary formats lock you into their tools
  - More likely to be embraced by the open source community
- What difference will it make?
  - Many advantages -- configurable, standard, open, limited only by the capacity of the storage device, amenable to scripting, etc.
- What resources might be required?
  - An undergraduate project to develop some ideas, or …
  - A Master’s project or thesis to develop a detailed framework and a non-trivial open source implementation

Task 1.3
Multimedia Traffic

Thoughts about multimedia traffic models at different time scales

1. Task statement

- Multimedia traffic studies carried out in COST Action 256 and elsewhere indicate that protocols behave differently at different time scales
- A study will be undertaken of the network behavior in different time scales, e.g., 1, 10, 100 second intervals, including the method of validation of simulation results
- Some multimedia traffic is semi-regular as it leaves the source; e.g., MPEG-4 segment with (IBB PBB PBB PBB) structure
- Traffic generators and protocol behaviors almost certainly differ at different time scales
- E.g., if we sum over every 12 frames then we are likely to lose the 'seasonal' effects



1. Task statement (cont.)

- Many types of multimedia traffic have characteristic statistical signatures induced by their encoders
- There is a significant base of research in this area
  - E.g., circa 1988 -- Verbiest, Nomura, Sen, Maglaris, many others
  - E.g., circa 1991 -- Grünenfelder, Manthorpe, Heyman, many others
  - E.g., 1995-present -- non-linear threshold AR models, fractional ARIMAs, DARs, GBARs, nested ARs, various LRD models, MMRPs, $M/G/\infty$ models, fluid flow models, TES, ARTA, … (**many** others)
  - Liu (ICC '01) -- technique based on decomposition/recombination for modeling MPEG autocorrelation functions
- Many of these models and techniques are valid; it is not clear which might be best for building realistic traffic generators
- We might be able to use a 'characteristic signatures' approach to build a representative set of parametric traffic models for standard multimedia types

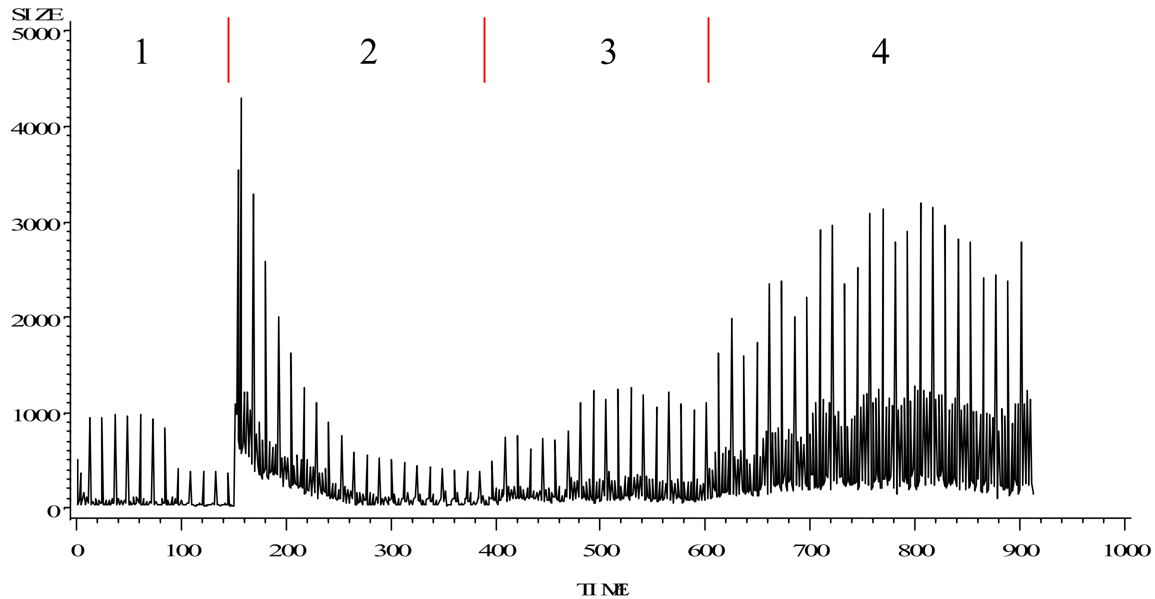
2. Possible approaches

- 1. Empirical approach
  - There are a number of collections of frame traces
  - E.g., TU Berlin has a library of frame size traces of long MPEG-4 and H.263 encoded videos
  - Trace used in the examples that follow is
    - Star Wars IV, MPEG-4, low quality frame trace
    - GOP structure is (IBB PBB PBB PBB)
    - URL is http://www-tkn.ee.tu-berlin.de/research/trace/pics/FrameTrace/mp4/

