

Performance Measurement DREN Activities

JET Workshop

13 April 2004

Phillip Dykstra

Chief Scientist

WareOnEarth Communications Inc.

phil@sd.wareonearth.com

DREN AMP

- Descendent of NLANR AMP
- Focus on testing and debugging high performance flows
- Test points with known high performance and no rate limits
- CDMA clocks for one-way testing
- End-user services such as nuttcp and owamp

DREN AMP Generations

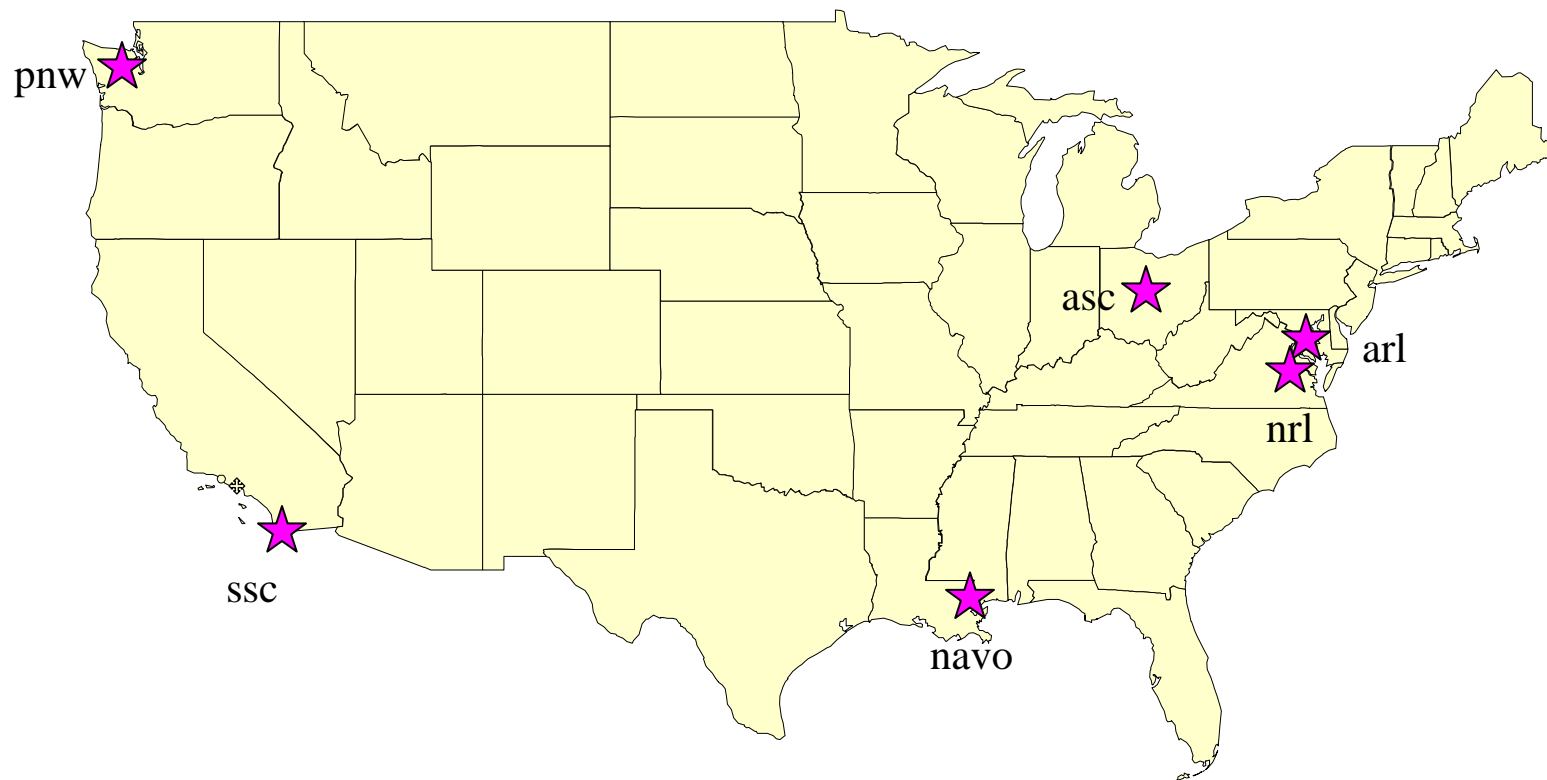
- 1st Generation, 2000
 - PII 400MHz, 64MB PC100, 32/33 PCI, FreeBSD, 100BaseT only
- 2nd Generation, 2001-2002
 - PIII 800MHz and 933 MHz, 512MB PC133, 64/66 PCI, Linux, OC3, OC12, GigE (GA620)
- 3rd Generation, 2003
 - P4 Xeon 2.6GHz, 1GB PC1600, 64/133 PCI-X, GigE (Intel PRO/1000 XF SX, BCM5701 copper)

