The speedup of SSFNET simulation over NS

Li Li

May 31, 2001

Contents

1 Abstract

SSFNet is a collection of SSF-based models for simulating Internet protocols and networks. It is designed at the logical IP-network level, and the simulation quantum is an IP packet. In this report I used SSFNet to simulate two sample network configurations. One of the configurations is with 64 and the other is with 27 nodes. Also the comparison between the simulation results of SSFNet and NS is presented in this report.

2 Introduction

2.1 SSFNet

SSF stands for Scalable Simulation Framework. SSFNet is a collection of Java SSF-based components for modeling and simulation of Internet protocols and networks at and above the IP packet level of detail. SSFNet models are self-configuring - that is, each SSFNet class instance can autonomously configure itself by querying a configuration database, which may be locally resident or available over the Web. The network configuration files are in the DML format. They are used to configure a complete model, and instantiate a simulation with the help of the scalable DML configuration database package that is distributed with the SSF simulators.

The principal classes used to construct virtually any Internet model are organized into two derivative frameworks, SSF.OS (for modeling of the host the operating system components, esp. protocols) and SSF.Net (for modeling network connectivity, creating node and link configurations).

The frameworks SSF.OS and SSF.Net hide all details of the discrete event simulator SSF API, allowing to implement the protocols like for a real operating system.

The pricipal classes of package SSF.OS are:

- ProtocolGraph
- ProtocolSession
- ProtocolMessage and packetEvent

The principal classes of package SSF.Net are:

- Net
- Host and Router
- NIC
- Link

Class Net loads all the model's DML configuration files and controls the orderly instantiation of the entire model: hosts and routes with their protocols, links connecting hosts and routers, as well as traffic scennarios and multiple random number streams. DML configuration is hierarchical, and allows recursive composition of larger networks from pre-configured sub-networks.

2.2 Real-time On-line Network Simulation Approach

The major difficulty in simulating large networks at the packet level is the enormous computational power needed to execute all events that packets undergo the network [4]. To overcome this difficulty, a method in which a large network is decomposed into parts and each part is simulated independently and simultaneously with the others was designed and implemented with NS at RPI. The simulation results with NS showed that this approach significantly speed up the simulation of large networks. Our goal is to replace ns with SSFNet to provide several order magnitude simulation speed improvements over the sequential NS.

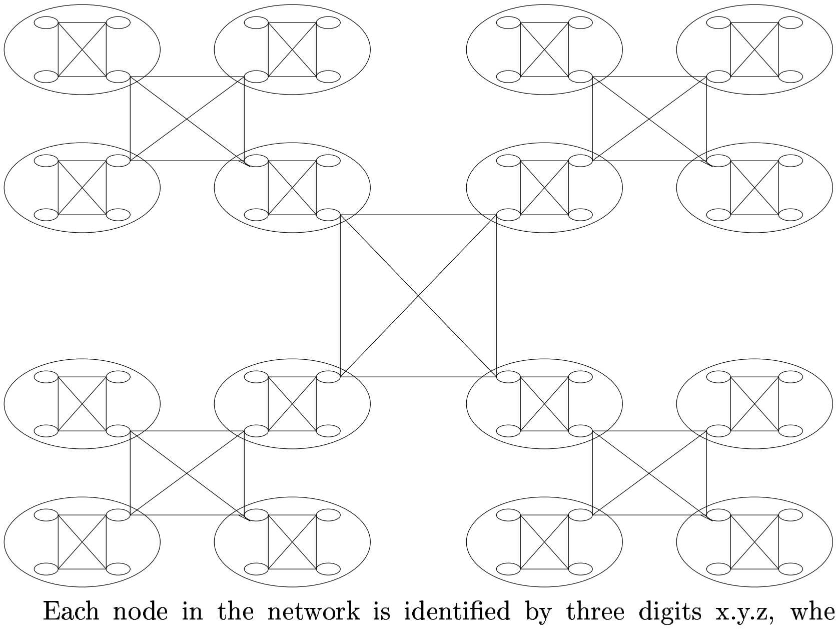
3 Network configurations

I use two sample network configurations, one with 64 and other with 27 nodes to test the speedup if SSFNet simulation over NS. Both of these networks are divided into classes of domains. The rate at which sources generate traffic are varied to generate temporal congestion in the network, especially at the nodes at the border of the domain. All sources produce packets of 500 bytes.

The 64-node network is designed with a great deal of symmetry. The smallest domain size is four nodes; there is full connectivity between these nodes. Four such domains together are considered as a larger domain in which there is full connectivity between the four sub-domains. Finally, four large domains are fully connected and form the entire network configuration.

The 27-node network is a PINNI [1] network with a hierarchical structure. Its smallest domain is composed of three nodes. Three such domains form a larger domain and three large domains form the entire network.

3.1 64-node network



Each node in the network is identified by three digits x.y.z, where $0 \leq x, y, z \leq 3$, that identify domain, sudomain and node rank within the subdomain to which the node belongs. Each node has nine flows originating from it. In addition, each node also acts as a sink to nine flows. The flows from node x.y.z go to nodes:

$$\begin{array}{lll} x.y.(z+1)\%4 & x.y.(z+2)\%4 & x.y.(z+3)\%4 \\ x.(y+1)\%4.z & x.(y+2)\%4.z & x.(y+3)\%4.z \\ (x+1)\%4.y.z & (x+2)\%4.y.z & (x+3)\%4.y.z \end{array}$$

Thus, this configuration forms a hierarchical and symmetrical structure on which the simulation is tested for scalability and speedup.

In a set of experiments, the sources at the borders of domains produce at the rate of 20000 packets/sec for half of the simulations time. The bandwidth of the link is 1.5Mbps. Thus, certain links are definitely congested and congestion may spread to some other links as well. For the other half of the simulation time, these sources produce 1000 packets per second. Since such flows require less bandwidth than provided by the links connected to each source. Congestion is not an issue. All other sources produce packets at the rate of 100 packets/sec for the entire simulation. For these experiments I defined sources that produced only CBR traffic and the speedup was measured by comparing simulation times to the simulation times with NS.

I conducted experiments with simulation time of 60 seconds, which is also used for the simulation with NS. The simulation speedup is approximately 38.

3.2 27-node network

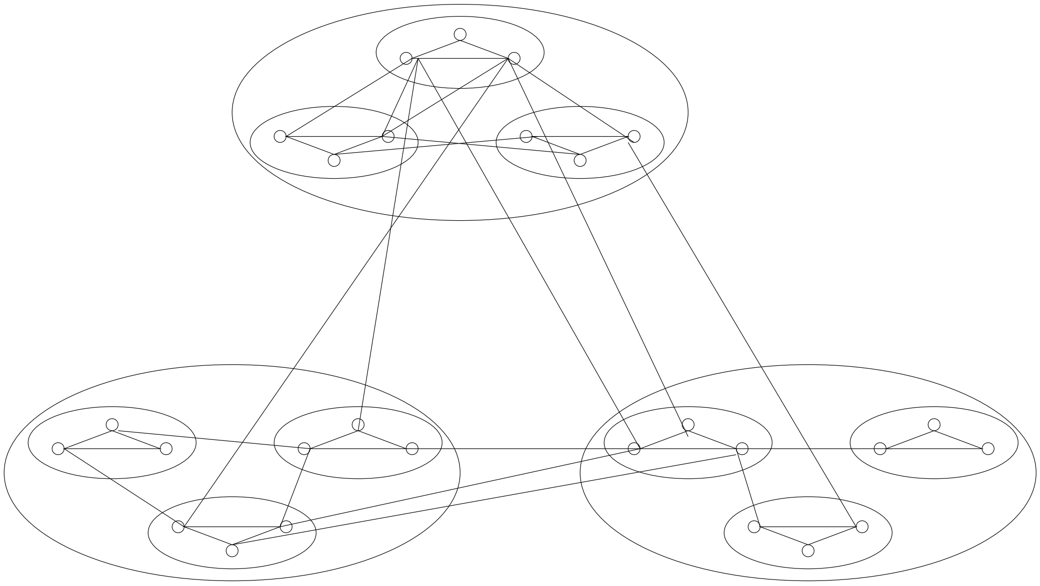


Figure 1: 64-node network configuration

The network configuration shown in Figure 2, the PINNI network adopted from [1] consists of 27 nodes arraged into 3 different levels of domains containing three, nine and 27 nodes, respectively. Each node has six flows to other nodes in the configuration and is receiving six flows from other nodes. The flows from a node x.y.z can be expressed as:

x.y.(z+1)%3   x.y.(z+2)%3
x.(y+1)%3.z   x.(y+2)%3.z
(x+1)%3.y.z   (x+2)%3.y.z

In these set of experiments, as above, the sources at the borders of domains produce packets at the rate of 20000 packets/sec for half of the simulation time. The bandwidth of the link is 1.5mbps. Thus, congestion is definitely produced on certain links shown above and congestion may be produced on certain other links. For the other half of the simulation, these sources produce 1000 packets which is less than the total bandwidth of the links connected to each of them. For these experiments I assume that all sources are producing CBR traffic. All other sources produce packets at the rate of 100 packets/sec for the entire simulation. I conducted experiments with simulation time of 60 seconds. The speedup of simulation with 27 nodes was well approximately 250 compared with simulation time of NS.

4 DML configurations

4.1 64-node DML configuration

```
_schema [_find .schemas.Net]
```

This line is to direct the DML parser to validate the network configuration, that is to verify that all attributes have required value types, that mandatory attributes are indeed present. This is very helpful for debugging.

```
Net [
frequency 1000000000
```

Net: Every network configuration is simply the value of Net attribute. frequency sets the physical unit value of the simulated time resolution as the integer number of dimensionless simulator "clock ticks" per simulated second: in this model I have a 1.5Mbps link meaning that nominally one bit is transmitted every 667 nanoseconds. Class SSF.Net.NIC (network interface card) computes the time needed to write a packet onto the link, thus needs time resolution at least of 667 nanoseconds, and our choice of 1ns is safe and will not introduce timing errors. the simulated time will advance by time intervals which are 1 nanoseconds.

```
randomstream [
generator "MersenneTwister"
stream "seedstartingstring1234567890"
reproducibility_level "timeline"
]
Net [id 0 _extends .networks.network16.Net]
Net [id 1 _extends .networks.network16.Net]
Net [id 2 _extends .networks.network16.Net]
Net [id 3 _extends .networks.network16.Net]
```

as I mentioned, the DML model is hierarchical. The 64-node network consists of 4 16-node subnet. And every subnet is exactly the same. The pre-configured 16-node network will be defined.

```
link [attach 0:3:4(7) attach 1:2:3(7) delay 0.01]
```

Class link models link-layer connectivity among a set of attached interfaces, The above link configuration will connect 4 16-node subnet as a 64-node network as figure 1. Every link configuration specifies one point to point connectivity. In this case just a point-to-point connection between two routers. Each attribute *attach* specifies a network interface (of a host or of a router) attached to the link. The value of *attach* (like attach 0:3:4(7) above) is the NHI address of the interface: here, 0:3:4(7) means ”interface with id=7 in host with id=4 in subnet 3 of net 0”. *delay* is the link transmission delay with value in seconds: here, 0.01s=1ms.

```
link [attach 0:3:4(8) attach 2:1:2(7) delay 0.01]
link [attach 0:3:4(9) attach 3:0:1(7) delay 0.01]
link [attach 1:2:3(8) attach 2:1:2(8) delay 0.01]
link [attach 1:2:3(9) attach 3:0:1(8) delay 0.01]
link [attach 2:1:2(9) attach 3:0:1(9) delay 0.01]
traffic [
pattern [
client 0:0:11
servers [nhi_range [from 0:0:22(0) to 0:0:24(0)] port 1600]
]
]
```

traffic is used to specify the global traffic scenarios for network models. Within the *traffic* attribute, there may be an arbitrary number of *pattern* sub-attributes. I don't have any use for it in this small network. Each *pattern* specifies two things: *client* specifies the NHI address of a client or a group of clients. *servers* specifies the NHI address of servers(including interface id) and the port number. Here I list three flows: the sink is at the node 0.0.1 and the three sources are 0.0.2, 0.0.3, and 0.0.4 respectively. The other flows are defined the same way.

```
]
networks [
network16 [
```

