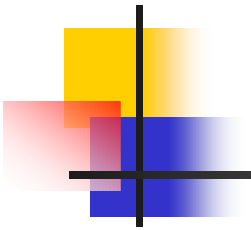
Linux Kernel Issues in End Host Systems

Wenji Wu, Matt Crawford

US-LHC End-to-End Networking Meeting

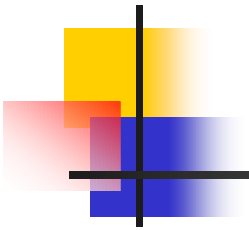
Fermi National Accelerator Lab, 2006

wenji@fnal.gov; crawdad@fnal.gov



Topics

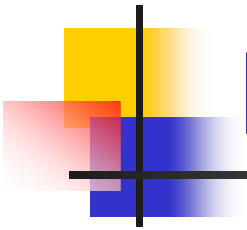
- Background
- Linux 2.6 Characteristics
- Kernel Memory Layout vs. Packet Receiving
- Kernel Preemptivity vs. Linux TCP Performance
- Interactivity vs. Fairness in Networked Linux Systems



Background

- What, Where, and How are the bottlenecks of Network Applications?
  - Networks?
  - Network End Systems?

Linux is widely used in the HEP community

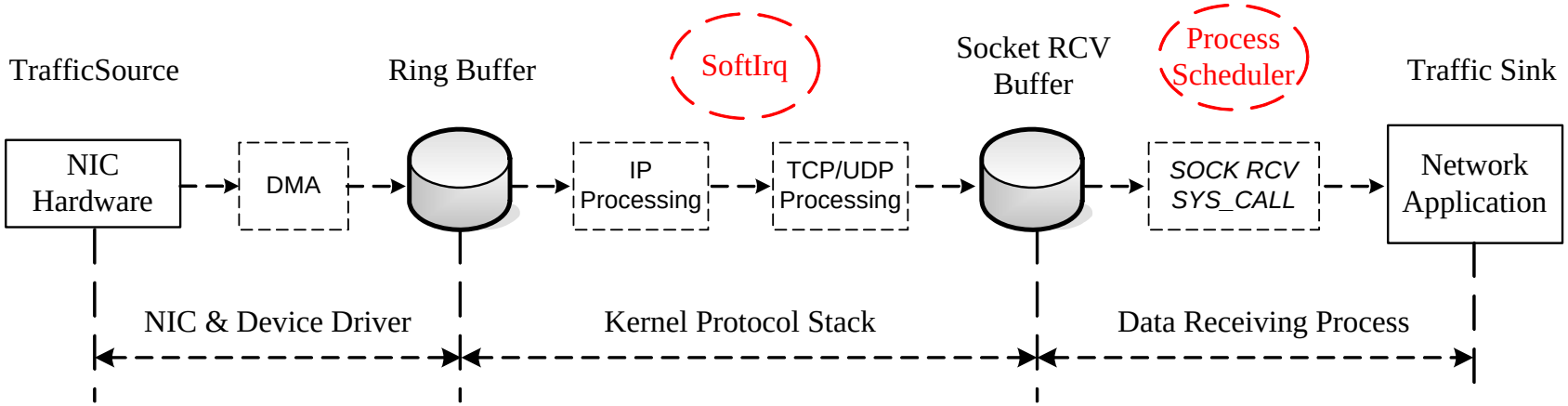


Linux 2.6 Characteristics

- Preemptible Kernel
- O(1) Scheduler
- Improved Interactivity, more responsive
- Improved Fairness
- Improved Scalability

Kernel Memory Layout vs. Packet Receiving

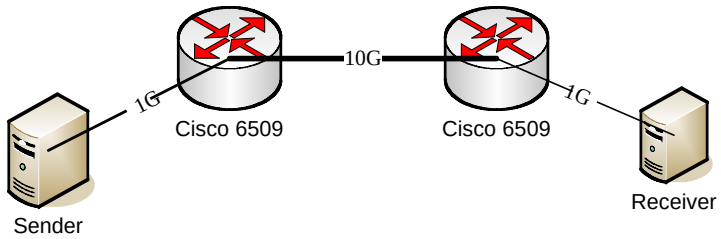
Linux Networking subsystem: Packet Receiving Process



- Stage 1: NIC & Device Driver
  - Packet is transferred from network interface card to ring buffer
- Stage 2: Kernel Protocol Stack
  - Packet is transferred from ring buffer to a socket receive buffer
- Stage 3: Data Receiving Process
  - Packet is copied from the socket receive buffer to the application

