Rechnernetze-Praktikum WS 2003/04

Moritz Steiner
Daniel Förderer
Sascha Schnaufer

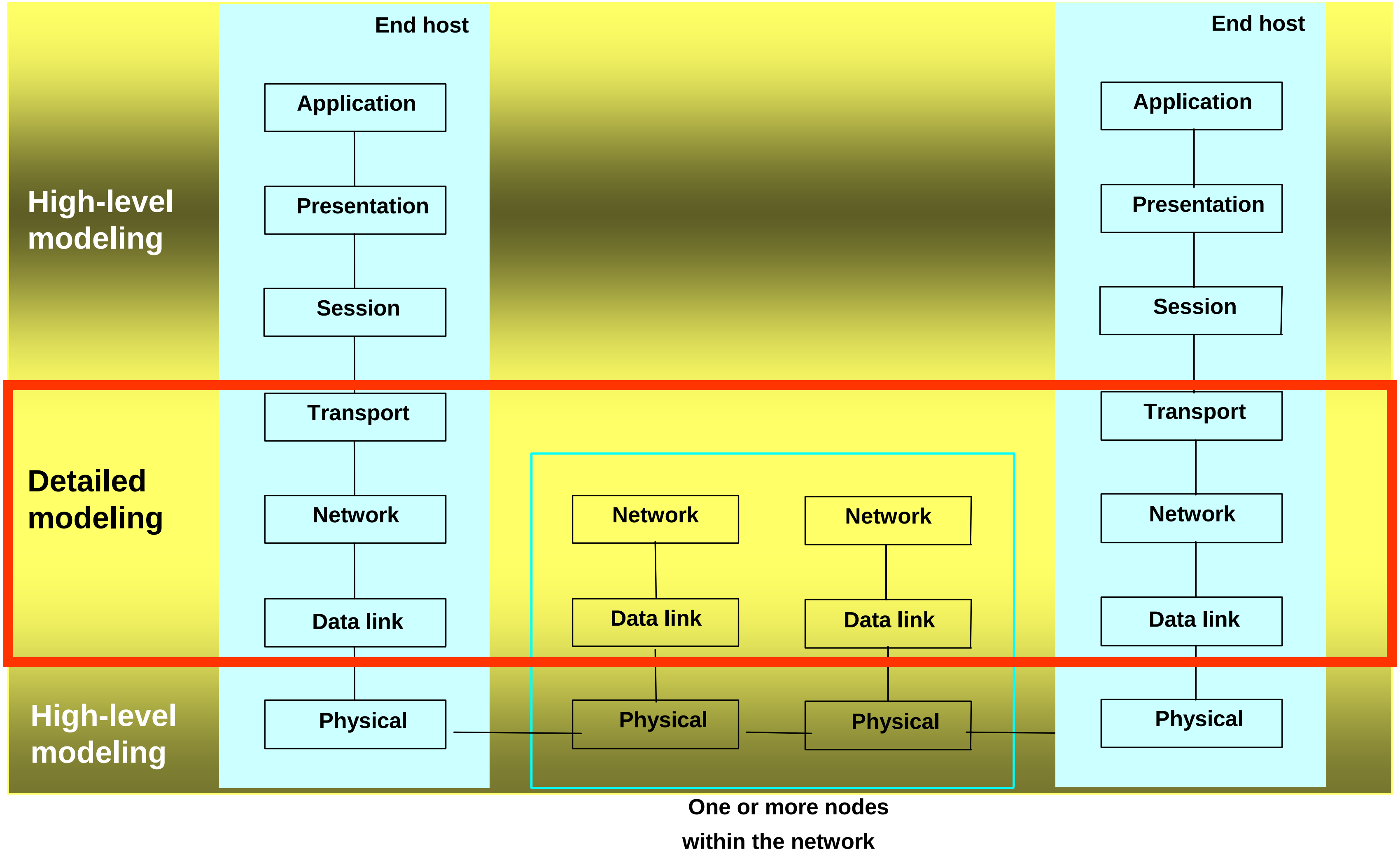
Lehrstuhl für Praktische Informatik IV

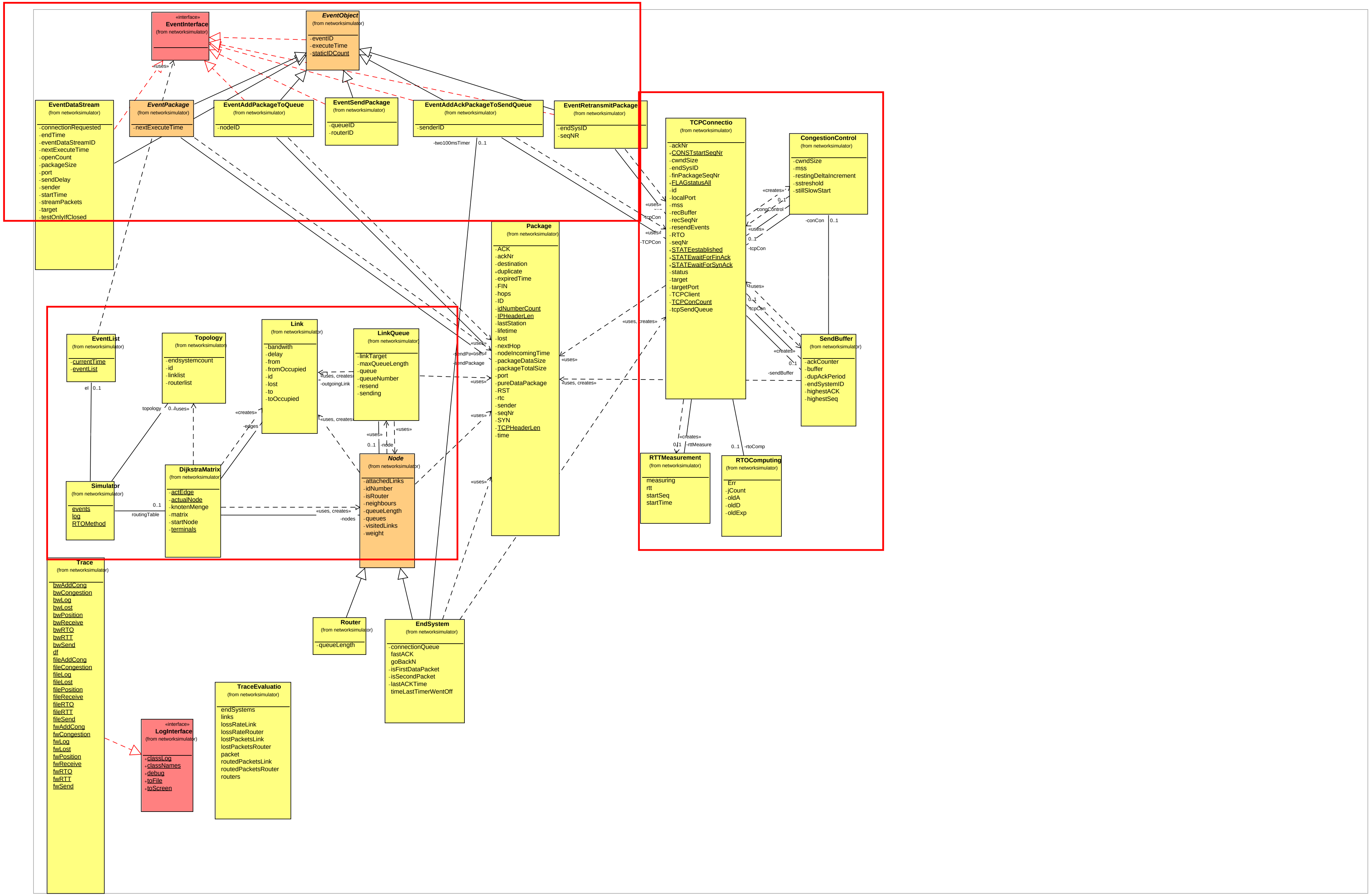
Universität Mannheim

Aufgabenstellung

1) Implementierung eines einfachen Netzwerksimulators
    - Knoten (Endsysteme, Router) und Links
        - Beliebige Topologien ⇨ aus Datei einlesen
    - Paketvermittlung
        - Zentrales Routing nach Dijkstra
    - ereignisgesteuerter Ablauf
2) Simulation des TCP – Protokolls
    - Verbindungsauf/abbau
    - Paketverluste
    - Congestion Control
3) Auswertung der Simulationsergebnisse
    - Szenarien und Parameter definieren
    - Simulationsablauf mitprotokollieren
    - graphische Auswertung

III Simulation of computer networks: level of detail

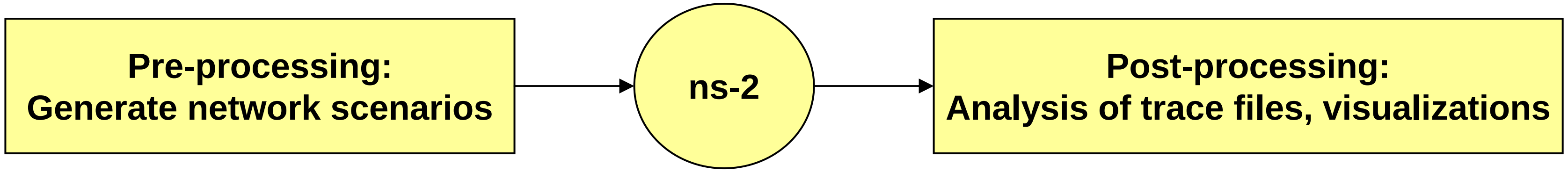



«interface»
EventInterface
(from networksimulator)
EventObject
(from networksimulator)
-eventID
-executeTime
-staticIDCount
«uses»
EventDataStream
(from networksimulator)
-connectionRequested
-endTime
-eventDataStreamID
-nextExecuteTime
-openCount
-packageSize
-port
-sendDelay
-sender
-startTime
-streamPackets
-target
-testOnlyIfClosed
EventPackage
(from networksimulator)
-nextExecuteTime
EventAddPackageToQueue
(from networksimulator)
-nodeID
EventSendPackage
(from networksimulator)
-queueID
-routerID
EventAddAckPackageToSendQueue
(from networksimulator)
-senderID
-two100msTimer
0..1
EventRetransmitPackage
(from networksimulator)
-endSysID
-seqNR
TCPConnectio
(from networksimulator)
-ackNr
+CONSTstartSeqNr
-cwndSize
-endSysID
-finPackageSeqNr
+FLAGstatusAll
-id
-localPort
-mss
-recBuffer
-recSeqNr
-resendEvents
-RTO
-seqNr
+STATEestablished
+STATEwaitForFinAck
+STATEwaitForSynAck
-status
-target
-targetPort
-TCPClient
-TCPConCount
-tcpSendQueue
CongestionControl
(from networksimulator)
-cwndSize
-mss
-restingDeltaIncrement
-sstreshold
-stillSlowStart
«creates»
0..1
-congControl
«uses»
0..1
-tcpCon
-conCon
0..1
«uses»
-tcpCon
«uses»
-TCPCon
Package
(from networksimulator)
-ACK
-ackNr
-destination
-duplicate
-expiredTime
-FIN
-hops
-ID
