ARP Traffic Study

Jim Rees, Manish Karir

Research and Development
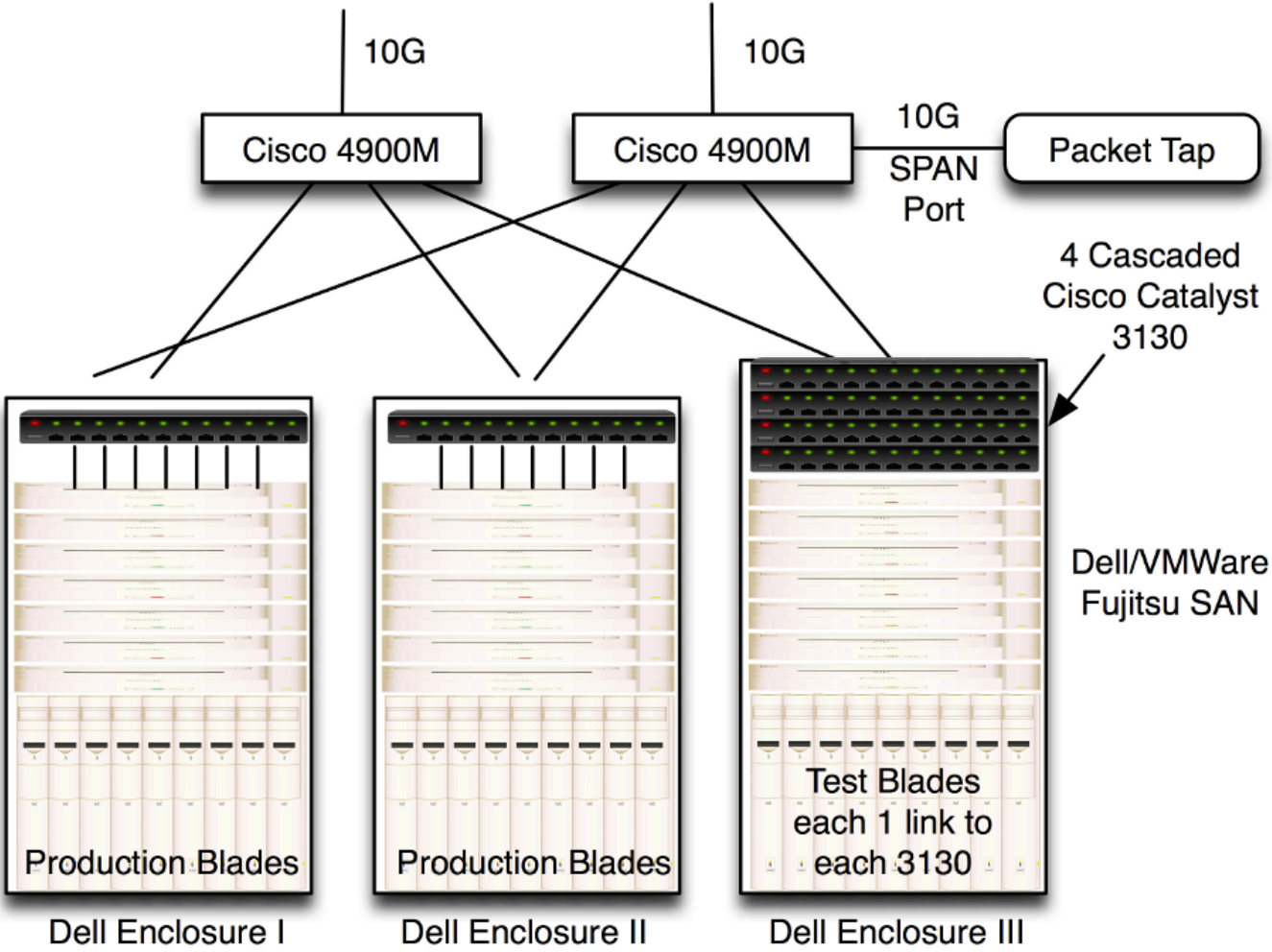
Merit Network Inc.

Outline

- Goals
- Experiment Setup and Architecture
- Experiments
  - Scanning
  - Number of hosts
  - Traffic load/patterns
  - Host failures
  - VM Migrations
  - Emulator
- Conclusions

ARP/ND Traffic Study

- Attempt to understand ARP behavior under various conditions

- Recreate various environments and observe ARP/ND traffic

- Build a model of ARP/ND based on experiments and collected data

- Build scalable ARP/ND emulator for large scale experiments which can mimic various environments for evaluating software/protocol impact, and perhaps proposed solutions

Network Topology

10G   10G

Cisco 4900M   Cisco 4900M   10G   Packet Tap
                                          SPAN
                                          Port

                                          4 Cascaded
                                          Cisco Catalyst
                                          3130

                                          Dell/VMWare
                                          Fujitsu SAN

                          Test Blades
                          each 1 link to
                          each 3130

Production Blades   Production Blades

Dell Enclosure I   Dell Enclosure II   Dell Enclosure III

Experiment Setup

- VMWare Cloud Director
  - Resource Pool – 100GHz, 200GB RAM, 3TB Disk
  - 60 Vapps with 10 VMs each
  - Each VM 4G disk 256MB RAM – Centos 5.5
  - Single flat network topology /22 network block
  - Roughly 500 VMs used in tests
- Herding the bot
  - Write command/control agent/controller software
  - Allows us to issue commands to all bots to start/stop services
 - Write traffic generator software to generate internal and external traffic load via a traffic matrix

Traffic Generator

- Runs on up to 500 hosts
- Each host listens for commands from controller and also runs web server
- Bot controller sends UDP broadcast message hosts to start/stop (no ARP traffic)
- Each host has list of other hosts to contact
- For each host on list:
  - wget https://host (v4/v6)
  - sleep(n seconds - configurable)
- Host list updated on each host prior to experiment