Experiment Settings

- Run *iperf* to send data in one direction between two computer systems;
- We have added instrumentation within Linux packet receiving path
- Compiling Linux kernel as background system load by running ***make –nj***
- Receive buffer size is set as 40M bytes



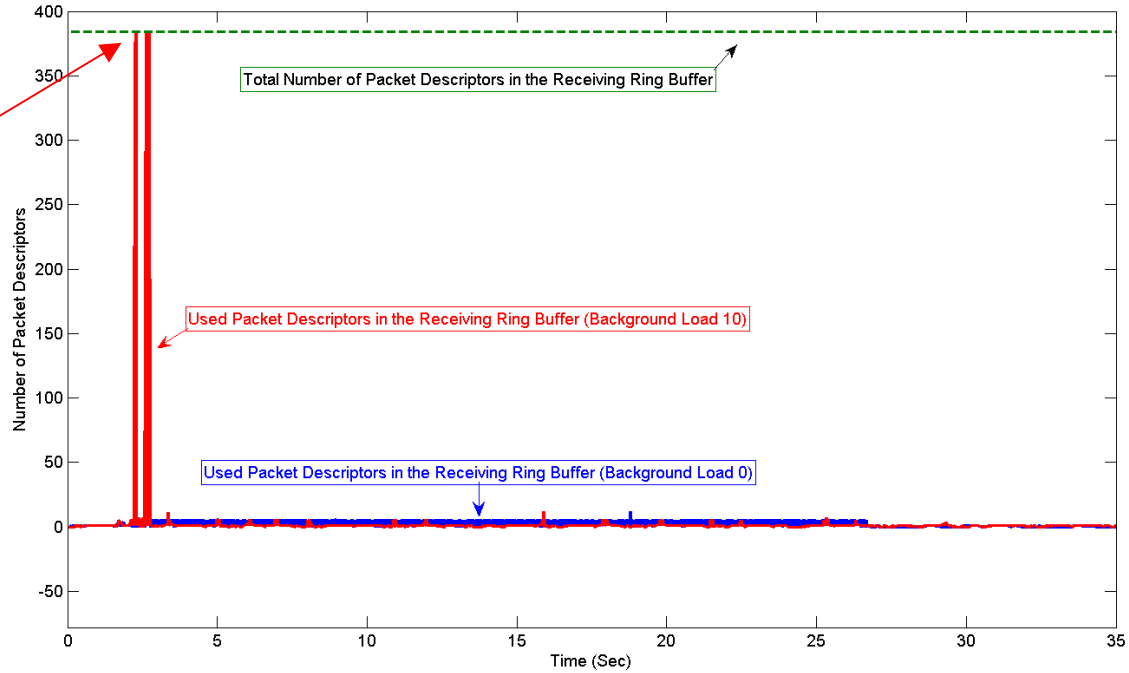
Fermi Test Network

	Sender	Receiver
CPU	Two Intel Xeon CPUs (3.0 GHz)	One Intel Pentium II CPU (350 MHz)
System Memory	3829 MB	256MB
NIC	Tigon, 64bit-PCI bus slot at 66MHz, 1Gbps/sec, twisted pair	Syskonnect, 32bit-PCI bus slot at 33MHz, 1Gbps/sec, twisted pair

