

NS internals

Velibor Markovski

Communication Networks Laboratory

School of Engineering Science

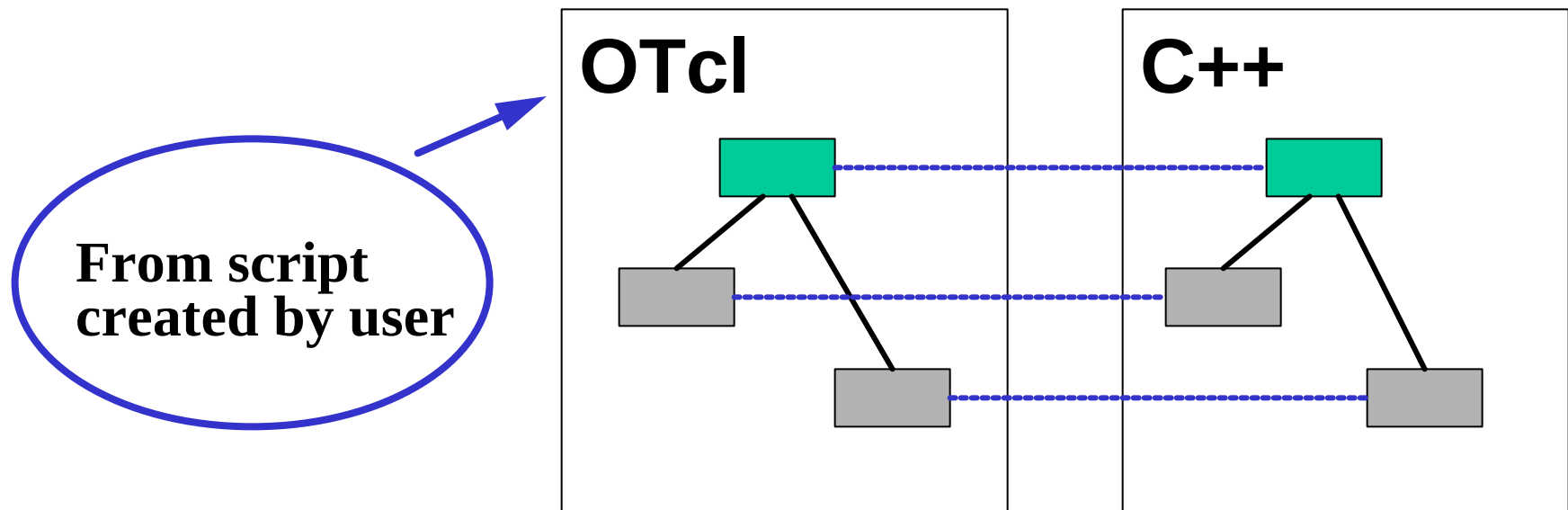
Simon Fraser University

Software architecture

- Split programming model
(C++ and OTcl)
- Simulation kernel ® C++
- Simulation definition,
configuration and control ® OTcl
- Tradeoff: performance vs. flexibility

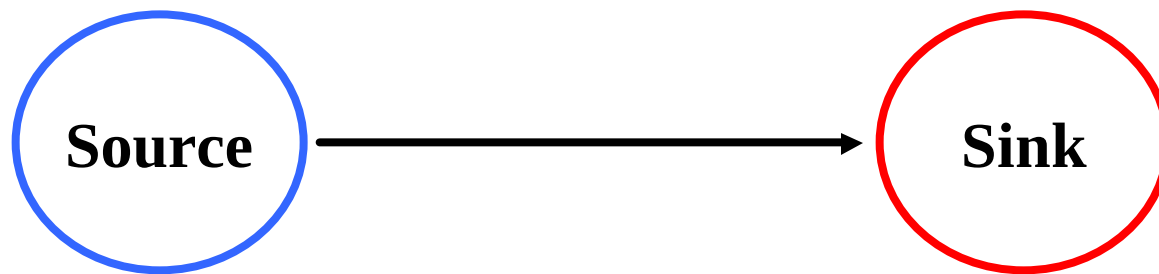
Class mirroring

- OTcl class hierarchy
- C++ class hierarchy



Simple script

- Two nodes
- CBR traffic from source to destination
- UDP used as transport protocol



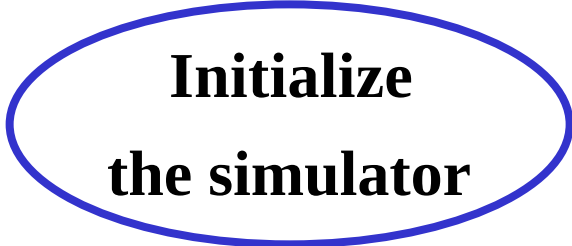
Simple script

```
set ns [new Simulator]
$ns use-scheduler List
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

Simple script - initialization

**set ns [new Simulator]
\$ns use-scheduler List**

set n0 [\$ns node]
set n1 [\$ns node]
\$ns duplex-link \$n0 \$n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
\$ns attach-agent \$n0 \$cbr0
\$cbr0 set packetSize_ 500
\$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
\$ns attach-agent \$n1 \$null0
\$ns connect \$cbr0 \$null0
\$ns at 0.5 "\$cbr0 start"
\$ns at 4.5 "\$cbr0 stop"



**Initialize
the simulator**

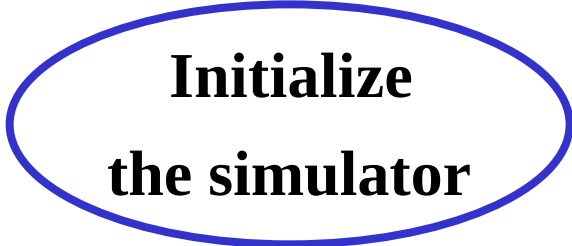
Simple script - initialization

