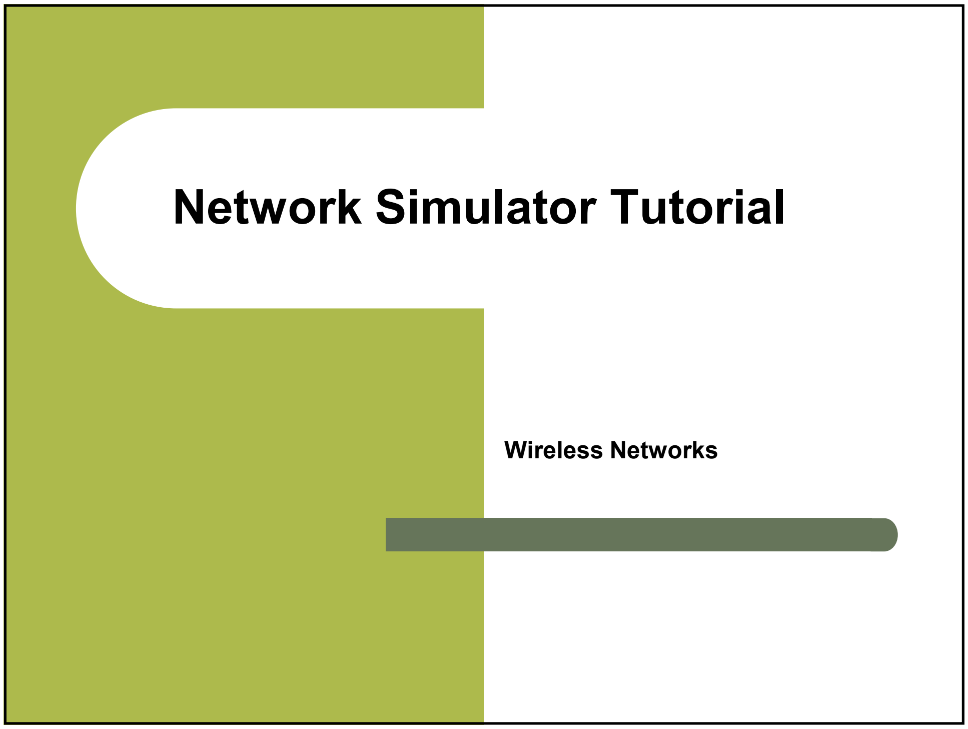
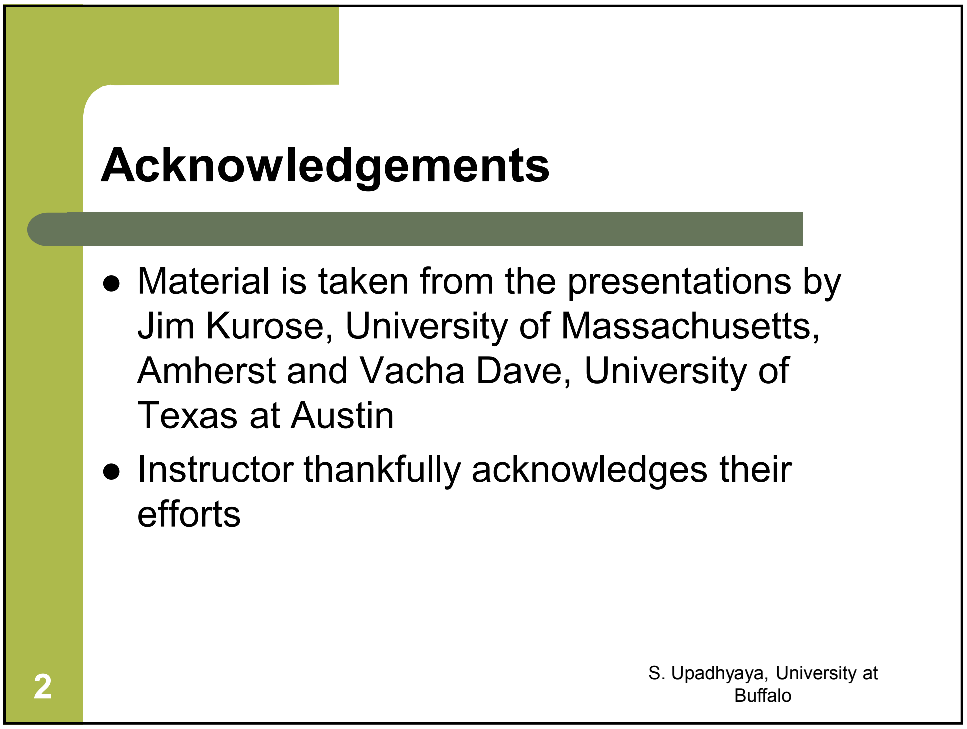


Network Simulator Tutorial

Wireless Networks



Acknowledgements

- Material is taken from the presentations by Jim Kurose, University of Massachusetts, Amherst and Vacha Dave, University of Texas at Austin
- Instructor thankfully acknowledges their efforts

2

S. Upadhyaya, University at Buffalo

Network Simulation

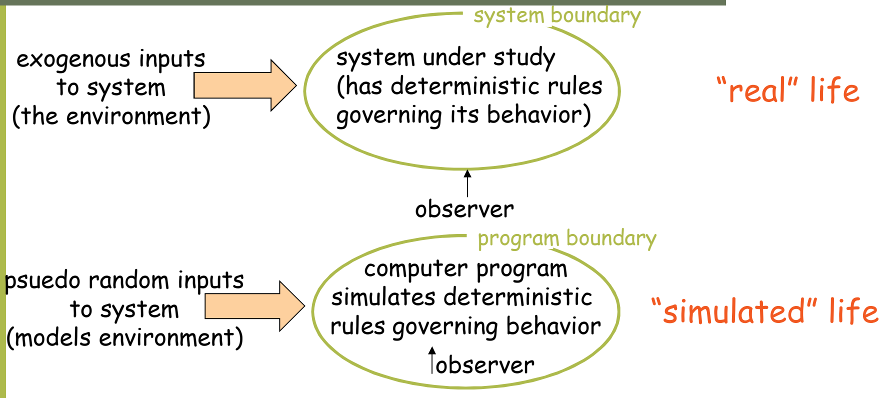
Motivation:

- Learn fundamentals of evaluating network performance via simulation

Overview:

- fundamentals of discrete event simulation
- ns-2 simulation

What is simulation?



system boundary

exogenous inputs to system (the environment)

system under study (has deterministic rules governing its behavior)

"real" life

observer

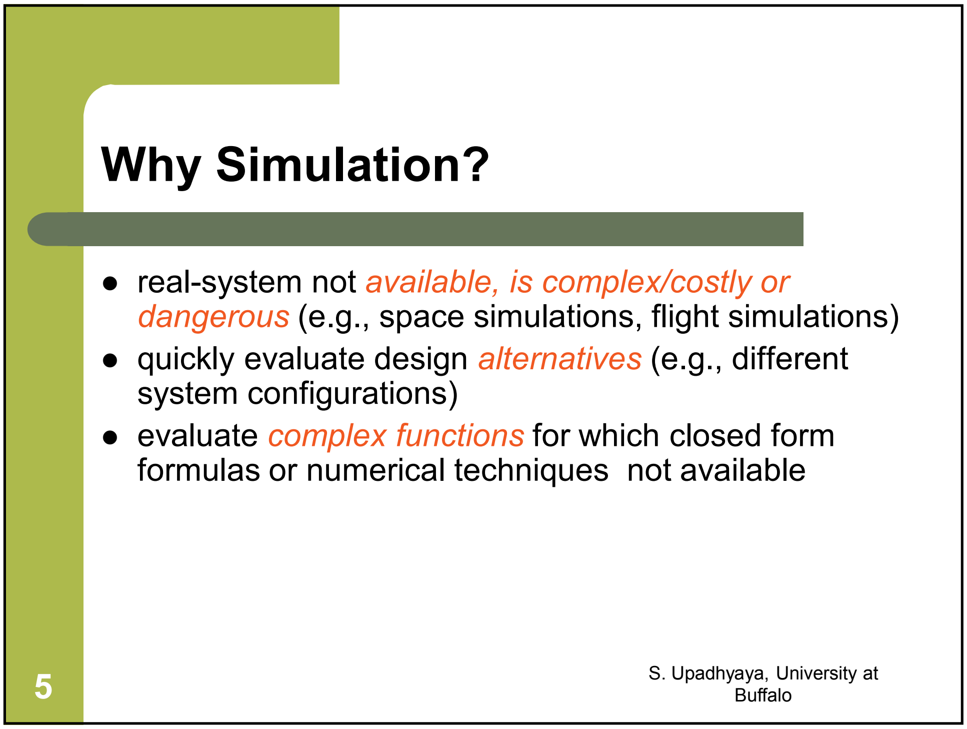
program boundary

psuedo random inputs to system (models environment)

computer program simulates deterministic rules governing behavior

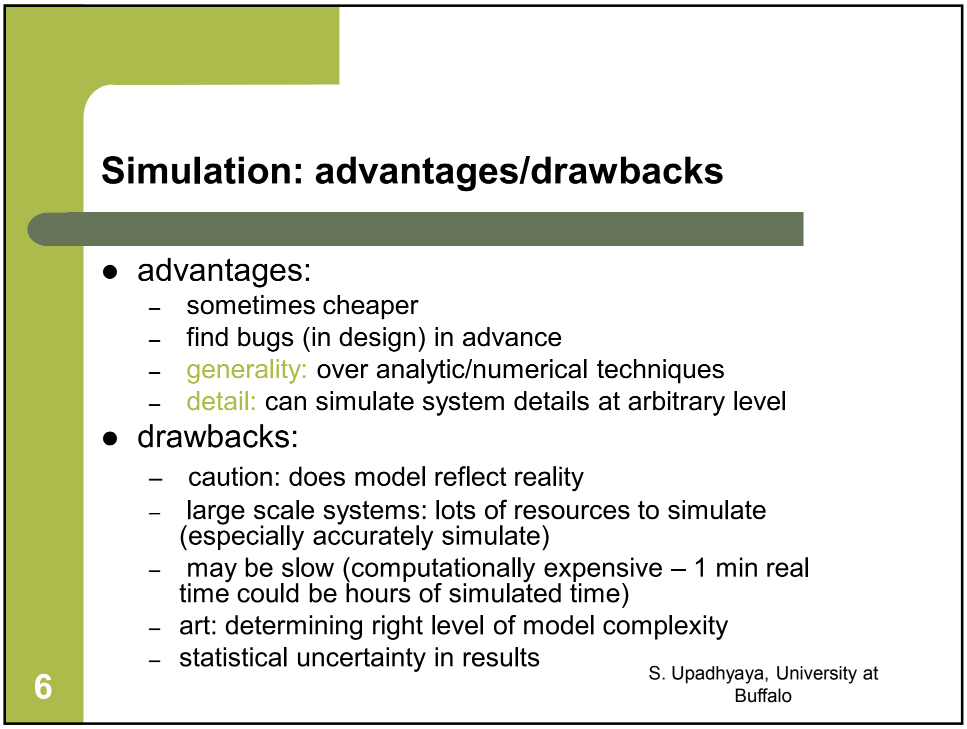
observer

"simulated" life



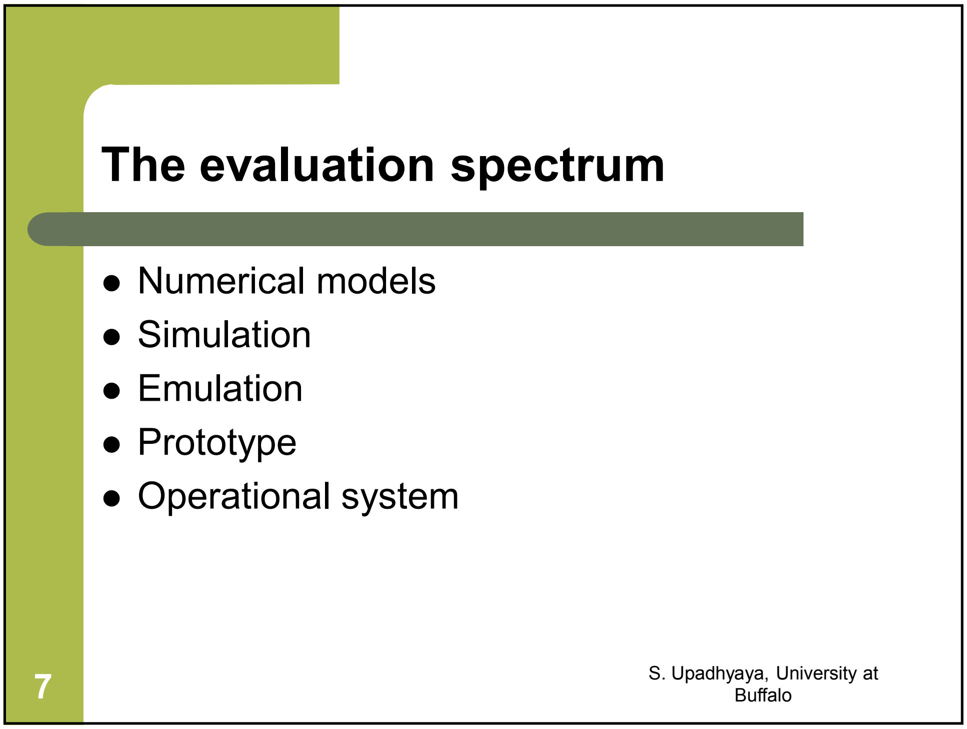
Why Simulation?

