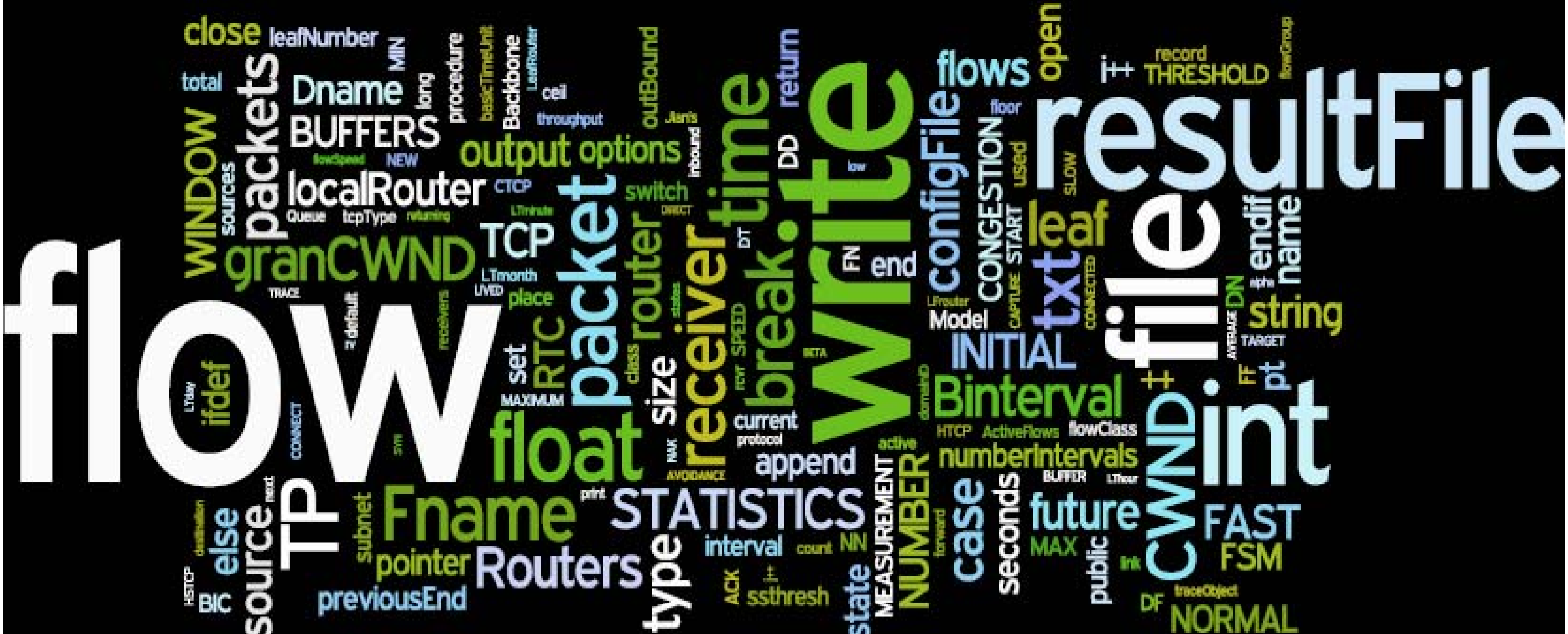
How to model a TCP/IP network using only 20 parameters



visual hash of *MesoNet* source code from http://www.wordle.net/

Kevin Mills, NIST

CxS Study Group – November 17, 2009

Outline

- Goal – Problem – Solution
- Scale Reduction: Theory and Practice
- Overview of the 20 *MesoNet* Parameters
- Parameter Explanations in 5 Categories
  - Network (4 parameters)
  - Sources & Receivers (4 parameters)
  - User Behavior (6 parameters)
  - Protocols (3 parameters)
  - Simulation & Measurement Control (3 parameters)
- Discuss Simulation Resource Requirements
- Future Work

Goal – Problem – Solution

- **Goal** – compare proposed Internet congestion control algorithms under a wide range of controlled, repeatable conditions

- **Problem** – real network not controllable and repeatable; test beds *currently* too small; *most* network simulation models have large search space and require infeasible memory and processing resources for large, fast networks; more tractable fluid-flow simulators are *currently* inaccurate

- **Solution** – design a reduced scale network simulation model (***MesoNet***) that is easy to configure and tractable to compute

Joint work with Jian Yuan and Edward Schwartz

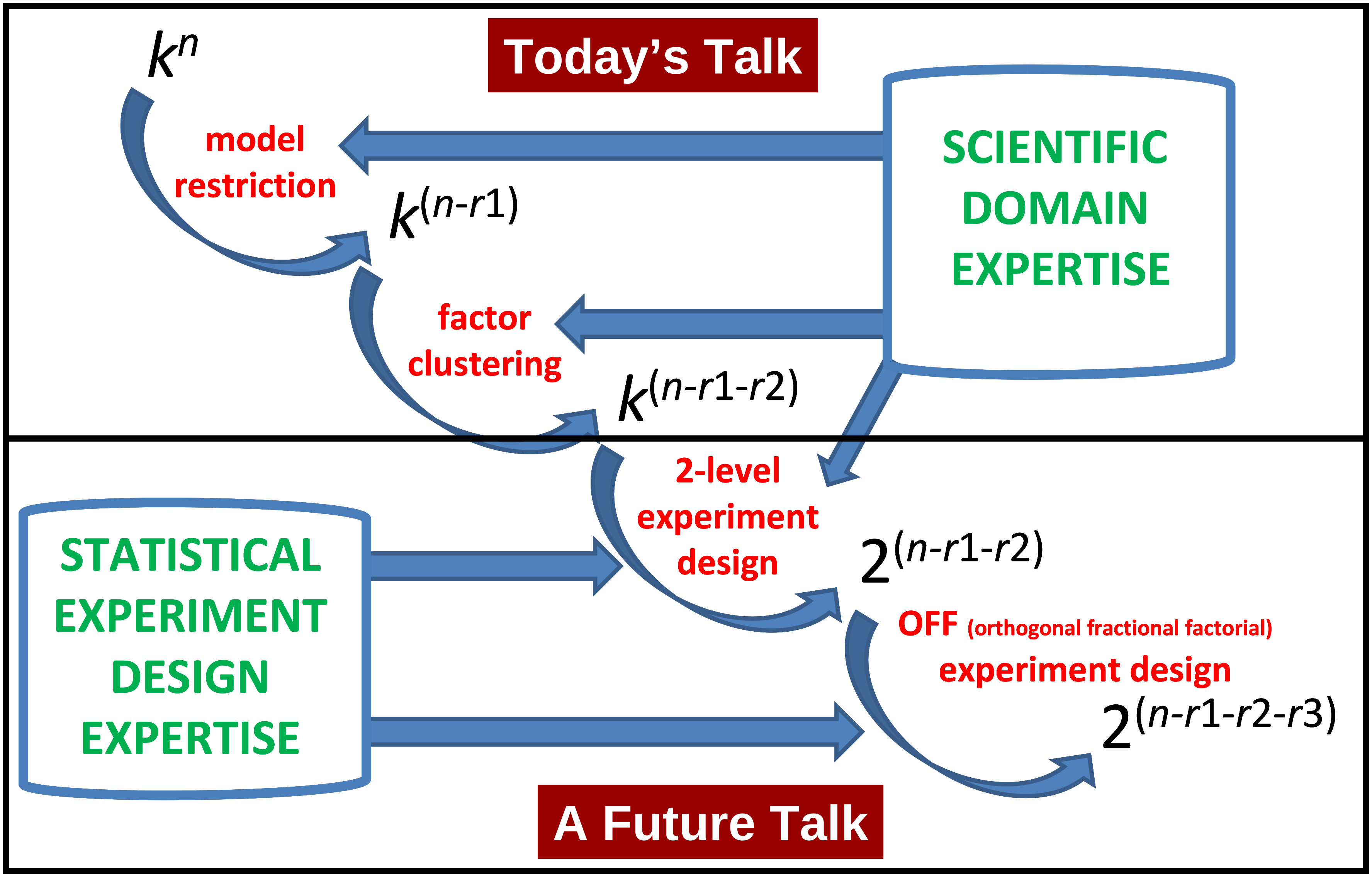
The Function ☺ of a Simulation Model

$$\underbrace{y_1, \ldots, y_m}_{\text{Response State-Space}} = f(\underbrace{x_{1|[1,\ldots,k]}, \ldots, x_{n|[1,\ldots,k]}}_{\text{Stimulus State-Space}})$$

