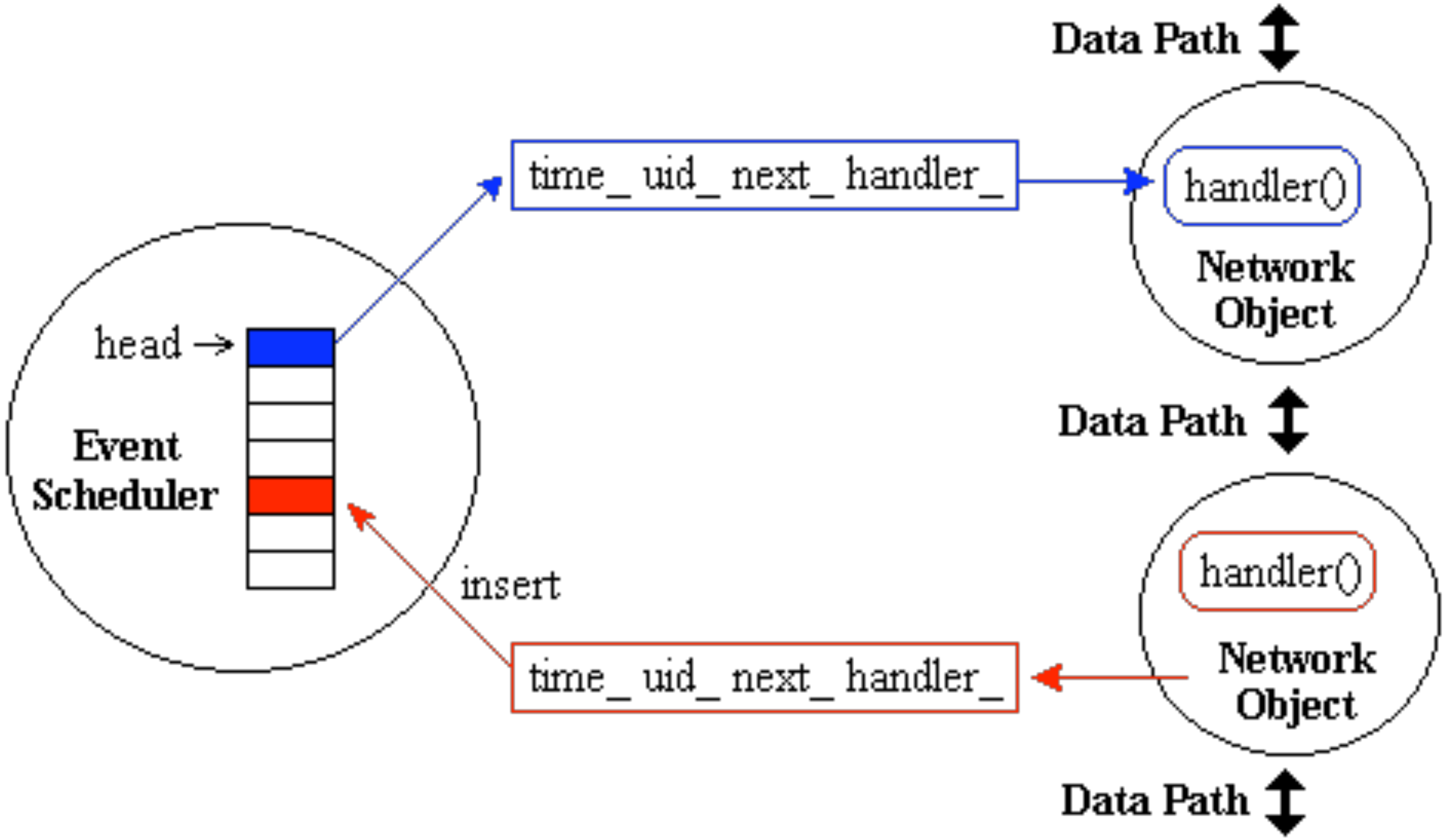
The Network Simulator NS2
(II parte)

casalicchio@ing.uniroma2.it

http://www.ce.uniroma2.it/courses/MMI/

http://www.uniroma2.it/didattica/MMI

Event Scheduler



Scheduler

- List scheduler
- Heap scheduler
- Calendar scheduler (default)

- **All made the same task with different performances**

List Scheduler

- The list scheduler (Scheduler/List../ns-2/scheduler.cc) implements the scheduler using a simple linked-list structure.
- The list is kept in time-order (earliest to latest), so event insertion and deletion require scanning the list to find the appropriate entry.
- Choosing the next event for execution requires trimming the first entry off the head of the list.
  - This implementation preserves event execution in a FIFO manner for simultaneous events.

