Deep Routing Simulation

Investigator: Alan Herbert

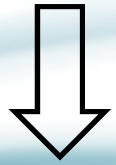
Supervisor: Barry Irwin

Structure

- What is it?
- What has been tried?
- What can we learn from past attempts?
- How I went about my implementation.
- Video Demonstration
- Results
- Conclusion
- Questions

Deep Routing Simulation, What's That?

- Routing Simulation: Having the ability to simulate the routing of packets.
- Deep: Refers to the depth at which it can route effectively.



(In this case, about 250 hops without issues)

OK ... But Why Simulate?

- Keep costs to a minimum.
- Pick up errors that are runtime specific.
- Maths and Stats aren't always your friend.

(trust me ... I know)

- Replacement of hardware implementations.
- There are more ...

There Are 3 Main Flavours

- Hardware Specific:

  All simulation done on specific hardware.
- Software Specific:

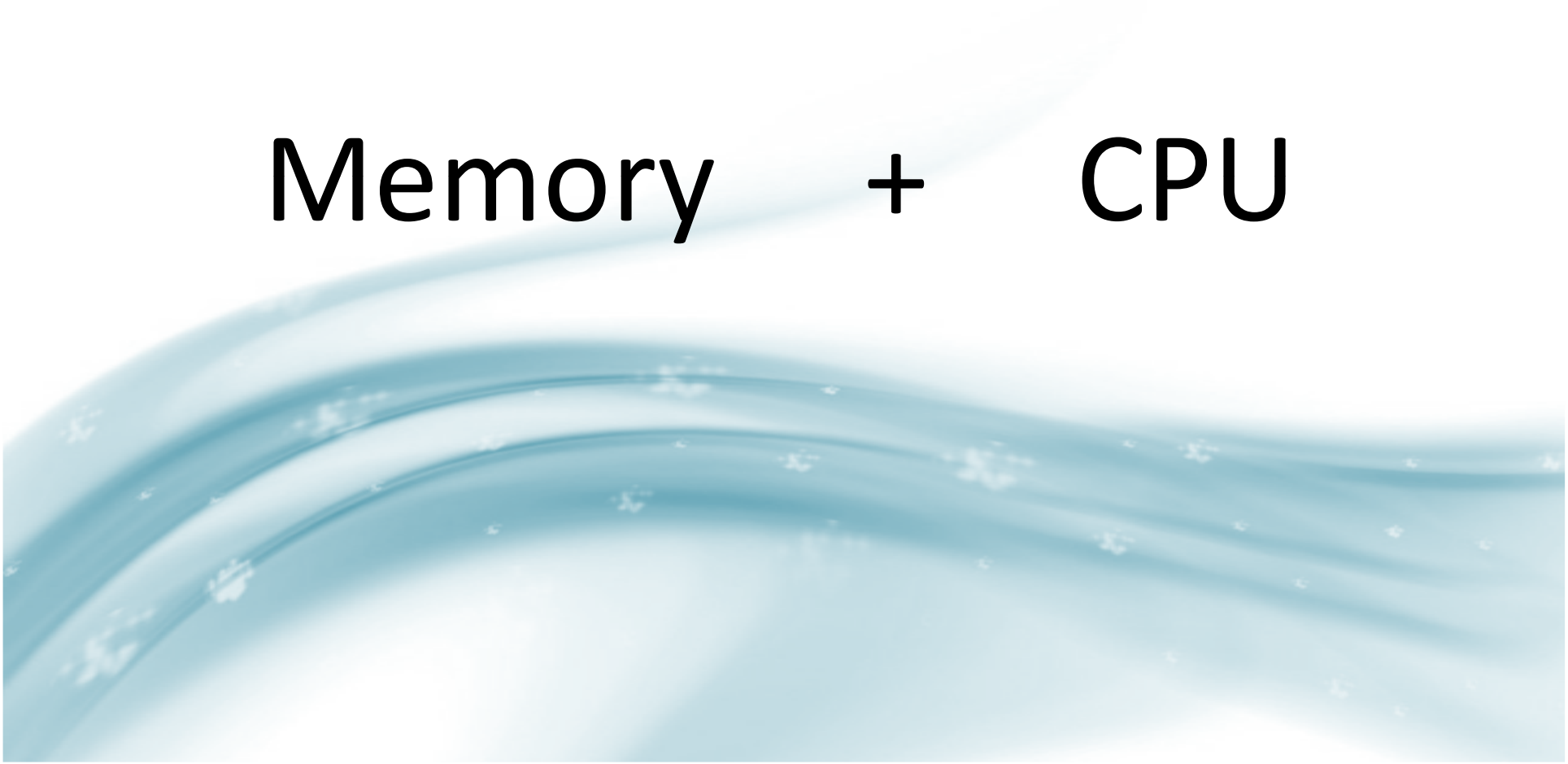
  All simulation done for a specific platform.
- Hybrids:

  Through interfacing to co-processors, GPUs and FPGAs.

Always 1 Problem ...

(In this case 2)

Memory + CPU



2 Problems ... 2 Times the Work!

So I took a look at two approaches:

- Static Routing Implementation:

  Calculate the routes before hand and store in memory.

- Dynamic Routing Implementation:

  Calculate the route at run time and use more CPU.

But ... but why?

There was a need to know whether a CPU or Memory intensive routing simulator is a better option and in what situations.

This was more of a subplot though ...

So What is Common Ground?

(Besides a coffee shop somewhere)

Both implementations make use of:

- Threads to handle packets.
- A console to make interfacing easy.
- Setup through configuration scripts.
- Network interfacing through libpcap and libnet.

Video Demo Time

Video 1: Build 10 000 routes and traceroute via a connected VM

1. Compile and Run.
2. Build the 10 000 routes from configuration script “cfg10k”.
3. Show routes are built by viewing a route in simulated network.
4. Run the simulator by activating packet sniffer.
5. Trace a route using VM (this gets no host response).
6. Bind host to the traced routes final node.
7. Retrace the route using VM (this gets a response and ends).
8. Stop and Exit.

Video Demo Time

Video 2: Build, bind 2 VM's, ping and packet loss

1. Run and recreate routes as in Video 1.
2. Check routes exist and then bind VM's.
3. Run simulator.
4. Make VM's ping each others bound node IP.
5. Toggle packet drop rates of a hop in route. (50% -> 100% -> 0%)
6. Stop and Exit.

Some Results

More specifically:

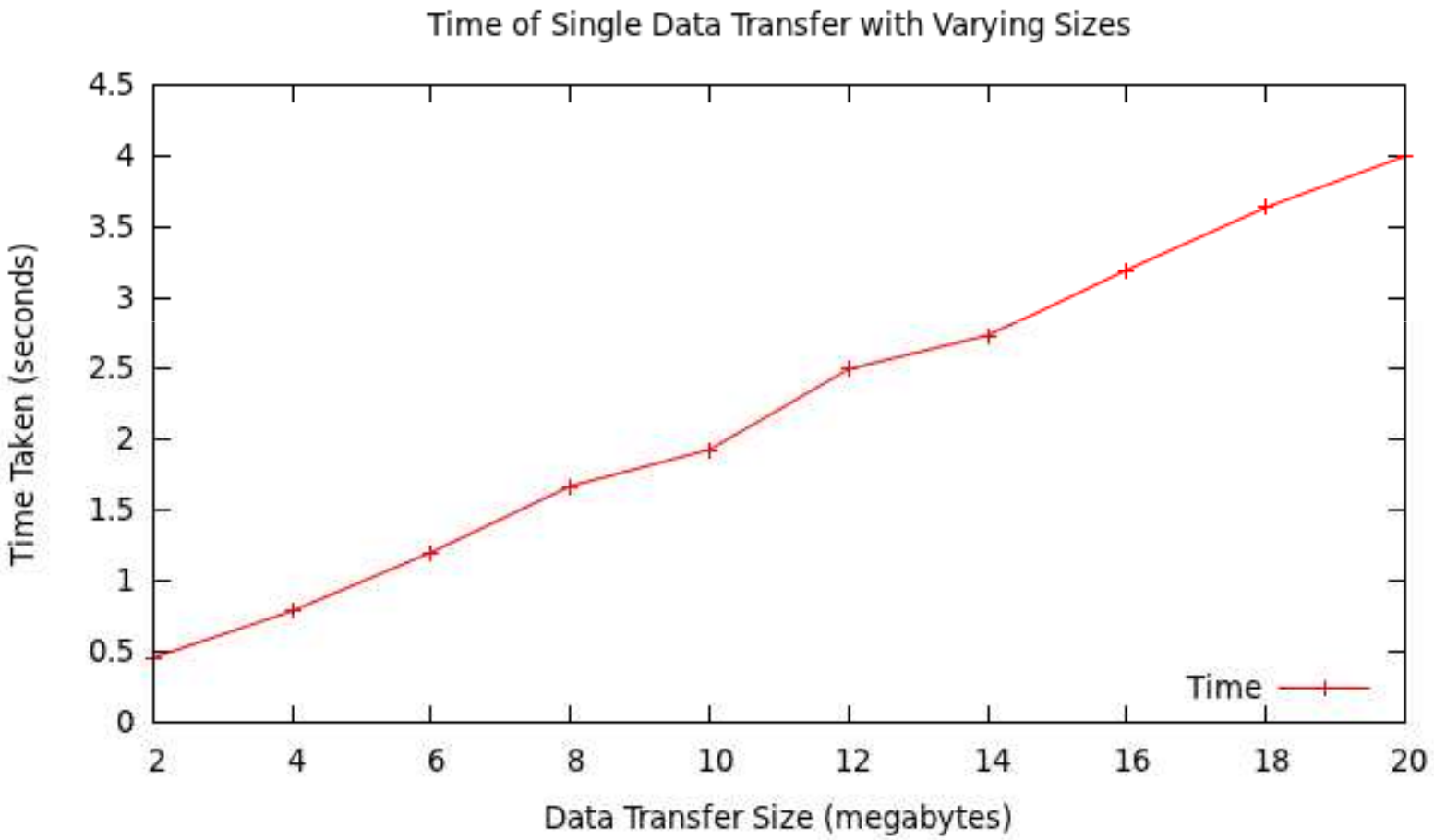
- Throughput
- Memory Usage
- CPU Usage

Note:

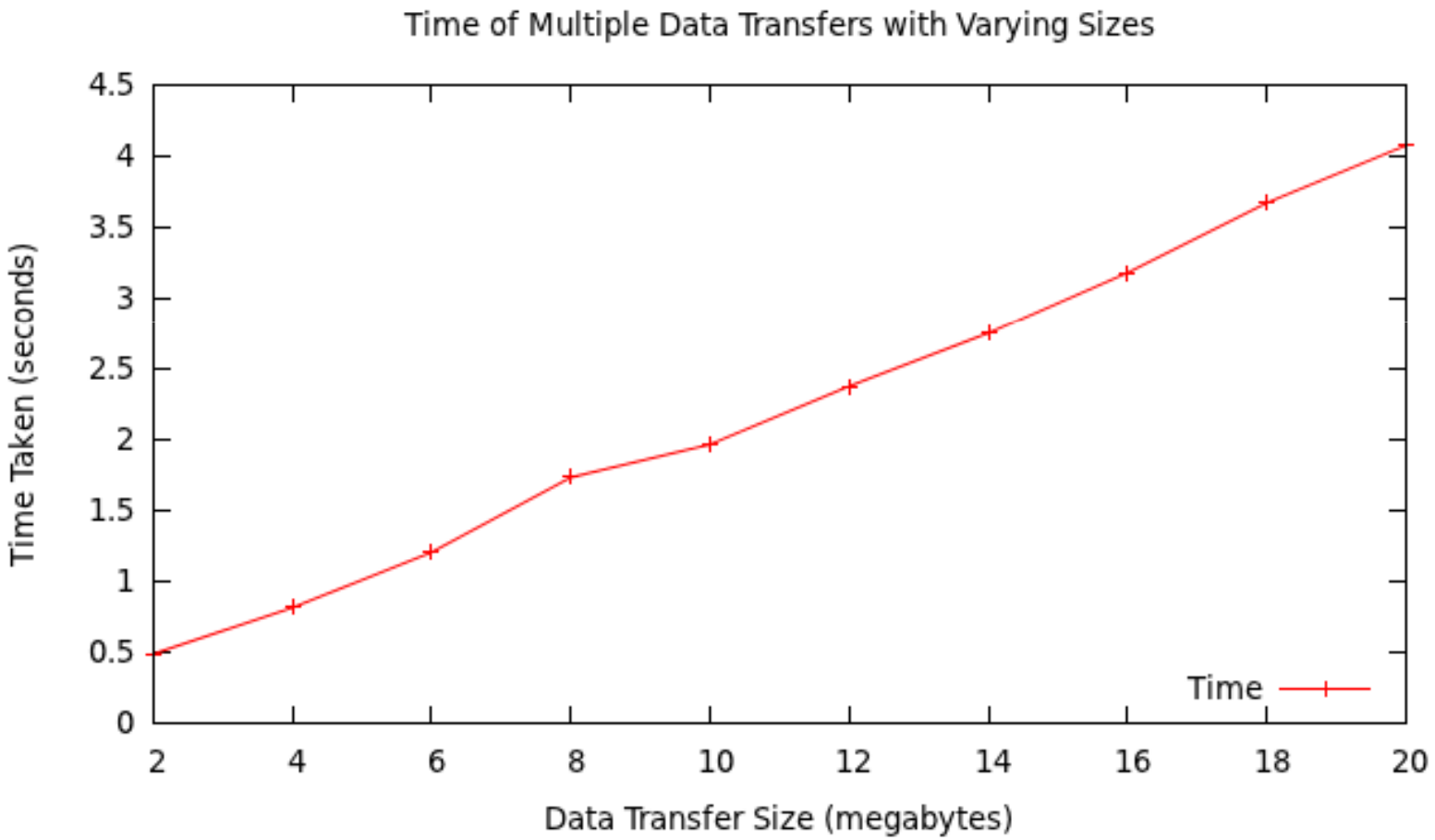
10000 Endpoints in Simulated Network

18739 Nodes were created

Throughput


Time of Single Data Transfer with Varying Sizes
Time Taken (seconds)
Data Transfer Size (megabytes)
Time
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
2
4
6
8
10
12
14
16
18
20