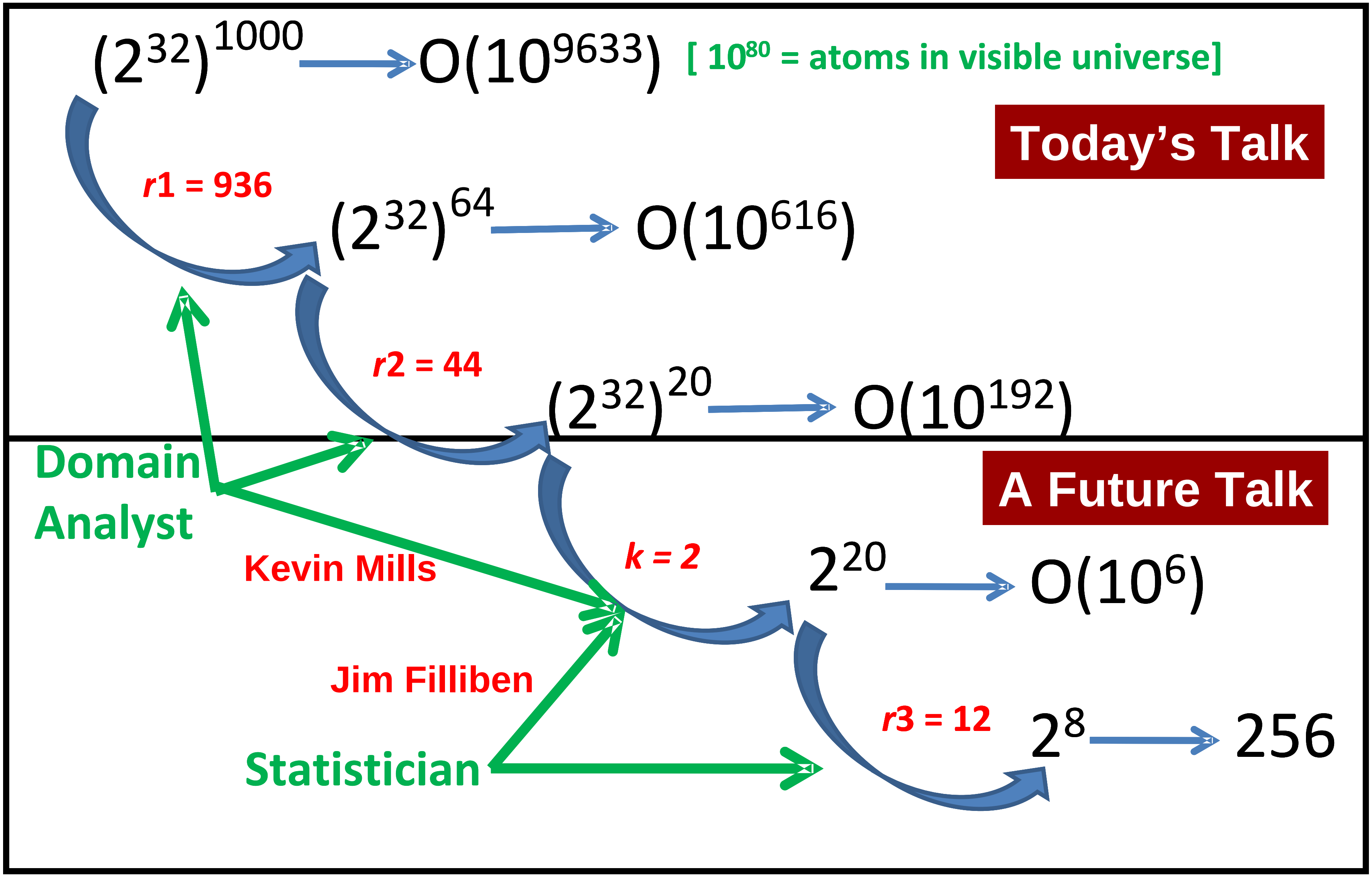
Response State-Space

Stimulus State-Space

Theory – Scale Reduction in Two Parts



Practice – Scale Reduction in Two Parts

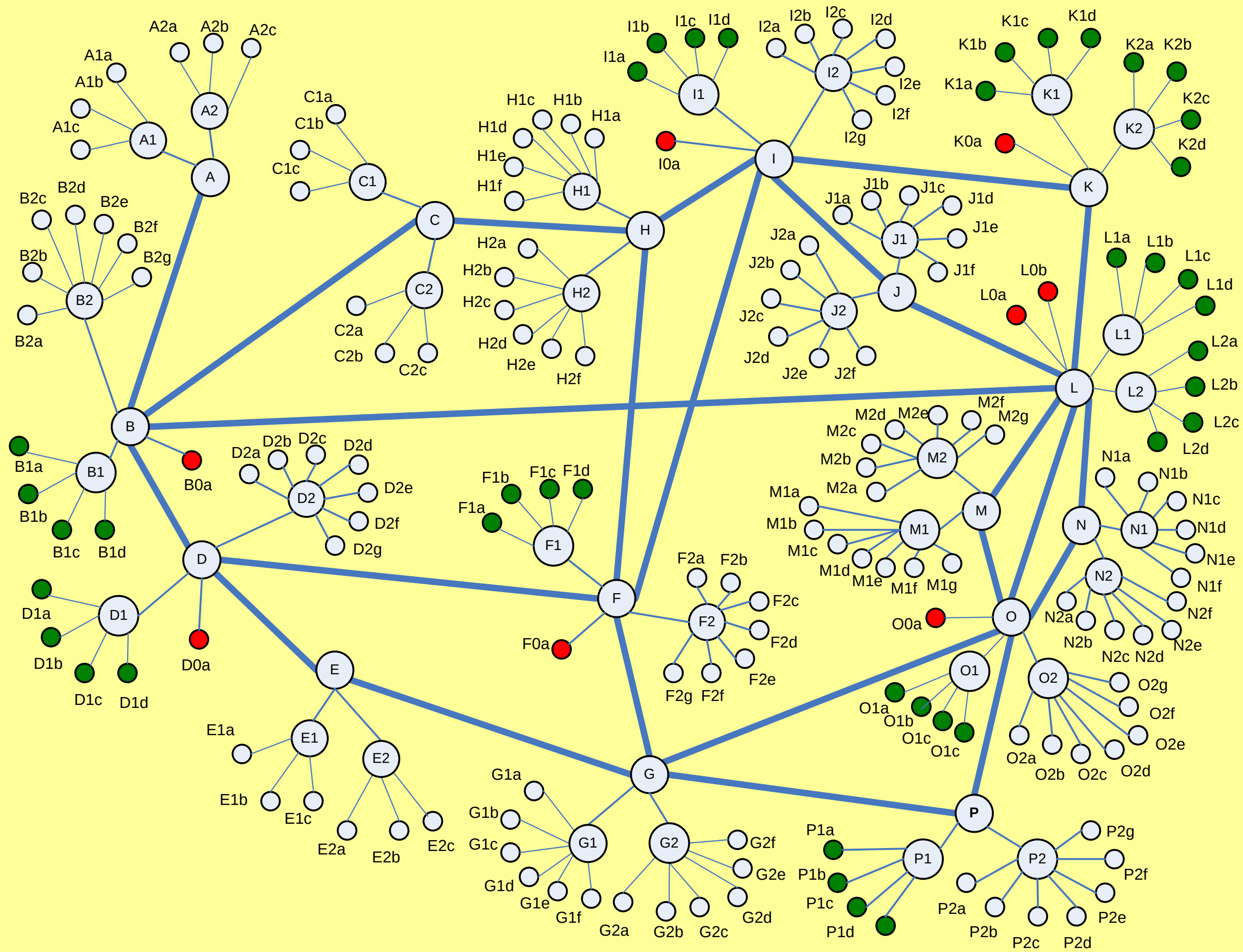


MesoNet – a TCP/IP network model using only 20 parameters

x1	**Network Speed**	Network Parameters
x2	**Propagation Delay**	
x3	**Buffer Provisioning**	
x4	**Topology**	
x5	**Source & Receiver Interface Speeds**	Sources & Receivers
x6	**Number of Sources & Receivers**	
x7	**Distribution of Sources**	
x8	**Distribution of Receivers**	
x9	**Web browsing File Sizes**	User Behavior
x10	**Larger download Proportion & File Sizes**	
x11	**Think Time**	
x12	**User Patience**	
x13	**Temporal Congestion on Very Fast paths**	
x14	**Long-Lived Flows**	
x15	**Assignment of Congestion-Control Algorithm**	Protocols
x16	**Initial Congestion Window**	
x17	**Initial Slow-Start Theshold**	
x18	**Measurement Interval Size**	Simulation & Measurement Control
x19	**Simulation Duration**	
x20	**Startup Pattern**	

Network Parameters

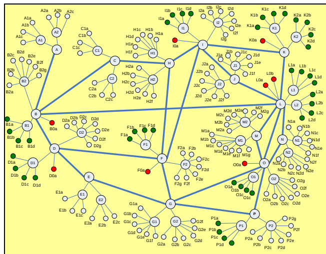
Parameter X4 is the Topology = Routers + Links + Routes + Propagation Delays



