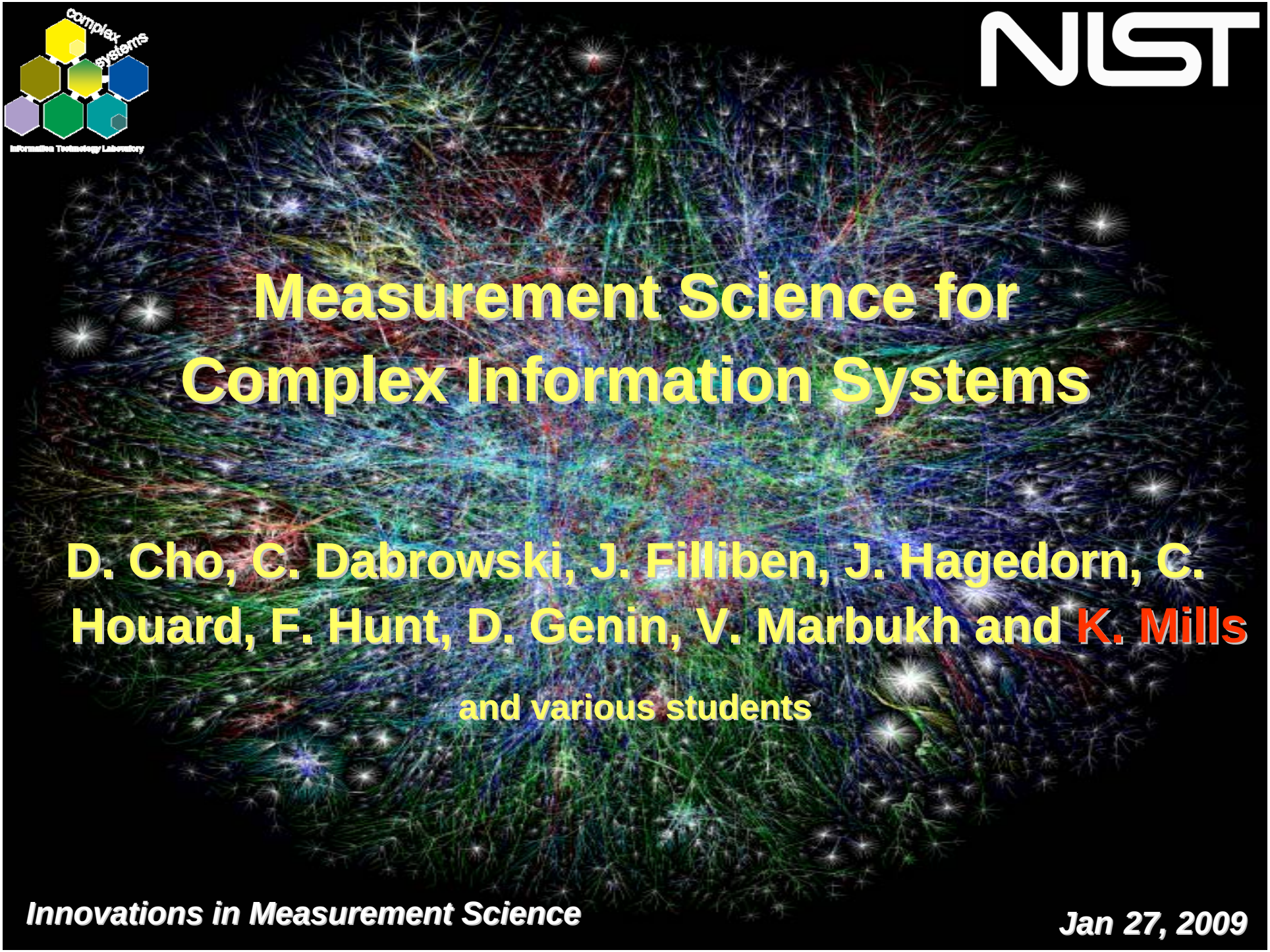
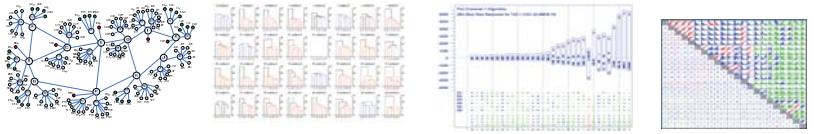

complex systems
Information Technology Laboratory
NIST
Measurement Science for
Complex Information Systems
D. Cho, C. Dabrowski, J. Filliben, J. Hagedorn, C. Houard, F. Hunt, D. Genin, V. Marbukh and K. Mills
and various students
Innovations in Measurement Science
Jan 27, 2009



Plan for Presentation

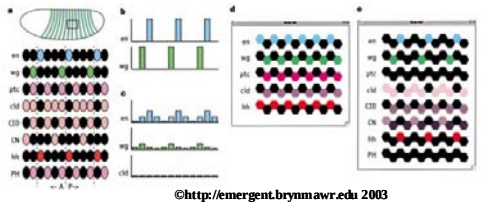
- Introduce NIST project to develop Measurement Science for Complex Information Systems

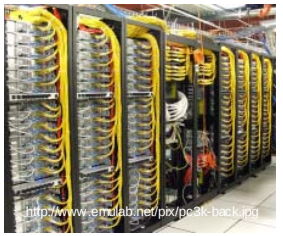
- Show an application of measurement science to compare seven alternate congestion-control algorithms for the Internet

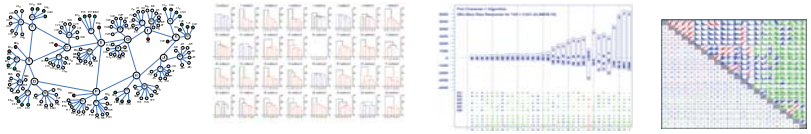
What are complex systems?

Large collections of interconnected components whose interactions lead to macroscopic behaviors

- Biological systems (e.g., slime molds, ant colonies, embryos)
- Physical systems (e.g., earthquakes, avalanches, forest fires)
- Social systems (e.g., transportation networks, cities, economies)
- Information systems (e.g., Internet, Web services, compute grids)





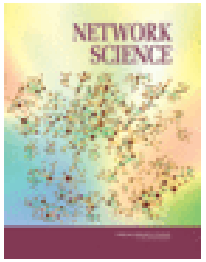


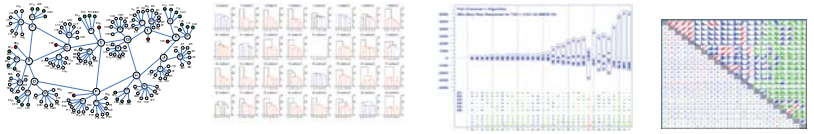
What is the problem?

No one understands how to measure, predict or control macroscopic behavior in complex information systems

"[Despite] society's profound dependence on networks, fundamental knowledge about them is primitive. [G]lobal communication … networks have quite advanced technological implementations but their behavior under stress still cannot be predicted reliably…. There is no science today that offers the fundamental knowledge necessary to design large complex networks [so] that their behaviors can be predicted prior to building them."

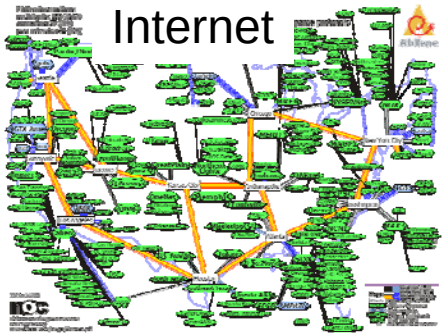
— Network Science, NRC report released in 2006





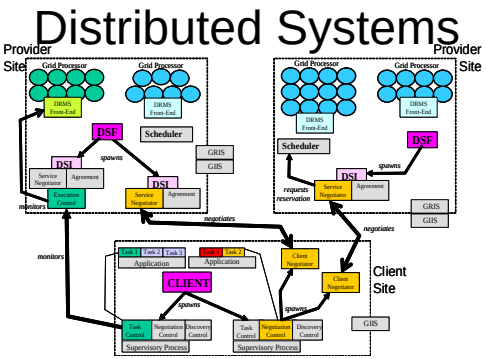
What is the new idea?

Leverage models and mathematics from the physical sciences to define a systematic method to measure, understand and control macroscopic behavior in the Internet and distributed software systems built on the Internet

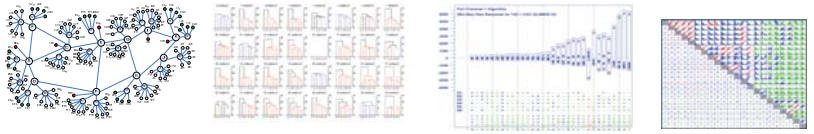


Technical Approach

- Evaluate models and analysis methods
  - Computationally tractable?
  - Reveal macroscopic behavior?
  - Establish causality?
- Evaluate distributed control techniques
  - Can economic mechanisms elicit desired behaviors?
  - Can biological mechanisms organize elements?

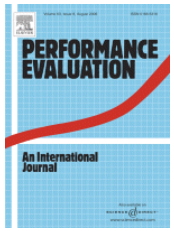


INFORMATION TECHNOLOGY LABORATORY NIST



Previous NIST Groundwork (2000-2005)

Preliminary investigation to identify hard technical issues



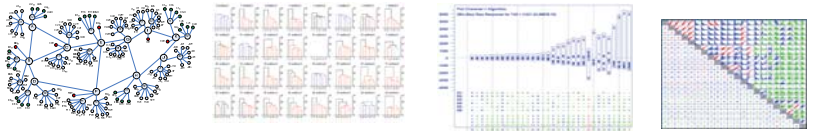
Yuan and Mills, *A Cross-Correlation-based Method for Spatial-Temporal Traffic Analysis*, July 2005

IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING

Yuan and Mills, *Monitoring the Macroscopic Effects of Distributed Denial of Service (DDoS) Flooding Attacks*, October 2005

Yuan and Mills, *Simulating Timescale Dynamics of Network Traffic Using Homogeneous Modeling*, May-June 2006

Complex Dynamics in Communications Networks, December 2005
(including *Macroscopic Dynamics in Large-Scale Data Networks* by Yuan and Mills)



Why is this hard? Why can we succeed?

Hard Issues	Plausible Approaches
H1. Model scale	A1. Scale-reduction techniques
H2. Model validation	A2. Sensitivity analysis & key comparisons
H3. Tractable analysis	A3. Cluster analysis and statistical analyses
H4. Causal analysis	A4. Evaluate analysis techniques
H5. Controlling behavior	A5. Evaluate distributed control regimes

Project Start Date: October 2006

Multidisciplinary Project

Disciplinary Expertise

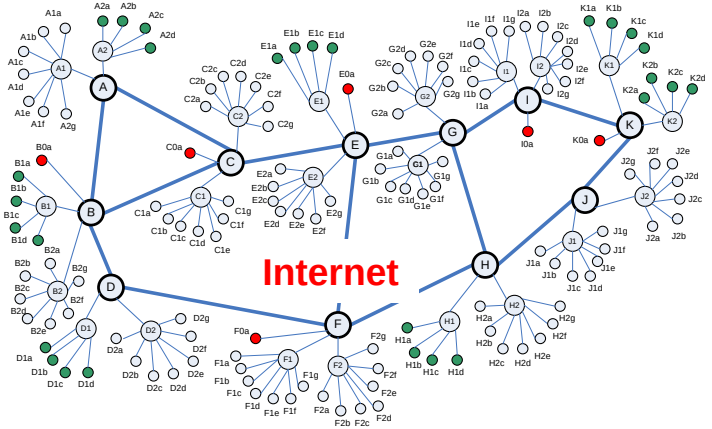
Modeling Methods	
Analytical	Simulation
D. Genin	C. Dabrowski
F. Hunt	K. Mills
V. Marbukh	

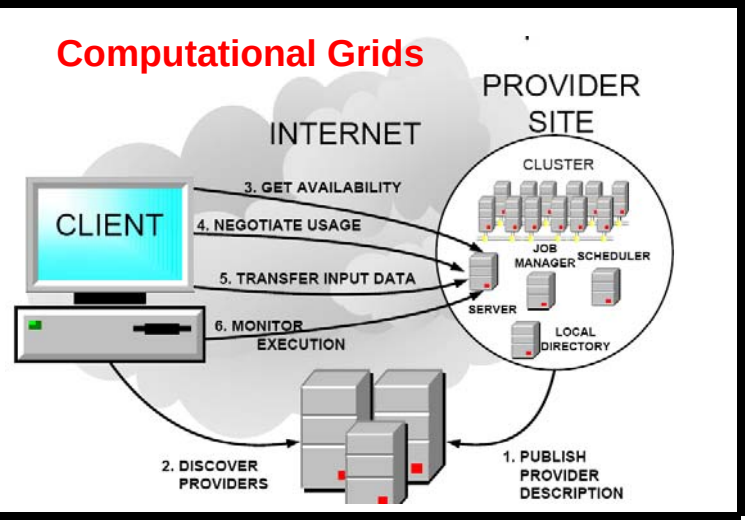
Experiment Design Methods
J. Filliben

Data Analysis Methods
D. Y. Cho
J. Filliben

Visualization Methods
J. Hagedorn
C. Houard

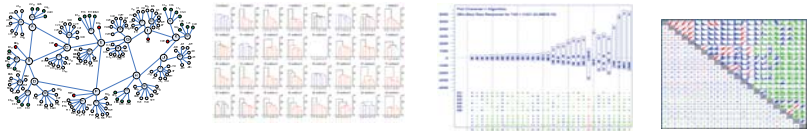
Problem Domains





Problem Approaches

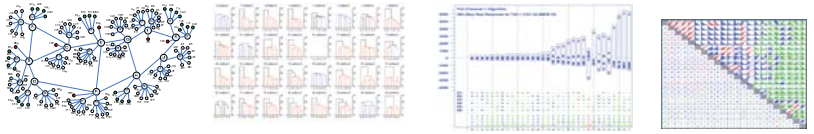
- Fluid-Flow Modeling
- Markov Modeling
- **Mesoscopic Modeling**
- Mean-Field Approximation
- Perturbation Analysis
- **Orthogonal Fractional Factorial Design**
- **Sensitivity Analysis**
- **Clustering Analysis**
- **Principal Components Analysis**
- **Correlation Analysis**
- Multidimensional Data Visualization



Sample Challenge Problems Under Investigation

- Predict effect on global behavior and user experience from adopting proposed replacement congestion-control algorithms for the Internet* (Filliben, Cho, Houard & Mills)
- Evaluate accuracy of proposed fluid-flow models for TCP, characterize limits of applicability of such models and propose improved analytical models (Genin & Marbukh)
- Devise efficient Markov models to accurately simulate large-scale systems and apply perturbation analysis to predict system changes that could lead to undesired behaviors (Dabrowski & Hunt)
- Investigate the use of economic methods for resource allocation in large distributed systems (e.g., computational grids and networks) (Dabrowski, Marbukh & Mills)

*Later I use this challenge problem to illustrate some of our approaches



Sample Artifacts Produced by the Project

MesoNet – a medium scale network simulator that includes seven congestion control algorithms: BIC, CTCP, FAST, HSTCP, H-TCP, Scalable TCP and TCP Reno