- 2. Theoretical approach
  - What happens when you aggregate to coarser time scales?
  - Is there theory for temporal aggregation for a particular model family?
  - Amemiya & Wu, Brewer, Granger & Morris, and others developed some theory for temporal aggregation of ARMA models in early 1970s
  - Can this be extended to ARIMA models, SARIMAs, fractional ARIMAs, etc. that are more typical of encoded multimedia traffic?

3. Empirical approach example

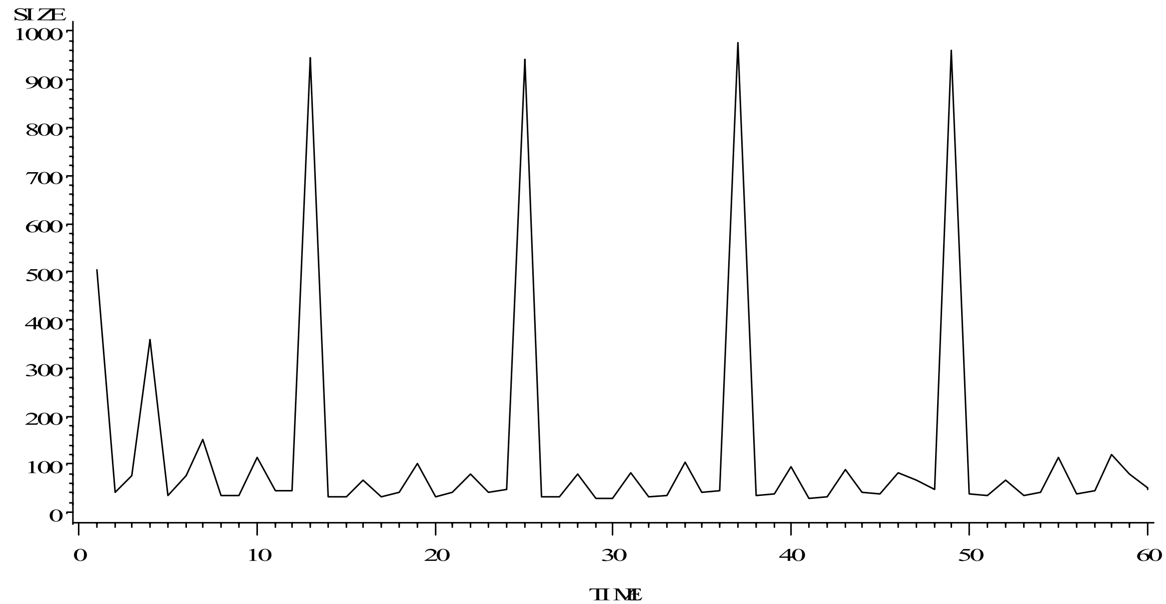
- Examine a 30 second subset of the trace (frame size × time)
- Four distinctly different patterns



- Non-stationary in mean and variance, with seasonal components induced by the encoder
  - The trace can be modeled in many ways
  - E.g., a multiplicative seasonal autoregressive integrated moving average process (S-ARIMA × S-ARIMA) is one candidate
  - Transformations (e.g., differencing, seasonal differencing, logarithmic, Box-Cox) may be required in order to analyze the trace