- real-system not *available, is complex/costly or dangerous* (e.g., space simulations, flight simulations)
- quickly evaluate design *alternatives* (e.g., different system configurations)
- evaluate *complex functions* for which closed form formulas or numerical techniques not available



Simulation: advantages/drawbacks

- advantages:
  - sometimes cheaper
  - find bugs (in design) in advance
  - generality: over analytic/numerical techniques
  - detail: can simulate system details at arbitrary level
- drawbacks:
  - caution: does model reflect reality
  - large scale systems: lots of resources to simulate (especially accurately simulate)
  - may be slow (computationally expensive – 1 min real time could be hours of simulated time)
  - art: determining right level of model complexity
  - statistical uncertainty in results

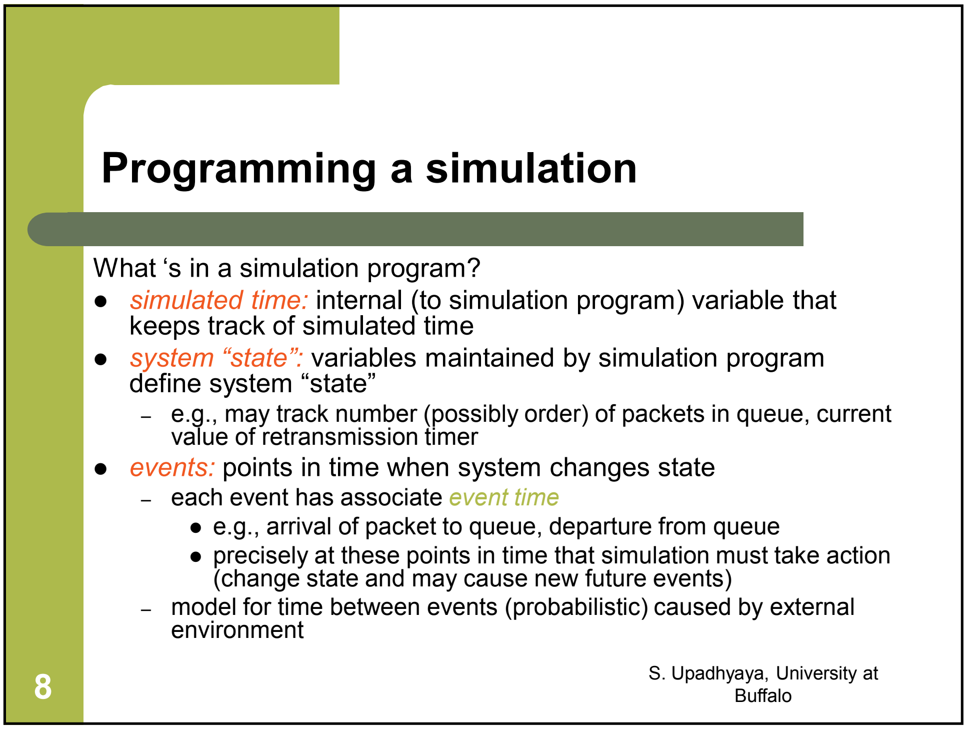


The evaluation spectrum

- Numerical models
- Simulation
- Emulation
- Prototype
- Operational system

7

S. Upadhyaya, University at Buffalo



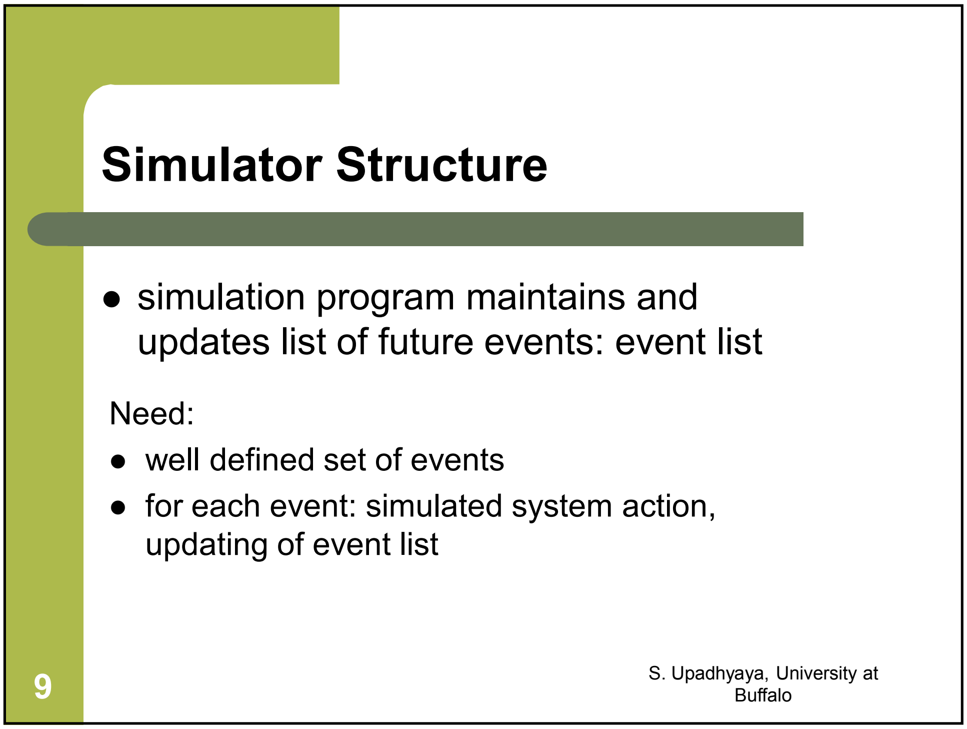
Programming a simulation

What 's in a simulation program?

- *simulated time:* internal (to simulation program) variable that keeps track of simulated time
- *system "state":* variables maintained by simulation program define system "state"
  - e.g., may track number (possibly order) of packets in queue, current value of retransmission timer
- *events:* points in time when system changes state
  - each event has associate *event time*
    - e.g., arrival of packet to queue, departure from queue
    - precisely at these points in time that simulation must take action (change state and may cause new future events)
  - model for time between events (probabilistic) caused by external environment

8

S. Upadhyaya, University at Buffalo



Simulator Structure

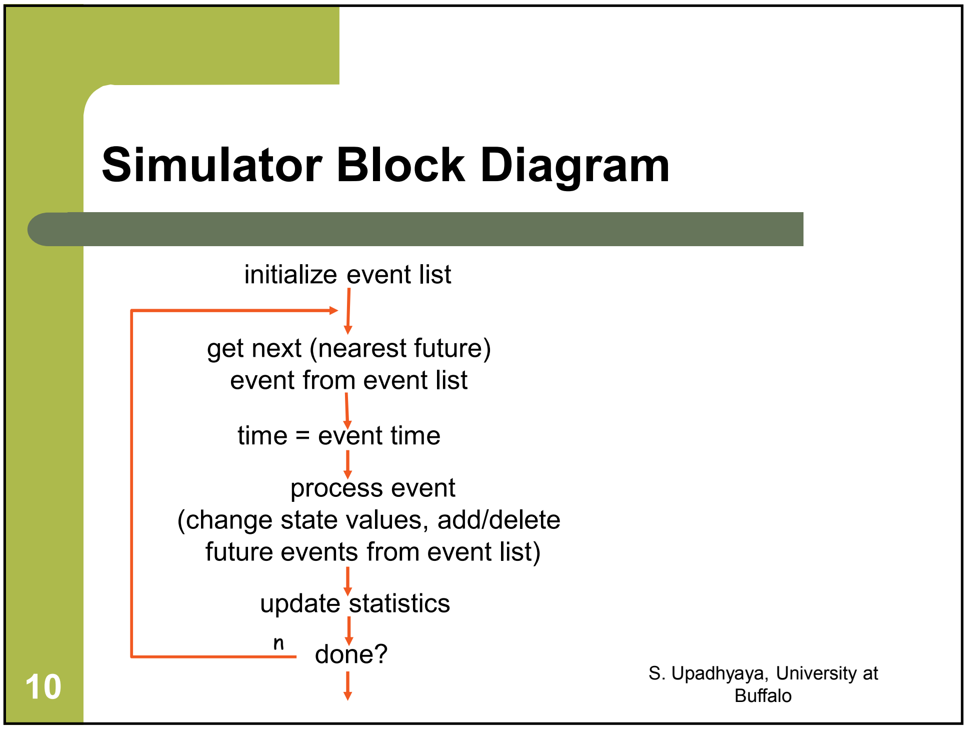
- simulation program maintains and updates list of future events: event list

Need:

- well defined set of events
- for each event: simulated system action, updating of event list

9

S. Upadhyaya, University at Buffalo

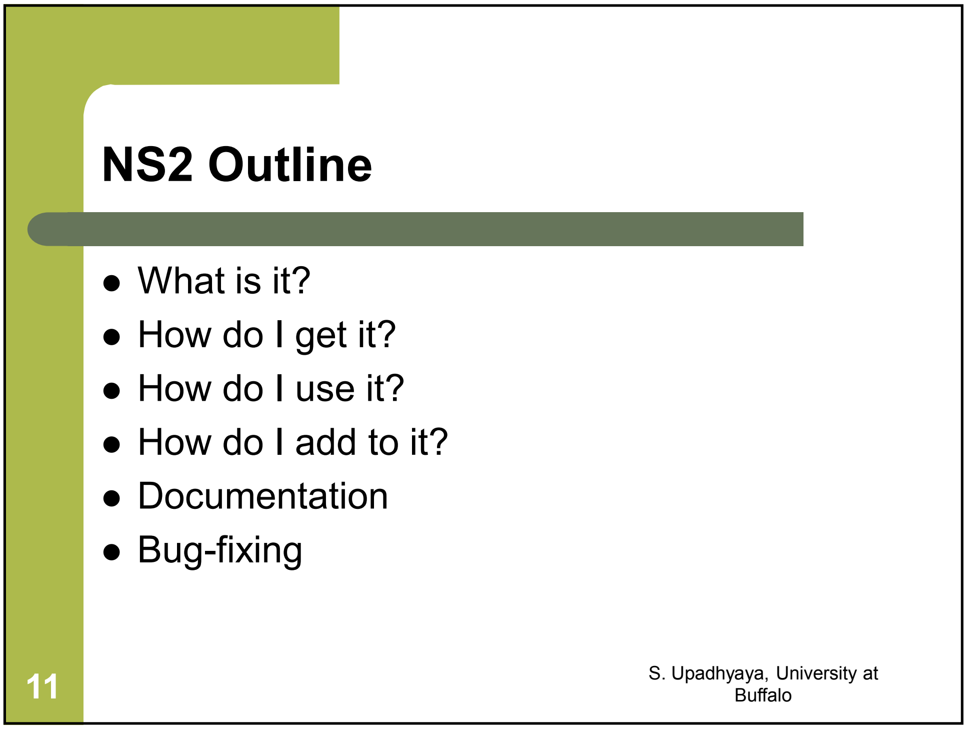


Simulator Block Diagram

initialize event list
get next (nearest future) event from event list
time = event time
process event (change state values, add/delete future events from event list)
update statistics
n
done?

