BITS Pilani
Dubai Campus

Simulation of Wired Network

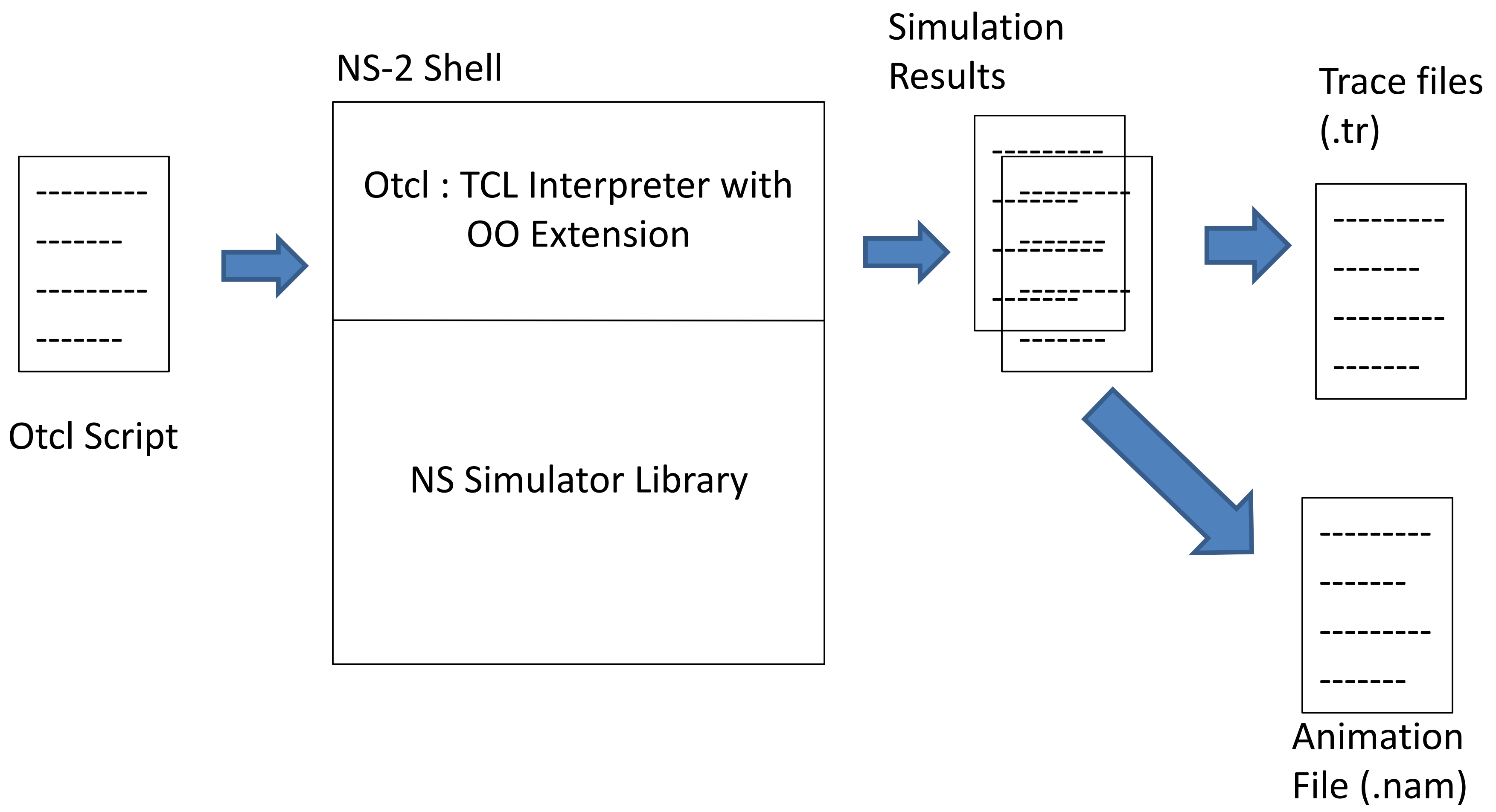
CS F303

Dr. Pranav M. Pawar

Introduction to NS-2

- Object Oriented simulator
- Work at packet level
- Widely used in research community
- Use two languages
  - TCL
  - C++