Here the 16-node network is defined.

```
Net [
Net [id 0 _extends .networks.network4.Net]
Net [id 1 _extends .networks.network4.Net]
Net [id 2 _extends .networks.network4.Net]
Net [id 3 _extends .networks.network4.Net]
```

The 16-node network itself is composed of 4 4-node small network as I defined as above.

```
link [attach 0:4(7) attach 1:3(7) delay 0.01]
link [attach 0:4(8) attach 2:2(7) delay 0.01]
link [attach 0:4(9) attach 3:1(7) delay 0.01]
link [attach 1:3(8) attach 2:2(8) delay 0.01]
link [attach 1:3(9) attach 3:1(8) delay 0.01]
link [attach 2:2(9) attach 3:1(9) delay 0.01]
```

The above link configuration forms a 16-node network as I defined.

```
traffic [
pattern [
client 0:11
servers [nhi 1:21(0) port 1600]
]
]
```

Again, I only list one flow here as an example. As the model is hierarchical so there are flows and one flow in every subnet. The sources are 0.1.1, 1.1.1, 2.1.1 3.3.3 and the sinks are 0.0.1, 1.0.1, 2.0.1, 3.0.1 respectively.

```
]
]
network4 [
Net [
```

Here I define the network configuration of 4-node net.

```
router [
idrange [from 1 to 4]
graph [ProtocolSession [name ip use SSF.OS.IP]]
interface [ idrange [from 4 to 9] buffer 8000 _extends .dictionary.15BaseT]
interface [ idrange [from 0 to 3] buffer 16000 _extends .dictionary.10BaseT]
route [dest default interface 4]
]
```

Every node in the configuration is actually a LAN. In SSFNet it consists of one router, two clients and two servers. A Router is a special case of a Host with multiple NICS. In our model, the four node are fully connected and some of them are needed to connect to nodes in other domains. And 4 interfaces are needed to connect to 2 clients and two servers for every router. Thus, I define 9 interfaces for it. For some nodes, there are several empty interfaces , which is allowed in SSFNet. Every host must have an id number which is specified by *id.* Above I specified four identical routers as elements of an array with *id* indices 1, 2, 3 and 4 using attribute *idrange.* The upper ProtocolSession entry instructs to load the class SSF.OS.IP, which installs a simplified model of the IP protocol in the router. I used the same construct to specify that each router has 6 15BaseT interfaces with id=4, 5, 6, 7, 8, 9 and 4 10BaseT interfaces with id=0, 1, 2, 3. Attribute *buffer* specifies the size of interface output queue buffer in bytes. Finally, using the attribute *route* I specified that if an IP packet's address is not in the IP routing table, the router should forward it via id=4.

```
link [attach 1(4) attach 2(4) delay 0.01]
link [attach 1(5) attach 3(4) delay 0.01]
link [attach 1(6) attach 4(4) delay 0.01]
link [attach 2(5) attach 3(5) delay 0.01]
link [attach 2(6) attach 4(5) delay 0.01]
link [attach 3(6) attach 4(6) delay 0.01]
```

The above links form a fully connected 4-node network.

```
link [attach 1(0) attach 11(0) delay 0]
link [attach 1(1) attach 21(0) delay 0]
link [attach 1(2) attach 31(0) delay 0]
link [attach 1(3) attach 41(0) delay 0]
link [attach 2(0) attach 12(0) delay 0]
link [attach 2(1) attach 22(0) delay 0]
link [attach 2(2) attach 32(0) delay 0]
link [attach 2(3) attach 42(0) delay 0]
link [attach 3(0) attach 13(0) delay 0]
link [attach 3(1) attach 23(0) delay 0]
link [attach 3(2) attach 33(0) delay 0]
link [attach 3(3) attach 43(0) delay 0]
link [attach 4(0) attach 14(0) delay 0]
link [attach 4(1) attach 24(0) delay 0]
link [attach 4(2) attach 34(0) delay 0]
link [attach 4(3) attach 44(0) delay 0]
```

The above links connect two clients and two server to every route.

```
host [idrange [from 11 to 14]
_extends .dictionary.standardClient
]
```

In the code fragment above, the line *_extends .dictionary.standardClient* is equivalent to the value of the attribute .dictionary.standardClient. And I specified four identical clients with id=11, 12, 13, 14.

```
host [idrange [from 21 to 24]
_extends .dictionary.standardServer
]
```

The above host configuration defines four servers.

```
host [idrange [from 31 to 34]
_extends .dictionary.standardClient1
]
```

In order to generate different traffic load at different time, I define another four clients.

```
host [idrange [from 41 to 44]
_extends .dictionary.standardServer1
]
```

The above host configuration defines another four servers.

```
]
]
]
dictionary [
10BaseT [
bitrate 10000000
latency 0 ]
```

The above code fragment defines the interface of 10BaseT. *bitrate* specifies the bandwidth in bps. And this 10BaseT interself doesn't introduce any delay.

```
15BaseT [
bitrate 1500000
latency 0.01 ]
```

The bandwidth of every link between two nodes is 1.5Mbps. It is defined as above. *latency* specifies the delay in second.

```
udpinit [
max_datagram_size 10000
debug false
]
appsession [
request_size 256
datagram_size 500
```

The packet size is 500 bytes.

```
send_interval 0.01
```

The traffic is 100 packets/sec.

```
show_report true
debug true
]
appsession1 [
request_size 256
datagram_size 500
send_interval 0.00005
show_report true
debug true
]
standardClient [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
```

The above code specified that the clients are UDP client.

```
start_time 0.1 # earliest time to send request to server
start_window 0.0 # send request to server at 0.1s
file_size 3000000 # requested payload bytes.
_find .dictionary.appsession.request_size
```

The above code fragment means "replace this line with *request_size* followed by the value of the attribute .dictionary.appsession.request_size".

```
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
```

In the above code, *name* is a symbolic tag, by which a protocol implementation finds its configuration parameter.*use* names the SSFNet class that should be loaded to do the protocol's job.

```
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardClient1 [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
start_time 1.0
start_window 1.0
file_size 298500000
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardServer [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name server use SSF.OS.UDP.test.udpStreamServer
port 1600 # specifies the port number
client_limit 10 # specifies the maximum clients connected to a server
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.send_interval
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
```

```
    ]
    ]
    standardServer1 [
    interface [id 0 _extends .dictionary.10BaseT]
    route [dest default interface 0]
    graph [
    ProtocolSession [
    name server use SSF.OS.UDP.test.udpStreamServer
    port 1600
    client_limit 10
    _find .dictionary.appsession1.request_size
    _find .dictionary.appsession1.datagram_size
    _find .dictionary.appsession1.send_interval
    _find .dictionary.appsession1.show_report
    _find .dictionary.appsession1.debug
    ]
    ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
    ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
    _find .dictionary.udpinit]
    ProtocolSession [name ip use SSF.OS.IP]
    ]
    ]
    ]
```

4.2 27-node DML configuration

```
_schema [_find .schemas.Net]
    Net [
    frequency 1000000000
    randomstream [
    generator ”MersenneTwister”
    stream ”seedstartingstring1234567890”
    reproducibility_level ”timeline”
    ]
    Net [id 1 _extends .networks.network91.Net]
    Net [id 2 _extends .networks.network92.Net]
    Net [id 3 _extends .networks.network93.Net]
```

The 27-node network is not as symmtric as 64-node. It consists of 3 smaller 9-node network. but every network is different from another one. Thus I have to define every network independently.

```
    link [attach 1:1:2(8) attach 2:2:1(6) delay 0.01]
    link [attach 1:1:3(8) attach 2:3:1(7) delay 0.01]
    link [attach 1:2:3(7) attach 3:3:2(7) delay 0.01]
    link [attach 1:1:3(9) attach 3:1:1(6) delay 0.01]
    link [attach 1:3:2(7) attach 3:2:1(7) delay 0.01]
    link [attach 2:2:3(6) attach 3:1:2(6) delay 0.01]
```

```
link [attach 2:3:2(7) attach 3:1:2(7) delay 0.01]
link [attach 2:1:3(6) attach 3:2:1(8) delay 0.01]
link [attach 2:3:3(6) attach 3:1:3(8) delay 0.01]
```

Again, every link configuration specifies one point to point connectivity.

```
traffic [
pattern [
client 1:1:11
servers [nhi_range [from 1:1:22(0) to 1:1:23(0)] port 1600]
]
]
```

The traffic configuration is the same as that of 64-node. And I only list two flows here as an example.

```
]
networks [
network91 [
Net [
Net [id 1 _extends .networks.network3.Net]
Net [id 2 _extends .networks.network3.Net]
Net [id 3 _extends .networks.network3.Net]
link [attach 1:2(6) attach 2:1(6) delay 0.01]
link [attach 1:2(7) attach 2:2(6) delay 0.01]
link [attach 1:3(6) attach 2:2(7) delay 0.01]
link [attach 1:3(7) attach 3:2(6) delay 0.01]
link [attach 2:2(8) attach 3:3(6) delay 0.01]
link [attach 2:3(6) attach 3:1(6) delay 0.01]
]
]
```

The above net configuration defines the first 9-node network.

```
network92 [
Net [
Net [id 1 _extends .networks.network3.Net]
Net [id 2 _extends .networks.network3.Net]
Net [id 3 _extends .networks.network3.Net]
link [attach 1:1(6) attach 2:2(6) delay 0.01]
link [attach 1:2(6) attach 3:1(6) delay 0.01]
link [attach 2:2(7) attach 3:2(6) delay 0.01]
]
]
```

The above is the configuration of the second 9-node network.

```
network93 [
Net [
Net [id 1 _extends .networks.network3.Net]
Net [id 2 _extends .networks.network3.Net]
Net [id 3 _extends .networks.network3.Net]
link [attach 1:3(6) attach 2:1(6) delay 0.01]
link [attach 1:3(7) attach 3:2(6) delay 0.01]
]
]
```