3. Empirical approach example (cont.)

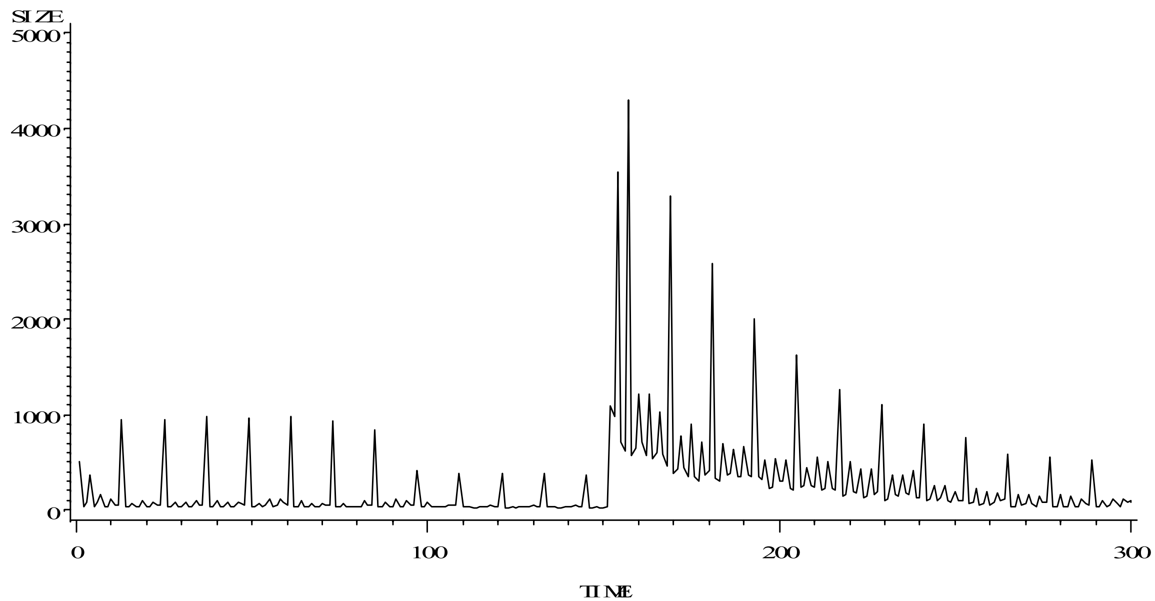
- 2-second segment
- Very regular pattern
  IBB PBB PBB PBB
  (frame size × time)
- $S(t)$ = size at time t
- $e(t)$ = actual $S(t)$ – predicted $S(t)$
- $e(i) \sim$ i.i.d.$(0,\sigma(e))$



- A reasonable model (based on conditional least squares) is:
  - Model: $(1 - B^{12}) \times S(t) = (1 - 0.697\, B^3) \times e(t) \quad \equiv$
    $S(t) - S(t-12) = e(t) - 0.697\, e(t-3)$
  - Seasonal differencing at lag 12, moving average term at lag 3; this captures the IBB PBB PBB PBB structure
  - Non-significant mean

3. Empirical approach example (cont.)

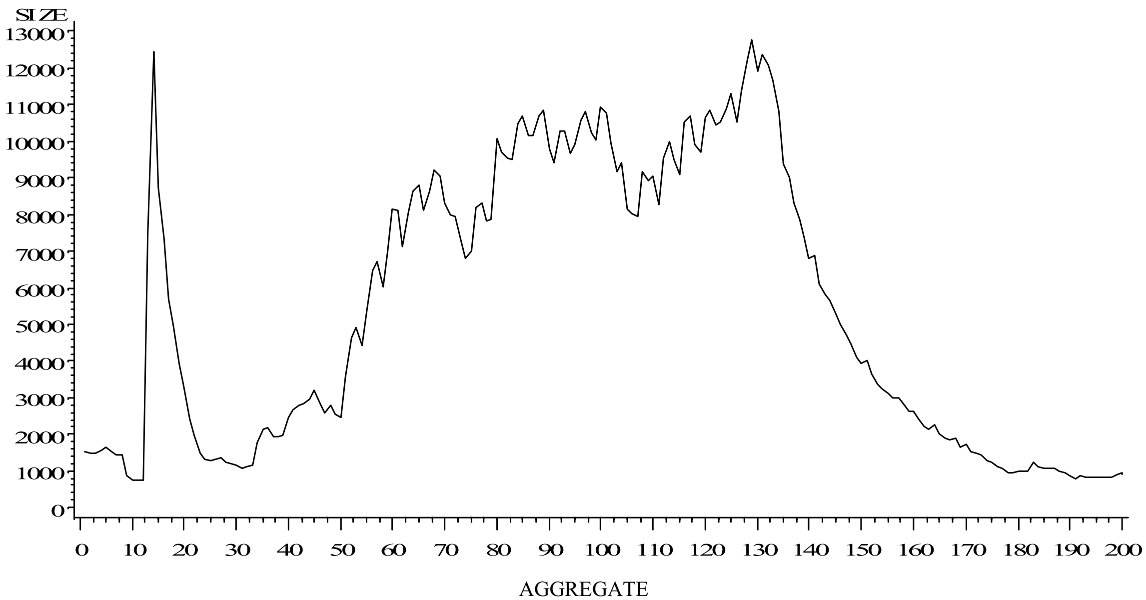
- 10-second segment
- Two patterns (frame size $\times$ time)
- Change near N = 190



