Invited Talk on “Research Directions for the Network Research Community”

The Software Metaphor for LAN PHY ≠ WAN PHY:
Why High-Speed Networking in Clusters ≠ High-Speed Networking in Grids

Wu-chun (Wu) Feng
feng@lanl.gov

RADIANT: Research And Development in Advanced Network Technology
http://www.lanl.gov/radiant
Computer & Computational Sciences Division
Los Alamos National Laboratory
University of California

LA-UR 01-6042 W. Feng, Los Alamos National Laboratory
The Software Metaphor for LAN PHY ≠ WAN PHY: Why High-Speed Networking in Clusters ≠ High-Speed Networking in Grids

Alternate “National Enquirer” Title

Why the High-End Scientific Computing Community “Secretly” Despises the Networking Community

Primarily due to the “The Wizard Gap”
Matt Mathis, PSC
Outline

• Who Are We and What Do We Do?
  - So Many Research Directions, So Little Time …

• Background
  - High-Performance Computing (HPC)
  - High-Performance Networking (HPN)

• Why HPN in Supercomputers & Clusters ≠ HPN in Grids
  - Host-Interface Bottlenecks → Supercomputers & Clusters
  - Adaptation Bottlenecks → Grids

• Conclusion

• Relevant Publications & Software Distributions

• Acknowledgements: Current Collaborators
Who Are We and What Do We Do?

- Team of 4 techno-geeks, 3 internal collaborators, gaggle of grad students.

- **High-Performance Networking**
  - User-Level Network Interfaces (ST OS-Bypass / Elan RDMA)
  - High-Performance IP & Flow- and Congestion-Control in TCP

- **(Passive) Network Monitoring & Measurement at Gb/s Speeds & Beyond**
  - MAGNeT: Monitor for Application-Generated Network Traffic
  - TICKET: Traffic Information-Collecting Kernel with Exact Timing

- **Cyber-Security**
  - IRIS: Inter-Realm Infrastructure for Security
  - SAFE: Steganographic Analysis, Filtration, and Elimination

- **Performance Evaluation of Commodity Clusters & Interconnects**

- **Fault Tolerance & Self-Healing Clusters (using the network)**
  - Buffered Co-Scheduling & Communication-Induced Checkpointing

- **Network Architecture**
  - MINI Processors: Memory-Integrated Network-Interface Processors
  - Smart Routers

- For more information, go to our out-of-date web site at [http://www.lanl.gov/radiant](http://www.lanl.gov/radiant). (We anticipate updating the web site by SC 2001.)
Who Are We and What Do We Do?

• Team of 4 techno-geeks, 3 internal collaborators, gaggle of grad students.

• **High-Performance Networking**
  - User-Level Network Interfaces (ST OS-Bypass / Elan RDMA)
  - High-Performance IP & Flow- and Congestion-Control in TCP

• **(Passive) Network Monitoring & Measurement at Gb/s**
  - MAGNeT: Monitor for Application-Generated Network Traffic
  - TICKET: Traffic Information-Collecting Kernel with Exact Timing

• **Cyber-Security**
  - IRIS: Inter-Realm Infrastructure for Security
  - SAFE: Steganographic Analysis, Filtration, and Elimination

• **Performance Evaluation of Commodity Clusters & Interconnects**

• **Fault Tolerance & Self-Healing Clusters (using the network)**
  - Buffered Co-Scheduling & Communication-Induced Checkpointing

• **Network Architecture**
  - MINI Processors: Memory-Integrated Network-Interface Processors
  - Smart Routers

• For more information, go to our out-of-date web site at [http://www.lanl.gov/radiant](http://www.lanl.gov/radiant). (We anticipate updating the web site by SC 2001.)
What is High-Performance Computing (HPC)?

- **Tightly-Coupled Supercomputers**
  - LLNL’s ASCI White, SDSC’s Blue Horizon, PSC’s TCS

- **High-End Clusters / PC Clusters**
  - NCSA’s Titan (to be used as part of DTF), LANL’s Avalon

- **Distributed Clusters & MicroGrids**
  - Intel’s internal microgrid

- **Computational Grids / Virtual Supercomputers**
  - Industry: United Devices (SETI@Home), Entropia, Parabon
  - Academia: Earth System Grid, Particle Physics Data Grid, Distributed Terascale Facility.

