Detecting Uncooperative Ethernet Elements using Accurate Round-trip Time Measurements

J.P. Delport*
Information and Computer Security Architectures (ICSA) Research Group,
University of Pretoria, South Africa
Email: jpdelport@csir.co.za
Telephone: (012) 841-3689
Fax: (012) 841-3453

Martin S. Olivier
Information and Computer Security Architectures (ICSA) Research Group,
University of Pretoria, South Africa
Email: martin@mo.co.za
Telephone: (012) 420-2052
Fax: (012) 362-5188

***Abstract*— Knowledge of a network's entities and the physical connections between them (a network's physical topology) can be useful in a variety of network scenarios and applications. Specifically, topology information can be used by administrators to detect unauthorised physical modifications to a network. Gathering accurate topology information manually can be a tedious and error-prone, if not impossible task.**

In this paper an active probing technique is used to obtain packet timing information from a set of experimental Ethernet LANs. Packets are sent from a probing host to a target node in a variety of network configurations and the total time from sending out the packet to receiving a reply is measured and stored. The stored timing data is then used to determine the influence of common Ethernet network elements on packet round-trip times. Real-Time Linux is used to obtain low-level and fine grained timing control over the probing host's network card.

I. Introduction

A network's nodes and the physical interconnections between them define the network's physical topology. Topology information can be valuable in a variety of situations, it can be used for network administration and planning [1, 2], protocol and routing algorithm development, performance prediction [3] and as a basis for accurate network simulation.

Network security aspects including threat detection, protection and reaction can also benefit from accurate network topology information. For instance, a snapshot of a network's normal topology would allow the detection of unauthorised topology modifications. Topology information can also help in making decisions about firewall or intrusion detection system placement.

Topology information can be obtained at either the network (Open System Interconnect layer 3) or data-link (OSI layer 2) layer and information from both levels can be useful in certain scenarios. Network layer information might be useful for routing optimisation, whereas data-link layer topology information might be useful for server siting [2].

Manual topology discovery is becoming increasingly difficult (if not impossible) because of the size and dynamic behaviour of networks. Automatic topology discovery tools and algorithms will therefore play an important part in network management and security.

As a subproblem of physical topology discovery, the focus of this paper is on detecting the influence and presence of layer 2 Ethernet elements using packet timing information. To this end, packets are sent from a probing host to a single target node in a variety of network configurations and accurate round-trip times are recorded and processed.

The rest of the paper is organised as follows: Section II provides some background information and discusses related work. Section III describes how a probing host was configured for experiments and Section IV discusses the actual experiments and results. Finally, Section V concludes the paper and discusses potential future work.

II. Background and Related Work

Since its conception in 1973 by Bob Metcalfe at the Xerox Palo Alto Research Center [4], Ethernet has evolved (and continues to do so) into a ubiquitous data communication standard. Ethernet LANs currently dominate in corporate environments and Ethernet ports account for a very high percentage of Internet endpoints.

Naturally, the problem of discovering the physical topology of Ethernet networks has received research attention; however, most research efforts assume that layer 2 elements are cooperative and intelligent. The most common technique for physical topology discovery involves collecting forwarding tables from Ethernet elements using Simple Network Management Protocol (SNMP) Management Information Bases (MIBs) [3, 2, 5]. The forwarding table data is then processed using algorithms to generate a physical network topology map. Proprietary techniques for topology discovery are also used by network equipment vendors, but can only be utilised in homogeneous environments if custom extensions to the SNMP are involved [3]. As Ethernet networks grow they are segmented further and further using more and more layer 2 elements. Alternative techniques for detecting these potentially *dumb*, uncooperative or heterogeneous elements could therefore prove valuable.

* J.P. Delport is also an employee of CSIR Defencetek, Meiring Naude Road, Pretoria, South Africa.