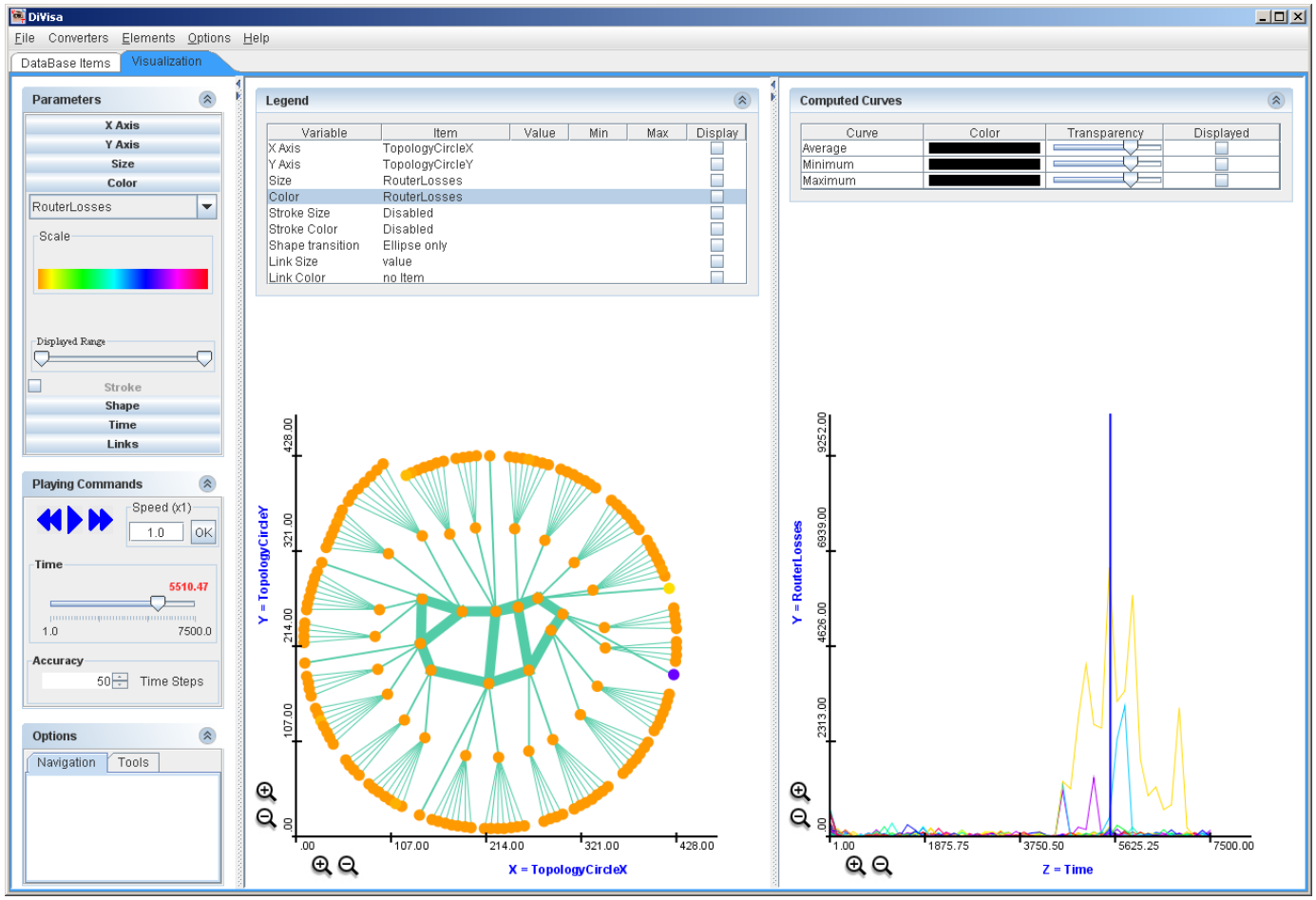
EconoGrid – a detailed simulation model of a standards-based Grid compute economy

Flexi-Cluster – a flexible simulation model of a compute cluster that includes alternate, replaceable functions for pricing, admission control, scheduling and queue management

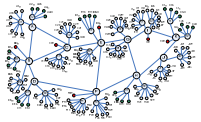
Markov-Model Rewriter – software to systematically perturb a Markov model with bounds defined by a user

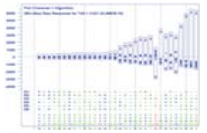
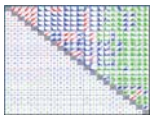
DiVisa – software for interactive exploration of multidimensional data

Interactive Exploration of Multidimensional Data



Download from – http://math.nist.gov/mcsd/savg/software/divisa/



Sample Papers Produced by the Project (1 of 2)

C. Dabrowski, “Reliability in Grid Computing Systems”, in *Concurrency and Computation: Practice and Experience*, Wiley-Blackwell, in press.

D. Genin and V. Marbukh, "Do Current Fluid Approximation Models Capture TCP Instability?“, submitted to ICC 2009, Dresden, Germany, June 14 -18.

C. Dabrowksi and F. Hunt, “Using Markov Chain Analysis to Study Dynamic Behavior in Large-Scale Grid Systems”, *Proceedings of the 7th Australasian Symposium on Grid Computing and e-Research*, Wellington, New Zealand, Jan. 2009.

C. Dabrowski and F. Hunt, Markov Chain Analysis for Large-Scale Grid Systems, NIST Technical Report (under review).

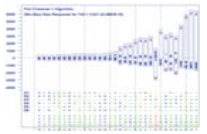
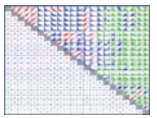
D. Genin and V. Marbukh, "Metastability in cellular networks with migrating users: emergence and implications for performance.“ GLOBECOM 2008, New Orleans, Nov. 31 - Dec. 4.

K. Mills and C. Dabrowski, “Can Economics-based Resource Allocation Prove Effective in a Computation Marketplace?", *Journal of Grid Computing*, Vol. 6, No. 3, September 2008, pp. 291-311.

F. Hunt and V. Marbukh, “Dynamic Routing and Congestion Control Through Random Assignment of Routes”, *Proceedings of the 5th International Conference on Cybernetics and Information Technologies*, Systems and Applications: CITSA 2008, Orlando FL, July 2008.

V. Marbukh and K. Mills, "Demand Pricing & Resource Allocation in Market-based Compute Grids: A Model and Initial Results", *Proceedings of the 7th International Conference on Networking*, IEEE, April 2008, pp. 752-757.



Sample Papers Produced by the Project (2 of 2)

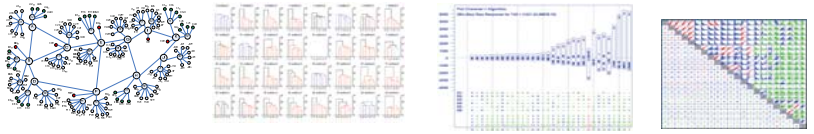
Marbukh and S. Klink, "Decentralized Control of Large-Scale Networks as a Game with Local Interactions: Cross-Layer TCP/IP Optimization", *2nd International Conference on Performance Evaluation Methodologies and Tools*, Nantes, France, October 23-25, 2007.

V. Marbukh, "Utility Maximization for Resolving Throughput/Reliability Trade-offs in an Unreliable Network with Multipath Routing", *2nd International Conference on Performance Evaluation Methodologies and Tools*, Nantes, France, October 23-25, 2007.

K. Mills, "A Brief Survey of Self-Organization in Wireless Sensor Networks", *Wireless Communications and Mobile Computing*, Wiley Interscience, Vol. 7, No. 7, September 2007, pp. 823-834.

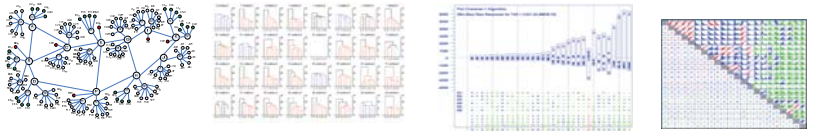
V. Marbukh and K. Mills, "On Maximizing Provider Revenue in Market-Based Compute Grids", *Proceedings of the 3rd International Conference on Networking and Services*, Athens, Greece, June 19-25, 2007.

K. Mills and C. Dabrowski, "Investigating Global Behavior in Computing Grids", Self-Organizing Systems, Lecture Notes in Computer Science, Volume 4124 ISBN 978-3-540-37658-3, pp. 120-136.



Challenge Problem: Study of Proposed Replacement Congestion-Control Algorithms for the Internet

- Modeling the network
  - Parameter state-space reduction techniques
  - Response state-space reduction techniques
  - Orthogonal fractional-factorial experiment design
  - Sensitivity analysis
- Modeling congestion-control algorithms
  - Unified model with phase and procedure alignment
  - Validation against empirical measurements
- Comparing congestion-control algorithms
  - Cluster analysis
  - Detailed analysis of individual responses
  - Condition-response summary analysis
  - Causality analysis – through domain expertise

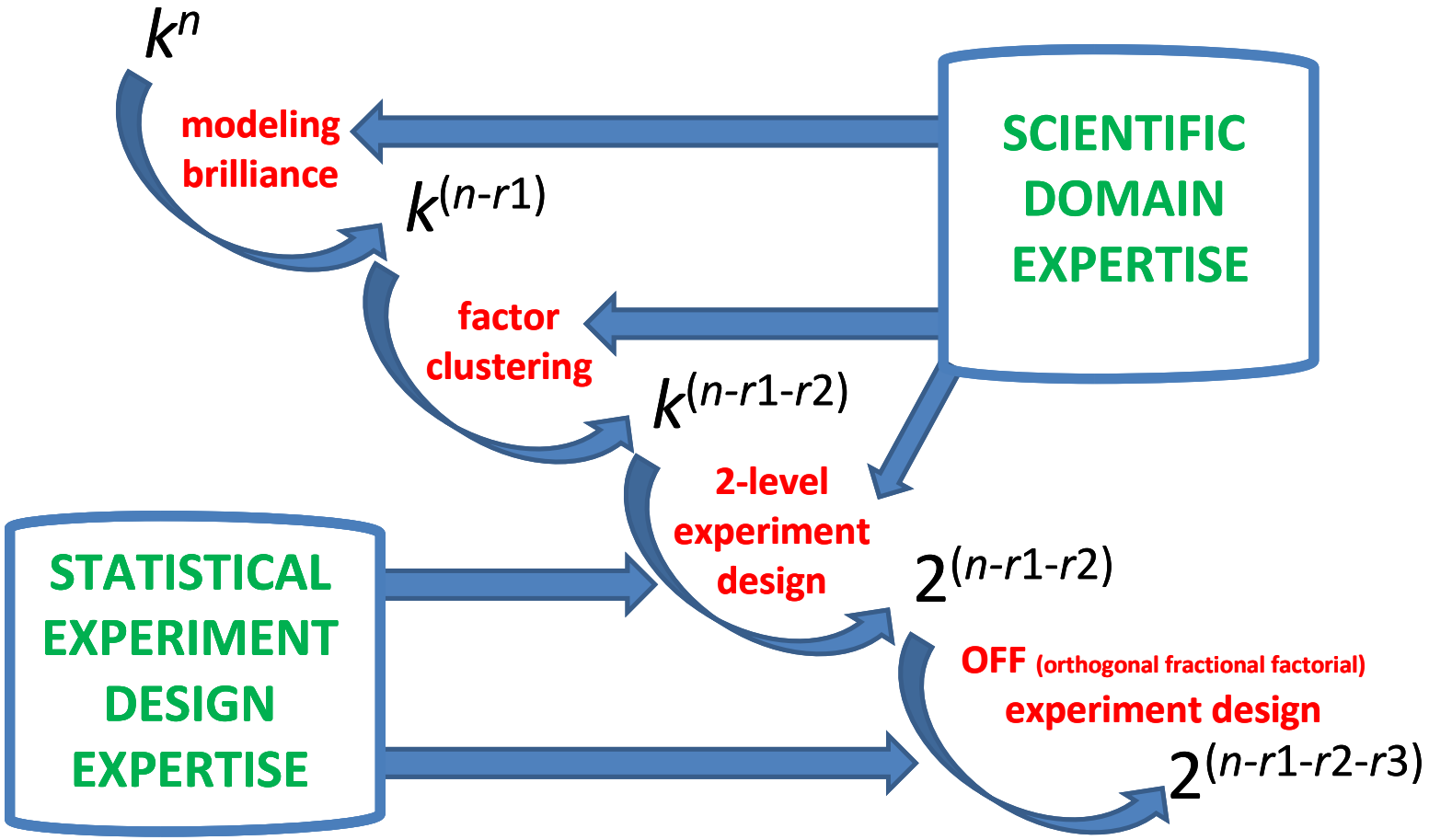


The Modeling State-Space Problem

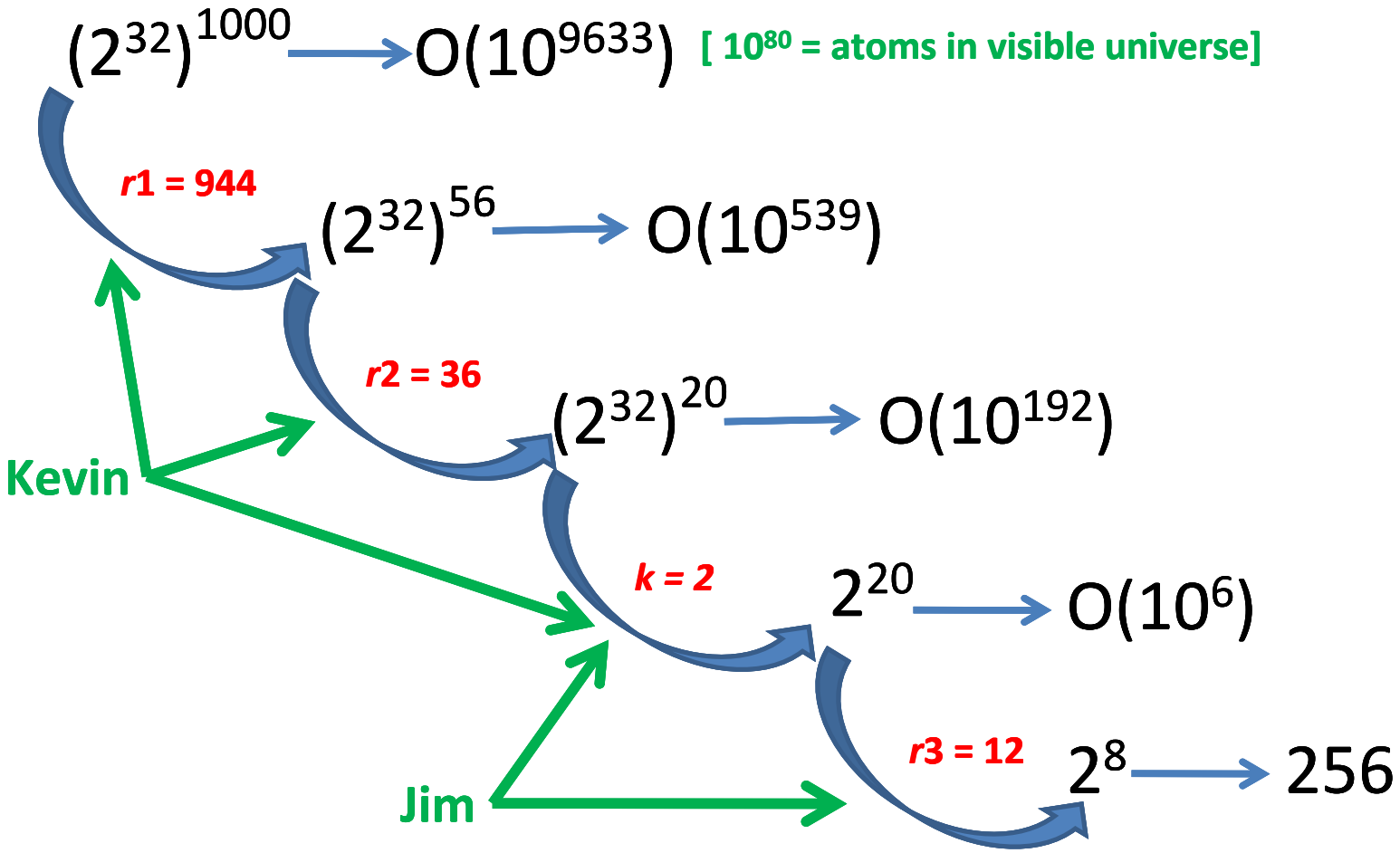
$$\underbrace{y_1, \ldots, y_m}_{\text{Response State-Space}} = f(\underbrace{k \cdot x_1, \ldots, k \cdot x_n}_{\text{Stimulus State-Space}})$$

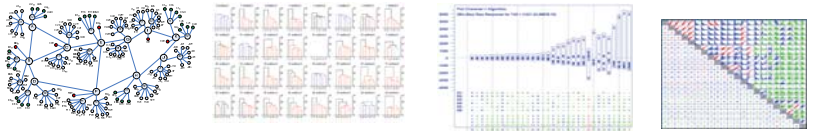
n	Number of inputs (i.e., stimulus factors)
k	Factor range (i.e., number of values each factor can assume)
m	Number of outputs (i.e., responses)

Parameter State-Space Reduction



Parameter State-Space Reduction: MesoNet Example





2^{20-12} OFF Design Improves Computational Feasibility of Searching Parameter State Space

Assumptions	
Processing Time for One Run	8 CPU-Hours
Number of Available CPUs	48

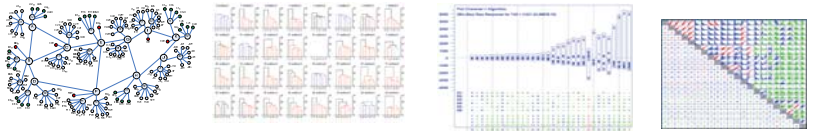
2-Level Experiment Design Requires 20 Years*