However, all the above platforms will continue to exist over the next decade, e.g., NCSA’s Titan will be a cluster in its own right as well as a grid node in DTF.
HPC → High-Performance Networking (HPN)

• Problems in Achieving HPN in HPC
  ➢ Why HPN in Supercomputers & Clusters ≠ HPN in Grids

• Tightly-Coupled Supercomputers & PC Clusters
  ➢ Network Environment: Generally, SANs/LANs using non-IP.
    (Exception: Beowulf clusters that use IP.)
  ➢ Why non-IP routing? Host-interface bottlenecks.
  → Latency is *generally* more of an issue than bandwidth.

• Computational Grids
  ➢ Network Environment: WAN using TCP/IP.
  ➢ Why is performance so lousy? Adaptation bottlenecks.
  → Bandwidth is *generally* more of an issue than latency.
Host-Interface Bottlenecks

- Software
  - Host can only send & receive packets as fast as OS can process them.
    - Excessive copying. (A known fact.)
    - Excessive CPU utilization. (See next slide.)

- Hardware (PC)
  - PCI I/O bus. 64 bit, 66 MHz = 4.2 Gb/s.
  - Solutions? PCI-X, InfiniBand, 3GIO/Arapahoe, Hypertransport, MINI Processors?

10GigE packet inter-arrival: 1.2 μs
(assuming 1500-byte MTUs)
Null system call in Linux: 5-10 μs
Host-Interface Bottlenecks

10GigE packet inter-arrival: 1.2 µs
(assuming 1500-byte MTUs)
Null system call in Linux: 5-10 µs

- Software
  - Host can only send & receive packets as fast as OS can process them.
    - Excessive copying.
    - Excessive CPU utilization.
- Hardware
  - PCI I/O bus: 64-bit, 66 MHz = 4.2 Gb/s
  - Solutions? PCI-X, InfiniBand, 3GIO/Arapahoe, Hypertransport, MINI Processors?

We have reached a crossover point with current software and hardware – network speeds are outstripping the ability of the CPU to keep up.
666-MHz Alpha with Linux
(Courtesy: USC/ISI)

Even jumbograms suffer from high CPU utilization …
Host-Interface Bottleneck (Software)

• First-Order Approximation
  - deliverable bandwidth = maximum-sized packet / interrupt latency
  - e.g., 1500-byte MTU / 50 $\mu$s = 30 MB/s = 240 Mb/s
  - 625 Mb/s – 900+ Mb/s

• Problems
  - Maximum-sized packet (or MTU) is only 1500 bytes for Ethernet.
  - Interrupt latency to process a packet is quite high.
  - CPU utilization for network tasks is too high.

• Solutions Intended to Boost TCP/IP Performance
  - Eliminate excessive copying, e.g., “zero-copy” stack, OS-bypass w/ RDMA.
  - Reduce frequency of interrupts, e.g., high-perf. IP, interrupt coalescing, jumbograms, OS-bypass.
  - Increase effective MTU size, e.g., high-perf. IP, interrupt coalescing, jumbograms.
  - Reduce interrupt latency, e.g., high-perf. IP, push checksums into hardware, “zero-copy”
  - Reduce CPU utilization, e.g., offload protocol processing to NIC $\Rightarrow$ high-performance IP.
Solutions to Boost TCP/IP Performance

(many non-TCP & non-standard)

- Interrupt Coalescing
  - Increases bandwidth (BW) at the expense of even higher latency.

- Jumbograms
  - Increases BW with minimal increase in latency, but at the expense of potentially more blocking in switches/routers and lack of interoperability. J. Cain (Cisco): It is very difficult to build switches to switch large packets such as a jumbogram.

- ULNI or OS-Bypass Protocol with RDMA
  - Increases BW & decreases latency by an order of magnitude or more.
  - Integrate OS-bypass into TCP?
    VIA over TCP (IETF Internet Draft, GigaNet, July 2000).

- Interrupt Latency Reduction (possible remedy for TCP)
  - Provide “zero-copy” TCP (à la OS-bypass) but OS still middleman.
  - Push protocol processing into hardware, e.g., checksums.

- High-Performance IP (to be described later)
  - Reduce CPU utilization, increase bandwidth, decrease latency.
Solutions to Boost TCP/IP Performance  
(many non-TCP & non-standard)

- **Interrupt Coalescing**  
  - Increases bandwidth (BW) at the expense of even higher latency.

- **Jumbograms**  
  - Increases BW with minimal increase in latency, but at the expense of potentially more blocking in switches/routers and lack of interoperability. J. Cain (Cisco): It is very difficult to build switches to switch large packets such as a jumbogram.