3rd Generation AMP Systems

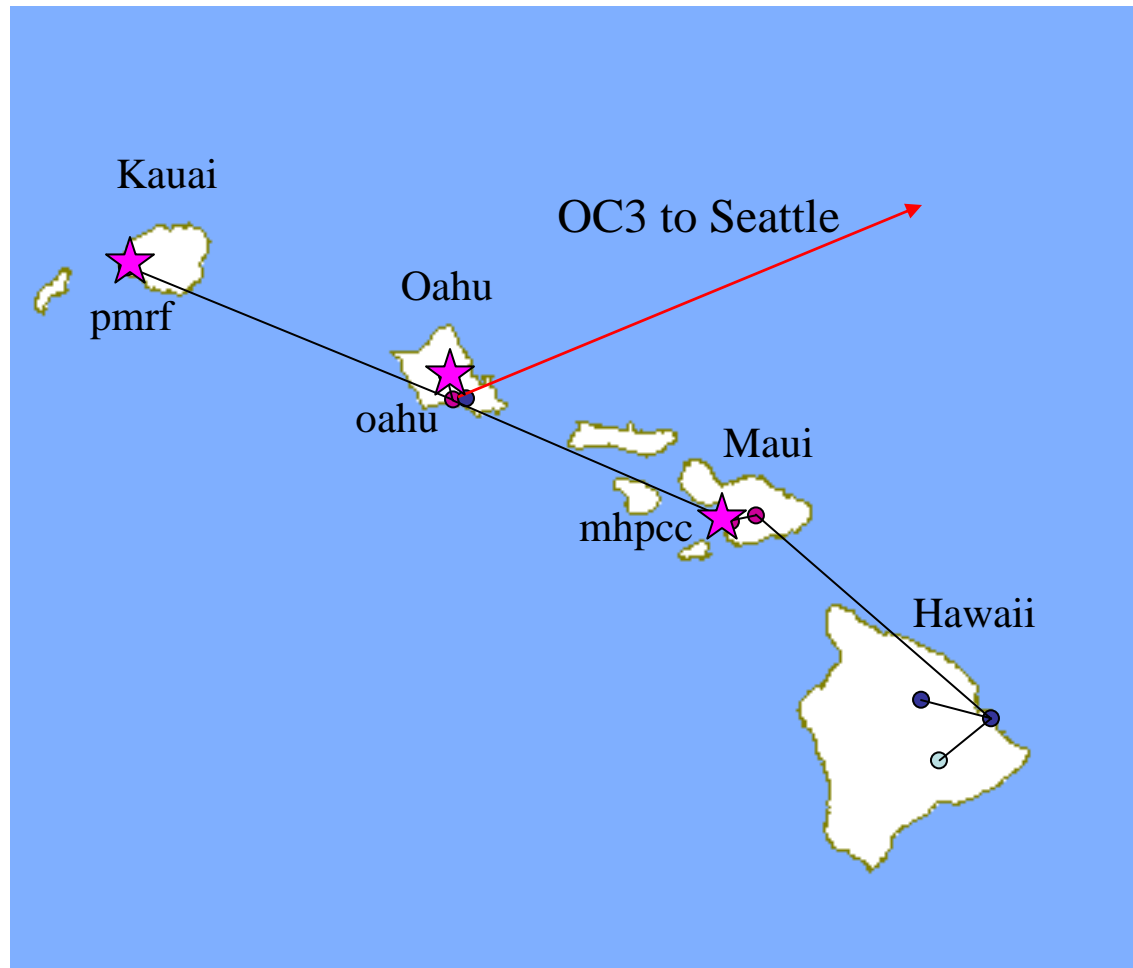
- Deployed 10 new systems since Feb 2003
 - 7 CONUS, 3 Hawaii
- Dell PowerEdge 2650 based
 - 2.6 GHz Xeon, 1 GB DDR memory
 - GigE: Broadcom copper, Intel Pro/1000 XF
 - ATM: Marconi HE622 OC12