Initial Experiments

- Effect of external scans
- Effect of number of hosts and relative traffic load
- Effect of machine failures
- Effect of traffic load with same number of hosts
- Effect of VM migrations
- IPv6 neighbor discovery behavior (for all above)
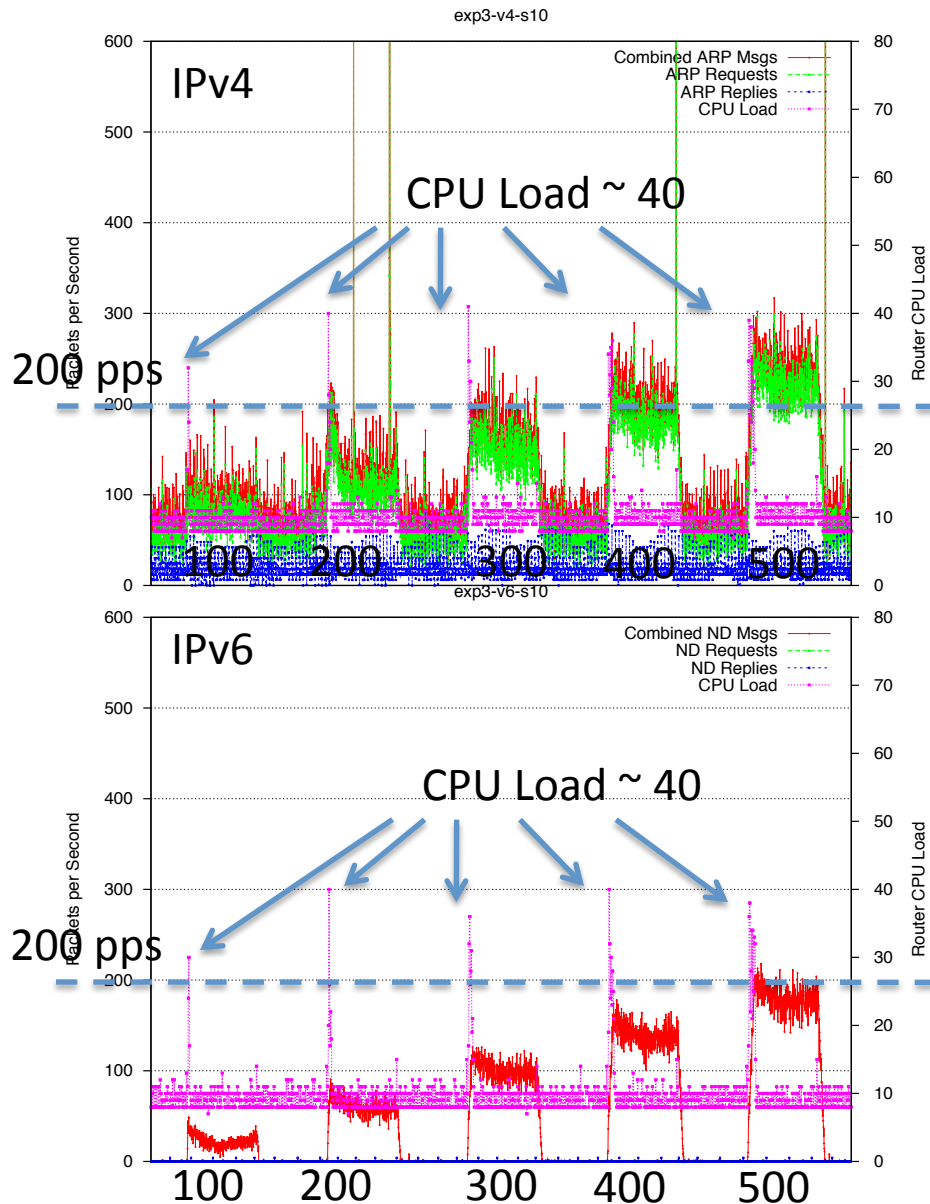
I: Effect of External Scans
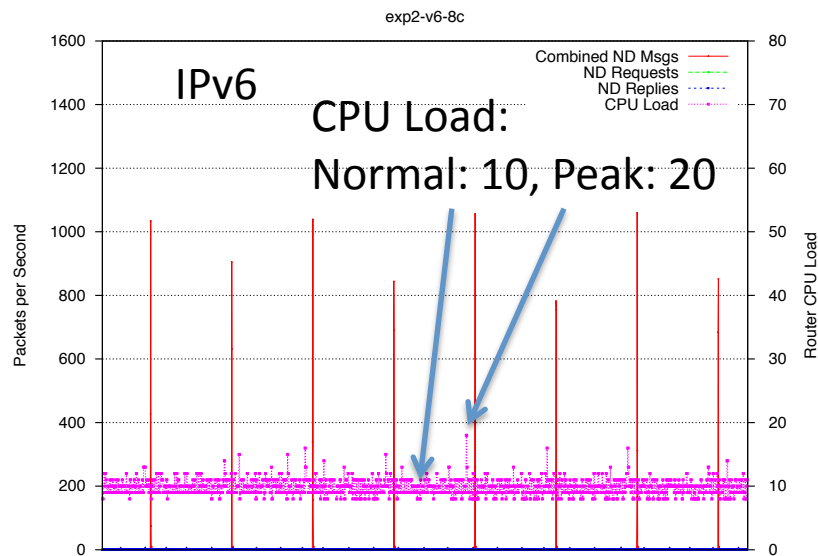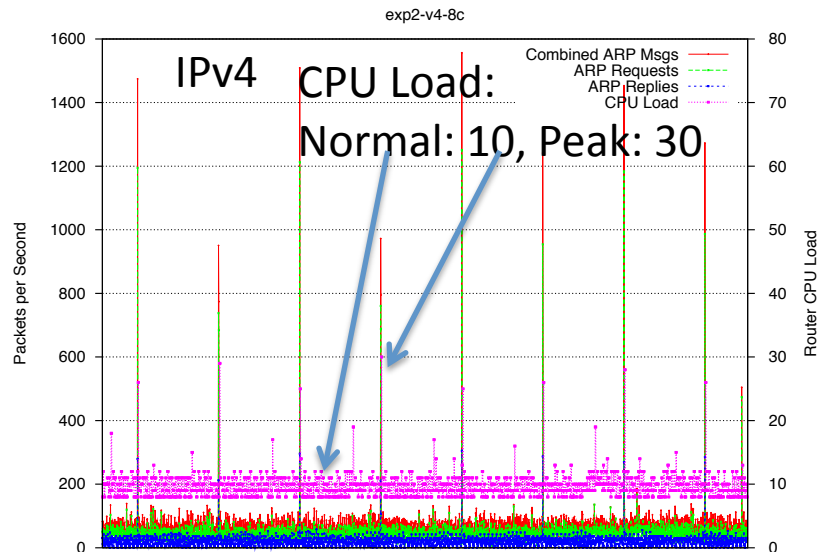
- nmap scans from outside the network