Heap Scheduler

- The heap scheduler (Scheduler/Heap../ns-2/scheduler.cc) implements the scheduler using a heap structure.
- This structure is superior to the list structure for a large number of events, as insertion and deletion times are in ***O(log n)*** for ***n*** events.

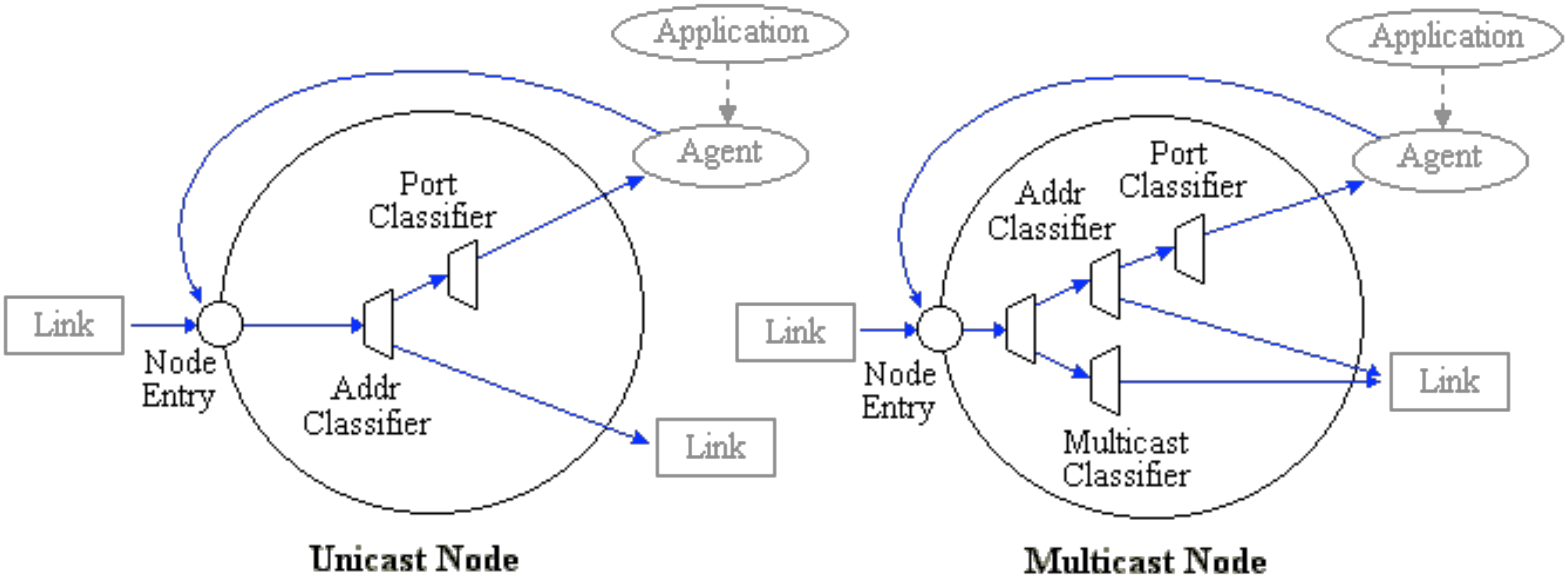
Calendar scheduler

- The calendar queue scheduler (Scheduler/Calendar../ns-2/ scheduler.cc) uses a data structure analogous to a one-year desk calendar, in which events on the same month/day of multiple years can be recorded in one day.
- The implementation of Calendar queues in ns v2 was contributed by David Wetherall (presently at MIT/LCS).
- The calendar queue scheduler since ns v2.33 is improved by the following three algorithms:

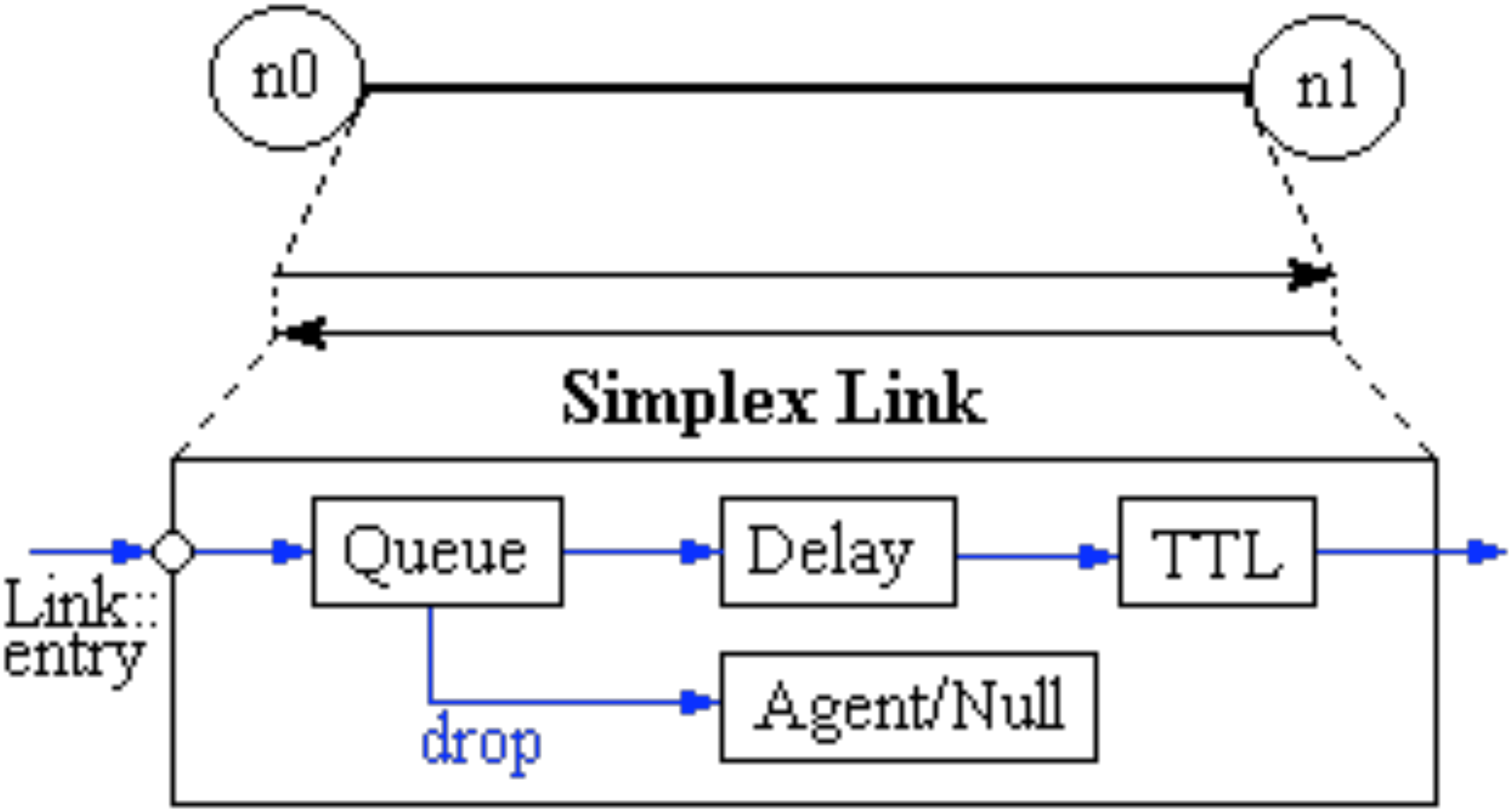
Time Precision

- **Precision of the scheduler clock can be defined as the smallest time-scale of the simulator that can be correctly represented**.
- The **clock variable** for ns is represented by a **double**.
  - As per the IEEE std for floating numbers, a double, consisting of 64 bits must allocate the following bits between its sign, exponent and mantissa fields: sign=1bit exponent =11bit mantissa=52bit
  - Any floating number can be represented in the form $\mathbf{X * 2^n}$ where **X** is the mantissa and **n** is the exponent.
- Thus the precision of timeclock in ns can be defined as $\mathbf{1/(2^{52})}$.