Sender & Receiver Features

Receive ring buffer

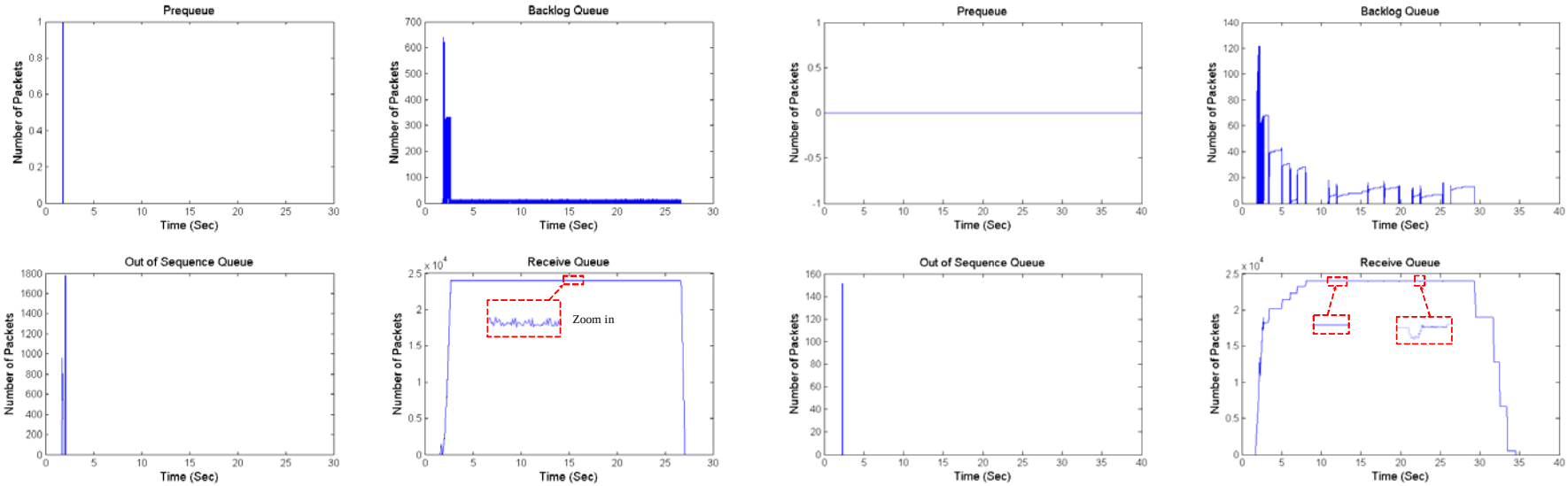
Running out packet descriptors



Total number of packet descriptors in the reception ring buffer of the NIC is 384

Receive ring buffer could run out of its packet descriptors: Performance Bottleneck!

Various TCP Receive Buffer Queues



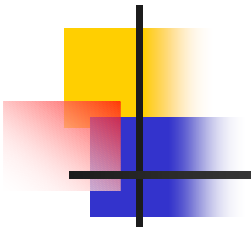
Background Load 0

Background Load 10

Receive buffer size is set as 40M bytes

What do the results mean?

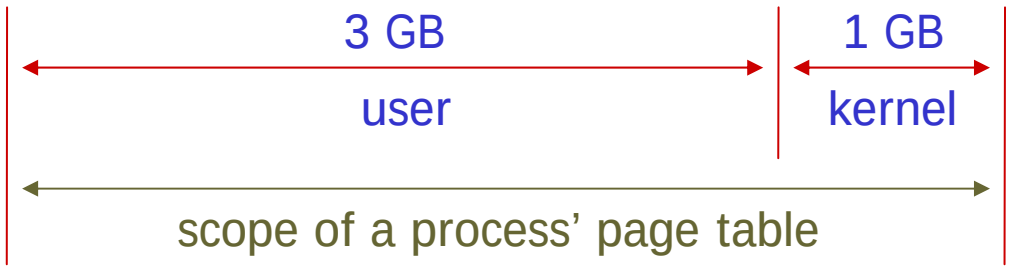
How to configure socket receive buffer size?



- We usually configure the socket receive buffer to the BDP.
- In real world, system administrators often configure /proc/net/ipv4/tcp_rmem high to accommodate high BDP connections.

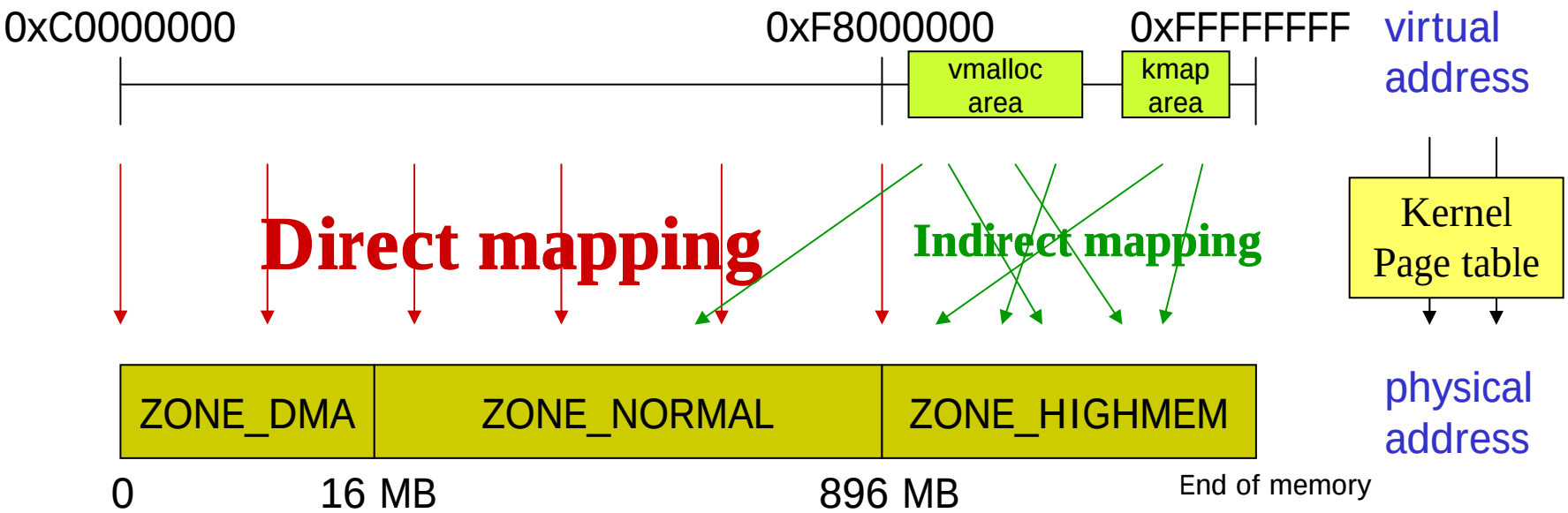
What could be wrong?