- **ULNI or OS-Bypass Protocol with RDMA**  
  - Increases BW & decreases latency by an order of magnitude or more.  
  - Integrate OS-bypass into TCP?  
    VIA over TCP (IETF Internet Draft, GigaNet, July 2000).

- **Interrupt Latency Reduction** (possible remedy for TCP)  
  - Provide “zero-copy” TCP (*a la* OS-bypass) but OS still middleman.  
  - Push protocol processing into hardware, e.g., checksums.

- **High-Performance IP** (to be described later)  
  - Reduce CPU utilization, increase bandwidth, decrease latency.
Solutions to Boost TCP/IP Performance
(many non-TCP & non-standard)

- Interrupt Coalescing
  - Increases bandwidth (BW) at the expense of even higher latency.

- Jumbograms
  - Increases BW with minimal increase in latency, but at the expense of potentially more blocking in switches/routers and lack of interoperability. J. Cain (Cisco): It is very difficult to build switches to switch large packets such as a jumbogram.

- ULNI or OS-Bypass Protocol with RDMA
  - Increases BW & decreases latency by an order of magnitude or more.
  - Integrate OS-bypass into TCP?
    VIA over TCP (IETF Internet Draft, GigaNet, July 2000).

- Interrupt Latency Reduction (possible remedy for TCP)
  - Provide “zero-copy” TCP (a la OS-bypass) but OS still middleman.
  - Push protocol processing into hardware, e.g., checksums.

- High-Performance IP (to be described later)
  - Reduce CPU utilization, increase bandwidth, decrease latency.

No! No! No!

These issues will be subsumed by high-performance IP.
(It may not be the ideal solution, but it is the “legacy” solution, much like the x86 architecture or the Windows OS.)
Solutions to Boost TCP/IP Performance
(many non-TCP & non-standard)

• Interrupt Coalescing
  ➢ Increases BW at the expense of even higher latency.

• Jumbograms
  ➢ Increases BW with minimal increase in latency, but at the expense of potentially more blocking in switches/routers and lack of interoperability. J. Cain (Cisco): It is very difficult to build switches to switch large packets.

• ULNI or OS-Bypass Protocol with RDMA
  ➢ Increases BW & decreases latency by an order of magnitude or more.
  ➢ Integrate OS-bypass into TCP? VIA over TCP (IETF Internet Draft, GigaNet, July 2000).

• Interrupt Latency Reduction (possible remedy for TCP)
  ➢ Provide “zero-copy” TCP (a la OS-bypass) but OS still middleman.
  ➢ Push protocol processing into hardware, e.g., checksums.

• High-Performance IP (to be described later)
  ➢ Reduce CPU utilization, increase bandwidth, decrease latency.

No!
No!
No!

These issues will be subsumed by high-performance IP. (It may not be the ideal solution, but it is the “legacy” solution, much like the x86 architecture or the Windows OS.)

Why not?
What Can OS-Bypass Protocols Do?

• Problems with TCP for HPC in the Mid-1990s
  ➢ Computing Paradigm: Cluster or supercomputer + computational grid
  ➢ Network Environment: System-area network (SAN) + wide-area network (WAN)
  ➢ TCP (mid-90s): Latency: $O(1000 \ \mu s)$. BW: $O(10 \ \text{Mb/s})$.
  ➢ TCP (today): Latency: $O(100 \ \mu s)$. BW: $O(500 \ \text{Mb/s})$.
  ➢ TCP (optimized): Latency: $50 \ \mu s$. BW: $1.42 \ \text{Gb/s}$. [Quadrics TCP/IP, LANL]
  Problem: ULNIs do not scale to WAN.

• Solution
  ➢ No automated routing (IP, ARP) & no congestion control.
  ➢ User-level network interfaces (ULNIs) or OS-bypass protocols w/ RDMA.
    ➢ Active Messages, FM, PM, U-Net. Recently, VIA (Compaq, Intel, µsoft)
  ➢ ULNI (mid-90s): Latency: $O(10 \ \mu s)$. BW: $O(600-800 \ \text{Mb/s})$.
  ➢ ULNI Performance [Quadrics Elan OS-Bypass w/ RDMA, LANL]
    ➢ Latency: $1.9 \ \mu s$. BW: $3.14 \ \text{Gb/s}$.
    ➢ User-Level Latency: $4.5 \ \mu s$. User-Level BW: $2.46 \ \text{Gb/s}$.
What Can OS-Bypass Protocols Do?

