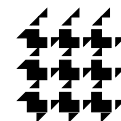


***ssfnet***

# Scalable Modeling & Simulations of Networks





# ssf and ssfnet

---

## **ssf**: a modern simulation platform

- scalable, parallel kernel, open API
- model configuration database framework
- random stream management & statistics
- pure **Java** (C++ version available)

**design & performance demonstrated at earlier PI meetings**

**network modeling layers: ssfnet**

**---- core ssf & ssfnet completed**



# ssfnet - overview 1

---

## modeling aids:

- network configuration **verification** and management
- total **reproducibility**: management of random number streams
- strongest random number generators, statistics package
- automated **IP address generation** (VLSM/CIDR)
- **traffic pattern** configuration

## practical good news:

- parallel performance from a **desktop PC** to a **supercomputer**
- fast **topology construction** and protocol configuration
- **no** programming style restrictions



# ssfnet - overview 2

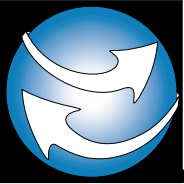
---

## walk-through example of multi-scale modeling:

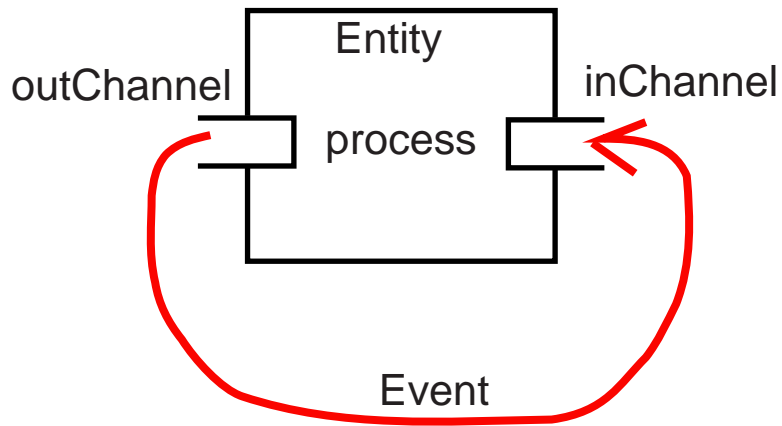
- USA-scale multi-AS network
- full Internet protocol suite
- empirical model of the Web traffic

## current research:

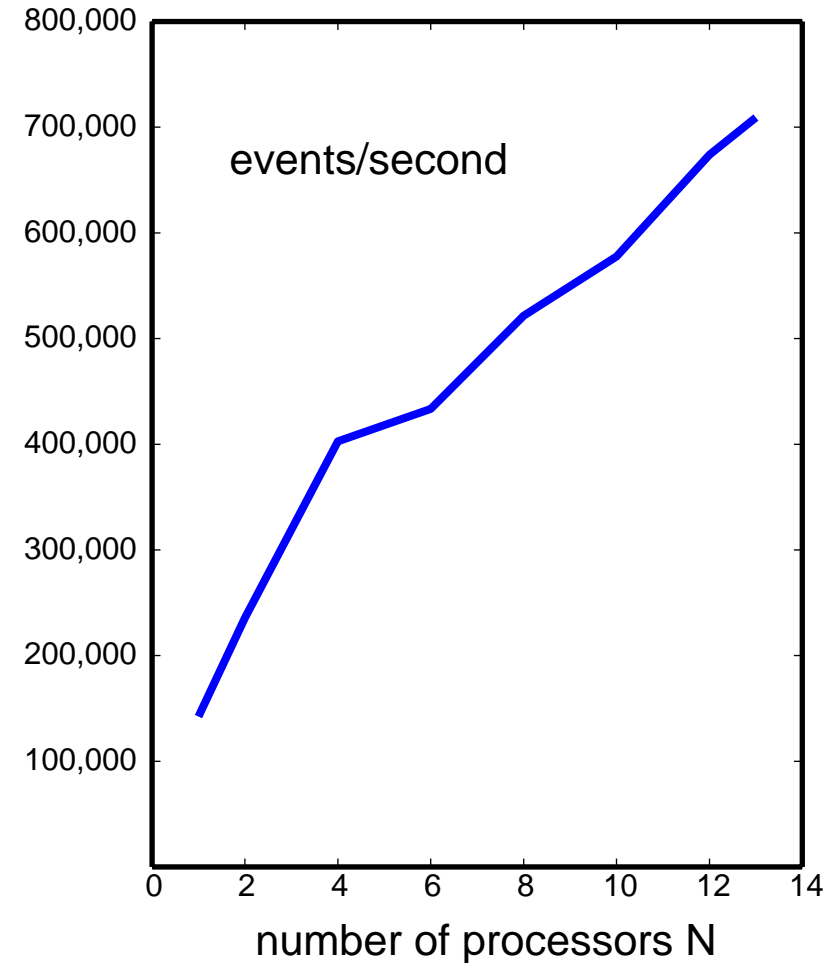
- **networkers' IDE** (integrated development environment)
- multiresolution analysis of **spatio-temporal network dynamics**
- wireless Internet problems