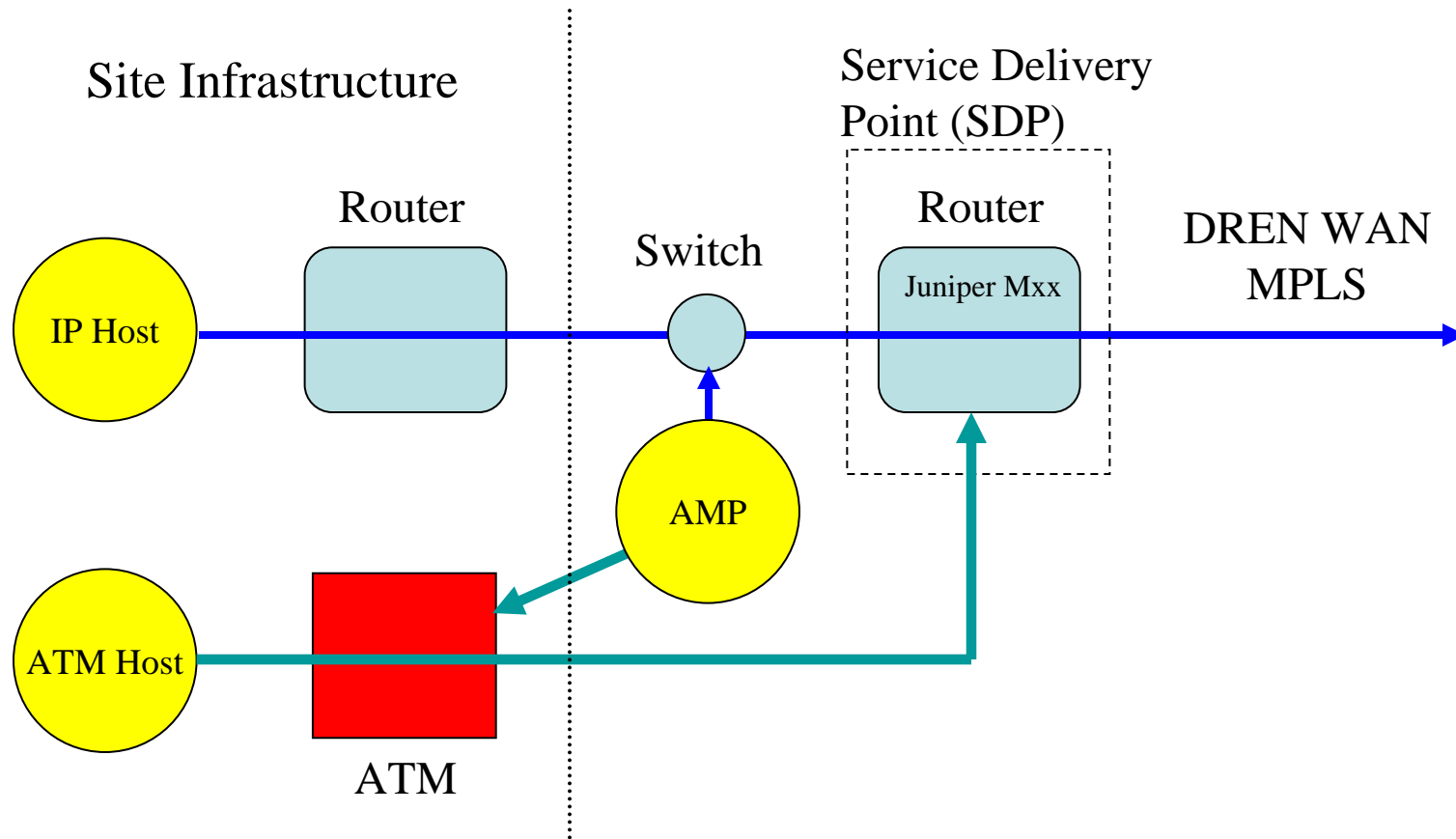
CONUS OC12 AMP Systems



Hawaii OC12 AMP Systems

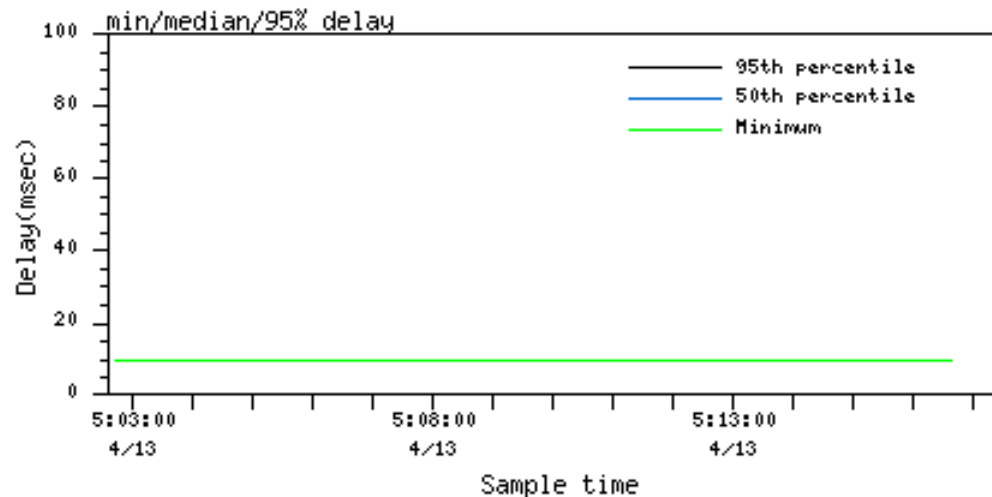


AMP System Placement