10

S. Upadhyaya, University at Buffalo

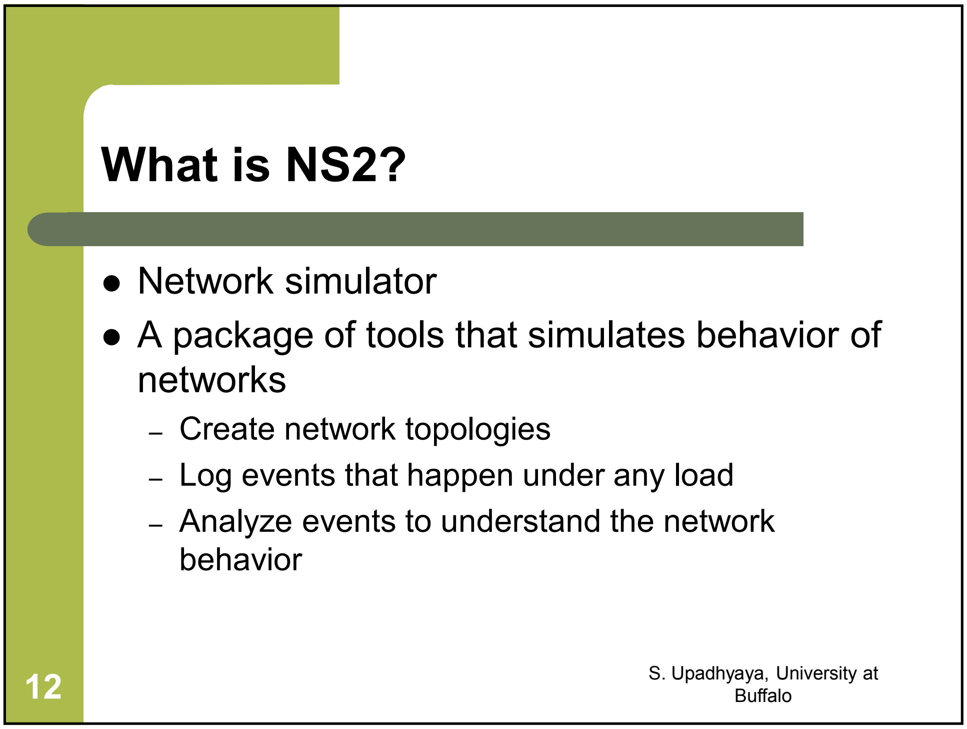


NS2 Outline

- What is it?
- How do I get it?
- How do I use it?
- How do I add to it?
- Documentation
- Bug-fixing

11

S. Upadhyaya, University at Buffalo

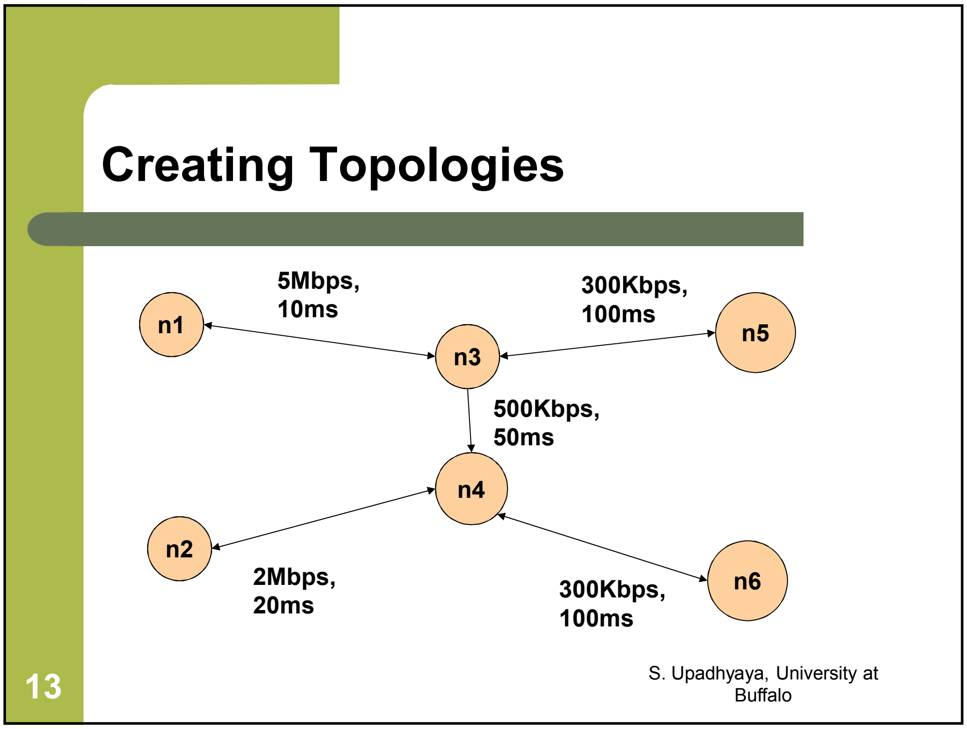


What is NS2?

- Network simulator
- A package of tools that simulates behavior of networks
  - Create network topologies
  - Log events that happen under any load
  - Analyze events to understand the network behavior

12

S. Upadhyaya, University at Buffalo



Creating Topologies

n1

5Mbps,
10ms

n3

300Kbps,
100ms

n5

500Kbps,
50ms

n4

n2

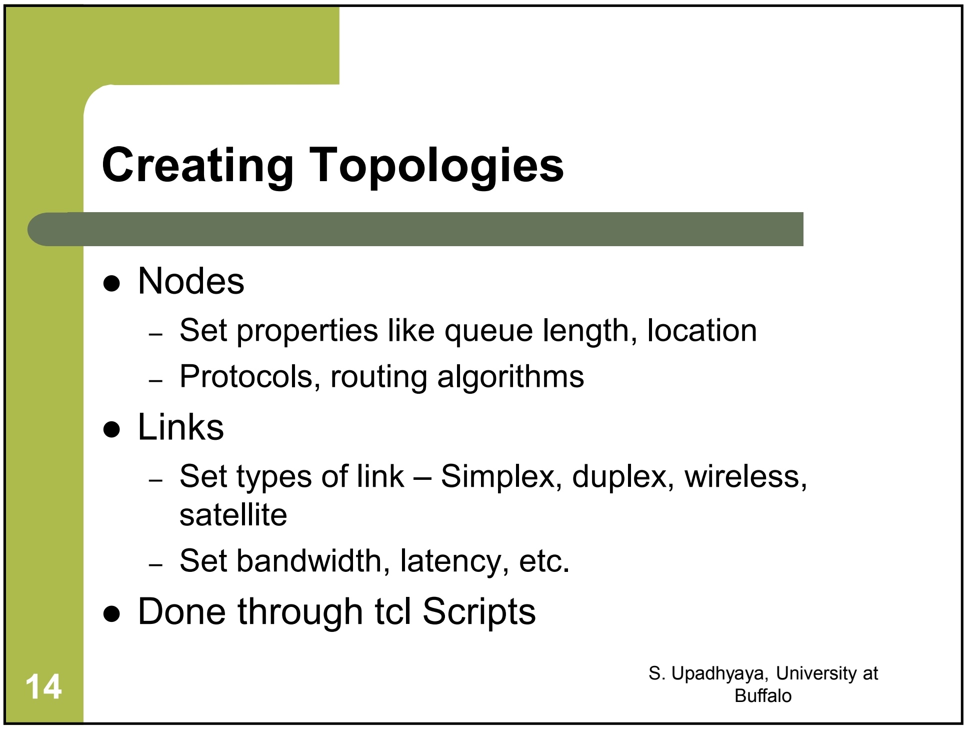
2Mbps,
20ms

300Kbps,
100ms

n6

13

S. Upadhyaya, University at
Buffalo



Creating Topologies

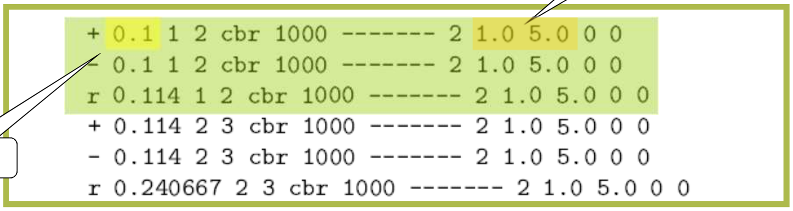
- Nodes
  - Set properties like queue length, location
  - Protocols, routing algorithms
- Links
  - Set types of link – Simplex, duplex, wireless, satellite
  - Set bandwidth, latency, etc.
- Done through tcl Scripts

14

S. Upadhyaya, University at
Buffalo

Observing Network Behavior

- Observe behavior by tracing “event
  - Eg. packet received, packet drop etc



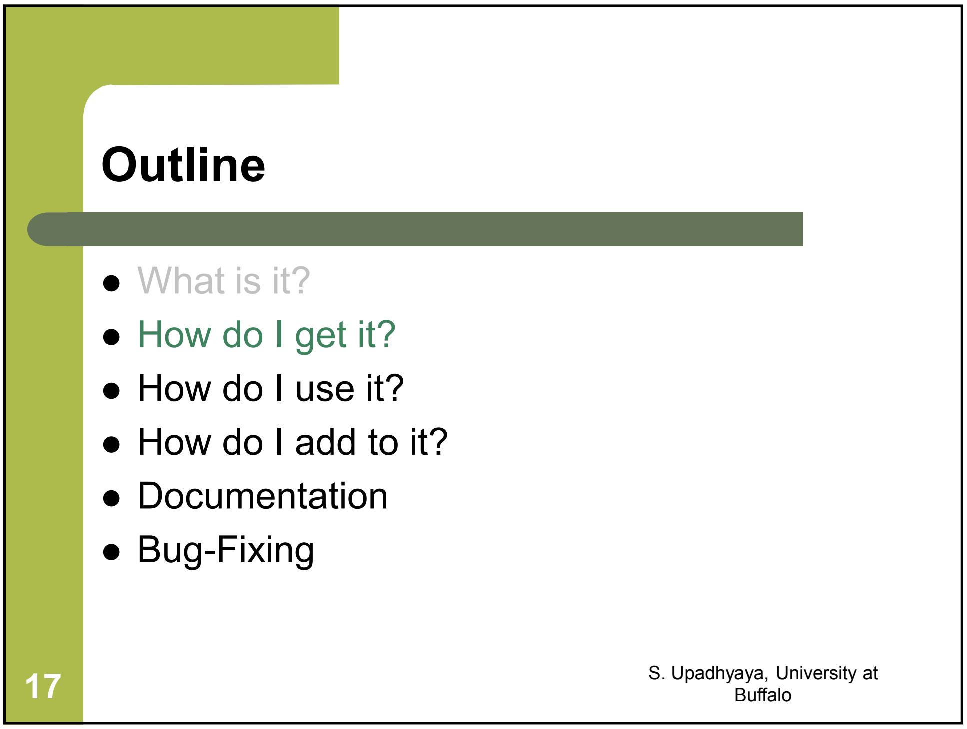
```
+ 0.1 1 2 cbr 1000 ------- 2 1.0 5.0 0 0
- 0.1 1 2 cbr 1000 ------- 2 1.0 5.0 0 0
r 0.114 1 2 cbr 1000 ------- 2 1.0 5.0 0 0
+ 0.114 2 3 cbr 1000 ------- 2 1.0 5.0 0 0
- 0.114 2 3 cbr 1000 ------- 2 1.0 5.0 0 0
r 0.240667 2 3 cbr 1000 ------- 2 1.0 5.0 0 0
```