3 Router Tiers: Backbone – Point of Presence (PoP) – Access

3 Access Router Classes: Typical – Fast – Directly Connected

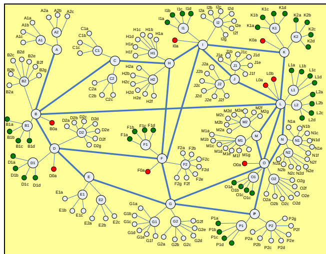
1 ingress/egress path from access routers to backbone routers



Link Characteristics – Propagation Delay and Cost Metric

Link #	Link Endpoints	Delay (ms)	Cost Metric
1	A-B	21	50
2	B-C	25	10
3	B-D	8	50
4	B-L	75	223
5	C-H	12	100
6	D-E	10	10
7	D-F	33	108
8	E-G	33	100
9	F-G	7	10
10	F-H	12	50
11	F-I	22	55
12	G-O	23	104
13	G-P	19	110
14	I-H	14	10
15	I-J	8	50
16	I-K	22	147
17	J-L	20	60
18	K-L	7	50
19	L-M	12	50
20	L-N	6	39
21	L-O	14	10
22	M-O	6	10
23	N-O	8	10
24	O-P	14	10

- Packets incur propagation delay when transiting a link
- Cost metric used to compute routes from source backbone router to destination backbone router



Dijkstra's SPF Backbone Routes Computed from Cost Metrics

# Backbone Paths	Avg. Backbone Path Length	Avg. Source-Receiver Path Length
240	3.63	9.53

Sample Backbone Paths for Routers A Through H

PATHS FOR NODE A
A:B @ A->B $ 50
A:C @ A->B->C $ 60
A:D @ A->B->D $ 100
A:E @ A->B->D->E $ 110
A:F @ A->B->D->F $ 208
A:G @ A->B->D->E->G $ 210
A:H @ A->B->C->H $ 160
A:I @ A -> B -> C -> H -> I $ 170
A:J @ A->E $220
A:K @ A->B->C->H->I->K $ 317
A:L @ A->B->L $ 273
A:M @ A->B->L->M $ 323
A:N @ A->B->L->N $ 312
A:O @ A->B->D->E->G->O $ 314
A:P @ A->B->D->E->G->P $ 320

PATHS FOR NODE B
B:A @ B->A $ 50
B:C @ B->C $ 10
B:D @ B->D $ 50
B:E @ B->D->E $ 60
B:F @ B->D->F $ 158
B:G @ B->D->E->G $ 160
B:H @ B->C->H $ 110
B:I @ B -> C -> H -> I $ 120
B:J @ B->C $170
B:K @ B->C->H->I->K $ 267
B:L @ B->L $223
B:M @ B->L->M $ 273
B:N @ B->L->N $ 262
B:O @ B->D->E->G->O $ 264
B:P @ B->D->E->G->P $ 270

PATHS FOR NODE C
C:A @ C->B->A $ 60
C:B @ C->B $ 10
C:D @ C->B->D $ 60
C:E @ C->B->D->E $ 70
C:F @ C->H->F $ 150
C:G @ C->H->F->G $ 160
C:H @ C->H $ 100
C:I @ C -> H -> I $ 110
C:J @ C->H $160
C:K @ C->H->I->K $ 257
C:L @ C->H $220
C:M @ C->H->I->J->L->M $ 270
C:N @ C->H->I->J->L->N $ 259
C:O @ C->H->F->G->O $ 264
C:P @ C->H->F->G->P $ 270

PATHS FOR NODE D
D:A @ D->B->A $ 100
D:B @ D->B $ 50
D:C @ D->B->C $ 60
D:E @ D->E $ 10
D:F @ D->F $ 108
D:G @ D->E->G $ 110
D:H @ D->F->H $ 158
D:I @ D -> F -> I $ 163
D:J @ D->F $213
D:K @ D->F->I->K $ 310
D:L @ D->E->G->O->N->L $ 263
D:M @ D->E->G->O->M $ 224
D:N @ D->E->G->O->N $ 224
D:O @ D->E->G->O $ 214
D:P @ D->E->G->P $ 220

PATHS FOR NODE E
E:A @ E->D->B->A $ 110
E:B @ E->D->B $ 60
E:C @ E->D->B->C $ 70
E:D @ E->D $ 10
E:F @ E->G->F $ 110
E:G @ E->G $ 100
E:H @ E->G->F->H $ 160
E:I @ E -> G -> F -> I $ 165
E:J @ E->G $215
E:K @ E->G->O->N->L->K $ 303
E:L @ E->G->O->N->L $ 253
E:M @ E->G->O->M $ 214
E:N @ E->G->O->N $ 214
E:O @ E->G->O $ 204
E:P @ E->G->P $ 210

PATHS FOR NODE F
F:A @ F->D->B->A $ 208
F:B @ F->D->B $ 158
F:C @ F->H->C $ 150
F:D @ F->D $ 108
F:E @ F->G->E $ 110
F:G @ F->G $ 10
F:H @ F->H $ 50
F:I @ F -> I $ 55
F:J @ F -> I -> J $ 105
F:K @ F->I -> K $ 202
F:L @ F->G->O->N->L $ 163
F:M @ F->G->O->M $ 124
F:N @ F->G->O->N $ 124
F:O @ F->G->O $ 114
F:P @ F->G->P $ 120

PATHS FOR NODE G
G:A @ G->E->D->B->A $ 210
G:B @ G->E->D->B $ 160
G:C @ G->F->H->C $ 160
G:D @ G->E->D $ 110
G:E @ G->E $ 100
G:F @ G->F $ 10
G:H @ G->F->H $ 60
G:I @ G -> F -> I $ 65
G:J @ G->F $115
G:K @ G->O->N->L->K $ 203
G:L @ G->O->N->L $ 153
G:M @ G->O->M $ 114
G:N @ G->O->N $ 114
G:O @ G->O $ 104
G:P @ G->P $ 110

PATHS FOR NODE H
H:A @ H->C->B->A $ 160
H:B @ H->C->B $ 110
H:C @ H->C $ 100
H:D @ H->F->D $ 158
H:E @ H->F->G->E $ 160
H:F @ H->F $ 50
H:G @ H->F->G $ 60
H:I @ H -> I $ 10
H:J @ H->I -> J $60
H:K @ H-> $157
H:L @ H->I -> L$ 120
H:M @ H->I->J->L->M $ 170
H:N @ H->I->J->L->N $ 159
H:O @ H->F->G->O $ 164
H:P @ H->F->G->P $ 170

Parameter x2 scales all propagation delays

Link #	Link Endpoints	Delay (ms)	x2 = 2	x2 = 0.5
1	A-B	21	42	10.5
2	B-C	25	50	12.5
3	B-D	8	16	4
4	B-L	75	150	37.5
5	C-H	12	24	6
6	D-E	10	20	5
7	D-F	33	66	16.5
8	E-G	33	66	16.5
9	F-G	7	14	3.5
10	F-H	12	24	6
11	F-I	22	44	11
12	G-O	23	46	11.5
13	G-P	19	38	9.5
14	I-H	14	28	7
15	I-J	8	16	4
16	I-K	22	44	11
17	J-L	20	40	10
18	K-L	7	14	3.5
19	L-M	12	24	6
20	L-N	6	12	3
21	L-O	14	28	7
22	M-O	6	12	3
23	N-O	8	16	4
24	O-P	14	28	7

Engineering Relationship Among Router Speeds

(*MesoNet* simplification – only routers have speeds)

Speed Relationships

Router Class	Speed
Backbone	r1 x BBspeedup
PoP	r1/r2
Access	r1/r2/r3
Fast Access	r1/r2/r3 x Bfast
Directly Connected Access	r1/r2/r3 x Bdirect

Sample Parameters Values

Parameter	Value
r1	x1
r2	4
r3	10
BBspeedup	2
Bdirect	10
Bfast	2

Given a fixed r2, r3, BBspeedup, Bdirect and Bfast,

Parameter x1 scales all router speeds

Router Class	Speed	x1 = 800	x1 = 1600
Backbone	r1 x BBspeedup	1600	3200
PoP	r1/r2	400	800
Access	r1/r2/r3	40	80
Fast Access	r1/r2/r3 x Bfast	80	160
Directly Connected Access	r1/r2/r3 x Bdirect	400	800

Domain View of Network Speed

(*MesoNet* simplification – packets have no size)

Assuming speeds dimensioned in packets per milliseconds and packets sized at 1500^ 8-bit bytes:

Router Class	x1 = 800	Speed (Gbps)	x1 = 1600	Speed (Gbps)
Backbone	1600	19.2	3200	38.4
PoP	400	4.8	800	9.6
Access	40	0.48	80	0.96
Fast Access	80	0.96	160	1.92
Directly Connected Access	400	4.8	800	9.6