(2^{20} runs x 8 CPU-Hours Per Run)/48 CPUs = 174,762.67 Hours

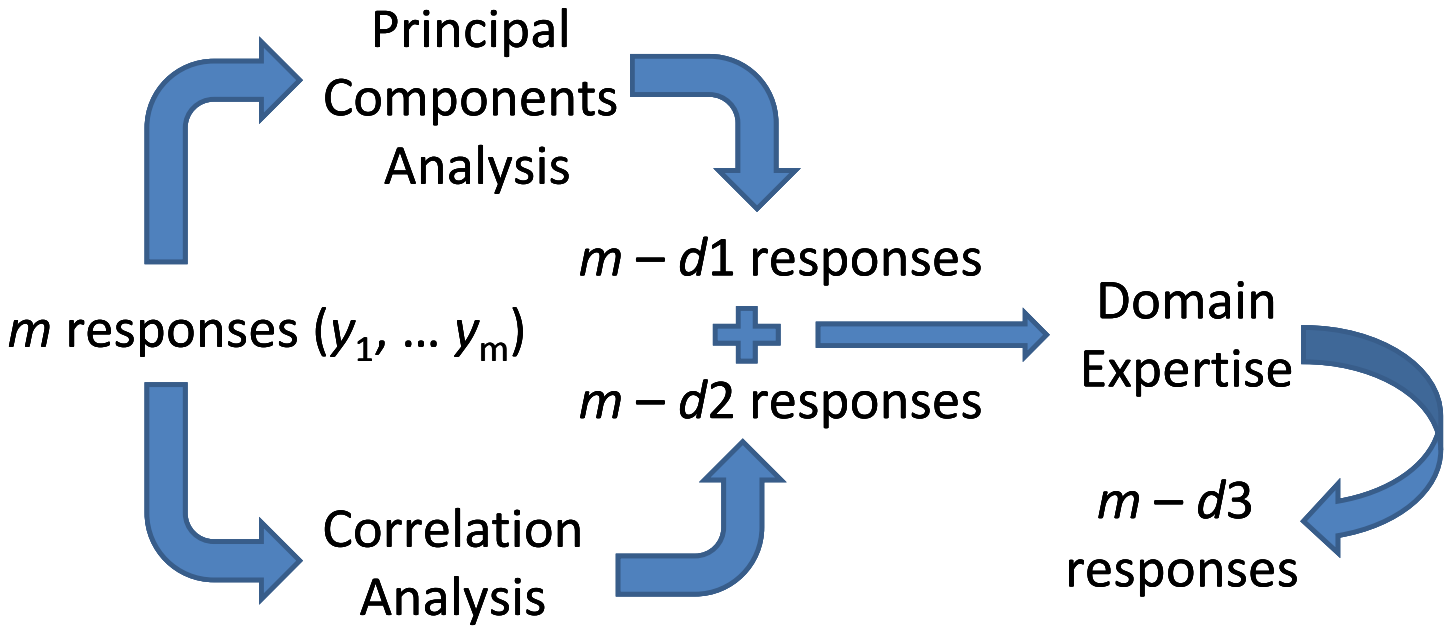
OFF Experiment Design Reduces Requirement To Under 2 Days

(2^{8} runs x 8 CPU-Hours Per Run)/48 CPUs = 42.67 Hours

***If we had 1,000 CPUs to dedicate to this problem, then we could compute the 2^{20} runs in just under 1 year**



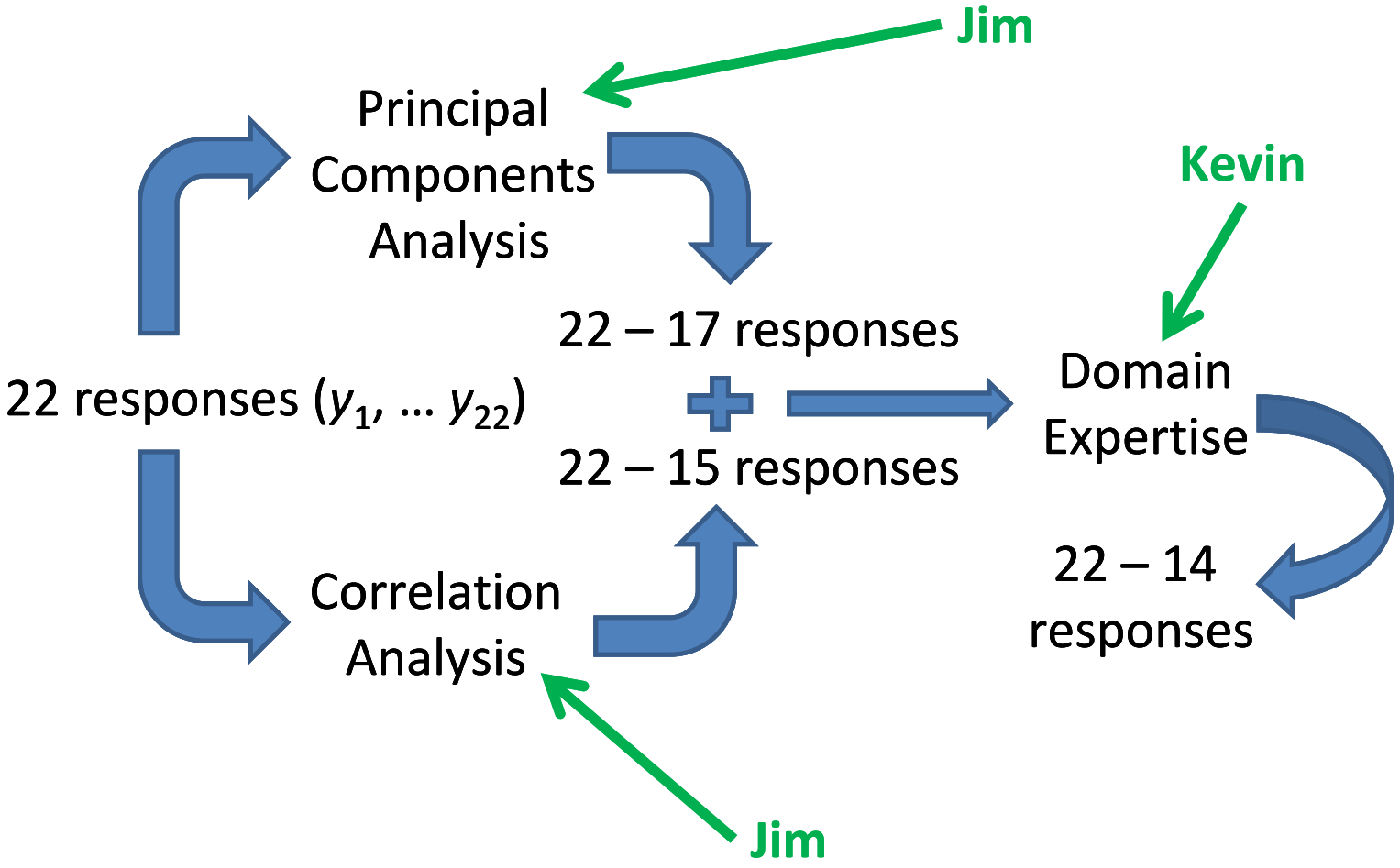
Response State-Space Reduction

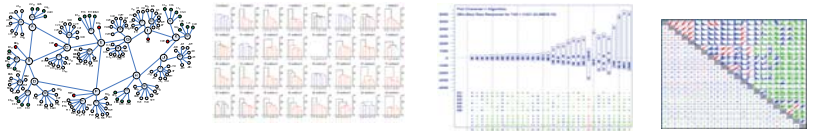


DATA ANALYSIS EXPERTISE

SCIENTIFIC DOMAIN EXPERTISE

Response State-Space Reduction: MesoNet Example





22-Dimension Response State Space from MesoNet

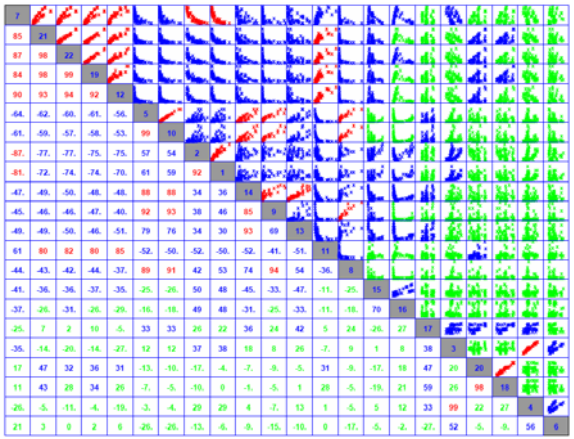
Macroscopic Network Behavior

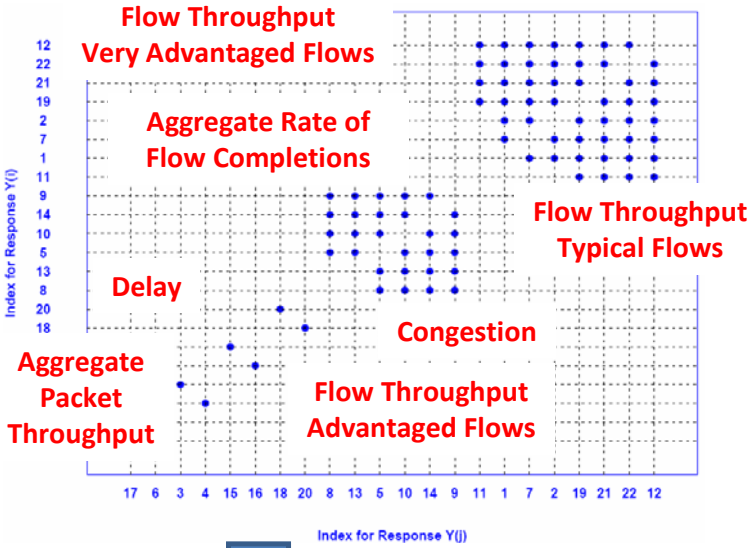
Response	Definition
y1	Active Flows – flows attempting to transfer data
y2	Proportion of potential flows that were active: Active Flows/All Sources
y3	Packets entering the network per measurement interval
y4	Packets leaving the network per measurement interval
y5	Loss Rate: y4/(y3+y4)
y6	Flows Completed per measurement interval
y7	Flow-Completion Rate: y6/(y6+y1)
y8	Connection Failures per measurement interval
y9	Connection-Failure Rate: y8/(y8+y1)
y10	Retransmission Rate
y11	Congestion Window per Flow
y12	Window Increases per Flow per measurement interval
y13	Negative Acknowledgments per Flow per measurement interval
y14	Timeouts per Flow per measurement interval
y15	Smoothed Round-Trip Time
y16	Relative queuing delay: y15/(41*x1)

User Experience

Response	Definition
y17	Average Throughput for active D-D Flows
y18	Average Throughput for active D-F Flows
y19	Average Throughput for active D-N Flows
y20	Average Throughput for active F-F Flows
y21	Average Throughput for active F-N Flows
y22	Average Throughput for active N-N Flows

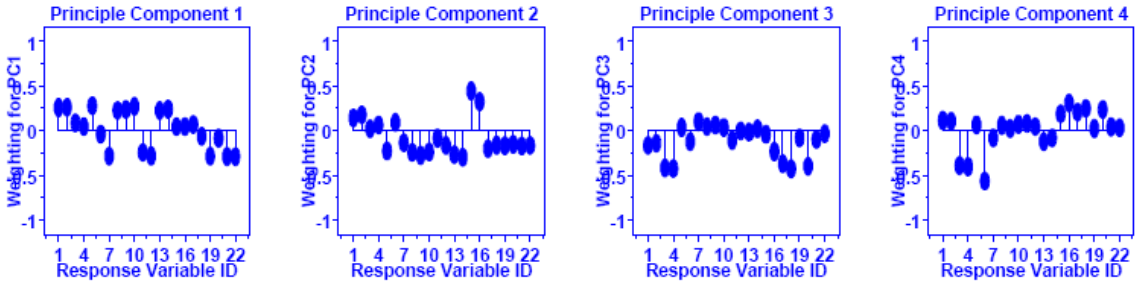
Correlation Analysis Identifies 7 Dimensions





Response	Definition
y4	Average number of packet output per measurement interval
y6	Average number of flows completed per measurement interval
y10	Average retransmission rate
y15	Average smoothed round-trip time
y17	Average instantaneous throughput for D-D flows
y20	Average instantaneous throughput for F-F flows
y22	Average instantaneous throughput for N-N flows

Principal Components Analysis Suggests 4 Dimensions



PC1 – Congestion

Response	Definition
y1	Average number of active flows
y2	Proportion of possible flows that are active
y5	Loss rate
y7	Flow-completion rate
y10	Retransmission rate
y11	Average congestion window
y12	Window-increase rate
y13	Negative-acknowledgment rate
y14	Timeout rate
y19	Average instantaneous throughput for D-N flows
y21	Average instantaneous throughput for F-N flows
y22	Average instantaneous throughput for N-N flows

PC2 – Delay

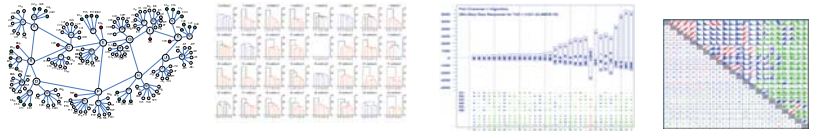
Response	Definition
y15	Smoothed round-trip time
y16	Relative queuing delay

PC4 – Aggregate Throughput

Response	Definition
y3	Packets input
y4	Packets output
y6	Flows completed per measurement interval

PC3 – Throughput for Advantaged Flows

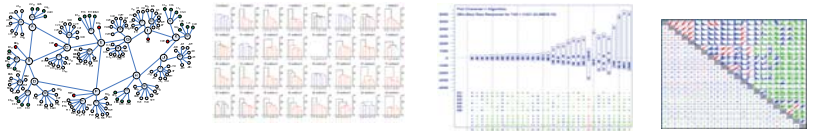
Response	Definition
y3	Packets input
y4	Packets output
y17	Average instantaneous throughput for D-D flows
y18	Average instantaneous throughput for D-F flows
y20	Average instantaneous throughput for F-F flows



Domain Expert Selects 8 Dimensions

Response	Definition
y1	Average number of active flows
y4	Average number of packet output per measurement interval
y6	Average number of flows completed per measurement interval
y10	Average retransmission rate
y15	Average smoothed round-trip time
y17	Average instantaneous throughput for D-D flows
y20	Average instantaneous throughput for F-F flows
y22	Average instantaneous throughput for N-N flows

Domain Expert Sides with the Correlation Analysis and Adds One Response (y1)

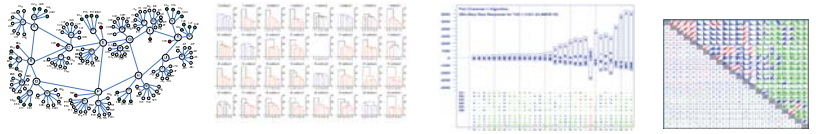


Sensitivity Analysis as a Model Validation Step

What parameters (or combinations of parameters) determine a model's responses?

Does the influence of parameters on responses make sense to a domain expert?

Can any unexpected responses be explained and do the explanations make sense?



11 of 20 MesoNet Parameters Selected for Sensitivity Analysis

* selected parameter

other parameters fixed

Network Parameters	
Topology	Next slide
Propagation Delay	*
Speed	*
Buffer Sizing	*

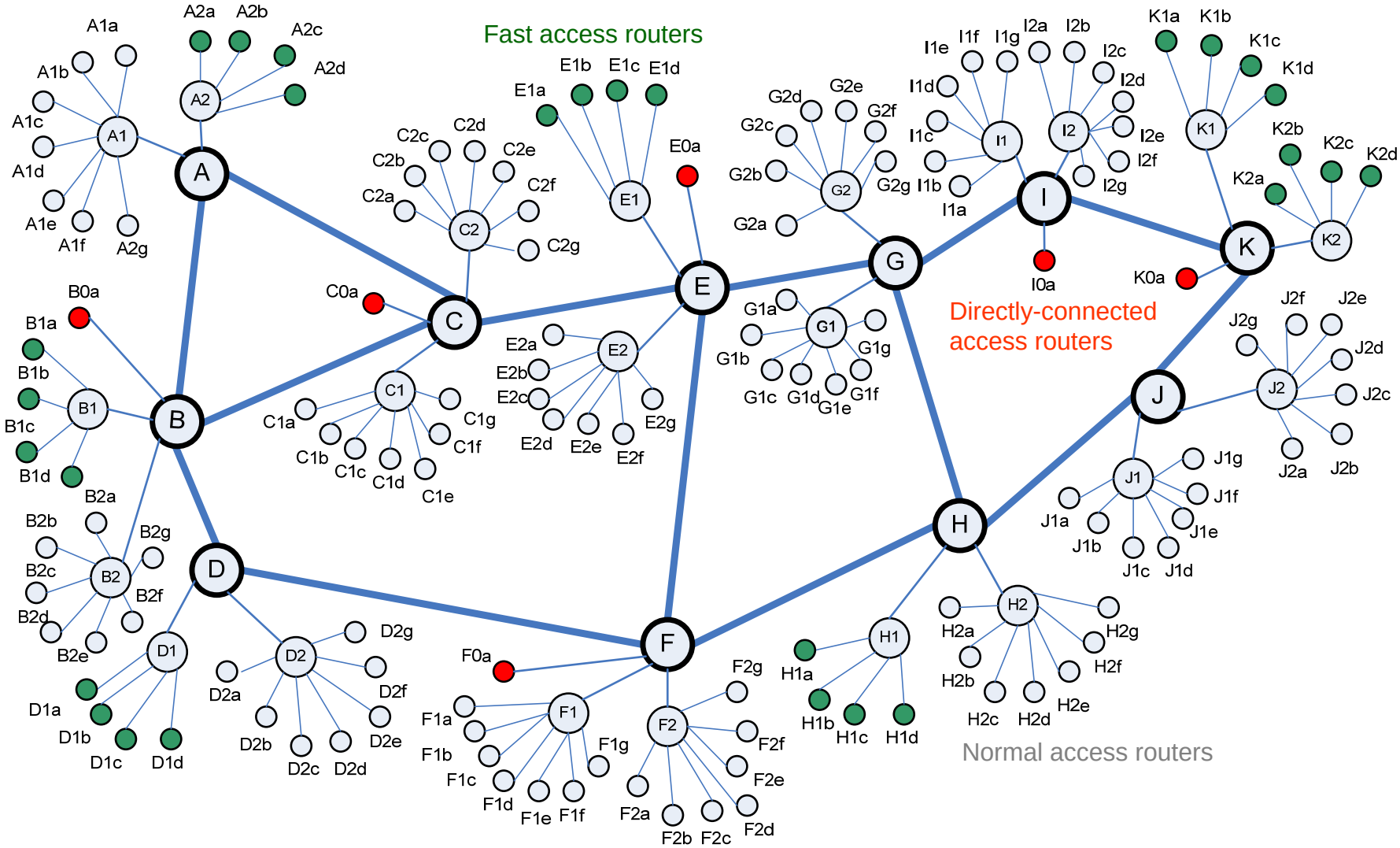
User Behavior	
Think Time	*
File-Size Distribution	*
Pattern of Long-Lived Flows	None