Throughput

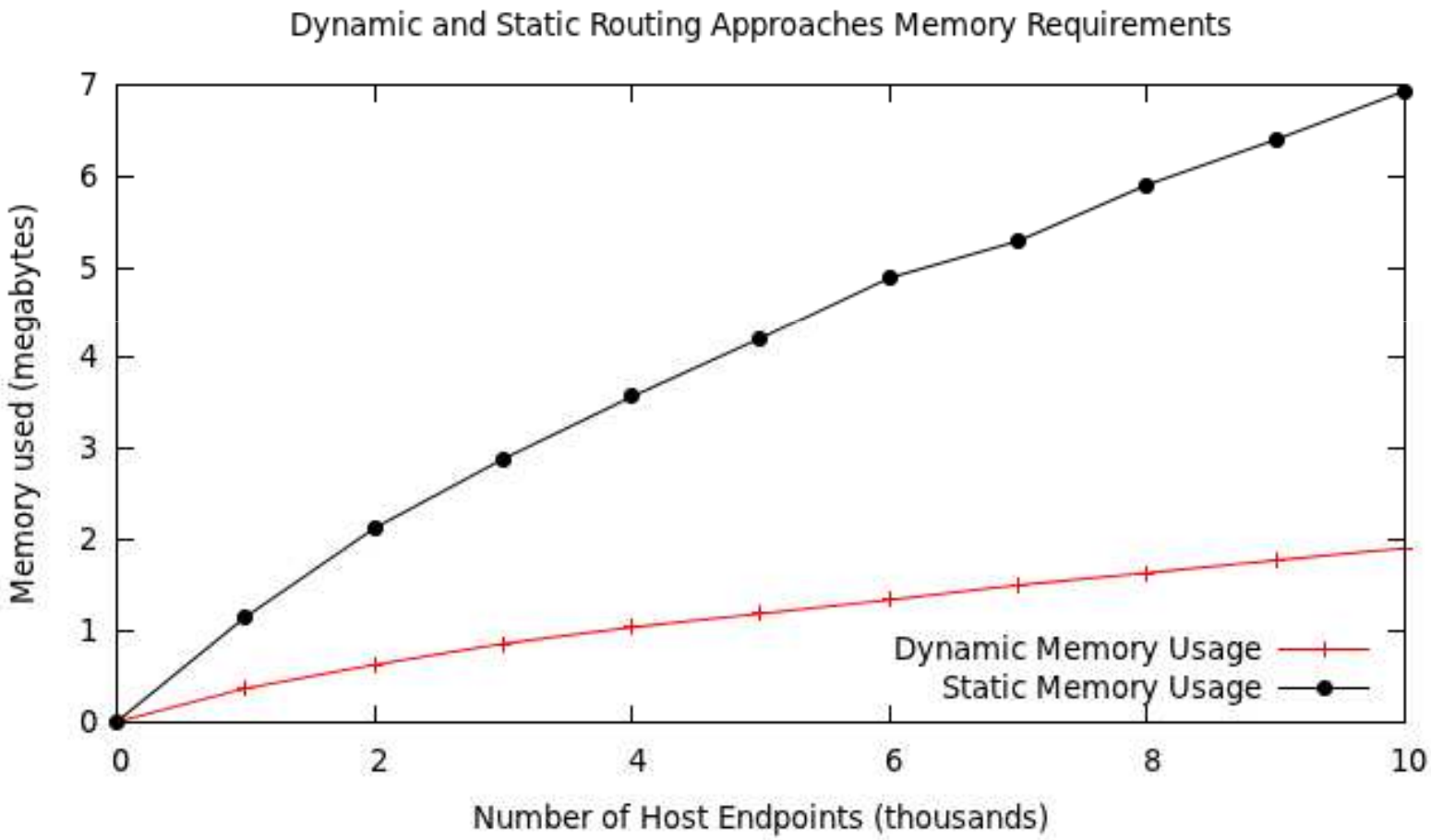

Time of Multiple Data Transfers with Varying Sizes
Time Taken (seconds)
Data Transfer Size (megabytes)
Time
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
2
4
6
8
10
12
14
16
18
20

Throughput

Done under Single Transfer and Parallel Transfer:

Single Transfer Average:	39.54Mbps
Parallel Transfer Average:	38.98Mbps
Combined Average:	39.26Mbps

