ภาคผนวก ค รวมสคริปต์ OTcl ที่ใช้สำหรับทดสอบในบทที่ 5

#OTcl สคริปต์สำหรับการทดลองเรื่อง OM และ DDM ทอ พอโลยี 1

```
set ns [new Simulator]

set nf [open out.nam w]
set tf [open out.tr w]
set df1_5 [open drop1_5.tr w]
set df5_2 [open drop5_2.tr w]
set df2_5 [open drop2_5.tr w]
set df5_3 [open drop5_3.tr w]
set df5_4 [open drop5_4.tr w]

$ns namtrace-all $nf
$ns trace-all $tf

set sumqa 0
set sumqb 0
set sumqc 0
set sumqd 0
set numpkt 0

set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]
set node5 [$ns node]

$ns simplex-link $node1 $node5 1kb 10ms DropTail
$ns simplex-link $node5 $node1 1kb 10ms DropTail
$ns simplex-link $node5 $node3 1kb 10ms DropTail
$ns simplex-link $node3 $node5 1kb 10ms DropTail
$ns simplex-link $node5 $node2 1kb 10ms DropTail
$ns simplex-link $node2 $node5 1kb 10ms DropTail
$ns simplex-link $node5 $node4 1kb 10ms DropTail
$ns simplex-link $node4 $node5 1kb 10ms DropTail


$ns queue-limit $node1 $node5 10000
$ns queue-limit $node5 $node2 10000
$ns queue-limit $node2 $node5 10000
$ns queue-limit $node5 $node3 10000
$ns queue-limit $node5 $node4 10000


set drop1_5 [$ns create-trace Drop $df1_5 $node1 $node5]
$ns drop-trace $node1 $node5 $drop1_5
set drop5_2 [$ns create-trace Drop $df5_2 $node5 $node2]
$ns drop-trace $node5 $node2 $drop5_2
set drop2_5 [$ns create-trace Drop $df2_5 $node2 $node5]
$ns drop-trace $node2 $node5 $drop2_5
set drop5_3 [$ns create-trace Drop $df5_3 $node5 $node3]
$ns drop-trace $node5 $node3 $drop5_3
set drop5_4 [$ns create-trace Drop $df5_4 $node5 $node4]
$ns drop-trace $node5 $node4 $drop5_4
```

```
$ns duplex-link-op $node1 $node5 orient right
$ns duplex-link-op $node5 $node2 orient right
$ns duplex-link-op $node5 $node3 orient right-up
$ns duplex-link-op $node5 $node4 orient right-down


set qa [$ns monitor-queue $node1 $node5 ""]
set qb [$ns monitor-queue $node5 $node2 ""]
set qc [$ns monitor-queue $node2 $node5 ""]
set qd [$ns monitor-queue $node5 $node3 ""]

set expo0 [new RandomVariable/Exponential]
$expo0 set avg_ 1.26
set expo1 [new RandomVariable/Exponential]
$expo1 set avg_ 110

set udp_1 [new Agent/UDP]
set udp_2 [new Agent/UDP]
set udp_3 [new Agent/UDP]
set udp_4 [new Agent/UDP]
set udp_5 [new Agent/UDP]
set udp_6 [new Agent/UDP]
$udp_1 set packetSize_ 1000
$udp_2 set packetSize_ 1000
$udp_3 set packetSize_ 1000
$udp_4 set packetSize_ 1000
$udp_5 set packetSize_ 1000
$udp_6 set packetSize_ 1000

$ns attach-agent $node1 $udp_1
$ns attach-agent $node2 $udp_2
$ns attach-agent $node2 $udp_3
$ns attach-agent $node2 $udp_5
$ns attach-agent $node4 $udp_4
$ns attach-agent $node3 $udp_6

$ns connect $udp_1 $udp_2
$ns connect $udp_3 $udp_4
$ns connect $udp_5 $udp_6

set obj_2 [new Application/TransObj]
set obj_3 [new Application/RecvObj]
set obj_4 [new Application/RecvObj]
set obj_5 [new Application/RecvObj]
set myapp_1 [new Application/SimObj]

set fed1 [new Application/FedOM]
set fed1_s  [new Application/FedOM]
set fed2 [new Application/FedOM]
set fed2_s [new Application/FedOM]

$fed1 attachsimobj $myapp_1
$fed1 attach-agent $udp_1
$fed1_s attachsimobj $obj_2