The above is the configuration of the third 9-node network. For every 9-node network, it consists three 3-node fully connected network which will be defined.

```
network3 [
Net [
router [
idrange [from 1 to 3]
graph [ProtocolSession [name ip use SSF.OS.IP]]
interface [idrange [from 4 to 9] buffer 8000 _extends .dictionary.15BaseT]
interface [idrange [from 0 to 3] buffer 16000 _extends .dictionary.10Baset]
route [dest default interface 4]
]
link [attach 1(4) attach 2(4) delay 0.01]
link [attach 1(5) attach 3(4) delay 0.01]
link [attach 2(5) attach 3(5) delat 0.01]
link [attach 1(0) attach 11(0) delay 0]
link [attach 1(1) attach 21(0) delay 0]
link [attach 1(2) attach 31(0) delay 0]
link [attach 1(3) attach 41(0) delay 0]
link [attach 2(0) attach 12(0) delay 0]
link [attach 2(1) attach 22(0) delay 0]
link [attach 2(2) attach 32(0) delay 0]
link [attach 2(3) attach 42(0) delay 0]
link [attach 3(0) attach 13(0) delay 0]
link [attach 3(1) attach 23(0) delay 0]
link [attach 3(2) attach 33(0) delay 0]
link [attach 3(3) attach 43(0) delay 0]
host [idrange [from 11 to 13]
_extends .dictionary.standardClient
]
host [idrange [from 21 to 23]
_extends .dictionary.standardServer
]
host [idrange [from 31 to 33]
_extends .dictionary.standardClient1
]
host [idrange [from 41 to 43]
_extends .dictionary.standardServer1
]
]
]
]
dictionary [
10BaseT [
bitrate 10000000
```

```
latency 0
]
15BaseT [
bitrate 1500000
latency 0.01 ]
udpinit [
max_datagram_size 10000
debug false
]
appsession [
request_size 256
datagram_size 500
send_interval 0.01
show_report true
debug true
]
appsession1 [
request_size 256
datagram_size 500
send_interval 0.00005
show_report true
debug true
]
standardClient [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
start_time 1.0
start_window 1.0
file_size 3000000
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardClient1 [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
```

```
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
start_time 1.0
start_window 1.0
file_size 298500000
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster _find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardServer [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name server use SSF.OS.UDP.test.udpStreamServer
port 1600
client_limit 10
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.send_interval
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardServer1 [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name server use SSF.OS.UDP.test.udpStreamServer
port 1600
client_limit 10
_find .dictionary.appsession1.request_size
_find .dictionary.appsession1.datagram_size
_find .dictionary.appsession1.send_interval
_find .dictionary.appsession1.show_report
_find .dictionary.appsession1.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
]
```

4.3 How to run SSFNet models

On a UNIX system, the command to run the model is:

```
java -classpath :/ssfnet1.2.2/src:/ssfnet1.2.2/lib/ssfnet.jar
:/ssfnet1.2.2/lib/ssf.jar:/ssfnet1.2.2/lib/dml.jar
:/ssfnet1.2.2/lib/cernlite.jar:/ssfnet1.2.2/lib/regexp.jar
SSF.Net.Net 60 networks.dml net.dml
```

Above, "60" is the value of the requested simulation end time in seconds; and net.dml is the official DML SSFNet schema file that is used for automatic validation of the input file, networks.dml.

At the start of the simulation run, SSF.Net.Net object will use the services of the DML library to load the content of the files networks.dml and net.dml into a runtime Configuration database object. After that, SSF.Net.Net will systematically instantiate and configure all simulation objects (Hosts, Routers, protocols, and network links). Once all simulation objects have been instantiated, the initialization phase begins, and finally SSF.Net.Net invokes its method startAll(), and the simulation begins with simulation time value equal to 0 sec.

5 Results analysis

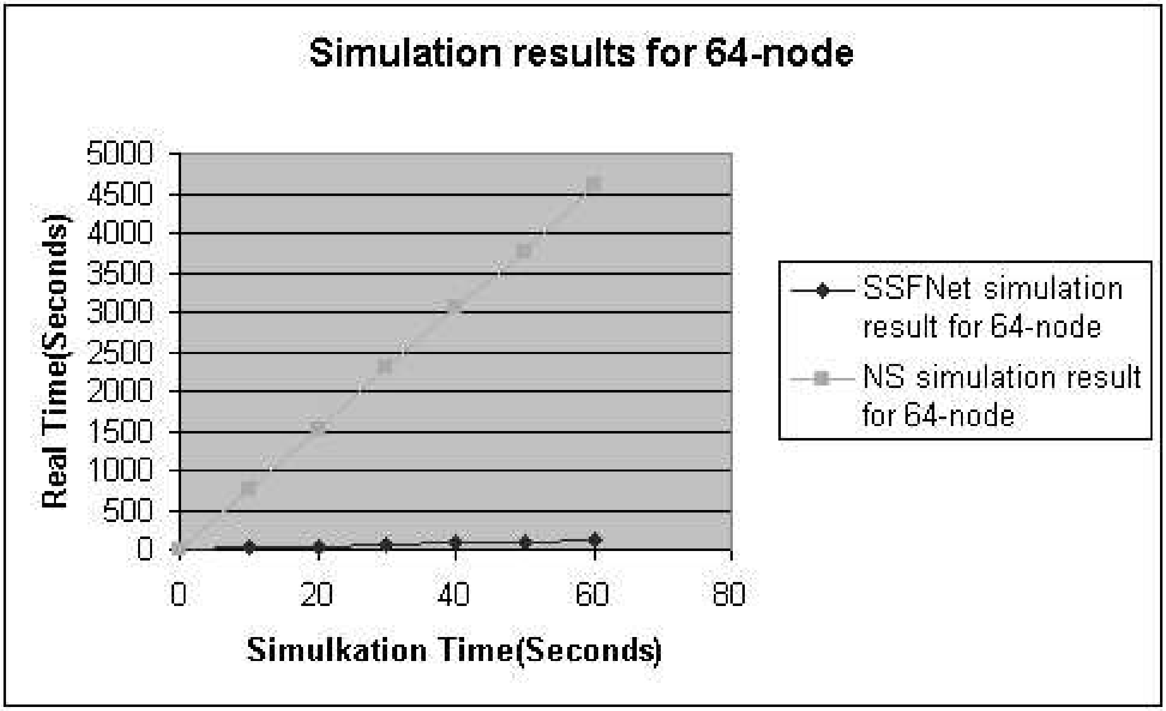
All the SSFNet simulation results are got on srs.cs.rpi.edu. For every configuration, simulation is run 5 times.

time(s)	1	2	3	4	5	Average
4 nodes	5.385	5.278	5.482	5.503	5.521	5.434
16 nodes	9.903	9.814	9.891	9.712	9.878	9.840
64 nodes	87.040	88.028	86.795	87.537	87.238	87.328

The above results are for 64-node network configuration. I also run the simulation for 4-node subconfiguration and 16-node subconfiguration. For the subconfiguration, I define an artificial node sink that is the destination of each flow outside the subconfiguration and all sources of traffic from outside into subconfiguration.

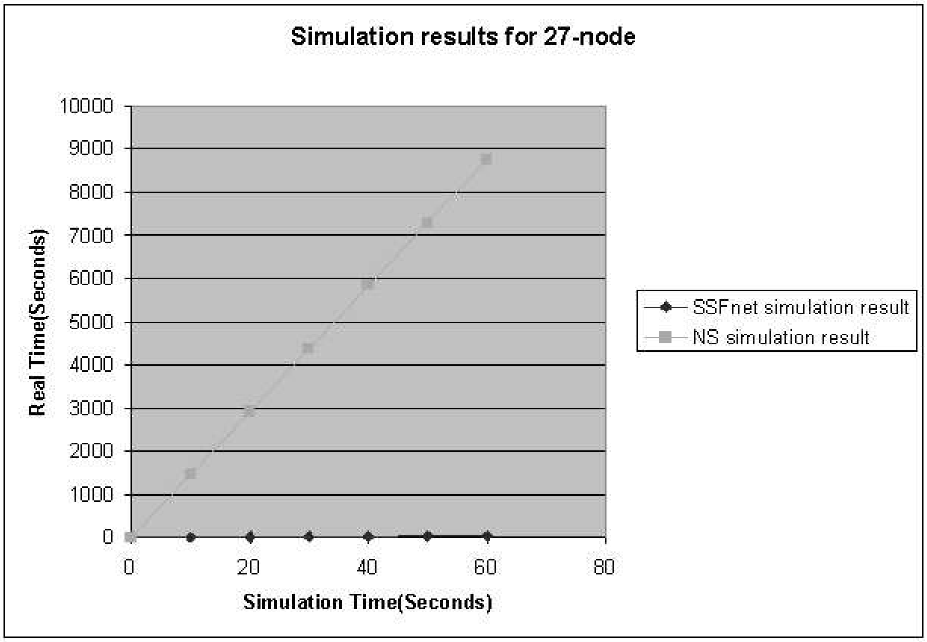
time(s)	1	2	3	4	5	Average
3 nodes	5.638	5.468	5.550	5.849	5.487	5.598
9 nodes(1)	6.818	6.747	7.240	7.012	7.243	7.012
9 nodes(2)	6.878	6.954	6.788	7.986	7.001	6.921
9 nodes(3)	7.123	6.813	6.974	6.851	6.969	6.946
27 nodes	27.579	27.725	27.029	27.541	27.869	27.549

The above table are for 27-node network configuration. Also 3-node subconfiguration and 9-node subconfiguration are presented. As 27-node network configuration is asymmetrical, 3 different 9-node subconfiguration are run respectively.


Simulation results for 64-node
Real Time(Seconds)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
20
40
60
80
Simulkation Time(Seconds)
SSFNet simulation result for 64-node
NS simulation result for 64-node

Real Time(Sec)	10	20	30	40	50	60
Simulation Time(Sec)	19.638	40.994	63.967	85.607	102.122	123.173

From the results, I can see that the simulation speed up with SSFNet over NS for 64-node configuration is approximately 38.


Simulation results for 27-node
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Real Time(Seconds)
0
20
40
60
80
Simulation Time(Seconds)
SSFnet simulation result
NS simulation result

Real Time(Sec)	10	20	30	40	50	60
Simulation Time(Sec)	7.206	12.782	17.989	24.290	30.623	36.506

From the results, I can see that the simulation speed up with SSFNet over NS for 27-node configuration is approximately 250.

6 Conclusions and future work

If this report, I used SSFNet to simulate two network configurations. From the above results, I can conclude that SSFNet is a very powerful discrete-event simulation tool.It significantly speeds up the simulation of the networks comparing with NS.

Future work will focus on implementing the Real-time On-line simulation method with SSFNet. Also I plan to extend our implementation to include TCP traffic.

7 Acknowledgements

First and last, I would like to thank prof. Boleslaw Szymanski, who inspired me to work on this project. Also my appreciation goes to Yu Liu, Kiran madnani and Anand Sastry, who helped me during the whole work. Finally I would like to thank Jim Cowie, who is one of the authors of SSFNet and ever helped me with SSFNet installation configuration.

