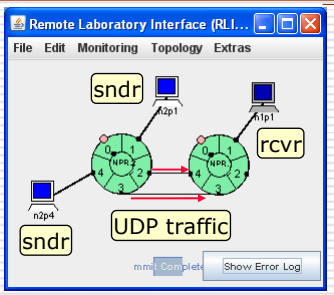
The Open Network Lab (Part 2)

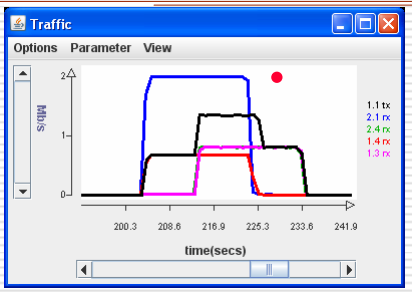
Ken Wong
Applied Research Laboratory
Computer Science and Engineering Department
http://www.arl.wustl.edu/~kenw
kenw@arl.wustl.edu
http://www.onl.wustl.edu (ONL)

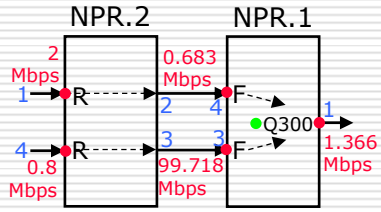
National Science Foundation ANI-023826, CNS-0551651, REL-0632580

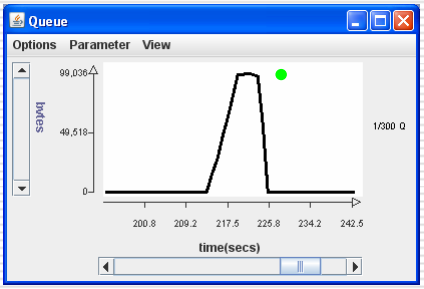
Washington University in St.Louis

Example 1

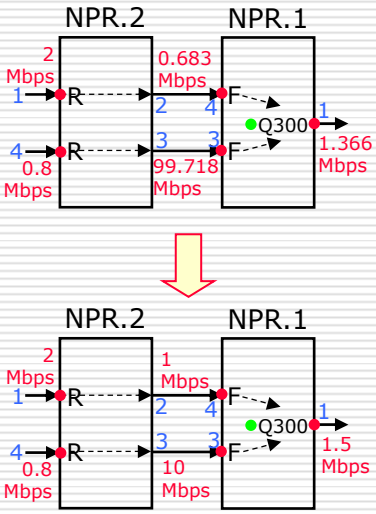






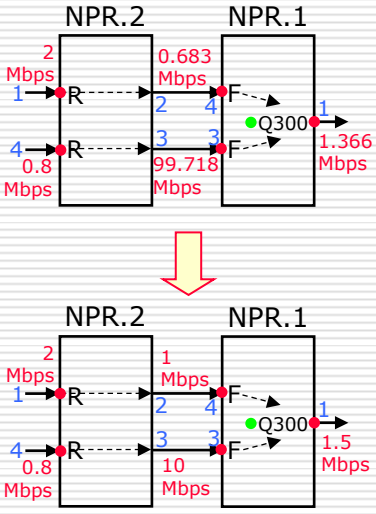


Example 1 (Modified)



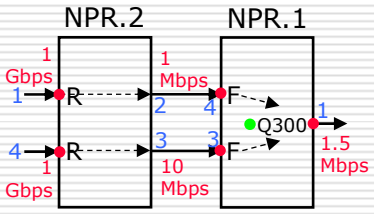
- Traffic
  - » Max size pkts = 1500 bytes
  - » n2p4 starts 10 sec after n2p1
  - » n2p1: 2 Mbps = 167 pps
  - » n2p4: 0.8 Mbps = 67 pps
- Port Capacities
  - » port 2/2:
  - » port 2/3:
  - » port 1/1:
- Queueing Rates
  - » port 2/2:
  - » port 2/3:
  - » port 1/1:
- Q300
  - » Capacity: 100,000B
  - » Overflows in sec

Example 1 (Modified)