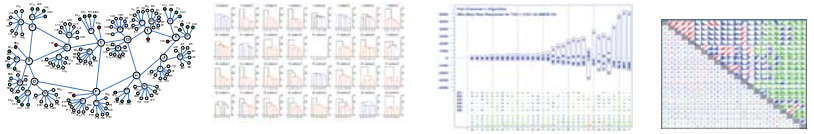
Protocol Parameters	
Initial Congestion Window	2 packets
Initial Slow-Start Threshold	*
Type of Slow Start Regime	Limited slow start

Characteristics of Sources & Receivers	
Number of Sources	*
Distribution of Sources	*
Number of Receivers	*
Distribution of Receivers	*
Distribution of Network-Interface Speeds	*
Distribution of Congestion-Control Mechanisms	All TCP
Startup Pattern for Sources	25% at a time

Simulation Control	
Measurement-Interval Size	200 ms
Number of Measurement Intervals	6000
Random-Number Seed	200000

Representative Heterogeneous Four-Tier Topology Selected





2^{11-5} Orthogonal Fraction Factorial (OFF) Experiment Design

Design Template

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
-1	-1	-1	-1	-1	-1	-1	+1	-1	+1	+1
+1	-1	-1	-1	-1	-1	-1	-1	+1	+1	-1
-1	+1	-1	-1	-1	-1	-1	-1	+1	-1	+1
+1	+1	-1	-1	-1	-1	-1	+1	-1	-1	-1
-1	-1	+1	-1	-1	-1	+1	-1	-1	+1	+1
+1	-1	+1	-1	-1	-1	+1	+1	+1	+1	-1
-1	+1	+1	-1	-1	-1	+1	+1	+1	-1	+1
+1	+1	+1	-1	-1	-1	+1	-1	-1	-1	-1
-1	-1	-1	+1	-1	-1	+1	-1	-1	-1	-1
+1	-1	-1	+1	-1	-1	+1	+1	+1	-1	+1
-1	+1	-1	+1	-1	-1	+1	+1	+1	+1	-1
+1	+1	-1	+1	-1	-1	+1	-1	-1	+1	+1
-1	-1	+1	+1	-1	-1	-1	+1	-1	-1	-1
+1	-1	+1	+1	-1	-1	-1	-1	+1	-1	+1
-1	+1	+1	+1	-1	-1	-1	-1	+1	+1	-1
+1	+1	+1	+1	-1	-1	-1	+1	-1	+1	+1
-1	-1	-1	-1	+1	-1	+1	+1	-1	-1	-1
+1	-1	-1	-1	+1	-1	+1	-1	+1	-1	+1
-1	+1	-1	-1	+1	-1	+1	-1	+1	+1	-1
+1	+1	-1	-1	+1	-1	+1	+1	-1	+1	+1
-1	-1	+1	-1	+1	-1	-1	-1	-1	-1	-1
+1	-1	+1	-1	+1	-1	-1	+1	+1	-1	+1
-1	+1	+1	-1	+1	-1	-1	+1	+1	+1	-1
+1	+1	+1	-1	+1	-1	-1	-1	-1	+1	+1
-1	-1	-1	+1	+1	-1	+1	-1	+1	-1	+1
+1	-1	-1	+1	+1	-1	+1	+1	-1	-1	-1
-1	+1	-1	+1	+1	-1	+1	+1	-1	+1	+1
+1	+1	-1	+1	+1	-1	+1	-1	+1	+1	-1
-1	-1	+1	+1	+1	-1	-1	+1	+1	+1	-1
+1	-1	+1	+1	+1	-1	-1	-1	-1	+1	+1
-1	+1	+1	+1	+1	-1	-1	-1	-1	+1	-1
+1	+1	+1	+1	+1	-1	-1	+1	+1	-1	+1
-1	-1	-1	-1	-1	+1	+1	+1	+1	-1	-1
+1	-1	-1	-1	-1	+1	+1	-1	-1	+1	+1
-1	+1	-1	-1	-1	+1	+1	-1	-1	+1	-1
+1	+1	-1	-1	-1	+1	+1	+1	+1	-1	+1
-1	-1	+1	-1	-1	+1	-1	-1	+1	+1	+1
+1	-1	+1	-1	-1	+1	-1	+1	-1	-1	-1
-1	+1	+1	-1	-1	+1	-1	+1	-1	+1	-1
+1	+1	+1	-1	-1	+1	-1	-1	+1	-1	+1
-1	-1	-1	+1	-1	+1	+1	-1	+1	-1	+1
+1	-1	-1	+1	-1	+1	+1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	+1	+1	-1	-1	+1
+1	+1	-1	+1	-1	+1	+1	-1	+1	+1	-1
-1	-1	+1	+1	-1	+1	-1	+1	+1	+1	-1
+1	-1	+1	+1	-1	+1	-1	-1	-1	-1	+1
-1	+1	+1	+1	-1	+1	-1	-1	-1	-1	+1
+1	+1	+1	+1	-1	+1	-1	+1	+1	+1	-1
-1	-1	-1	-1	+1	+1	+1	+1	+1	+1	-1
+1	-1	-1	-1	+1	+1	+1	-1	-1	-1	+1
-1	+1	-1	-1	+1	+1	+1	-1	-1	-1	+1
+1	+1	-1	-1	+1	+1	+1	+1	+1	+1	-1
-1	-1	+1	-1	+1	+1	-1	+1	-1	+1	-1
+1	-1	+1	-1	+1	+1	-1	-1	+1	-1	+1
-1	+1	+1	-1	+1	+1	-1	+1	-1	-1	+1
+1	+1	+1	-1	+1	+1	-1	-1	+1	+1	-1
-1	-1	-1	+1	+1	+1	-1	-1	+1	-1	-1
+1	-1	-1	+1	+1	+1	-1	+1	-1	-1	+1
-1	+1	-1	+1	+1	+1	-1	+1	-1	+1	-1
+1	+1	-1	+1	+1	+1	-1	-1	+1	+1	+1
-1	-1	+1	+1	+1	+1	+1	+1	+1	-1	-1

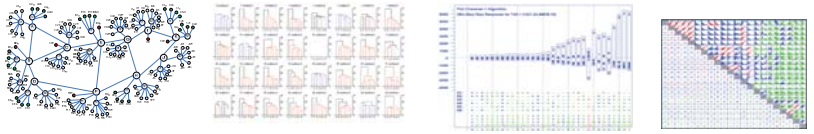
Full Factorial Design Requires 2^{11} = 2048 runs

We can only afford 2^{6} = 64 runs; thus,

we require a 2^{11-5} OFF Experiment Design

No confounding of main effects with two-term interactions

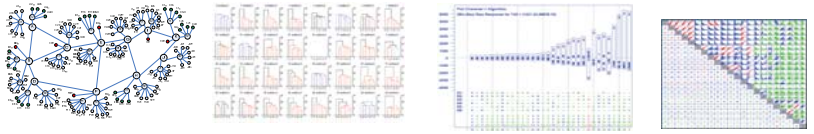
Main effects may be confounded with three-term and higher interactions



Values Selected to Replace +1 and -1 in Design Template

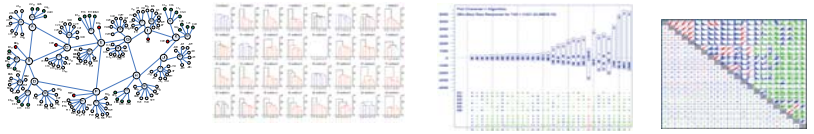
	Parameter	Factor	+1	-1
Network Factors	Multiplier for Propagation Delay	x1	2	1
	Backbone Router Speed (ppms)*	x2	400	800
	Buffer Sizing Algorithm	x3	RTTxC	*RTT*x*C*/SQR(*n*)
User Factors	Average File Size (packets)	x4	100	50
	Average Think Time (ms)	x5	5000	2000
	Probability of a Larger File*	x6	0.01	0.02
Source & Receiver Factors	Probability of Fast Network Interface*	x7	0.2	0.4
	Multiplier on Number Sources Per Access Router	x8	3	2
	Distribution of Sources	x9	P2P	WEB
	Distribution of Receivers	x10	P2P	WEB
Protocol Factors	Initial Slow-Start Threshold (packets)	x11	1.07×10^9	43

*By convention +1 is coded with larger value and -1 is coded with smaller value. Here, coding was reversed in three cases. This makes no difference in the construction of the experiment, but must be managed during data analysis



Speed of All Routers Derived from Speed of Backbone Routers (Topology Tiers 1 to 3)

Router Type	Parameter	Equation	+1	-1
Backbone	x2	= x2	400 ppms	800 ppms
POP	R2 (= 4)	= x2/R2	100 ppms	200 ppms
Typical Access	R3 (= 10)	= x2/R2/R4	10 ppms	20 ppms
Fast Access	FA (= 2)	= x2/R2/R4xFA	20 ppms	40 ppms
Directly Connected Access	DC (= 10)	= x2/R2/R4xDC	100 ppms	200 ppms



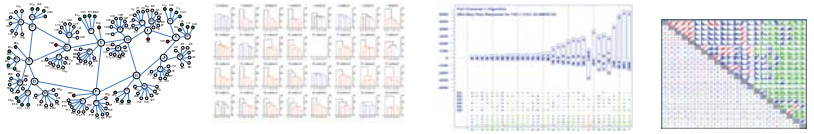
Number & Distribution of Sources & Receivers (Tier 4)

Sources

x8	x9	x10	Total Sources	% under D Routers	% under F Routers	% under N Routers
2	P2P	P2P	27,800	4.32	20.14	75.54
3	P2P	P2P	41,700	4.32	20.14	75.54
2	WEB	WEB	18,560	6.46	48.27	45.25
3	WEB	WEB	27,840	6.46	48.27	45.25
2	P2P	WEB	27,800	4.32	20.14	75.54
3	P2P	WEB	41,700	4.32	20.14	75.54
2	WEB	P2P	18,560	6.46	48.27	45.25
3	WEB	P2P	27,840	6.46	48.27	45.25

Receivers

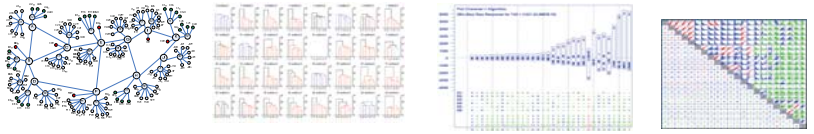
x8	x9	x10	Total Receivers	% under D Routers	% under F Routers	% under N Routers
2	P2P	P2P	111,200	4.32	20.14	75.54
3	P2P	P2P	166,800	4.32	20.14	75.54
2	WEB	WEB	146,400	2.45	11.47	86.06
3	WEB	WEB	219,600	2.45	11.47	86.06
2	P2P	WEB	146,400	2.45	11.47	86.06
3	P2P	WEB	219,600	2.45	11.47	86.06
2	WEB	P2P	111,200	4.32	20.14	75.54
3	WEB	P2P	166,800	4.32	20.14	75.54



Resulting Distribution of Flow Classes

x8	x9	x10	% DD Flows	% DF Flows	% DN Flows	% FF Flows	% FN Flows	% NN Flows
2	P2P	P2P	0.186	1.74	6.52	4.05	30.43	57.06
3	P2P	P2P	0.186	1.74	6.52	4.05	30.43	57.06
2	WEB	WEB	0.159	1.92	6.67	5.53	46.74	38.95
3	WEB	WEB	0.159	1.92	6.67	5.53	46.74	38.95
2	P2P	WEB	0.106	0.99	5.57	2.31	26.00	65.01
3	P2P	WEB	0.106	0.99	5.57	2.31	26.00	65.01
2	WEB	P2P	0.279	3.38	6.83	9.72	45.58	34.18
3	WEB	P2P	0.279	3.38	6.83	9.72	45.58	34.18

Flow classes defined by relative locations of source & receiver



Influence of other Factors on Average Buffer Sizes

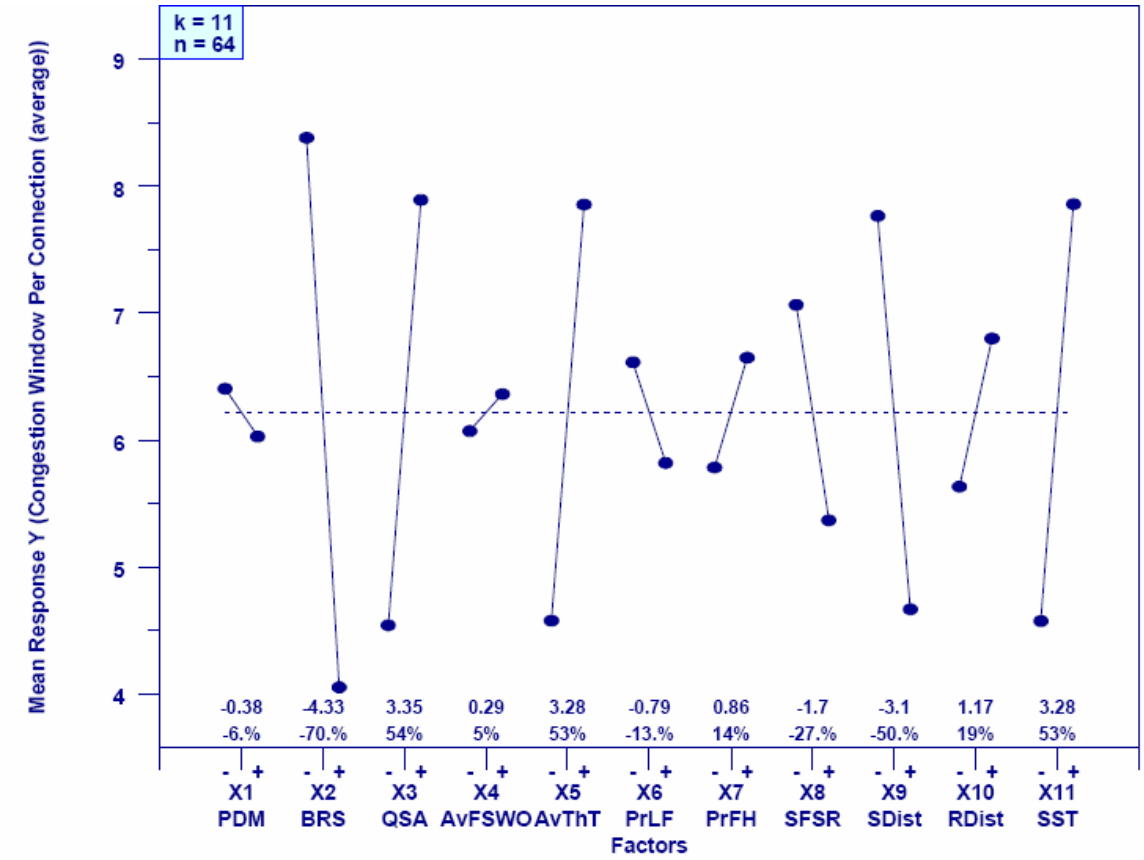
RTT x C

x1	x2	Backbone Router Buffers (avg.)	POP Router Buffers (avg.)	Access Router Buffers (avg.)
1	400	16277	4070	647
2	400	32553	8139	1294
1	800	32553	8139	1294
2	800	65106	16277	2588