- Problems with TCP for HPC in the Mid-1990s
  - Computing Paradigm: Cluster or supercomputer + computational grid
  - Network Environment: System-area network (SAN) + wide-area network (WAN)
  - TCP (mid-90s): Latency: $O(1000 \mu s)$. BW: $O(10 \text{ Mb/s})$.
  - TCP (today): Latency: $O(100 \mu s)$. BW: $O(500 \text{ Mb/s})$.
  - TCP (optimized): Latency: 50 $\mu$s. BW: 1.42 Gb/s. [Quadrics TCP/IP, LANL]
  - Problem: ULNIs do not scale to WAN.

- Solution
  - User-level network interfaces (ULNIs) or OS-bypass protocols w/ RDMA.
    - Active Messages, FM, PM, U-Net. Recently, VIA (Compaq, Intel, µsoft)
  - ULNI (mid-90s): Latency: $O(10 \mu s)$. BW: $O(600-800 \text{ Mb/s})$.
  - ULNI Performance [Quadrics Eian OS-Bypass w/ RDMA, LANL]
    - Latency: 1.9 $\mu$s.
    - BW: 3.14 Gb/s.
    - User-Level Latency: 4.5 $\mu$s. User-Level BW: 2.46 Gb/s.
Solutions to Boost TCP/IP Performance
(many non-TCP & non-standard)

- **Interrupt Coalescing**
  - Increases BW at the expense of even higher latency.

- **Jumbograms**
  - Increases BW with minimal increase in latency, but at the expense of potentially more blocking in switches/routers and lack of interoperability.
  - Jeff Cain (Cisco): It is very difficult to build switches to switch large packets efficiently.

- **ULNI or OS-Bypass Protocol with RDMA**
  - Increases BW & decreases latency by an order of magnitude or more.
  - Integrate OS-bypass into TCP?
    - VIA over TCP (IETF Internet Draft, GigaNet, July 2000).

- **Interrupt Latency Reduction** (possible remedy for TCP)
  - Provide “zero-copy” TCP (*a la* OS-bypass) but OS still middleman.
  - Push protocol processing into hardware, e.g., checksums.

- **High-Performance IP** (to be described NOW)
  - Reduce CPU utilization, increase bandwidth, decrease latency.

---

No! No! No!

Why not? It does *not* scale to WANs.
TCP/IP is a ubiquitously deployed protocol.

These issues will be subsumed by high-performance IP.
(It may not be the ideal solution, but it is the “legacy” solution, much like the x86 architecture or the Windows OS.)
My MTU Is Bigger Than Your MTU, So There!

• What is the MTU size for the Quadrics network?
  ➢ 320 bytes! Yeah, that’s right … ~20% of an Ethernet MTU.
• What’s their secret? The virtual MTU size is on the order of 64KB.

*Bob Grow (Intel)* said, “If there’s a magic solution, we’ll adopt it.” 😊
• High-Performance IP over Gigabit Ethernet → 10GigE?
  ➢ Lightweight Protocol Off-Loading
    ▪ Configure device driver to accept *virtual MTUs (vMTU)* of up to 64 KB → TCP/IP transmits up to 64-KB vMTU to device driver. *Result: Minimize CPU overhead for fragmentation.*
    ▪ Make the firmware on the NIC do the fragmentation.
  ➢ Implemented with Alteon GigE AceNICs.
Summary: Software-Based Host-Interface Bottleneck

• Better performance in SAN? OS-bypass with RDMA

• Problems
  ➢ It does not scale to WANs in support of grids.
  ➢ TCP/IP is the ubiquitously-deployed protocol suite.

• Solutions
  ➢ Encapsulate (tunnel) ULNI/RDMA in TCP/IP over the WAN.
  ➢ Use TCP/IP but implement a high-performance IP for SANs and a “more adaptive” TCP. (To be discussed in “Adaptation Bottlenecks” slide.)

• Clusters vs. Grids
Host-Interface Bottleneck (Hardware)

- PCI = Pretty Crappy Interface 😊
  - Theoretical Peak Bandwidth
    - PCI 2.2, 32/33: 1.06 Gb/s (133 MB/s)
    - PCI 2.2, 64/33: 2.13 Gb/s (266 MB/s)
    - PCI 2.2, 64/66: 4.26 Gb/s (533 MB/s) → 2.64 Gb/s (330 MB/s)
    - PCI-X 1.0, 64/100: 6.40 Gb/s (800 MB/s)
    - PCI-X 1.0, 64/133: 8.53 Gb/s (1066 MB/s)

- Passive-monitoring TICKET is hardware-limited to 2.64 Gb/s right now … we’d love to have a 10GigE NIC to monitor the backbone traffic at SC 2001 ;-).