^default maximum transfer unit size for the Internet is 1500 bytes

Buffer Size Determination

<u>Choice of 4 Buffer Provisioning Algorithms</u>

Buffers = $RTT \times C$ — recommended practice

Buffers = $(RTT \times C)/\sqrt{n}$ — suggested by researchers from Stanford

Buffers = $((RTT \times C) + ((RTT \times C)/\sqrt{n}))/2$ — interpolation

Buffers = <integer value> — directly set

Qfactor = buffer size scaling factor

BufferSize = *Buffers* x *Qfactor*

For selected algorithm, x3 = *Qfactor* scales buffer size

For specified *Qfactor*, x3 = algorithm determines buffer size

RTT = average round-trip propagation delay of the topology
C = capacity of a given router type – determined by network speed settings
n = expected number of flows transiting the router

Sources & Receivers

Approximate Number of Sources & Receivers

baseSources = target^ # sources under each access router
ΔU = scaling factor for target # sources

sources = O(*baseSources* x ΔU x # access routers)
receivers = O(*baseSources* x ΔU x # access routers x 4)

baseSources = 100 # access routers = 170

ΔU	# sources	# receivers
1	O(17,000)	O(68,000)
3	O(51,000)	O(204,000)

Setting x6 (= ΔU) scales the number of sources and receivers in the network

^# sources and receivers becomes known exactly only when taking into account the distribution of sources and receivers under various access router types

Access Router Classes & Source Distribution

Access Router Classes		Probability Source Under
N-class	Access routers with typical speed	*probNs*
F-class	Access routers with fast speed	*probNsf*
D-class	Directly connected access routers	1 – *probNs* - *probNsf*

Target # sources under 3 access routers: 1 in each class) = 3 x *baseSources* x ΔU

sources under each N-class router = 3 x *baseSources* x ΔU x *probNs*

sources under each F-class router = 3 x *baseSources* x ΔU x *probNsf*

sources under each D-class router = 3 x *baseSources* x ΔU x (1 - *probNs* - *probNsf)*

Given ΔU = 3, *baseSources* = 100, p*robNs* = 0.1, *probNsf* = 0.6

router class	sources/router	# routers	# sources	% sources
N-class	90	122	10,980	31.6
F-class	540	40	21,600	62.2
D-class	270	8	2,160	6.2

Total Sources in Network 34,740

Setting x7 = (*probNs*, *probNsf*) scales the distribution of sources and adjusts the number of sources in the topology
Setting x8 = (*probNr*, *probNrf*) performs a similar function on receivers

Source/Receiver Distributions Influence Traffic Patterns

Given ΔU = 3, *baseSources* = 100, p*robNs* = 0.1, *probNsf* = 0.6, *probNr* = 0.8, *probNrf* = 0.1

router class	% sources	% receivers
N-class	31.6	95.3
F-class	62.2	3.9
D-class	6.2	0.8

Flow class	% flows
DD flows	0.05
DF flows	0.74
DN flows	6.1
FF flows	2.4
FN flows	60.5
NN flows	30.1

FF, FN, DN and DF flows represent Web-centric traffic and NN flows represent P2P-like traffic

DD flows represent traffic exchange with large data repositories

Source & Receiver Interface Speeds

Assuming packet size is 1500 8-bit bytes:

Example

Interface Speeds	ppms	Mbps
Hbase	8	96
Hfast	80	960

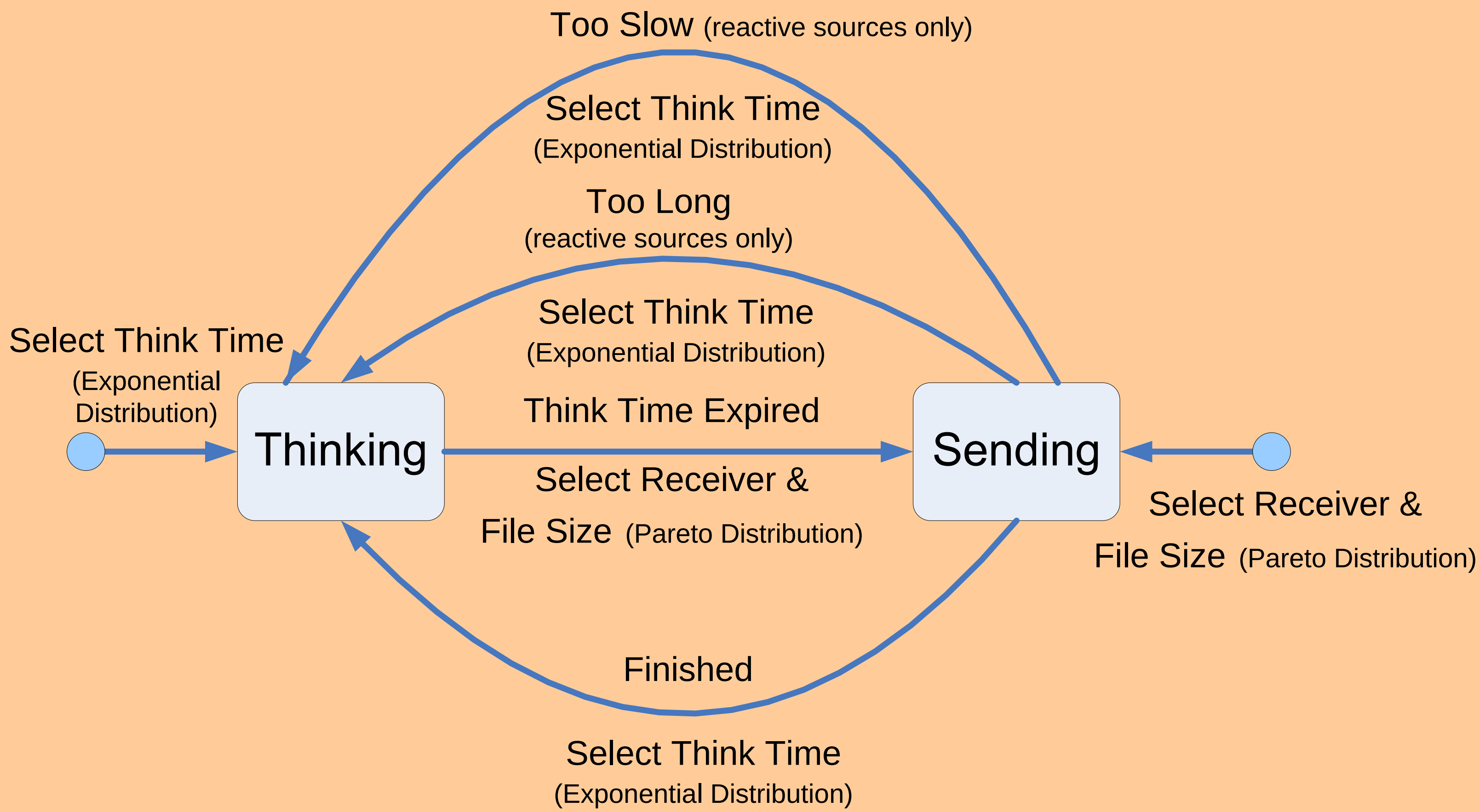
Model permits two interface speeds

x5 specifies the probability that a source/receiver has an *Hfast* interface

(1- x5) gives the probability that a source/receiver has an *Hbase* interface

User Behavior

Source Behavior^



x11 specifies average Think Time **x12 specifies probability source is reactive**

^Note: this simplified diagram omits a flow connection phase that occurs before sending and also the potential for the connection phase to fail – after which source enters Thinking

File Size Selection

Characterize Web Objects (λon, α)

Size of Web Objects

λon	average size (packets)
α	shape of Pareto distribution

Probability of Web Object
(1 – *Fp* – *Sp* – *Mp*)

Characterize Larger Files [(*Fx*, *Fp*), (*Sx*, *Sp*), (*Mx*, *Mp*)]

Larger File Size Multipliers

Fx	documents
Sx	software downloads
Mx	movies

Larger File Probabilities

Fp	documents
Sp	software downloads
Mp	movies

Given a fixed α

x9 specifies λon

Given fixed *Fx*, *Sx*, *Mx*

x10 specifies (*Fp*, *Sp*, *Mp*)

Specifying Intense Spatiotemporal Traffic

During selected time periods, DD flows can transfer *jumbo* files

Jumbo File Characteristics

Jx	size multiplier for jumbo files
Jon	fraction of simulated time after which jumbo file transfers begin
Joff	fraction of simulated time after which jumbo file transfers end

Characterize Jumbo Files (*Jx*, *Jon, Joff*)

Given a fixed *Jx*

x13 specifies (*Jon*, *Joff*)

Specifying Long-Lived Flows

Behavior of each long-lived flow is monitored in detail

Characteristics of Long-Lived Flows

n	number of long-lived flows
LLon[*n*]	fraction of simulated time after which each long-lived flow starts
SourceType[*n*]	congestion control algorithm used by each long-lived flow
SourceLocation[*n*]	access router under which each long-lived source is located
ReceiverLocation[*n*]	access router under which each long-lived receiver is located

x14 specifies (*n, LLon, SourceType, SourceLocation, ReceiverLocation*)