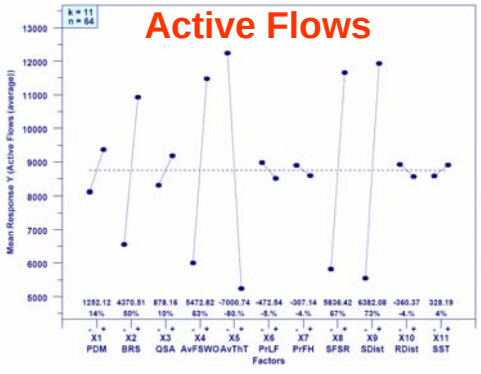
RTT x C/ SQR(n)

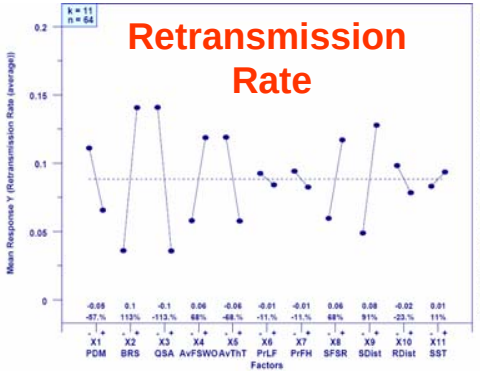
x1	x2	Backbone Router Buffers (avg.)	POP Router Buffers (avg.)	Access Router Buffers (avg.)
1	400	182	68	27
2	400	364	135	53
1	800	364	135	53
2	800	728	270	105

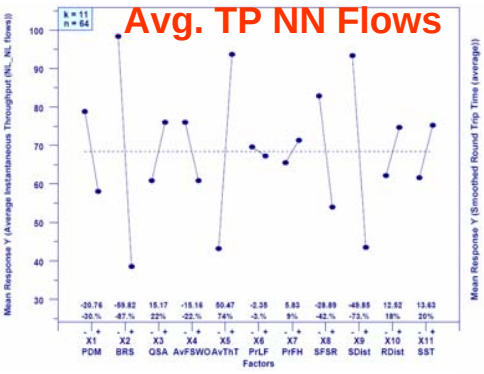
Sensitivity Analysis Relies on Main Effects Plots

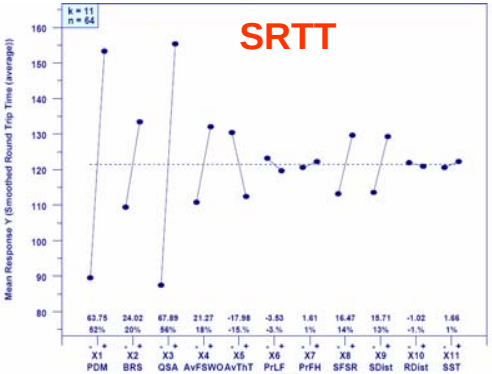


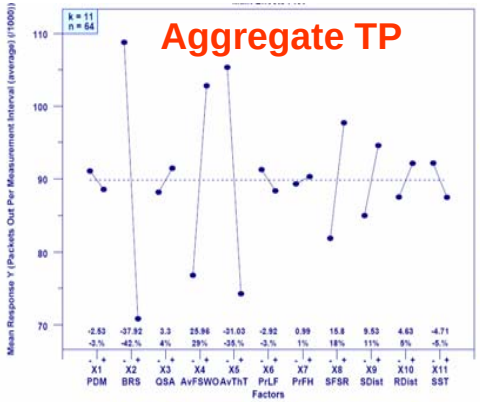
Sensitivity Analysis Driven by Correlation Analysis

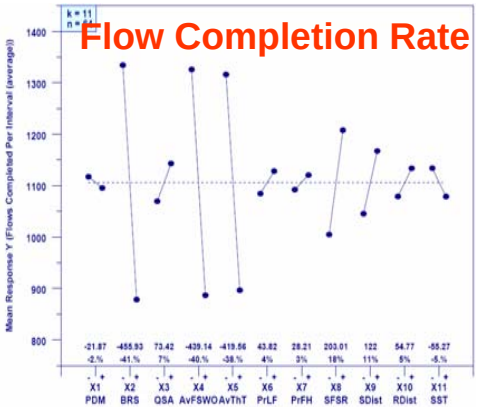


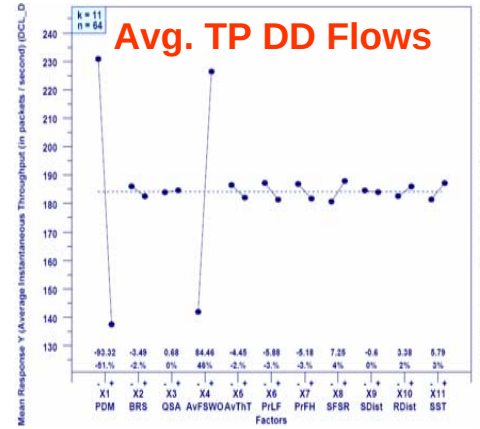


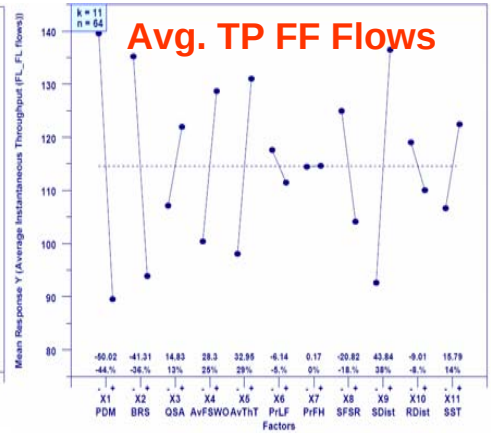






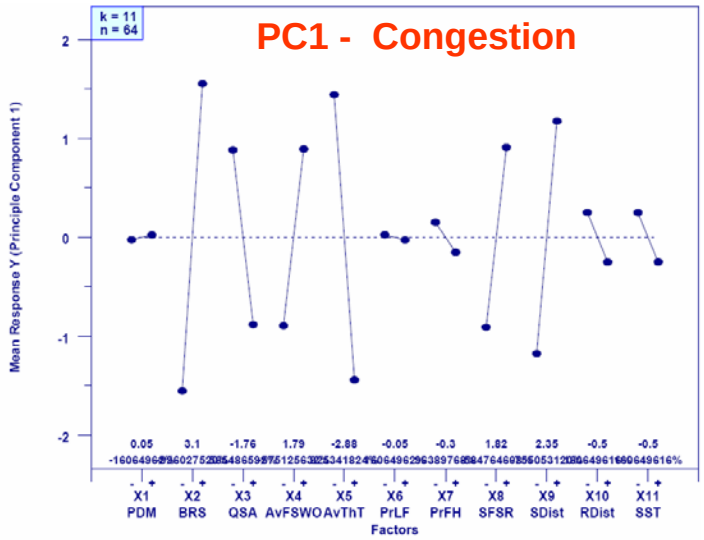


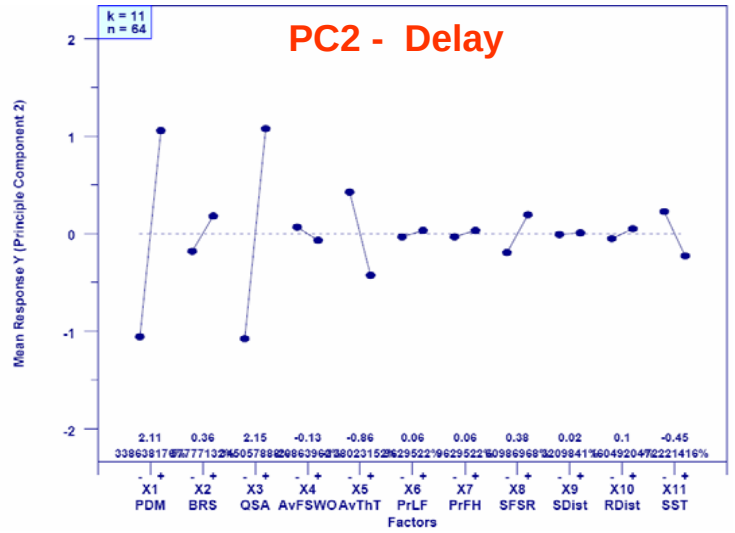


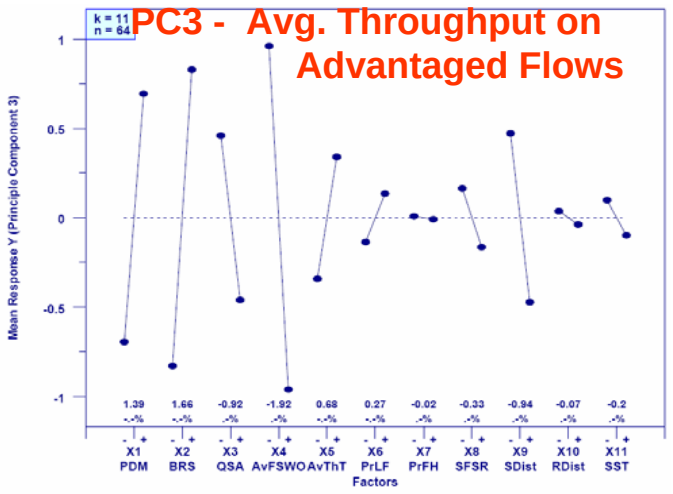


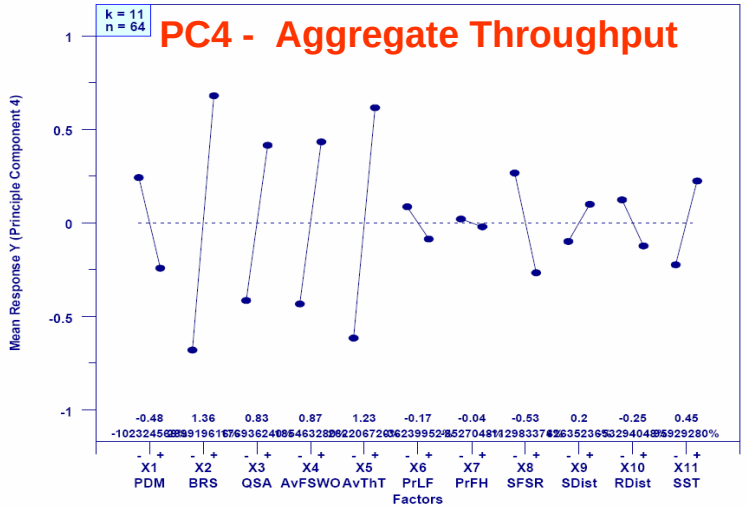
Seven Responses Chosen by Correlation Analysis + 1

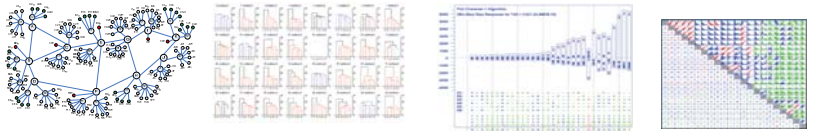
Sensitivity Analysis Driven by Principal Components Analysis











Major Factors Influencing Model Behavior

(1) Network Speed

(2) File Size

(3) Think Time

(4) Number of Sources

(5) Propagation Delay

(6) Distribution of Sources

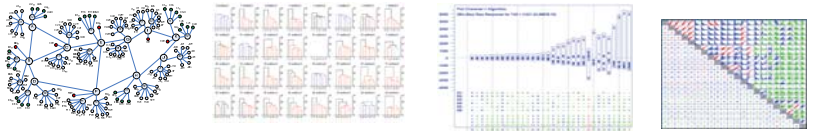
(7) Buffer Size – small buffer size reduces delay variability & larger buffer size has greater effect under high network speed

Correlation-based Analysis

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
y4	9.5	1	8	3	2	9.5	11	4	5	6.5	6.5
y6	11	1	6	2	3	9	10	4	5	7.5	7.5
y10	7	1.5	1.5	5	5	10	10	5	3	8	10
y15	2	3	1	4	5	8	10	6	7	10	10
y17	1	8	10.5	2	8	5	5	3	10.5	8	5
y20	1	3	8	5	4	10	11	6	2	9	7
Average Rank	5.25	2.92	5.83	3.5	4.5	8.58	9.50	4.67	5.42	8.17	7.67
Ordinal Rank	5	1	7	2	3	10	11	4	6	9	8

Principal Components-based Analysis

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
PC1	10.5	1	6	5	2	10.5	9	4	3	7.5	7.5
PC2	2	6	1	7	3	8.5	8.5	4	11	10	5
PC3	3	2	5	1	6	8	10.5	7	4	9	10.5
PC4	5	1	4	3	2	10	11	7	9	8	6
Average Rank	5.13	2.50	4.00	4.00	3.25	9.25	9.75	5.50	6.75	8.63	7.25
Ordinal Rank	5	1	4	4	2	10	11	6	7	9	8



Unified Model for Congestion-Control Algorithms

LIFE OF A TCP FLOW

CONNECTION PHASE	TRANSFER PHASE	
	SLOW START	CONGESTION AVOIDANCE

MS Windows® TCP

Symbol	Definition
syn_{INT}	Timeout interval for initial SYN
syn_{MAX}	Maximum number of SYNs to send
syn_{SENT}	Number of SYNs that have been sent
syn_{TO}	Timeout for current SYN
time	Current time

InitiateConnection ≡
- $syn_{MAX} \leftarrow 3$
- $syn_{INT} \leftarrow 3\ s$
- $syn_{TO} \leftarrow \textbf{time} + syn_{INT}$
- $syn_{SENT} \leftarrow 1$
- **send**(SYN)

Timeout ≡
- *if* $syn_{SENT} < syn_{MAX}$
  - $syn_{INT} \leftarrow 2 \times syn_{INT}$
  - $syn_{TO} \leftarrow \textbf{time} + syn_{INT}$
  - $syn_{SENT} \leftarrow syn_{SENT} + 1$
  - **send**(SYN)
- **signal**(ConnectionFailure) *otherwise*

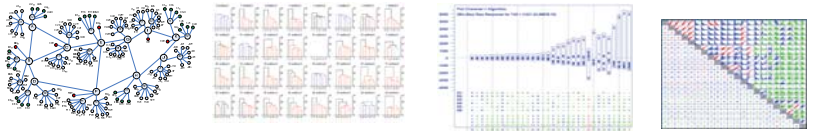
Symbol	Definition
cwnd	Current congestion window
$cwnd_{INT}$	Initial congestion window (we use $cwnd_{INT} = 2$)
sst	Current slow-start threshold
sst_{MAX}	Threshold to switch from exponential to logarithmic increase (varies with experiment)
sst_{INT}	Threshold to terminate initial slow start (varies with experiment)

InitiateTransferPhase ≡
- $cwnd \leftarrow cwnd_{INT}$
- $sst \leftarrow sst_{INT}$

ACK ∧ (*cwnd* < *sst*) ≡
- $cwnd \leftarrow cwnd + 1$ *if* $cwnd < sst_{MAX}$
- $cwnd \leftarrow cwnd + \frac{1}{\left(\frac{cwnd}{0.5 \times sst_{MAX}}\right)}$ *otherwise*

Standard or Limited

- Window increase procedures (ACK)
- Window decrease procedures (Loss)
- Timeout procedures
- Optional periodic procedures:
  CTCP, FAST, H-TCP
- Optional mode switch between TCP & alternate procedures:
  BIC, CTCP, HSTCP, H-TCP, Scalable TCP



Sample: CTCP Congestion-Avoidance Model

Symbol	Definition
α_C	Window increase (α_C = 0.125) weight for CTCP
A_C	Actual throughput ($cwnd/SRTT_C$) experienced on CTCP flow
β_C	Window decrease (β_C = 0.5) weight for CTCP
CD_C	Boolean denoting whether early congestion has been detected (**true**) or not (**false**)
γ_C	CTCP gamma threshold (γ_C = 30) for detecting early congestion
D_C	Difference between expected and actual throughput experienced on CTCP flow
$dwnd$	CTCP delay window
E_C	Expected throughput ($cwnd/minRTT_C$) on CTCP flow
ζ_C	CTCP zeta parameter (ζ_C = 0.1) defining reduction speed in delay window
k_C	Exponent (k_C = 0.8) for CTCP window-increase procedures
LW_C	Low-window threshold (LW_C = 41) for applying CTCP procedures
$minRTT_C$	Minimum round-trip time experienced on CTCP flow
$SRTT_C$	Average Smoothed Round-Trip Time experienced on CTCP flow

Mode Switch Procedures

$$\textbf{SelectProcedures} \equiv \begin{cases} \textbf{TCPcongestionAvoidance} & \textit{if } cwnd < LW_C \\ \textbf{CTCPcongestionAvoidance} & \textit{otherwise} \end{cases}$$

Increase Procedures

$$\textbf{ACK} \equiv \begin{cases} cwnd \leftarrow cwnd + \dfrac{1}{(cwnd + dwnd)} \\ cwnd \leftarrow cwnd + dwnd \end{cases}$$

Decrease Procedures

$$\textbf{Loss} \equiv \begin{cases} cwnd \leftarrow \dfrac{cwnd}{2} + dwnd \\ CD_C \leftarrow \textbf{true} \\ sst \leftarrow cwnd \end{cases}$$