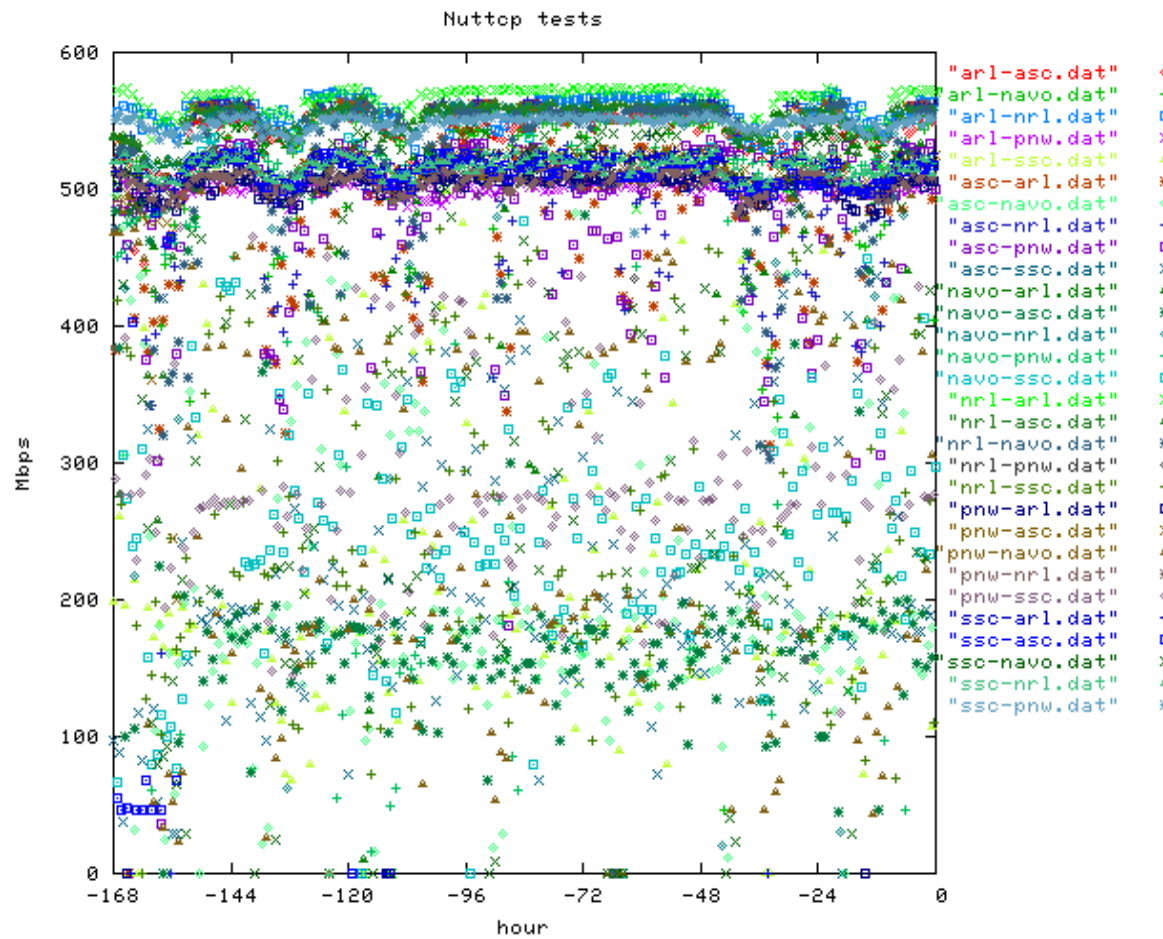
AMP Measurement

- Delay and Loss measurements have become pretty boring
 - Have you looked at an Abilene graph lately?