-idNumberCount
-IPHeaderLen
-lastStation
-lifetime
-lost
-nextHop
-nodeIncomingTime
-packageDataSize
-packageTotalSize
-port
-pureDataPackage
-RST
-rtc
-sender
-seqNr
-SYN
-TCPHeaderLen
-time
«uses»
«uses, creates»
«uses, creates»
«uses»
-sendPa«uses»
-sendPackage
«uses»
«uses»
«uses»
«uses»
0..1
-tcpCon
«creates»
0..1
-sendBuffer
SendBuffer
(from networksimulator)
-ackCounter
-buffer
-dupAckPeriod
-endSystemID
-highestACK
-highestSeq
«creates»
0..1
-rttMeasure
0..1
-rtoComp
RTTMeasurement
(from networksimulator)
measuring
rtt
startSeq
startTime
RTOComputing
(from networksimulator)
Err
-jCount
-oldA
-oldD
-oldExp
EventList
(from networksimulator)
-currentTime
-eventList
el
0..1
Topology
(from networksimulator)
-endsystemcount
-id
-linklist
-routerlist
topology
0..1
«uses»
Link
(from networksimulator)
-bandwith
-delay
-from
-fromOccupied
-id
-lost
-to
-toOccupied
«uses, creates»
-outgoingLink
«uses, creates»
«creates»
-edges
LinkQueue
(from networksimulator)
-linkTarget
-maxQueueLength
-queue
-queueNumber
-resend
-sending
«uses»
«uses»
0..1
-node
Simulator
(from networksimulator)
events
log
RTOMethod
0..1
routingTable
DijkstraMatrix
(from networksimulator)
-actEdge
-actualNode
-knotenMenge
-matrix
-startNode
-terminals
«uses, creates»
-nodes
Node
(from networksimulator)
-attachedLinks
-idNumber
-isRouter
-neighbours
-queueLength
-queues
-visitedLinks
-weight
Trace
(from networksimulator)
bwAddCong
bwCongestion
bwLog
bwLost
bwPosition
bwReceive
bwRTO
bwRTT
bwSend
df
fileAddCong
fileCongestion
fileLog
fileLost
filePosition
fileReceive
fileRTO
fileRTT
fileSend
fwAddCong
fwCongestion
fwLog
fwLost
fwPosition
fwReceive
fwRTO
fwRTT
fwSend
Router
(from networksimulator)
-queueLength
EndSystem
(from networksimulator)
-connectionQueue
fastACK
goBackN
-isFirstDataPacket
-isSecondPacket
-lastACKTime
timeLastTimerWentOff
TraceEvaluatio
(from networksimulator)
endSystems
links
lossRateLink
lossRateRouter
lostPacketsLink
lostPacketsRouter
packet
routedPacketsLink
routedPacketsRouter
routers
«interface»
LogInterface
(from networksimulator)
+classLog
+classNames
+debug
+toFile
+toScreen