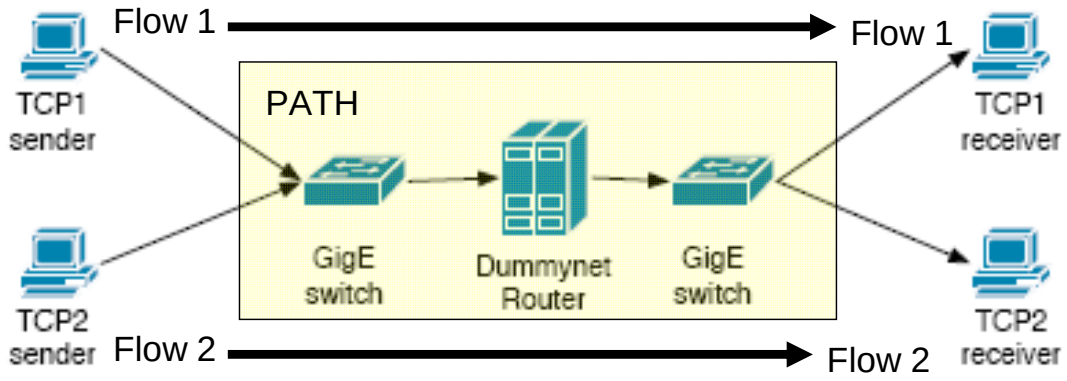
Timeout Procedures

$$\textbf{Timeout} \equiv \begin{cases} sst \leftarrow max\left(\dfrac{cwnd}{2}, cwnd_{INT}\right) \\ cwnd \leftarrow cwnd_{INT} \\ dwnd \leftarrow \textbf{0} \\ CD_C \leftarrow \textbf{true} \end{cases}$$

Periodic Procedures

$$\textbf{every}(SRTT_C) \equiv \begin{cases} E_C \leftarrow \dfrac{cwnd}{minRTT_C} \\ A_C \leftarrow \dfrac{cwnd}{SRTT_C} \\ D_C \leftarrow (E_C - A_C) \times minRTT_C \\ \textit{if } CD_C = \textbf{true} \\ \quad \begin{cases} dwnd \leftarrow \textbf{min}\left[\textbf{0}, cwnd \times (\textbf{1} - \beta_C) - \dfrac{cwnd}{2}\right] \\ CD_C \leftarrow \textbf{false} \end{cases} \\ dwnd \leftarrow dwnd + \textbf{min}\left(\textbf{0}, \alpha_C \times cwnd^{k_C} - \textbf{1}\right) \quad \textit{if } CD_C = \textbf{false} \wedge D_C < \gamma_C \\ dwnd \leftarrow \textbf{min}\left[\textbf{0}, dwnd - (\zeta_C \times D_C)\right] \quad \textit{otherwise} \\ cwnd \leftarrow \textbf{max}(\textbf{int\_max}, cwnd + dwnd) \end{cases}$$

Empirical Studies of Six Alternate Congestion-Control Algorithms



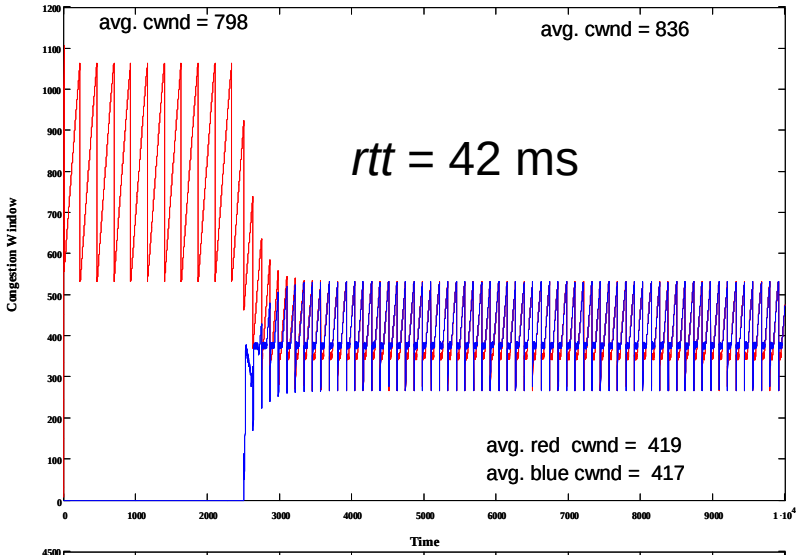
1. Yee-Ting, Leith and Shorten, "Experimental Evaluation of TCP Protocols for High-Speed Networks", *IEEE/ACM Transactions on Networking*, (15)5, October 2007, pp. 1109 – 1122.
   - Covers BIC, FAST, HSTCP, H-TCP, Scalable TCP
   - Uses implementations within Linux kernel

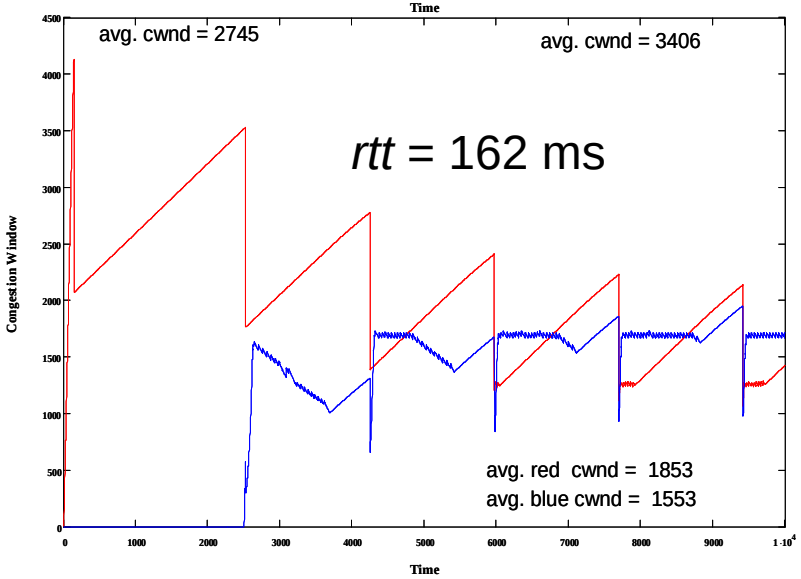
2. Leith, Andrew, Quetchenbach, Shorten and Lavi, "Experimental Evaluation of Delay/Loss-based TCP Congestion Control Algorithms", *Proceedings of the 6th International Workshop on Protocols for Fast Long-Distance Networks* (PFLDnet 2008), March 5-7,2008, Manchester, UK.
   - Covers CTCP and TCP Illinois
   - Uses CTCP implementation in Windows® VISTA and TCP Illinois implementation within Linux kernel

PARAMETERS

- Number of flows sharing path
- Start time of each flow
- Bottleneck bandwidth
- Round-trip propagation delay
- Number of buffers

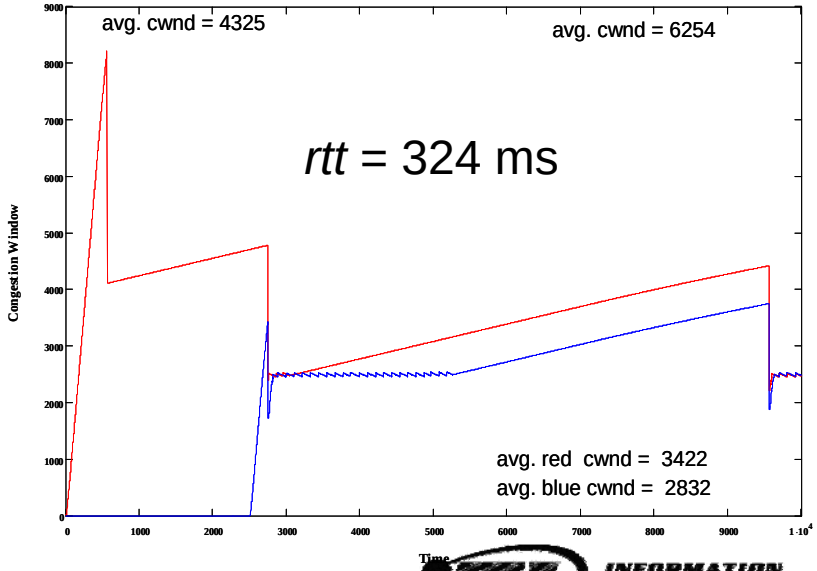
Simulated Behavior of MesoNet CTCP Model

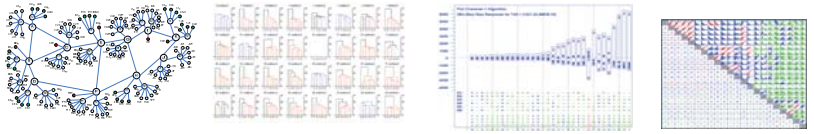




MesoNet CTCP simulation behavior agrees with empirical results

Other MesoNet congestion-control algorithms also agree with empirical results





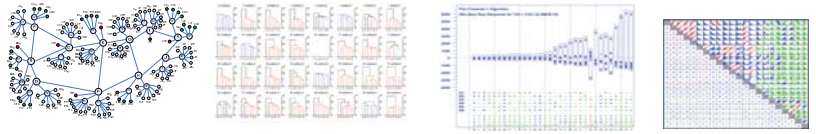
Comparing Alternate Congestion-Control Algorithms in a Large (up to 278,000 sources), Fast (up 192 Gbps) Homogeneous Network

7 Algorithms

Identifier	Label	Name of Congestion-Avoidance Algorithm
1	BIC	Binary Increase Congestion Control
2	CTCP	Compound Transmission Control Protocol
3	FAST	Fast Active-Queue Management Scalable Transmission Control Protocol
4	HSTCP	High-Speed Transmission Control Protocol
5	HTCP	Hamilton Transmission Control Protocol
6	Scalable	Scalable Transmission Control Protocol
7	TCP	Transmission Control Protocol (Reno)

3 Path Classes

Path Class	Flow Type	Definition
Very Fast	DD	Source & receiver under directly connected access router
Fast	DF	Source or receiver under directly connected access router and correspondent under fast access router
	FF	Source & receiver under fast access router
Typical	DN	Source or receiver under directly connected access router and correspondent under normal access router
	FN	Source or receiver under fast access router and correspondent under normal access router
	NN	Source & receiver under normal access router



Input Factors & Network Parameters

6 Robustness Factors

Identifier	Definition	PLUS (+1) Value	Minus (-1) Value
x1	Network Speed	8000	4000
x2	Think Time	5000	2500
x3	Source Distribution	Uniform (.33/.33/.33)	Skewed (.1/.6/.3)
x4	Propagation Delay	2	1
x5	File Size	100	50
x6	Buffer Sizing Algorithm	RTTxCapacity	RTTxCapacity/SQR(*N*)

Router Speeds

Router	PLUS (+1)	Minus (-1)
Backbone	192 Gbps	96 Gbps
POP	24 Gbps	12 Gbps
Normal Access	2.4 Gbps	1.2 Gbps
Fast Access	4.8 Gbps	2.4 Gbps
Directly Connected Access	24 Gbps	12 Gbps

Number of Sources

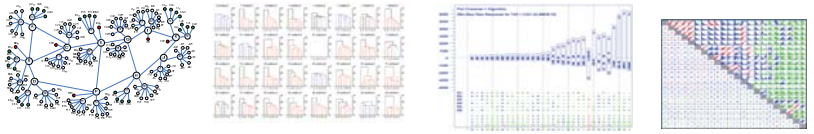
PLUS (+1)	Minus (-1)
278,000	174,600

Propagation Delays (ms)

	Min	Avg	Max
PLUS (+1)	12	81	200
Minus (-1)	6	41	100

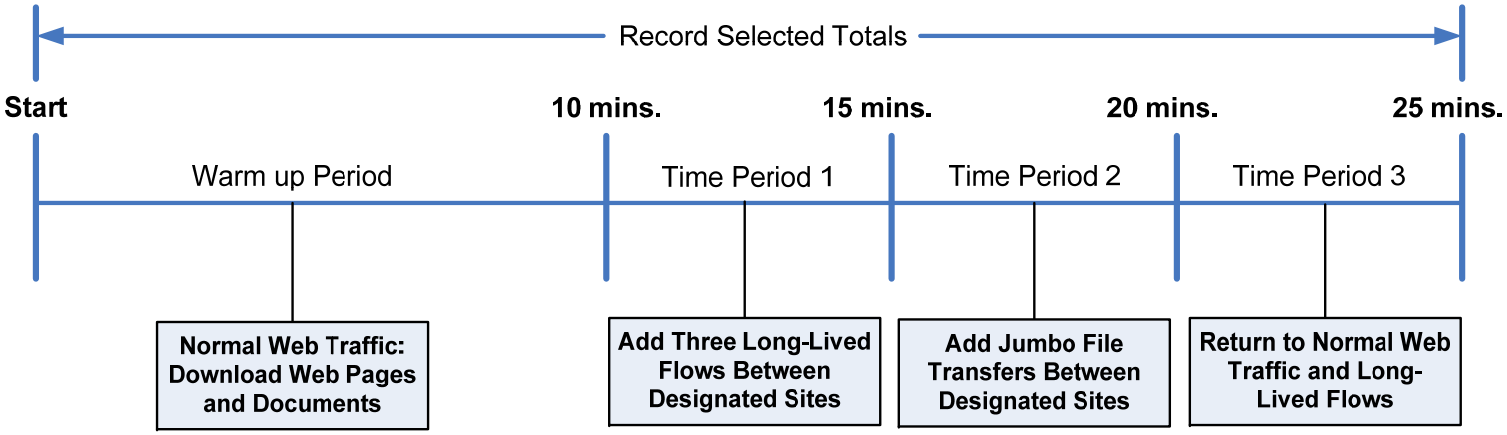
Buffer Sizes (packets)

	PLUS (+1)			Minus (-1)		
Router	Min	Avg	Max	Min	Avg	Max
Backbone	325,528	732,437	1,302,110	1,153	2,606	4,654
POP	40,691	91,555	162,764	221	505	908
Access	6,470	14,557	25,879	91	207	369



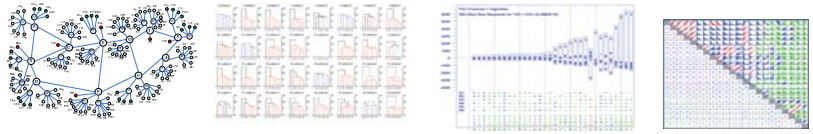
Simulation Scenario & Long-Lived Flows

Scenario



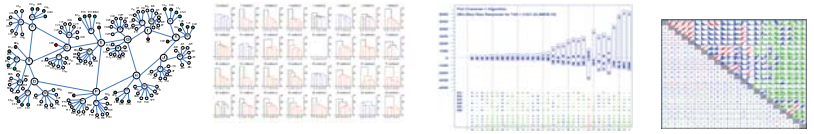
Long-Lived Flows

Identifier	Definition	Source Router	Receiver Router	Start Time
L1	Long-distance flow	B0a	K0a	0.4 x 25 mins.
L2	Medium-distance flow	C0a	I0a	0.4 x 25 mins.
L3	Short-distance flow	E0a	F0a	0.4 x 25 mins.



