High-Speed Network Monitoring & Measurement with Commodity Parts

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

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

Adjunct Assistant Professor
Department of Computer & Information Science
The Ohio State University

DOE SciDAC
W. Feng, 4 January 2002, LA-UR-01-6042
Outline

- Who Are We and What Do We Do?
  ➢ So Many Research Directions, So Little Time …
- Network Monitoring & Measurement
  ➢ MAGNeT: Monitor for Application-Generated Network Traffic
    ✓ Design & (Prototype) Implementation
    ✓ MAGNeT Operation: A Look Under the Hood
    ✓ Performance Evaluation
    ✓ Related Work
    ✓ Fun with MAGNeT, i.e., Applications of MAGNeT
    ✓ Conclusion
  ➢ TICKET: Traffic Information-Collecting Kernel with Exact Timing
    ✓ General Overview
    ✓ Comparative Evaluation
    ✓ Conclusion
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 [http://www.lanl.gov/radiant](http://www.lanl.gov/radiant).
Selected Publications

• On the Compatibility of TCP Reno and TCP Vegas. To be submitted to *GLOBECOM 2002*.
Network Monitoring & Measurement

• MAGNeT
  - Monitor for Application-Generated Network Traffic
  - Goals
    - To monitor traffic immediately after being generated by the application (i.e., unmodulated traffic) and throughout the protocol stack to see how traffic gets modulated.
    - To create a library of application-generated network traces to test network protocols.

• TICKET
  - Traffic Information-Collecting Kernel with Exact Timing
  - Goals
    - To provide high-speed and high-fidelity network capture to support research in traffic characterization and to provide insight into future protocol design.
    - To monitor, troubleshoot, or tune production networks.
  - Coincidentally Achieved Goal: Functionally reconfigurable.
Why Monitor Traffic?

• Research & Development
  ➢ Guide the design of routers, e.g., buffer sizes, packet scheduling, active queue management.
  ➢ Provide insight into the development of protocols and/or protocol enhancements.
  ➢ Develop traffic shapers and/or reduce DOS attacks.

• Operations & Management
  ➢ Network tuning.
  ➢ Security monitoring.
  ➢ “Appropriate use” monitoring.
What Good is a MAGNeT?

• Existing Monitors …
  - Focus on specific areas of the stack.
  - Capture traffic after modulation.
  - Produce inaccurate timestamps.
  - Cannot keep up with GigE / 10GigE.
  - … more later …

• Network Models
  - Built on existing traffic traces.
Network Models

- **Traditional Network Models** (1970s to mid-1990s)
  - Source: Poisson-distributed inter-arrivals and file-size distributions.

- **Contemporary Network Model** (mid-1990s to now)
  - Source: Heavy-tailed (e.g., Pareto) inter-arrivals and file-size distributions → Network: Self-similar (or fractal)

- **Problem**: What is the correct model?

- **Solution**: Re-examine traffic traces.

**What is a traffic trace?**
What goes on here???

Problem: Monitoring (adversely) modulated traffic.
Solution: MAGNeT
MAGNeT Design Goals

- Monitoring Traffic (at each layer)
  - To / from applications.
  - Passing through the protocol stack.
  - Entering / leaving the network (like `tcpdump`).
- Fine-Granularity Timestamps
- High Performance, Low Overhead
- Flexibility
  - Events & Protocols Easy to Add
MAGNeT Design Alternatives

- API and Static Library
  - Requires modified applications.
  - Only captures traffic from a single application.
- Shared-Library Hijacking
  - Requires tricky dynamic linking.
  - Only captures application traffic.
- Modified Kernel
  - Requires kernel re-compile.
  - Captures traffic from unmodified applications.

Note: Related research on dynamically instrumented kernel at the University of Wisconsin, Prof. Barton Miller.
MAGNeT Design

**Kernel**
- Record application, stack, and network traffic.
- One-time kernel re-build.
  - No application modifications.
  - No re-compilation of apps.
  - No re-linking required.
- Always available.
- Low overhead.