```
set ns [poseidon 6: /opt/tmp/velibor/ns/examples> ns
$ns use % set ns [new Simulator]
set n0 [$r _o4
set n1 [$r
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

Simple script - initialization

**set ns [new Simulator]
\$ns use-scheduler List**

set n0 [\$ns node]
set n1 [\$ns node]
\$ns duplex-link \$n0 \$n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
\$ns attach-agent \$n0 \$cbr0
\$cbr0 set packetSize_ 500
\$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
\$ns attach-agent \$n1 \$null0
\$ns connect \$cbr0 \$null0
\$ns at 0.5 "\$cbr0 start"
\$ns at 4.5 "\$cbr0 stop"



**Initialize
the simulator**

Simple script - initialization

```
set ns [new Simulator]  
$ns use-scheduler List  
set n0 [$ns node]  
set n1 [$ns node]  
$ns duplex-link $n0 $n1 1Mb 10ms DropTail  
set cbr0 [new Agent/CBR]  
$ns attach-agent $n0 $cbr0  
$cbr0 set packetSize_ 500  
$cbr0 set interval_ 0.005  
set null0 [new Agent/Null]  
$ns attach-agent $n1 $null0  
$ns connect $cbr0 $null0  
$ns at 0.5 "$cbr0 start"  
$ns at 4.5 "$cbr0 stop"
```

Initialize
the simulator

Use linked list
as event scheduler

Simple script - initialization

```
poseidon 6: /opt/tmp/velibor/ns/examples> ns
set ns [new Simulator]
$ns u _o4
set n0 [new Agent]
set n1 [new Agent]
$ns du _o4 info vars
set cbr [new CBR]
$ns att _o4 allocAddr_ nullAgent_ scheduler_
$nbr0 s _o4 packetManager_
$nbr0 s _o4 % _o4 set scheduler_
set nul [new Null]
$ns att _o11
$ns cor _o11 % _o11 info class
$ns at 0 $ns at 0 Scheduler/List
```

Simple script - initialization

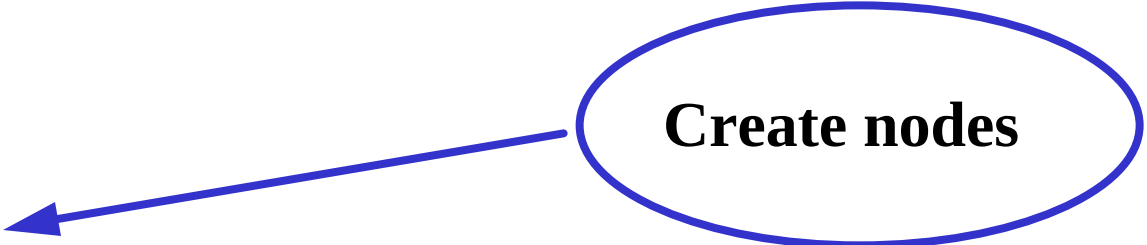
```
set ns [new Simulator]  
$ns use-scheduler List  
set n0 [$ns node]  
set n1 [$ns node]  
$ns duplex-link $n0 $n1 1Mb 10ms DropTail  
set cbr0 [new Agent/CBR]  
$ns attach-agent $n0 $cbr0  
$cbr0 set packetSize_ 500  
$cbr0 set interval_ 0.005  
set null0 [new Agent/Null]  
$ns attach-agent $n1 $null0  
$ns connect $cbr0 $null0  
$ns at 0.5 "$cbr0 start"  
$ns at 4.5 "$cbr0 stop"
```