Src Dst IP Address, Port

time

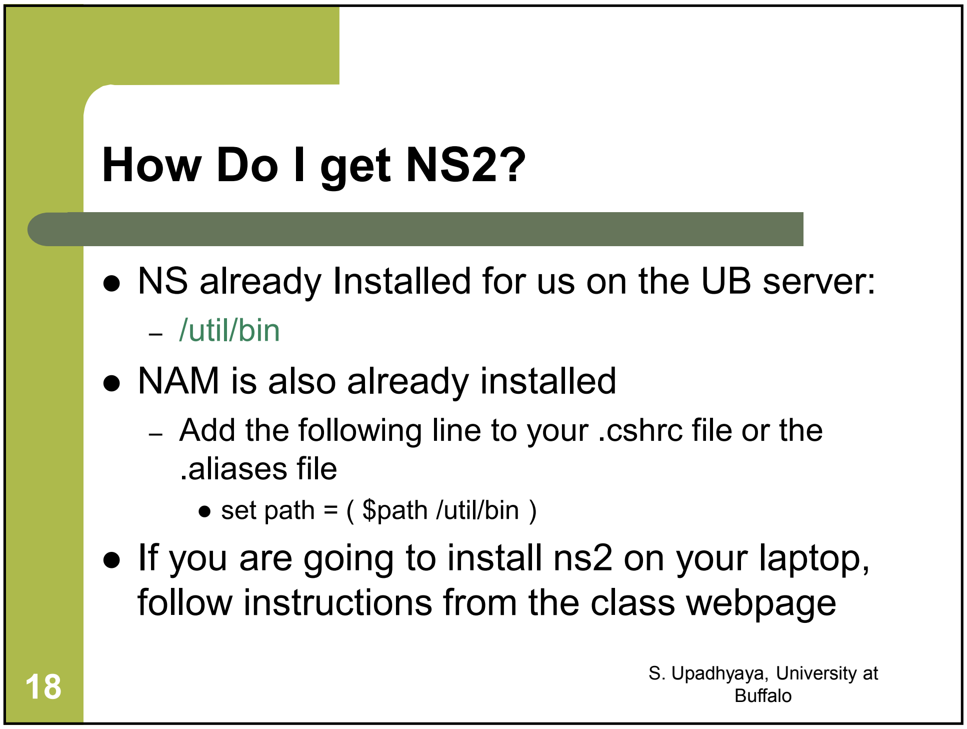
Observing Network Behavior

- NAM:
  - Network Animator
  - A visual aid showing how packets flow along the network
- We’ll see a demo..



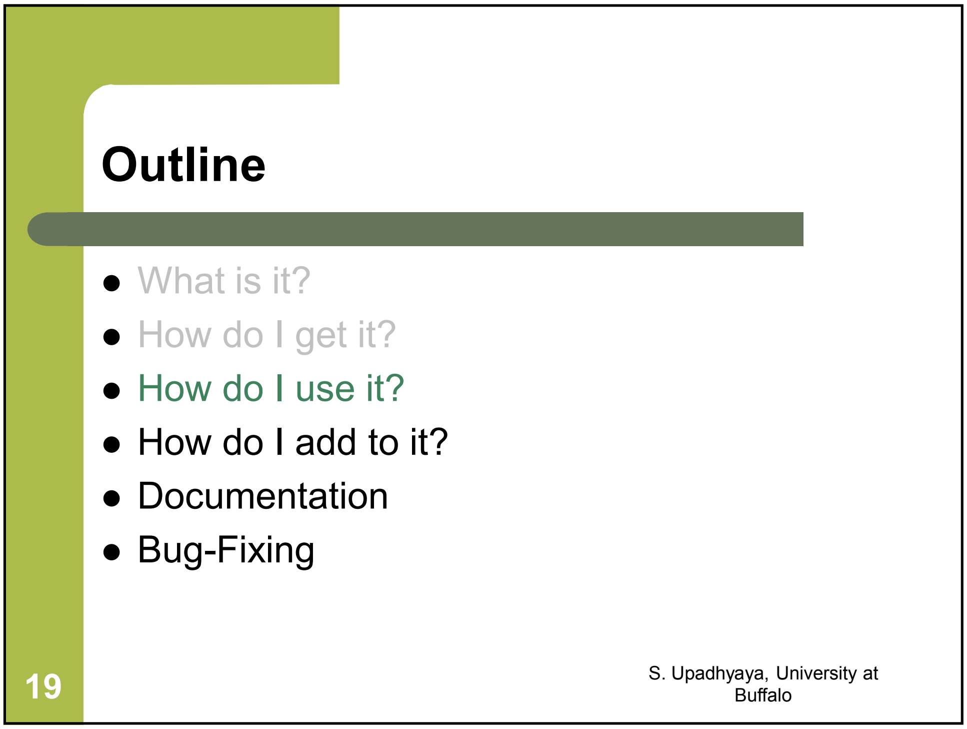
Outline

- What is it?
- How do I get it?
- How do I use it?
- How do I add to it?
- Documentation
- Bug-Fixing



How Do I get NS2?

- NS already Installed for us on the UB server:
  - /util/bin
- NAM is also already installed
  - Add the following line to your .cshrc file or the .aliases file
    - set path = ($path /util/bin)
- If you are going to install ns2 on your laptop, follow instructions from the class webpage

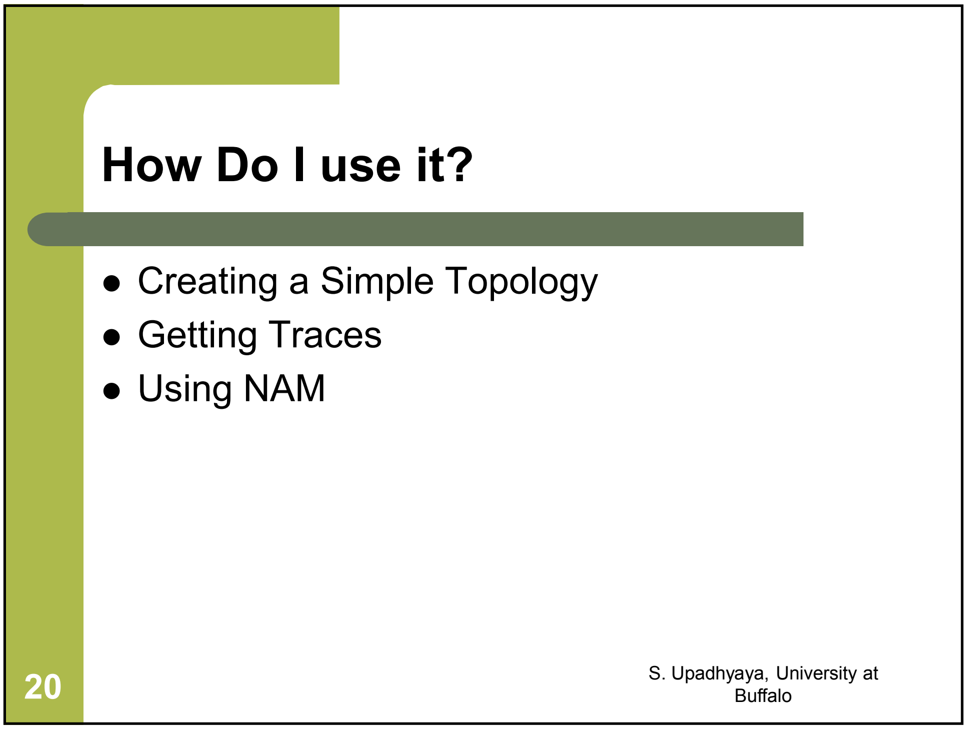


Outline

- What is it?
- How do I get it?
- How do I use it?
- How do I add to it?
- Documentation
- Bug-Fixing

19

S. Upadhyaya, University at Buffalo

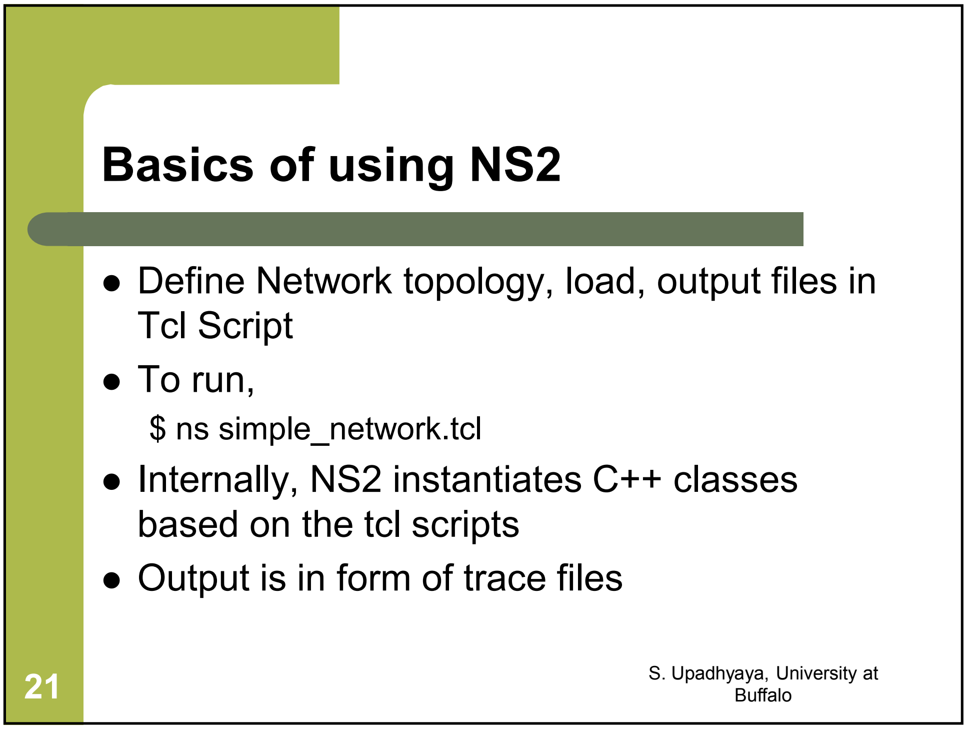


How Do I use it?

- Creating a Simple Topology
- Getting Traces
- Using NAM

20

S. Upadhyaya, University at Buffalo



Basics of using NS2

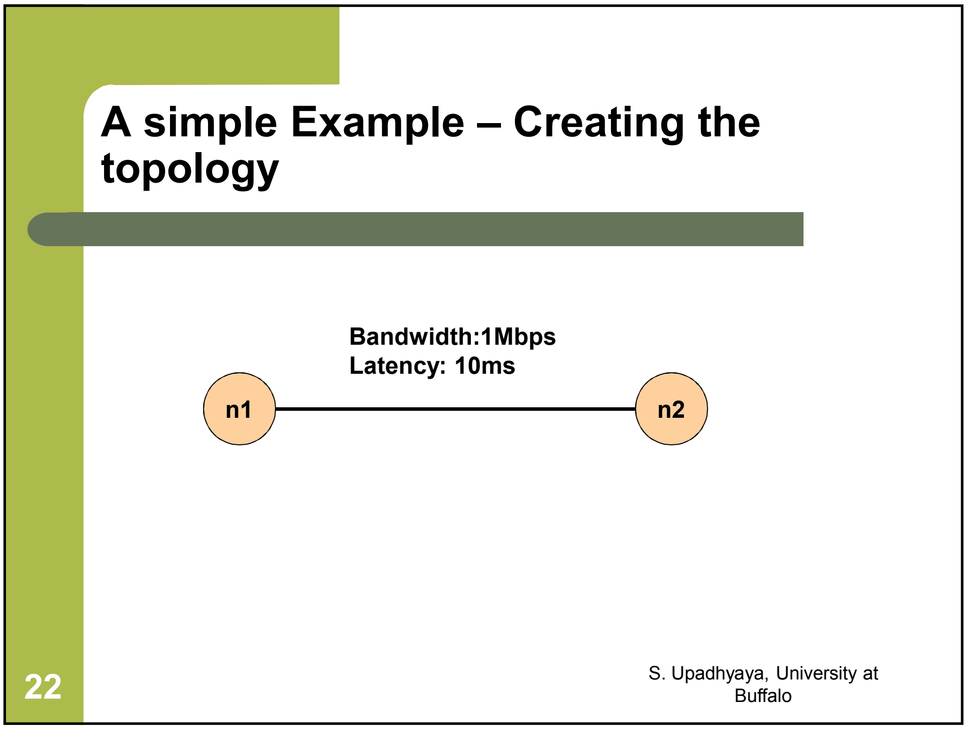
- Define Network topology, load, output files in Tcl Script
- To run,

```
$ ns simple_network.tcl
```