# Java SSF performance 1

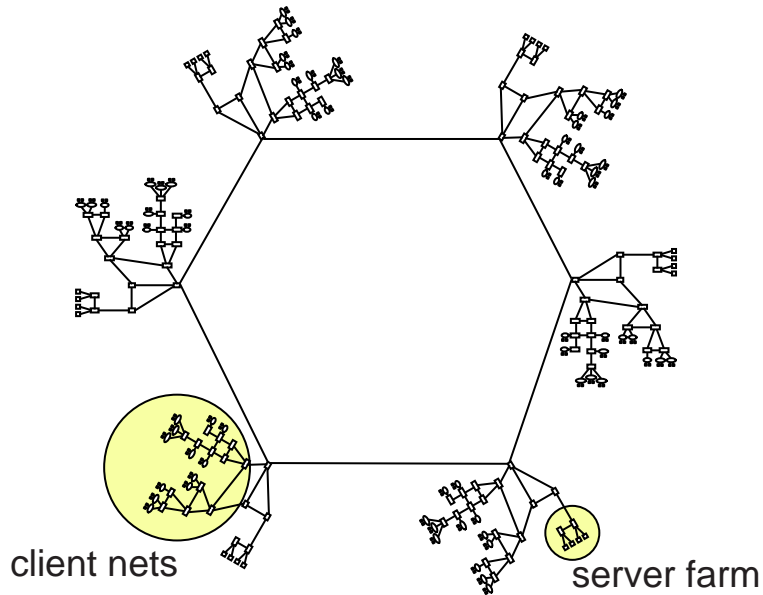


Basic event spin-loop experiment  
Java gets pretty fast by now!





# Java ssfnet - parallel performance



**N** networks (AS) connected as a ring

**OSPF** in each AS (24 clients, 4 servers, 18 routers)

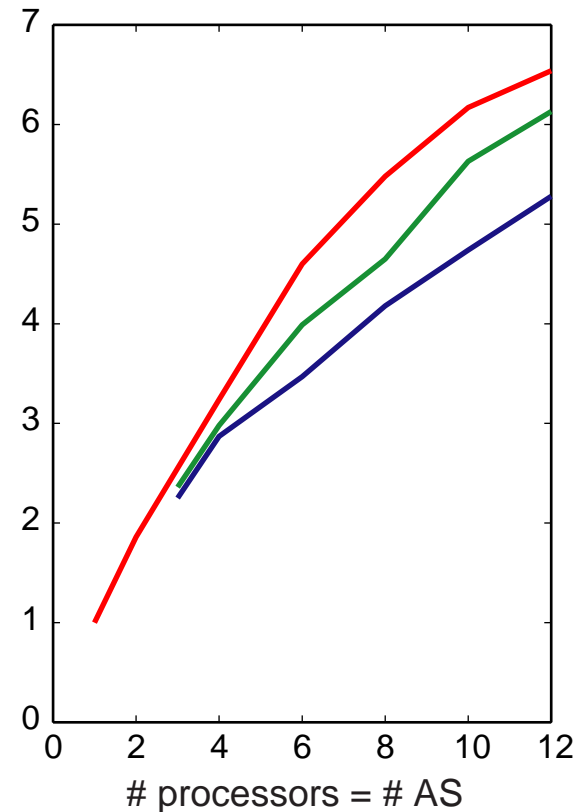
**BGP** between ASs

client-server **TCP** - large file transfer

- each clnt to **local** AS server —
- each clnt to **neighbor** AS server —
- each clnt to **random** AS server —

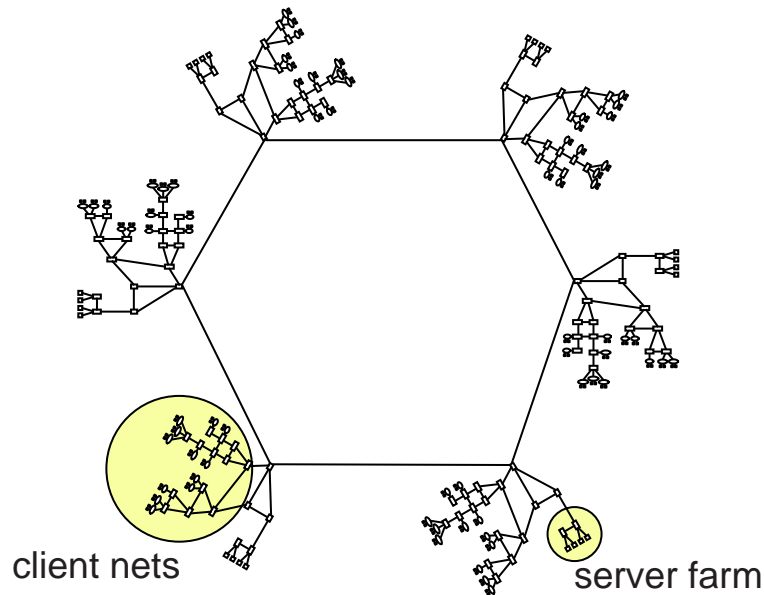
## parallel speedup

all AS on one processor exec time  
one AS per processor exec time





# Java ssfnet on a Windows PC



6 networks (AS) connected as a ring

OSPF in each AS (24 clients, 4 servers, 18 routers)

BGP between ASs

each client-server TCP - 3MB file transfer

3,000 TCP packets

each client connects to a **random** AS server

typical dual-processor PC  
1.7 times faster for parallel simulation

440 seconds



1 CPU

260 seconds



2 CPU



# network modeling packages

---

## **SSF.Net**

- host, router, queues, network interfaces
- links & LANs
- nested Net import, hierarchical component addressing
- IP address generation