Initialize
the simulator

Use linked list
as event scheduler

Simple script – node creation

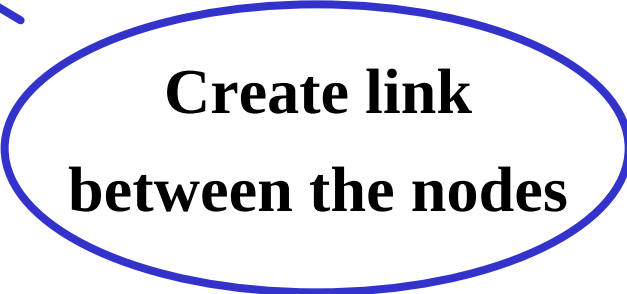
```
set ns [new Simulator]
$ns use-scheduler List
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```



Create nodes

Simple script – link creation

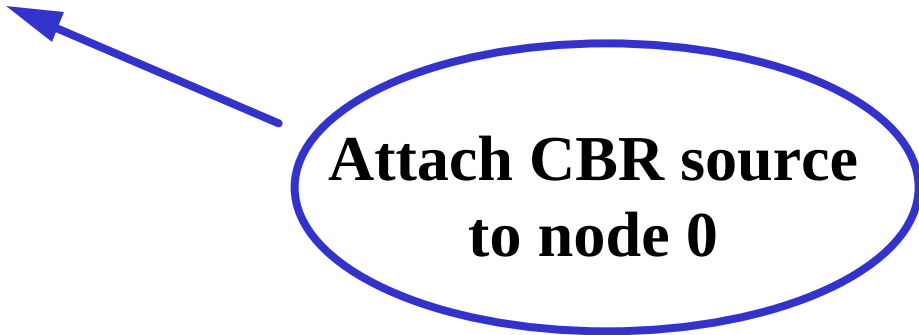
```
set ns [new Simulator]
$ns use-scheduler List
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```



**Create link
between the nodes**

Simple script – setting up source

```
set ns [new Simulator]
$ns use-scheduler List
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```



**Attach CBR source
to node 0**

Simple script – setting up sink

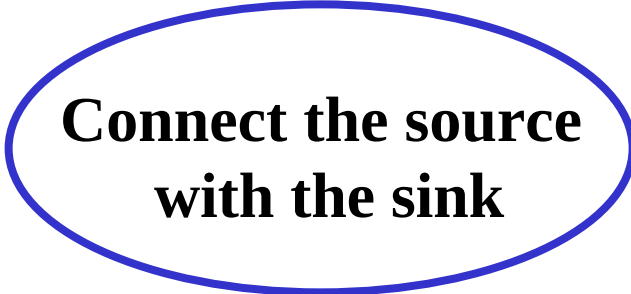
```
set ns [new Simulator]
$ns use-scheduler List
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```



**Attach sink to
Node 1**

Simple script – connecting nodes

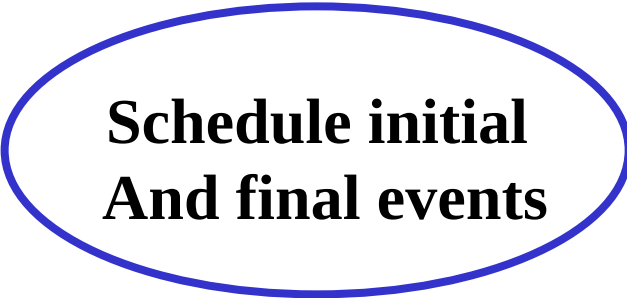
```
set ns [new Simulator]
$ns use-scheduler List
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```



**Connect the source
with the sink**

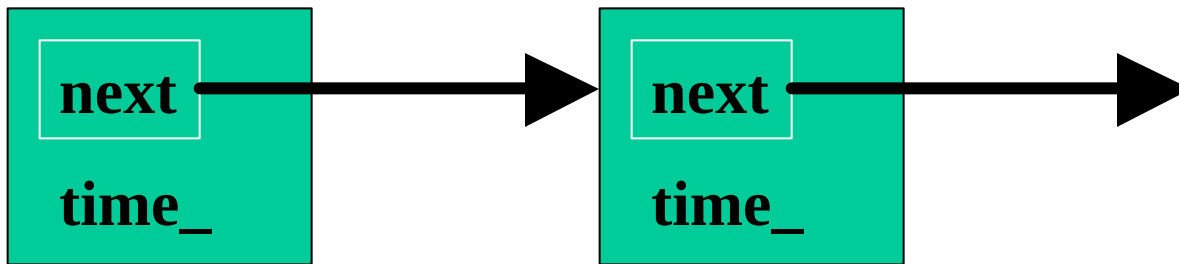
Simple script – schedule “at” events

```
set ns [new Simulator]
$ns use-scheduler List
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
$ns connect $cbr0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```