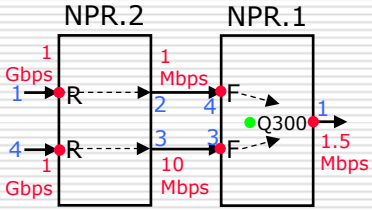
- Traffic
  - » Max size pkts = 1500 bytes
  - » n2p4 starts 10 sec after n2p1
  - » n2p1: 2 Mbps = 167 pps
  - » n2p4: 0.8 Mbps = 67 pps
- Port Capacities
  - » port 2/2: 1 Mbps = 84 pps
  - » port 2/3: 10 Mbps = 840 pps
  - » port 1/1: 1.5 Mbps = 126 pps
- Queueing Rates
  - » port 2/2: 2-1 = 1 Mbps
  - » port 2/3: 0 Mbps
  - » port 1/1: (1+0.8)-1.5 = 0.3 Mbps
    - 0.3 Mbps = 37,500 Bps = 25 pps
- Q300
  - » Capacity: 100,000B = 67 pkts
  - » Overflows in 67/25 = 2.7 sec

Example 2 (Back-To-Back Traffic)



- Different Traffic
  - » back-to-back UDP pkts from n2p1 and n2p4
  - » both flows start at the same time
  - » n2p1: 100 max-size pkts (1500 bytes each)
  - » n2p4: 100 max-size pkts (1500 bytes each)
- What queues must be changed so that no pkt drops occur?

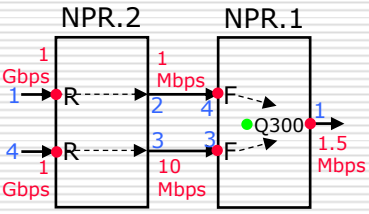
Example 2



- Different Traffic
  - » back-to-back UDP pkts from n2p1 and n2p4
  - » both flows start at the same time
  - » n2p1: 100 max-size pkts (1500 bytes each)
  - » n2p4: 100 max-size pkts (1500 bytes each)
- What queues must be changed so that no pkt drops occur?

- Transmission Volumes
  - » bytes from each source
  - » pragmatics: ONL UDP pkts
    - 8-byte UDP hdr, 20-byte IP hdr
    - 1470-byte UDP payload
    - ➔ actual pkt size = 1498 bytes
- File transfer time

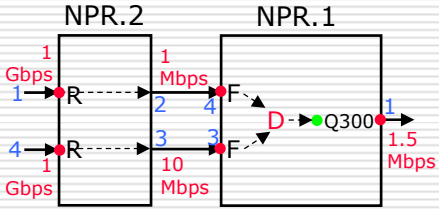
Example 2



- Different Traffic
  - » back-to-back UDP pkts from n2p1 and n2p4
  - » both flows start at the same time
  - » n2p1: 100 max-size pkts (1500 bytes each)
  - » n2p4: 100 max-size pkts (1500 bytes each)
- What queues must be changed so that no pkt drops occur?

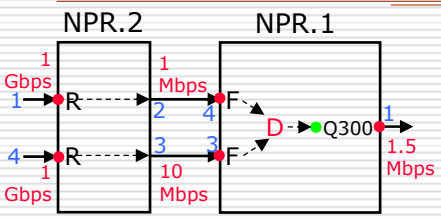
- Transmission Volumes
  - » bytes from each source
    - 150,000 bytes = 100x1500
  - » pragmatics: ONL UDP pkts
    - 8-byte UDP hdr, 20-byte IP hdr
    - 1470-byte UDP payload
    - → actual pkt size = 1498 bytes
- Q300 (NPR.1)
  - » Capacity (for 2 flows)
    - 300,000 bytes = 200 pkts
- Port 2/1, Datagram Queue
  - » Capacity (for 1 flow)
    - 150,000 bytes = 100 pkts
- File transfer time
  - » 1.6 sec = (1.2+1.2)/1.5

Example 3 (Forward Delay)