Linux Virtual Address Layout



- 3G/1G partition
  - The way Linux partition a 32-bit address space
  - Cover user and kernel address space at the same time
  - Advantage
    - Incurs no extra overhead (no TLB flushing) for system calls
  - Disadvantage
    - With 64 GB RAM, `mem_map` alone takes up 512 MB memory from lowmem (ZONE_NORMAL).

Partition of Physical Memory (Zone)

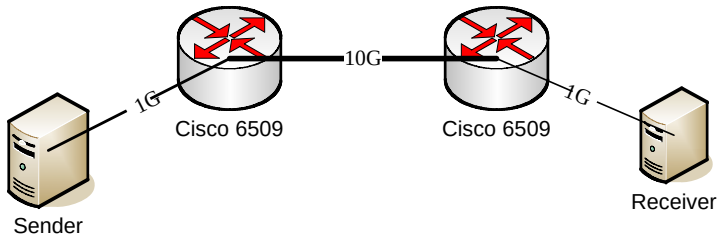


This figure shows the partition of physical memory its mapping to virtual address in 3G/1G layout

Kernel Preemptivity vs. Linux TCP Performance

Preemptivity vs. Linux TCP Performance Experiment Settings

- Run *iperf* to send data in one direction between two computer systems
- We have added instrumentation within Linux packet receiving path
- Compiling Linux kernel as background system load by running ***make –nj***
- Receive buffer size is set as 40M bytes