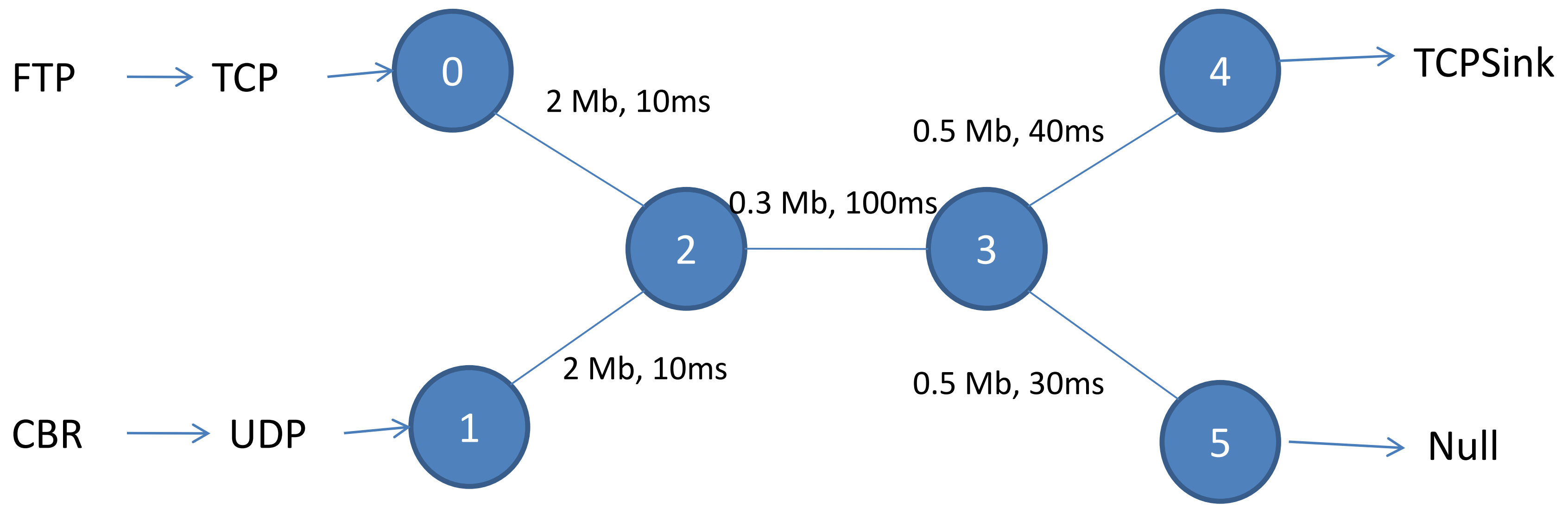
Simulation System Architecture



Running the Simulation Script

- Save the simulation script in specific folder.
- Open the terminal and go up to specific folder.
- Run the simulation script,
  - ns: command to run simulation script.
    - Syntax: ns filename.tcl
      - e.g. ns First_script_wired.tcl
- Run the nam file,
  - nam: command to run animation file
    - Syntax: nam filename.nam
      - e.g. nam s1.nam

Simulation Scenario



Simulation Script

```
#Create Simulator Object (Simulator is class in ns2)
   set ns [new Simulator]
#Define different colors for data flows (for NAM) ($ means
   reference)
   $ns color 1 Blue
   $ns color 2 Red
#Open the Event trace files
   set file1 [open out.tr w]
#trace-all for capturing event trace.
   $ns trace-all $file1
#Open the NAM trace file
   set file2 [open out.nam w]
#namtrace-all for capturing animation details.
   $ns namtrace-all $file2
```

Simulation Script (contd..)