$fed1_s attach-agent $udp_2

$fed2_s attachsimobj $obj_3
$fed2_3 attach-agent $udp_3
```

```
$fed2 attach-agent $udp_4

$obj_3 subscribe $obj_2

proc finish {} {
      global ns nf tf df1_5 df5_2 df2_5 df5_3 df5_4 sumqa sumqb sumqc
sumqd sumqe numpkt
      $ns flush-trace
      close $nf
      close $tf
      close $df1_5
      close $df5_2
      close $df2_5
      close $df5_3
      close $df5_4

      set now [$ns now]
      set sumqa [expr $sumqa * 1]
      #set sumqb [expr $sumqb * 1000]
      set sumqc [expr $sumqc * 1000]
      set sumqd [expr $sumqd * 1000]

      set avg_qalen [expr $sumqa / $now]
      #set avg_qblen [expr $sumqb / $now]
      set avg_qclen [expr $sumqc / $now]
      set avg_qdlen [expr $sumqd / $now]

      puts "avg_qalen = [format %#.4g $avg_qalen]"
      puts "avg_qblen = [format %#.4g $avg_qblen]"
      puts "avg_qclen = [format %#.4g $avg_qclen]"
      puts "avg_qdlen = [format %#.4g $avg_qdlen]"
      puts "numpkt=[format %#.4g $numpkt]"
      #exec nam out.nam &
      exit 0
}

proc sendpacket {} {
      global ns bytecount0 src expo1 expo0 myapp_1 numpkt

      set numpkt [expr $numpkt +1]
      set now [$ns now]
      $ns at [expr $now + [$expo0 value]] "sendpacket"
      set bytes [expr round ([$expo1 value])]
      $myapp_1 send $bytes

}

proc observe {} {
      global ns qa qb qc qd sumqa sumqb sumqc sumqd
      set now [$ns now]
      set qalen [$qa set pkts_]
      set qblen [$qb set pkts_]
      set qclen [$qc set pkts_]
      set qdlen [$qd set pkts_]

      set sumqa [expr $sumqa + $qalen ]
      set sumqb [expr $sumqb + $qblen ]
      set sumqc [expr $sumqc + $qclen ]
      set sumqd [expr $sumqd + $qdlen ]

      $ns at [expr $now + 1] "observe"

}
$ns at 0.00001 "sendpacket"
$ns at 0.000001 "observe"
$ns at 10000 "finish"
$ns run
```

#OTcl สคริปต์สำหรับการทดลองเรื่อง OM และ DDM ทอพอโลยี 2

```
set ns [new Simulator]

set nf [open out.nam w]
set tf [open out.tr w]
set df1_5 [open drop1_5.tr w]
set df5_2 [open drop5_2.tr w]
set df2_5 [open drop2_5.tr w]
set df5_6 [open drop2_5.tr w]
set df5_4 [open drop5_4.tr w]
set df6_3 [open drop2_5.tr w]


$ns namtrace-all $nf
$ns trace-all $tf

set sumqa 0
set sumqb 0
set sumqc 0
set sumqd 0
set numpkt 0

set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]
set node5 [$ns node]

$ns simplex-link $node1 $node5 1kb 10ms DropTail
$ns simplex-link $node5 $node1 1kb 10ms DropTail
$ns simplex-link $node3 $node6 1kb 10ms DropTail
$ns simplex-link $node6 $node3 1kb 10ms DropTail
$ns simplex-link $node5 $node6 1kb 10ms DropTail
$ns simplex-link $node6 $node5 1kb 10ms DropTail
$ns simplex-link $node5 $node2 1kb 10ms DropTail
$ns simplex-link $node2 $node5 1kb 10ms DropTail
$ns simplex-link $node5 $node4 1kb 10ms DropTail
$ns simplex-link $node4 $node5 1kb 10ms DropTail


$ns queue-limit $node1 $node5 10000
$ns queue-limit $node5 $node2 10000
$ns queue-limit $node2 $node5 10000
$ns queue-limit $node6 $node3 10000
$ns queue-limit $node5 $node4 10000
$ns queue-limit $node5 $node6 10000
```

```
set drop1_5 [$ns create-trace Drop $df1_5 $node1 $node5]