## **SSF.OS**

- ProtocolGraph - ProtocolSession - ProtocolMessage  
x-kernel-like protocol design framework
- protocol and application models





# ssfnet protocol validation

---

## ssfnet distribution includes validation tests

- test network configurations, HTML docs, traces
- TCP, UDP, BGP4, OSPF in distribution
- validation against RFCs and other test suites



# Example: ssfnet TCP validation

---

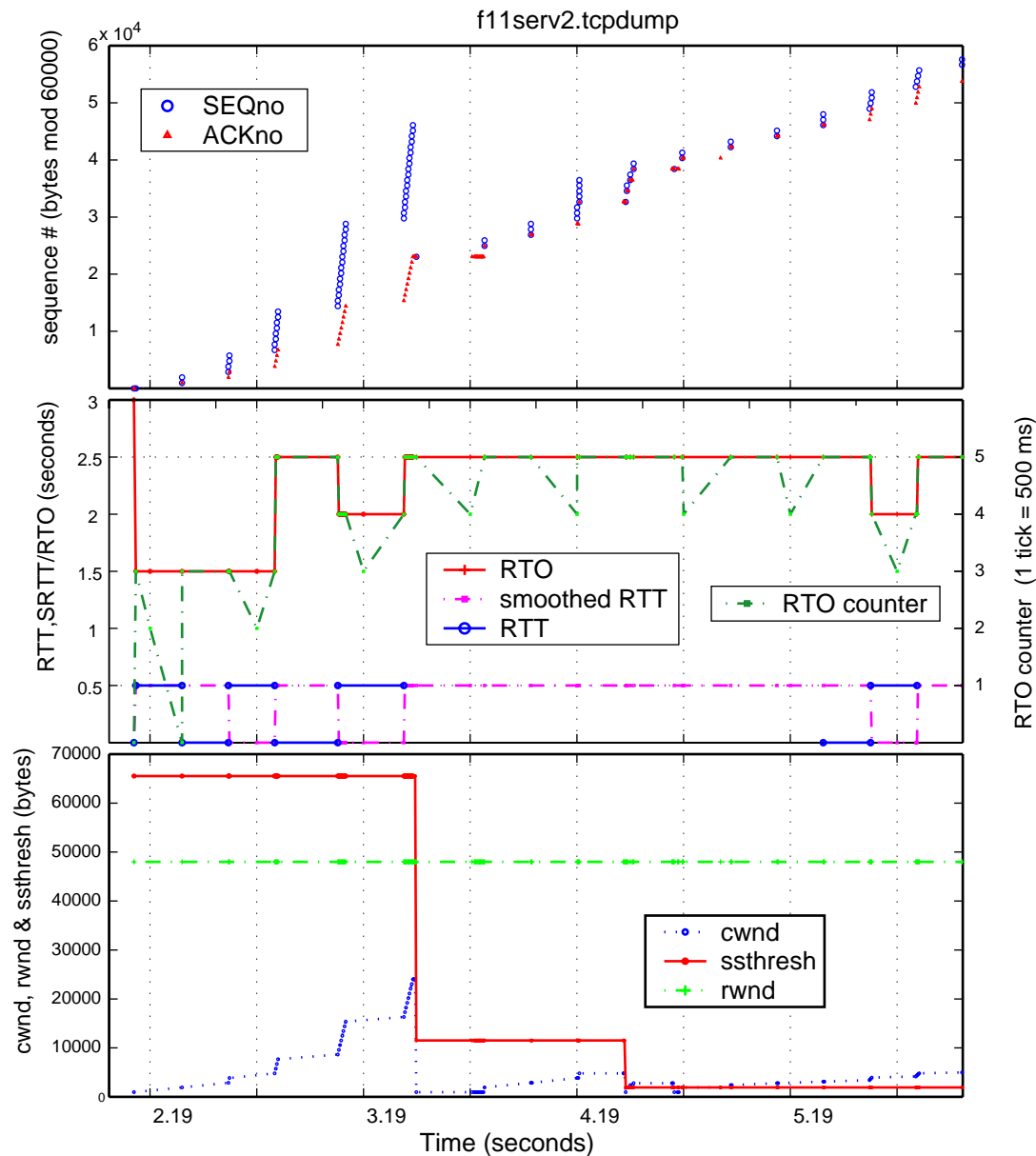
**generic Tahoe, Reno, delayed ACK option**

- BSD socket-like interface to app protocols
- validation against RFCs and ns-2 test suite (thanks!)
- full documentation and plots:

[www.winlab.rutgers.edu/~ato/ssf/tcp/](http://www.winlab.rutgers.edu/~ato/ssf/tcp/)



# TCP validation 2



binary tcpdump (standard)

fully instrumented  
TCP implementation  
(DML selectable)

matlab plotting  
in ssfnet distribution



# Modeling aids: SSF.DML

---

## Network configuration design/management database

- hierarchical object-oriented schemas: simple & powerful
- strong configuration reuse: substitution & multiple inheritance
- automatic verification of configuration correctness
- target for CAD and analysis tools
- separation of config database and executable code  
reduces modeling errors

**examples follow...**



# Modeling aids: SSF.Util.Random

---

## Total reproducibility of parallel simulations

- control of seed distribution via cryptographic techniques (MD) on random stream pathnames
- creation of random stream instances at multiple levels (per distribution, protocol, host, timeline)