- Measure ARP/ND traffic spikes

- Monitor CPU load on access switch

- Possible to cause transient spikes in ARP/ND traffic

- Effect on CPU load is greater for ARP than ND

I: Effect of External Scans (single)

CPU Load

Combined ARP/ND Traffic Rate
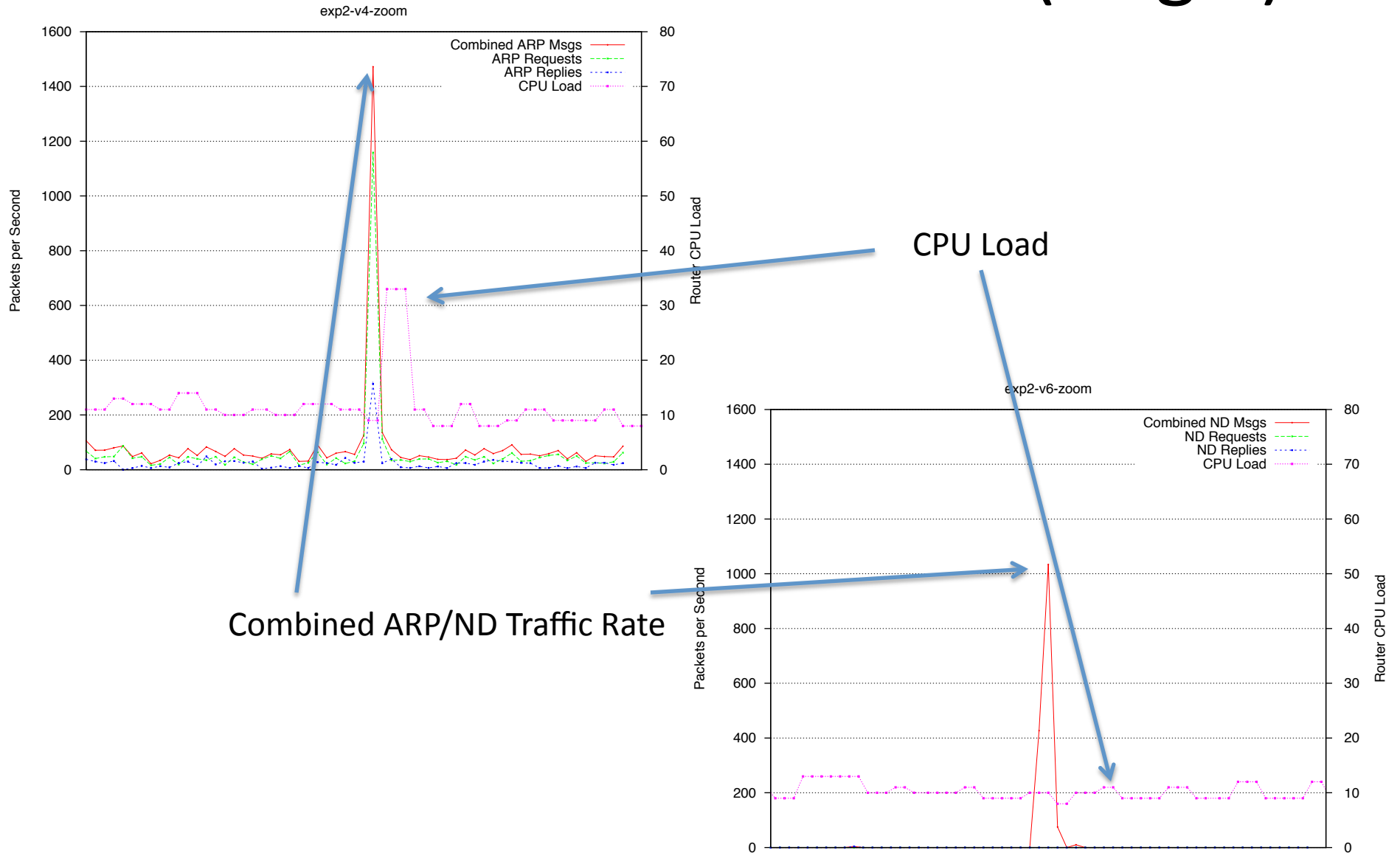
II: Effect of Number of Hosts