- Internally, NS2 instantiates C++ classes based on the tcl scripts
- Output is in form of trace files

21

S. Upadhyaya, University at Buffalo

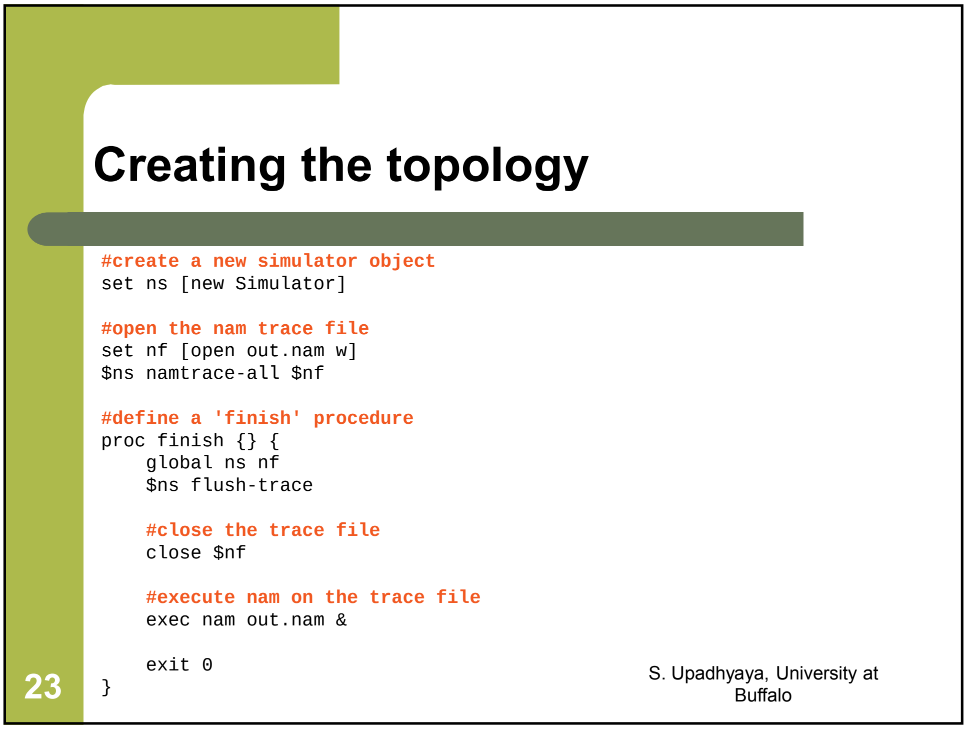


A simple Example – Creating the topology

Bandwidth:1Mbps
Latency: 10ms

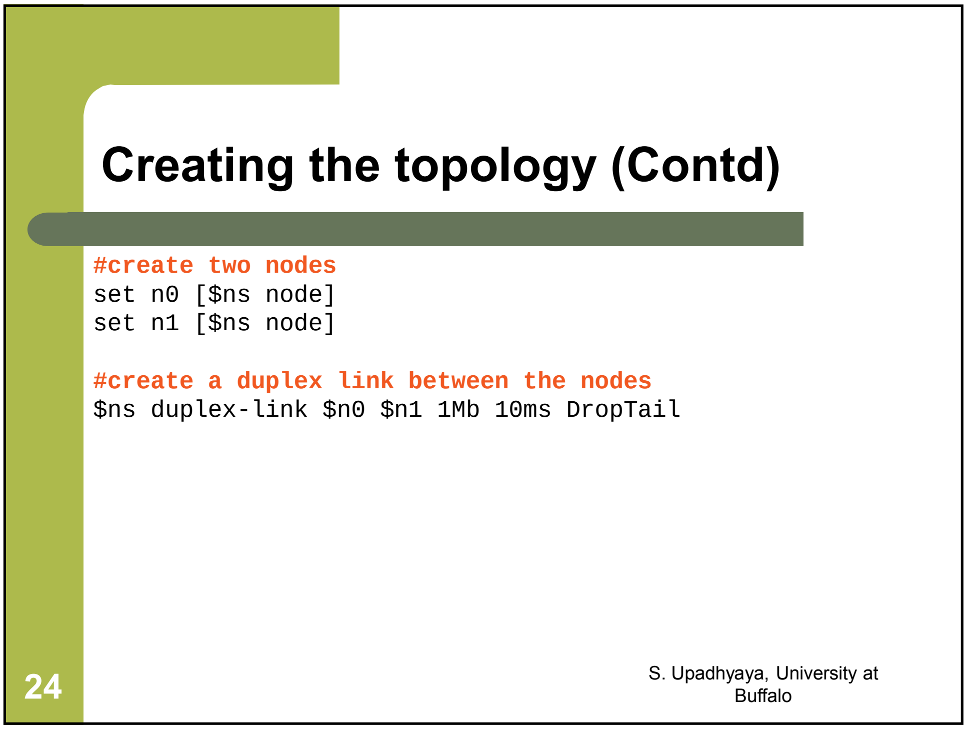
22

S. Upadhyaya, University at Buffalo



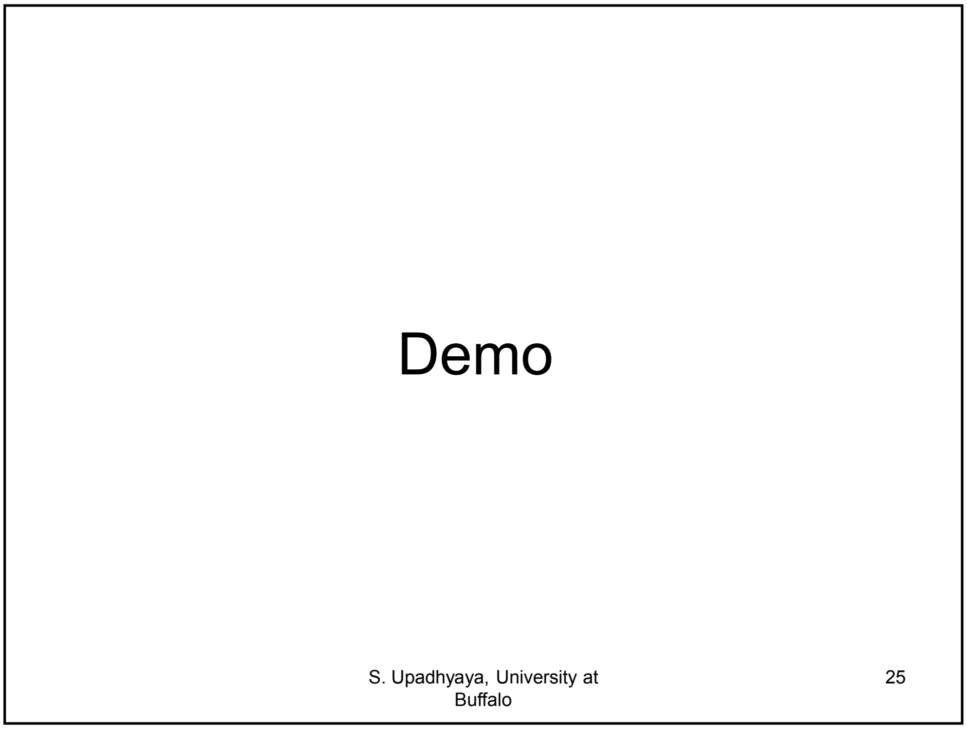
Creating the topology

```
#create a new simulator object
set ns [new Simulator]

#open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace

    #close the trace file
    close $nf

    #execute nam on the trace file
    exec nam out.nam &

    exit 0
}
```