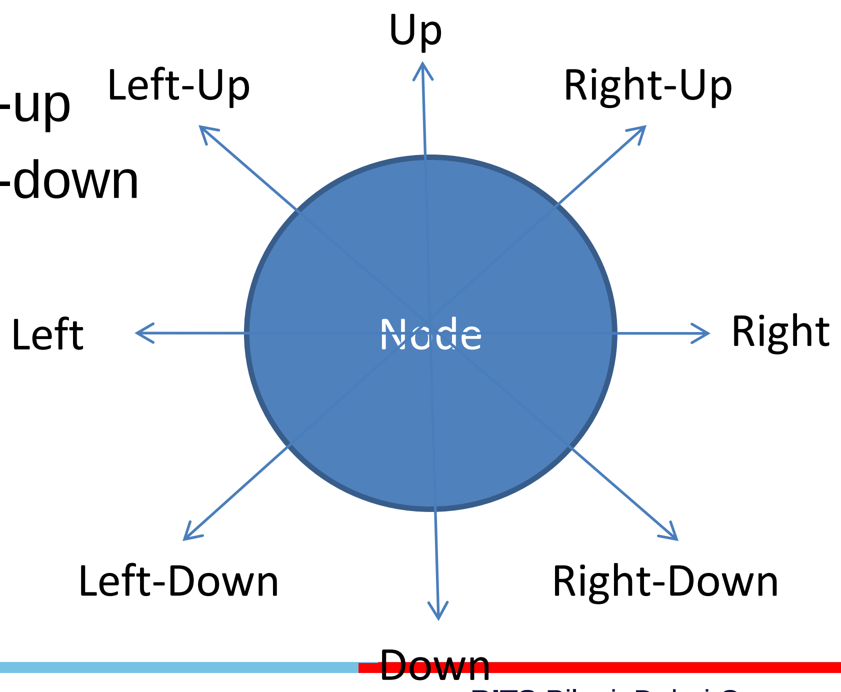
```
#Create six nodes
    set n0 [$ns node]
    set n1 [$ns node]
    set n2 [$ns node]
    set n3 [$ns node]
    set n4 [$ns node]
    set n5 [$ns node]
#Create links between the nodes
#Syntax: $ns duplex-link source destination data-rate
    propagation-delay queue-type
#DropTail class for simple queue.
     $ns duplex-link $n0 $n2 2Mb 10ms DropTail
     $ns duplex-link $n1 $n2 2Mb 10ms DropTail
     $ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
     $ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
     $ns duplex-link $n3 $n4 0.5Mb 40ms DropTail
     $ns duplex-link $n3 $n5 0.5Mb 30ms DropTail
```

Simulation Script (contd..)

#Give node position (for NAM)

#Syntax: $ns duplex-link-op source destination orient orientation-position

```
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down
```



Simulation Script (contd..)

```
#Set Queue Size of link (n2-n3) to 40
#Syntax: $ns queue-limt source destination queue-size
      $ns queue-limit $n2 $n3 40
#Setup a TCP connection (Source agent: TCP, Destination agent: TCPSink)
#Agent is superclass and TCP is subclass.
     set tcp [new Agent/TCP]
     $ns attach-agent $n0 $tcp
#Agent is superclass and TCPSink is subclass.
     set sink [new Agent/TCPSink]
     $ns attach-agent $n4 $sink
     $ns connect $tcp $sink
     $tcp set fid_ 1
     $tcp set packetSize_ 552
#Setup a FTP over TCP connection
#Application is superclass and FTP is subclass.
     set ftp [new Application/FTP]
     $ftp attach-agent $tcp
```

Simulation Script (contd..)

```
#Setup a UDP connection (Source agent: UDP, Destination
    agent: Null)
#Agent is superclass and UDP is subclass.
    set udp [new Agent/UDP]
    $ns attach-agent $n1 $udp
#Agent is superclass and Null is subclass.
    set null [new Agent/Null]
    $ns attach-agent $n5 $null
    $ns connect $udp $null
    $udp set fid_ 2
#Setup a CBR over UDP connection
#CBR(Constant Bit Rate) is subclass of Traffic and Traffic is
    subclass of Application
    set cbr [new Application/Traffic/CBR]
    $cbr attach-agent $udp
    $cbr set packet_size_ 1000
```

Simulation Script (contd..)

```
# Scheduling the event
    $ns at 0.1 "$cbr start"
    $ns at 1.0 "$ftp start"
    $ns at 624.0 "$ftp stop"
    $ns at 624.5 "$cbr stop“
# Call finish procedure
   $ns at 625.0 “finish”
# Run the simulation
   $ns run
```

```
#Define a 'finish' procedure
proc finish {} {
        global ns file1 file2
        $ns flush-trace
        close $file1
        close $file2
        exit 0  }
```