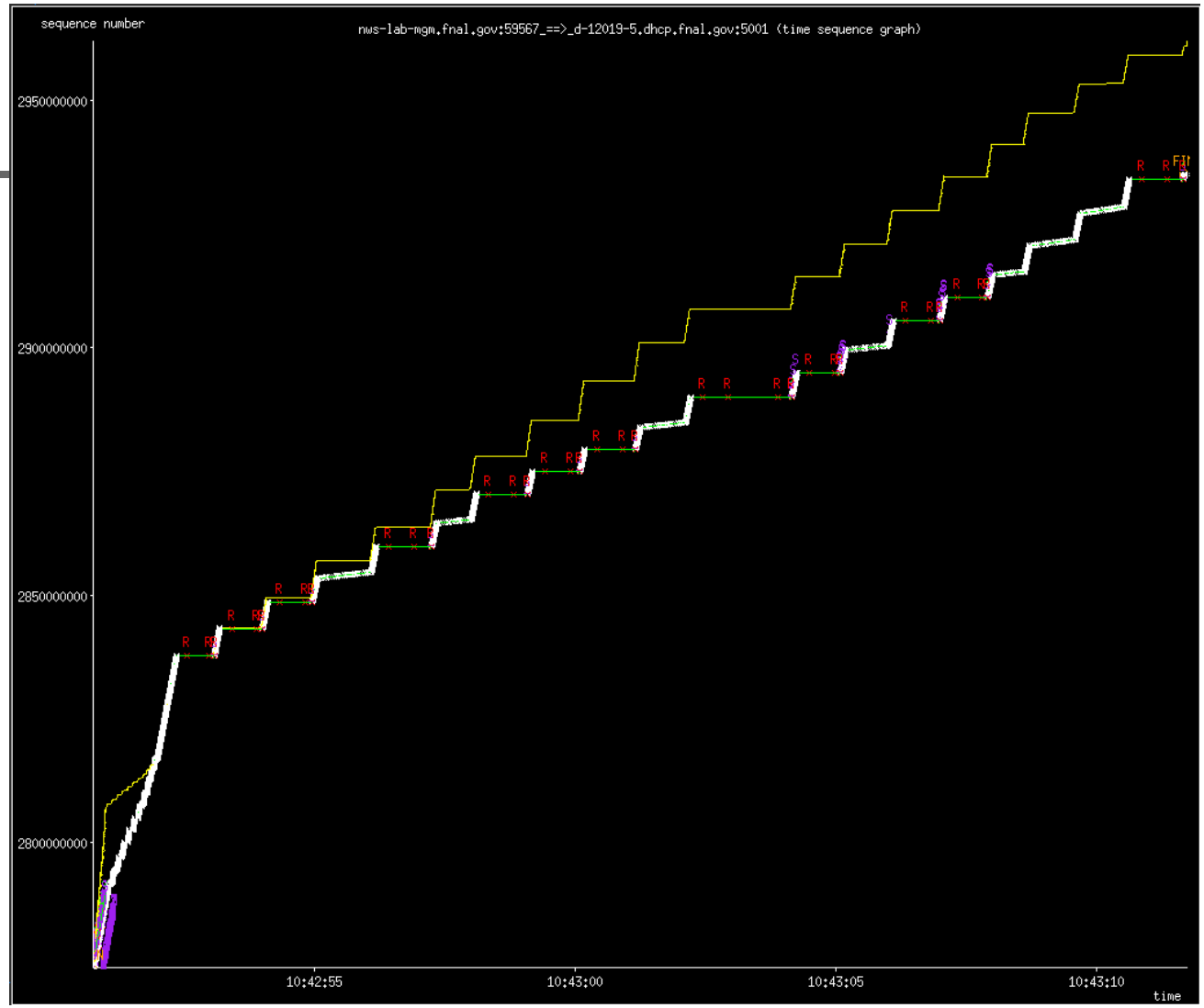
Fermi Test Network

	Sender	Receiver
CPU	Two Intel Xeon CPUs (3.0 GHz)	One Intel Pentium II CPU (350 MHz)
System Memory	3829 MB	256MB
NIC	Tigon, 64bit-PCI bus slot at 66MHz, 1Gbps/sec, twisted pair	Syskonnect, 32bit-PCI bus slot at 33MHz, 1Gbps/sec, twisted pair

Sender & Receiver Features

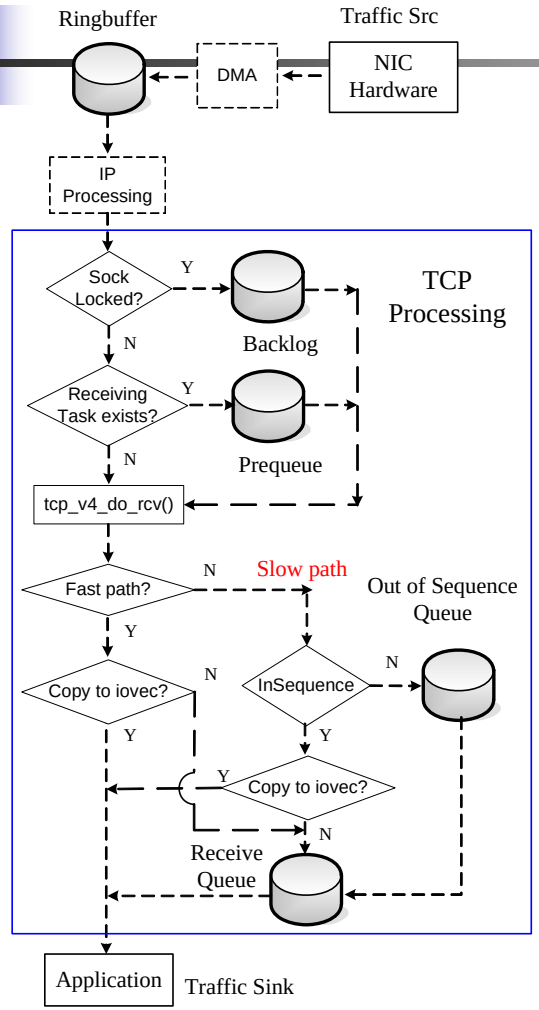
What, Why, and How?