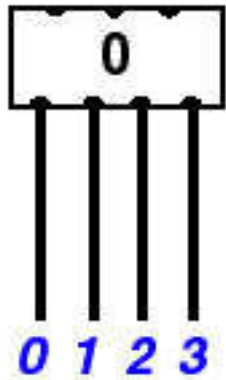
## Strongest, fast random number generators & statistics

- Mersenne Twister, Ranlux, Ranmar, Ranecu, RandomShuffle,..
- 24 common prob. distributions...statistics (CERN packages)

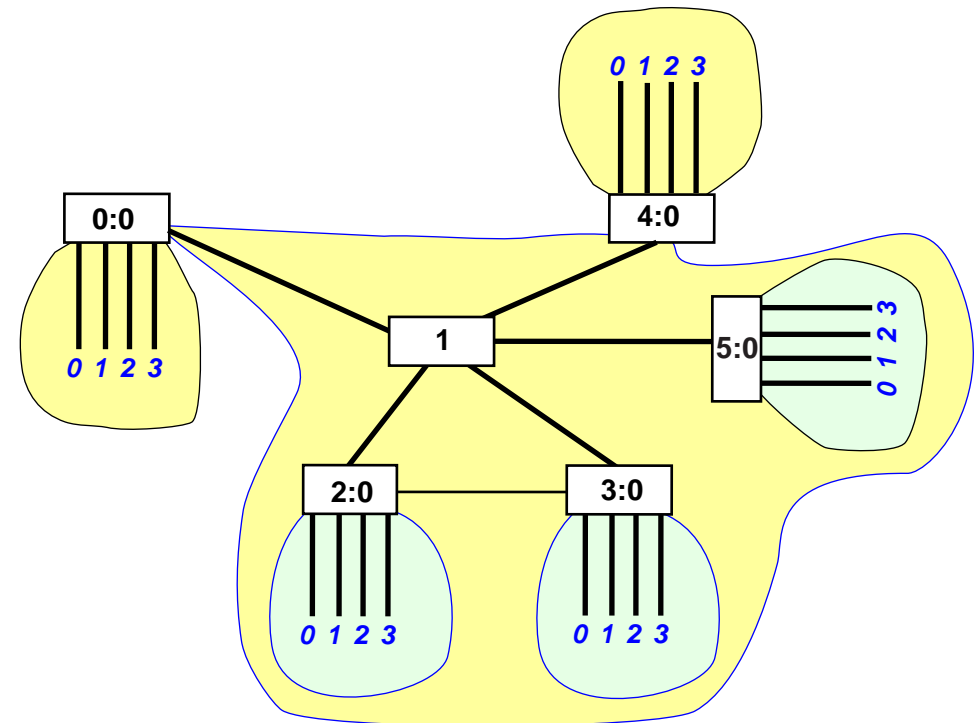
**examples follow...**



# walk-through a large model



router + 4 LANs,  
LAN = 25 hosts,  
= 100 hosts



add a router  
and connect together

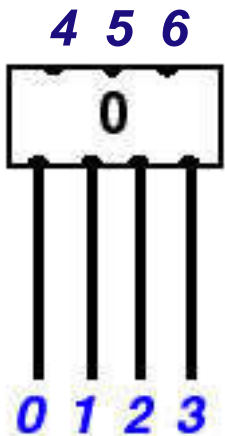


# walk through a large model 2

## DML database entry

from database of network cards.....  
inheritance

from database of protocol graphs.....  
substitution



```

networks [
  network100 [
    Net [
      router [ id 0
        interface [ id 0 _extends .dictionary.100Mb ] # LAN
        interface [ id 1 _extends .dictionary.100Mb ] # LAN
        .....
        interface [ id 5 _extends .dictionary.1Gb ] # to be attached
        interface [ id 6 _extends .dictionary.1Gb ] # to be attached
        _find .dictionary.routerGraph.graph
      ]
    ]
  ]
]

# all hosts are identical. 4 lans @ 25 hosts per lan = 100 hosts
host [
  idrange [from 1 to 25]
  _find .dictionary.client100Mb.interface
  _find .dictionary.client100Mb.graph
  nhi_route [dest default interface 0 next_hop 0(0)]
]

# LAN 0
link [ delay 0.001
  attach 0(0)
  attach 1(0) attach 2(0) attach 3(0) attach 4(0) attach 5(0)
  .....
  attach 21(0) attach 22(0) attach 23(0) attach 24 (0) attach 25(0)
]

```



# walk through a large model 3

**combine submodels together**

... make a small Autonomous System:

**1,300** client hosts, 4 big servers,

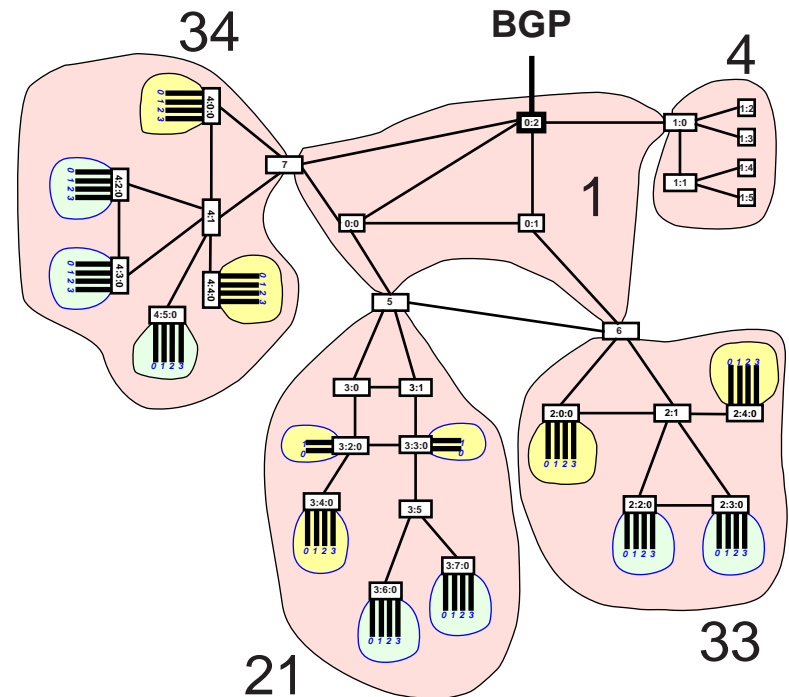
**26** internal routers (OSPF),

**1** AS boundary router (OSPF + BGP)

**DML database entry:**

```
as2 [  
  Net [  
    AS_status boundary  
    ospf_area 0
```

```
Net [id 0 __extends .networks.network1.Net] # backbone  
Net [id 1 __extends .networks.network4.Net] # server farm  
Net [id 2 __extends .networks.network33.Net]  
Net [id 3 __extends .networks.network21.Net]  
Net [id 4 __extends .networks.network34.Net]
```



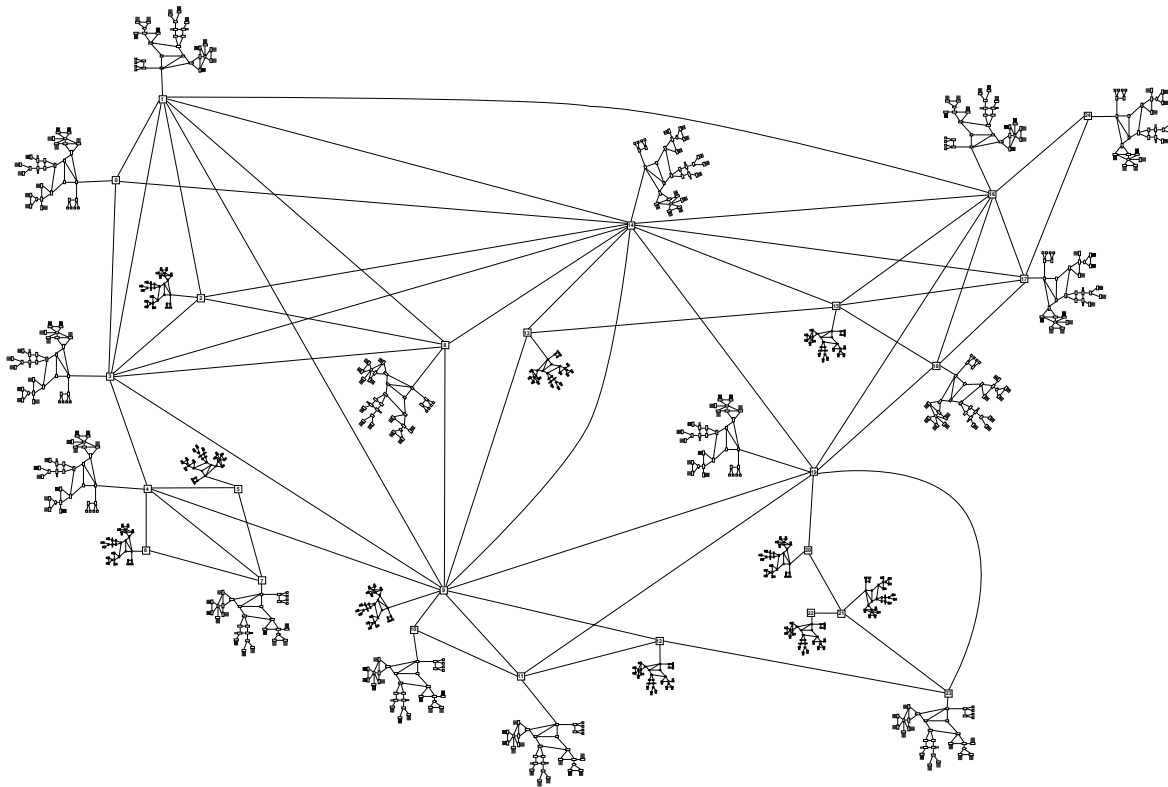
**.....Net import from a database  
- a powerful concept**





# walk-through a large model 4

now connect 25 copies of an **as2** network  
...and we have 33,325 hosts & routers  
... do you recognize the ISP ?