- Starting up up to 500 hosts 100,200,300,400,500

- Stop all hosts in between batch starts

- Measure startup ARP/ND traffic and CPU Load

- Increase overall traffic load roughly 5x from 100 – 500 hosts

- Initial spike in CPU load similar for both v4/v6

- Roughly linear increase in ARP/ND traffic

- Higher ARP traffic rate as compared with ND

IIb: Effect of Traffic

exp5-v4

exp5-v6
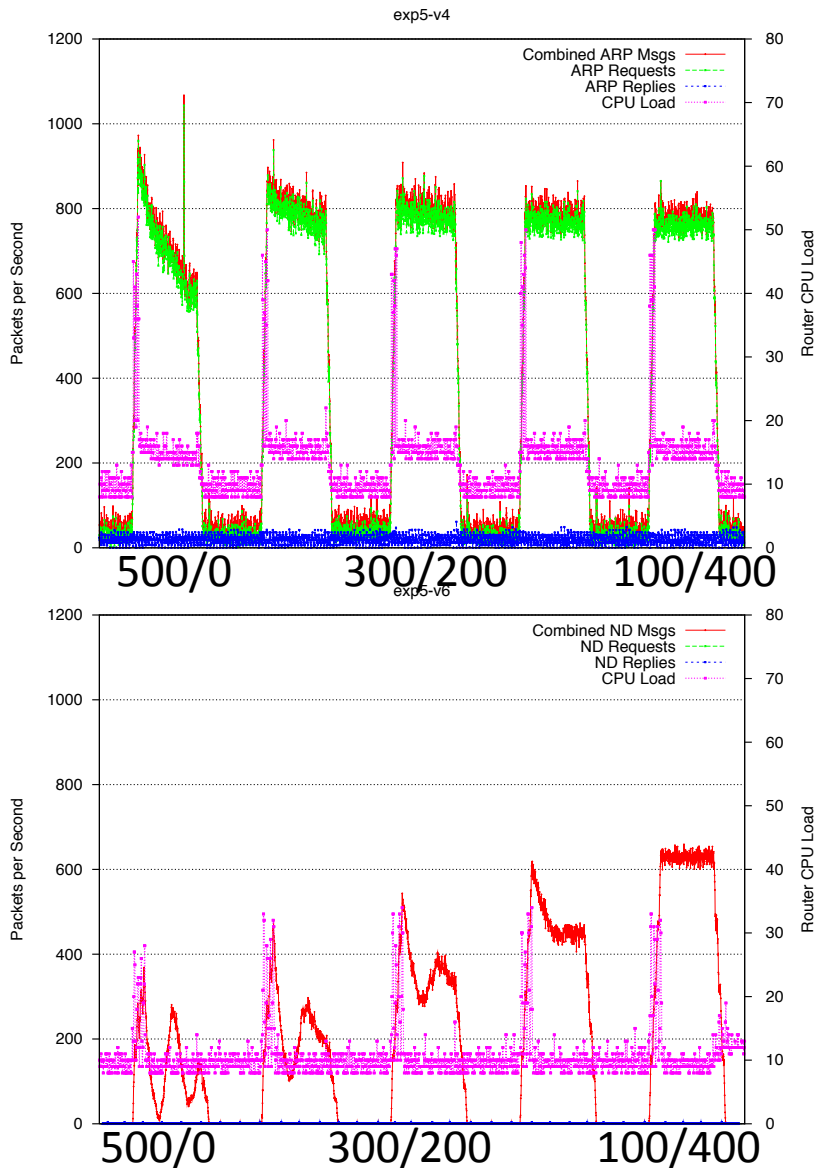
- Starting up 500 hosts
- Hosts attempt to contact varying number of active/inactive hosts (500/0, 400/100, 300/200, 200/300, 100/400)
- Highest possible rate zero sleep
- Stop all hosts in between changes to traffic pattern
- Measure startup ARP/ND traffic and CPU Load
- Initial spike in CPU load slightly higher for both ARP
- CPU load spike roughly same across traffic patterns
- Roughly linear increase in ARP/ND traffic
- Higher ARP traffic rate as compared with ND