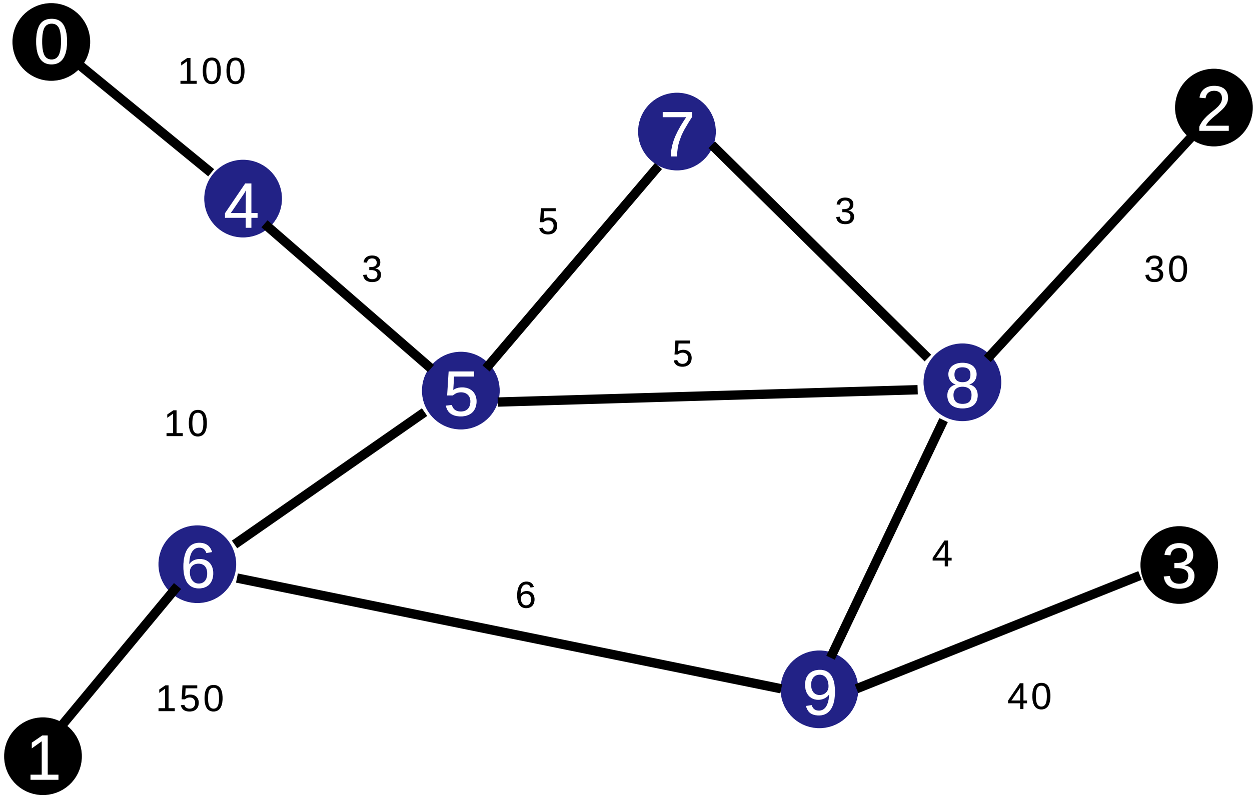
I Elements of ns 'package'

[Source: Ns Tutorial 2002, Padmaparna Haldar]

- Ns, the simulator itself
- Nam, the network animator
  - Visualize ns (or other) output
  - Nam editor: GUI interface to generate ns scripts
- Pre-processing:
  - Traffic and topology generators
- Post-processing:
  - Simple trace analysis, often in Awk, Perl, or Tcl


Pre-processing:
Generate network scenarios
ns-2
Post-processing:
Analysis of trace files, visualizations

Beispiel-Netz



network02.ns

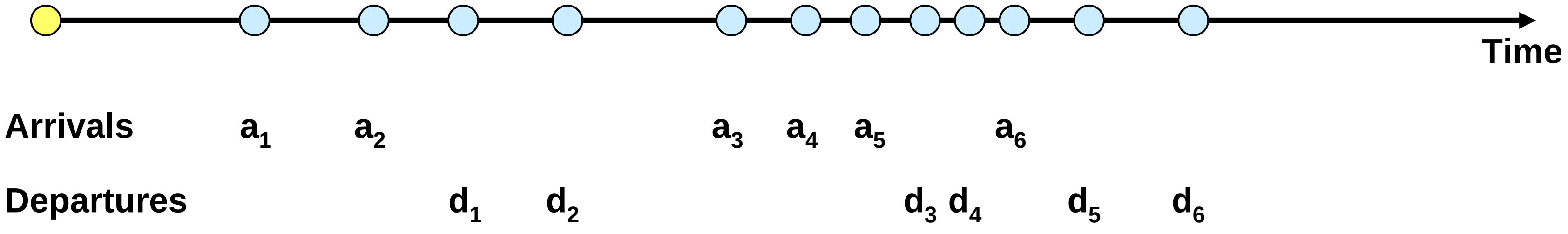
```
// hints:
// - indices start with 0,
// - nodes: first end-systems then routers
// - all links are symmetric
// format:
// <number of endsystems> <number of routers> <number of links>
// <queue length of router 0 in packets>
// <queue length of router 1 in packets>
// ...
// <from node> <to node> <delay in ms> <bandwidth in bytes/s> <packet loss probability in %>
// <from node> <to node> <delay in ms> <bandwidth in bytes/s> <packet loss probability in %>
// ...
4 6 11
9
9
5
10
10
10
0 4 100 56000 100
1 6 150 128000 0
2 8 30 28800 0
3 9 40 64000 0
4 5 3 10000000 0
5 6 10 10000000 0
5 7 5 20000000 0
5 8 5 10000000 0
6 9 6 10000000 0
7 8 3 1000000 0
8 9 4 10000000 0
```

Time Management

Zeit-Steuerung

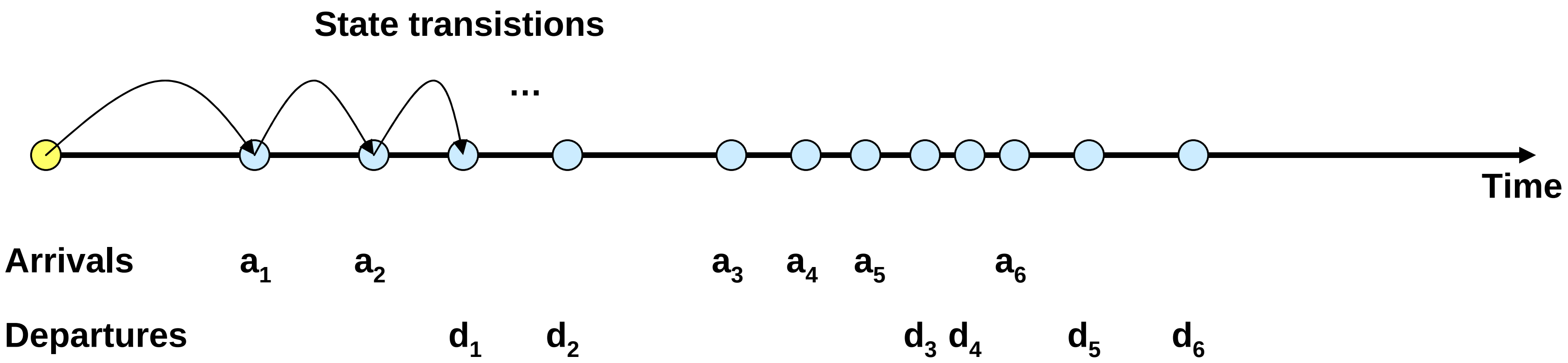
- virtuelle Zeit
- class EventList
  - verwaltet aktuelle Zeit
  - Liste aufsteigend sortierter TimerObjects
  - fortlaufende Bearbeitung des jeweils nächsten TimerObjects
- class EventObject
  - Informationen zur Timer-Verarbeitung
  - verschiedene Typen
- interface ExecuteEvent
  - Callback-Interface

II Introductory example: next-event time advance



- **Events:**
  - Packet arrivals
  - Departure: depends on arrival, delay, and service time
- **Next-event time advance mechanism:**
  - Simulation clock advances to next event
    - State of system is updated
    - Knowledge of the times of occurrence of future events is updated
    - Go to next event
  - Thus, periods of inactivity are ‚skipped‘.