ID	Source Type
0	select from probability distribution (see x15)
1	TCP
2	HSTCP
3	CTCP
4	Scalable TCP
5	FAST
6	HTCP
7	BIC

Protocols

Assignment of Congestion Control Algorithm

Probabilistically assign each source (except for long-lived flows) a congestion control algorithm

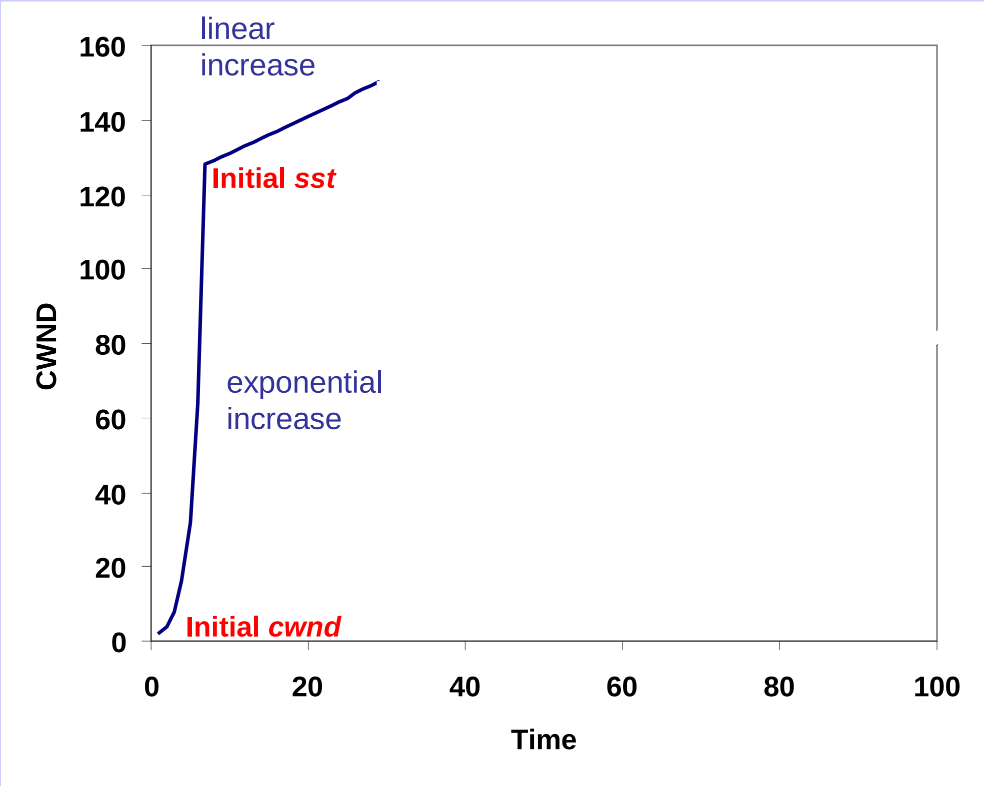
Congestion Control Algorithm	Probability
TCP	*prTCP*
HSTCP	*prHSTCP*
CTCP	*prCTCP*
Scalable TCP	*prSCALABLE*
FAST	*prFAST*
HTCP	*prHTCP*
BIC	*prBICTCP*

x15 specifies (*prTCP***,** *prHSTCP***,** *prCTCP, prSCALABLE, prFAST, prHTCP, prBICTCP***)**

Initial Congestion Window & Slow Start Threshold

Congestion window (*cwnd*) determines how many packets may be sent by a source in one RTT

At Initial Slow Start Threshold (*sst*) *cwnd* increase switches from exponential to linear



Initial *cwnd* determines how many packets may be sent prior to first acknowledgment

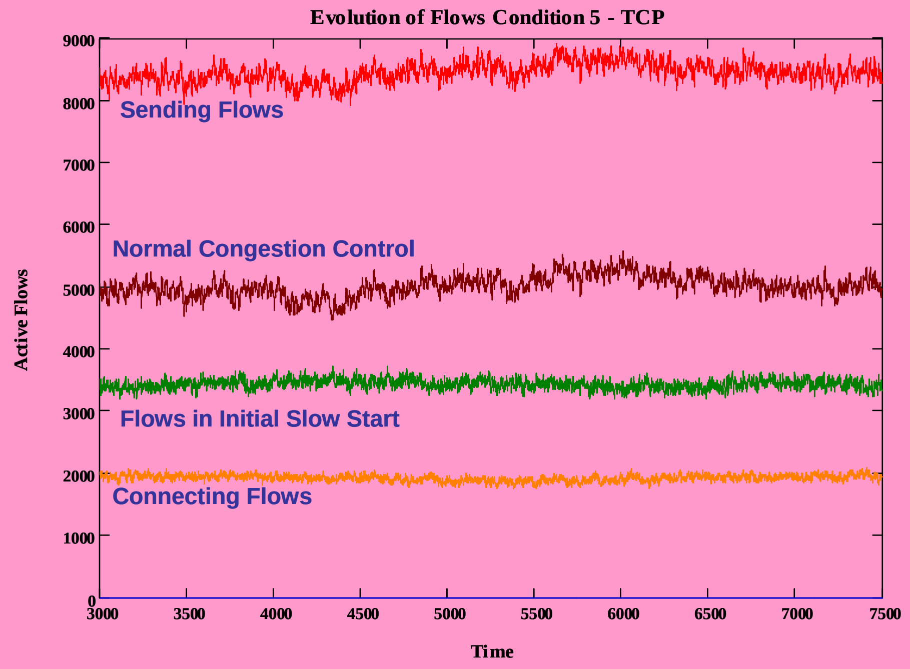
x16 specifies initial *cwnd*

x17 specifies initial *sst*

Simulation & Measurement Control

Measurement Interval Size & Simulation Duration

Measurements taken as time series sampled periodically: every measurement interval



Controlling Measurement & Simulation Length

M	measurement interval size
MI	number of measurement intervals
M x MI	simulation duration

x18 specifies *M*

x19 specifies *MI*

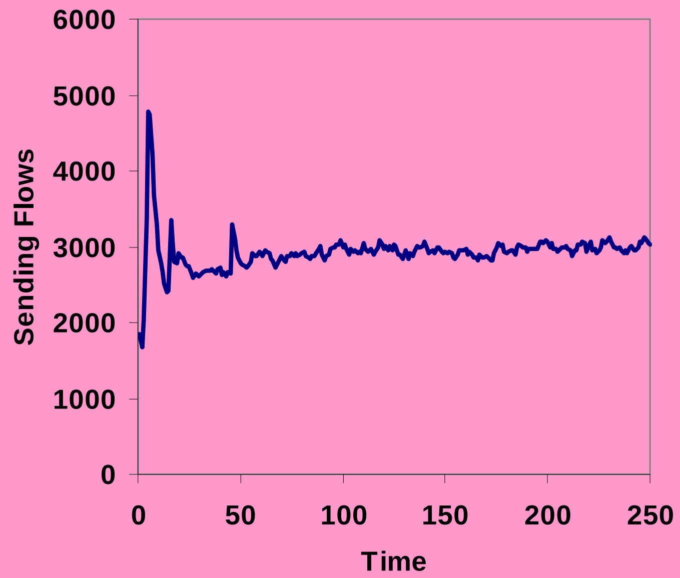
Source Startup Pattern

Sources enter the sending state at simulation startup according to a specified pattern

Initial Think Times for Sources

prON	probability a source starts in the sending state
prONsecond	probability a source starts sending after random time: mean 33% x think time
prONthird	probability a source starts sending after random time: mean 66% x think time
prREST	probability a source starts sending after random time: mean think time

prREST = (1 – *prON* – *prONsecond* – *prONthird*)



prON	0.25
prONsecond	0.08
prONthird	0.17
prREST	0.50

x20 specifies (*prON, prONsecond, prONthird)*

Two Sample Experiments

Joint work with Jim Filliben, Dong Yeon Cho and Edward Schwartz

Characteristic	**Sec. 8 High SST**	**Sec. 9 High SST**
Network Size (sources)	17,455 & 26,085	174,600 & 261,792
Backbone Speed (Gbps)	19.2 & 38.4	192 & 384
Packet Loss Rate	$1x10^{-4}$ to $1x10^{-2}$	$2x10^{-9}$ to $2x10^{-2}$
Initial Slow-Start Threshold	$2^{32}/2$	$2^{32}/2$
Alternate Congestion-Control Algorithms	BIC, CTCP, FAST, FAST-AT, HSTCP, HTCP, Scalable	BIC, CTCP, FAST, FAST-AT, HSTCP, HTCP, Scalable
Ratio (%) of Sources using Alternate Congestion-Control to Standard TCP Congestion-Control	30:70 & 70:30	30:70 & 70:30
Scenario	60 min. – 96-98% Web objects; 2-4% document transfers; smaller number of service-pack and movie downloads	60 min. – 96-98% Web objects; 2-4% document transfers; smaller number of service-pack and movie downloads