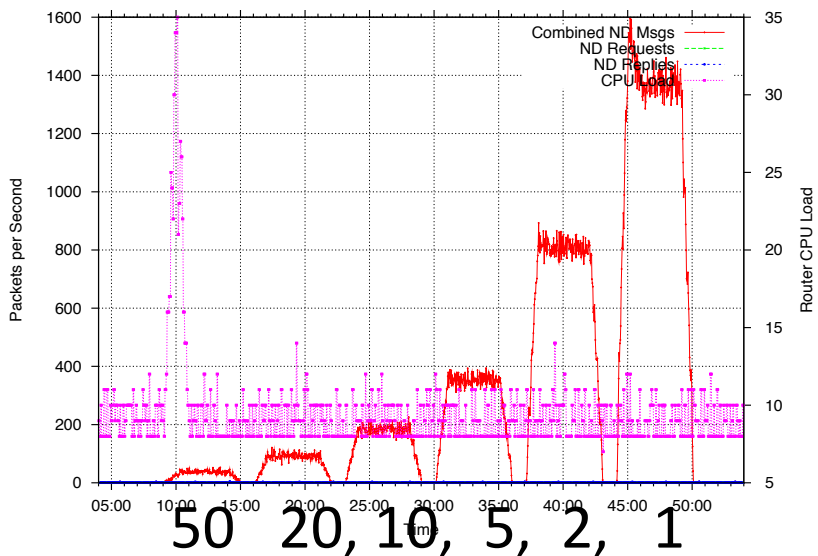# IIc: Effect of Number of Hosts/Traffic
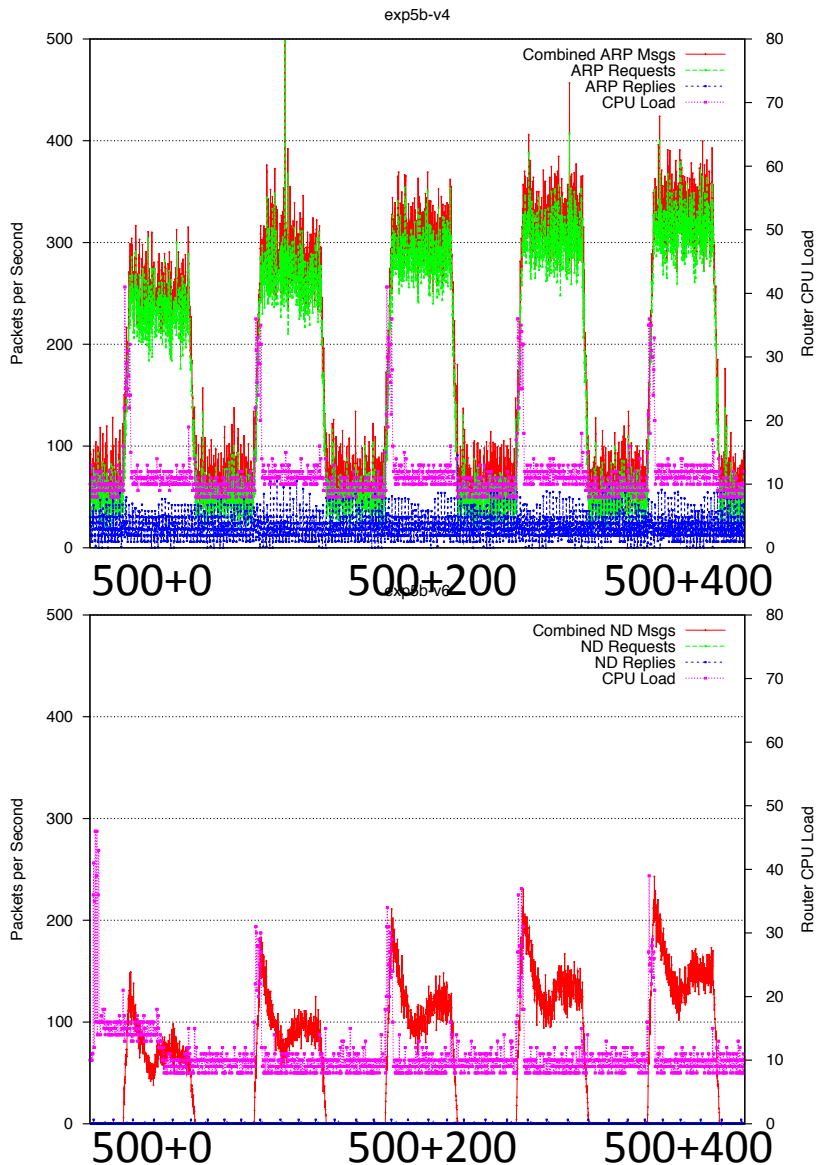
- Starting up all 500 hosts
- Contact list includes all 500 hosts plus a varying number of inactive hosts(0, 100, 200, 300, 400)
- Sleep 2 seconds between connects
- Stop all hosts in between experiments
- Measure startup ARP/ND traffic and CPU Load
- Initial spike in CPU load similar for both v4/v6
- Roughly linear increase in ARP/ND traffic
- Slight increase in CPU load for v6
- Higher ARP traffic rate as compared with ND
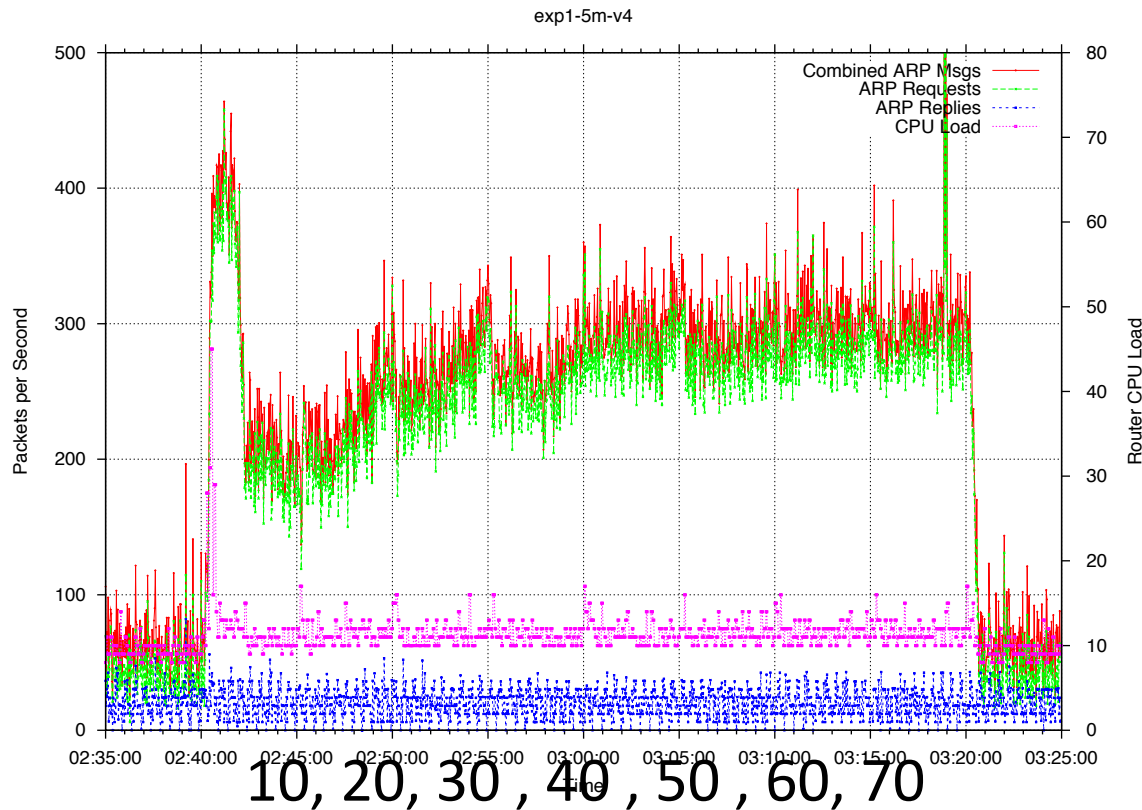
IId: Effect of Traffic Load
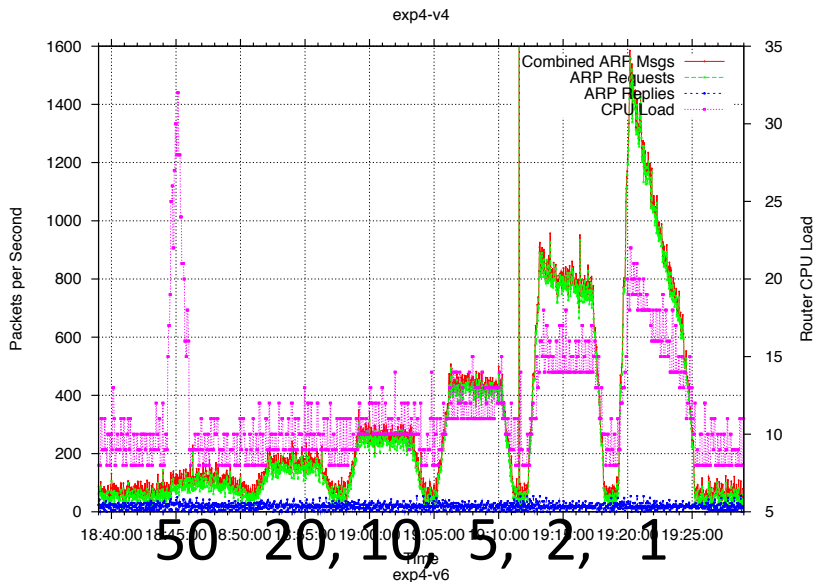
- Start Traffic generator on 500 hosts
- Vary the amount of load by reducing time between communication events (50, 20, 10, 5, 2, 1 seconds)
- Measure ARP/ND Traffic and CPU load on switch
- ARP/ND increases roughly linearly across experiments