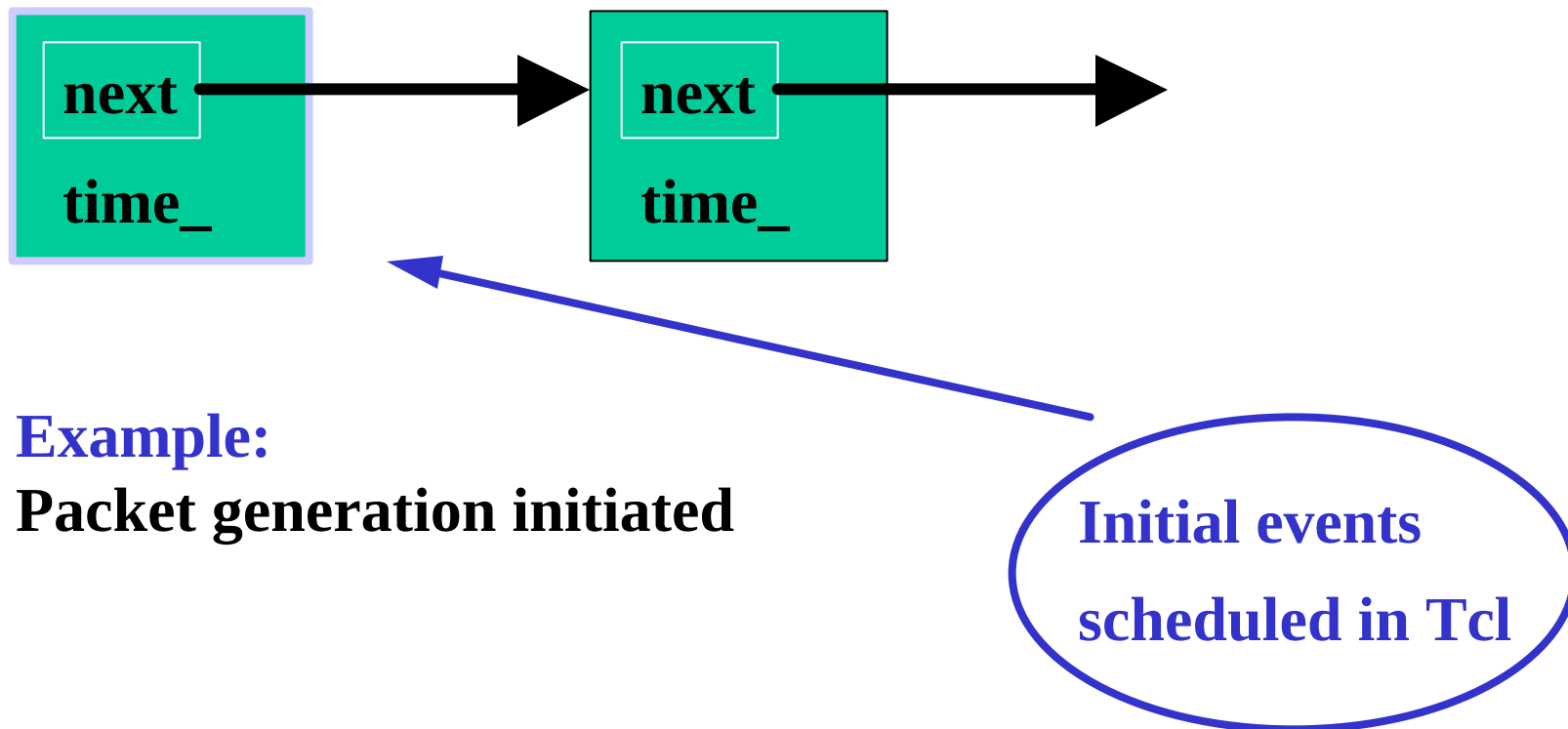
**Schedule initial
And final events**

Event scheduler

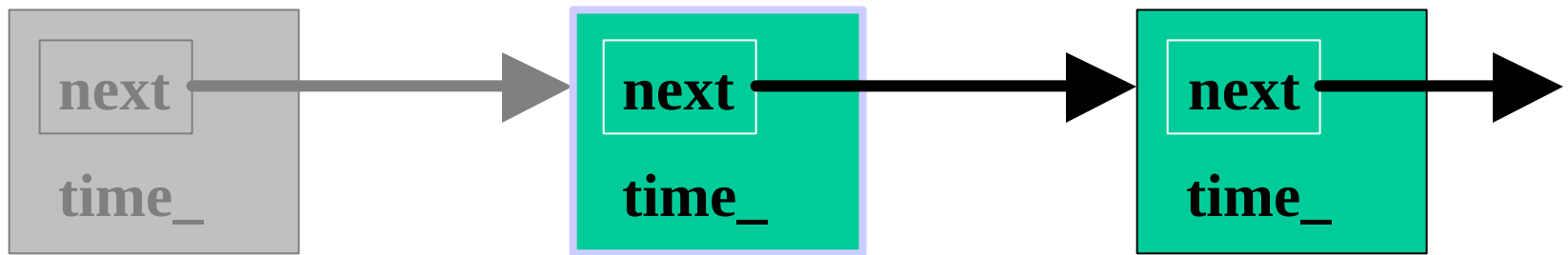


```
class Event {  
    Event* next_  
    Handler* handler_  
    double time_  
    int uid_  
    Event() : time_(0), uid_(0) {}  
};
```