$ns drop-trace $node1 $node5 $drop1_5
set drop5_2 [$ns create-trace Drop $df5_2 $node5 $node2]
$ns drop-trace $node5 $node2 $drop5_2
set drop2_5 [$ns create-trace Drop $df2_5 $node2 $node5]
$ns drop-trace $node2 $node5 $drop2_5
set $drop6_3 [$ns create-trace Drop $df6_3 $node5 $node4]
$ns drop-trace $node6 $node3 $drop6_3
set drop5_4 [$ns create-trace Drop $df5_4 $node5 $node4]
$ns drop-trace $node5 $node4 $drop5_4

$ns duplex-link-op $node1 $node5 orient right
$ns duplex-link-op $node5 $node2 orient right
$ns duplex-link-op $node5 $node6 orient right-up
$ns duplex-link-op $node6 $node3 orient right-up
$ns duplex-link-op $node5 $node4 orient right-down


set qa [$ns monitor-queue $node1 $node5 ""]
set qb [$ns monitor-queue $node5 $node2 ""]
set qc [$ns monitor-queue $node2 $node5 ""]
set qd [$ns monitor-queue $node5 $node6 ""]
set qe [$ns monitor-queue $node6 $node3 ""]


set expo0 [new RandomVariable/Exponential]
$expo0 set avg_ 1.26
set expo1 [new RandomVariable/Exponential]
$expo1 set avg_ 110

set udp_1 [new Agent/UDP]
set udp_2 [new Agent/UDP]
set udp_3 [new Agent/UDP]
set udp_4 [new Agent/UDP]
set udp_5 [new Agent/UDP]
set udp_6 [new Agent/UDP]
$udp_1 set packetSize_ 1000
$udp_2 set packetSize_ 1000
$udp_3 set packetSize_ 1000
$udp_4 set packetSize_ 1000
$udp_5 set packetSize_ 1000
$udp_6 set packetSize_ 1000

$ns attach-agent $node1 $udp_1
$ns attach-agent $node2 $udp_2
$ns attach-agent $node2 $udp_3
$ns attach-agent $node2 $udp_5
$ns attach-agent $node4 $udp_4
$ns attach-agent $node3 $udp_6

$ns connect $udp_1 $udp_2
$ns connect $udp_3 $udp_4
$ns connect $udp_5 $udp_6

set obj_2 [new Application/TransObj]
set obj_3 [new Application/RecvObj]
set obj_4 [new Application/RecvObj]
set obj_5 [new Application/RecvObj]
set myapp_1 [new Application/SimObj]

set fed1 [new Application/FedOM]
set fed1_s  [new Application/FedOM]
set fed2 [new Application/FedOM]
set fed2_s [new Application/FedOM]

$fed1 attachsimobj $myapp_1
$fed1 attach-agent $udp_1
$fed1_s attachsimobj $obj_2
$fed1_s attach-agent $udp_2

$fed2_s attachsimobj $obj_3
$fed2_3 attach-agent $udp_3
$fed2 attach-agent $udp_4

$obj_3 subscribe $obj_2

proc finish {} {
      global ns nf tf df1_5 df5_2 df2_5 df5_3 df5_4 sumqa sumqb sumqc
sumqd sumqe numpkt
      $ns flush-trace
      close $nf
      close $tf
      close $df1_5
      close $df5_2
      close $df2_5
      close $df5_3
      close $df5_4

      set now [$ns now]
      set sumqa [expr $sumqa * 1]
      #set sumqb [expr $sumqb * 1000]
      set sumqc [expr $sumqc * 1000]
      set sumqd [expr $sumqd * 1000]

      set avg_qalen [expr $sumqa / $now]
      #set avg_qblen [expr $sumqb / $now]
      set avg_qclen [expr $sumqc / $now]
      set avg_qdlen [expr $sumqd / $now]

      puts "avg_qalen = [format %#.4g $avg_qalen]"
      puts "avg_qblen = [format %#.4g $avg_qblen]"
      puts "avg_qclen = [format %#.4g $avg_qclen]"
      puts "avg_qdlen = [format %#.4g $avg_qdlen]"
      puts "numpkt=[format %#.4g $numpkt]"
      #exec nam out.nam &
      exit 0
}

proc sendpacket {} {