# III: Effect of Machine Failures

exp1-5m-v4

- Start Traffic generator on 400 hosts, 80 targets
- Shutdown targets in groups of 10
- Measure ARP traffic and CPU Load
- ARP traffic increases as number of failed machines increases

IV: Effect of VM Migrations
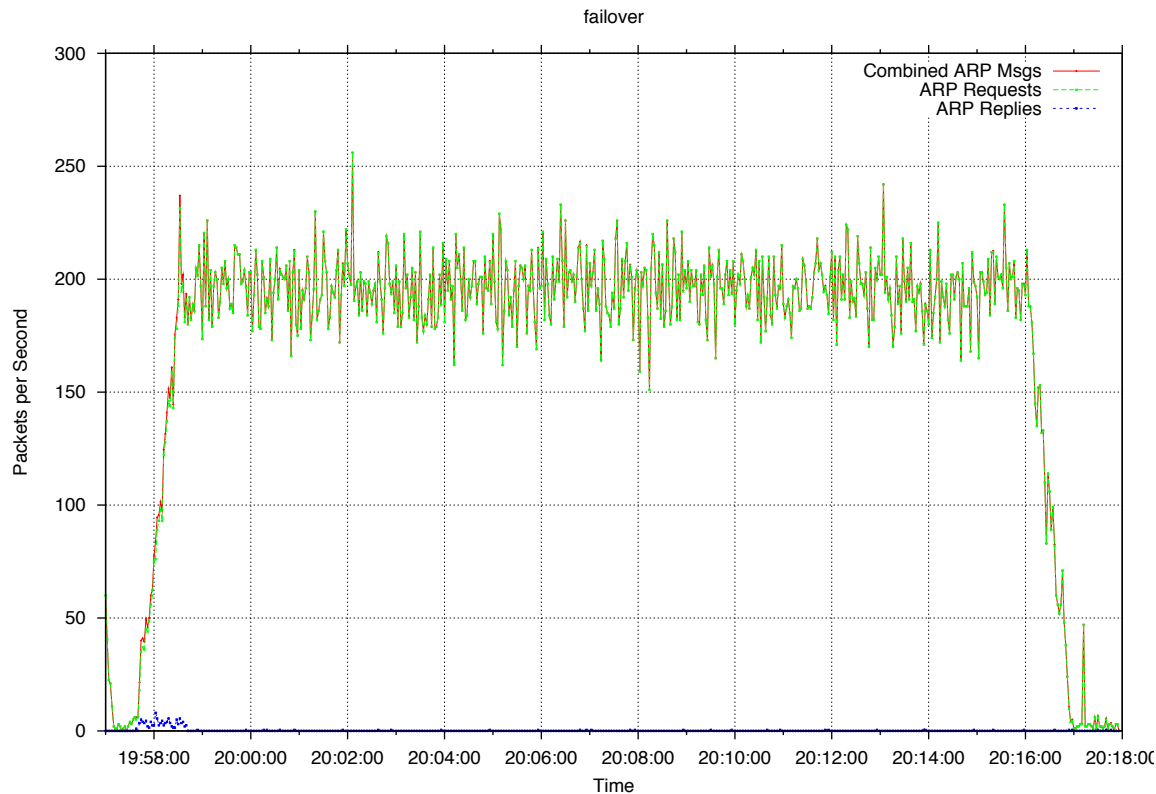
- Start Traffic generator on 500 hosts
- Roughly 36 VMs on 1 blade
- Place blade into maintenance mode forcing all VMs on that blade to migrate to other blades