Memory Usage

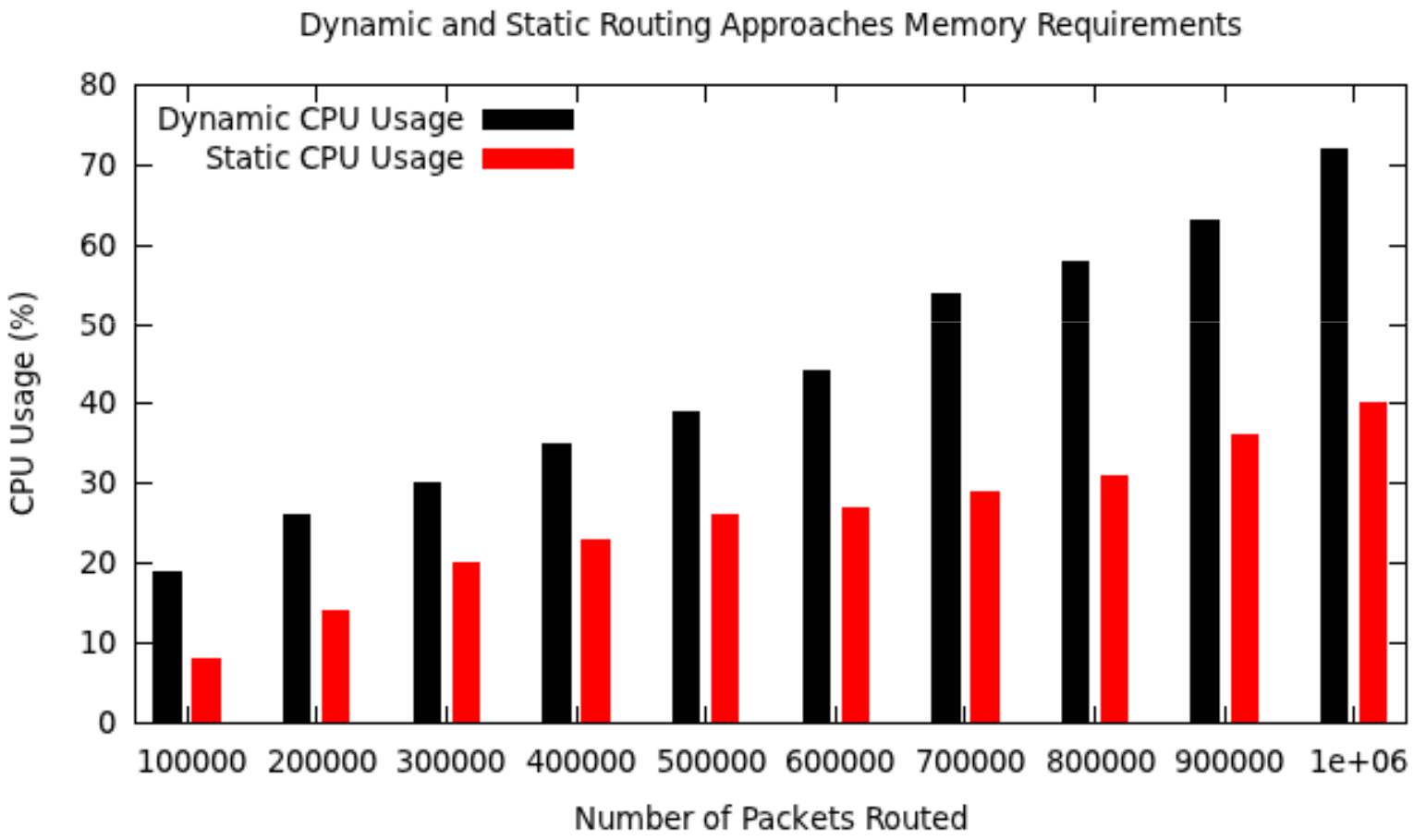


Memory Usage

This compares both Static and Dynamic approaches:

Static Routing Average:	±807 bytes/host
Dynamic Routing Average:	±225 bytes/host
(Static – Dynamic) =	±582 bytes/host