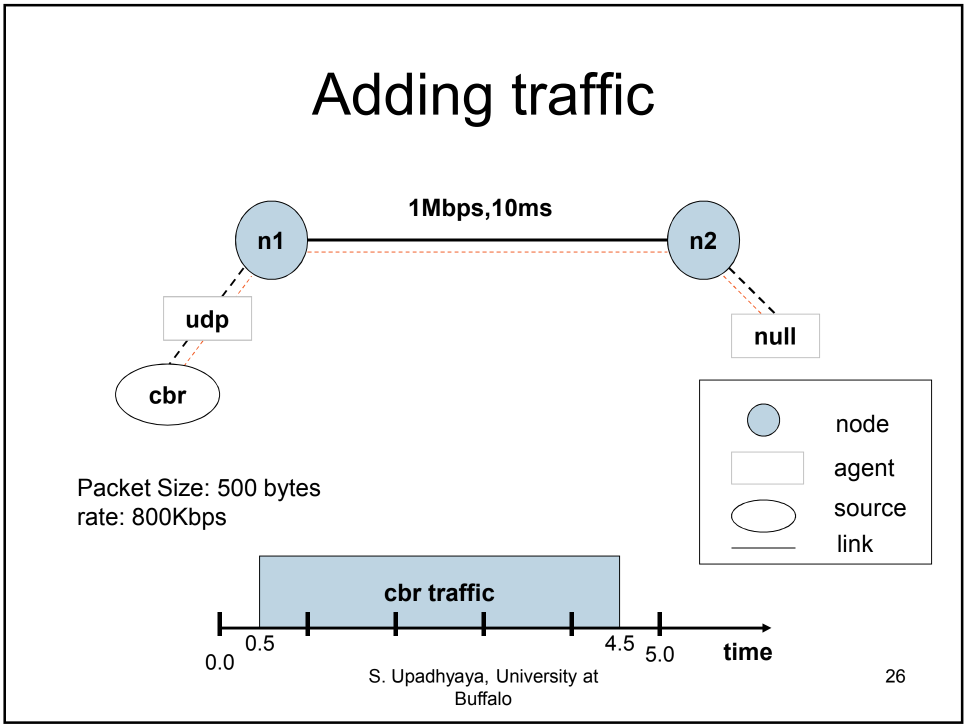


Creating the topology (Contd)

```
#create two nodes
set n0 [$ns node]
set n1 [$ns node]

#create a duplex link between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

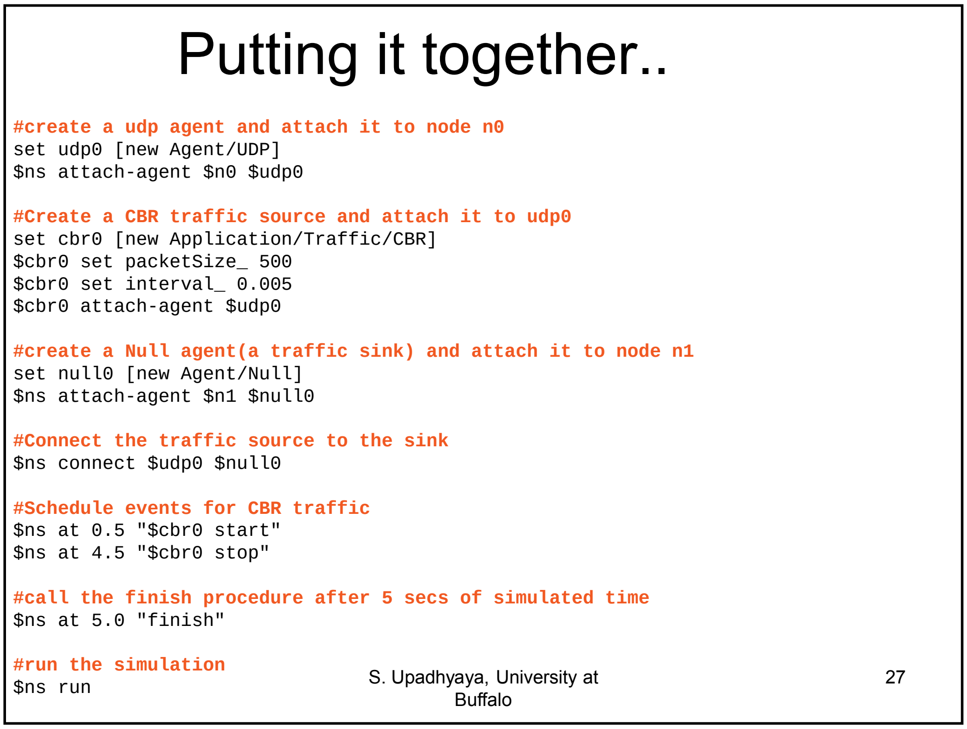


Demo



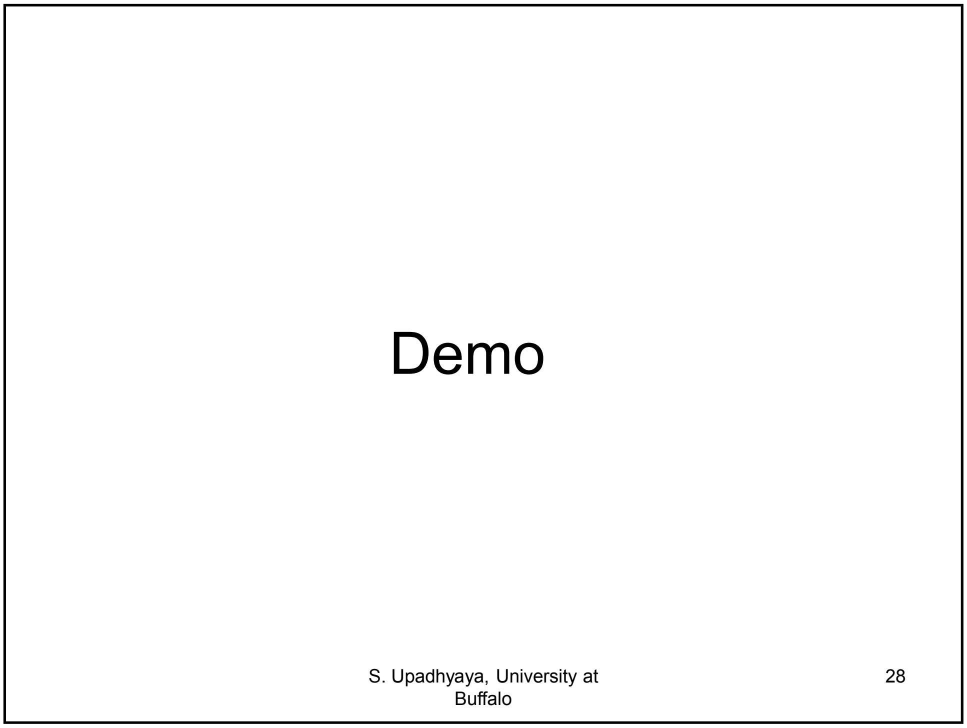
Adding traffic

Packet Size: 500 bytes
rate: 800Kbps



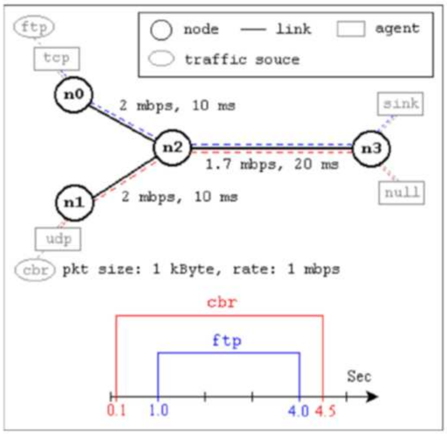
Putting it together..

```
#create a udp agent and attach it to node n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

#Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

#create a Null agent(a traffic sink) and attach it to node n1
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0

#Connect the traffic source to the sink
$ns connect $udp0 $null0

#Schedule events for CBR traffic
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"

#call the finish procedure after 5 secs of simulated time
$ns at 5.0 "finish"

#run the simulation
$ns run
```



Demo

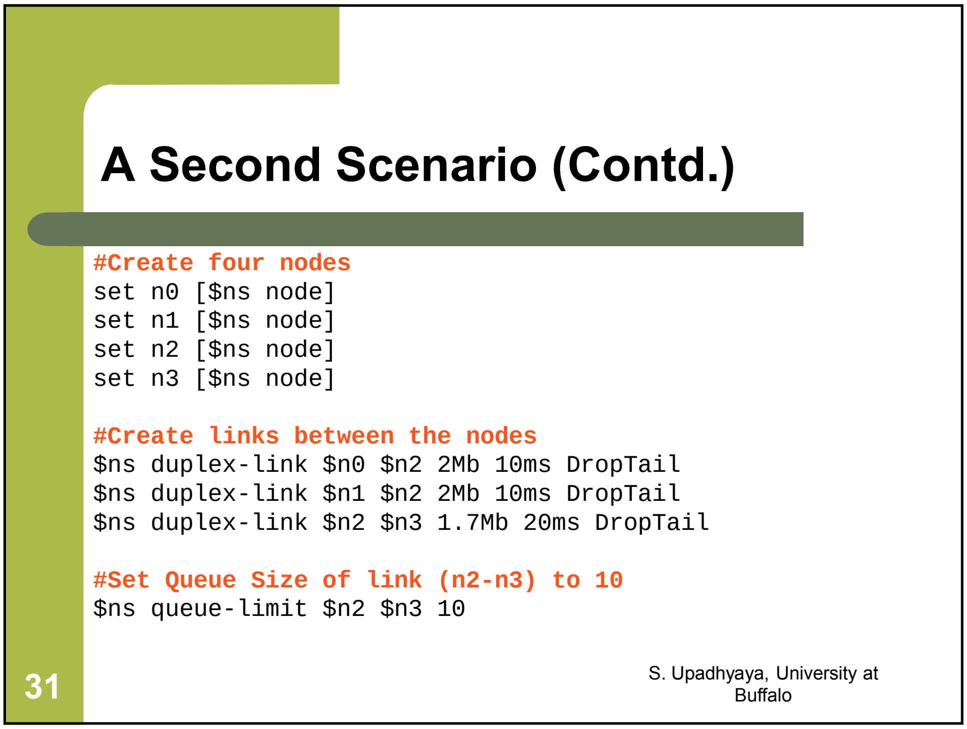
A second Scenario * (from NS by Example)



Taken from NS by Example by Jae Chung **and** Mark Claypool

A second Example (From NS by Example)

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
        global ns nf
        $ns flush-trace
        #Close the NAM trace file
        close $nf
        #Execute NAM on the trace file
        exec nam out.nam &
        exit 0
}
```

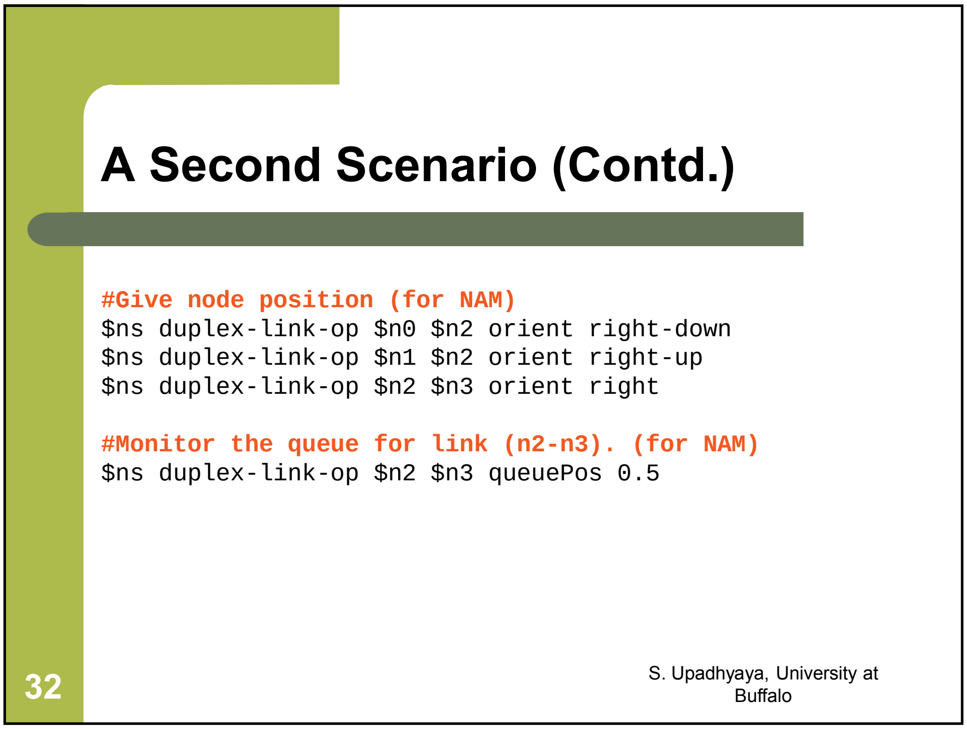


A Second Scenario (Contd.)

```
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10
```