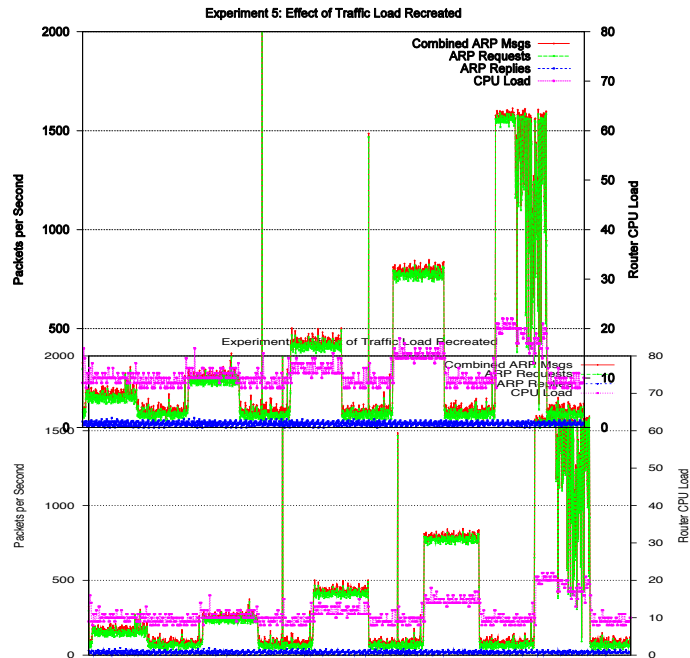- Measure ARP Traffic
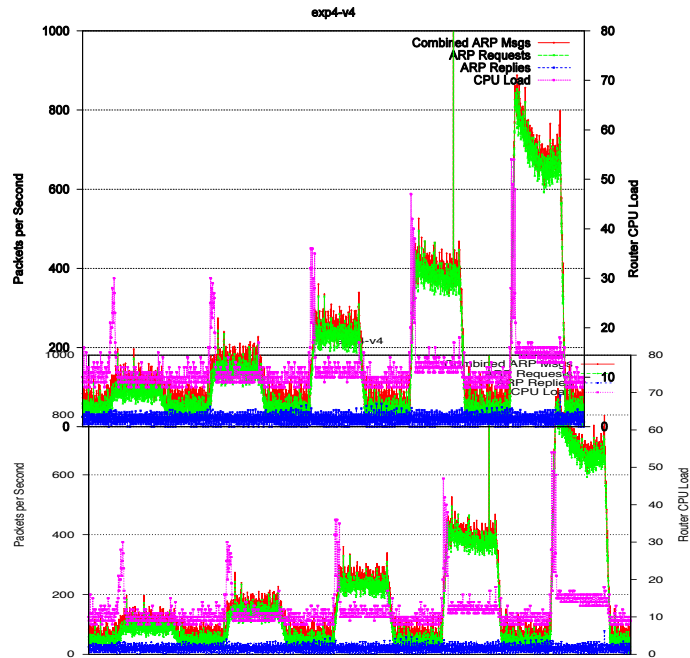
V: Highest possible Load
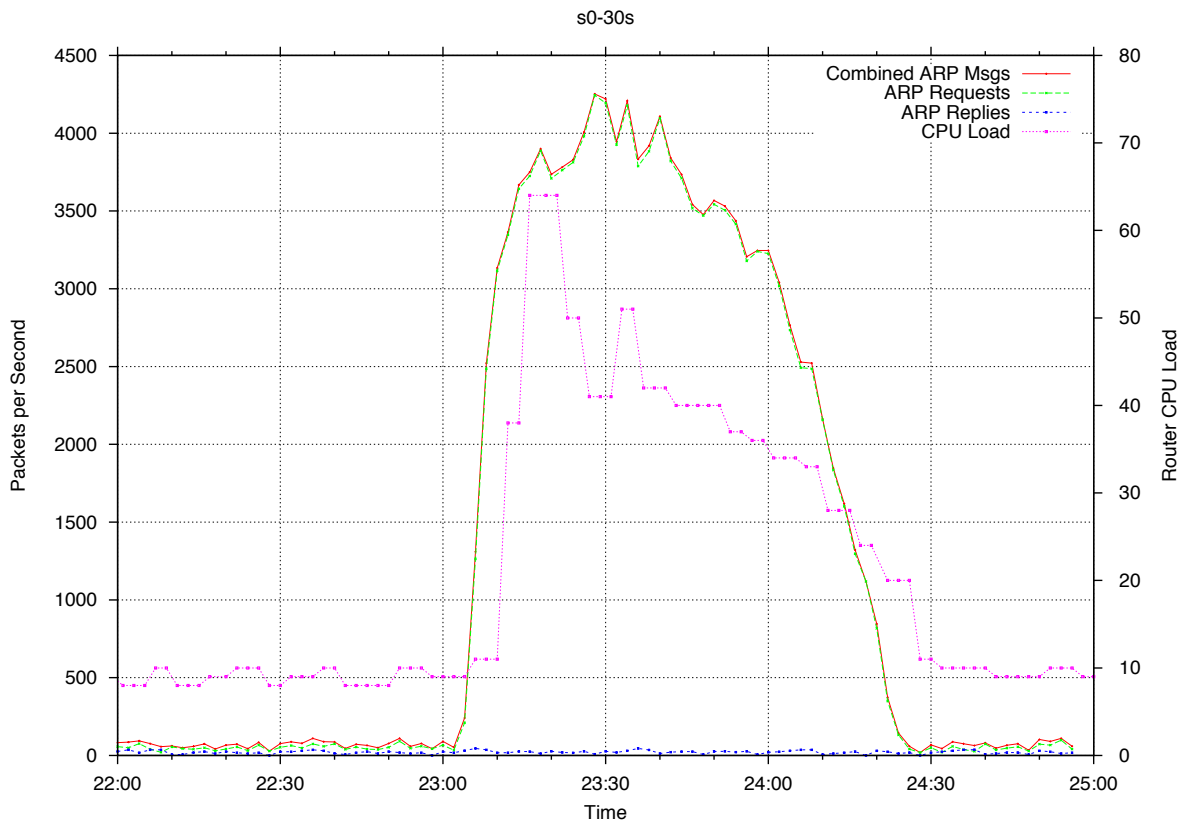
- Start traffic generator on 500 hosts
- All hosts communicate with all other hosts as fast as possible (sleep interval = 0)
- 30 second test
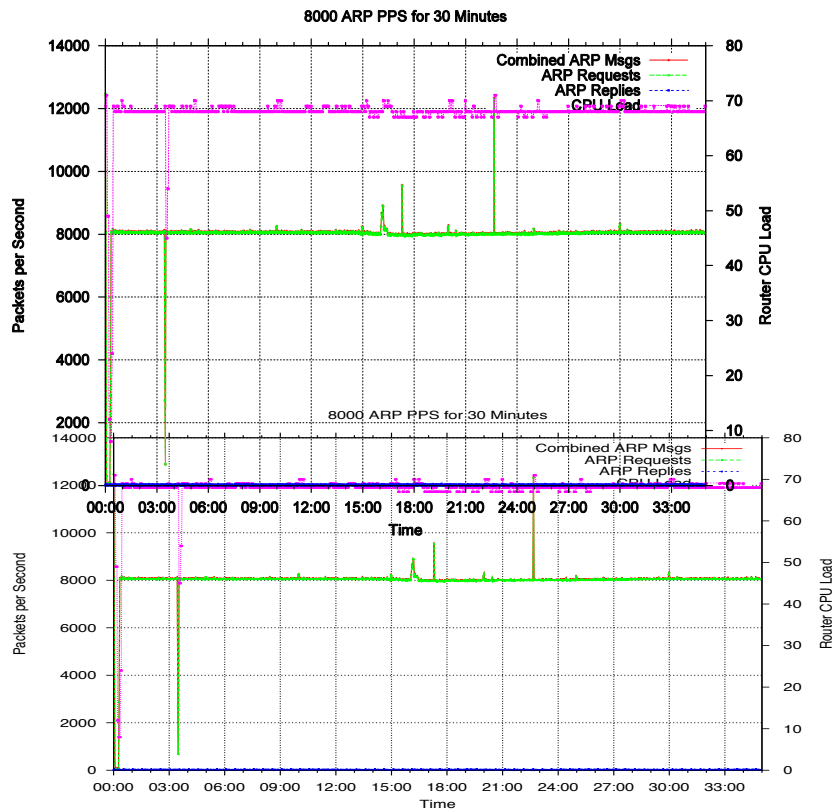- Maximum 4K pps of ARP pkts and a CPU load of 65