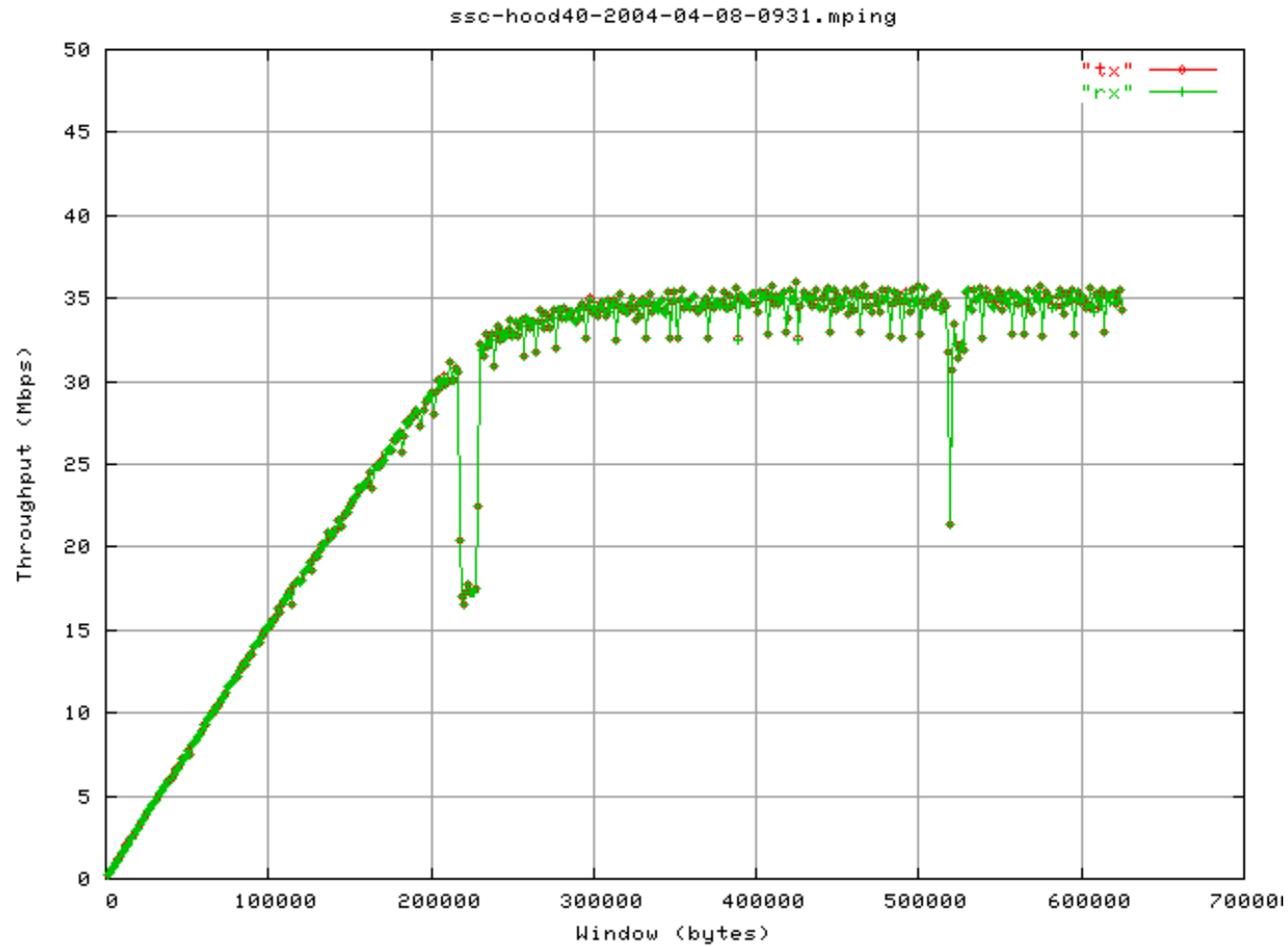


TCP WAN Tests

July 23-29 2003



Mping Tests

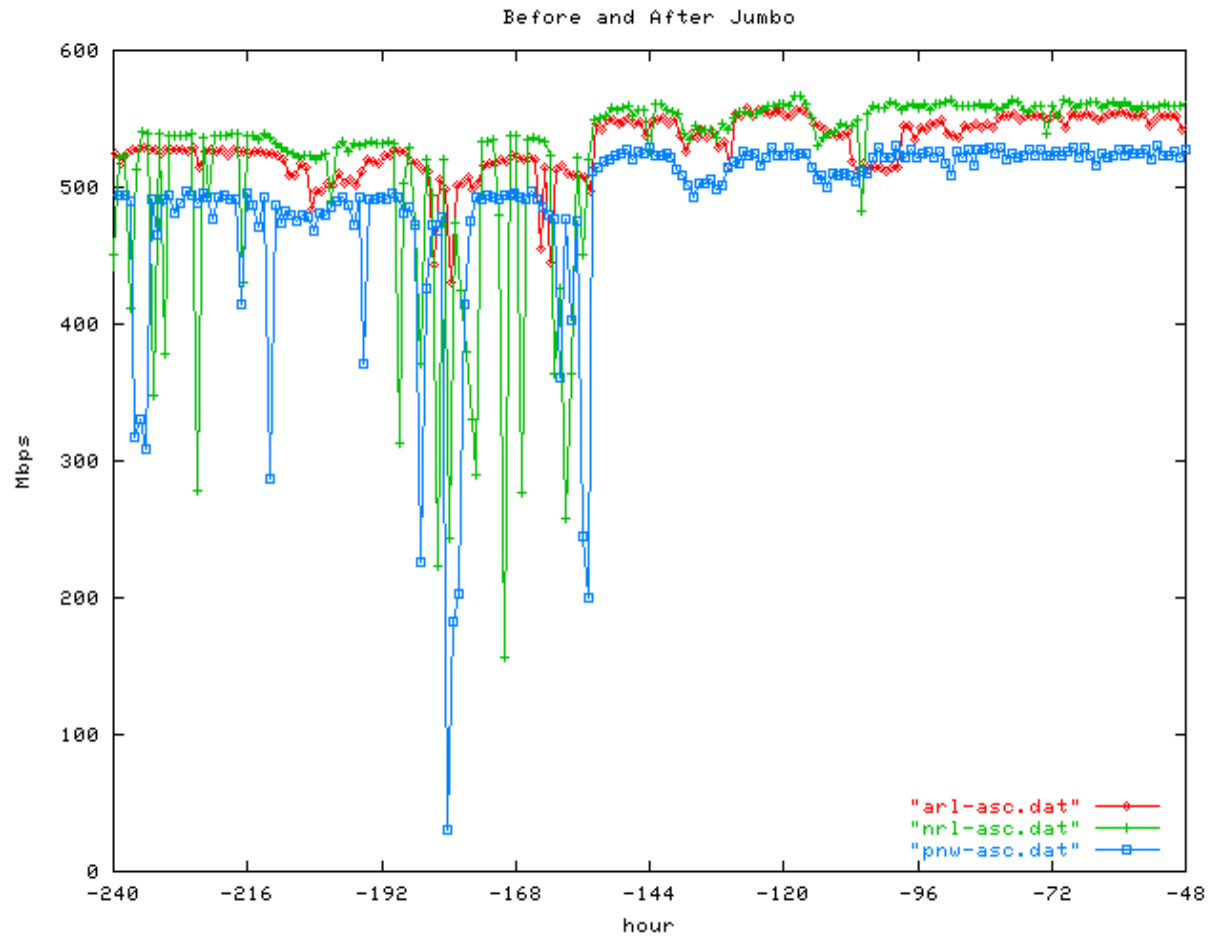


MTU Increase

- IP Maximum Transmission Unit (MTU) on the DREN WAN is now 9146 bytes
- After much debate, exactly 9000 is being used to the sites with GigE interfaces
 - Sites choose 1500, 4470, or 9000, others by exception
- Sites are encouraged to support 9000 on their GigE infrastructures