- A reasonable model (conditional least squares) is:
  - LS(t) = log(size) at time t; variance is non-stationary
  - Model: $(1 - B^1)(1 - B^{12}) \times LS(t) =$
    $(1 - 0.266\, B^1)(1 - 0.100\, B^3)(1 - 0.548\, B^{12}) \times e(t)$
  - Seasonal differencing, moving average terms, non-significant mean
  - Two distinct processes representing two distinct 'scenes'
  - Second process has interesting exponential-like decay

3. Empirical approach example (cont.)

- 80-second segment, aggregated
- 12 frames → 1 value (Σ frame sizes × time)
- Quite different when aggregated



- A reasonable model (conditional least squares) is:
  - Differencing required as mean is non-stationary; D(t) = S(t) - S(t-1)
  - Model: $(1 - B^1) \times S(t) =$
    $(1 + 0.331\ B^1)(1 - 0.191\ B^2)(1 - 0.181\ B^3) \times e(t)$
  - Third-order moving average on difference D(t)
  - Non-significant mean

4. Steps

- Analyze a number of multimedia traces
  - Different multimedia types, encodings, time scales
  - Characterize them statistically and tabulate the characterizations; ARIMA models are one approach; others may be better
  -

Multimedia type	Encoding	Timescale	Model
Video	MPEG-4	0.033 sec	SARIMA(s,p,d,q)
Video	MPEG-4	0.400 sec	IMA(0,1,q)

….

- Build realistic, parametric traffic generators for these models
  - A few tens of lines of per model may be sufficient
  - E.g., http://statistics.okstate.edu/bilder/stat5053/schedule/2.9.doc contains S-Plus code for SARIMA models
  - Code for a family of traffic models would be a valuable contribution
- Is LRD behavior present? If so, models must also capture LRD.
- Is there theory for temporal aggregation (comparable to Amemiya & Wu, Brewer, Granger & Morris results for ARMA models)?
- Resources -- requires a student with background in statistics

Task 2.2
Multi-layer Traffic Modeling/Models

A mini-testbed approach to multi-layer traffic modeling

1. Task statement

- Some recently proposed models for TCP and their potential in addressing multi-layer traffic modeling will be investigated
- Some focus on window sizing and network issues separately but not independently, and when they can be coupled with a source level-model the outcome might be a valuable step toward multi-layer modeling
- Work by Chadi Barakat and colleagues
  - http://www-sop.inria.fr/planete/personnel/Chadi.Barakat/
  - Work on modeling TCP at the packet, session and system levels
- Work by Jitendra Padhye, Sally Floyd, Jim Roberts, etc.
  - Padhye et al., “Modeling TCP throughput: a simple model and its empirical validation”, Proc. SIGCOMM ‘98
  - Padhye and Floyd, “On inferring TCP behavior”, Proc. SIGCOMM ‘01
  - Jim Roberts (http://perso.rd.francetelecom.fr/roberts/Publications.html)

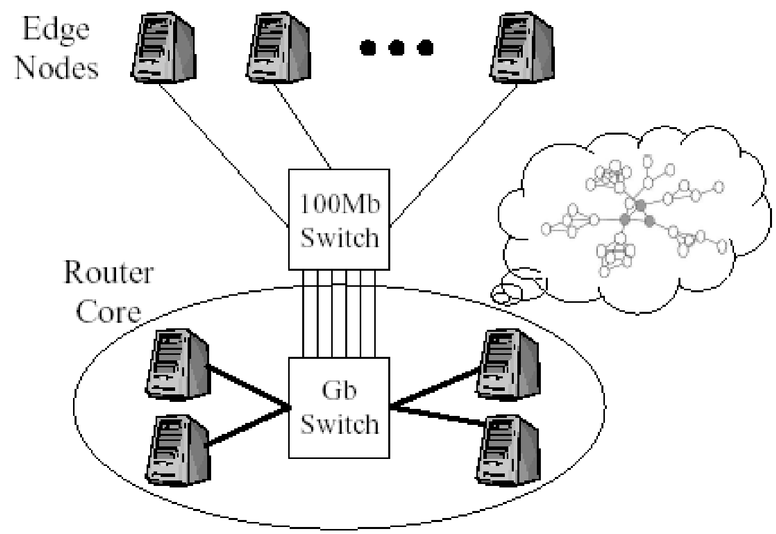
1. Task statement (cont.)