Emulator Topology
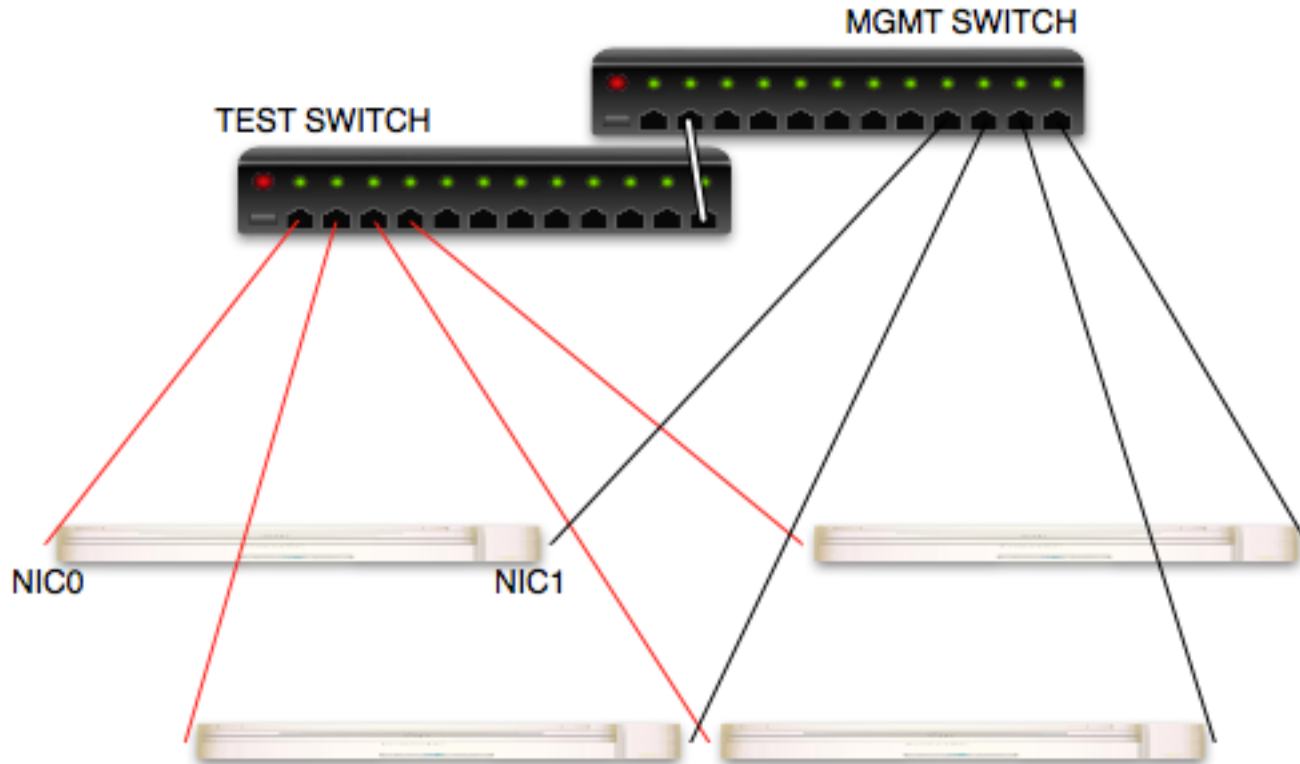
Emulator has two parts, a sender and a responder

Emulator Validation

Real (top) vs emulator (bottom)

High Emulator Load

- 8000 pps for 30 minutes

- Equivalent to several thousand real hosts depending on traffic profile

- Sustained 70% switch cpu load

Conclusion and Next Steps

- There appears to be roughly linear relationship between ARP traffic and number of hosts and traffic
- There can be scenarios where machine failures in a data center can lead to higher ARP traffic rates
- Overall ND appears to be lower impact than ARP in terms of traffic volume, number of hosts, and CPU load on access switch as compared with ARP
- Detailed analysis of collected ARP/ND traffic
- Build model of ARP/ND traffic behavior based on experimental data
- Build ARP/ND traffic emulator software to scale up the experiments
- Move away from limitations of managing real VM hosts
- Repeat measurements for live production environment and compare with our experimental setup