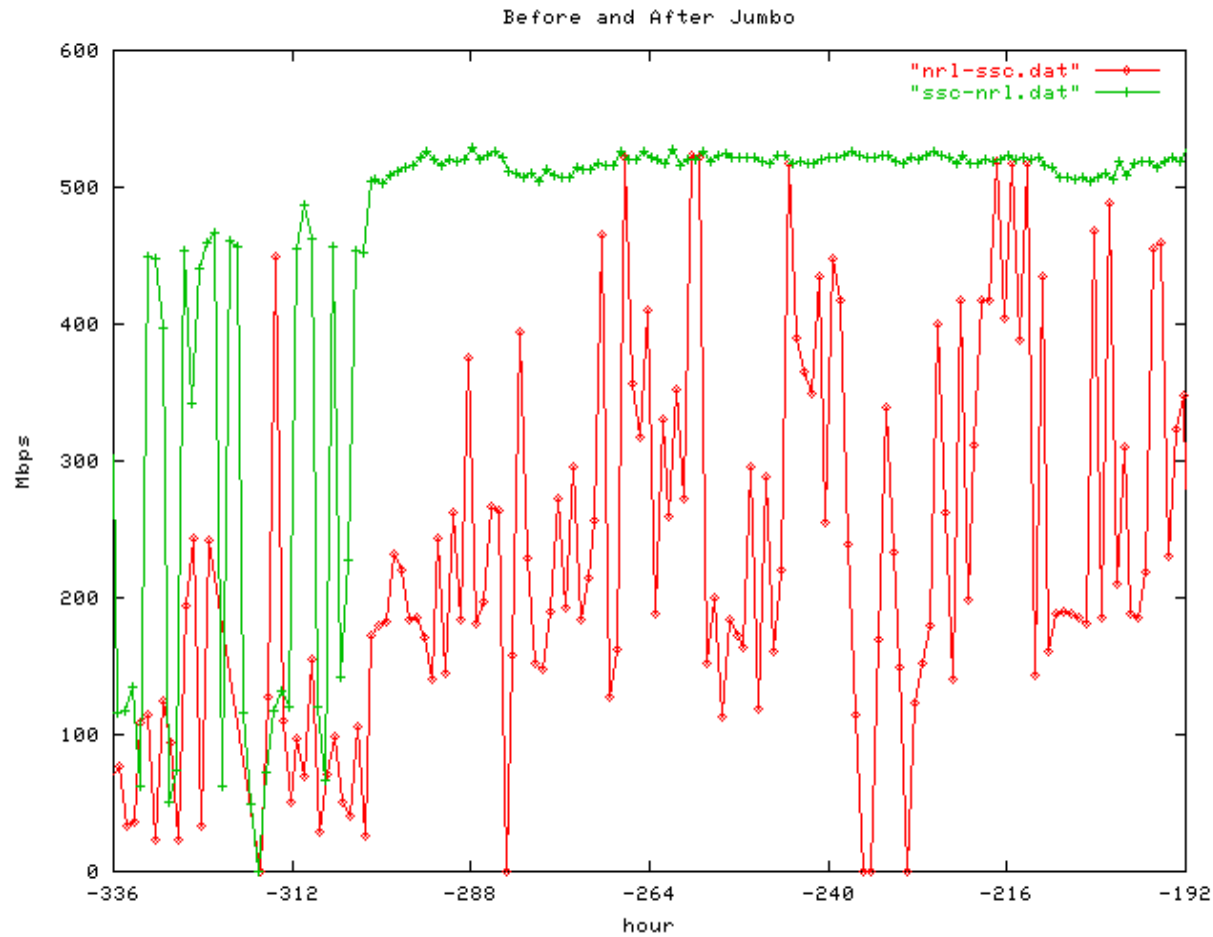
Impact of Jumbo, into ASC

Jumbo enabled 23 July 2003



Impact of Jumbo, SSC to NRL

Jumbo enabled 17 July 2003



nuttcp

- <ftp://ftp.lcp.nrl.navy.mil/pub/nuttcp>
- Bill Fink's descendent of ttcp, nttcp
- Servers return all results to clients
 - No need for remote accounts or logs
- Third party support
- Highly recommended!