References

[1] S.Bhatt, R. Fujimoto, A. Ogielski, K. Perumalla, "Parallel Simulation Techniques for Large-Scale Networks" IEEE Communications Magazine 1998

[2] Boleslaw Szymanski, Yu Liu, Anand Sastry, Kiran Madnani, "Real-time On-line Network Simulation" Department of Computer Science Technical Report at RPI March 2001

[3] Jim Cowie, Andy Ogielski, SSFNet Tutorial on www.ssfnet.org

[4] L.A.Law, M.G. McComas, "Simulation Software for Communication networks: the State of the Art" IEEE Communication Magazine, vol. 32, pp.44-50, 1994

[5] Andy T. Ogielski. SSFNet. Presentation at the DARPA Next Generation Internet Conference, Arlington, VA, Dec. 15-17, 1999

8 Appendix

8.1 27-node network DML source

```
_schema [_find .schemas.Net]
    Net [
    frequency 1000000000
    randomstream [
    generator ”MersenneTwister”
    stream ”seedstartingstring1234567890”
    reproducibility_level ”timeline”
    ]
    Net [id 1 _extends .networks.network91.Net]
    Net [id 2 _extends .networks.network92.Net]
    Net [id 3 _extends .networks.network93.Net]
    link [attach 1:1:2(8) attach 2:2:1(6) delay 0.01]
    link [attach 1:1:3(8) attach 2:3:1(7) delay 0.01]
    link [attach 1:2:3(7) attach 3:3:2(7) delay 0.01]
    link [attach 1:1:3(9) attach 3:1:1(6) delay 0.01]
    link [attach 1:3:2(7) attach 3:2:1(7) delay 0.01]
    link [attach 2:2:3(6) attach 3:1:2(6) delay 0.01]
    link [attach 2:3:2(7) attach 3:1:2(7) delay 0.01]
    link [attach 2:1:3(6) attach 3:2:1(8) delay 0.01]
    link [attach 2:3:3(6) attach 3:1:3(8) delay 0.01]
    traffic [
    pattern [
    client 1:1:11
    servers [nhi_range [from 1:1:22(0) to 1:1:23(0)] port 1600]
    ]
    pattern[
    client 1:1:12
    servers [nhi 1:1:21(0) port 1600]
    ]
    pattern[
    client 1:1:12
    servers [nhi 1:1:23(0) port 1600]
    ]
    pattern[
    client 1:1:13
    servers [nhi 1:1:21(0) port 1600]
    ]
    pattern[
    client 1:1:13
    servers [nhi 1:1:22(0) port 1600]
    ]
    pattern [
client 1:2:11
servers [nhi_range [from 1:2:22(0) to 1:2:23(0)] port 1600]
]
pattern[
client 1:2:12
servers [nhi 1:2:21(0) port 1600]
]
pattern[
client 1:2:12
servers [nhi 1:2:23(0) port 1600]
]
pattern[
client 1:2:13
servers [nhi 1:2:21(0) port 1600]
]
pattern[
client 1:2:13
servers [nhi 1:2:22(0) port 1600]
]
pattern [
client 1:3:11
servers [nhi_range [from 1:3:22(0) to 1:3:23(0)] port 1600]
]
pattern[
client 1:3:12
servers [nhi 1:3:21(0) port 1600]
]
pattern[
client 1:3:12
servers [nhi 1:3:23(0) port 1600]
]
pattern[
client 1:3:13
servers [nhi 1:3:21(0) port 1600]
]
pattern[
client 1:3:13
servers [nhi 1:3:22(0) port 1600]
]
pattern [
client 2:1:11
servers [nhi_range [from 2:1:22(0) to 2:1:23(0)] port 1600]
]
pattern[
client 2:1:12
servers [nhi 2:1:21(0) port 1600]
```

```
]
pattern[
client 2:1:12
servers [nhi 2:1:23(0) port 1600]
]
pattern[
client 2:1:13
servers [nhi 2:1:21(0) port 1600]
]
pattern[
client 2:1:13
servers [nhi 2:1:22(0) port 1600]
]
pattern [
client 2:2:11
servers [nhi_range [from 2:2:22(0) to 2:2:23(0)] port 1600]
]
pattern[
client 2:2:12
servers [nhi 2:2:21(0) port 1600]
]
pattern[
client 2:2:12
servers [nhi 2:2:23(0) port 1600]
]
pattern[
client 2:2:13
servers [nhi 2:2:21(0) port 1600]
]
pattern[
client 2:2:13
servers [nhi 2:2:22(0) port 1600]
]
pattern [
client 2:3:11
servers [nhi_range [from 2:3:22(0) to 2:3:23(0)] port 1600]
]
pattern[
client 2:3:12
servers [nhi 2:3:21(0) port 1600]
]
pattern[
client 2:3:12
servers [nhi 2:3:23(0) port 1600]
]
pattern[
client 2:3:13
servers [nhi 2:3:21(0) port 1600]
]
pattern[
client 2:3:13
servers [nhi 2:3:22(0) port 1600]
]
pattern [
client 3:1:11
servers [nhi_range [from 3:1:22(0) to 3:1:23(0)] port 1600]
]
pattern[
client 3:1:12
servers [nhi 3:1:21(0) port 1600]
]
pattern[
client 3:1:12
servers [nhi 3:1:23(0) port 1600]
]
pattern[
client 3:1:13
servers [nhi 3:1:21(0) port 1600]
]
pattern[
client 3:1:13
servers [nhi 3:1:22(0) port 1600]
]
pattern [
client 3:2:11
servers [nhi_range [from 3:2:22(0) to 3:2:23(0)] port 1600]
]
pattern[
client 3:2:12
servers [nhi 3:2:21(0) port 1600]
]
pattern[
client 3:2:12
servers [nhi 3:2:23(0) port 1600]
]
pattern[
client 3:2:13
servers [nhi 3:2:21(0) port 1600]
]
pattern[
client 3:2:13
servers [nhi 3:2:22(0) port 1600]
```

```
]
pattern [
client 3:3:11
servers [nhi_range [from 3:3:22(0) to 3:3:23(0)] port 1600]
]
pattern[ client 3:3:12
servers [nhi 3:3:21(0) port 1600]
]
pattern[
client 3:3:12
servers [nhi 3:3:23(0) port 1600]
]
pattern[
client 3:3:13
servers [nhi 3:3:21(0) port 1600]
]
pattern[
client 3:3:13
servers [nhi 3:3:22(0) port 1600] ]
pattern [
client 1:1:11
servers [nhi 1:2:21(0) port 1600]
]
pattern [
client 1:1:31
servers [nhi 1:2:41(0) port 1600]
]
pattern [
client 1:1:11
servers [nhi 1:3:21(0) port 1600]
]
pattern [
client 1:1:31
servers [nhi 1:3:41(0) port 1600]
]
pattern [
client 1:1:12
servers [nhi 1:2:22(0) port 1600]
]
pattern [
client 1:1:32
servers [nhi 1:2:42(0) port 1600]
]
pattern [
client 1:1:12
servers [nhi 1:3:22(0) port 1600]
```

```
]
pattern [
client 1:1:32
servers [nhi 1:3:42(0) port 1600]
]
pattern [
client 1:1:13
servers [nhi 1:2:23(0) port 1600]
]
pattern [
client 1:1:33
servers [nhi 1:2:43(0) port 1600]
]
pattern [
client 1:1:13
servers [nhi 1:3:23(0) port 1600]
]
pattern [
client 1:1:33
servers [nhi 1:3:43(0) port 1600]
]
pattern [
client 1:2:11
servers [nhi 1:1:21(0) port 1600]
]
pattern [
client 1:2:31
servers [nhi 1:1:41(0) port 1600]
]
pattern [
client 1:2:11
servers [nhi 1:3:21(0) port 1600]
]
pattern [
client 1:2:31
servers [nhi 1:3:41(0) port 1600]
]
pattern [
client 1:2:12
servers [nhi 1:1:22(0) port 1600]
]
pattern [
client 1:2:32
servers [nhi 1:1:42(0) port 1600]
]
pattern [
client 1:2:12
servers [nhi 1:3:22(0) port 1600]
]
pattern [
client 1:2:32
servers [nhi 1:3:42(0) port 1600]
]
pattern [
client 1:2:13
servers [nhi 1:1:23(0) port 1600]
]
pattern [
client 1:2:33
servers [nhi 1:1:43(0) port 1600]
]
pattern [
client 1:2:13
servers [nhi 1:3:23(0) port 1600]
]
pattern [
client 1:2:33
servers [nhi 1:3:43(0) port 1600] ]
pattern [
client 1:3:11
servers [nhi 1:1:21(0) port 1600]
]
pattern [
client 1:3:31
servers [nhi 1:1:41(0) port 1600]
]
pattern [
client 1:3:11
servers [nhi 1:2:21(0) port 1600]
]
pattern [
client 1:3:31
servers [nhi 1:2:41(0) port 1600]
]
pattern [
client 1:3:12
servers [nhi 1:1:22(0) port 1600]
]
pattern [
client 1:3:32
servers [nhi 1:1:42(0) port 1600]
]
```

```
pattern [
client 1:3:12
servers [nhi 1:2:22(0) port 1600]
]
pattern [
client 1:3:32
servers [nhi 1:2:42(0) port 1600]
]
pattern [
client 1:3:13
servers [nhi 1:1:23(0) port 1600]
]
pattern [
client 1:3:33
servers [nhi 1:1:43(0) port 1600]
]
pattern [
client 1:3:13
servers [nhi 1:2:23(0) port 1600]
]
pattern [
client 1:3:33
servers [nhi 1:2:43(0) port 1600] ]
pattern [
client 2:1:11
servers [nhi 2:2:21(0) port 1600]
]
pattern [
client 2:1:31
servers [nhi 2:2:41(0) port 1600]
]
pattern [
client 2:1:11
servers [nhi 2:3:21(0) port 1600]
]
pattern [
client 2:1:31
servers [nhi 2:3:41(0) port 1600]
]
pattern [
client 2:1:12
servers [nhi 2:2:22(0) port 1600]
]
pattern [
client 2:1:32
servers [nhi 2:2:42(0) port 1600] ]
```

```
pattern [
client 2:1:12
servers [nhi 2:3:22(0) port 1600]
]
pattern [
client 2:1:32
servers [nhi 2:3:42(0) port 1600]
]
pattern [
client 2:1:13