II Introductory example: event logic



State transitions

...

Time

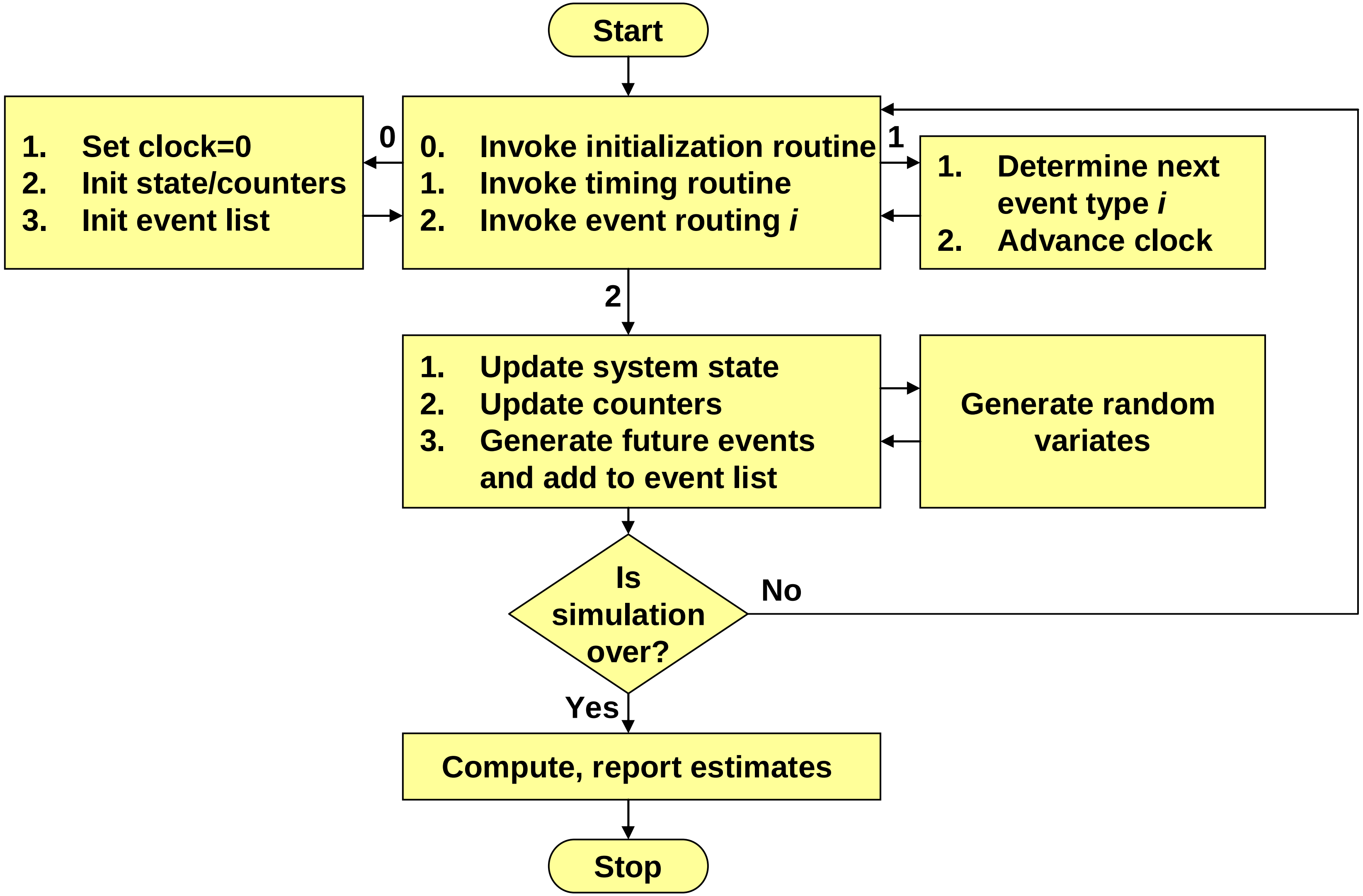
Arrivals a_1 a_2 a_3 a_4 a_5 a_6

Departures d_1 d_2 d_3 d_4 d_5 d_6

- Depending on the *event type,* a specific *event handler* is called that performs the appropriate state transition
- A state transition also includes generation of new events

```
// format:
// <start time in ms> <end time in ms> <source> <destination> <port> <packet size in bytes> <send rate in ms>
0 50000 0 2 21 1000 70
25000 75000 1 2 22 1000 140
//40000 10000 3 2 23 1000 60
```