Node and Routing

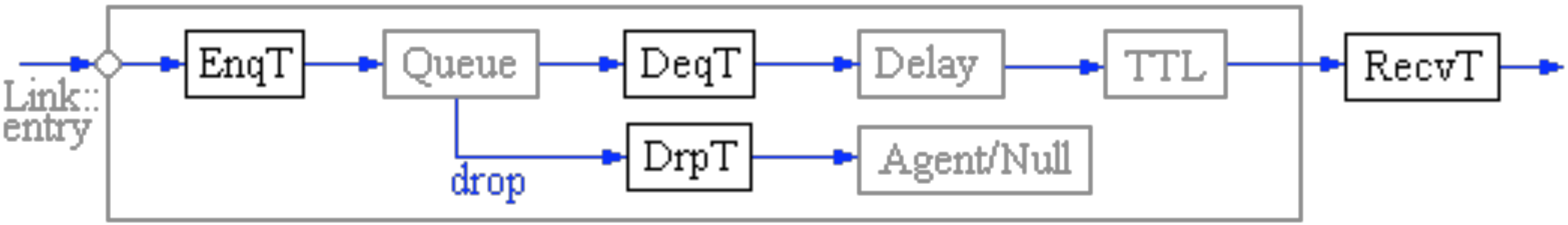


Link

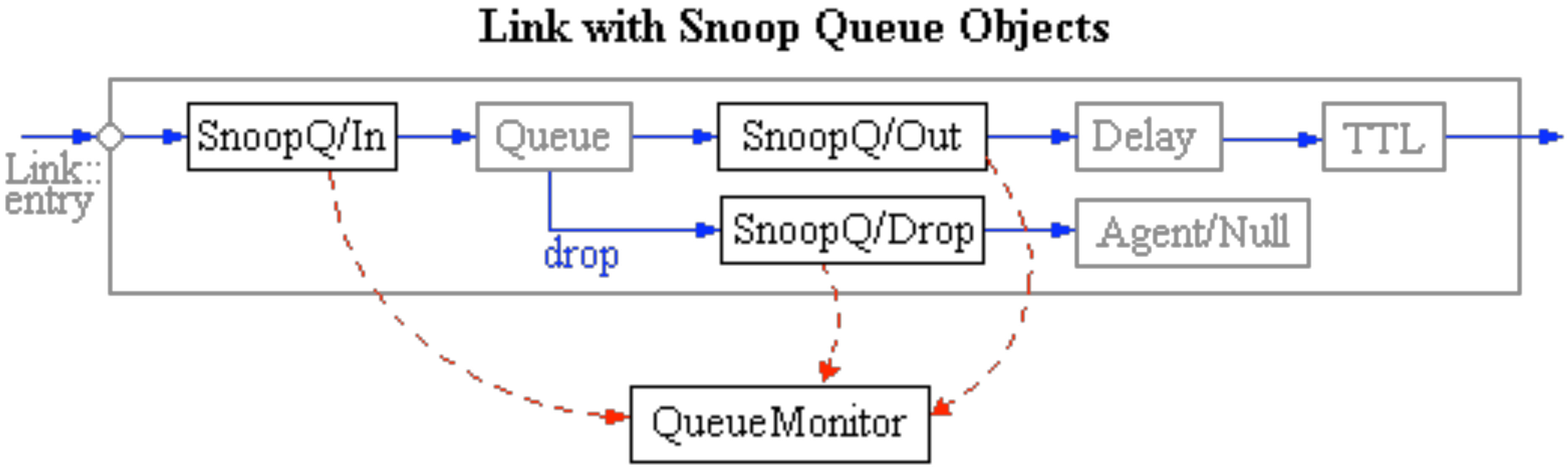

n0
n1
Simplex Link
Queue
Delay
TTL
Link::
entry
drop
Agent/Null

Tracing

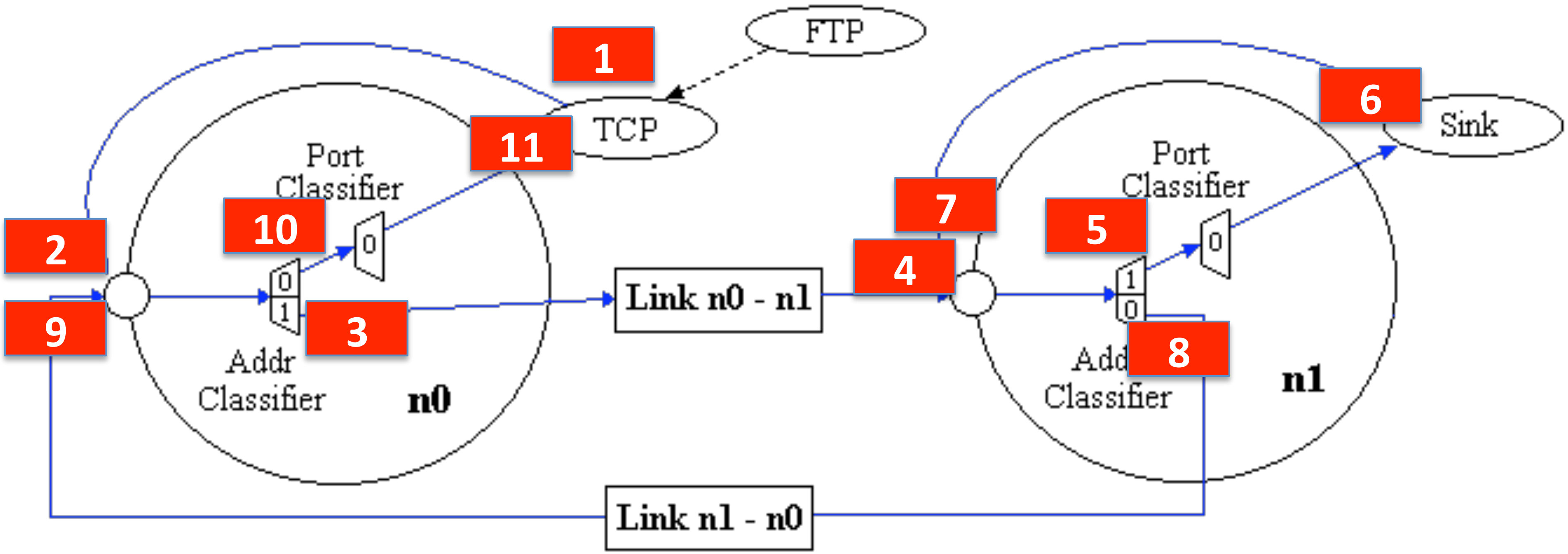
Link with Trace Objects



Queue Monitor



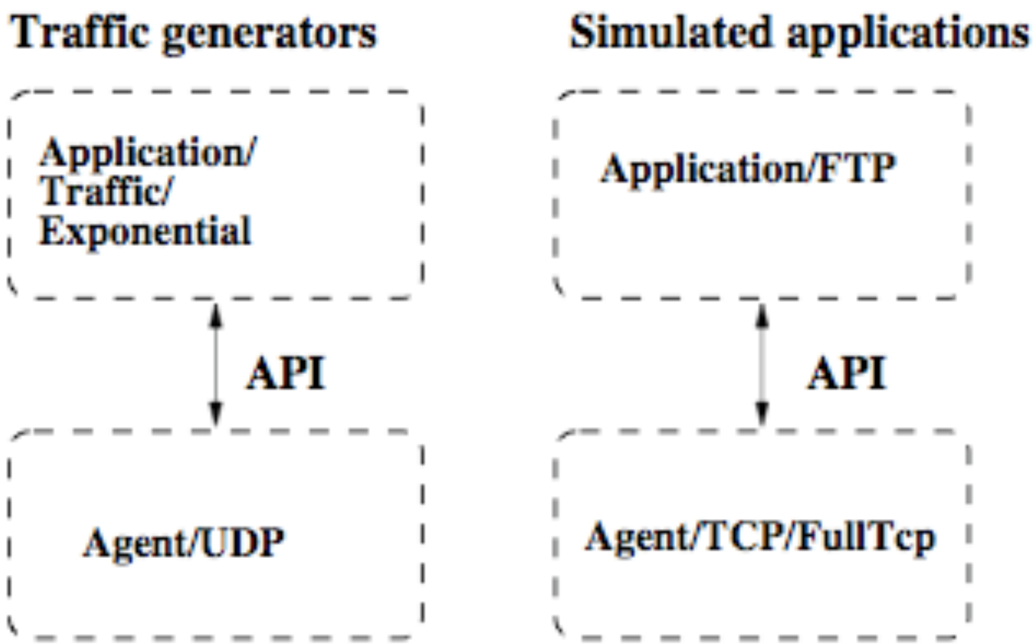
Packet Flow



FTP
1
TCP
11
Port
Classifier
10
0
2
0
1
9
3
Addr
Classifier
n0
Link n0 - n1
7
4
Port
Classifier
5
0
6
Sink
1
0
8
Addr
Classifier
n1
Link n1 - n0

Application

- App. to Agent: system call
  - send
  - sendmsg
  - close
  - listen
- Agent to App.: up call
  - recv
  - resume