**User**
- Save only data of interest.
  - Wrapper around specific applications, e.g., FTP.
  - Reduce filter time and storage space.
- Export monitoring service to any application.
- Run by user (or **cron**)
MAGNeT Operation

Kernel

- TCP
- IP
- Network
- magnet_add()
- kernel_buffer

User

- application
- send()
- recv()
- magnet_read
- Disk
Other protocols, events, etc. are easily added with minimal kernel hacking.
MAGNeT Event Records

```c
struct magnet_data {
    void *sockid;
    unsigned long long timestamp;
    unsigned int event;
    int size;
    union magnet_ext_data data;
};
```

Minimal Saved State: 24 bytes/event
MAGNeT Extra Data (Headers)

TCP
- Source Port
- Destination Port
- Send Window (snd_wnd)
- Smoothed Round Trip Time (srtt)
- Packets in flight
- Retransmitted packets
- Slow Start Threshold (snd_ssthresh)
- Congestion Window (snd_cwnd)
- Current Receiver Window (rcv_wnd)
- Send sequence number (write_seq)
- Sequence on top of receive buffer (copied_seq)
- Flags (SYN, FIN, PSH, RST, ACK, URG)

Size: 64 bytes / packet

IP
- Version
- Type of Service
- ID
- Fragment Offset
- Time To Live
- Protocol

Size: 8 bytes / packet
MAGNeT on Linux

- Linux 2.4.x.
  - Large installed base.
  - Source code readily available.
- Kernel- and User-Space Implementation
  - Minimize kernel overhead
  - Communicate via shared memory.
- Architecture Independent
  - Endian-aware.
  - Use generic kernel operations
    (e.g., getting CPU cycle counter)
  
  Alpha-tested on i386 & PowerPC architectures.
MAGNeT uses the timestamp field as a synchronization flag.
MAGNeT Experiments

- Two Machines: Dual 400-MHz Pentium IIs
- Networks
  - 100-Mb/s NetGear NIC.
  - 1000-Mb/s Alteon AceNIC.
- Configurations
  1. Linux 2.4.3 on sender and receiver (baseline).
  2. Linux 2.4.3 with (inactive) MAGNeT.
  3. Configuration 1 with `magnet-read` on receiver.
  4. Configuration 1 with `magnet-read` on sender.
  5. Configuration 1 with `tcpdump` on receiver.
  6. Configuration 1 with `tcpdump` on sender.
- Workload: `netperf` on sender, saturating the network.
- Events Monitored: App send/recv, TCP – IP, IP – data link
CPU Utilization

% Increase in CPU Utilization

100 Mb/s

1000 Mb/s

MAGNeTized
MAGNeT/recv
MAGNeT/send
tcpdump/recv
tcpdump/send
MAGNeT fails to record an event in only one case:
The kernel buffer is full when an event occurs.

Ways to Reduce MAGNeT Event Loss

1. Increase kernel buffer size
   - More buffer = More events before loss
   - Buffer is pinned in memory:
     More buffer = Less available physical RAM

2. Reduce `magnet_read` sleep time
   - Less delay = Less time for buffer to fill
   - Less delay = more CPU overhead
By comparison, `tcpdump/libpcap` loss rate is 15%
MAGNeT was motivated from a belief that the networking stack (i.e., TCP) adversely modulates the actual application traffic patterns.

Is this really the case?