- It would be interesting to combine good source and lower-layer models into two-layer or three-layer traffic models and to begin studying the convolutional and confounding effects
- It is a very difficult problem; you have a mix of characteristics …
  - Traffic sources (greedy, bursty, periodic, non-stationary, auto-correlated, LRD, stochastic, stream *vs.* elastic, multiplexing, …)
  - Control loops (many TCP variants, TCP dynamics, RTTs, retransmissions, effects of nested control loops as with TCP over ATM ABR, …)
  - Network elements (queuing, blocking, AQM, class or priority, …)
  - Cross-layer protocols (mobile and wireless, λ routing and IP routing in optical networks, …)
  - Bandwidth and other asymmetries, session lengths and distributions, …
  - Applications (terabyte and petabyte file transfers, HTTP sessions, …)
- … and an enormous number of potential convolutional, confounding, and interaction effects; some may be influential
- Some elegant solutions for specific problems, but we need insight

2. Network emulation

- A technique worth investigating is network emulation
  - 1. Core -- virtual links, arbitrary topologies, and arbitrary bandwidth, delay and error characteristics
  - 2. Edge -- Artificial traffic sources and/or realistic protocol stacks
  - References
    - "Report of NSF Workshop on Network Research Testbeds", http://www-net.cs.umass.edu/testbed_workshop/testbed_workshop_report_final.pdf, November 2002
    - Rizzo, "Dummynet: a simple approach to the evaluation of network protocols", ACM CCR, January 1997
    - White et al., "An integrated experimental environment for distributed systems and networks", *Proc. 5th Symp. Operating Systems Design and Implementation,* 2002
    - Vahdat et al., "Scalability and accuracy in a large-scale network emulator", *Proc. 5th Symp. Operating Systems Design and Implementation,* 2002
- A semi-empirical tool with great value for insight and validation; "Internet in a can" perspective

2. Network emulation (cont.)



From Vahdat et al., 2002

- Edge nodes running unmodified protocol stacks are physically interconnected by fast links
  - Can be commodity single-board computers or multi-tasking hosts
  - Run applications, OS, protocols
- Simulated core network routes packets, and subjects traffic to bandwidth, congestion, latency, loss, error, etc.
  - Emulates packet's end-to-end path across a specific network topology
  - A single core node emulates about 70K-120K packets/second [Vahdat]
  - Some emulators scale linearly with additional core nodes
  - Captures queuing, cross traffic, latency, jitter, loss, congestion, failure
- Supports thousands of edge-node instances and gigabits of bandwidth

3. Tradeoffs

- Advantages
  - Edges -- scalable and ultra-realistic as edge nodes run actual unmodified applications, OS, and protocol stacks
  - Can run multiple processes on the same node; feasible to run tens of instances of an application edge node on one physical device
  - Core -- scalable; provides realistic hop-by-hop emulation
  - Network topologies are specified by the user
  - Emulations are per-packet, and run in (near) real time and at (nearly) the same rate as in the modeled network
  - Long run times can be accommodated
  - Results are 100% reproducible; this is important for diagnoses
  - At least one implementation of an emulator may be available to academic institutions
- Issues
  - Cost
  - Footprint
  - Quasi-cluster architecture, concurrency
  - Availability

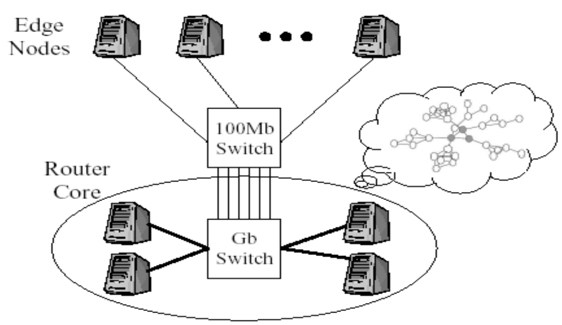
4. Relevance to Task 1.1

- Task 1.1 Statement #2: "Investigate hybrid simulation techniques in which one focuses the simulation on a portion of the network…, and models the remainder … using analytic techniques"