32 Conditions Simulated (2^{6-1} OFF Design)

Factor->	X1	X2	X3	X4	X5	X6
Condition	--	--	--	--	--	--
1	4000	2500	.1/.6/.3	1	50	RTTxCapacity/SQR(*N*)
2	8000	2500	.1/.6/.3	1	50	RTTxCapacity
3	4000	5000	.1/.6/.3	1	50	RTTxCapacity
4	8000	5000	.1/.6/.3	1	50	RTTxCapacity/SQR(*N*)
5	4000	2500	.3/.3/.3	1	50	RTTxCapacity
6	8000	2500	.3/.3/.3	1	50	RTTxCapacity/SQR(*N*)
7	4000	5000	.3/.3/.3	1	50	RTTxCapacity/SQR(*N*)
8	8000	5000	.3/.3/.3	1	50	RTTxCapacity
9	4000	2500	.1/.6/.3	2	50	RTTxCapacity
10	8000	2500	.1/.6/.3	2	50	RTTxCapacity/SQR(*N*)
11	4000	5000	.1/.6/.3	2	50	RTTxCapacity/SQR(*N*)
12	8000	5000	.1/.6/.3	2	50	RTTxCapacity
13	4000	2500	.3/.3/.3	2	50	RTTxCapacity/SQR(*N*)
14	8000	2500	.3/.3/.3	2	50	RTTxCapacity
15	4000	5000	.3/.3/.3	2	50	RTTxCapacity
16	8000	5000	.3/.3/.3	2	50	RTTxCapacity/SQR(*N*)
17	4000	2500	.1/.6/.3	1	100	RTTxCapacity
18	8000	2500	.1/.6/.3	1	100	RTTxCapacity/SQR(*N*)
19	4000	5000	.1/.6/.3	1	100	RTTxCapacity/SQR(*N*)
20	8000	5000	.1/.6/.3	1	100	RTTxCapacity
21	4000	2500	.3/.3/.3	1	100	RTTxCapacity/SQR(*N*)
22	8000	2500	.3/.3/.3	1	100	RTTxCapacity
23	4000	5000	.3/.3/.3	1	100	RTTxCapacity
24	8000	5000	.3/.3/.3	1	100	RTTxCapacity/SQR(*N*)
25	4000	2500	.1/.6/.3	2	100	RTTxCapacity/SQR(*N*)
26	8000	2500	.1/.6/.3	2	100	RTTxCapacity
27	4000	5000	.1/.6/.3	2	100	RTTxCapacity
28	8000	5000	.1/.6/.3	2	100	RTTxCapacity/SQR(*N*)
29	4000	2500	.3/.3/.3	2	100	RTTxCapacity
30	8000	2500	.3/.3/.3	2	100	RTTxCapacity/SQR(*N*)
31	4000	5000	.3/.3/.3	2	100	RTTxCapacity/SQR(*N*)
32	8000	5000	.3/.3/.3	2	100	RTTxCapacity



Processor Time Requirements

(Units are CPU days)

	Compute Servers ws11-ws14						Compute Servers ws9-ws10		
	BIC	CTCP	FAST	HTCP	TCP	Totals	HSTCP	Scalable	Totals
CPU time (32 runs)	91.5	97.2	93.4	96.4	94.2	472.5	108.6	110.5	219.1
Avg. CPU time (per run)	2.86	3.04	2.92	3.01	2.94	14.77	3.39	3.46	13.70 (6.85x2)
Min. CPU time (one run)	1.16	1.33	1.44	1.40	1.28		1.61	1.51	
Max. CPU time (one run)	5.94	5.85	5.17	5.84	5.63	28.42	6.57	6.61	26.37 (13.18x2)

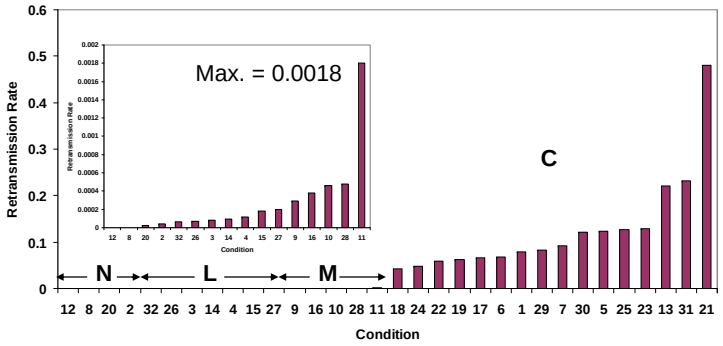
Experiment required about 15 days of wall-clock time spread over 48 processors

Simulated Workload

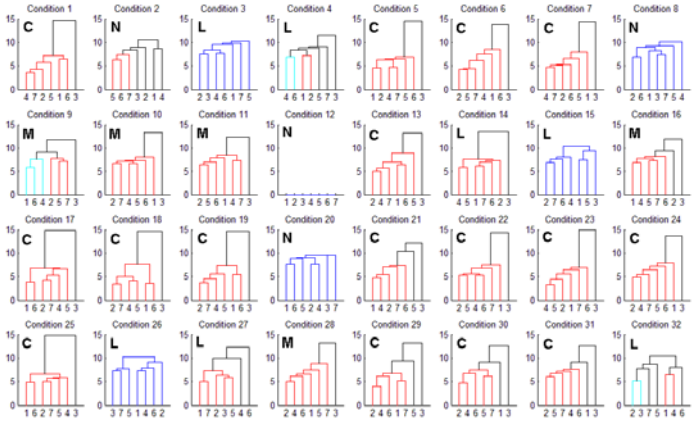
Statistic	Flows Completed	Data Packets Sent
Avg. Per Condition	74,033,116	6,912,373,746
Min. Per Condition	40,966,013	3,146,870,571
Max. Per Condition	154,914,953	11,917,420,154
Total All Runs	16,583,418,069	1,548,371,719,084

Characterizing the 32 Conditions Simulated

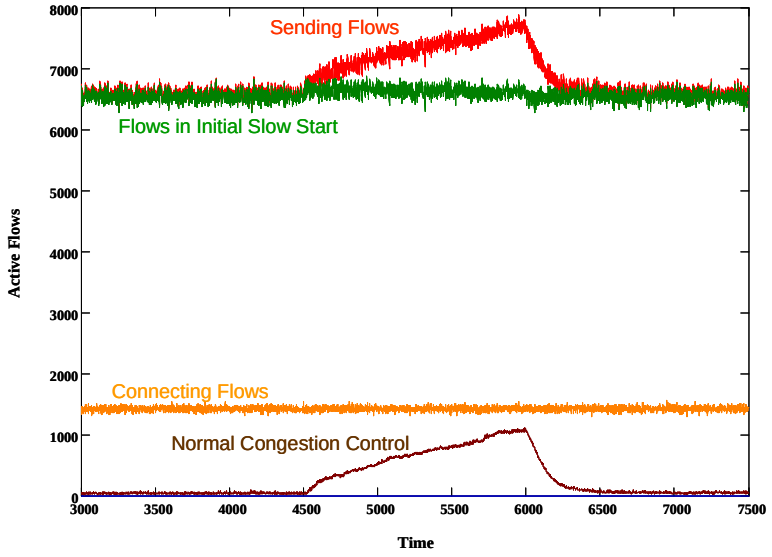
16 Uncongested and 16 Congested Conditions



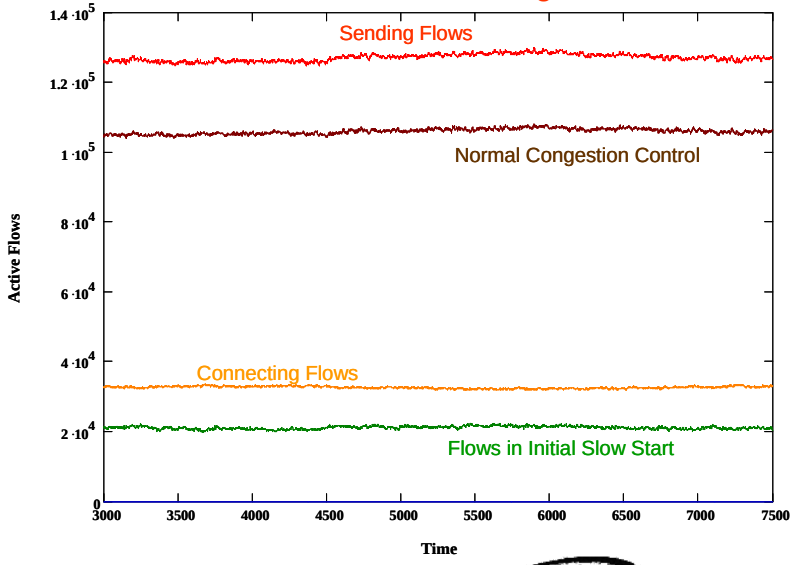
Cluster Analysis Annotated with Congestion Level