Run an nuttcp test

```
[phil@damp-mhpcc phil]$ nuttcp -v -t -T10 -w6m damp-rome
nuttcp-t: v3.6.2: socket
nuttcp-t: buflen=65536, nstream=1, port=5001 tcp -> damp-rome
nuttcp-t: time limit = 10.00 seconds
nuttcp-t: connect to 138.18.25.6
nuttcp-t: send window size = 12288000, receive window size = 65535
nuttcp-t: 47.334 MB in 9.99 real seconds = 4851.47 KB/sec = 39.7433 Mb/s
nuttcp-t: 758 I/O calls, msec/call = 13.50, calls/sec = 75.87
nuttcp-t: 0.0user 0.1sys 0:10real 1% 0i+0d 0maxrss 0+0pf 0+0csw

nuttcp-r: v3.6.2: socket
nuttcp-r: buflen=65536, nstream=1, port=5001 tcp
nuttcp-r: accept from 138.18.203.3
nuttcp-r: send window size = 65535, receive window size = 4194304
nuttcp-r: 47.334 MB in 11.52 real seconds = 4208.18 KB/sec = 34.4734 Mb/s
nuttcp-r: 32895 I/O calls, msec/call = 0.36, calls/sec = 2855.96
nuttcp-r: 0.0user 0.3sys 0:11real 3% 0i+0d 0maxrss 1+0pf 0+0csw
```

Reverse the Direction (-r)

```
[phil@damp-mhpcc phil]$ nuttcp -v -r -T10 -w6m damp-rome
nuttcp-r: v3.6.2: socket
nuttcp-r: buflen=65536, nstream=1, port=5001 tcp
nuttcp-r: accept from 138.18.25.6
nuttcp-r: send window size = 65535, receive window size = 12288000
nuttcp-r: 45.511 MB in 10.53 real seconds = 4425.60 KB/sec = 36.2545 Mb/s
nuttcp-r: 31256 I/O calls, msec/call = 0.34, calls/sec = 2968.18
nuttcp-r: 0.0user 0.1sys 0:10real 1% 0i+0d 0maxrss 1+0pf 0+0csw

nuttcp-t: v3.6.2: socket
nuttcp-t: buflen=65536, nstream=1, port=5001 tcp -> 138.18.203.3
nuttcp-t: time limit = 10.00 seconds
nuttcp-t: connect to 138.18.203.3
nuttcp-t: send window size = 4194304, receive window size = 65535
nuttcp-t: 45.511 MB in 9.99 real seconds = 4664.11 KB/sec = 38.2084 Mb/s
nuttcp-t: 729 I/O calls, msec/call = 14.04, calls/sec = 72.96
nuttcp-t: 0.0user 0.4sys 0:10real 4% 0i+0d 0maxrss 0+0pf 0+0csw
```


Watch It Run (-i1)

```
[phil@damp-mhpcc phil]$ nuttcp -v -i1 -t -T10 -w6m damp-rome
nuttcp-t: v3.6.2: socket
nuttcp-t: buflen=65536, nstream=1, port=5001 tcp -> damp-rome
nuttcp-t: time limit = 10.00 seconds
nuttcp-t: connect to 138.18.25.6
nuttcp-t: send window size = 12288000, receive window size = 65535
0.0939 MB / 0.99 sec = 0.7935 Mbps
1.0675 MB / 1.00 sec = 8.9548 Mbps
4.1704 MB / 1.00 sec = 34.9849 Mbps
4.8249 MB / 1.00 sec = 40.4764 Mbps
4.9879 MB / 1.00 sec = 41.8433 Mbps
4.9796 MB / 1.00 sec = 41.7737 Mbps
4.9768 MB / 1.00 sec = 41.7505 Mbps
4.9837 MB / 1.00 sec = 41.8086 Mbps
4.9879 MB / 1.00 sec = 41.8432 Mbps
4.9824 MB / 1.00 sec = 41.7969 Mbps
```

Being Brief (-b)

```
[phil@damp-mhpcc phil]$ nuttcp -t -T10 -w6000 -b damp-rome  
47.291 MB / 12.09 sec = 32.8137 Mbps 1 %TX 3 %RX
```

Testing With UDP (-u -R)

```
damp-kirt$ nuttcp -v -t -T1 -u -R30m damp-ssc2
```

```
nuttcp-t: v3.1.9: socket
```

```
nuttcp-t: buflen=8192, nstream=1, port=5001 udp -> damp-ssc2
```

```
nuttcp-t: time limit = 1 second
```

```
nuttcp-t: rate limit = 30.000 Mb/s
```

```
nuttcp-t: send window size = 65535, receive window size = 65535
```

```
nuttcp-t: 3.578 MB in 1.00 real seconds = 3662.08 KB/sec = 29.9997 Mb/s
```

```
nuttcp-t: 464 I/O calls, msec/call = 2.21, calls/sec = 463.76
```

```
nuttcp-t: 0.5user 0.4sys 0:01real 100% 0i+0d 0maxrss 0+0pf 0+0csw
```

```
nuttcp-r: v3.1.9: socket
```

```
nuttcp-r: buflen=8192, nstream=1, port=5001 udp
```

```
nuttcp-r: send window size = 65535, receive window size = 65535
```

```
nuttcp-r: 0.609 MB in 2.02 real seconds = 309.21 KB/sec = 2.5331 Mb/s
```

```
nuttcp-r: 82.98% data loss
```

```
nuttcp-r: 80 I/O calls, msec/call = 25.83, calls/sec = 39.64
```

```
nuttcp-r: 0.0user 0.0sys 0:02real 0% 0i+0d 0maxrss 0+0pf 0+0csw
```

Starting an nuttcp server

- `nuttcp -S`
- Allows remote users to run tests to/from that host without accounts