servers [nhi 2:2:23(0) port 1600]
]
pattern [
client 2:1:33
servers [nhi 2:2:43(0) port 1600]
]
pattern [
client 2:1:13
servers [nhi 2:3:23(0) port 1600]
]
pattern [
client 2:1:33
servers [nhi 2:3:43(0) port 1600]
]
pattern [
client 2:1:14
servers [nhi 2:2:24(0) port 1600]
]
pattern [
client 2:1:34
servers [nhi 2:2:44(0) port 1600] ]
pattern [
client 2:1:14
servers [nhi 2:3:24(0) port 1600]
]
pattern [
client 2:1:34
servers [nhi 2:3:44(0) port 1600]
]
pattern [
client 2:2:11
servers [nhi 2:1:21(0) port 1600]
]
pattern [
client 2:2:31
servers [nhi 2:1:41(0) port 1600]
```

```
]
pattern [
client 2:2:11
servers [nhi 2:3:21(0) port 1600]
]
pattern [
client 2:2:31
servers [nhi 2:3:41(0) port 1600]
]
pattern [
client 2:2:12
servers [nhi 2:1:22(0) port 1600]
]
pattern [
client 2:2:32
servers [nhi 2:1:42(0) port 1600]
]
pattern [
client 2:2:12
servers [nhi 2:3:22(0) port 1600]
]
pattern [
client 2:2:32
servers [nhi 2:3:42(0) port 1600] ]
pattern [
client 2:2:13
servers [nhi 2:1:23(0) port 1600]
]
pattern [
client 2:2:33
servers [nhi 2:1:43(0) port 1600]
]
pattern [
client 2:2:13
servers [nhi 2:3:23(0) port 1600]
]
pattern [
client 2:2:33
servers [nhi 2:3:43(0) port 1600]
]
pattern [
client 2:3:11
servers [nhi 2:1:21(0) port 1600]
]
pattern [
client 2:3:31
```

```
servers [nhi 2:1:41(0) port 1600]
]
pattern [
client 2:3:11
servers [nhi 2:2:21(0) port 1600]
]
pattern [
client 2:3:31
servers [nhi 2:2:41(0) port 1600]
]
pattern [
client 2:3:12
servers [nhi 2:1:22(0) port 1600]
]
pattern [
client 2:3:32
servers [nhi 2:1:42(0) port 1600]
]
pattern [
client 2:3:12
servers [nhi 2:2:22(0) port 1600]
]
pattern [
client 2:3:32
servers [nhi 2:2:42(0) port 1600]
]
pattern [
client 2:3:13
servers [nhi 2:1:23(0) port 1600]
]
pattern [
client 2:3:33
servers [nhi 2:1:43(0) port 1600]
]
pattern [
client 2:3:13
servers [nhi 2:2:23(0) port 1600]
]
pattern [
client 2:3:33
servers [nhi 2:2:43(0) port 1600]
]
pattern [
client 2:3:14
servers [nhi 2:1:24(0) port 1600]
]
```

```
pattern [
client 2:3:34
servers [nhi 2:1:44(0) port 1600]
]
pattern [ client 2:3:14
servers [nhi 2:2:24(0) port 1600]
]
pattern [
client 2:3:34
servers [nhi 2:2:44(0) port 1600]
]
pattern [
client 3:1:11
servers [nhi 3:2:21(0) port 1600]
]
pattern [
client 3:1:31
servers [nhi 3:2:41(0) port 1600]
]
pattern [
client 3:1:11
servers [nhi 3:3:21(0) port 1600]
]
pattern [
client 3:1:31
servers [nhi 3:3:41(0) port 1600]
]
pattern [
client 3:1:12
servers [nhi 3:2:22(0) port 1600]
] pattern [
client 3:1:32
servers [nhi 3:2:42(0) port 1600]
]
pattern [
client 3:1:12
servers [nhi 3:3:22(0) port 1600]
]
pattern [
client 3:1:32
servers [nhi 3:3:42(0) port 1600]
]
pattern [ client 3:1:13
servers [nhi 3:2:23(0) port 1600] ]
pattern [
client 3:1:33
servers [nhi 3:2:43(0) port 1600]
]
pattern [
client 3:1:13
servers [nhi 3:3:23(0) port 1600]
]
pattern [
client 3:1:33
servers [nhi 3:3:43(0) port 1600]
]
pattern [
client 3:2:11
servers [nhi 3:1:21(0) port 1600]
]
pattern [
client 3:2:31
servers [nhi 3:1:41(0) port 1600]
]
pattern [
client 3:2:11
servers [nhi 3:3:21(0) port 1600]
]
pattern [
client 3:2:31
servers [nhi 3:3:41(0) port 1600]
]
pattern [
client 3:2:12
servers [nhi 3:1:22(0) port 1600]
]
pattern [
client 3:2:32
servers [nhi 3:1:42(0) port 1600]
]
pattern [
client 3:2:12
servers [nhi 3:3:22(0) port 1600]
]
pattern [
client 3:2:32
servers [nhi 3:3:42(0) port 1600]
]
pattern [
client 3:2:13
servers [nhi 3:1:23(0) port 1600]
]
```

```
pattern [
client 3:2:33
servers [nhi 3:1:43(0) port 1600]
]
pattern [
client 3:2:13
servers [nhi 3:3:23(0) port 1600]
]
pattern [
client 3:2:33
servers [nhi 3:3:43(0) port 1600] ]
pattern [
client 3:3:11
servers [nhi 3:1:21(0) port 1600]
]
pattern [
client 3:3:31
servers [nhi 3:1:41(0) port 1600]
]
pattern [
client 3:3:11
servers [nhi 3:2:21(0) port 1600]
]
pattern [
client 3:3:31
servers [nhi 3:2:41(0) port 1600]
]
pattern [
client 3:3:12
servers [nhi 3:1:22(0) port 1600]
]
pattern [
client 3:3:32
servers [nhi 3:1:42(0) port 1600]
]
pattern [
client 3:3:12
servers [nhi 3:2:22(0) port 1600]
] pattern [
client 3:3:32
servers [nhi 3:2:42(0) port 1600]
]
pattern [
client 3:3:13
servers [nhi 3:1:23(0) port 1600]
]
```

```
pattern [
client 3:3:33
servers [nhi 3:1:43(0) port 1600]
]
pattern [
client 3:3:13
servers [nhi 3:2:23(0) port 1600]
]
pattern [
client 3:3:33
servers [nhi 3:2:43(0) port 1600]
]
pattern [
client 1:1:11
servers [nhi 2:1:21(0) port 1600]
]
pattern [
client 1:1:31
servers [nhi 2:1:41(0) port 1600]
]
pattern [
client 1:1:11
servers [nhi 3:1:21(0) port 1600]
]
pattern [
client 1:1:31
servers [nhi 3:1:41(0) port 1600]
]
pattern [
client 1:1:12
servers [nhi 2:1:22(0) port 1600]
]
pattern [
client 1:1:32
servers [nhi 2:1:42(0) port 1600]
]
pattern [
client 1:1:12
servers [nhi 3:1:22(0) port 1600]
]
pattern [
client 1:1:32
servers [nhi 3:1:42(0) port 1600]
]
pattern [
client 1:1:13
servers [nhi 2:1:23(0) port 1600]
]
pattern [
client 1:1:33
servers [nhi 2:1:43(0) port 1600]
]
pattern [
client 1:1:13
servers [nhi 3:1:23(0) port 1600]
]
pattern [
client 1:1:33
servers [nhi 3:1:43(0) port 1600]
]
pattern [
client 1:2:11
servers [nhi 2:2:21(0) port 1600]
]
pattern [
client 1:2:31
servers [nhi 2:2:41(0) port 1600]
]
pattern [
client 1:2:11
servers [nhi 3:2:21(0) port 1600]
]
pattern [
client 1:2:31
servers [nhi 3:2:41(0) port 1600]
]
pattern [
client 1:2:12
servers [nhi 2:2:22(0) port 1600]
]
pattern [
client 1:2:32
servers [nhi 2:2:42(0) port 1600]
]
pattern [
client 1:2:12
servers [nhi 3:2:22(0) port 1600]
]
pattern [
client 1:2:32
servers [nhi 3:2:42(0) port 1600]
]
```

```
pattern [
client 1:2:13
servers [nhi 2:2:23(0) port 1600]
]
pattern [
client 1:2:33
servers [nhi 2:2:43(0) port 1600]
]
pattern [
client 1:2:13
servers [nhi 3:2:23(0) port 1600]
]
pattern [
client 1:2:33
servers [nhi 3:2:43(0) port 1600]
]
pattern [
client 1:3:11
servers [nhi 2:3:21(0) port 1600]
]
pattern [
client 1:3:31
servers [nhi 2:3:41(0) port 1600]
]
pattern [
client 1:3:11
servers [nhi 3:3:21(0) port 1600]
]
pattern [
client 1:3:31
servers [nhi 3:3:41(0) port 1600]
]
pattern [
client 1:3:12
servers [nhi 2:3:22(0) port 1600]
]
pattern [
client 1:3:32
servers [nhi 2:3:42(0) port 1600]
]
pattern [
client 1:3:12
servers [nhi 3:3:22(0) port 1600]
]
pattern [
client 1:3:32
```

```
servers [nhi 3:3:42(0) port 1600]
]
pattern [
client 1:3:13
servers [nhi 2:3:23(0) port 1600]
]
pattern [
client 1:3:33
servers [nhi 2:3:43(0) port 1600]
]
pattern [
client 1:3:13
servers [nhi 3:3:23(0) port 1600]
]
pattern [
client 1:3:33
servers [nhi 3:3:43(0) port 1600]
]
pattern [
client 2:1:11
servers [nhi 1:1:21(0) port 1600]
]
pattern [
client 2:1:31
servers [nhi 1:1:41(0) port 1600]
]
pattern [
client 2:1:11
servers [nhi 3:1:21(0) port 1600]
]
pattern [ client 2:1:31
servers [nhi 3:1:41(0) port 1600]
]
pattern [
client 2:1:12
servers [nhi 1:1:22(0) port 1600]
]
pattern [
client 2:1:32
servers [nhi 1:1:42(0) port 1600]
]
pattern [
client 2:1:12
servers [nhi 3:1:22(0) port 1600]
]
pattern [
client 2:1:32
servers [nhi 3:1:42(0) port 1600]
]
pattern [
client 2:1:13
servers [nhi 1:1:23(0) port 1600]
]
pattern [
client 2:1:33
servers [nhi 1:1:43(0) port 1600]
]
pattern [
client 2:1:13
servers [nhi 3:1:23(0) port 1600]
]
pattern [
client 2:1:33
servers [nhi 3:1:43(0) port 1600]
]
pattern [
client 2:2:11