      global ns bytecount0 src expo1 expo0 myapp_1 numpkt

      set numpkt [expr $numpkt +1]
      set now [$ns now]
      $ns at [expr $now + [$expo0 value]] "sendpacket"
      set bytes [expr round ([$expo1 value])]
      $myapp_1 send $bytes
}

proc observe {} {
      global ns qa qb qc qd sumqa sumqb sumqc sumqd
      set now [$ns now]
      set qalen [$qa set pkts_]
      set qblen [$qb set pkts_]
      set qclen [$qc set pkts_]
      set qdlen [$qd set pkts_]

      set sumqa [expr $sumqa + $qalen ]
      set sumqb [expr $sumqb + $qblen ]
      set sumqc [expr $sumqc + $qclen ]
      set sumqd [expr $sumqd + $qdlen ]

      $ns at [expr $now + 1] "observe"

}
$ns at 0.00001 "sendpacket"
$ns at 0.000001 "observe"
$ns at 10000 "finish"
$ns run
```

#OTcl สคริปต์สำหรับการทดลองเรื่อง TM 1 Regulating Federate

```
set namfile out.nam
set tracefile     out.tr

set ns [new Simulator]

set nf [open $namfile w]
$ns namtrace-all $nf
set f [open $tracefile w]
$ns trace-all $f

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set router [$ns node]

$ns color 0 Blue
$ns color 1 Red
$ns color 2 Green
$ns color 3 Yellow
$ns duplex-link $n0 $router 2Mb 50ms DropTail
$ns duplex-link $router $n1 2Mb 5ms DropTail
$ns duplex-link $router $n2 2Mb 5ms DropTail

$ns duplex-link-op $n0 $router orient right
$ns duplex-link-op $router $n1 orient up
$ns duplex-link-op $router $n2 orient right

set size0 [new RandomVariable/Exponential]
$size0 set avg_ 100

set tcp0 [new Agent/TCP/FullTcp]
set tcp1 [new Agent/TCP/FullTcp]
set tcp2 [new Agent/TCP/FullTcp]
set tcp3 [new Agent/TCP/FullTcp]
$tcp0 set fid_ 0
$tcp1 set fid_ 1
$tcp2 set fid_ 2
$tcp3 set fid_ 3
$ns attach-agent $n0 $tcp0
$ns attach-agent $n1 $tcp1
$ns attach-agent $n1 $tcp2
$ns attach-agent $n2 $tcp3

$ns connect $tcp0 $tcp1
$ns connect $tcp3 $tcp2
$tcp0 listen
$tcp1 listen
$tcp2 listen
$tcp3 listen

set rti [new Application/TimeManager]

set debug [new Application/TcpApp $tcp0]
set fedagent0 [new Application/FedTMTar $tcp0]
set fedagent1 [new Application/FedTM/FedAgent $tcp1]
set fedagent2 [new Application/FedTM/FedAgent $tcp2]
set fedagent3 [new Application/FedTMTar $tcp3]

set simobj [new Application/SimObj]
$simobj attachsize $size0
set simobj_s [new Application/TransObj]
set simobj_r [new Application/RecvObj]

$fedagent0 attachsimobj $simobj
$fedagent1 attachsimobj $simobj_s
$fedagent2 attachsimobj $simobj_r
$fedagent0 connect $fedagent1
$fedagent2 connect $fedagent3
$fedagent0 lookahead 1
$fedagent1 lookahead 1
$fedagent2 lookahead 1
$fedagent3 lookahead 1
$rti attachfedagent $fedagent1
$rti attachfedagent $fedagent2

#$rti setcon $fedagent0 $fedagent1

$fedagent3 enable-autofwd
$fedagent0 enable-autofwd

$fedagent0 enable-trace
$fedagent1 enable-trace
$fedagent2 enable-trace
$fedagent3 enable-trace


$obj_s subscribe-by $obj_r

Application/TcpApp instproc showmsg {data} {
      global ns
      $ns trace-annotate "$self : $data "
}

proc finish {} {
      global ns nf f namfile
      $ns flush-trace
      close $nf
      close $f

      puts "running nam..."