31

S. Upadhyaya, University at Buffalo

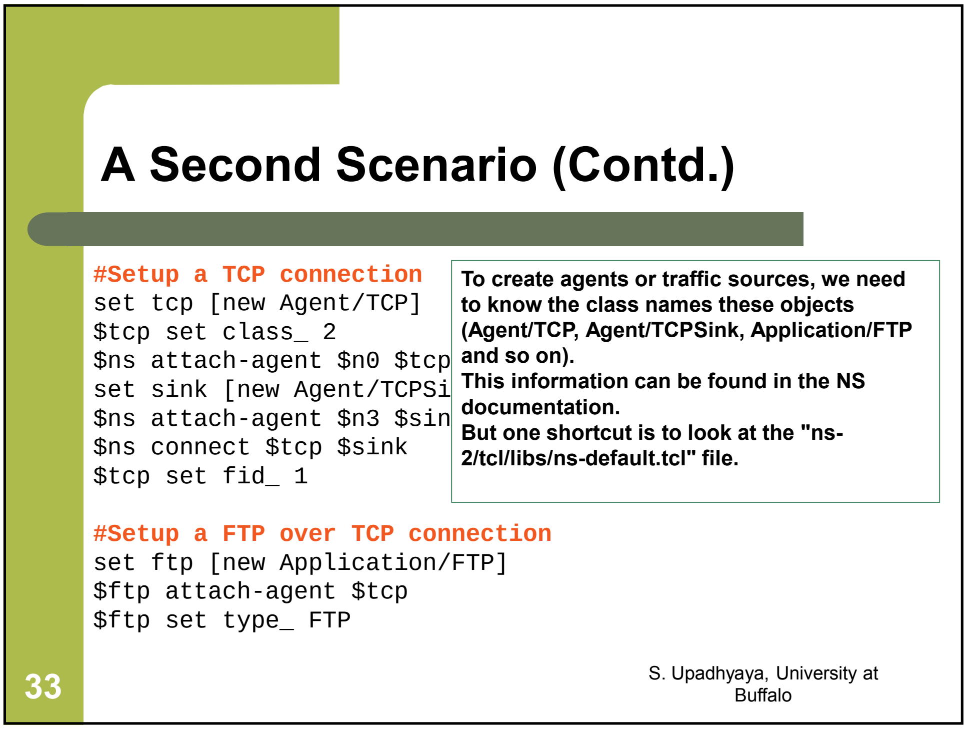


A Second Scenario (Contd.)

```
#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

32

S. Upadhyaya, University at Buffalo



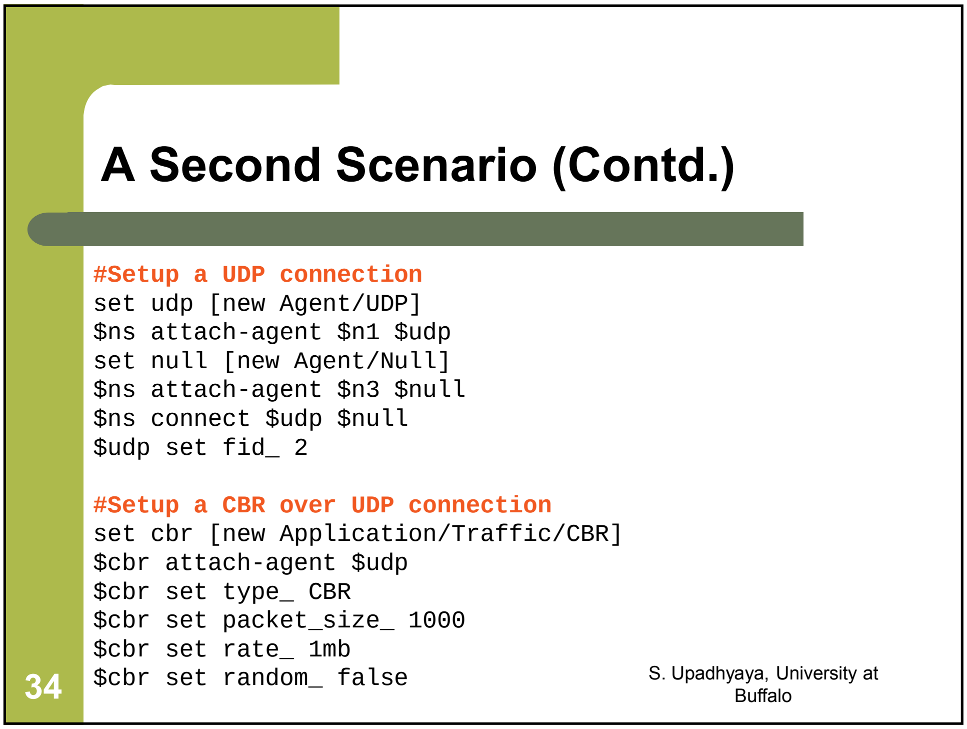
A Second Scenario (Contd.)

```
#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSi
$ns attach-agent $n3 $sin
$ns connect $tcp $sink
$tcp set fid_ 1

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
```

To create agents or traffic sources, we need to know the class names these objects (Agent/TCP, Agent/TCPSink, Application/FTP and so on).
This information can be found in the NS documentation.
But one shortcut is to look at the "ns-2/tcl/libs/ns-default.tcl" file.

33

S. Upadhyaya, University at Buffalo

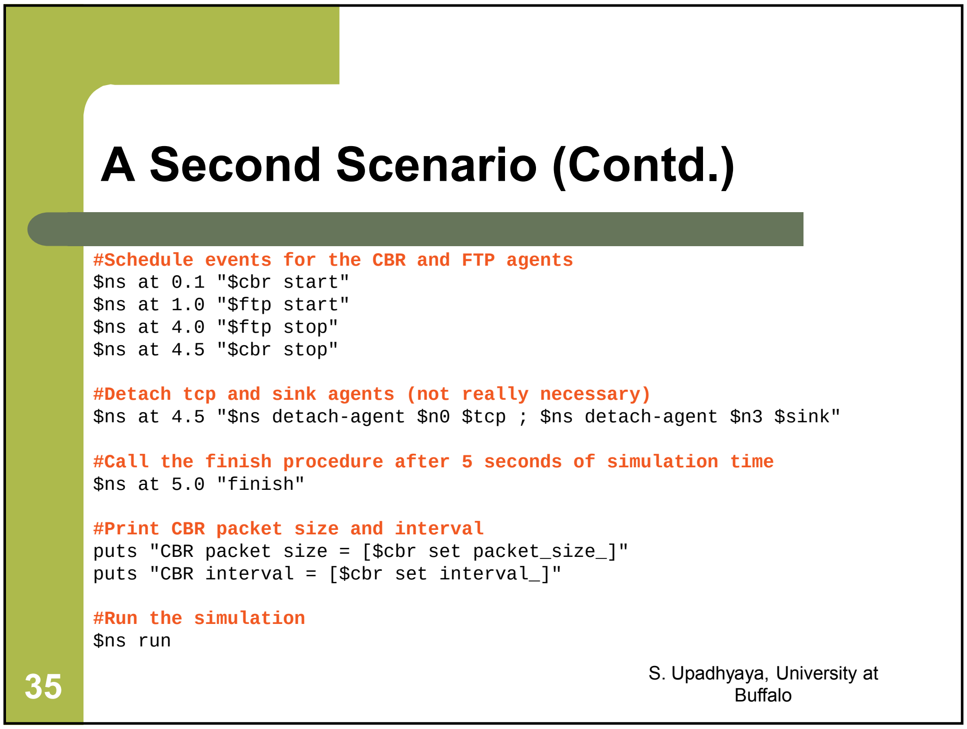


A Second Scenario (Contd.)

```
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false
```

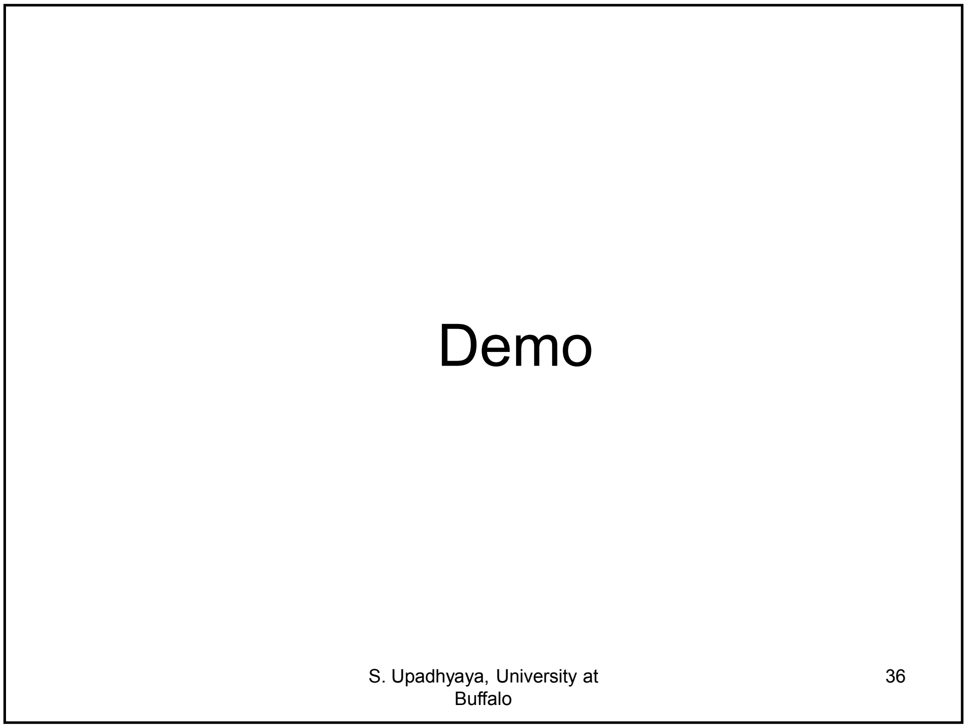
34

S. Upadhyaya, University at Buffalo



A Second Scenario (Contd.)