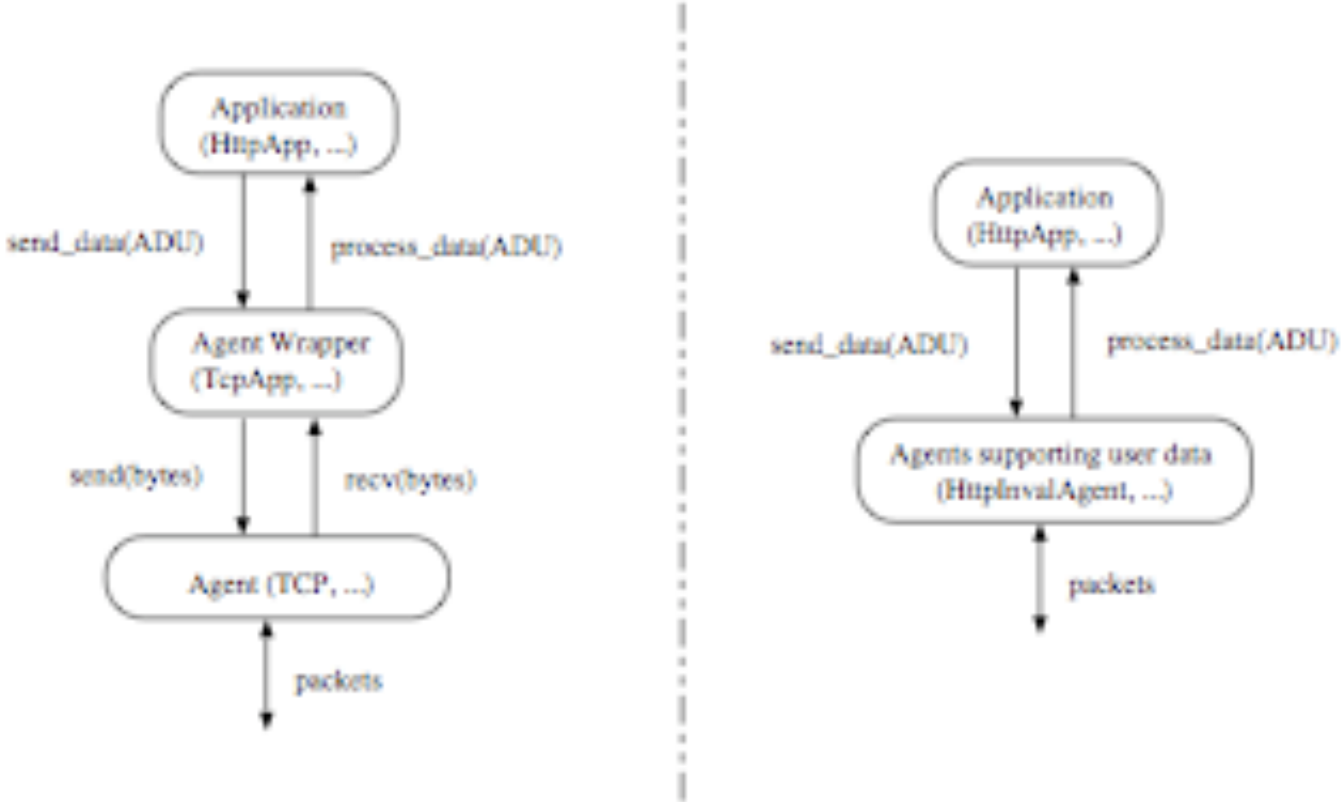


- **in *ns* there is no actual data being passed between applications**

Type of application

- HttpApp
- Application/TcpApp
- Application/FTP
- Application/Telnet
- Application/Traffic/*
  - Exponential On/Off
  - Pareto On/Off
  - CBR
  - Traffic trace

Web traffic



Application
(HttpApp, ...)
send_data(ADU)
process_data(ADU)
Agent Wrapper
(TcpApp, ...)
send(bytes)
recv(bytes)
Agent (TCP, ...)
packets
Application
(HttpApp, ...)
send_data(ADU)
process_data(ADU)
Agents supporting user data
(HttpInvalAgent, ...)
packets

Web page and PagePool

- Web Page
  - name
  - modification time
  - age
- Page Pool
  - Math
  - CompMath
  - ProxyTrace
  - Client

Esempio

- simple-webcache.tcl

Internet topologies

- To effectively engineer the Internet, crucial issues such as the large scale structure of its underlying physical topology,
  - its time evolution and the contribution of its individual components to its overall function need to be well understood.
- During the **design phase of an Internet-based technology**, extensive simulations are usually performed to assess its feasibility, in terms of efficiency and performance.
- In general, **Internet studies and simulations assume certain topological properties or use synthetically generated topologies.**
- If such studies are to give accurate guidance as to Internet-wide behavior of the protocols and algorithms being studied, **the chosen topologies must exhibit fundamental properties or invariants empirically found in the actual extant structure of the Internet.** Otherwise, correct conclusions cannot be drawn.

BRITE topology generator

- BRITE: Boston university Representative Internet Topology gEnerator
- http://www.cs.bu.edu/brite/

- Router topology model
- AS topology model
- Hierarchical
  - Top-down topology model
  - Bottom-up topology model

Router Waxman

- Basically refers to a generation model for a random topology using Waxman's probability model for interconnecting the nodes of the topology, which is given by:

$$P(u,v) = a * e^{-d/(b*L)}$$

where ***0 < a,b <= 1***, d is the Euclidean distance from node ***u*** to node ***v***, and ***L*** is the maximum distance between any two nodes.

Route Barabasi-Albert topology

- RouterBarabasiAlbert model that implements a model proposed by Barabási and Albert
- This model suggests two possible causes for the emergence of a power law in the frequency of outdegrees in network topologies: incremental growth and preferential connectivity.
  - **Incremental growth** refers to growing networks that are formed by the continual addition of new nodes, and thus the gradual increase in the size of the network.
  - **Preferential connectivity** refers to the tendency of a new node to connect to existing nodes that are highly connected or popular.
- RouterBarabasiAlbert interconnects the nodes according to the incremental growth approach. When a node ***i*** joins the network, the probability that it connects to a node ***j*** already belonging to the network is given by:

$$P(i,j) = d_i / S$$

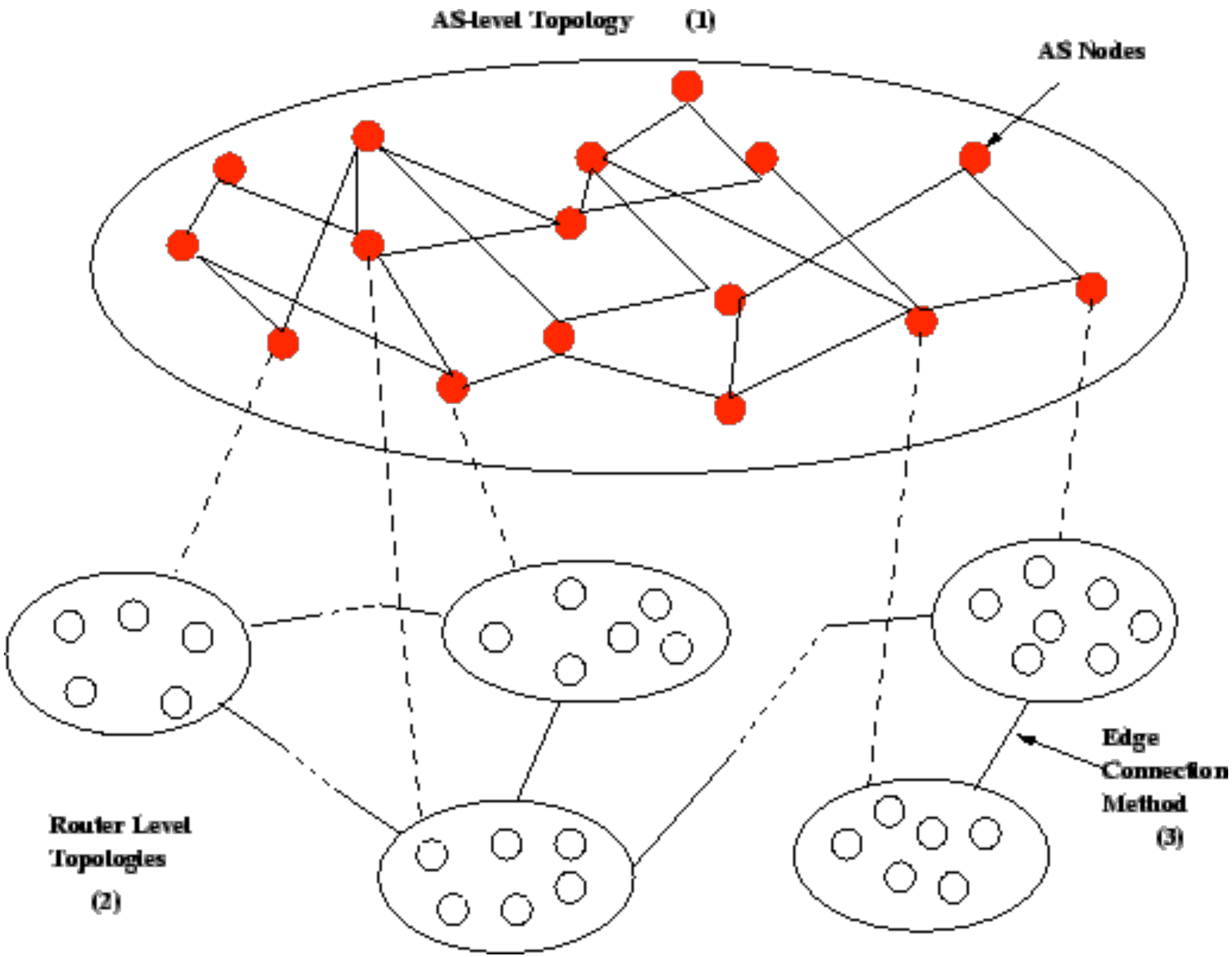
where d_i is the degree of the target node and S is the sum of outdegrees of all nodes that previously joined the network.

AS-level topology

- The provided AS-level models are very similar to the models provided for generating router-level topologies.
- The main difference between these router-level and AS-level models is the fact that AS models place AS nodes in the plane and these have the capability of containing associated topologies.
- Note that this does not mean that there are no AS-level and router-level models that differ substantially from each other. The idea of separating router-level from AS-level from the beginning is to allow for the flexibility of developing independent models for each scenario.
- The two AS-level models provided with the initial distribution of BRITE are **Waxman** and **BarabasiAlbert**

Top-down topology

- Top-down means that BRITE generates first an AS-level topology (1) according to one of the available flat AS-level models (e.g. Waxman, Imported File, etc.).
- Next, for each node in the AS-level topology BRITE will generate a router-level topology (2) using a different generation model from the available flat models that can be used at the router-level.
- (3) BRITE uses an edge connection mechanism to interconnect router-level topologies as dictated by the connectivity of the AS-level topology.
  - Performing this interconnection of router-level topologies in a representative way is an open research question.


AS-level Topology (1)
AS Nodes
Router Level
Topologies
(2)
Edge
Connection
Method
(3)