      exec nam $namfile &
      exit 0
}


$ns at 0.1 "$fedagent0 start"
$ns at 0.1 "$fedagent3 start"

$ns at 5.0 "finish"

$ns run
```

#OTcl สคริปต์สำหรับการทดลองเรื่อง TM 2 Regulating Federate

```
set namfile out.nam
set tracefile     out.tr

set ns [new Simulator]

set nf [open $namfile w]
$ns namtrace-all $nf
set f [open $tracefile w]
$ns trace-all $f

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set router [$ns node]

$ns color 0 Blue
$ns color 1 Red
$ns color 2 Green
$ns color 3 Yellow
$ns duplex-link $n0 $router 2Mb 50ms DropTail
$ns duplex-link $router $n1 2Mb 5ms DropTail
$ns duplex-link $router $n2 2Mb 5ms DropTail
$ns duplex-link $router $n3 2Mb 100ms DropTail

$ns duplex-link-op $n0 $router orient right
$ns duplex-link-op $router $n1 orient up
$ns duplex-link-op $router $n2 orient right

set size0 [new RandomVariable/Exponential]
$size0 set avg_ 100
set size1 [new RandomVariable/Exponential]
$size1 set avg_ 100
```

```
set tcp0 [new Agent/TCP/FullTcp]
set tcp1 [new Agent/TCP/FullTcp]
set tcp2 [new Agent/TCP/FullTcp]
set tcp3 [new Agent/TCP/FullTcp]
set tcp4 [new Agent/TCP/FullTcp]
set tcp5 [new Agent/TCP/FullTcp]

$tcp0 set fid_ 0
$tcp1 set fid_ 1
$tcp2 set fid_ 2
$tcp3 set fid_ 3
$tcp2 set fid_ 4
$tcp3 set fid_ 5

$ns attach-agent $n0 $tcp0
$ns attach-agent $n1 $tcp1
$ns attach-agent $n1 $tcp2
$ns attach-agent $n2 $tcp3
$ns attach-agent $n3 $tcp4
$ns attach-agent $n1 $tcp5

$ns connect $tcp0 $tcp1
$ns connect $tcp3 $tcp2
$ns connect $tcp4 $tcp5
$tcp0 listen
$tcp1 listen
$tcp2 listen
$tcp3 listen
$tcp4 listen
$tcp5 listen

set rti [new Application/TimeManager]

set debug [new Application/TcpApp $tcp0]
set fedagent0 [new Application/FedTMTar $tcp0]
set fedagent1 [new Application/FedTM/FedAgent $tcp1]
set fedagent2 [new Application/FedTM/FedAgent $tcp2]
set fedagent3 [new Application/FedTMTar $tcp3]
set fedagent4 [new Application/FedTMTar $tcp4]
set fedagent5 [new Application/FedTM/FedAgent $tcp5]

set simobj [new Application/SimObj]
$simobj attachsize $size0
set simobj_1 [new Application/SimObj]
$simobj_1 attachsize $size0

set simobj_s [new Application/TransObj]
set simobj_ls [new Application/TransObj]
set simobj_r [new Application/RecvObj]


$fedagent0 attachsimobj $simobj
$fedagent1 attachsimobj $simobj_s
$fedagent2 attachsimobj $simobj_r
$fedagent4 attachsimobj $simobj_1
$fedagent5 attachsimobj $simobj_ls

$fedagent0 connect $fedagent1
```

```
$fedagent2 connect $fedagent3
$fedagent4 connect $fedagent5
$fedagent0 lookahead 1
$fedagent1 lookahead 1
$fedagent2 lookahead 1
$fedagent3 lookahead 1
$fedagent4 lookahead 1
$fedagent5 lookahead 1

$rti attachfedagent $fedagent1
$rti attachfedagent $fedagent2
$rti attachfedagent $fedagent5

#$rti setcon $fedagent0 $fedagent1

$fedagent3 enable-autofwd
$fedagent0 enable-autofwd
$fedagent4 enable-autofwd
$fedagent0 enable-trace
$fedagent1 enable-trace
$fedagent2 enable-trace
$fedagent3 enable-trace
$fedagent4 enable-trace
$fedagent5 enable-trace


$obj_s subscribe-by $obj_r
$obj_ls subscribe-by $obj_r

Application/TcpApp instproc showmsg {data} {
      global ns
      $ns trace-annotate "$self : $data "
}

proc finish {} {
      global ns nf f namfile
      $ns flush-trace
      close $nf
      close $f

      puts "running nam..."