7 Congestion Control Algorithms x 32 Conditions

2^{9-4} Orthogonal Fractional Factorial Design

Note: factor designators x1 to x9 not the same as used in this talk

Factor->	x1	x2	x3	x4	x5	x6	x7	x8	x9
Condition	--	--	--	--	--	--	--	--	--
1	8000	1	0.5	5000	100	0.04/0.004/0.0004	0.7	0.7	3
2	16000	1	0.5	5000	100	0.04/0.004/0.0004	0.3	0.3	2
3	8000	2	0.5	5000	100	0.02/0.002/0.0002	0.7	0.3	2
4	16000	2	0.5	5000	100	0.02/0.002/0.0002	0.3	0.7	3
5	8000	1	1	5000	100	0.02/0.002/0.0002	0.3	0.7	2
6	16000	1	1	5000	100	0.02/0.002/0.0002	0.7	0.3	3
7	8000	2	1	5000	100	0.04/0.004/0.0004	0.3	0.3	3
8	16000	2	1	5000	100	0.04/0.004/0.0004	0.7	0.7	2
9	8000	1	0.5	7500	100	0.02/0.002/0.0002	0.3	0.3	3
10	16000	1	0.5	7500	100	0.02/0.002/0.0002	0.7	0.7	2
11	8000	2	0.5	7500	100	0.04/0.004/0.0004	0.3	0.7	2
12	16000	2	0.5	7500	100	0.04/0.004/0.0004	0.7	0.3	3
13	8000	1	1	7500	100	0.04/0.004/0.0004	0.7	0.3	2
14	16000	1	1	7500	100	0.04/0.004/0.0004	0.3	0.7	3
15	8000	2	1	7500	100	0.02/0.002/0.0002	0.7	0.7	3
16	16000	2	1	7500	100	0.02/0.002/0.0002	0.3	0.3	2
17	8000	1	0.5	5000	150	0.02/0.002/0.0002	0.3	0.3	2
18	16000	1	0.5	5000	150	0.02/0.002/0.0002	0.7	0.7	3
19	8000	2	0.5	5000	150	0.04/0.004/0.0004	0.3	0.7	3
20	16000	2	0.5	5000	150	0.04/0.004/0.0004	0.7	0.3	2
21	8000	1	1	5000	150	0.04/0.004/0.0004	0.7	0.3	3
22	16000	1	1	5000	150	0.04/0.004/0.0004	0.3	0.7	2
23	8000	2	1	5000	150	0.02/0.002/0.0002	0.7	0.7	2
24	16000	2	1	5000	150	0.02/0.002/0.0002	0.3	0.3	3
25	8000	1	0.5	7500	150	0.04/0.004/0.0004	0.7	1	2
26	16000	1	0.5	7500	150	0.04/0.004/0.0004	0.3	0.3	3
27	8000	2	0.5	7500	150	0.02/0.002/0.0002	0.7	0.3	3
28	16000	2	0.5	7500	150	0.02/0.002/0.0002	0.3	0.7	2
29	8000	1	1	7500	150	0.02/0.002/0.0002	0.3	0.7	3
30	16000	1	1	7500	150	0.02/0.002/0.0002	0.7	0.3	2
31	8000	2	1	7500	150	0.04/0.004/0.0004	0.3	0.3	2
32	16000	2	1	7500	150	0.04/0.004/0.0004	0.7	0.7	3

(7 x 32 =) 224 Runs for Small, Slow Network and (7 x 32 =) 224 Runs for Large, Fast Network

MesoNet Tractable?

Statistic	Small, Slow Network with High Initial Slow-Start Threshold: Flows Completed	Small, Slow Network with High Initial Slow-Start Threshold: Data Packets Sent	Large, Fast Network with High Initial Slow-Start Threshold: Flows Completed	Large, Fast Network with High Initial Slow-Start Threshold: Data Packets Sent
Avg. Per Condition	11,466,429	3,414,017,482	116,317,093	33,351,040,358
Min. Per Condition	7,258,056	2,138,998,764	72,944,797	21,069,357,409
Max. Per Condition	17,390,781	5,048,119,166	175,947,632	50,932,067,100
Total all Runs	2,568,480,122	764,739,915,978	26,055,028,851	7,470,633,040,199

35 weeks of CPU time competed in 1 week

131 months of CPU time competed in 3 months

48 processors available to run simulations

	Small, Slow Network with High Initial Slow-Start Threshold	Large, Fast Network with High Initial Slow-Start Threshold
CPU hours (224 Runs)	5,857.18	94,355.28
Avg. CPU hours (per run)	26.15	421.23
Min. CPU hours (one run)	12.58	203.04
Max. CPU hours (one run)	43.97	739.04
Avg. Memory Usage (Mbytes)	196.56	2,392.41

Future Work

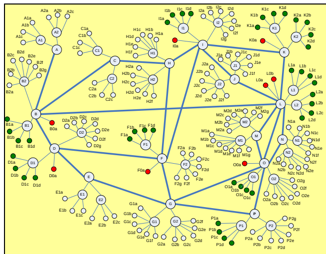
- Conduct 20-factor sensitivity analysis of *MesoNet* [*data analysis underway*]
- Use *MesoNet* to compare proposed Internet congestion control algorithms [*report drafted*]

JOINT WORK BETWEEN CxS and CNS Programs

- Develop a reduced scale simulation model for cloud computing IaaS (infrastructure-as-a-service) [*studying literature, code and deployments*]
- Conduct sensitivity analysis of IaaS model
- Compare proposed IaaS resource allocation algorithms [*studying literature and code*]

Joint work with Chris Dabrowski, Jim Filliben and Peter Mell

Backup Slides



Dijkstra’s SPF Backbone Routes Computed from Cost Metrics (2 of 2)

PATHS FOR NODE I

I : A @ I -> H -> C -> B -> A $ 170
I : B @ I -> H -> C -> B $ 120
I : C @ I -> H -> C $ 110
I : D @ I -> F -> D $ 163
I : E @ I -> F -> G -> E $ 165
I : F @ I -> F $ 55
I : G @ I -> F -> G $ 65
I : H @ I -> H $ 10
I : J @ I -> J $ 50
I : K @ I -> K $ 147
I : L @ I -> J -> L $ 110
I : M @ I -> J -> L -> M $ 160
I : N @ I -> J -> L -> N $ 149
I : O @ I -> J -> L -> N -> O $ 159
I : P @ I -> J -> L -> N -> O -> P $ 169

PATHS FOR NODE J

J : A @ J -> I -> H -> C -> B -> A $ 220
J : B @ J -> I -> H -> C -> B $ 170
J : C @ J -> I -> H -> C $ 160
J : D @ J -> I -> F -> D $ 213
J : E @ J -> I -> F -> G -> E $ 215
J : F @ J -> I -> F $ 105
J : G @ J -> I -> F -> G $ 115
J : H @ J -> I -> H $ 60
J : I @ J -> I $ 50
J : K @ J -> L -> K $ 110
J : L @ J -> L $ 60
J : M @ J -> L -> M $ 110
J : N @ J -> L -> N $ 99
J : O @ J -> L -> N -> O $ 109
J : P @ J -> L -> N -> O -> P $ 119

PATHS FOR NODE K

K:A @ K->I->H->C->B->A $ 317
K:B @ K->I->H->C->B $ 267
K:C @ K->I->H->C $ 257
K:D @ K->I->F->D $ 310
K:E @ K->L->N->O->G->E $ 303
K:F @ K->I -> F $ 202
K:G @ K->L->N->O->G $ 203
K:H @ K->I $157
K:I @ K -> I $ 147
K:J @ K->L $110
K:L @ K->L $50
K:M @ K->L->M $ 100
K:N @ K->L->N $ 89
K:O @ K->L->N->O $ 99
K:P @ K->L->N->O->P $ 109

PATHS FOR NODE L

L:A @ L->B->A $ 273
L:B @ L->B $223
L:C @ L->J -> H -> C $ 220
L:D @ L->N->O->G->E->D $ 263
L:E @ L->N->O->G->E $ 253
L:F @ L->N->O->G->F $ 163
L:G @ L->N->O->G $ 153
L:H @ L->J -> H $ 120
L:I @ L -> J -> I $ 110
L:J @ L -> J $ 60
L:K @ L->K $50
L:M @ L->M $ 50
L:N @ L->N $ 39
L:O @ L->N->O $ 49
L:P @ L->N->O->P $ 59