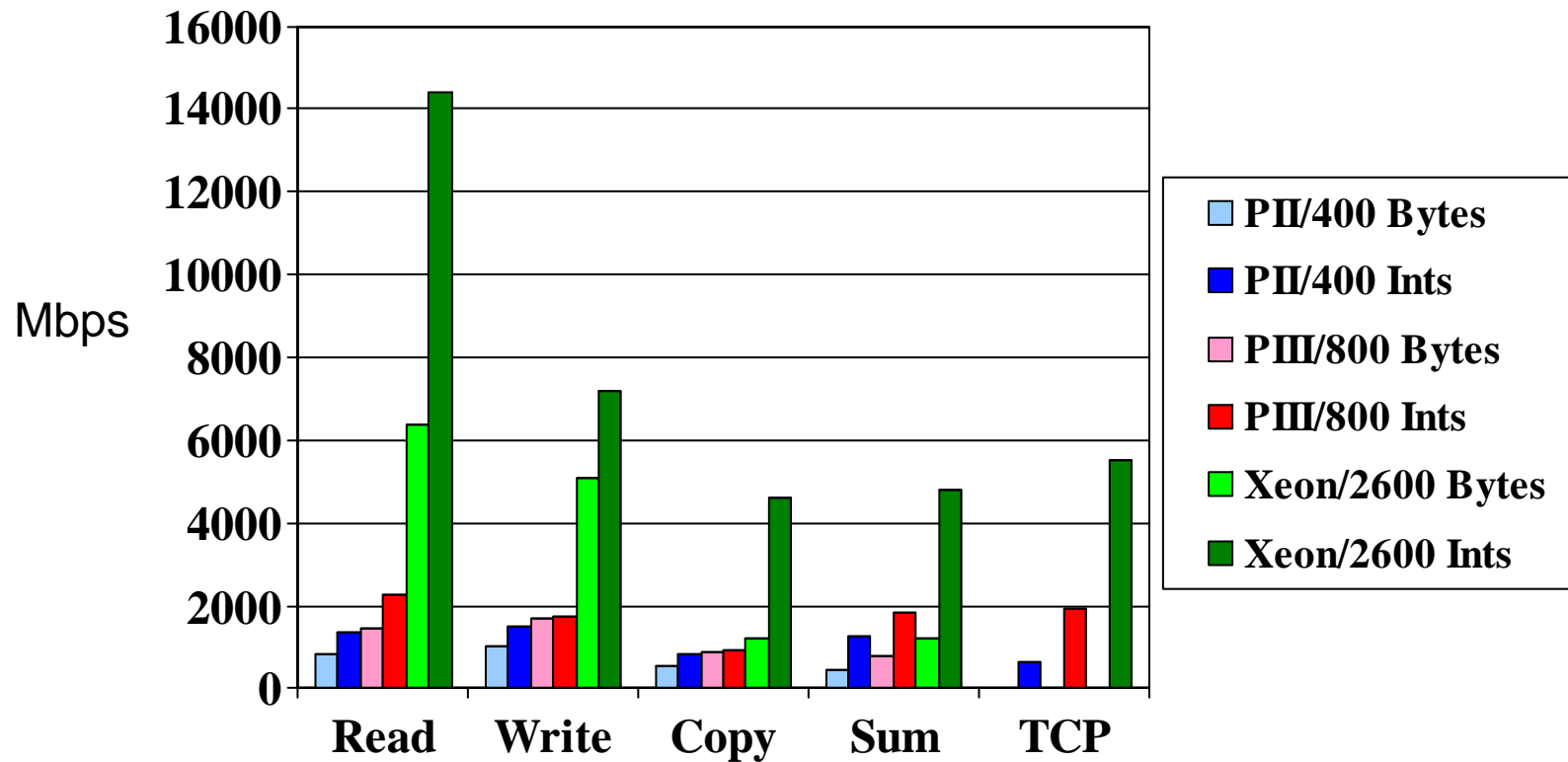
10GigE AMP Systems

- 10GigE NICs in five AMPs by June 2004
 - Intel of S2io TBD
- Handles OC48, but not full 10 Gbps
 - PCI-X 1.0 ~8 Gbps limit
- Hope to have a PCI Express solution by Fall 2004 for full 10 Gbps

High Performance Interconnects

- Box-to-Box
 - 10GigE
 - InfiniBand – SANS / large data
- Board-to-Board
 - PCI Express – Pentium/PCI legacy (low cost)
 - RapidIO Serial
- Chip-to-Chip
 - HyperTransport – Apple, PowerPC/AMD/MIPS
 - RapidIO Parallel – embedded/deterministic

AMP System Memory Bandwidth



Performance Challenges

- Full bandwidth 10GigE
 - Generate, test, encrypt, monitor (NID)
- Making everything work with IPv6
- How to live without “QoS”
 - TCP without large queues!
 - Priority without starvation
- Getting sites to tune systems
 - Or not: auto tune, UDP tools, abstract storage

Performance Challenges

- Jumbo larger than 9000
- Improving routing and reducing latency between nets
 - IDRPs anyone?
- Measuring and debugging Internet performance
 - Would love to have some test systems at .com sites