Initial events



Initial and follow-up events

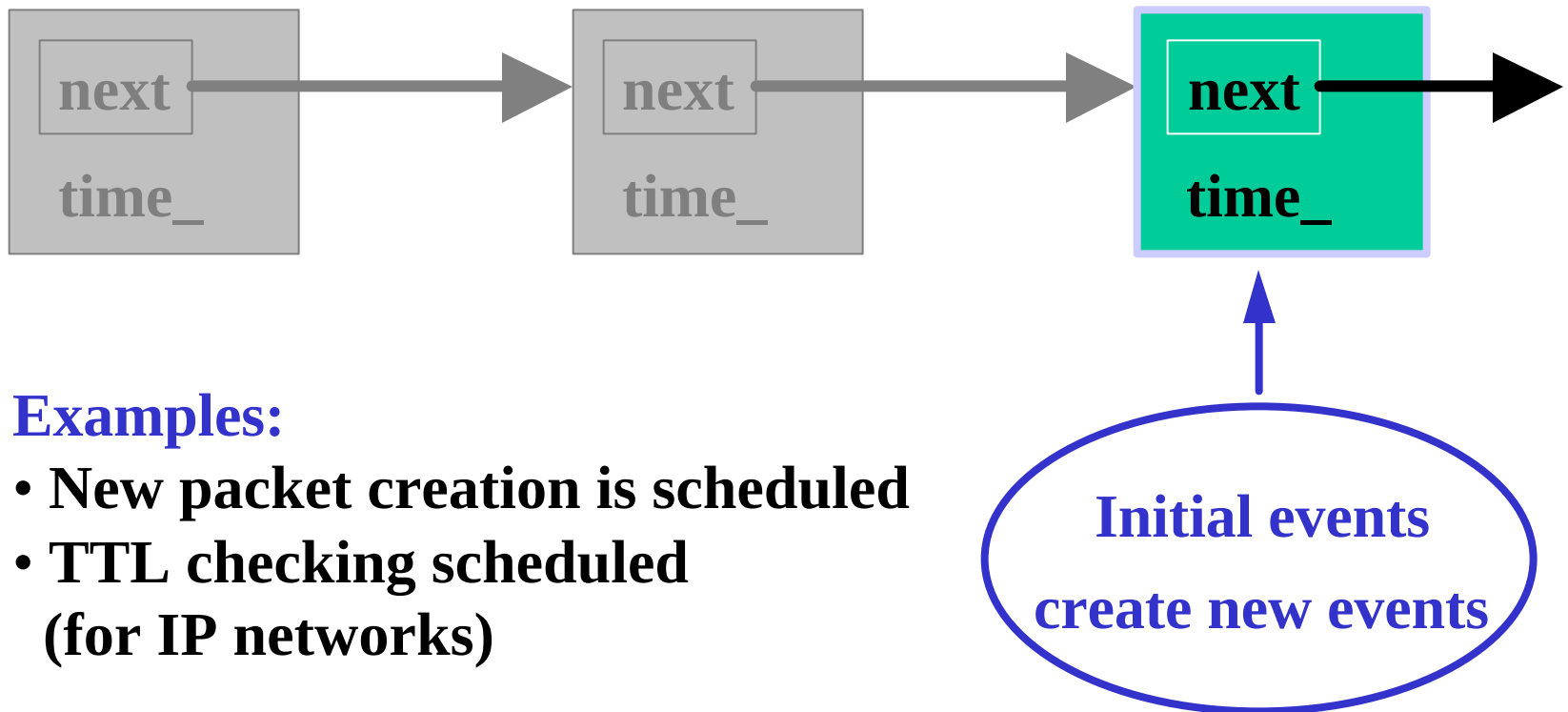


Examples:

- New packet creation is scheduled
- TTL checking scheduled (for IP networks)