# walk through a large model 5

## Traffic pattern databases

### local

```
client100Mb [  
  graph [  
    ProtocolSession [  
      name client use SSF.OS.WWW.httpClient  
  
      # HTTP session empirical model.  
      inter_session_time [  
        distribution [  
          name "Exponential"  
          lambda 0.01  
        ]  
      ]  
      inter_page_time [  
        distribution [  
          name "Pareto"  
          k 25.0 alpha 2.0  
        ]  
      ]  
    ]  
  ]  
  ..... more
```

### global

```
randomstream [  
  generator "MersenneTwister"  
  stream "seedstarter4"  
  reproducibility_level "host"  
]  
  
traffic [  
  pattern [  
    client 10  
    servers [ port 80  
      nhi_range [  
        from 10:1:2(0) to 10:1:5(0)  
      ]  
    ]  
  ]  
  ..... more
```



# Current research: net dynamics

---

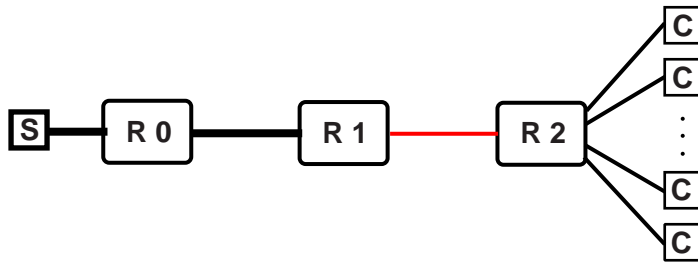
## AT&T Research - DIMACS - Princeton

- collaboration on multiresolution analysis of spatio-temporal behavior of large networks (Willinger, Feldmann, Daubechies, Ogielski and others)
- towards "new theory of networks"
- very challenging - drives ideas for new methodologies & tools
- simulated & real traffic studies

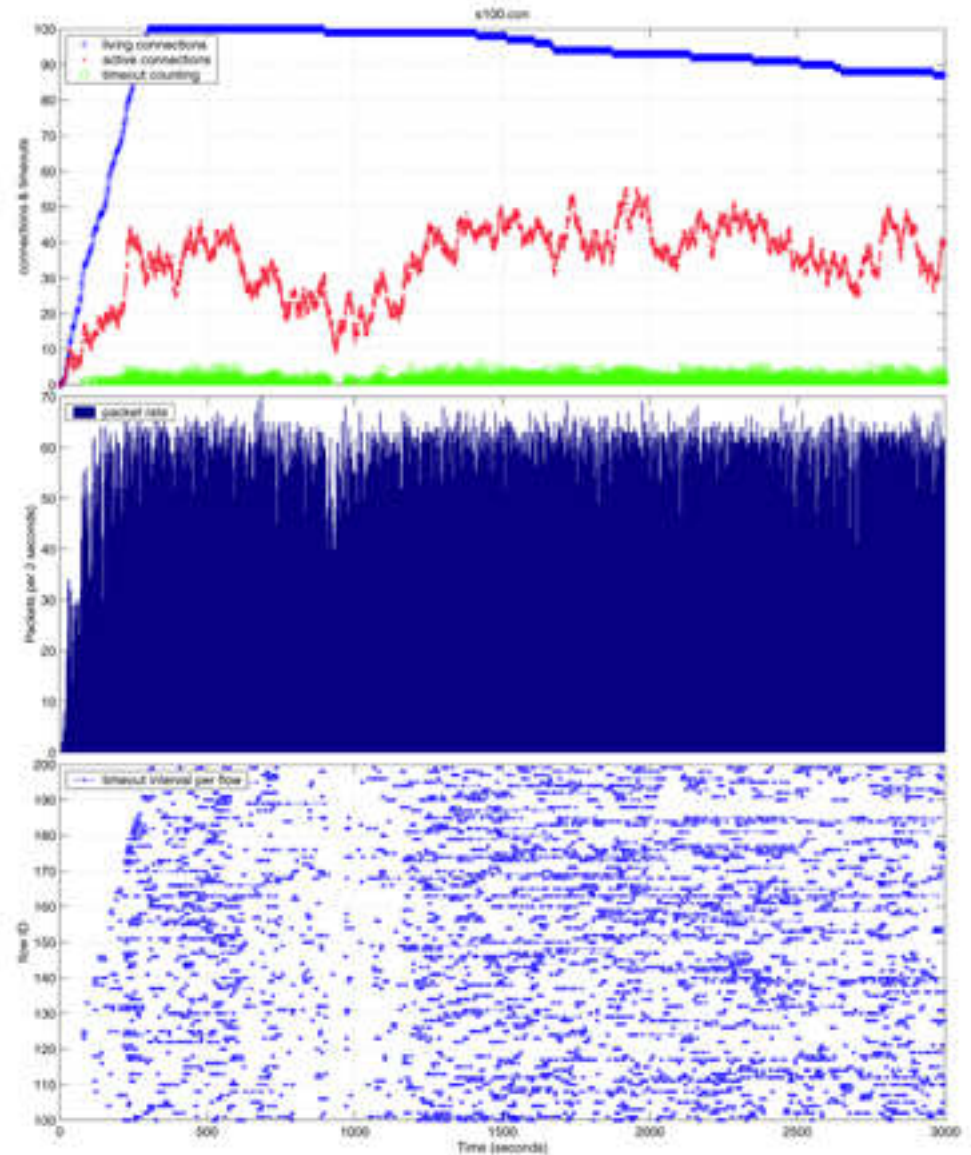
**examples follow...**



# Scalable Simulation Framework



**Challenge:**  
manipulate & visualize  
1000s of flows





# Current research: wireless Internet

---

## **WINLAB & affiliated corporations (Motorola, Lucent, Cisco,...)**

- SSF is the platform for various wireless Internet and 3G projects
- TCP over adaptive radio link protocols (like cdma2000)
- ad-hoc Bluetooth-based IP networks
- signal level multicell 3G systems: waveform to IP