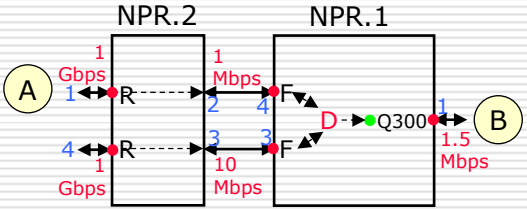
- Effect of 50 msec delay along forward path?
  - » arrival of first pkt to n1p1
  - » finish time
- Interpacket Times
  - » port 1/4:
  - » port 1/3:
  - » n1p1:
- Bandwidth-Delay Products
  - » n1p4-n1p1:
  - » n1p3-n1p1:

Example 3 (Forward Delay)

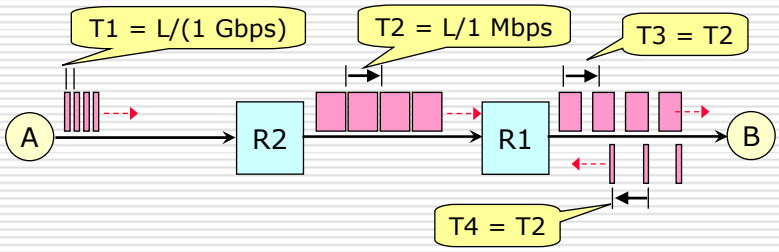


- Effect of 50 msec delay along forward path?
  - » arrival of first pkt to n1p1
    - delayed by 50 msec
  - » finish time
    - 50 msec longer than without delay
- Interpacket Times
  - » port 1/4: 1/(84 pps) = 11.9 msec
  - » port 1/3: 1/(8400 pps) = 0.119 msec
  - » n1p1: 1/(126 pps) = 7.93 msec
- Bandwidth-Delay Products
  - » n1p4-n1p1: 1 Mbps x 50 msec = 50 Kb = 33.3 pkts
  - » n1p3-n1p1: 10 Mbps x 50 msec = 500 Kb = 333 pkts

Example 4 (ACK Delay)



- Suppose
  - » Small ACK pkt returned along reverse path for each data pkt



ONL Issues

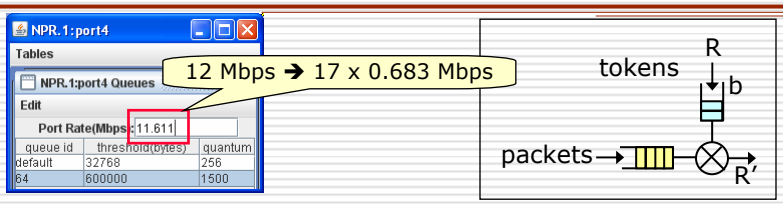
- Port rate
  - » Integer multiple of 0.683 Mbps
    - e.g., 12 Mbps ➔ 11.611
- Receiver socket buffer size
  - » Default: 84 KB
  - » Avoid ignoring incoming pkts for too long
  - » Use setsockopt() to increase buffer size

```
n = 500 * 1500;          // 500 max-size pkts
rc = setsockopt( sockfd, SOL_SOCKET, SO_RCVBUF, &n, sizeof(n));
```

- C/C++ code examples (onlusr:~kenw/src/)
  - » pkt-pair/
  - » xstcp/

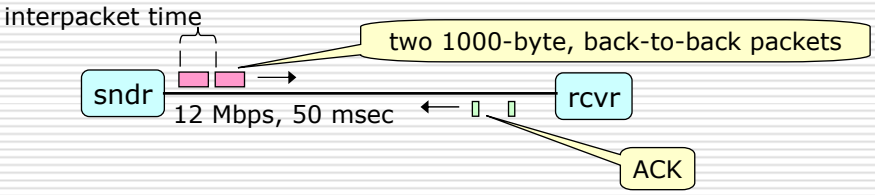
The Meaning of Port Rate

Queue Table ➔ Port Rate



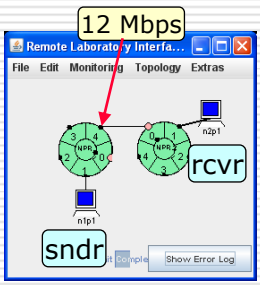
- Controlled by a token bucket model
  - » R: long-term average link rate (RLI parameter)
    - 54 Kbps granularity
  - » b: maximum bucket depth (4000 bytes)
  - » R': physical link rate (1 Gbps)
- Effect
  - » avg output rate is R
  - » peak rate is R'
- Operational definition
  - » fill token bucket at rate R
  - » transmit pkt when #tokens ≥ length of pkt at head of queue
- *Link regulator* model, NOT link emulator