servers [nhi 1:2:21(0) port 1600]
]
pattern [
client 2:2:31
servers [nhi 1:2:41(0) port 1600]
]
pattern [
client 2:2:11
servers [nhi 3:2:21(0) port 1600]
]
pattern [
client 2:2:31
servers [nhi 3:2:41(0) port 1600]
]
pattern [
client 2:2:12
servers [nhi 1:2:22(0) port 1600]
]
pattern [
client 2:2:32
servers [nhi 1:2:42(0) port 1600]
]
pattern [
client 2:2:12
servers [nhi 3:2:22(0) port 1600]
```

```
]
pattern [
client 2:2:32
servers [nhi 3:2:42(0) port 1600]
]
pattern [
client 2:2:13
servers [nhi 1:2:23(0) port 1600]
]
pattern [
client 2:2:33
servers [nhi 1:2:43(0) port 1600]
]
pattern [
client 2:2:13
servers [nhi 3:2:23(0) port 1600]
]
pattern [
client 2:2:33
servers [nhi 3:2:43(0) port 1600]
]
pattern [
client 2:3:11
servers [nhi 1:3:21(0) port 1600]
]
pattern [
client 2:3:31
servers [nhi 1:3:41(0) port 1600]
]
pattern [
client 2:3:11
servers [nhi 3:3:21(0) port 1600]
]
pattern [
client 2:3:31
servers [nhi 3:3:41(0) port 1600]
]
pattern [
client 2:3:12
servers [nhi 1:3:22(0) port 1600]
]
pattern [
client 2:3:32
servers [nhi 1:3:42(0) port 1600]
]
pattern [
client 2:3:12
servers [nhi 3:3:22(0) port 1600]
]
pattern [
client 2:3:32
servers [nhi 3:3:42(0) port 1600]
]
pattern [
client 2:3:13
servers [nhi 1:3:23(0) port 1600]
]
pattern [
client 2:3:33
servers [nhi 1:3:43(0) port 1600]
]
pattern [
client 2:3:13
servers [nhi 3:3:23(0) port 1600]
]
pattern [
client 2:3:33
servers [nhi 3:3:43(0) port 1600]
]
pattern [
client 3:1:11
servers [nhi 2:1:21(0) port 1600]
]
pattern [
client 3:1:31
servers [nhi 2:1:41(0) port 1600]
]
pattern [
client 3:1:12
servers [nhi 1:1:22(0) port 1600]
]
pattern [
client 3:1:32
servers [nhi 1:1:42(0) port 1600]
]
pattern [
client 3:1:12
servers [nhi 2:1:22(0) port 1600]
]
pattern [
client 3:1:32
servers [nhi 2:1:42(0) port 1600]
```

```
]
pattern [
client 3:1:13
servers [nhi 1:1:23(0) port 1600]
]
pattern [
client 3:1:33
servers [nhi 1:1:43(0) port 1600]
]
pattern [
client 3:1:13
servers [nhi 2:1:23(0) port 1600]
]
pattern [
client 3:1:33
servers [nhi 2:1:43(0) port 1600]
]
pattern [
client 3:2:11
servers [nhi 1:2:21(0) port 1600]
]
pattern [
client 3:2:31
servers [nhi 1:2:41(0) port 1600]
]
pattern [
client 3:2:11
servers [nhi 2:2:21(0) port 1600]
]
pattern [
client 3:2:31
servers [nhi 2:2:41(0) port 1600]
]
pattern [
client 3:2:12
servers [nhi 1:2:22(0) port 1600]
]
pattern [
client 3:2:32
servers [nhi 1:2:42(0) port 1600]
]
pattern [
client 3:2:12
servers [nhi 2:2:22(0) port 1600]
]
pattern [
client 3:2:32
servers [nhi 2:2:42(0) port 1600]
]
pattern [
client 3:2:13
servers [nhi 1:2:23(0) port 1600]
]
pattern [
client 3:2:33
servers [nhi 1:2:43(0) port 1600]
]
pattern [
client 3:2:13
servers [nhi 2:2:23(0) port 1600]
]
pattern [
client 3:2:33
servers [nhi 2:2:43(0) port 1600]
]
pattern [
client 3:3:11
servers [nhi 1:3:21(0) port 1600]
]
pattern [
client 3:3:31
servers [nhi 1:3:41(0) port 1600]
]
pattern [
client 3:3:11
servers [nhi 2:3:21(0) port 1600]
]
pattern [
client 3:3:31
servers [nhi 2:3:41(0) port 1600]
]
pattern [
client 3:3:12
servers [nhi 1:3:22(0) port 1600]
]
pattern [
client 3:3:32
servers [nhi 1:3:42(0) port 1600]
]
pattern [
client 3:3:12
servers [nhi 2:3:22(0) port 1600]
```

```
]
pattern [
client 3:3:32
servers [nhi 2:3:42(0) port 1600]
]
pattern [
client 3:3:13
servers [nhi 1:3:23(0) port 1600]
]
pattern [
client 3:3:33
servers [nhi 1:3:43(0) port 1600]
]
pattern [
client 3:3:13
servers [nhi 2:3:23(0) port 1600]
]
pattern [
client 3:3:33
servers [nhi 2:3:43(0) port 1600]
]
]
]
networks [
network91 [
Net [
Net [id 1 _extends .networks.network3.Net]
Net [id 2 _extends .networks.network3.Net]
Net [id 3 _extends .networks.network3.Net]
link [attach 1:2(6) attach 2:1(6) delay 0.01]
link [attach 1:2(7) attach 2:2(6) delay 0.01]
link [attach 1:3(6) attach 2:2(7) delay 0.01]
link [attach 1:3(7) attach 3:2(6) delay 0.01]
link [attach 2:2(8) attach 3:3(6) delay 0.01]
link [attach 2:3(6) attach 3:1(6) delay 0.01]
]
]
network92 [
Net [
Net [id 1 _extends .networks.network3.Net]
Net [id 2 _extends .networks.network3.Net]
Net [id 3 _extends .networks.network3.Net]
link [attach 1:1(6) attach 2:2(6) delay 0.01]
link [attach 1:2(6) attach 3:1(6) delay 0.01]
link [attach 2:2(7) attach 3:2(6) delay 0.01]
]
```

```
]
network93 [
Net [
Net [id 1 _extends .networks.network3.Net]
Net [id 2 _extends .networks.network3.Net]
Net [id 3 _extends .networks.network3.Net]
link [attach 1:3(6) attach 2:1(6) delay 0.01]
link [attach 1:3(7) attach 3:2(6) delay 0.01]
]
]
network3 [
Net [
router [
idrange [from 1 to 3]
graph [ProtocolSession [name ip use SSF.OS.IP]]
interface [idrange [from 4 to 9] buffer 8000 _extends .dictionary.15BaseT]
interface [idrange [from 0 to 3] buffer 16000 _extends .dictionary.10Baset]
route [dest default interface 4]
]
link [attach 1(4) attach 2(4) delay 0.01]
link [attach 1(5) attach 3(4) delay 0.01]
link [attach 2(5) attach 3(5) delat 0.01]
link [attach 1(0) attach 11(0) delay 0]
link [attach 1(1) attach 21(0) delay 0]
link [attach 1(2) attach 31(0) delay 0]
link [attach 1(3) attach 41(0) delay 0]
link [attach 2(0) attach 12(0) delay 0]
link [attach 2(1) attach 22(0) delay 0]
link [attach 2(2) attach 32(0) delay 0]
link [attach 2(3) attach 42(0) delay 0]
link [attach 3(0) attach 13(0) delay 0]
link [attach 3(1) attach 23(0) delay 0]
link [attach 3(2) attach 33(0) delay 0]
link [attach 3(3) attach 43(0) delay 0]
host [idrange [from 11 to 13]
_extends .dictionary.standardClient
]
host [idrange [from 21 to 23]
_extends .dictionary.standardServer
]
host [idrange [from 31 to 33]
_extends .dictionary.standardClient1
]
host [idrange [from 41 to 43]
_extends .dictionary.standardServer1
]
```

```
]
]
]
dictionary [
10BaseT [
bitrate 10000000
latency 0
]
15BaseT [
bitrate 1500000
latency 0.01
]
udpinit [
max_datagram_size 10000
debug false
]
appsession [
request_size 256
datagram_size 500
send_interval 0.01
show_report true
debug true
]
appsession1 [
request_size 256
datagram_size 500
send_interval 0.00005
show_report true
debug true
]
standardClient [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
start_time 1.0
start_window 1.0
file_size 3000000
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardClient1 [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
start_time 1.0
start_window 1.0
file_size 298500000
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardServer [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name server use SSF.OS.UDP.test.udpStreamServer
port 1600
client_limit 10
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.send_interval
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardServer1 [
interface [id 0 _extends .dictionary.10BaseT]
```

```
    route [dest default interface 0]
    graph [
    ProtocolSession [
    name server use SSF.OS.UDP.test.udpStreamServer
    port 1600
    client_limit 10
    _find .dictionary.appsession1.request_size
    _find .dictionary.appsession1.datagram_size
    _find .dictionary.appsession1.send_interval
    _find .dictionary.appsession1.show_report
    _find .dictionary.appsession1.debug
    ]
    ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
    ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster _find .dictionary.udpinit] ProtocolSession [name ip use SSF.OS.IP] ]
    ]
    ]
```