**Initial events
create new events**

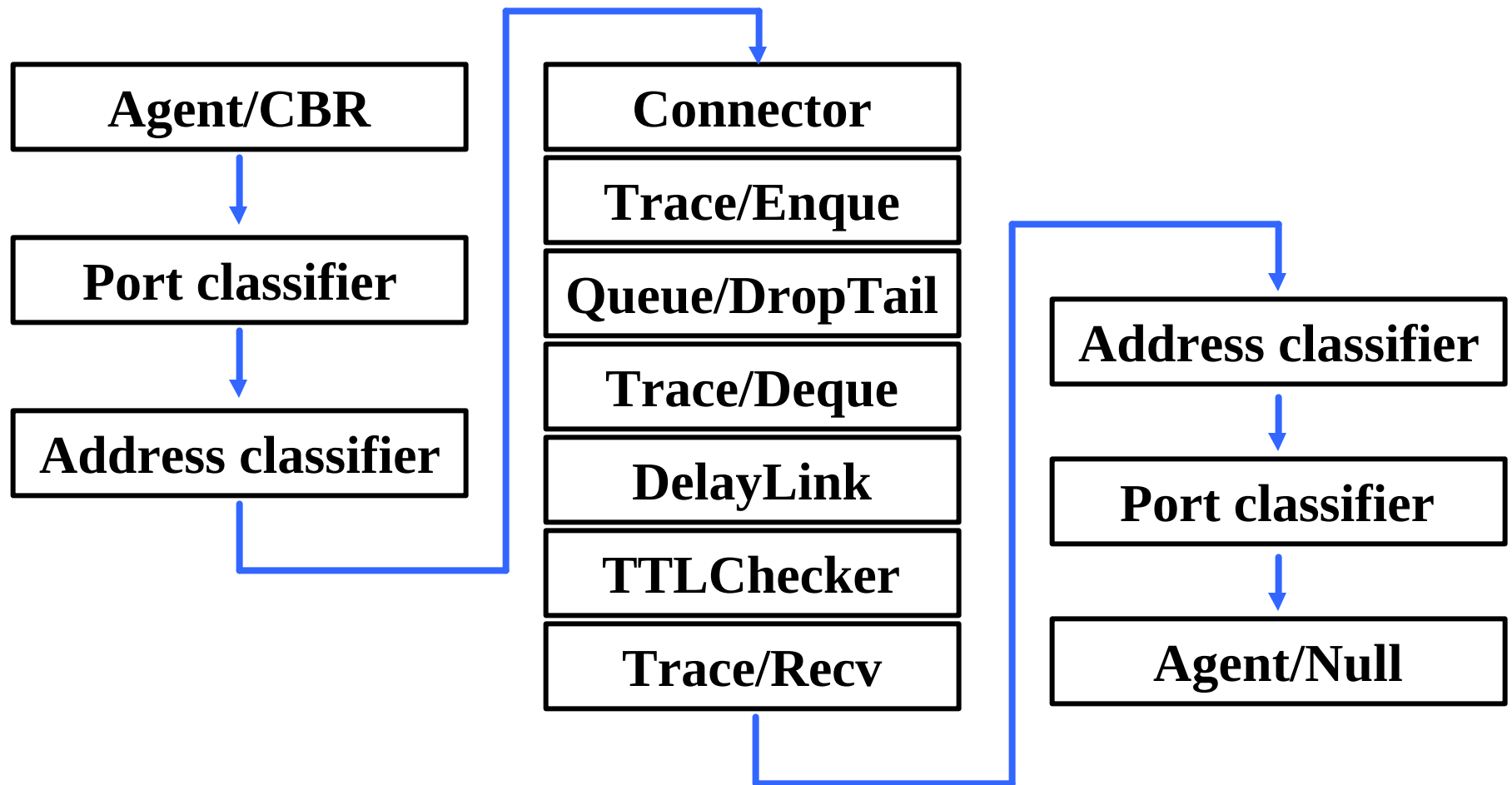
Initial and follow-up events



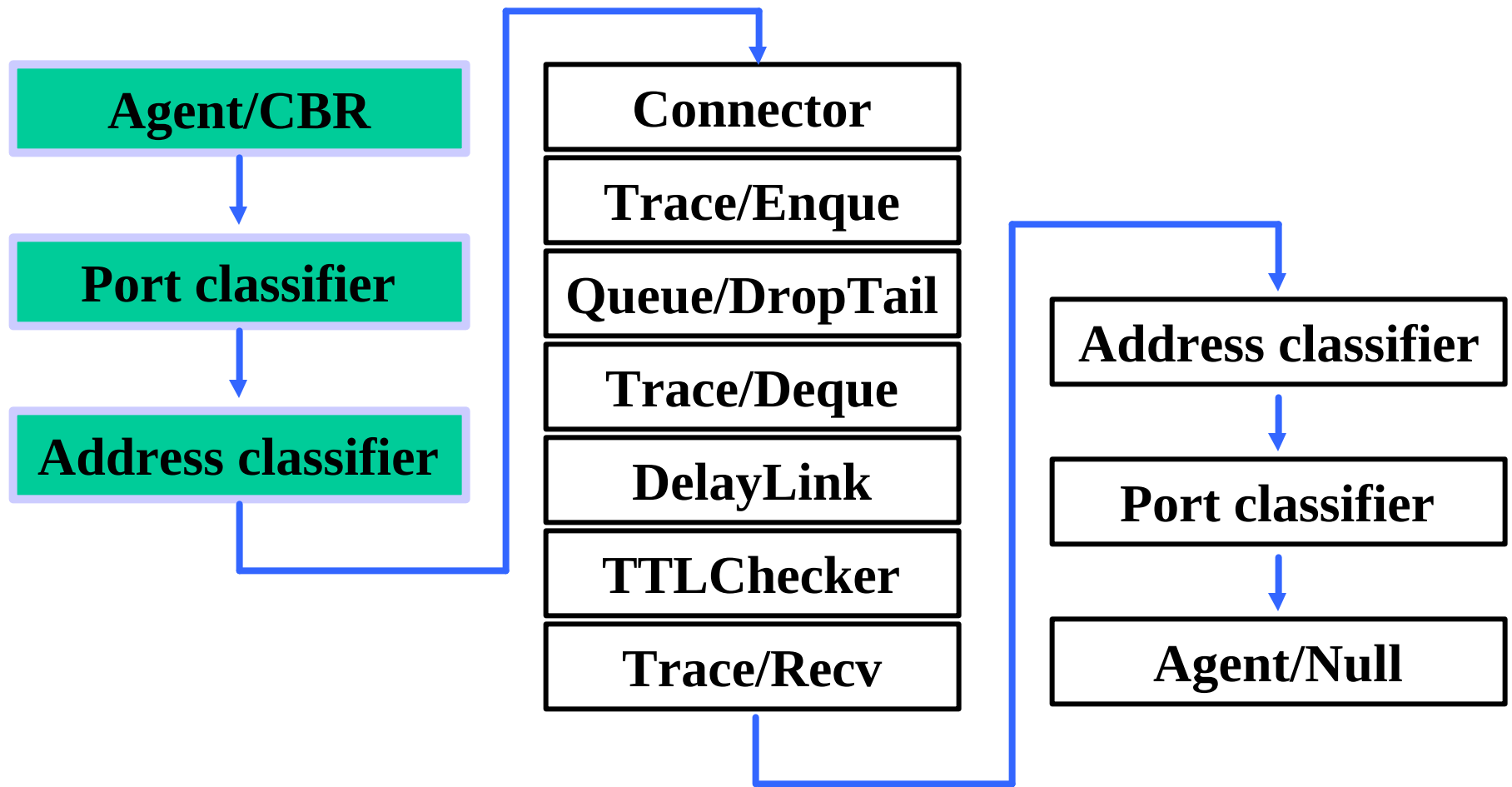
Scheduler's main functions

- Schedule events at certain time
 - Insert events in the list
- Dequeue events from the list
- Dispatch dequeued events
 - Call appropriate event handler