PATHS FOR NODE M

M:A @ M->L->B->A $ 323
M:B @ M->L->B $ 273
M:C @ M->L->J->I->H->C $ 270
M:D @ M->O->G->E->D $ 224
M:E @ M->O->G->E $ 214
M:F @ M->O->G->F $ 124
M:G @ M->O->G $ 114
M:H @ M->L->J->I->H $ 170
M:I @ M -> L -> J -> I $ 160
M:J @ M->L->J $ 110
M:K @ M->L->K $ 100
M:L @ M->L $ 50
M:N @ M->O->N $ 20
M:O @ M->O $ 10
M:P @ M->O->P $ 20

PATHS FOR NODE N

N:A @ N->L->B->A $ 312
N:B @ N->L->B $ 262
N:C @ N->L->J->I->H->C $ 259
N:D @ N->O->G->E->D $ 224
N:E @ N->O->G->E $ 214
N:F @ N->O->G->F $ 124
N:G @ N->O->G $ 114
N:H @ N->L->J->I->H $ 159
N:I @ N -> L -> J -> I $ 149
N:J @ N->I $99
N:K @ N->L->K $ 89
N:L @ N->L $ 39
N:M @ N->O->M $ 20
N:O @ N->O $ 10
N:P @ N->O->P $ 20

PATHS FOR NODE O

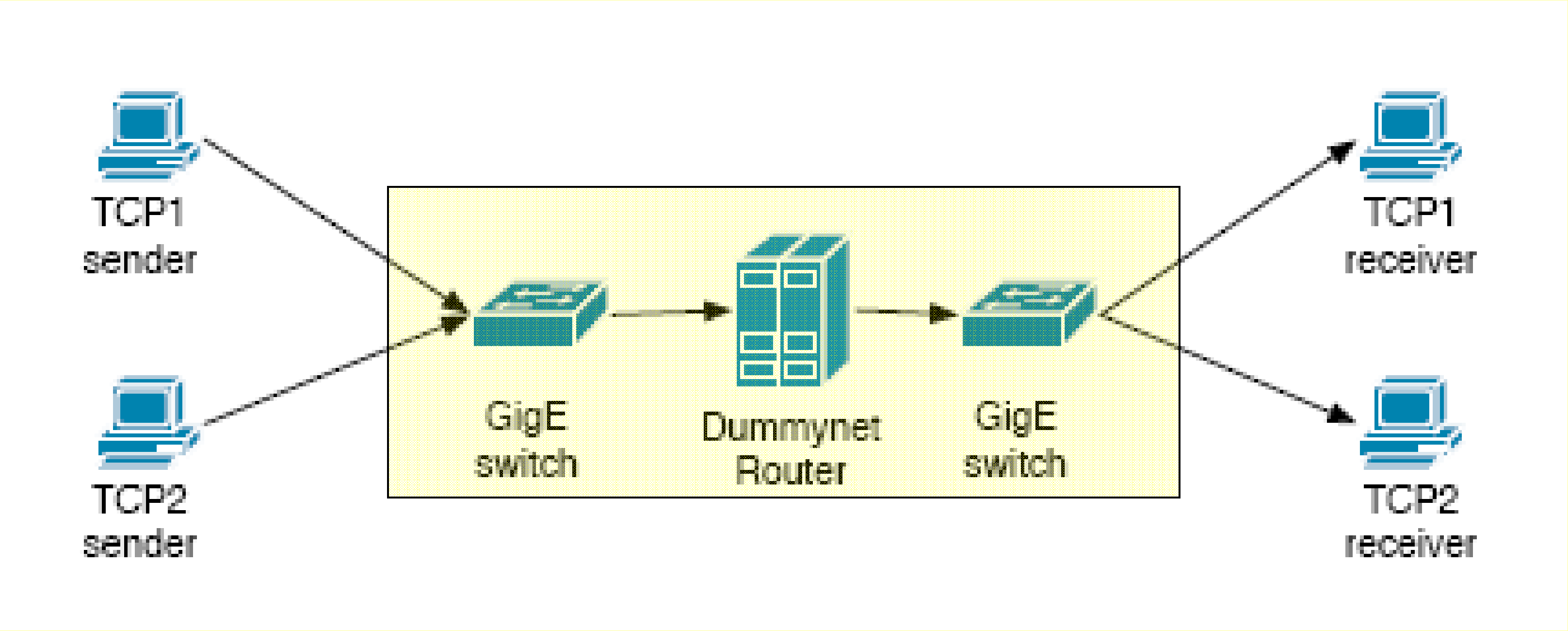
O:A @ O->G->E->D->B->A $ 314
O:B @ O->G->E->D->B $ 264
O:C @ O->G->F->H->C $ 264
O:D @ O->G->E->D $ 214
O:E @ O->G->E $ 204
O:F @ O->G->F $ 114
O:G @ O->G $ 104
O:H @ O->G->F->H $ 164
O:I @ O -> N -> L -> J -> I $ 159
O:J @ O->N->L->J $ 109
O:K @ O->N->L->K $ 99
O:L @ O->N->L $ 49
O:M @ O->M $ 10
O:N @ O->N $ 10
O:P @ O->P $ 10

PATHS FOR NODE P

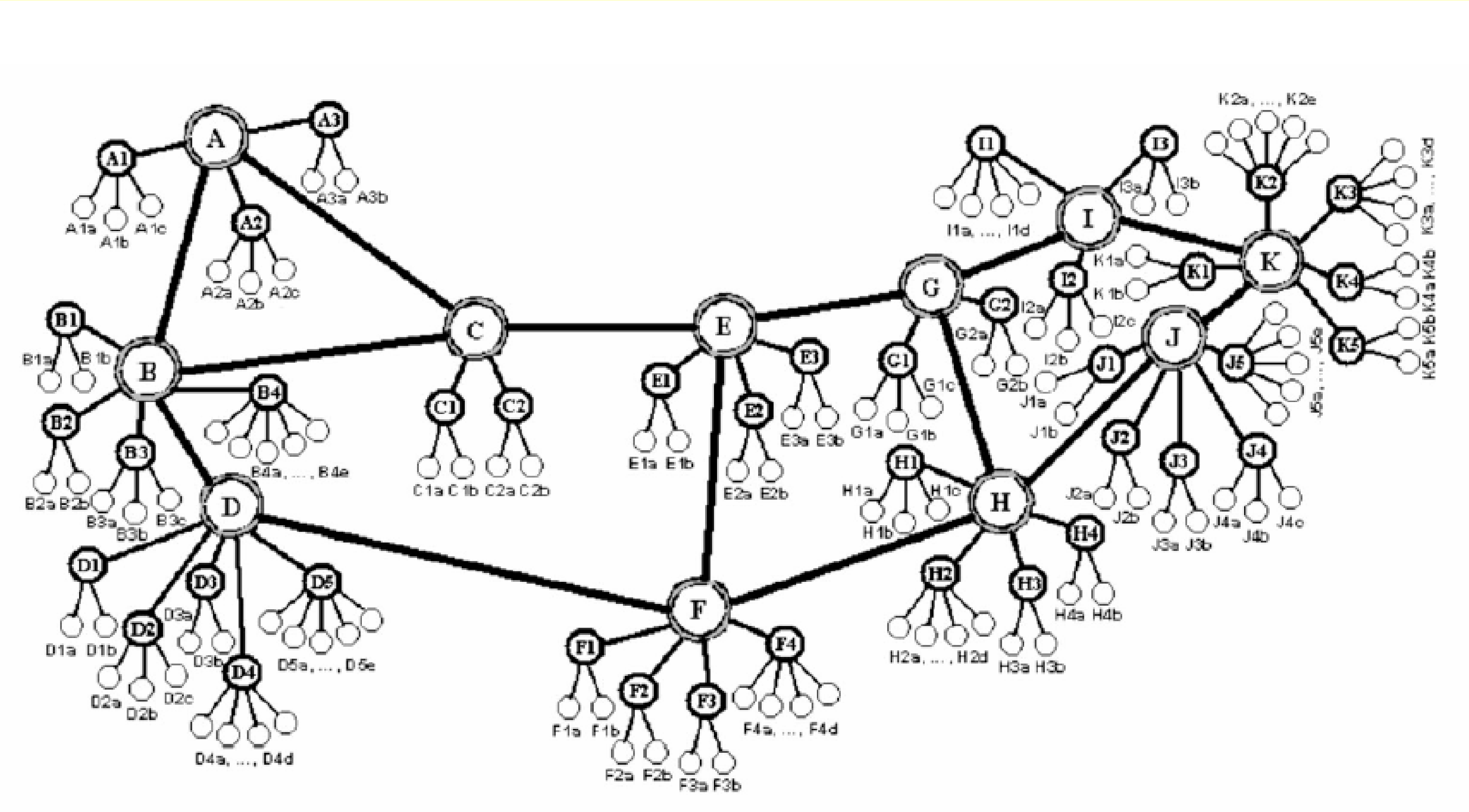
P:A @ P->G->E->D->B->A $ 320
P:B @ P->G->E->D->B $ 270
P:C @ P->G->F->H->C $ 270
P:D @ P->G->E->D $ 220
P:E @ P->G->E $ 210
P:F @ P->G->F $ 120
P:G @ P->G $ 110
P:H @ P->G->F->H $ 170
P:I @ P -> O -> N -> L -> J -> I $ 169
P:J @ P->O->N->L->J $ 119
P:K @ P->O->N->L->K $ 109
P:L @ P->O->N->L $ 59
P:M @ P->O->M $ 20
P:N @ P->O->N $ 20
P:O @ P->O $ 10