I Discrete event simulation: flow diagram



Start

1. Set clock=0
2. Init state/counters
3. Init event list

0. Invoke initialization routine
1. Invoke timing routine
2. Invoke event routing *i*

1. Determine next event type *i*
2. Advance clock

1. Update system state
2. Update counters
3. Generate future events and add to event list

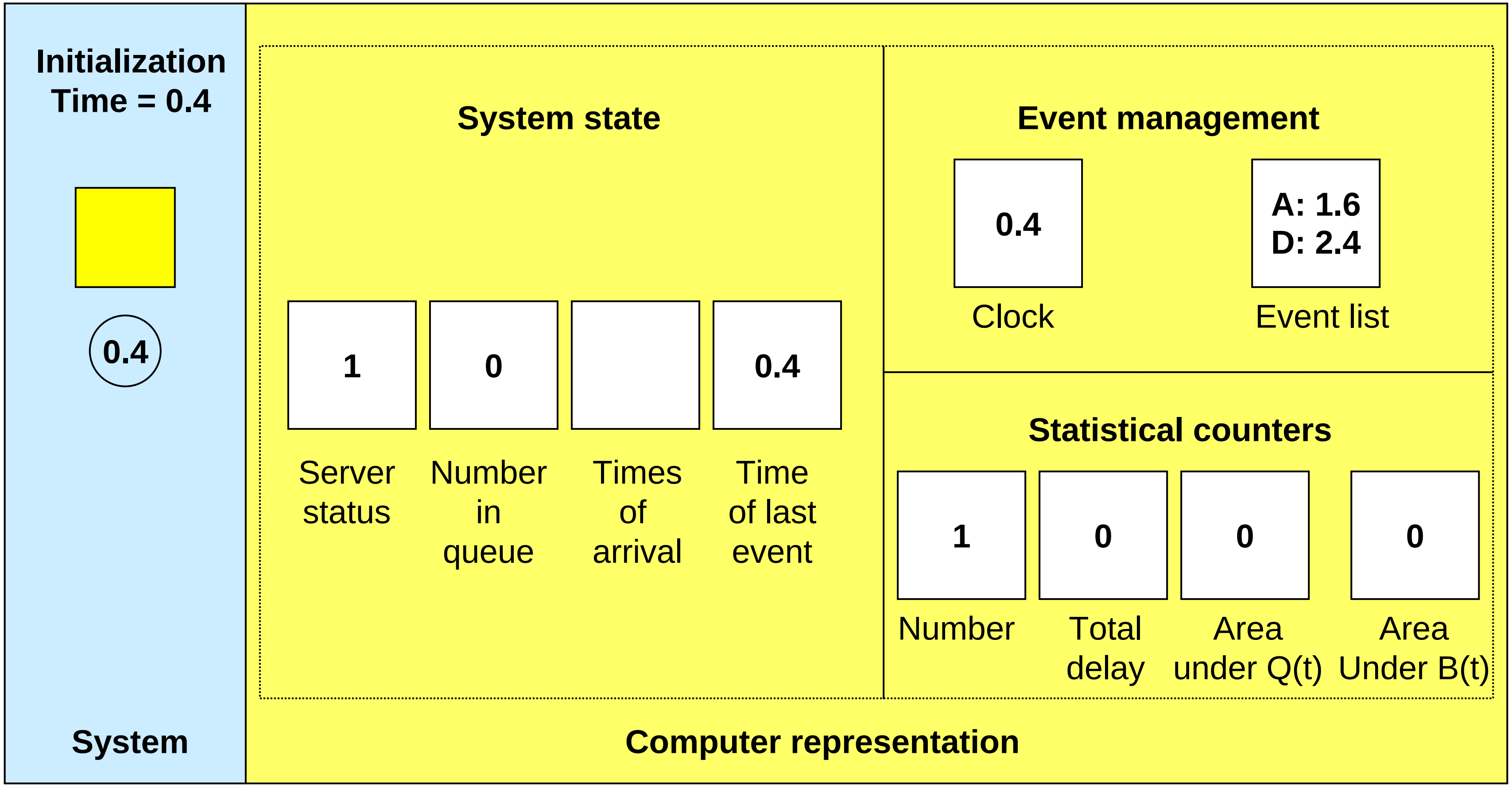
Generate random variates

Is simulation over?

Compute, report estimates

Stop

II Introductory example: execute model



Initialization
Time = 0.4

0.4

System

System state

1
Server status

0
Number in queue

Times of arrival

0.4
Time of last event

Event management

0.4
Clock

A: 1.6
D: 2.4
Event list

Statistical counters

1
Number

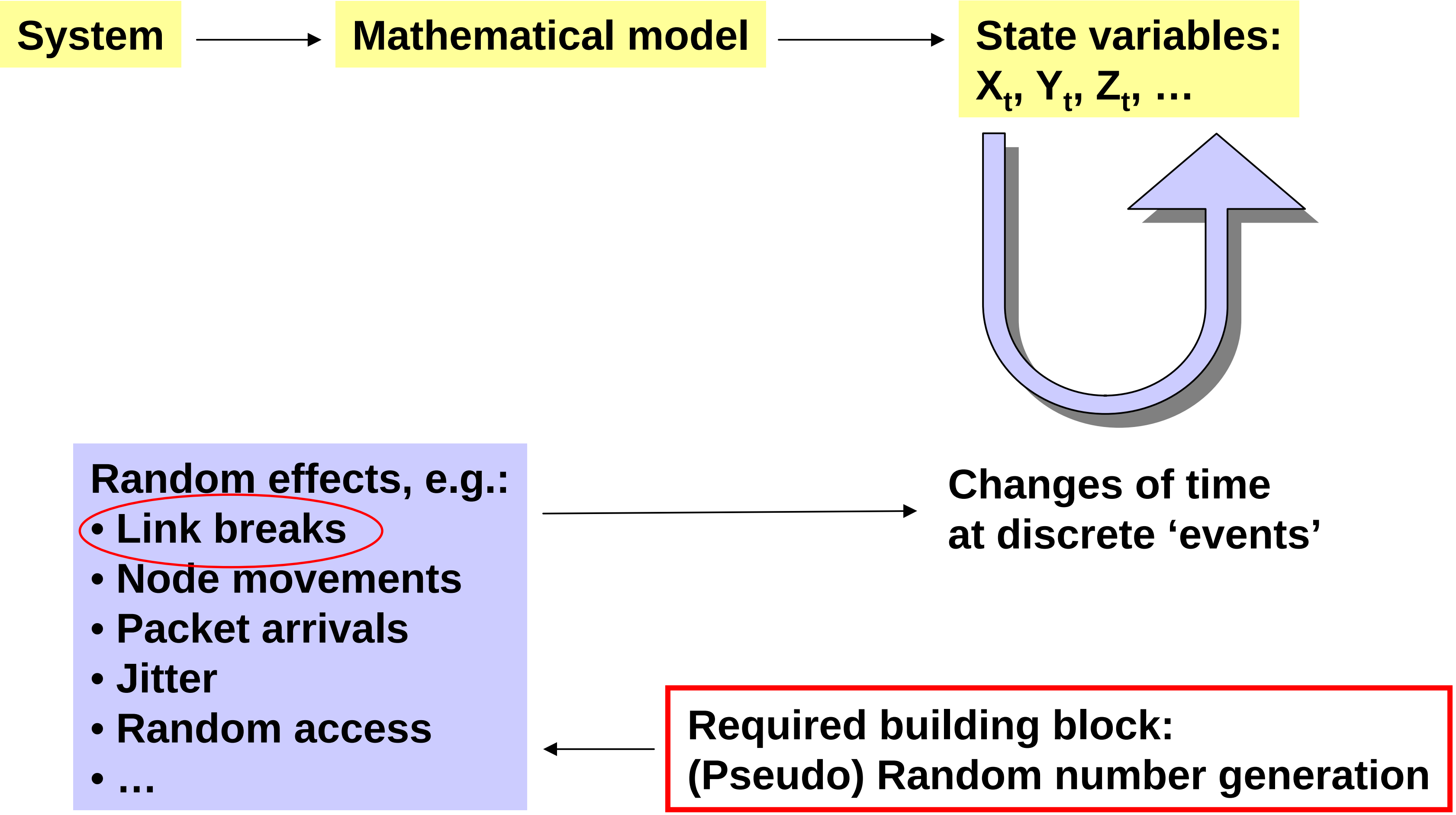
0
Total delay

0
Area under Q(t)

0
Area Under B(t)

Computer representation

I Why do we need ‘random numbers’?



System

Mathematical model

State variables:
X_t, Y_t, Z_t, …

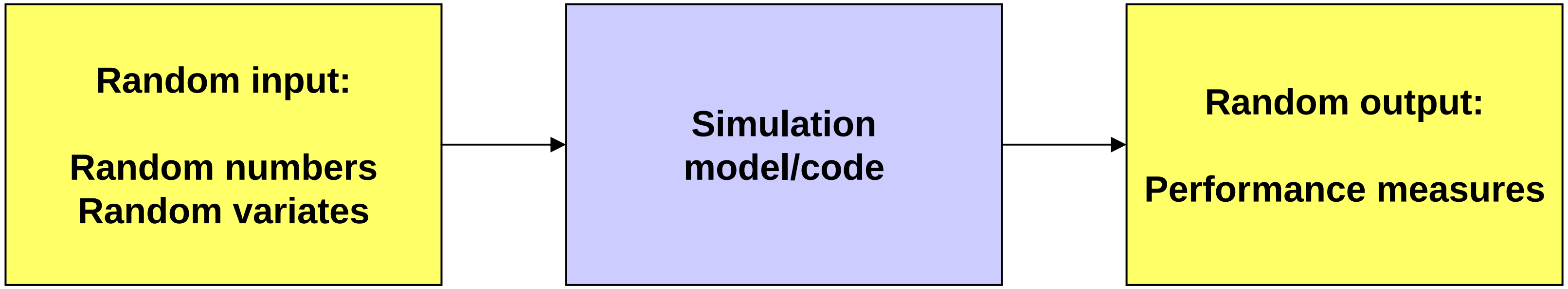
Random effects, e.g.:

- Link breaks
- Node movements
- Packet arrivals
- Jitter
- Random access
- …

Changes of time
at discrete ‘events’

Required building block:
(Pseudo) Random number generation

A.I Problem statement cont‘d



Random input:

Random numbers
Random variates

Simulation
model/code

Random output:

Performance measures

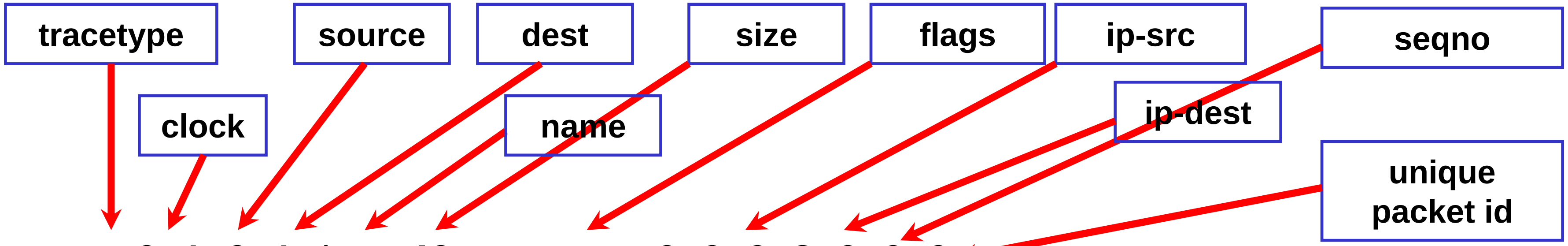
- Do not get ‚exact‘ answers
- Two different runs of the same model: different numerical results

A.I General set-up

$$\text{n Experiments}\left\{\overbrace{\underbrace{\begin{array}{cc} Y_{11} & \dots \\ Y_{21} & \dots \\ \dots & \dots \\ Y_{n1} & \dots \end{array}}_{\text{Warm-up period}}\begin{array}{ccc} Y_{1i} & \dots & Y_{1m} \\ Y_{2i} & \dots & Y_{2m} \\ \dots & \dots & \dots \\ Y_{ni} & \dots & Y_{nm} \end{array}}^{\text{m Observations}}\right.$$

Challenge:
Choose n, m and warm-up period approriately!

II The packet tracefile (wired)



tracetype

clock

source

dest

name

size

flags

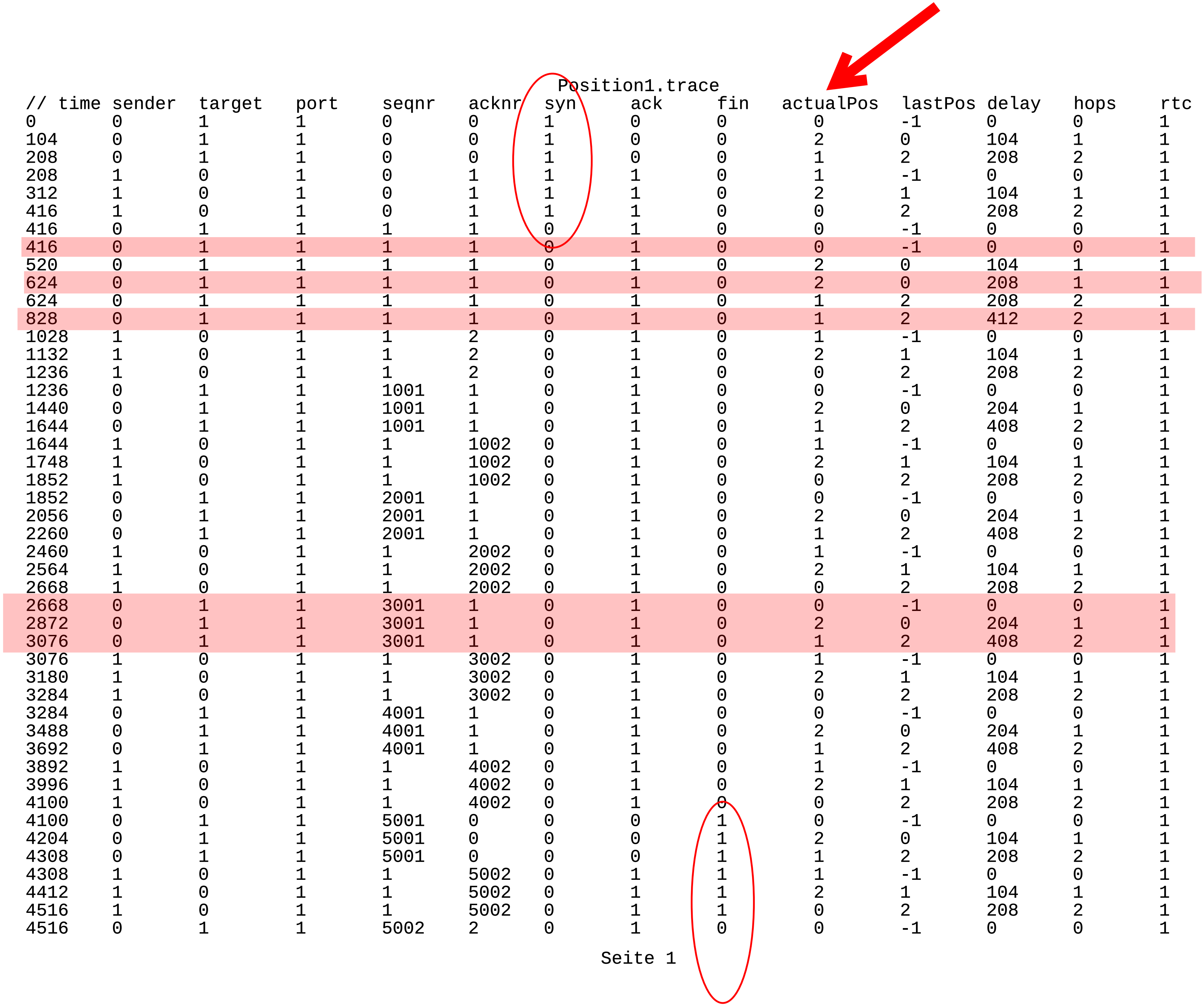
ip-src

ip-dest

seqno

unique packet id

```
+ 0.1 0 1 tcp 40 ------- 0 0.0 3.0 0 0
- 0.1 0 1 tcp 40 ------- 0 0.0 3.0 0 0
r 0.120064 0 1 tcp 40 ------- 0 0.0 3.0 0 0
+ 0.120064 1 2 tcp 40 ------- 0 0.0 3.0 0 0
- 0.120064 1 2 tcp 40 ------- 0 0.0 3.0 0 0
r 0.220704 1 2 tcp 40 ------- 0 0.0 3.0 0 0
+ 0.220704 2 3 tcp 40 ------- 0 0.0 3.0 0 0
- 0.220704 2 3 tcp 40 ------- 0 0.0 3.0 0 0
r 0.240768 2 3 tcp 40 ------- 0 0.0 3.0 0 0
...
d 1.371968 1 2 tcp 1040 ------- 0 0.0 3.0 23 38
```