Wired trace file format

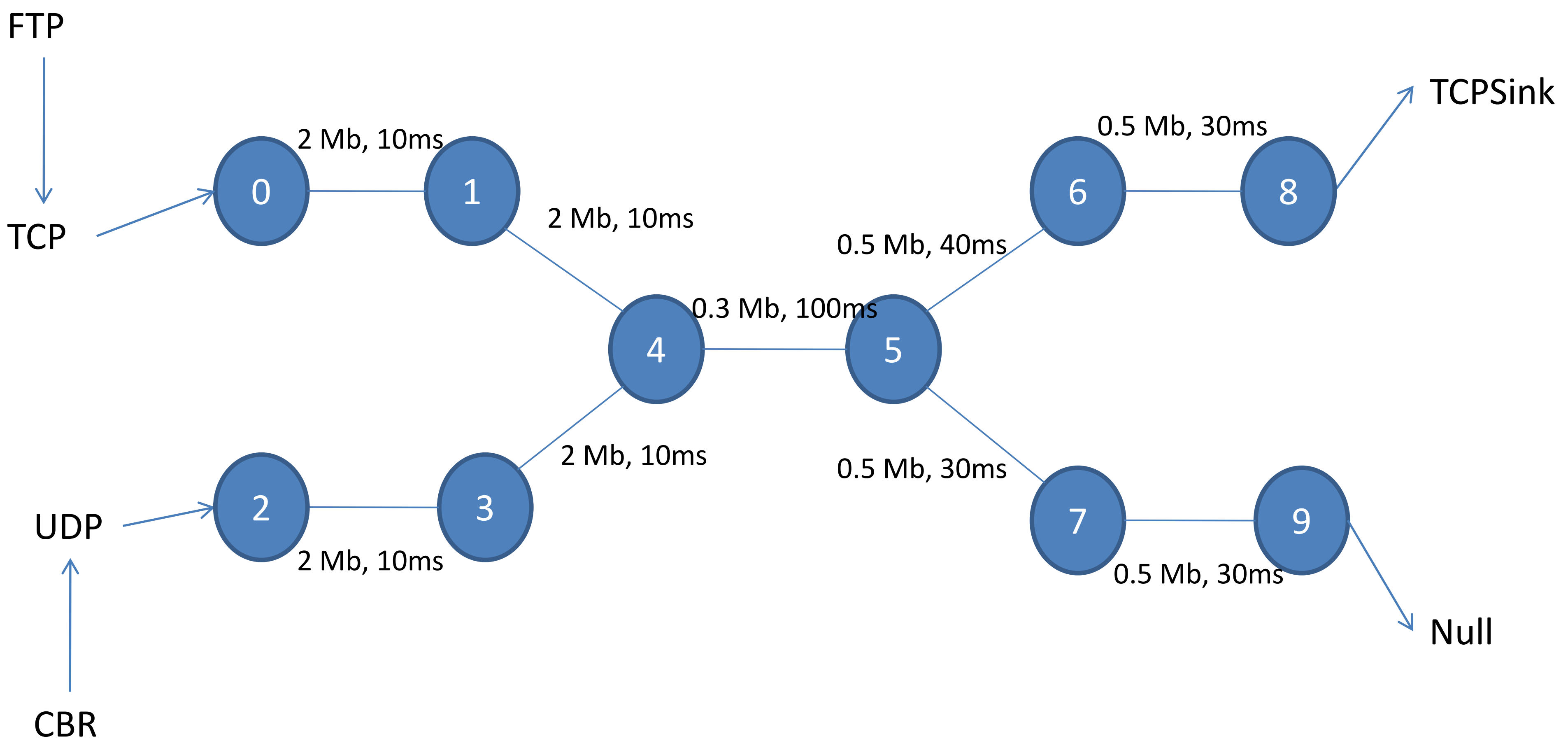
event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dst addr	seq num	pkt id

```
r : receive (at to_node)
+ : enqueue (at queue)
- : dequeue (at queue)
d : drop    (at queue)
```

```
src_addr : node.port (3.0)
dst_addr : node.port (0.0)
```

Event	Time	From-Node	To-Node	Pkt-Type	Pkt-Size	Flags	Fid	Src-addr	Dest-addr	Seq-num	Pkt-id
-	1.06	0	2	tcp	1040	-------	1	0.0	3.0	2	124
r	1.07	1	2	cbr	1000	-------	2	1.0	3.1	120	122
+	1.07	2	3	cbr	1000	-------	2	1.0	3.1	120	122
d	1.07	2	3	cbr	1000	-------	2	1.0	3.1	120	122

Self Practice Example



Appendix

- Steps for making your machine ready for NS-2
  - Install Ubuntu linux on your machine (Using Virtual Box).
    - Steps for installing ubuntu (recommended version: Ubuntu 20.04.2):
      https://itsfoss.com/install-linux-in-virtualbox
  - Login (using user name and password which you selected during installation) to Ubuntu Linux
  - Open terminal (Right click on desktop screen and select open terminal).
  - NS-2 installation
    - Type following command on terminal and follow instructions
      - sudo apt-get install ns2
    - Install nam
      - sudo apt-get install nam

Sources

- https://www.isi.edu/nsnam/ns/tutorial/
- https://www.geeksforgeeks.org/basics-of-ns2-and-otcltcl-script/
- https://www.isi.edu/nsnam/ns/
- Steps for installing ubuntu (recommended version: Ubuntu 20.04.2):

  https://itsfoss.com/install-linux-in-virtualbox/

innovate achieve lead

BITS Pilani
Dubai Campus

Thank You!