Background Load 10

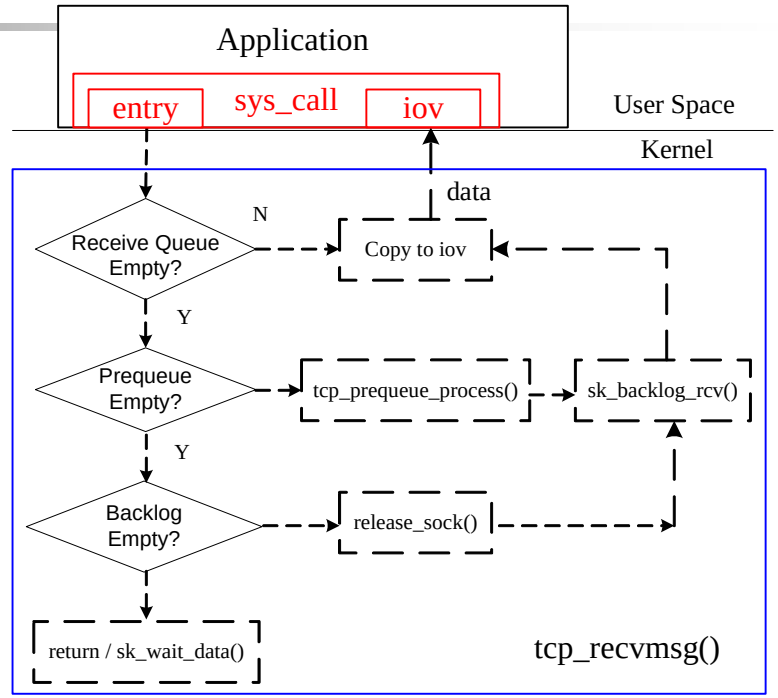


Tcptrace time-sequence diagram from the sender side

Kernel Protocol Stack – TCP



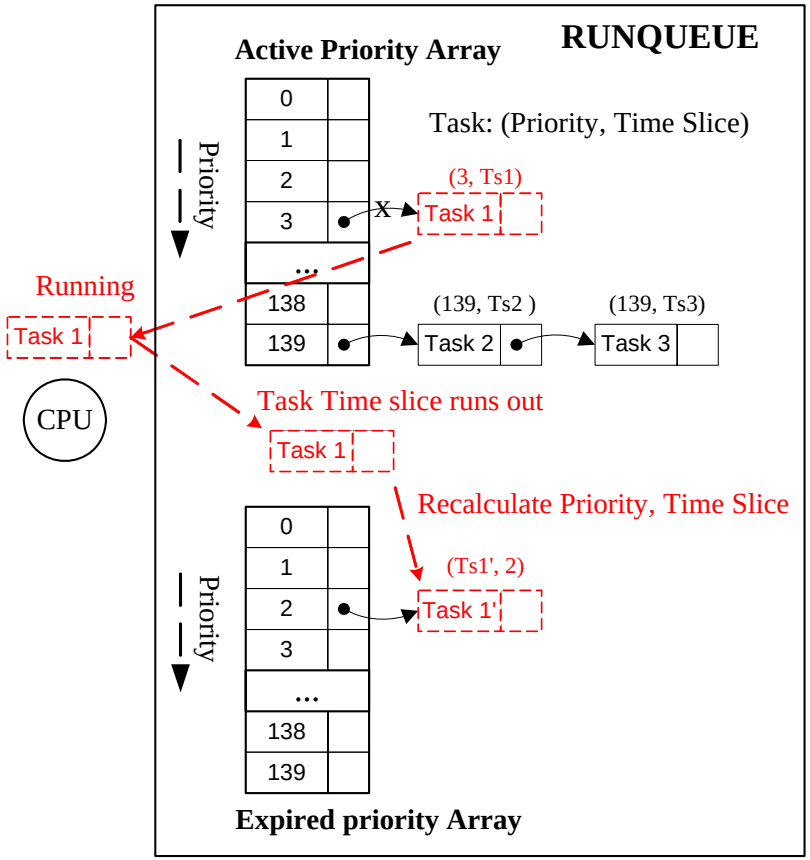
TCP Processing- Interrupt context

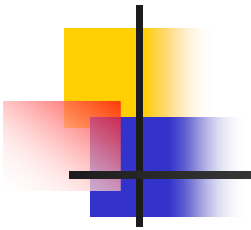


TCP Processing- Process context

Except in the case of prequeue overflow, Prequeue and Backlog queues are processed within the process context!