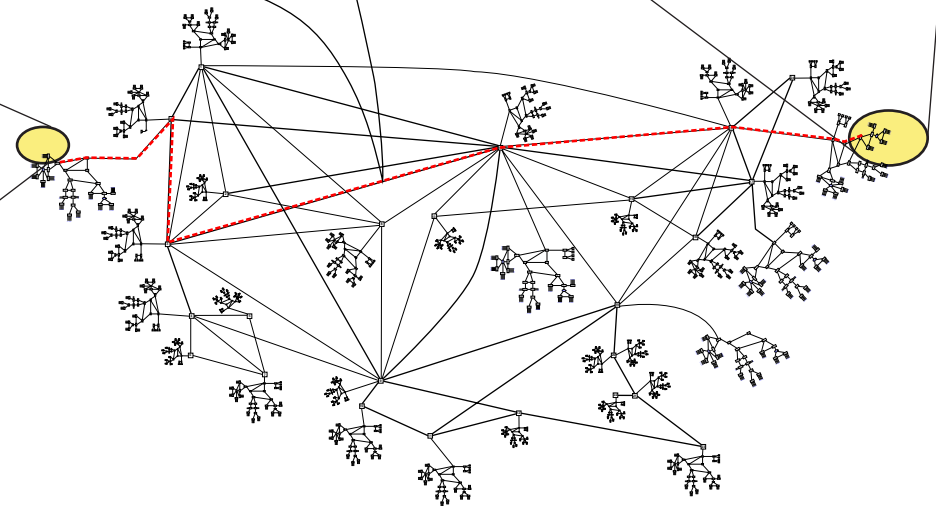
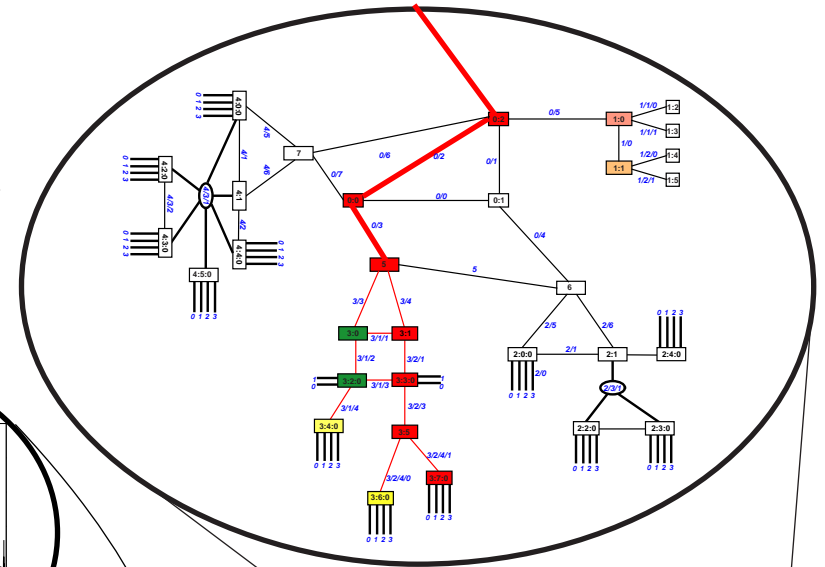
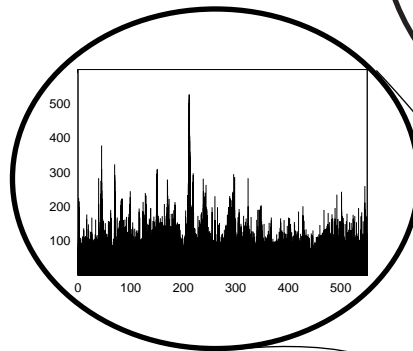
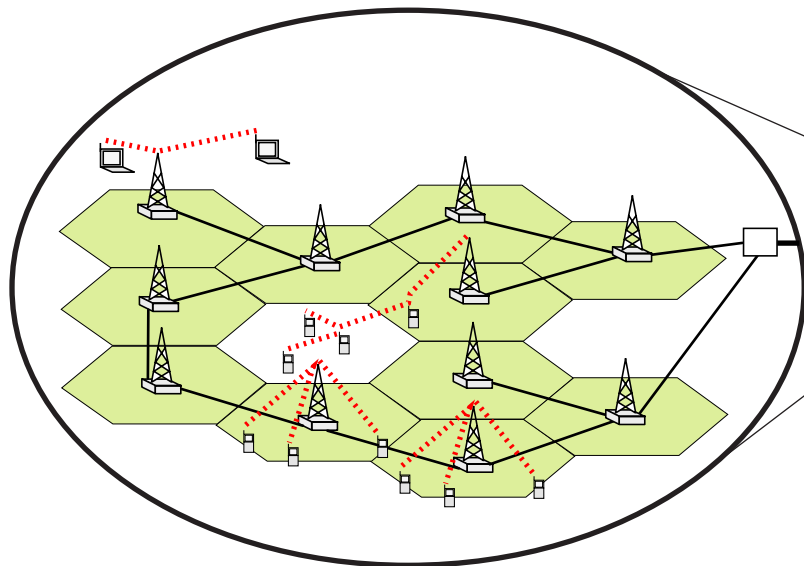
**examples follow...**