CPU Usage

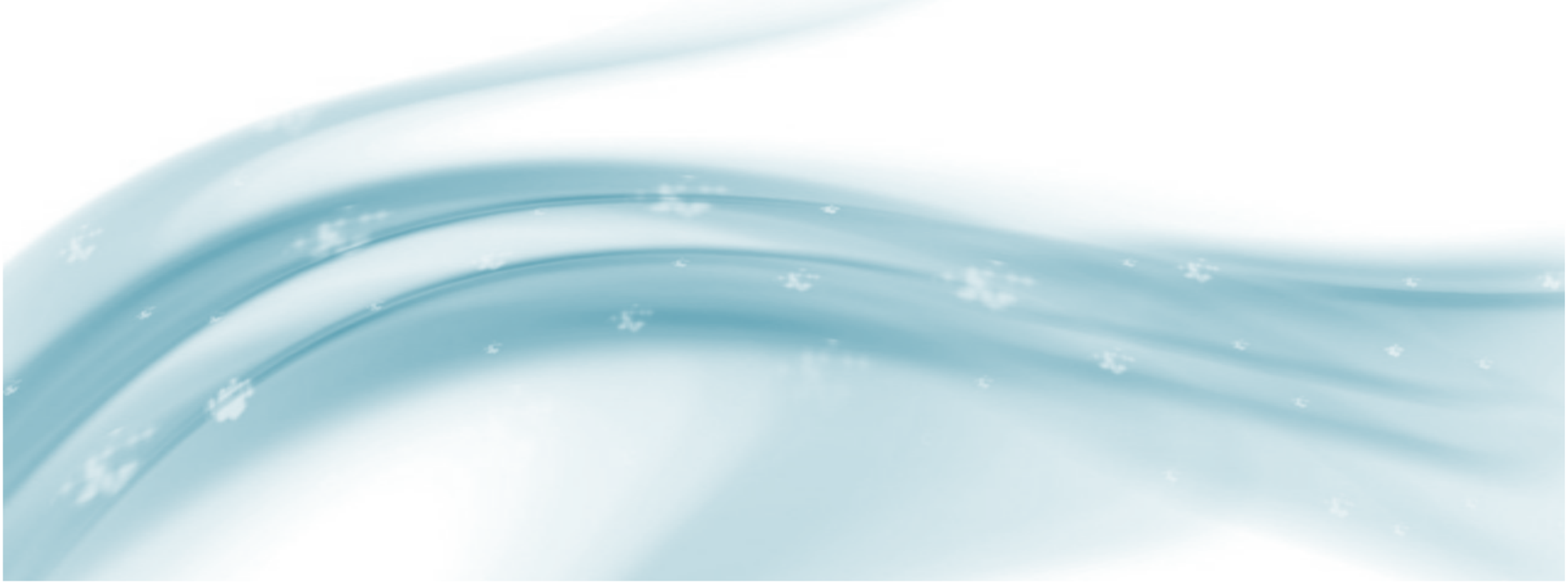

Dynamic and Static Routing Approaches Memory Requirements
Dynamic CPU Usage
Static CPU Usage
CPU Usage (%)
0
10
20
30
40
50
60
70
80
100000
200000
300000
400000
500000
600000
700000
800000
900000
1e+06
Number of Packets Routed

CPU Usage

Predicted 100% CPU Usage for both routing styles:

Static Routing Prediction: 2.2 million packets

Dynamic Routing Prediction: 1.2 million packets



So What Do These Figures Mean?

- Dynamic routing requires 72% less memory than the Static approach.
- Dynamic routing requires 45% more CPU resources than the Static approach.

So it seems that a dynamic routing style is in favour but this is still application specific as one has to take in to account processing delay.

Is It Actually Usable?

Throughput Comparison:

This Routing Simulator:	40 Mbps
Average LAN Bandwidth:	100 Mbps
Average Korean Bandwidth:	17 Mbps
Average World Bandwidth:	2 Mbps

In Short

Local Internet Simulation

Global Interconnect Simulation

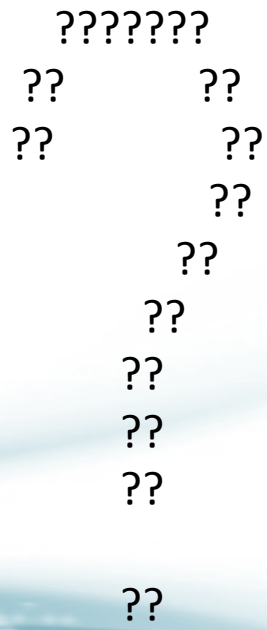
Office/Institute LAN Simulation

Something Pretty Awesome

Based on figures taken from http://www.isc.org/solutions/survey

- In July 2012 there were roughly 900 000 000 findable nodes in the Internet.
- Using the dynamic routing approach this would use just over 92Gb of memory.
- This would fit into 128Gb of memory, a number that now days is not so far fetched.

Questions



Thanks for the year ☺