Evolution of Flow States in Uncongested Condition 4

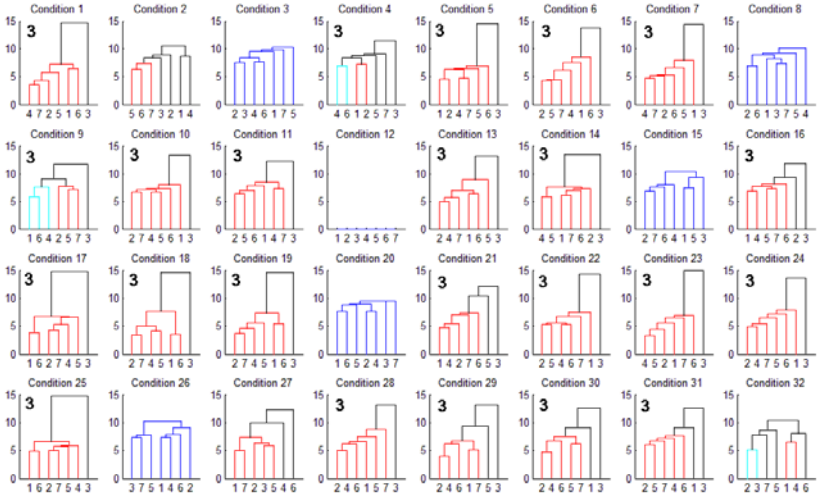


Evolution of Flow States in Congested Condition 5

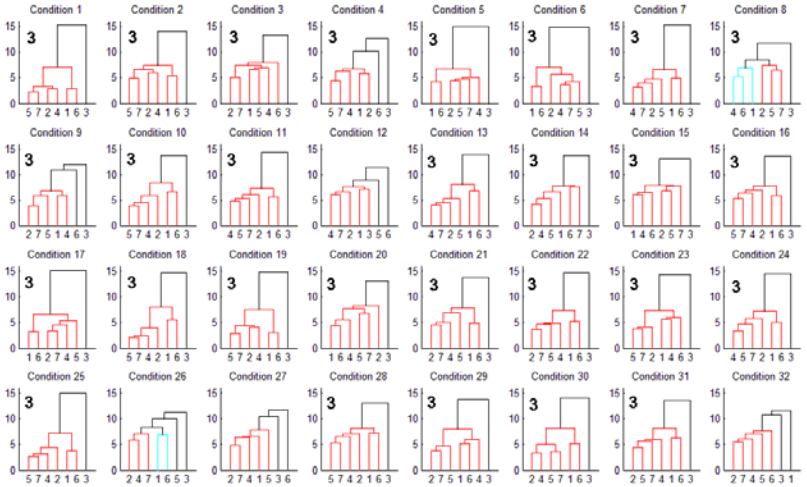


Cluster Analyses – Algorithm 3 (FAST) Stands Out

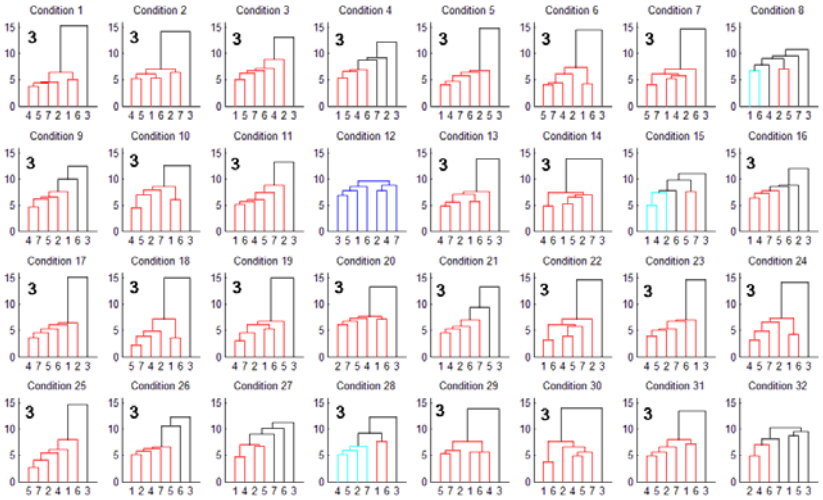
Time Period 1



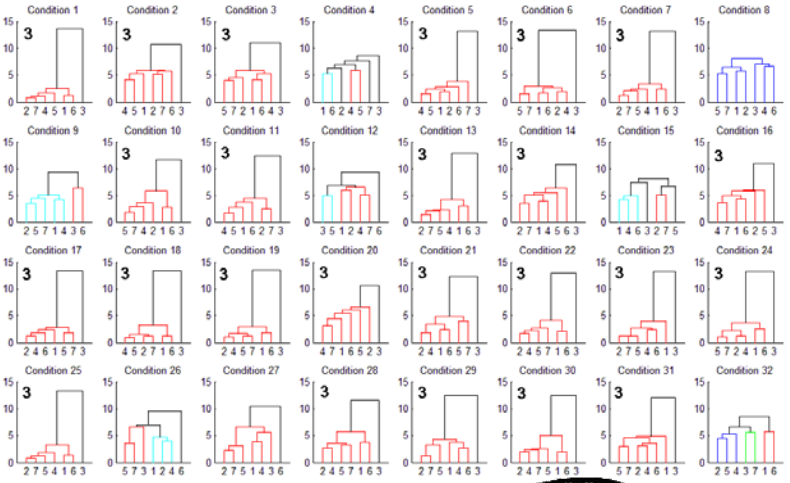
Time Period 2



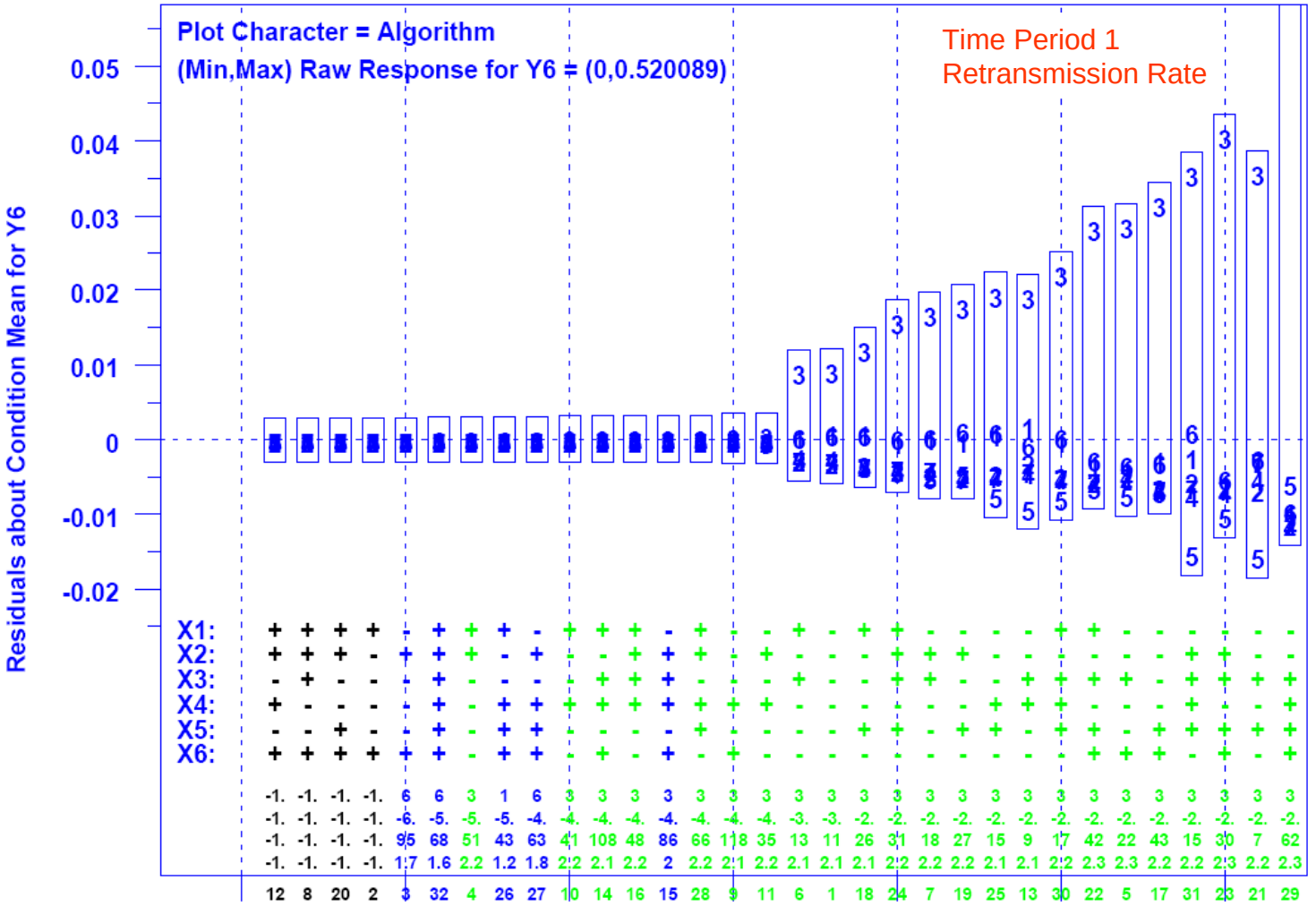
Time Period 3



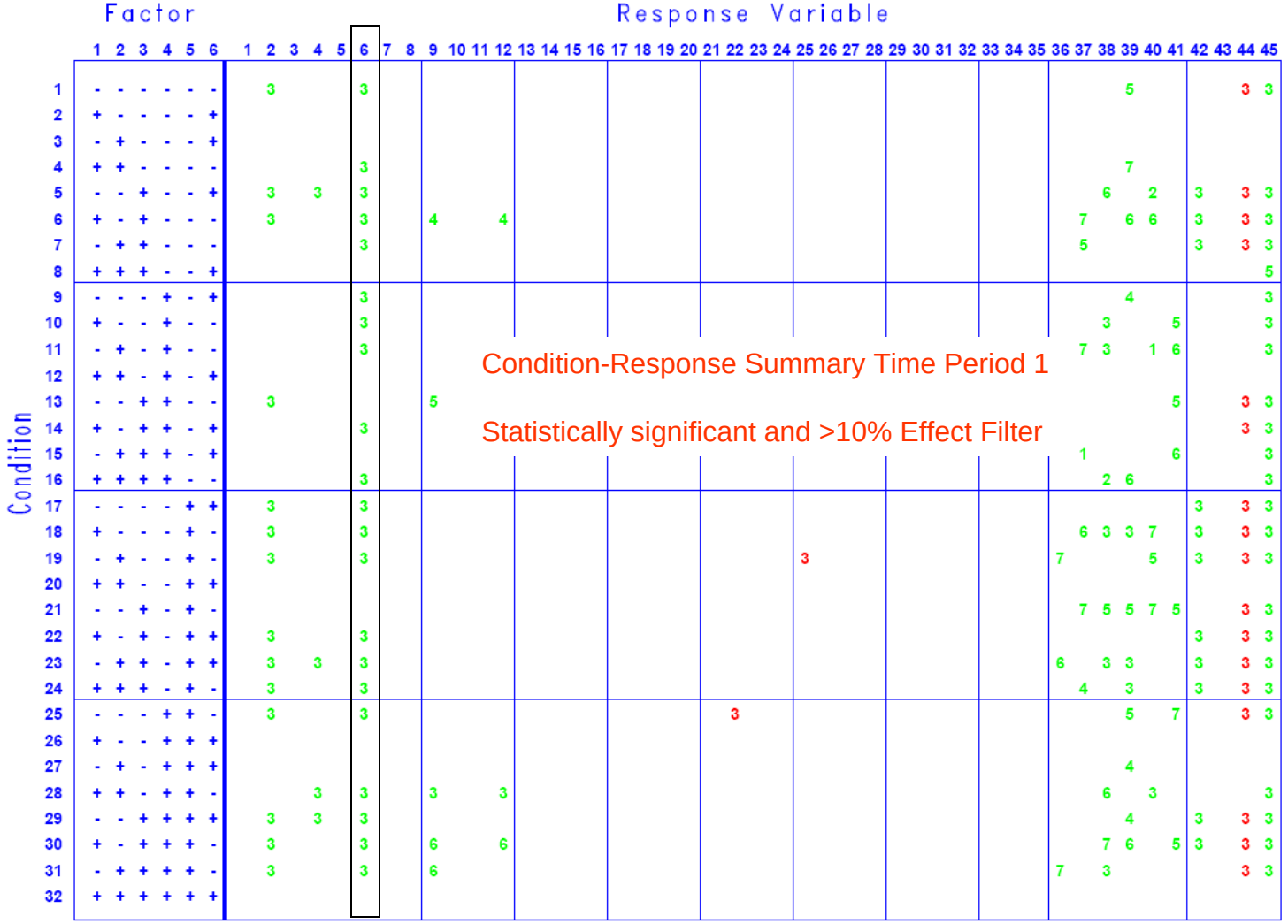
Aggregate Responses over 25 minutes



Approach to Detailed Analysis of Individual Responses

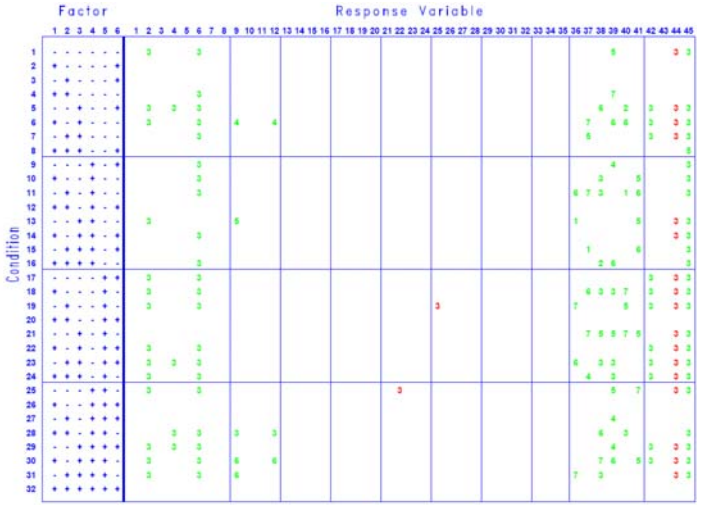


Approach to Summarizing Detailed Analyses of Responses

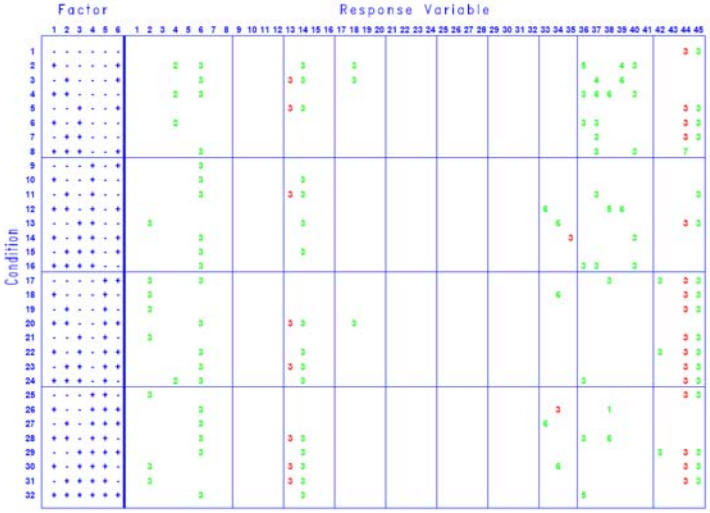


Condition-Response Summaries – Algorithm 3 (FAST) Stands Out

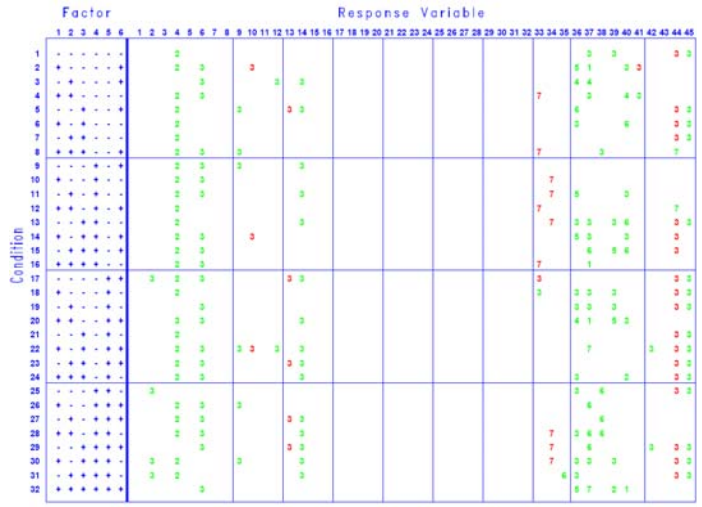
Time Period 1 – 10% Filter



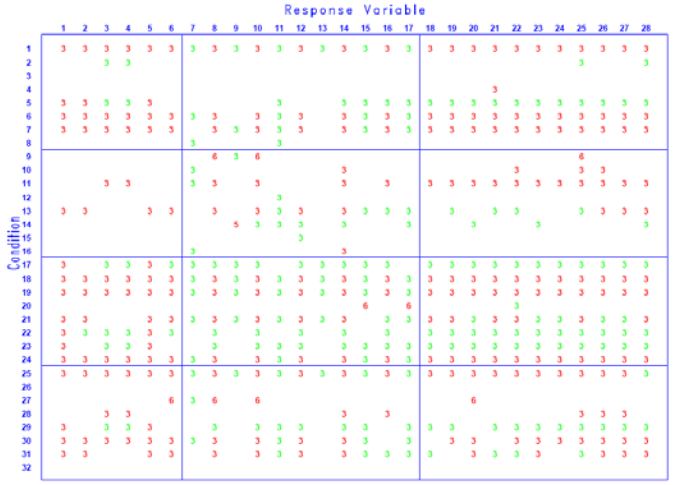
Time Period 2 – 30% Filter



Time Period 3 – 30% Filter

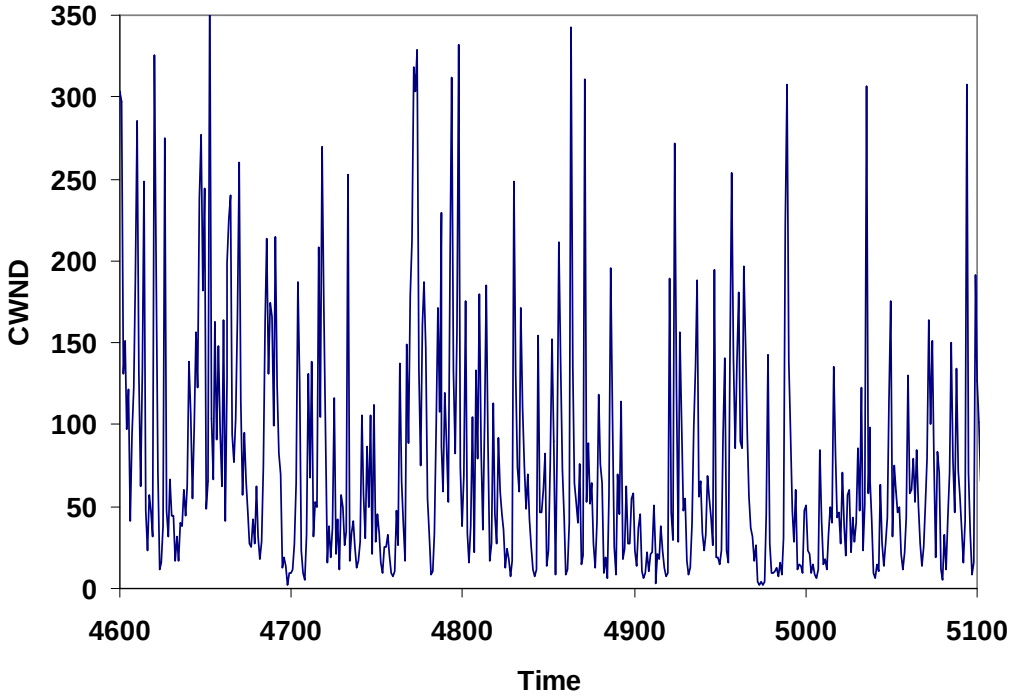


Aggregate Responses – No Filter



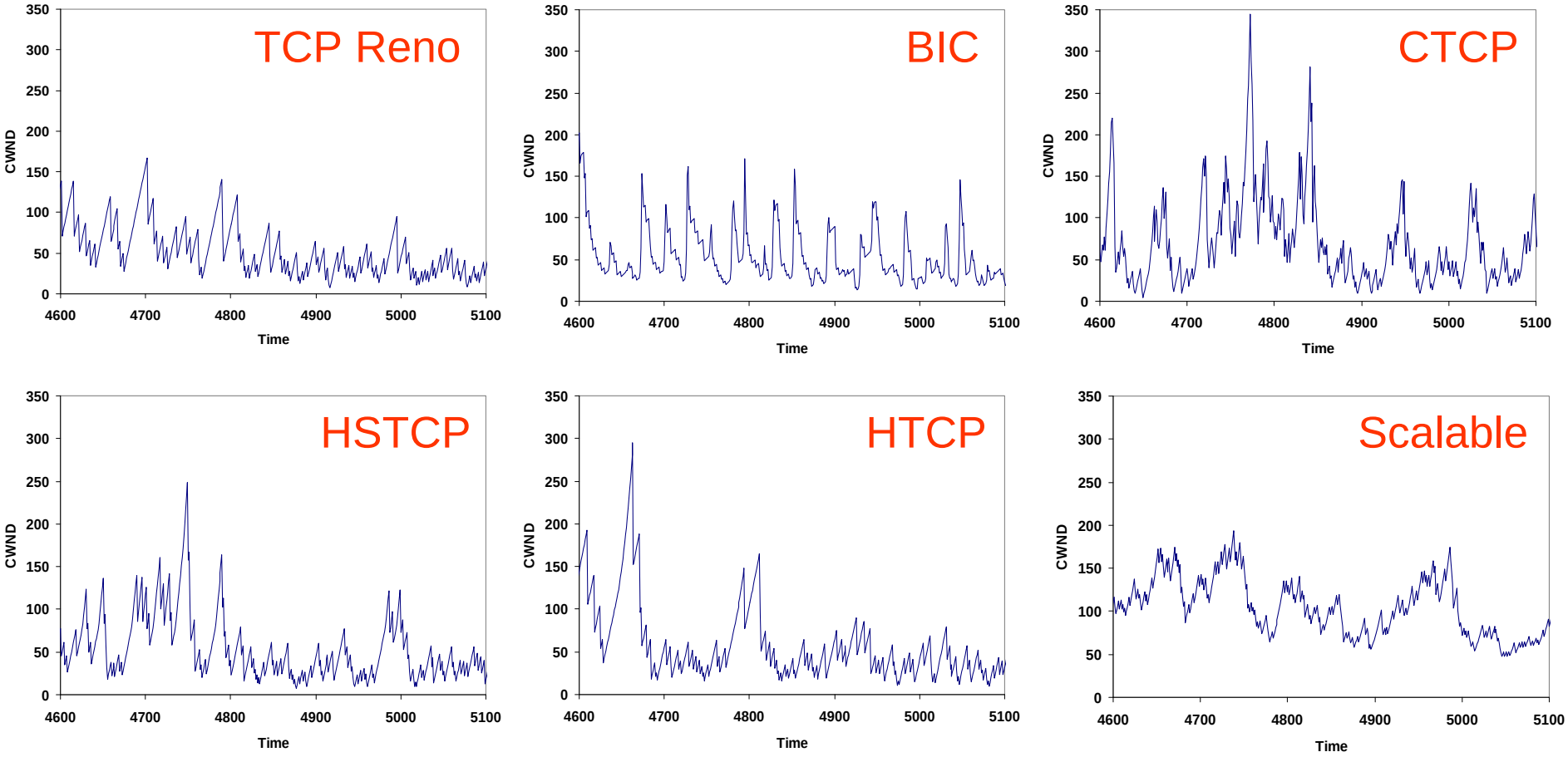
Why does Algorithm 3 (FAST) Stand Out?

FAST does not respond well to congestion, where too many flows compete for insufficient buffers – leading to rapid oscillation in flow congestion windows

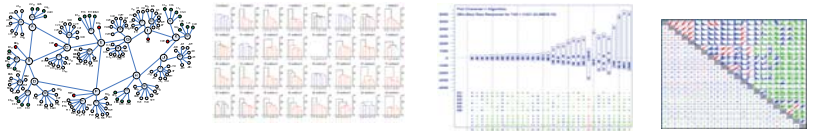


Evolution of congestion window under FAST for long-lived flow L2 during 500 measurement intervals within Time Period 2 under (the most congested) condition 21

Other Congestion-Control Algorithms Adjust Less Rapidly

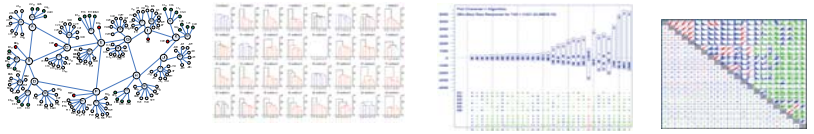


Evolution of congestion window under other congestion-control algorithms for long-lived flow L2 during the same 500 measurement intervals within Time Period 2 under (the most congested) condition 21



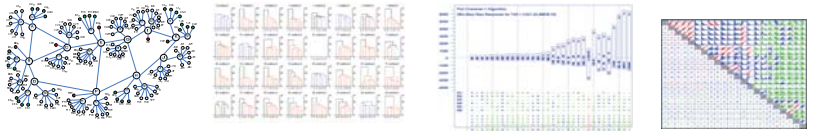
Summary of FAST Behavior under Congestion

- Rapid oscillatory adjustment of congestion window leads to:
  - Larger congestion-window increase rate
  - Higher retransmission rate (including for SYNs)
  - Larger number of flows pending in the connecting state
- Practical implications include:
  - Flows take longer to connect
  - Flows take longer to complete
  - Goodput is lower for flows transiting congested areas
  - Fewer (10^5 to 10^7, depending on condition) flows complete in a 25 minute period



Future Work on Congestion-Control Algorithms

- Does MesoNet reveal the same behavior when modeling a smaller, slower network with a much lower initial slow-start threshold?
- How do the congestion-control algorithms compare in a relatively uncongested, heterogeneous network with a wider range of traffic classes (e.g., web objects, documents, service packs and movie downloads)?
- Does the sensitivity analysis change when considering all 20 MesoNet parameters?



Summary of Presentation

- Introduced NIST project to develop Measurement Science for Complex Information Systems
- Showed an application of measurement science to compare seven alternate congestion-control algorithms for the Internet
- Identified a potential for FAST algorithm to behave undesirably under congested conditions
  - Explained the root cause for the potential undesirable behavior
  - Explained why other congestion-control algorithms are not likely to exhibit the same potential undesirable behavior