- May be a valuable adjunct to several of our tasks

- Provides insight and validation to very difficult problems in (near) real time; e.g., cross-layer and multi-layer protocol analysis, development and testing

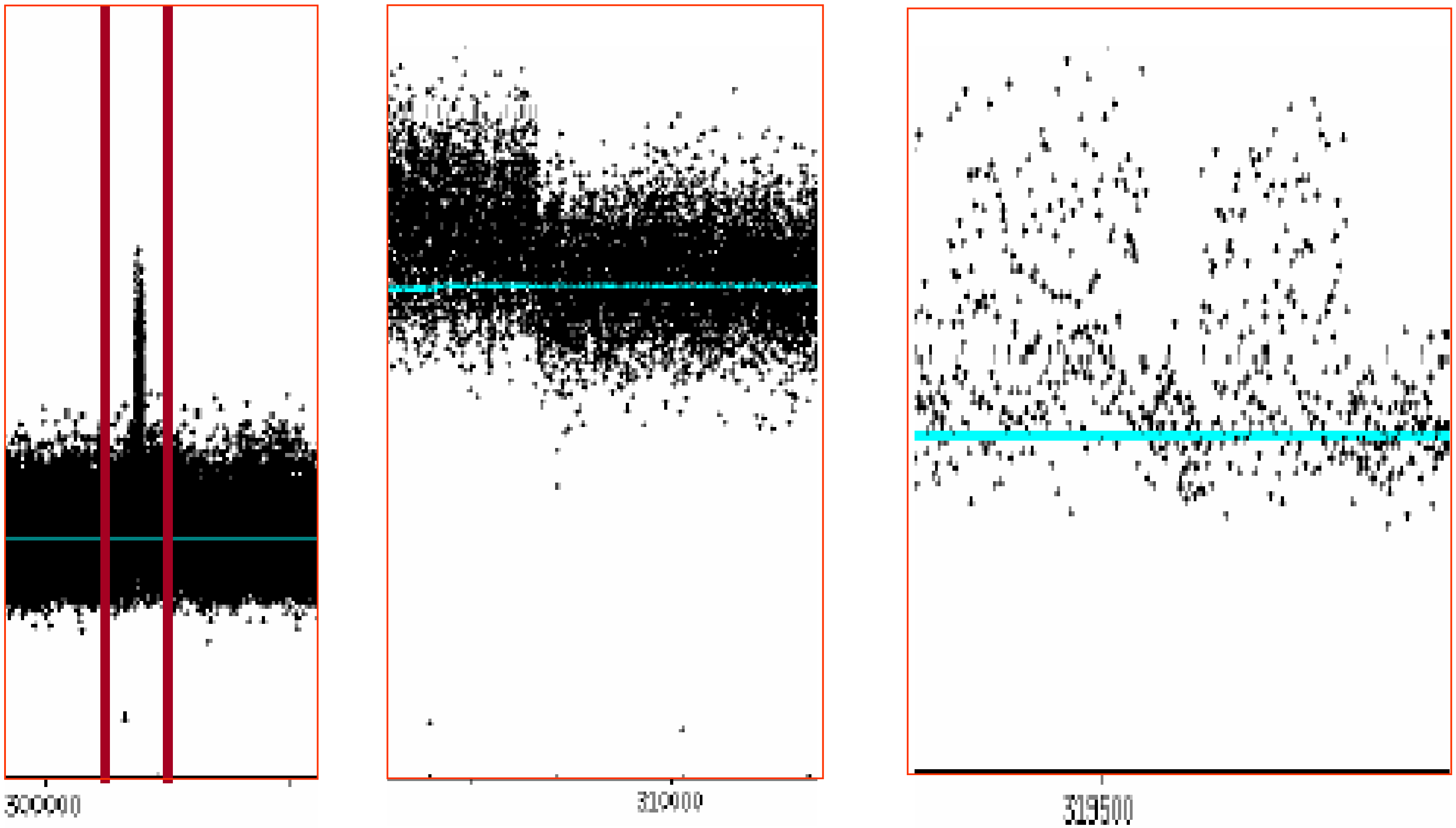
- May be amenable to grid computing



From Vahdat et al., 2002

4. Relevance to Task 1.1 (cont.)

- Long run times may generate realistic abnormal events usually not found in DES; e.g.,



Data from traces and presentation by Prof. Don Smith, UNC-CH, USA