Linux Scheduling Mechanism

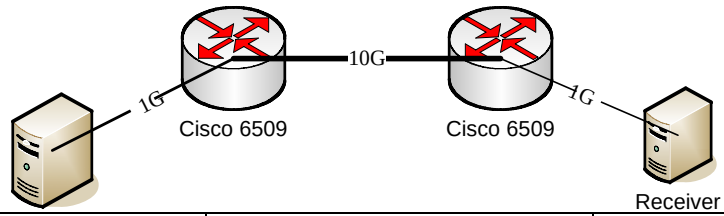




Interactivity vs. Fairness in Networked Linux Systems

Interactivity vs. Fairness Experiment Settings

- Run *iperf* to send data in one direction between two computer systems
- We have added instrumentation within Linux kernel
- Compiling Linux kernel as background system load by running ***make –nj***
- Receive buffer size is set as 40M bytes



Fermi Test Network

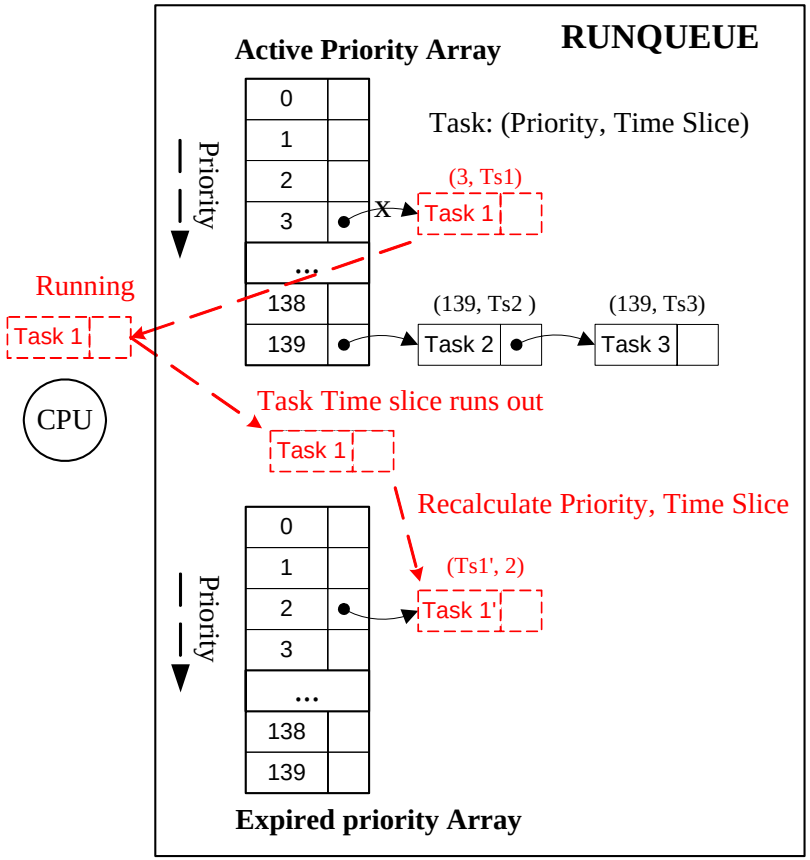
	Fast Sender	Slow Sender	Receiver
CPU	Two Intel Xeon CPUs (3.0 GHz)	One Intel Pentium IV CPU (2.8 GHz)	One Intel Pentium III CPU (1 GHz)
System Memory	3829 MB	512MB	512MB
NIC	Syskonnect, 32bit-PCI bus slot at 33MHz, 1Gbps/sec, twisted pair	Intel PRO/1000, 32bit-PCI bus slot at 33 MHz 1Gbps/sec, twisted pair	3COM, 3C996B-T, 32bit-PCI bus slot at 33MHz, 1Gbps/sec, twisted pair

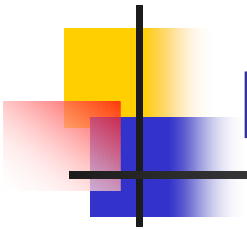
Sender & Receiver Features

What? Why? How?

	Slow Sender		Fast Sender	
Load	**Throughput**	**CPU Share**	**Throughput**	**CPU Share**
BL0	436 Mbps	78.489%	464 Mbps	99.228%
BL1	443 Mbps	81.573%	241 Mbps	49.995%
BL2	438 Mbps	80.613%	159 Mbps	34.246%
BL4	430 Mbps	79.217%	97.0 Mbps	20.859%
BL8	440 Mbps	81.093%	74.2 Mbps	15.375%