Other Sample *MesoNet* Topologies

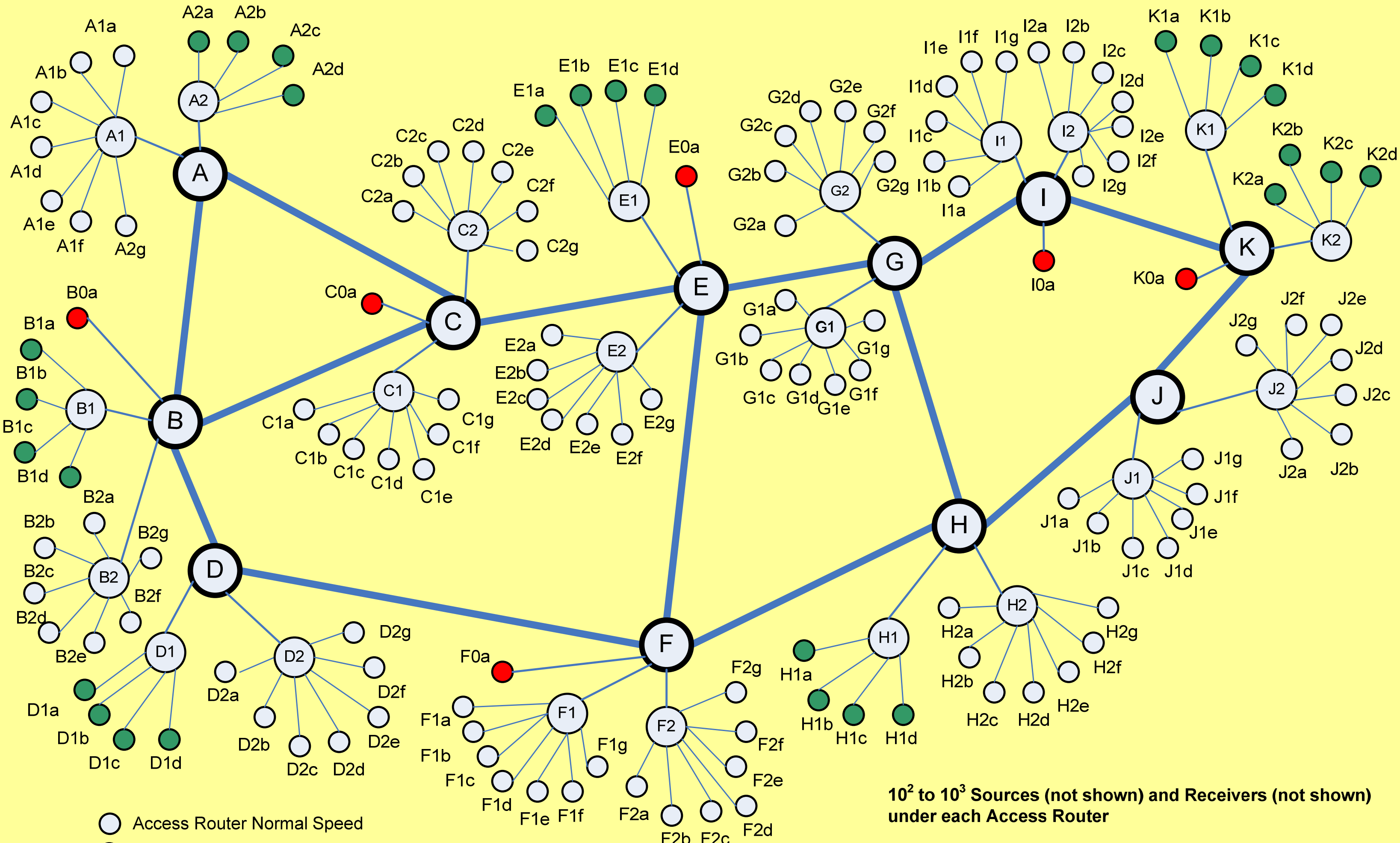
Sample Dumbbell Topology from an Empirical Study



Sample Simulated Topology (I) from *MesoNet*



Sample Simulated Topology (IIc & IId) from *MesoNet*

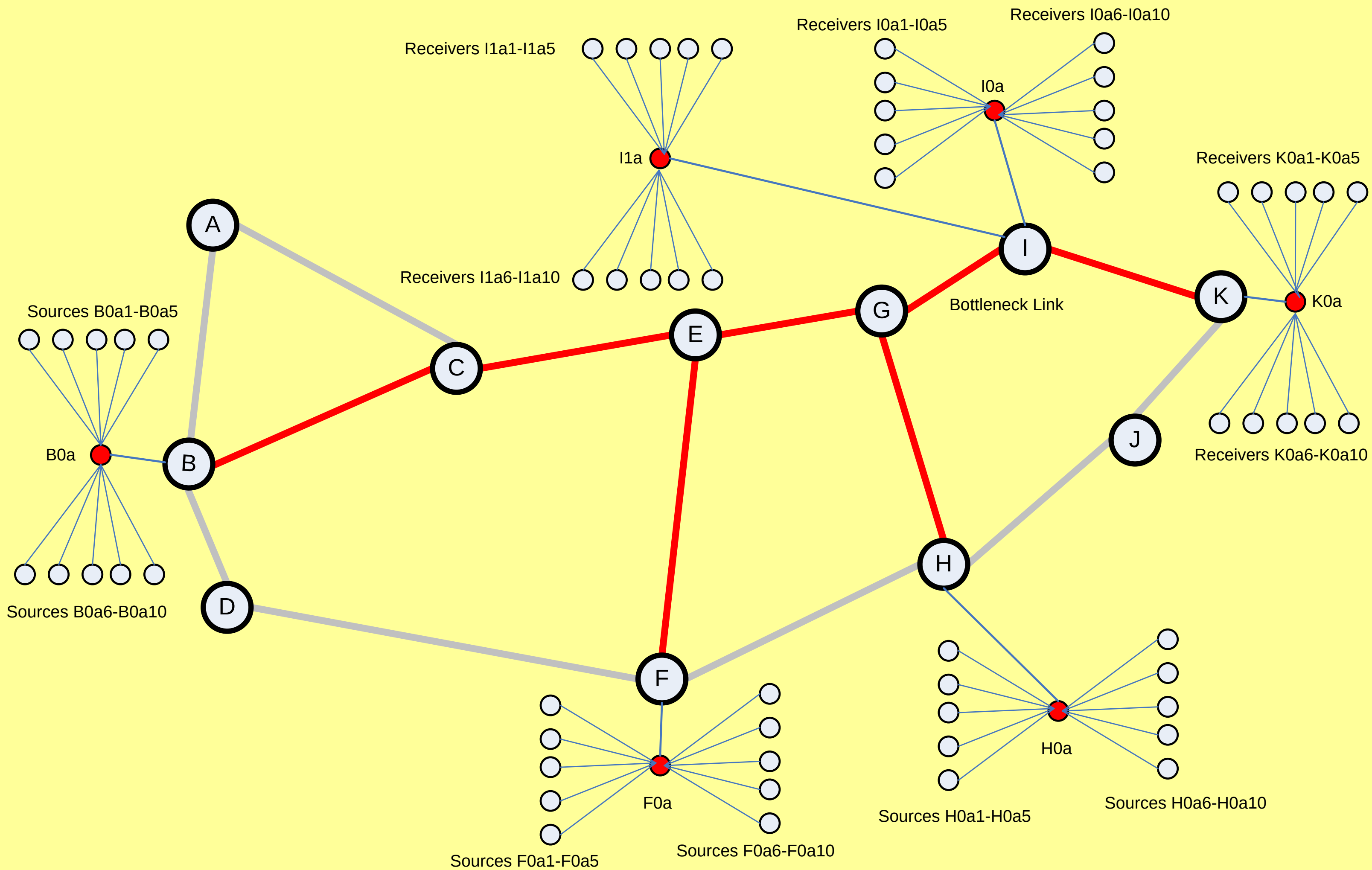


- Access Router Normal Speed
- Access Router n x Normal Speed
- Access Router m x Normal Speed

IIc – asymmetric routes

IId – SPF based on propagation delay

Sample Simulated Topology (ST) from *MesoNet*



SPF based on propagation delays