- Solutions? More or less out of our control …
  - InfiniBand, 3GIO/Arapahoe, Hypertransport, MINI Processors: Memory-Integrated Network-Interface Processors.
HPC → High-Performance Networking (HPN)

• Problems in Achieving HPN in HPC
  ➢ HPN in Supercomputers & Clusters ≠ HPN in Grids

• Tightly-Coupled Supercomputers & PC Clusters
  ➢ Network Environment: Generally, SANs/LANs using non-IP.
    (Exception: Beowulf clusters that use IP.)
  ➢ Why non-IP routing? Host-interface bottlenecks.
  → Latency is generally more of an issue than bandwidth.

• Computational Grids
  ➢ Network Environment: WAN using TCP/IP.
  ➢ Why is performance so lousy? Adaptation bottlenecks.
  → Bandwidth is generally more of an issue than latency.
Adaptation Bottlenecks

• Flow Control
  - No adaptation is currently being done in any “standard” TCP with one exception.
    - The recent release of Linux 2.4.x does “sender-based auto-tuning” of aggregated TCP connections.
    - Primary benefit is to web servers, not high-performance, bulk-data transfer.
  - Static-sized buffer is supposed to work for both the LAN & WAN.

• Congestion Control
  - Adaptation mechanisms will not scale, particularly TCP Reno.
  - Adaptation mechanisms induce burstiness to the aggregate traffic stream.
Flow-Control Adaptation

- **Issues**
  - No adaptation currently being done in any “standard” TCP.
  - 32-KB static-sized buffer that is supposed to work for both LAN & WAN.

- **Problem**: Large bandwidth-delay products require flow-control windows as large as 1024-KB to fill the network pipe.

- **Consequence**: As little as 3% of network pipe is filled.

- **Solutions**
  - **Manual** tuning of buffers at send and receive end-hosts.
    - Too small → low bandwidth. Too large → waste memory (LAN).
    - [http://www.psc.edu/networking/perf_tune.html](http://www.psc.edu/networking/perf_tune.html)
  - **Automatic** tuning of buffers.
    - Auto-tuning: Sender-based flow control.
      - [Semke, Mahdavi, & Mathis, PSC, 1998.](#) → Web100 & Net100.
    - Dynamic right-sizing: Receiver-based flow control.
      - [Fisk & Feng, LANL, 1999.](#)
    - ENABLE: “Database” of BW-delay products
      - [Tierney et al., LBNL, 2001.](#)
  - **Network striping & pipelining** with default buffers.
    - [UIC, 2000 & GridFTP @ ANL, 2001.](#)
**Congestion-Control Adaptation**

- Adaptation mechanisms will *not* scale due to
  - Additive increase / multiplicative decrease algorithm (see next slide).
    - *Induces* bursty (i.e., self-similar or fractal) traffic.

- **TCP Reno congestion control**
  - Bad: Allow/induce congestion.
  - Detect & recover from congestion.
  - Analogy: *“Deadlock detection & recovery” in OS.*
  - Result: *“At best” 75% utilization in steady state* (assuming no buffering).

- **TCP Vegas congestion control**
  - Better: Approach congestion but try to *avoid* it.
  - Usually results in better network utilization.
  - Analogy: *“Deadlock avoidance” in OS.*

---

Utilization vs. Time

```
100%  50%  100%  50%
```

W. Feng, Los Alamos National Laboratory
“Optimal” Bandwidth

- The future performance of computational grids (as well as clusters & supercomputers trying to get away from ULNI scalability problems) looks bad if we continue to rely on the widely-deployed TCP Reno.

  Example: High BW-delay product: 1 Gb/s WAN * 100 ms RTT = 100 Mb

- Additive increase
  - when window size is 1 → 100% increase in window size.
  - when window size is 1000 → 0.1% increase in window size.

Re-convergence to “optimal” bandwidth takes nearly 7 minutes!
(Performance is awful if network uncongested.)