8.2 16-node subnetwork DML source

```
_schema [_find .schemas.Net]
    Net [
    frequency 1000000000
    randomstream [
    generator ”MersenneTwister”
    stream ”seedstartingstring1234567890”
    reproducibility_level ”host” ]
    Net [id 0 _extends .networks.network4.Net]
    Net [id 1 _extends .networks.network4.Net]
    Net [id 2 _extends .networks.network4.Net]
    Net [id 3 _extends .networks.network4.Net]
    router [
    idrange [from 250 to 297]
    graph [ProtocolSession [name ip use SSF.OS.IP]]
    interface [id 2 buffer 8000 _extends .dictionary.15BaseT]
    interface [idrange [from 0 to 1] buffer 16000 _extends .dictionary.10BaseT]
    route [dest default interface 2]
    ]
    router [
    id 298
    graph [ProtocolSession [name ip use SSF.OS.IP]]
    interface [idrange [from 2 to 17] buffer 8000 _extends .dictionary.15BaseT]
    interface [idrange [from 0 to 1] buffer 16000 _extends .dictionary.10BaseT]
    route [dest default interface 2]
    ]
    link [attach 0:4(7) attach 1:3(7) delay 0.01]
link [attach 0:4(8) attach 2:2(7) delay 0.01]
link [attach 0:4(9) attach 3:1(7) delay 0.01]
link [attach 1:3(8) attach 2:2(8) delay 0.01]
link [attach 1:3(9) attach 3:1(8) delay 0.01]
link [attach 2:2(9) attach 3:1(9) delay 0.01]
link [attach 0:1(10) attach 250(2) delay 0.01]
link [attach 0:1(11) attach 251(2) delay 0.01]
link [attach 0:1(12) attach 252(2) delay 0.01]
link [attach 0:2(10) attach 253(2) delay 0.01]
link [attach 0:2(11) attach 254(2) delay 0.01]
link [attach 0:2(12) attach 255(2) delay 0.01]
link [attach 0:3(10) attach 256(2) delay 0.01]
link [attach 0:3(11) attach 257(2) delay 0.01]
link [attach 0:3(12) attach 258(2) delay 0.01]
link [attach 0:4(10) attach 259(2) delay 0.01]
link [attach 0:4(11) attach 260(2) delay 0.01]
link [attach 0:4(12) attach 261(2) delay 0.01]
link [attach 1:1(10) attach 262(2) delay 0.01]
link [attach 1:1(11) attach 263(2) delay 0.01]
link [attach 1:1(12) attach 264(2) delay 0.01]
link [attach 1:2(10) attach 265(2) delay 0.01]
link [attach 1:2(11) attach 266(2) delay 0.01]
link [attach 1:2(12) attach 267(2) delay 0.01]
link [attach 1:3(10) attach 268(2) delay 0.01]
link [attach 1:3(11) attach 269(2) delay 0.01]
link [attach 1:3(12) attach 270(2) delay 0.01]
link [attach 1:4(10) attach 271(2) delay 0.01]
link [attach 1:4(11) attach 272(2) delay 0.01]
link [attach 1:4(12) attach 273(2) delay 0.01]
link [attach 2:1(10) attach 274(2) delay 0.01]
link [attach 2:1(11) attach 275(2) delay 0.01]
link [attach 2:1(12) attach 276(2) delay 0.01]
link [attach 2:2(10) attach 277(2) delay 0.01]
link [attach 2:2(11) attach 278(2) delay 0.01]
link [attach 2:2(12) attach 279(2) delay 0.01]
link [attach 2:3(10) attach 280(2) delay 0.01]
link [attach 2:3(11) attach 281(2) delay 0.01]
link [attach 2:3(12) attach 282(2) delay 0.01]
link [attach 2:4(10) attach 283(2) delay 0.01]
link [attach 2:4(11) attach 284(2) delay 0.01]
link [attach 2:4(12) attach 285(2) delay 0.01]
link [attach 3:1(10) attach 286(2) delay 0.01]
link [attach 3:1(11) attach 287(2) delay 0.01]
link [attach 3:1(12) attach 288(2) delay 0.01]
link [attach 3:2(10) attach 289(2) delay 0.01] link [attach 3:2(11) attach 290(2) delay 0.01]
link [attach 3:2(12) attach 291(2) delay 0.01]
link [attach 3:3(10) attach 292(2) delay 0.01]
link [attach 3:3(11) attach 293(2) delay 0.01]
link [attach 3:3(12) attach 294(2) delay 0.01]
link [attach 3:4(10) attach 295(2) delay 0.01]
link [attach 3:4(11) attach 296(2) delay 0.01]
link [attach 3:4(12) attach 297(2) delay 0.01]
link [attach 0:1(13) attach 298(2) delay 0.01]
link [attach 0:2(13) attach 298(3) delay 0.01]
link [attach 0:3(13) attach 298(4) delay 0.01]
link [attach 0:4(13) attach 298(5) delay 0.01]
link [attach 1:1(13) attach 298(6) delay 0.01]
link [attach 1:2(13) attach 298(7) delay 0.01]
link [attach 1:3(13) attach 298(8) delay 0.01]
link [attach 1:4(13) attach 298(9) delay 0.01]
link [attach 2:1(13) attach 298(10) delay 0.01]
link [attach 2:2(13) attach 298(11) delay 0.01]
link [attach 2:3(13) attach 298(12) delay 0.01]
link [attach 2:4(13) attach 298(13) delay 0.01]
link [attach 3:1(13) attach 298(14) delay 0.01]
link [attach 3:2(13) attach 298(15) delay 0.01]
link [attach 3:3(13) attach 298(16) delay 0.01]
link [attach 3:4(13) attach 298(17) delay 0.01]
link [attach 250(0) attach 100(0) delay 0]
link [attach 251(0) attach 101(0) delay 0]
link [attach 252(0) attach 102(0) delay 0]
link [attach 253(0) attach 103(0) delay 0]
link [attach 254(0) attach 104(0) delay 0]
link [attach 255(0) attach 105(0) delay 0]
link [attach 256(0) attach 106(0) delay 0]
link [attach 257(0) attach 107(0) delay 0]
link [attach 258(0) attach 108(0) delay 0]
link [attach 259(0) attach 109(0) delay 0]
link [attach 260(0) attach 110(0) delay 0]
link [attach 261(0) attach 111(0) delay 0]
link [attach 262(0) attach 112(0) delay 0]
link [attach 263(0) attach 113(0) delay 0]
link [attach 264(0) attach 114(0) delay 0]
link [attach 265(0) attach 115(0) delay 0]
link [attach 266(0) attach 116(0) delay 0]
link [attach 267(0) attach 117(0) delay 0]
link [attach 268(0) attach 118(0) delay 0]
link [attach 269(0) attach 119(0) delay 0]
link [attach 270(0) attach 120(0) delay 0]
link [attach 271(0) attach 121(0) delay 0]
link [attach 272(0) attach 122(0) delay 0]
link [attach 273(0) attach 123(0) delay 0]
link [attach 274(0) attach 124(0) delay 0]
link [attach 275(0) attach 125(0) delay 0]
link [attach 276(0) attach 126(0) delay 0]
link [attach 277(0) attach 127(0) delay 0]
link [attach 278(0) attach 128(0) delay 0]
link [attach 279(0) attach 129(0) delay 0]
link [attach 280(0) attach 130(0) delay 0]
link [attach 281(0) attach 131(0) delay 0]
link [attach 282(0) attach 132(0) delay 0]
link [attach 283(0) attach 133(0) delay 0]
link [attach 284(0) attach 134(0) delay 0]
link [attach 285(0) attach 135(0) delay 0]
link [attach 286(0) attach 136(0) delay 0]
link [attach 287(0) attach 137(0) delay 0]
link [attach 288(0) attach 138(0) delay 0]
link [attach 289(0) attach 139(0) delay 0]
link [attach 290(0) attach 140(0) delay 0]
link [attach 291(0) attach 141(0) delay 0]
link [attach 292(0) attach 142(0) delay 0]
link [attach 293(0) attach 143(0) delay 0]
link [attach 294(0) attach 144(0) delay 0]
link [attach 295(0) attach 145(0) delay 0]
link [attach 296(0) attach 146(0) delay 0]
link [attach 297(0) attach 147(0) delay 0]
link [attach 298(0) attach 148(0) delay 0]
link [attach 250(1) attach 150(0) delay 0]
link [attach 251(1) attach 151(0) delay 0]
link [attach 252(1) attach 152(0) delay 0]
link [attach 253(1) attach 153(0) delay 0]
link [attach 254(1) attach 154(0) delay 0]
link [attach 255(1) attach 155(0) delay 0]
link [attach 256(1) attach 156(0) delay 0]
link [attach 257(1) attach 157(0) delay 0]
link [attach 258(1) attach 158(0) delay 0]
link [attach 259(1) attach 159(0) delay 0]
link [attach 260(1) attach 160(0) delay 0]
link [attach 261(1) attach 161(0) delay 0]
link [attach 262(1) attach 162(0) delay 0]
link [attach 263(1) attach 163(0) delay 0]
link [attach 264(1) attach 164(0) delay 0]
link [attach 265(1) attach 165(0) delay 0]
link [attach 266(1) attach 166(0) delay 0]
link [attach 267(1) attach 167(0) delay 0]
link [attach 268(1) attach 168(0) delay 0]
link [attach 269(1) attach 169(0) delay 0]
link [attach 270(1) attach 170(0) delay 0]
link [attach 271(1) attach 171(0) delay 0]
link [attach 272(1) attach 172(0) delay 0]
link [attach 273(1) attach 173(0) delay 0]
link [attach 274(1) attach 174(0) delay 0]
link [attach 275(1) attach 175(0) delay 0]
link [attach 276(1) attach 176(0) delay 0]
link [attach 277(1) attach 177(0) delay 0]
link [attach 278(1) attach 178(0) delay 0]
link [attach 279(1) attach 179(0) delay 0]
link [attach 280(1) attach 180(0) delay 0]
link [attach 281(1) attach 181(0) delay 0]
link [attach 282(1) attach 182(0) delay 0]
link [attach 283(1) attach 183(0) delay 0]
link [attach 284(1) attach 184(0) delay 0]
link [attach 285(1) attach 185(0) delay 0]
link [attach 286(1) attach 186(0) delay 0]
link [attach 287(1) attach 187(0) delay 0]
link [attach 288(1) attach 188(0) delay 0]
link [attach 289(1) attach 189(0) delay 0]
link [attach 290(1) attach 190(0) delay 0]
link [attach 291(1) attach 191(0) delay 0]
link [attach 292(1) attach 192(0) delay 0]
link [attach 293(1) attach 193(0) delay 0]
link [attach 294(1) attach 194(0) delay 0]
link [attach 295(1) attach 195(0) delay 0]
link [attach 296(1) attach 196(0) delay 0]
link [attach 297(1) attach 197(0) delay 0]
link [attach 298(1) attach 198(0) delay 0]
host [idrange [from 100 to 147]
_extends .dictionary.standardServer
]
host [id 148 _extends .dictionary.standardClient]
host [idrange [from 150 to 197]
_extends .dictionary.standardServer1
]
host [id 198 _extends .dictionary.standardClient1]
traffic [
pattern [
client 0:11
servers [nhi_range [from 0:22(0) to 0:24(0)] port 1600]
]
pattern[
client 0:12
servers [nhi 0:21(0) port 1600] ]
pattern[
client 0:12
servers [nhi 0:23(0) port 1600]
]
pattern[
client 0:12
servers [nhi 0:24(0) port 1600]
]
pattern[
client 0:13
servers [nhi 0:21(0) port 1600]
]
pattern[
client 0:13
servers [nhi 0:22(0) port 1600]
]
pattern[
client 0:13
servers [nhi 0:24(0) port 1600]
]
pattern[
client 0:14
servers [nhi_range [from 0:21(0) to 0:23(0)] port 1600]
]
pattern [
client 1:11
servers [nhi_range [from 1:22(0) to 1:24(0)] port 1600]
]
pattern[
client 1:12
servers [nhi 1:21(0) port 1600]
]
pattern[
client 1:12
servers [nhi 1:23(0) port 1600]
]
pattern[
client 1:12
servers [nhi 1:24(0) port 1600]
]
pattern[
client 1:13
servers [nhi 1:21(0) port 1600]
]
pattern[
client 1:13
servers [nhi 1:22(0) port 1600]
```

```
]
pattern[
client 1:13
servers [nhi 1:24(0) port 1600]
]
pattern[
client 1:14
servers [nhi_range [from 1:21(0) to 1:23(0)] port 1600]
]
pattern [
client 2:11
servers [nhi_range [from 2:22(0) to 2:24(0)] port 1600]
]
pattern[
client 2:12
servers [nhi 2:21(0) port 1600]
]
pattern[
client 2:12
servers [nhi 2:23(0) port 1600]
]
pattern[
client 2:12
servers [nhi 2:24(0) port 1600]
]
pattern[
client 2:13
servers [nhi 2:21(0) port 1600]
]
pattern[
client 2:13
servers [nhi 2:22(0) port 1600]
]
pattern[
client 2:13
servers [nhi 2:24(0) port 1600]
]
pattern[
client 2:14
servers [nhi_range [from 2:21(0) to 2:23(0)] port 1600]
]
pattern [
client 3:11
servers [nhi_range [from 3:22(0) to 3:24(0)] port 1600]
]
pattern[
client 3:12
servers [nhi 3:21(0) port 1600]
]
pattern[
client 3:12
servers [nhi 3:23(0) port 1600]
]
pattern[
client 3:12
servers [nhi 3:24(0) port 1600]
]
pattern[
client 3:13
servers [nhi 3:21(0) port 1600]
]
pattern[
client 3:13
servers [nhi 3:22(0) port 1600]
]
pattern[
client 3:13
servers [nhi 3:24(0) port 1600]
]
pattern[
client 3:14
servers [nhi_range [from 3:21(0) to 3:23(0)] port 1600]
]
pattern [
client 0:11
servers [nhi 1:21(0) port 1600]
]
pattern [
client 0:31
servers [nhi 1:41(0) port 1600]
]
pattern [
client 0:11
servers [nhi 2:21(0) port 1600]
]
pattern [
client 0:31
servers [nhi 2:41(0) port 1600]
]
pattern [
client 0:11
servers [nhi 3:21(0) port 1600]
```

```
]
pattern [
client 0:31
servers [nhi 3:41(0) port 1600]
]
pattern [
client 0:12
servers [nhi 1:22(0) port 1600]
]
pattern [
client 0:32
servers [nhi 1:42(0) port 1600]
]
pattern [
client 0:12
servers [nhi 2:22(0) port 1600]
]
pattern [
client 0:32