The Packet-Pair Problem

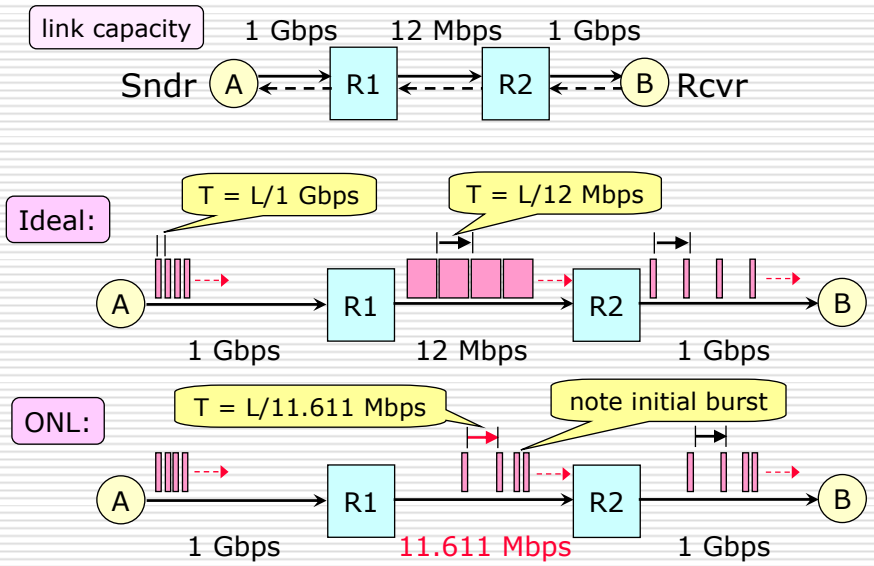


- Keshav's idea
- Rcvr receives the two pkts at times t0 and t1
  - » What will the *interpacket time* t1-t0 be?
  - » *average interbit time*?

ONL Experiment:



ONL Packet-Pair Experiment (1)



ONL Packet-Pair Experiment (2)

Puts $n1p2, etc. into Linux bash environment

```
onl
onlusr> source ~onl/.topology
bash: export: `NPR.1=NPR.9': not a valid identifier
bash: export: `NPR.2=NPR.10': not a valid identifier
onlusr> ssh $n2p1
Last login: Wed Oct  8 16:22:35 2008 from onlusr.arl.wustl.edu
onl052> cd src/pkt-pair
onl052> pp-rcvr
```

NPR.1:port4

Tables

NPR.1:port4 Queues

Edit

Port Rate(Mbps): 11.611

queue id	threshold(bytes)	quantum
default	32768	256
64	600000	1500

```
onl
onlusr> source ~onl/.topology
bash: export: `NPR.1=NPR.9': not a valid identifier
bash: export: `NPR.2=NPR.10': not a valid identifier
onlusr> ssh $n1p1
Last login: Wed Oct  8 18:13:01 2008 from onlusr
onl051> cd src/pkt-pair
onl051> pp-sndr -r n2p1 -n 7
RTTs (msec):            2103.76 2103.74 2103.74 2103.73 2103.72 2103.72 2103.72
Sndr Interpkt Times (usec):       20      10      15      13      13      13
Rcvr Interpkt Times (usec):       17      19     210    1037    1028    1047
Bandwidth est. at sndr (Mbps):  599.20 1198.40  798.93  921.85  921.85  998.67
Bandwidth est. at rcvr (Mbps):  704.94  630.74   57.07   11.56   11.66   11.45
onl051>
```

rcvr: n2p1
#pkts: 7

20-byte IP hdr
8-byte UDP hdr
1470-byte payload