```
#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation
$ns run
```



Demo

Outline

- What is it?
- How do I get it?
- How do I use it?
- How do I add to it?
- Documentation
- Bug-Fixing

How can I add to NS2?

- Adding Protocols to NS2 is possible
  - Need to create the C++ class
  - Need to create the OTcl Linkage
- More info at:
  - http://www.isi.edu/nsnam/ns/tutorial/index.html
  - Tutorial about how to add a simple protocol to NS2

Outline

- What is it?
- How do I get it?
- How do I use it?
- How do I add to it?
- Documentation
- Bug-fixing

Documentation – NS2 Documentation

- NS2 Manual
  - Information about Otcl interpreter, C++ class hierarchy, parameters for various protocols
  - http://www.isi.edu/nsnam/ns/doc/index.html
  - Very detailed, useful when looking for something specific, like:
    - What are the shadowing models available for wireless? How do I select them?
    - How do I make my routing strategy to be Distance Vector routing?

Documentation – NS2 documentation

- NS2 Tutorial by Marc Greis
  - http://www.isi.edu/nsnam/ns/tutorial/index.html
  - Good starting point for understanding the overall structure of NS2
  - Examples:
    - What is the relation between c++ classes and Otcl classes?
    - basic info on instantiating NS2 instance, tcl scripting

41

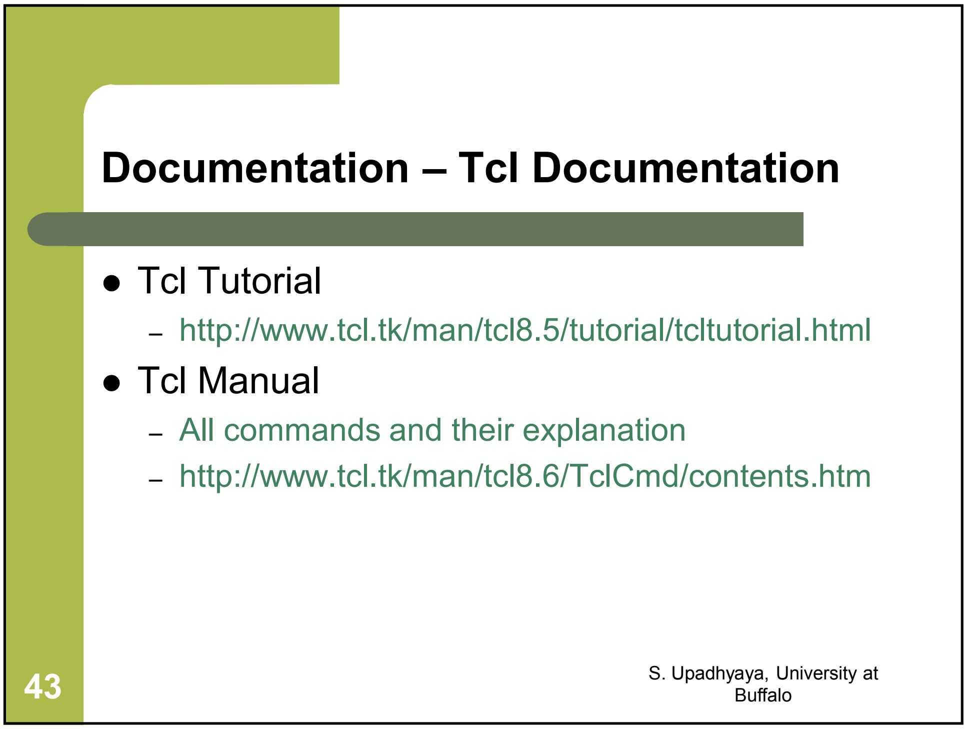
S. Upadhyaya, University at Buffalo

Documentation – NS2 Documentation

- NS2 for beginners
  - http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf
  - More detailed than Marc Greis' Tutorial
  - More info on getting it up and running – rather than internals
  - Examples:
    - What does each line of a tcl script do?
    - Most common examples of trace formats that are useful

42

S. Upadhyaya, University at Buffalo

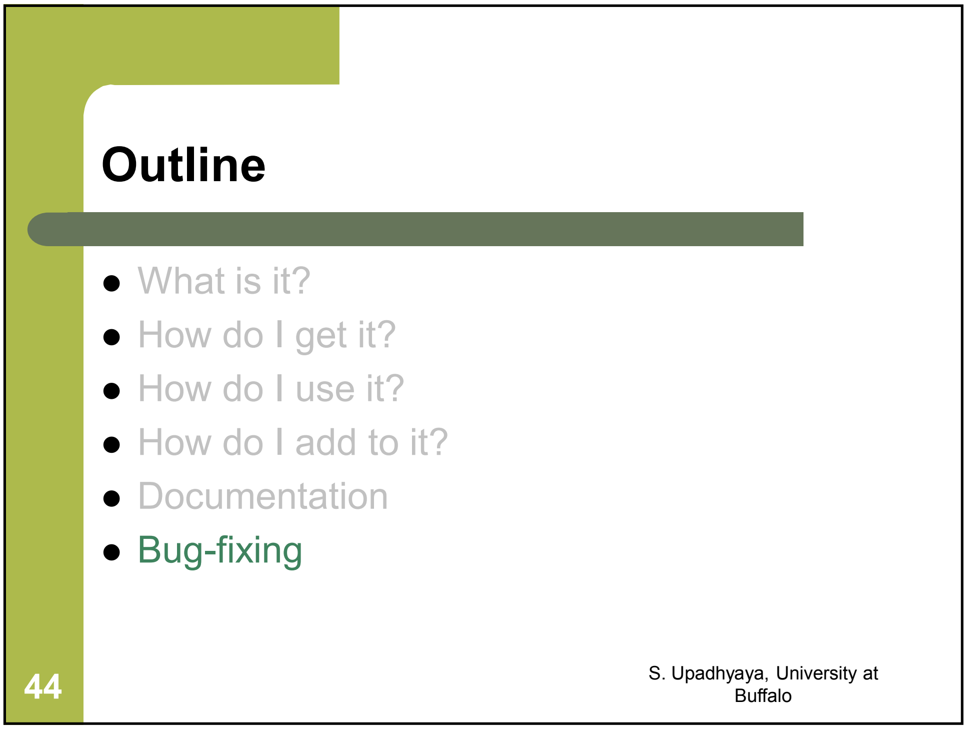


Documentation – Tcl Documentation

- Tcl Tutorial
  - http://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html
- Tcl Manual
  - All commands and their explanation
  - http://www.tcl.tk/man/tcl8.6/TclCmd/contents.htm

43

S. Upadhyaya, University at Buffalo

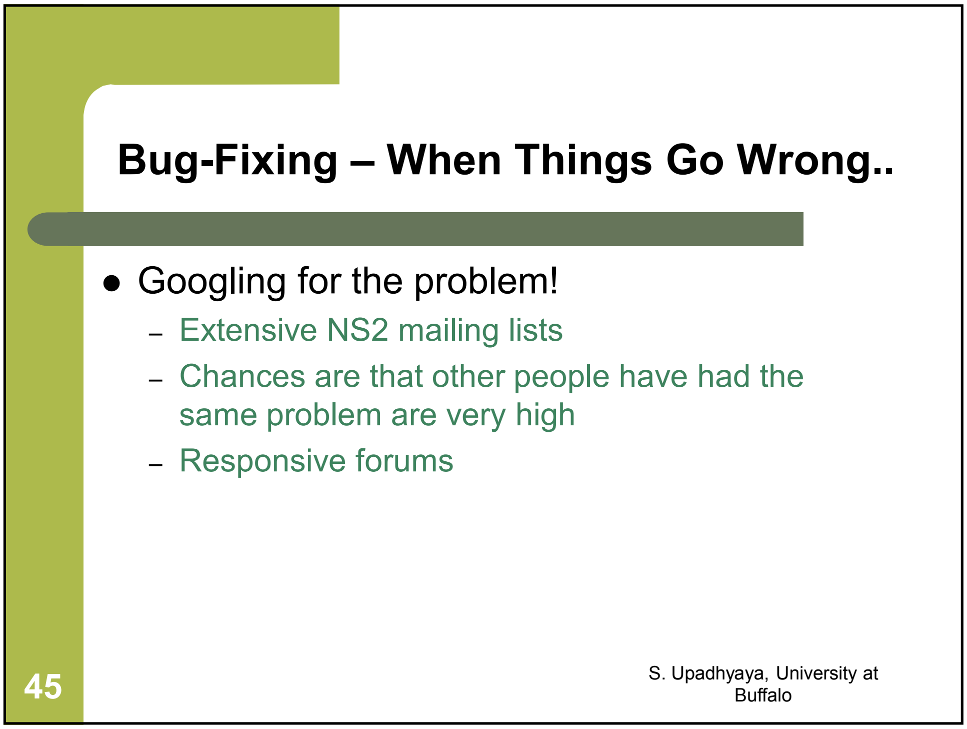


Outline

- What is it?
- How do I get it?
- How do I use it?
- How do I add to it?
- Documentation
- Bug-fixing

44

S. Upadhyaya, University at Buffalo

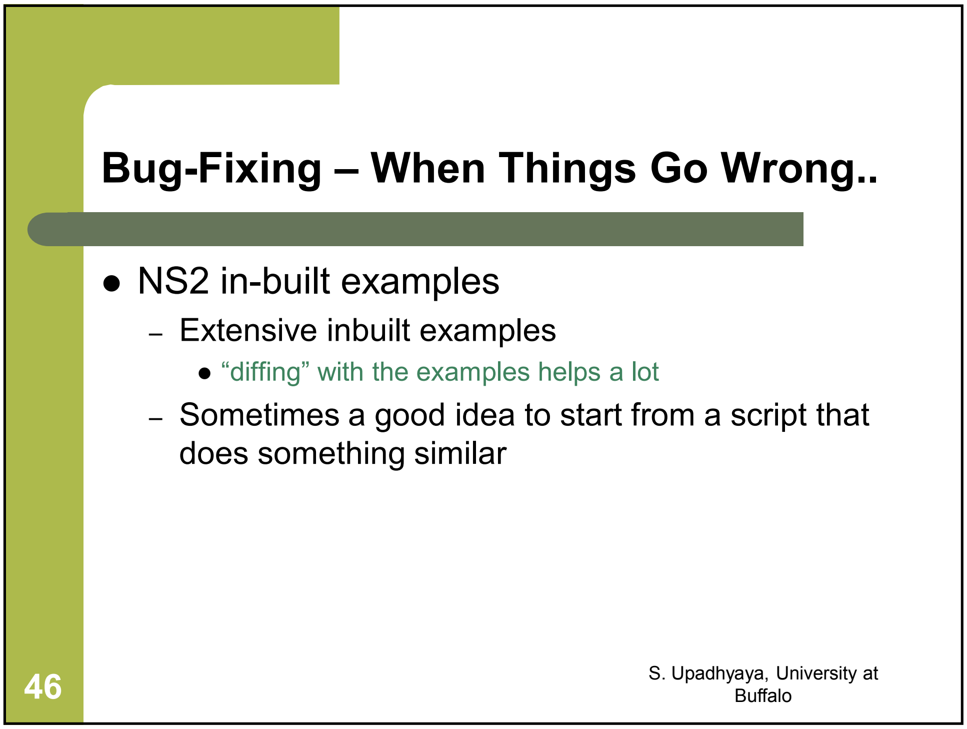


Bug-Fixing – When Things Go Wrong..

- Googling for the problem!
  - Extensive NS2 mailing lists
  - Chances are that other people have had the same problem are very high
  - Responsive forums

45

S. Upadhyaya, University at Buffalo

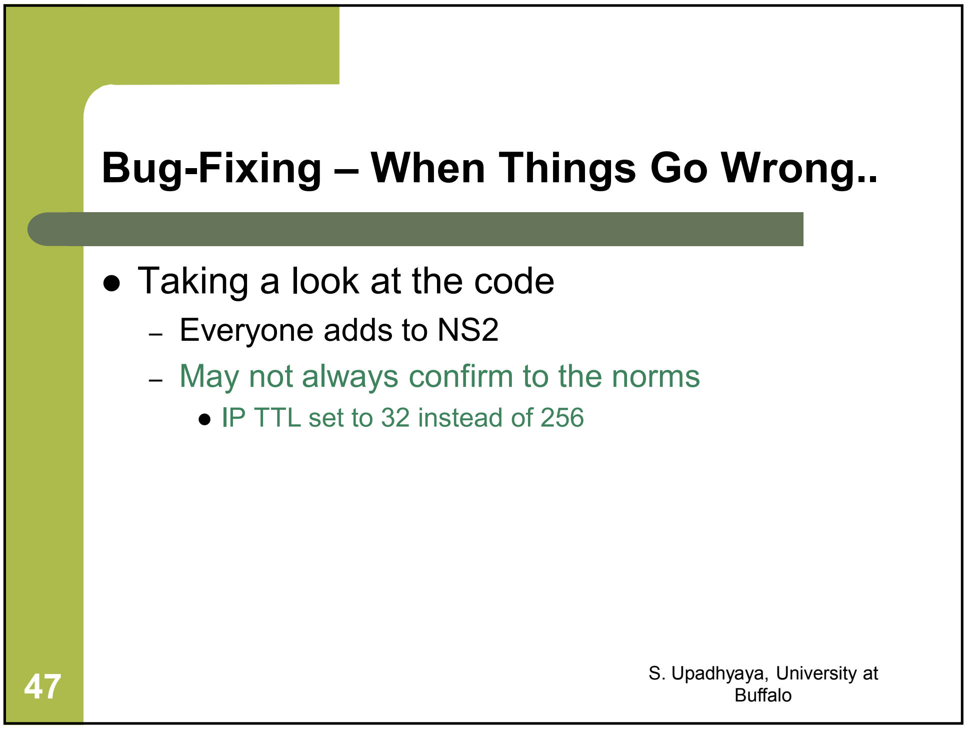


Bug-Fixing – When Things Go Wrong..

- NS2 in-built examples
  - Extensive inbuilt examples
    - “diffing” with the examples helps a lot
  - Sometimes a good idea to start from a script that does something similar

46

S. Upadhyaya, University at Buffalo

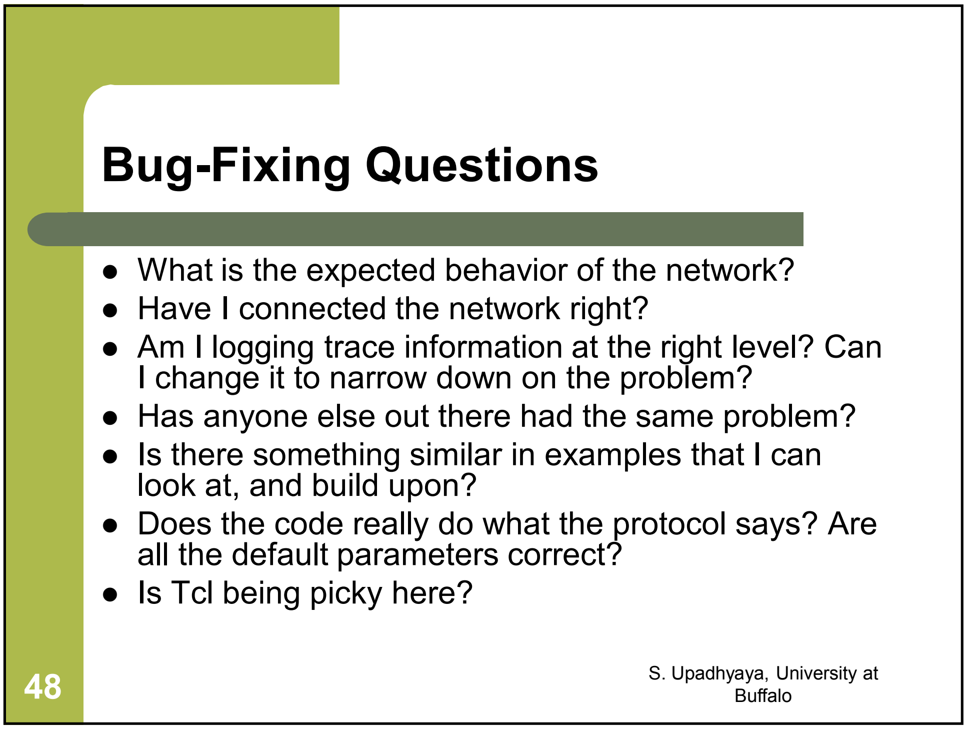


Bug-Fixing – When Things Go Wrong..

- Taking a look at the code
  - Everyone adds to NS2
  - May not always confirm to the norms
    - IP TTL set to 32 instead of 256

47

S. Upadhyaya, University at Buffalo



Bug-Fixing Questions

- What is the expected behavior of the network?
- Have I connected the network right?
- Am I logging trace information at the right level? Can I change it to narrow down on the problem?
- Has anyone else out there had the same problem?
- Is there something similar in examples that I can look at, and build upon?
- Does the code really do what the protocol says? Are all the default parameters correct?
- Is Tcl being picky here?

48

S. Upadhyaya, University at Buffalo