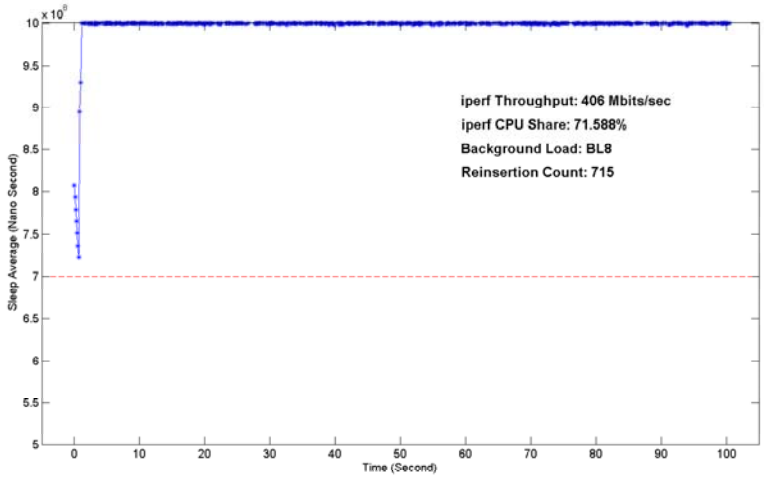
Linux Scheduling Mechanism



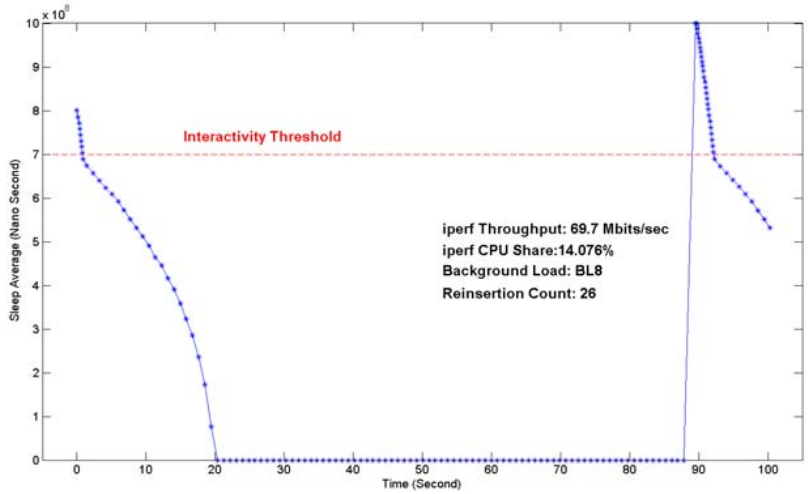


Network applications vs. interactivity

- A *sleep_avg* is stored for each process: a process is credited for its sleep time and penalized for its runtime. A process with high sleep_avg is considered interactive, and low sleep_avg is non-interactive.
- network packets arrive at the receiver independently and discretely; the “relatively fast” non-interactive network process might frequently sleep to wait for network packets. Though each sleep lasts for a short period of time, the wait-for-packet sleeps occur frequently, more than enough to lead to the interactivity status.
- The current Linux interactivity mechanism provides the possibilities that a non-interactive network process could consume a high CPU share, and at the same time be incorrectly categorized as “interactive.”



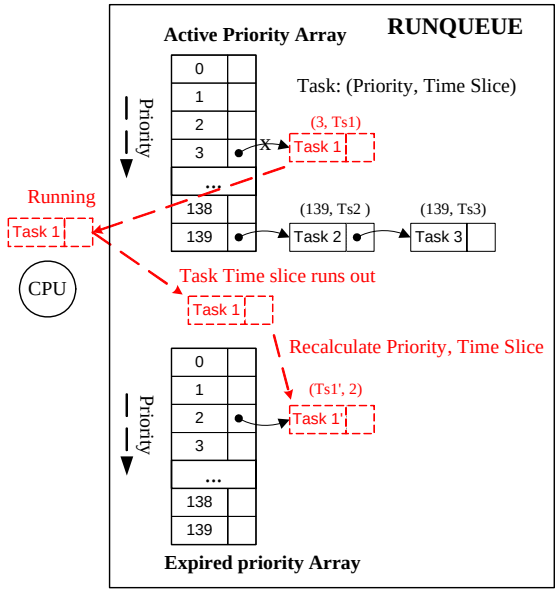
Slow Sender

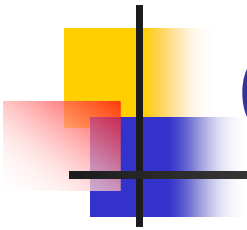


Fast Sender

With Slow Sender, iperf in the receiver is always categorized as interactive.

With fast sender, iperf in the receiver is categorized as non-interactive most of the time.





Contacts

- Wenji Wu, wenji@fnal.gov
- Matt Crawford, crawdad@fnal.gov

Wide Area Systems, Fermilab, 2006