Position1.trace

```
// time sender  target   port    seqnr   acknr  syn     ack     fin   actualPos lastPos delay   hops    rtc
0       0       1        1       0       0      1       0       0        0       -1      0       0       1
104     0       1        1       0       0      1       0       0        2       0       104     1       1
208     0       1        1       0       0      1       0       0        1       2       208     2       1
208     1       0        1       0       1      1       1       0        1       -1      0       0       1
312     1       0        1       0       1      1       1       0        2       1       104     1       1
416     1       0        1       0       1      1       1       0        0       2       208     2       1
416     0       1        1       1       1      0       1       0        0       -1      0       0       1
416     0       1        1       1       1      0       1       0        0       -1      0       0       1
520     0       1        1       1       1      0       1       0        2       0       104     1       1
624     0       1        1       1       1      0       1       0        2       0       208     1       1
624     0       1        1       1       1      0       1       0        1       2       208     2       1
828     0       1        1       1       1      0       1       0        1       2       412     2       1
1028    1       0        1       1       2      0       1       0        1       -1      0       0       1
1132    1       0        1       1       2      0       1       0        2       1       104     1       1
1236    1       0        1       1       2      0       1       0        0       2       208     2       1
1236    0       1        1       1001    1      0       1       0        0       -1      0       0       1
1440    0       1        1       1001    1      0       1       0        2       0       204     1       1
1644    0       1        1       1001    1      0       1       0        1       2       408     2       1
1644    1       0        1       1       1002   0       1       0        1       -1      0       0       1
1748    1       0        1       1       1002   0       1       0        2       1       104     1       1
1852    1       0        1       1       1002   0       1       0        0       2       208     2       1
1852    0       1        1       2001    1      0       1       0        0       -1      0       0       1
2056    0       1        1       2001    1      0       1       0        2       0       204     1       1
2260    0       1        1       2001    1      0       1       0        1       2       408     2       1
2460    1       0        1       1       2002   0       1       0        1       -1      0       0       1
2564    1       0        1       1       2002   0       1       0        2       1       104     1       1
2668    1       0        1       1       2002   0       1       0        0       2       208     2       1
2668    0       1        1       3001    1      0       1       0        0       -1      0       0       1
2872    0       1        1       3001    1      0       1       0        2       0       204     1       1
3076    0       1        1       3001    1      0       1       0        1       2       408     2       1
3076    1       0        1       1       3002   0       1       0        1       -1      0       0       1
3180    1       0        1       1       3002   0       1       0        2       1       104     1       1
3284    1       0        1       1       3002   0       1       0        0       2       208     2       1
3284    0       1        1       4001    1      0       1       0        0       -1      0       0       1
3488    0       1        1       4001    1      0       1       0        2       0       204     1       1
3692    0       1        1       4001    1      0       1       0        1       2       408     2       1
3892    1       0        1       1       4002   0       1       0        1       -1      0       0       1
3996    1       0        1       1       4002   0       1       0        2       1       104     1       1
4100    1       0        1       1       4002   0       1       0        0       2       208     2       1
4100    0       1        1       5001    0      0       0       1        0       -1      0       0       1
4204    0       1        1       5001    0      0       0       1        2       0       104     1       1
4308    0       1        1       5001    0      0       0       1        1       2       208     2       1
4308    1       0        1       1       5002   0       1       1        1       -1      0       0       1
4412    1       0        1       1       5002   0       1       1        2       1       104     1       1
4516    1       0        1       1       5002   0       1       1        0       2       208     2       1
4516    0       1        1       5002    2      0       1       0        0       -1      0       0       1
```

Seite 1

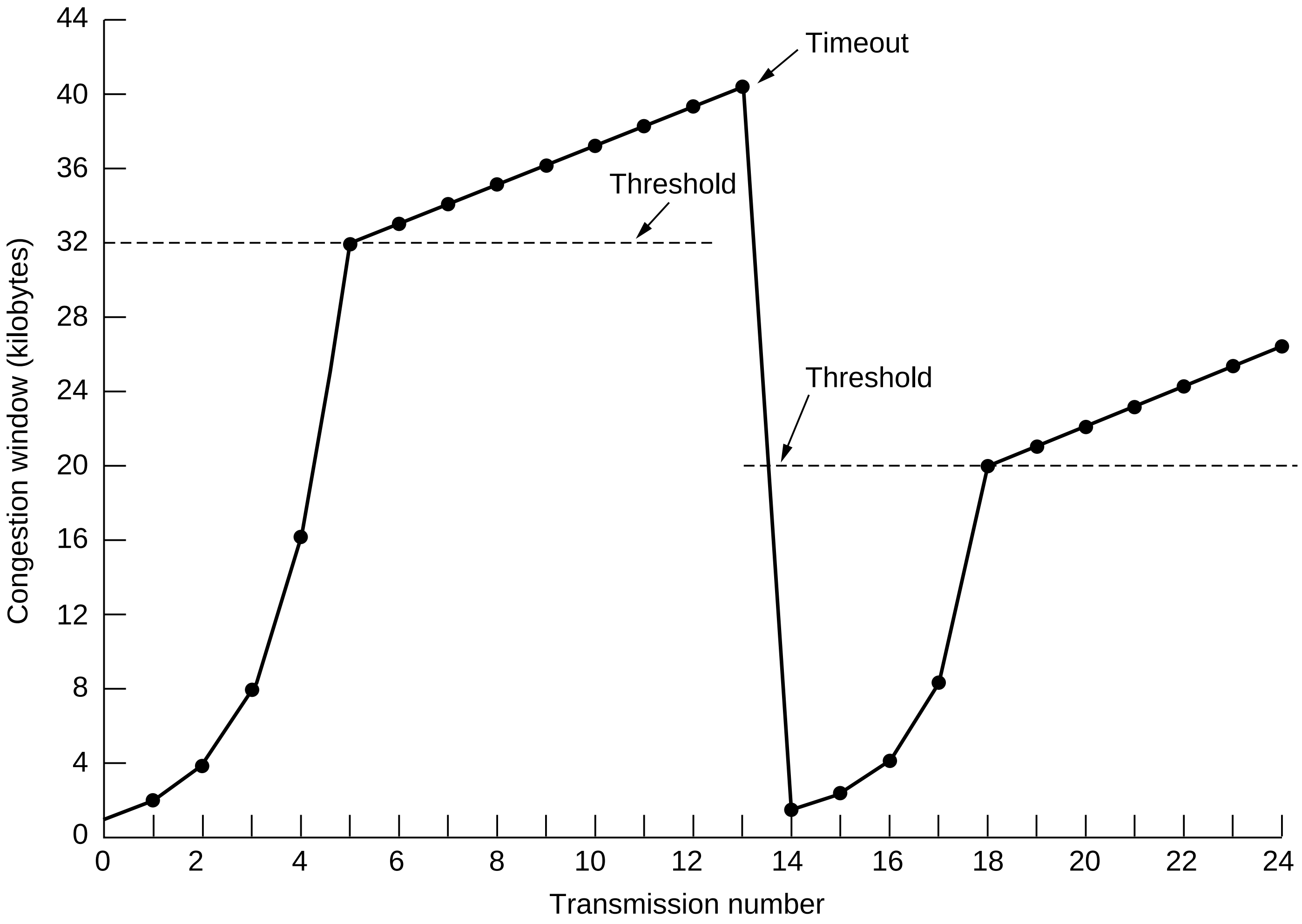
```
// <time> <sender> <destination> <port> <sequence number> <RTO> <measured RTT>
1236     1       0       1       1       3948    816
1852     1       0       1       1       3990    616
2668     1       0       1       1       3646    816
3284     1       0       1       1       3551    616
4100     1       0       1       1       3178    816
```