Solutions: (1) Faster converging congestion control. (2) Larger MTU.
AIMD Congestion Control

• Stable & fair (under certain assumptions of synchronized feedback) but
  - Not well-suited for emerging applications (e.g., streaming & real-time audio and video)
    - Its reliability and ordering semantics increase end-to-end delays and delay variations.
    - Multimedia applications *generally* do not react well to the large and abrupt reductions in transmission rate caused by AIMD.
  - Solutions
    - Deploy “TCP-friendly” (non-AIMD) congestion-control algorithms, e.g., binomial congestion-control algorithms such as inverse increase / additive decrease (Bansal & Balakrishnan, MIT).
    - Adopt some version of the TCP Vegas congestion-control mechanism in the Internet. (Easier said than done …)
Conclusion: How To Take Advantage of 10 Gigabit Ethernet?

- **Host-Interface Bottleneck**
  - **Software**
    - A host can only send and receive packets as fast as the OS can process the packets.
  - **Hardware (PC)**
    - PCI I/O bus. 64 bit, 66 MHz = 4.2 Gb/s. **Based on past trends, the I/O bus will continue to be a bottleneck.**

- **Adaptation Bottlenecks**
  - **Flow Control**
    - No adaptation currently being done in any standard TCP.
    - Static-sized window/buffer is supposed to work for both the LAN and WAN. **Solutions exist but are not widely deployed.**
  - **Congestion Control**
    - Adaptation mechanisms will not scale, particularly TCP Reno (although TCP Reno w/ SACK helps immensely). **TCP Vegas? Binomial congestion control?**
Conclusion: How To Take Advantage of 10 Gigabit Ethernet?

- Host-Interface Bottleneck
  - Software
    - A host can only send and receive packets as fast as the OS can process them.
  - Hardware (PCI I/O bus)
    - 64 bit, 66 MHz = 4.2 Gb/s.

- Adaptation Bottlenecks
  - Flow Control
    - No adaptation currently being done in any standard TCP.
    - Static-sized window/buffer is supposed to work for both the LAN and WAN.
  - Congestion Control
    - Adaptation mechanisms will not scale, particularly TCP Reno (although TCP Reno w/ SACK helps immensely).
    - BW problems potentially solvable. Latency? What happens when we go optical to the chip? Solutions exist but are not widely deployed.
    - Based on past trends, the I/O bus will continue to be a bottleneck.

Maybe we can stop the high-end application users from "secretly" despising the networking community. 😊
A Few Recent & Relevant Publications …

• On the Compatibility of TCP Reno and TCP Vegas, Submitted to *INFOCOM 2002*.

Be forewarned! Only the first publication is currently available on-line at [http://www.lanl.gov/radiant](http://www.lanl.gov/radiant). This will be rectified by SC 2001, November 2001.
Relevant Software Distribution (GPL)

• Dynamic Right-Sizing (DRS)
  ➢ In Kernel Space
    ▪ Linux 2.2.x DRS patch implemented over a year ago but “too unpolished” to release.
    ▪ Linux 2.4.x: Plan to release at SC 2001 via CD-ROM as well as via http://www.lanl.gov/radiant
  ➢ In User Space
    ▪ Integration of kernel-level DRS technique into FTP.

• Other software “on the loading dock” to be shrink-wrapped by SC 2001: IRIS, MAGNeT, TICKET.
Acknowledgements: Current Collaborators

- **Indiana University**
  - TICKET: Beta-test site (1/02)
- **Rice University**
  - Network Traffic Characterization
- **University of Illinois at U-C**
  - Ubiquitous Computing
  - Cyber-Security
- **University of Texas at Austin**
  - Fault Tolerance & Self-Healing Clusters via the Network
- **University of Maryland**
  - Analytic Modeling of TCP Protocols

- **Argonne National Laboratory**
  - drsFTP → GridFTP?
- **SDSC / NPACI**
  - TICKET: Beta-test site
  - drsFTP: Beta-test site
- **SLAC**
  - Monitoring and Measurement

- **RLX Technologies**
  - (Commodity) Supercomputing in Small Spaces
- **Quadrics**
  - High-Speed Interconnects
- **United Devices**
  - Cyber-Security for Grids

---

*For more information on our research, go to* [http://www.lanl.gov/radiant](http://www.lanl.gov/radiant)
Potential Partnerships with Industry

• UC-CoRE: UC Communications Research Program
  ➢ Note: Los Alamos National Laboratory and SDSC are operated by the University of California.

• Industry Benefits
  ➢ Immediate leveraging of R&D funds.
  ➢ California and federal tax credits.
  ➢ Access to UC’s & LANL’s world-class faculty and research resources.
  ➢ Expansion of company R&D capacity through partnership with UC.
  ➢ Intellectual property rights.
That’s All Folks!