      exec nam $namfile &
      exit 0
}


$ns at 0.1 "$fedagent0 start"
$ns at 0.1 "$fedagent3 start"
$ns at 0.1 "$fedagent4 start"

$ns at 5.0 "finish"

$ns run
```

#OTcl สคริปต์สำหรับการทดลองเรื่อง การจำลองของ federation ขนาดใหญ่

```
set ns [new Simulator]
set nf [open out.nam w]
set tf [open out.tr w]

$ns namtrace-all $nf
$ns trace-all $tf
set numrout 5
set numcon 5

for {set i 0} {$i < $numrout} {incr i} {
      set r(s$i) [$ns node]
      for {set j 0} {$j < $numrout} {incr j} {
            set confed($i,$j) [$ns node ]
      }
}
set rtinode [$ns node]
set regnode [$ns node]
set routerx [$ns node]
$ns duplex-link $routerx $r(s0) 1000kb 10ms DropTail
$ns duplex-link-op $routerx $r(s0)  orient right
$ns duplex-link $rtinode $routerx 1000kb 10ms DropTail
$ns duplex-link-op $rtinode $routerx orient right
$ns duplex-link $regnode $routerx 1000kb 10ms DropTail
$ns duplex-link-op $regnode $routerx orient down

set tcpReg0 [new Agent/TCP/FullTcp]
set tcpRReg0 [new Agent/TCP/FullTcp]
set fedReg0 [new Application/FedTMTar $tcpReg0]
set fedRReg0 [new Application/FedTM/FedAgent $tcpRReg0]
set simobj [new Application/SimObj]
$simobj attachsize $size0
set simobj_s [new Application/TransObj]
$fedReg0 attachsimobj $simobj
$fedRReg0 attachsimobj $simobj_s

for {set i 0} {$i < [expr $numrout-1]} {incr i} {
      set j [expr $i+1]
      $ns duplex-link $r(s$i) $r(s$j) 2Mb 5ms DropTail
      $ns duplex-link-op $r(s$i) $r(s$j) orient right
}
for {set i 0} {$i < $numcon} {incr i 1} {
      for {set j 0} {$j < $numrout} {incr j} {
            $ns duplex-link $r(s$i) $confed($i,$j) 2Mb 5ms DropTail
            $ns duplex-link-op $r(s$i) $confed($i,$j) orient right

            set tcpRti($i,$j) [new Agent/TCP/FullTcp]
            set tcpC($i,$j) [new Agent/TCP/FullTcp]
            set fedRti($i,$j) [new Application/FedTM/FedAgent
$tcpRti($i,$j)]
            set fedC($i,$j) [new Application/FedTMTar $tcpC($i,$j)]
            $ns attach-agent $rtinode $tcpRti($i,$j)
            $ns attach-agent $confed($i,$j) $tcpC($i,$j)

            set simobj_r($i,$j) [new Application/RecvObj]
            $fedRti($i,$j) attachsimobj $simobj_r($i,$j)
            fedRti($i,$j) lookahead 1
            fedC($i,$j) lookahead 1
            $simobj_s subscribe-by $simobj_r($i,$j)
      }

}


proc finish {} {
      global ns nf tf
      $ns flush-trace
      close $nf
      close $tf
      exec nam out.nam &
      exit 0
}



$ns at 0.1 "$fedReg0 start"
$ns at 0.1 "$fedC(0,0) start"
$ns at 0.1 "$fedC(0,1) start"
$ns at 0.1 "$fedC(0,2) start"
$ns at 0.1 "$fedC(0,3) start"
$ns at 0.1 "$fedC(0,4) start"

$ns at 0.1 "$fedC(1,0) start"
$ns at 0.1 "$fedC(1,1) start"
$ns at 0.1 "$fedC(1,2) start"
$ns at 0.1 "$fedC(1,3) start"
$ns at 0.1 "$fedC(1,4) start"

$ns at 0.1 "$fedC(2,0) start"
$ns at 0.1 "$fedC(2,1) start"
$ns at 0.1 "$fedC(2,2) start"
$ns at 0.1 "$fedC(2,3) start"
$ns at 0.1 "$fedC(2,4) start"

$ns at 0.1 "$fedC(3,0) start"
$ns at 0.1 "$fedC(3,1) start"
$ns at 0.1 "$fedC(3,2) start"
$ns at 0.1 "$fedC(3,3) start"
$ns at 0.1 "$fedC(3,4) start"

$ns at 0.1 "$fedC(4,0) start"
$ns at 0.1 "$fedC(4,1) start"
$ns at 0.1 "$fedC(4,2) start"
$ns at 0.1 "$fedC(4,3) start"
$ns at 0.1 "$fedC(4,4) start"

$ns at 5 "finish"
$ns run
```