IV Extracting information: plot example

$ ./getCWND.pl > cwnd.txt	**generate Data File**
$ vi plot.gnuplot	**edit gnuplot script**
$ gnuplot plot.gnuplot	**run gnuplot script**

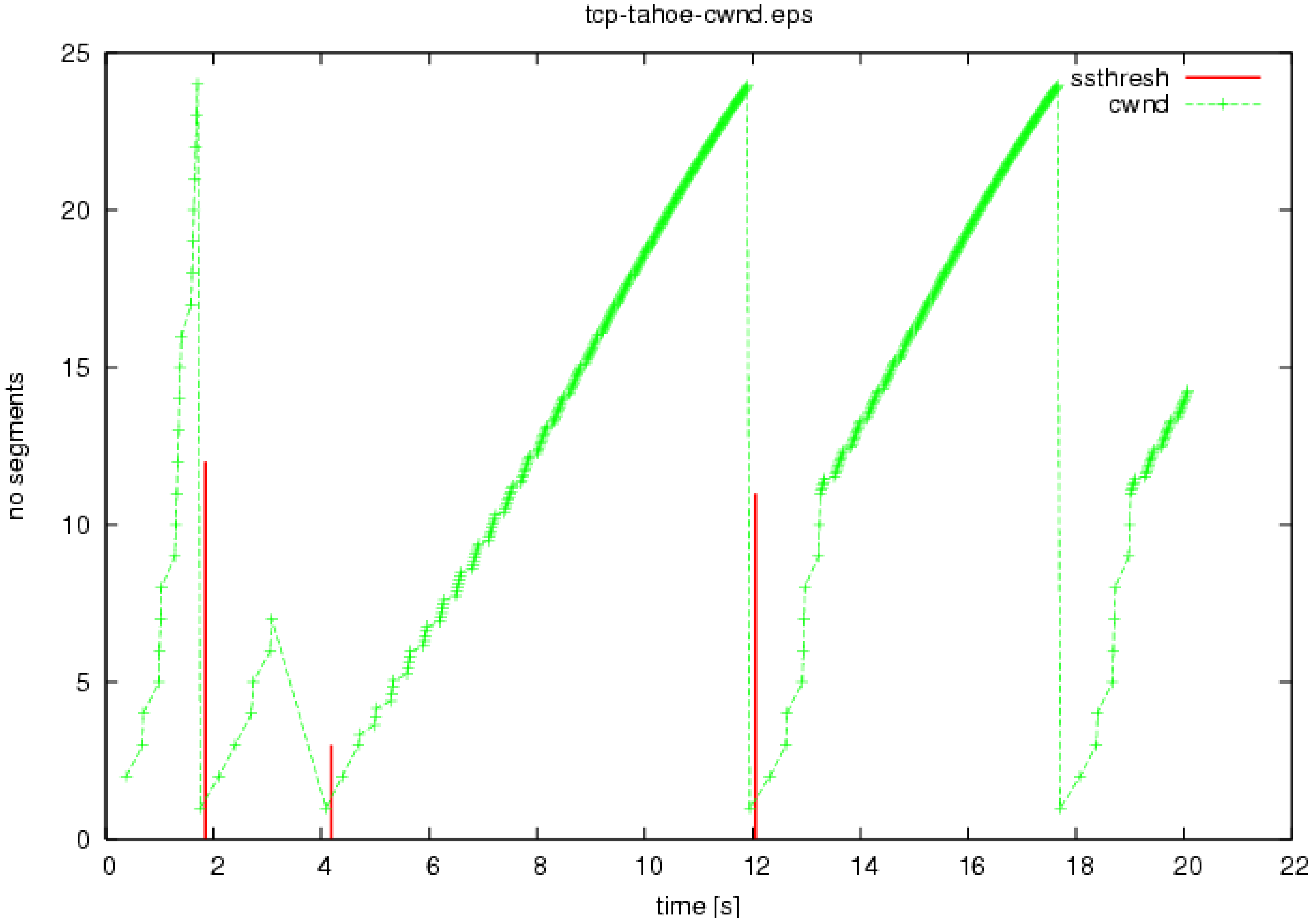
```
set term post eps mono
set out "cwnd.eps"
set title "TCP Tahoe"
set xlabel "time"
set ylabel "cwnd"

plot "cwnd.txt" title "Contention Window Size" with lines
```

I TCP Tahoe congestion control example

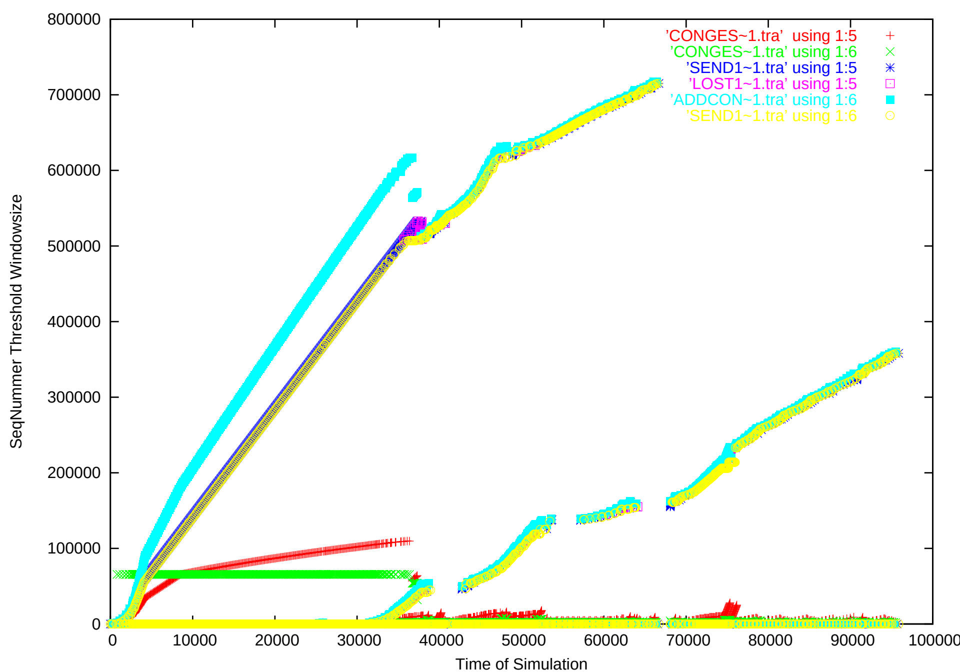


[Tanenbaum, CN3]

IV Resulting graphs: Tahoe



Note: shown is the cwmd whenever it is changed, NOT every packet.



'CONGES~1.tra' using 1:5 +
'CONGES~1.tra' using 1:6 ×
'SEND1~1.tra' using 1:5 ✳
'LOST1~1.tra' using 1:5 □
'ADDCON~1.tra' using 1:6 ■
'SEND1~1.tra' using 1:6 ○
SeqNummer Threshold Windowsize
0
100000
200000
300000
400000
500000
600000
700000
800000
Time of Simulation
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000