# ssfnet2 - integration of Internet & Wireless models

**Scale and Details matter!**

**wireless access + wired global net**  
**scalable design & analysis tools**

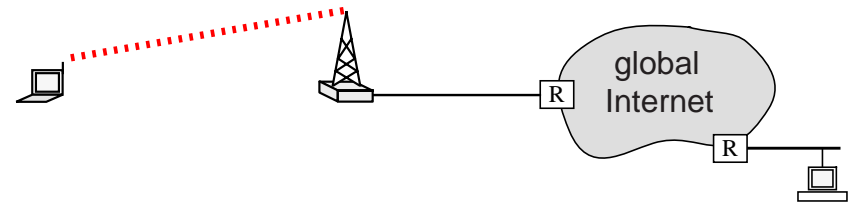
*Internet software radios,  
inter-protocol interactions,  
traffic, service design,...*



# Research on TCP - RLP interactions: towards better protocols for wireless Internet

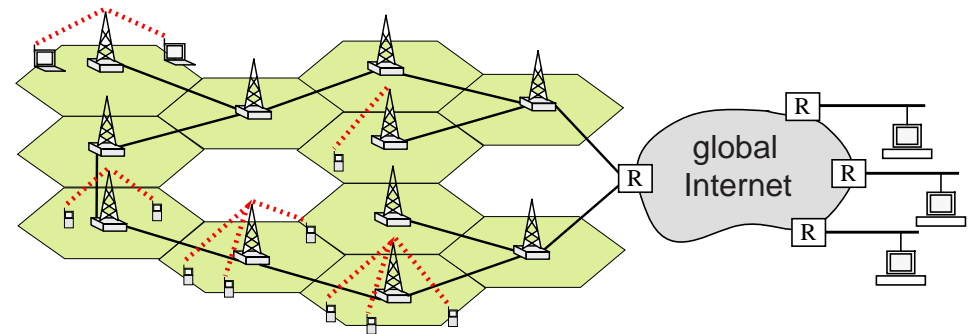
## phase 1: correlated fading losses

*1 wireless host, 1 TCP connection,  
RLP, fading, simplified IP cloud*



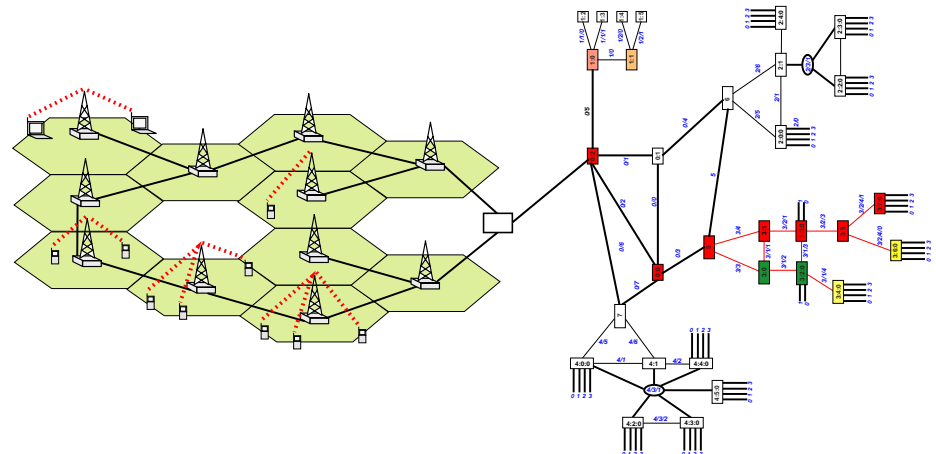
## phase 2: add multiaccess

*N wireless hosts, many connections,  
MAC + RLP, fading, interference,  
simplified IP cloud*



## phase 3: add wired congestion

*phase 2 plus realistic IP cloud:  
correlated wireline delays/losses*





# Current research: network IDE

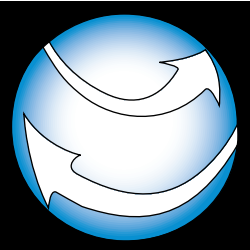
---

## Integrated Design Environment for networking research

- graphical **DML-based network editor**  
topology generation, protocol configuration, database generation,  
validation (prototype exists, needs some more work)
- High-volume stream data collection (~ 1000s of tcpdumps)  
and multiscale analysis package  
(challenging...)

**dual-use for model & real networks**





# ssfnet '99 development team

---

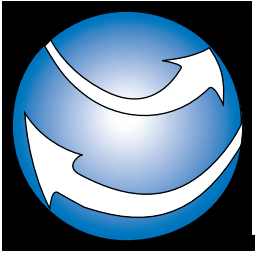
**Andy Ogielski** DIMACS & WINLAB  
**James Cowie** Cooperating System  
**David Nicol** Dartmouth College

## **Students (wired Internet)**

Myongsu Choe  
Philip Kwok  
Hongbo Liu  
Brian Premore

## **Students (wireless Internet)**

Yong Bai  
Vikram Kaul  
Wenfeng Zhang



# ssfnet web site

---

ssfnet software distribution

**[www.ssfnet.org](http://www.ssfnet.org)**

Open source modeling packages, protocols, networks.