An obvious (but simple) example:

- FTP Linux 2.2.18 kernel from Los Alamos to Dallas with MAGNeT running on the sender …
Modulated Traffic

![Graph showing modulated traffic pattern]

- **Size (Bytes)** vs **Time (Seconds)**
- Delivered to network: solid line
- Application send() call: dashed line
Really Modulated Traffic

FTP
10K
TCP₁
1480
TCP₂
1015
TCP₃
919
TCP₄
760
Related Work

- Monitors
  - tcpdump, **tcpdump**, Coral Software Suite
  - RMON
  - TCP Kernel Monitor
  - tcpmon

- Traffic Repositories
  - Internet Traffic Archive
    - Low-speed, low-utilization aggregate traffic
    - Oftentimes over shared 10-Mb/s Ethernet.
  - Internet Traffic Data Repository
Fun with MAGNeT

- Potential Uses of MAGNeT
  - Collect **real** application traffic traces.
    - No modulation by existing protocols.
  - Debug & tune protocol implementations (or kernel events in general)
    - Run-time protocol state information easily available.
  - Provide information to network-aware applications.
  - Support security scanning.
    - Unobtrusive, high-fidelity network monitoring on a per-machine basis.
    - Campus-wide monitoring with no central bottleneck.
  - Analyze network traffic
    - Poisson, self-similar (fractal), multi-fractal?
Future Work

• Collection of traces of application-generated traffic across campus.
• Run-time vs. compile-time configuration.
• Kernel-thread implementation?
  ➢ Suggestion by Andrea Arcangeli (SuSeLinux)
• Automatic handling of CPU clock-rate changes (*a la* Intel SpeedStep).
MAGNeT Conclusion

• Existing traces cannot provide protocol-independent insight.
  - Modulation effects can be substantial.
  - Existing (modulated) traffic traces may be misleading.
• MAGNeT can capture protocol-independent traffic traces (as well as kernel events in general)
  - It provides a flexible, low-overhead infrastructure.
  - It can be used throughout the network stack.
• Status
  - Alpha prototype has been completed and tested.
  - GPL software distribution to follow once approval is received.
Motivation for TICKET

• `tcpdump` & `tcpdump-based` Monitors
  - Unable to monitor *and* record traffic at gigabit-per-second (Gb/s) speeds and nanosecond granularity, particularly with low-end commodity parts.
  - Field test of `tcpdump` in February 2001:
    ~300 Mb/s with $O$(msec) timestamp granularity.

• Commercial Monitors, e.g., NetScout nGenius
  - Able to keep up at gigabit-per-second speeds *but* with $O$(sec) granularity and with a $200K$ price tag.
  - Goal: Design a high-speed (Gb/s), high-fidelity (nanosecond granularity), and cost-efficient ($2K$) network monitor.
Comparison

• Price
  - TICKET: $2K
  - tcpdump: $1K
  - Commercial Offering (e.g., NetScout nGenius): $200K

• Performance
  - TICKET: 600-1000 Mbps (problem with multicast back-pressure)
  - tcpdump: 300 Mbps
  - Commercial Offering: 2000 Mbps

• Price/Performance
  - TICKET: $2.00-$3.33 / Mbps
  - tcpdump: $2.50 / Mbps
  - Commercial Offering: $165.00 / Mbps
Comparison

• Granularity of Measurements
  ➢ TICKET: \(O(\text{ns})\).
  ➢ tcpdump: \(O(\text{ms})\).
  ➢ Commercial Offerings: \(O(\text{s})\).

• Flexibility
  ➢ TICKET: Can be configured to be a network intrusion detector and a WAN emulator among other things.
  ➢ tcpdump and commercial offerings only monitor and measure traffic.

• Boot Time
  ➢ TICKET: 10 seconds
  ➢ tcpdump: 120-180 seconds
  ➢ Commercial Offerings: ???
TICKET Conclusion

• The current generation of network monitors cannot simultaneously address the following issues:
  - High speeds, e.g., Gb/s.
  - High fidelity, e.g., nanoseconds.
  - Low cost, e.g., $1K-$2K.
  - Versatility, e.g., able to function as more than a monitor.

• Status
  - Alpha prototype has been completed and tested.
  - GPL software distribution to follow once approval is received.
  - Patent to be filed.