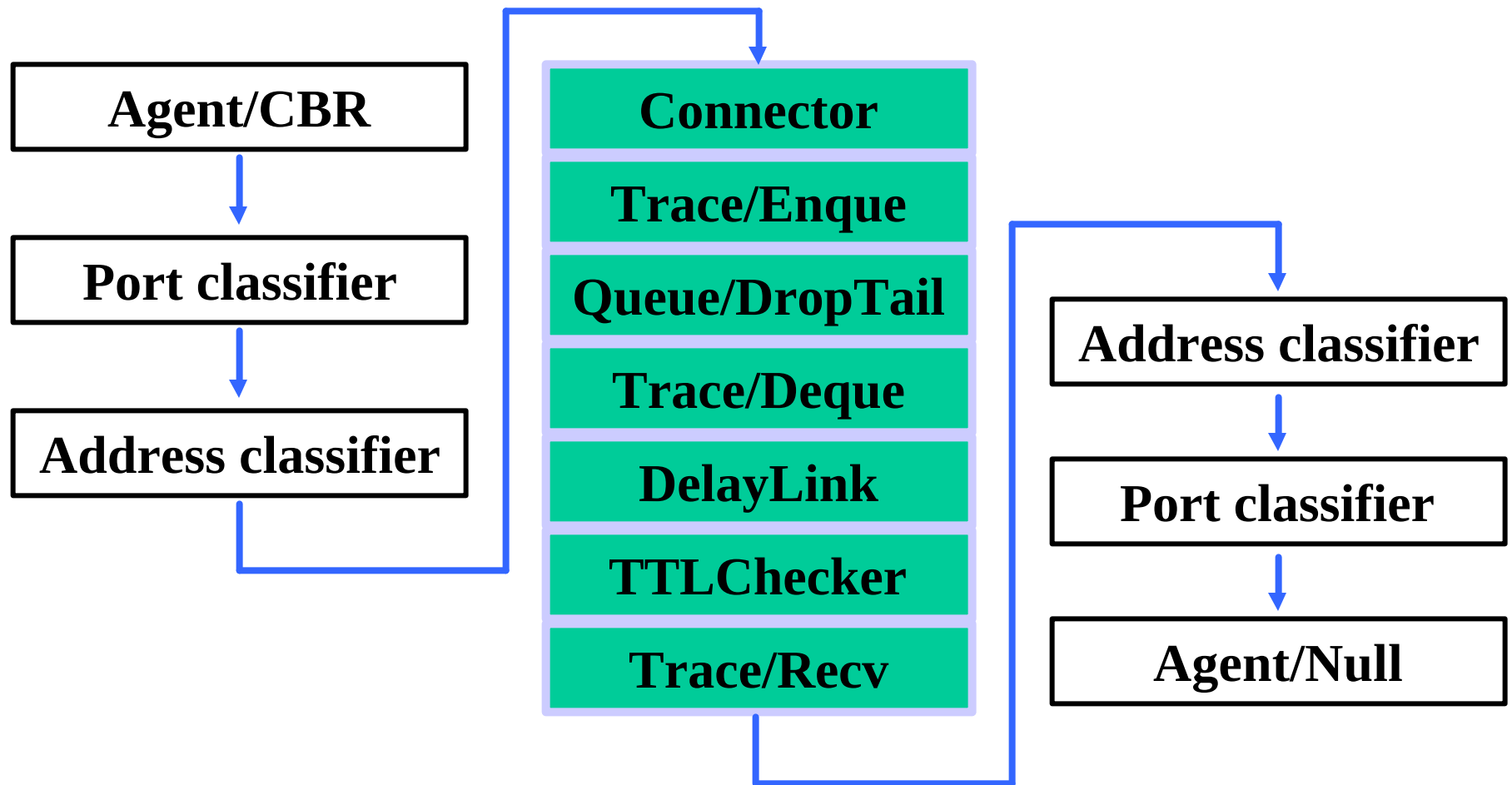
Back to our example



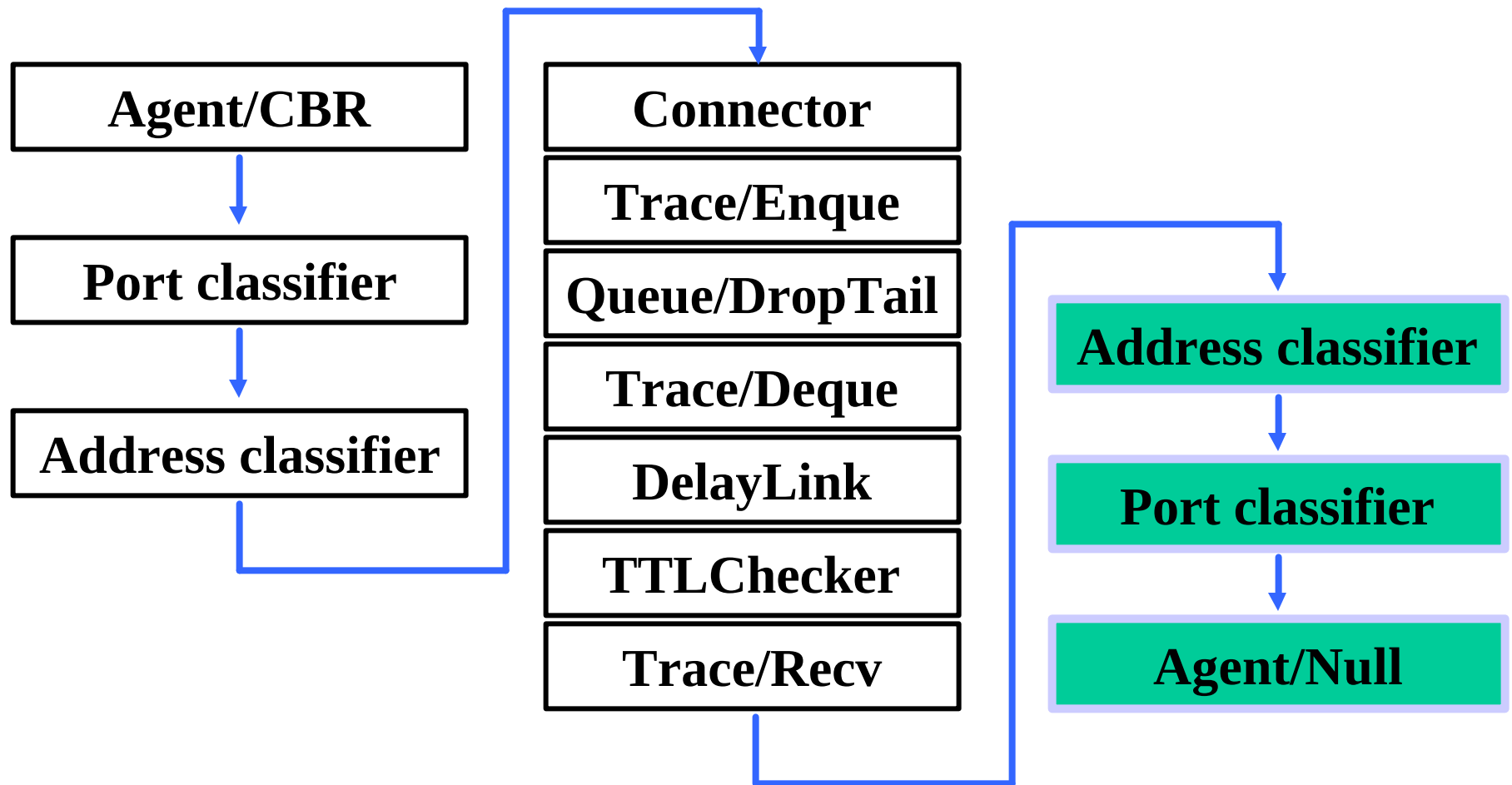
Back to our example



Back to our example



Back to our example



Even more details...

```
(gdb) b Scheduler::run
```

```
Breakpoint 1 at 0x9c054: file scheduler.cc, line 81.
```

```
(gdb) run
```

```
Breakpoint 1, Scheduler::run (this=0x473a08) at scheduler.cc:81
```

```
81         instance_ = this;
```

```
(gdb) l scheduler.cc:79,86
```

```
79     Scheduler::run()
```

```
80     {
```

```
81         instance_ = this;
```

```
82         Event *p;
```

```
83         while ((p = deque()) && !halted_) {
```

```
84             dispatch(p);
```

```
85         }
```

```
86     }
```

```
(gdb)
```

Complex software structure

Approximate numbers:

- 27K lines of C++ code (ns-2.1b5)
- 12K lines of OTcl code
- 18K lines of test suites and examples

Why simulators at all?

- Inexpensive
- Flexible:
 - with scenario generation
 - with link speeds
 - with topologies
- Access to ALL data

Alternatives to simulation

- Testbeds
- Emulation (testbed-simulator mix)
- Live Internet tests

FOR MORE INFO...

- <http://www-mash.cs.berkeley.edu/ns/ns.html>
- Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Halдар, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala,
[“Improving Simulation for Network Research,”](#)
Technical Report 99-702, University of Southern California, March 1999.
- Vern Paxson and Sally Floyd,
[“Why We Don't Know How To Simulate The Internet,”](#)
Proceedings of the 1997 Winter Simulation Conference, Phoenix, Arizona, December 1997.