servers [nhi 2:42(0) port 1600]
]
pattern [
client 0:12
servers [nhi 3:22(0) port 1600]
]
pattern [
client 0:32
servers [nhi 3:42(0) port 1600]
]
pattern [
client 0:13
servers [nhi 1:23(0) port 1600]
]
pattern [
client 0:33
servers [nhi 1:43(0) port 1600]
]
pattern [
client 0:13
servers [nhi 2:23(0) port 1600]
]
pattern [
client 0:33
servers [nhi 2:43(0) port 1600]
]
pattern [
client 0:13
servers [nhi 3:23(0) port 1600]
]
pattern [
client 0:33
servers [nhi 3:43(0) port 1600]
]
pattern [
client 0:14
servers [nhi 1:24(0) port 1600]
]
pattern [
client 0:34
servers [nhi 1:44(0) port 1600]
]
pattern [
client 0:14
servers [nhi 2:24(0) port 1600]
]
pattern [
client 0:34
servers [nhi 2:44(0) port 1600]
]
pattern [
client 0:14
servers [nhi 3:24(0) port 1600]
]
pattern [
client 0:34
servers [nhi 3:44(0) port 1600]
]
pattern [
client 1:11
servers [nhi 0:21(0) port 1600]
]
pattern [
client 1:31
servers [nhi 0:41(0) port 1600]
]
pattern [
client 1:11
servers [nhi 2:21(0) port 1600]
]
pattern [
client 1:31
servers [nhi 2:41(0) port 1600]
```

```
]
pattern [
client 1:11 servers [nhi 3:21(0) port 1600]
]
pattern [
client 1:31
servers [nhi 3:41(0) port 1600]
]
pattern [
client 1:12
servers [nhi 0:22(0) port 1600]
]
pattern [
client 1:32
servers [nhi 0:42(0) port 1600] ]
pattern [
client 1:12
servers [nhi 2:22(0) port 1600]
]
pattern [
client 1:32
servers [nhi 2:42(0) port 1600]
]
pattern [
client 1:12
servers [nhi 3:22(0) port 1600]
]
pattern [
client 1:32
servers [nhi 3:42(0) port 1600]
]
pattern [
client 1:13
servers [nhi 0:23(0) port 1600]
]
pattern [
client 1:33
servers [nhi 0:43(0) port 1600]
]
pattern [
client 1:13
servers [nhi 2:23(0) port 1600]
]
pattern [
client 1:33
servers [nhi 2:43(0) port 1600]
```

```
]
pattern [
client 1:13
servers [nhi 3:23(0) port 1600]
]
pattern [
client 1:33
servers [nhi 3:43(0) port 1600]
]
pattern [
client 1:14
servers [nhi 0:24(0) port 1600]
]
pattern [
client 1:34
servers [nhi 0:44(0) port 1600]
]
pattern [
client 1:14
servers [nhi 2:24(0) port 1600]
]
pattern [
client 1:34
servers [nhi 2:44(0) port 1600]
]
pattern [
client 1:14
servers [nhi 3:24(0) port 1600]
]
pattern [
client 1:34
servers [nhi 3:44(0) port 1600]
]
pattern [
client 2:11
servers [nhi 1:21(0) port 1600]
]
pattern [
client 2:31
servers [nhi 1:41(0) port 1600]
]
pattern [
client 2:11
servers [nhi 0:21(0) port 1600]
]
pattern [
client 2:31
servers [nhi 0:41(0) port 1600]
]
pattern [
client 2:11
servers [nhi 3:21(0) port 1600]
]
pattern [
client 2:31
servers [nhi 3:41(0) port 1600]
]
pattern [
client 2:12
servers [nhi 1:22(0) port 1600]
]
pattern [
client 2:32
servers [nhi 1:42(0) port 1600]
]
pattern [
client 2:12
servers [nhi 0:22(0) port 1600]
]
pattern [
client 2:32
servers [nhi 0:42(0) port 1600]
]
pattern [
client 2:12
servers [nhi 3:22(0) port 1600]
]
pattern [
client 2:32
servers [nhi 3:42(0) port 1600]
]
pattern [
client 2:13
servers [nhi 1:23(0) port 1600]
]
pattern [
client 2:33
servers [nhi 1:43(0) port 1600]
]
pattern [
client 2:13
servers [nhi 0:23(0) port 1600]
```

```
]
pattern [
client 2:33
servers [nhi 0:43(0) port 1600]
]
pattern [
client 2:13
servers [nhi 3:23(0) port 1600]
]
pattern [
client 2:33
servers [nhi 3:43(0) port 1600]
]
pattern [
client 2:14
servers [nhi 1:24(0) port 1600]
]
pattern [
client 2:34
servers [nhi 1:44(0) port 1600]
]
pattern [
client 2:14
servers [nhi 0:24(0) port 1600]
]
pattern [
client 2:34
servers [nhi 0:44(0) port 1600]
]
pattern [
client 2:14
servers [nhi 3:24(0) port 1600]
]
pattern [
client 2:34
servers [nhi 3:44(0) port 1600]
]
pattern [
client 3:11
servers [nhi 1:21(0) port 1600]
]
pattern [
client 3:31
servers [nhi 1:41(0) port 1600]
]
pattern [
client 3:11
servers [nhi 2:21(0) port 1600]
]
pattern [
client 3:31
servers [nhi 2:41(0) port 1600]
]
pattern [
client 3:11
servers [nhi 0:21(0) port 1600]
]
pattern [
client 3:31
servers [nhi 0:41(0) port 1600]
]
pattern [
client 3:12
servers [nhi 1:22(0) port 1600]
]
pattern [
client 3:32
servers [nhi 1:42(0) port 1600]
] pattern [
client 3:12
servers [nhi 2:22(0) port 1600]
]
pattern [
client 3:32
servers [nhi 2:42(0) port 1600]
]
pattern [
client 3:12
servers [nhi 0:22(0) port 1600]
]
pattern [
client 3:32
servers [nhi 0:42(0) port 1600]
]
pattern [
client 3:13
servers [nhi 1:23(0) port 1600]
]
pattern [
client 3:33
servers [nhi 1:43(0) port 1600]
]
```

```
pattern [
client 3:13
servers [nhi 2:23(0) port 1600]
]
pattern [
client 3:33
servers [nhi 2:43(0) port 1600]
]
pattern [
client 3:13
servers [nhi 0:23(0) port 1600]
]
pattern [
client 3:33
servers [nhi 0:43(0) port 1600]
]
pattern [
client 3:14
servers [nhi 1:24(0) port 1600]
]
pattern [
client 3:34
servers [nhi 1:44(0) port 1600]
]
pattern [
client 3:14
servers [nhi 2:24(0) port 1600]
]
pattern [
client 3:34
servers [nhi 2:44(0) port 1600]
]
pattern [
client 3:14
servers [nhi 0:24(0) port 1600]
]
pattern [
client 3:34
servers [nhi 0:44(0) port 1600]
]
pattern [
client 148
servers [nhi_range [from 0:21(0) to 0:24(0)] port 1600]
]
pattern [
client 148
```

```
servers [nhi_range [from 1:21(0) to 1:24(0)] port 1600]
]
pattern [
client 148
servers [nhi_range [from 2:21(0) to 2:24(0)] port 1600]
]
pattern [
client 148
servers [nhi_range [from 3:21(0) to 3:24(0)] port 1600]
]
pattern [
client 198
servers [nhi_range [from 0:21(0) to 0:24(0)] port 1600]
]
pattern [
client 198
servers [nhi_range [from 1:21(0) to 1:24(0)] port 1600]
]
pattern [
client 198
servers [nhi_range [from 2:21(0) to 2:24(0)] port 1600]
]
pattern [
client 198
servers [nhi_range [from 3:21(0) to 3:24(0)] port 1600]
]
pattern [
client 0:11
servers [nhi_range[from 100(0) to 102(0)] port 1600] ]
pattern [
client 0:12
servers [nhi_range[from 103(0) to 105(0)] port 1600] ]
pattern [
client 0:13
servers [nhi_range[from 106(0) to 108(0)] port 1600]
]
pattern [
client 0:14
servers [nhi_range[from 109(0) to 111(0)] port 1600]
]
pattern [
client 1:11
servers [nhi_range[from 112(0) to 114(0)] port 1600] ]
pattern [
client 1:12
servers [nhi_range[from 115(0) to 117(0)] port 1600]
```

```
]
pattern [
client 1:13
servers [nhi_range[from 118(0) to 120(0)] port 1600]
]
pattern [
client 1:14
servers [nhi_range[from 121(0) to 123(0)] port 1600]
]
pattern [
client 2:11
servers [nhi_range[from 124(0) to 126(0)] port 1600]
]
pattern [
client 2:12
servers [nhi_range[from 127(0) to 129(0)] port 1600]
]
pattern [
client 2:13
servers [nhi_range[from 130(0) to 132(0)] port 1600]
]
pattern [
client 2:14
servers [nhi_range[from 133(0) to 135(0)] port 1600]
]
pattern [
client 3:11
servers [nhi_range[from 136(0) to 138(0)] port 1600]
]
pattern [
client 3:12
servers [nhi_range[from 139(0) to 141(0)] port 1600]
]
pattern [
client 3:13
servers [nhi_range[from 142(0) to 144(0)] port 1600] ]
pattern [
client 3:14
servers [nhi_range[from 145(0) to 147(0)] port 1600] ]
pattern [
client 0:31
servers [nhi_range[from 150(0) to 152(0)] port 1600] ]
pattern [
client 0:32
servers [nhi_range[from 153(0) to 155(0)] port 1600]
]
```

```
pattern [
client 0:33
servers [nhi_range[from 156(0) to 158(0)] port 1600]
]
pattern [
client 0:34
servers [nhi_range[from 159(0) to 161(0)] port 1600]
]
pattern [
client 1:31
servers [nhi_range[from 162(0) to 164(0)] port 1600]
]
pattern [
client 1:32
servers [nhi_range[from 165(0) to 167(0)] port 1600]
]
pattern [
client 1:33
servers [nhi_range[from 168(0) to 170(0)] port 1600]
]
pattern [
client 1:34
servers [nhi_range[from 171(0) to 173(0)] port 1600]
]
pattern [
client 2:31
servers [nhi_range[from 174(0) to 176(0)] port 1600]
]
pattern [
client 2:32
servers [nhi_range[from 177(0) to 179(0)] port 1600]
]
pattern [
client 2:33
servers [nhi_range[from 180(0) to 182(0)] port 1600]
]
pattern [
client 2:34
servers [nhi_range[from 183(0) to 185(0)] port 1600]
]
pattern [
client 3:31
servers [nhi_range[from 186(0) to 188(0)] port 1600]
]
pattern [
client 3:32
```

```
servers [nhi_range[from 189(0) to 191(0)] port 1600]
]
pattern [
client 3:33
servers [nhi_range[from 192(0) to 194(0)] port 1600]
]
pattern [
client 3:34
servers [nhi_range[from 195(0) to 197(0)] port 1600]
]
]
]
networks [
network4 [
Net [
router [
idrange [from 1 to 4]
graph [ProtocolSession [name ip use SSF.OS.IP]]
interface [ idrange [from 4 to 13] buffer 8000 _extends .dictionary.15BaseT]
interface [ idrange [from 0 to 3] buffer 16000 _extends .dictionary.10BaseT]
route [dest default interface 4] ]
link [attach 1(4) attach 2(4) delay 0.01]
link [attach 1(5) attach 3(4) delay 0.01]
link [attach 1(6) attach 4(4) delay 0.01]
link [attach 2(5) attach 3(5) delay 0.01]
link [attach 2(6) attach 4(5) delay 0.01]
link [attach 3(6) attach 4(6) delay 0.01]
link [attach 1(0) attach 11(0) delay 0] link [attach 1(1) attach 21(0) delay 0]
link [attach 1(2) attach 31(0) delay 0]
link [attach 1(3) attach 41(0) delay 0]
link [attach 2(0) attach 12(0) delay 0]
link [attach 2(1) attach 22(0) delay 0]
link [attach 2(2) attach 32(0) delay 0]
link [attach 2(3) attach 42(0) delay 0]
link [attach 3(0) attach 13(0) delay 0]
link [attach 3(1) attach 23(0) delay 0]
link [attach 3(2) attach 33(0) delay 0]
link [attach 3(3) attach 43(0) delay 0]
link [attach 4(0) attach 14(0) delay 0]
link [attach 4(1) attach 24(0) delay 0]
link [attach 4(2) attach 34(0) delay 0]
link [attach 4(3) attach 44(0) delay 0]
host [idrange [from 11 to 14]
_extends .dictionary.standardClient
]
host [idrange [from 21 to 24]
```

```
_extends .dictionary.standardServer
]
host [idrange [from 31 to 34]
_extends .dictionary.standardClient1
]
host [idrange [from 41 to 44]
_extends .dictionary.standardServer1
]
]
]
]
dictionary [
10BaseT [
bitrate 10000000
latency 0
]
15BaseT [
bitrate 1500000
latency 0.01
]
udpinit [
max_datagram_size 10000
debug false
]
appsession [
request_size 256
datagram_size 500
send_interval 0.01
show_report true
debug true
]
appsession1 [
request_size 256
datagram_size 500
send_interval 0.00005
show_report true
debug true
]
standardClient [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
start_time 1.0
start_window 1.0
```

```
file_size 3000000
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardClient1 [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name client use SSF.OS.UDP.test.udpStreamClient
start_time 1.0
start_window 1.0
file_size 298500000
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardServer [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name server use SSF.OS.UDP.test.udpStreamServer
port 1600
client_limit 20
_find .dictionary.appsession.request_size
_find .dictionary.appsession.datagram_size
_find .dictionary.appsession.send_interval
_find .dictionary.appsession.show_report
_find .dictionary.appsession.debug
]
```

```
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
]
]
standardServer1 [
interface [id 0 _extends .dictionary.10BaseT]
route [dest default interface 0]
graph [
ProtocolSession [
name server use SSF.OS.UDP.test.udpStreamServer
port 1600
client_limit 20
_find .dictionary.appsession1.request_size
_find .dictionary.appsession1.datagram_size
_find .dictionary.appsession1.send_interval
_find .dictionary.appsession1.show_report
_find .dictionary.appsession1.debug
]
ProtocolSession [name socket use SSF.OS.Socket.socketMaster]
ProtocolSession [name udp use SSF.OS.UDP.udpSessionMaster
_find .dictionary.udpinit]
ProtocolSession [name ip use SSF.OS.IP]
] ]
]
```