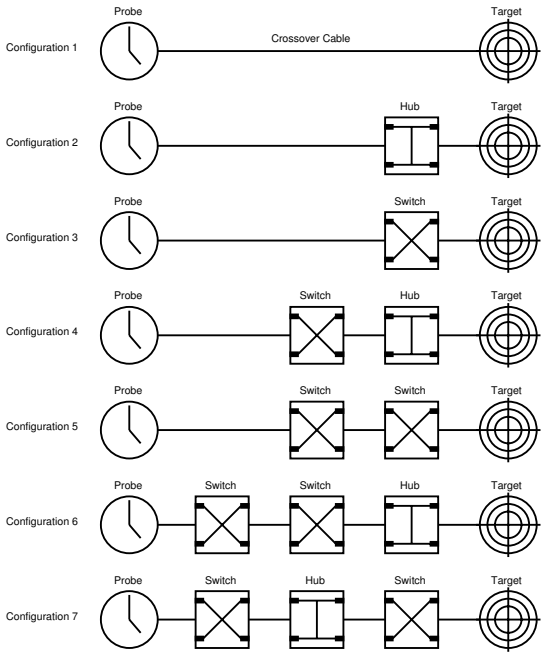

Fig. 1. Experimental Network Configurations

Layer 2 network elements in an Ethernet are essentially transparent at higher network layers; for instance, two hosts on an Transmission Control Protocol/Internet Protocol (TCP/IP) subnet can communicate directly without knowledge of the bridges, switches or hubs forming the physical link between them. Layer 2 elements do however influence the temporal behaviour of packets in the network.

Ethernet hubs electrically mirror incoming traffic to all ports, effectively creating a shared medium between nodes connected to it. The shared medium can only be used for half-duplex communication, which makes detection of hubs using packet timing information possible. Ethernet bridges and switches eliminate the shared medium and create full-duplex links between nodes connected to them by storing and forwarding packets. Packets can however only be forwarded once completely received by the bridge or switch, thereby creating delays in packet delivery which can be exploited to detect these elements.

III. Methodology

In order to detect the influence of layer 2 network elements on packet delivery times, packets must be handled by the elements and the time of packet transmission and reception must be recorded. Experimental Ethernet LAN configurations as shown in Figure 1 were constructed for this purpose.

Two methods for measuring the delay experienced by a packet as it travels across the network were considered. Packets can either be sent directly from a probing host to a target node or packets that evoke a response from a target node can be sent from a probing host. The first method requires the cooperation of the target node as well as accurate clock synchronisation between the probing host and the target node in order to calculate the packet delay time. The second method has to contend with the uncertainty in the target node response time, but was chosen for its simplicity.

Suitable packets, usable by the second method to record round-trip times, now had to be found. Address Resolution Protocol (ARP [6]) packets were chosen for this purpose for the following reasons:

- Every host on an Ethernet LAN wishing to communicate using TCP/IP is required to listen to ARP broadcasts requesting its IP address and to reply with its hardware Medium Access Control (MAC) address [7]. Targets for probing are therefore abundant.
- ARP request packets are easy to generate and replies are easy to parse, thereby simplifying the host application software.
- ARP packets are normally handled low down in a node's network stack, thereby increasing the predictability of reply times from the probed target.
- Valid ARP requests can be encapsulated in Ethernet frames of various sizes, allowing for experiments with different packet sizes.

The probing host has to accurately record the time that an ARP request was transmitted and the time that an ARP reply was received. An ideal store-and-forward element on a 100Mbit Ethernet LAN would introduce a unidirectional packet delay that is equal to the transmission time of the packet. On a 100Mbit Ethernet LAN a 1000-byte packet takes roughly 80 microseconds to transmit. Detecting this sub-millisecond delay necessitated the use of a probing host with access to a high resolution clock and fine grained control over its network card. RTLinux/GPL, a hard real-time kernel that runs Linux as its idle thread [8], combined with device drivers from the RTLinux Ethernet Device Drivers (REDD) project [9] were used to satisfy this need.

The Ethernet device driver for the Realtek 8139 chipset (found on quite a number of inexpensive 100Mbit Ethernet network cards) was modified to allow the interrupt handler to accurately record the time after a complete packet was transmitted or received. The resolution of the RTLinux clock used for the time-stamping was around 1 microsecond. After a packet or packet group was successfully transmitted, the driver waited 10 milliseconds for replies to arrive. A user-space application was created that communicated with the real-time driver in order to send commands and to receive the timing information for packets where a reply was received. The commands to the real-time driver contained the following:

- The target MAC address.
- The target IP address.
- The number of ARP packets to send.
- A list of packet lengths (Ethernet frame sizes).
- A list of delay times to be used between the sending of adjacent packets.

The round-trip times were saved into comma separated text files and Matlab® was used to read the files, process the data and create plots.

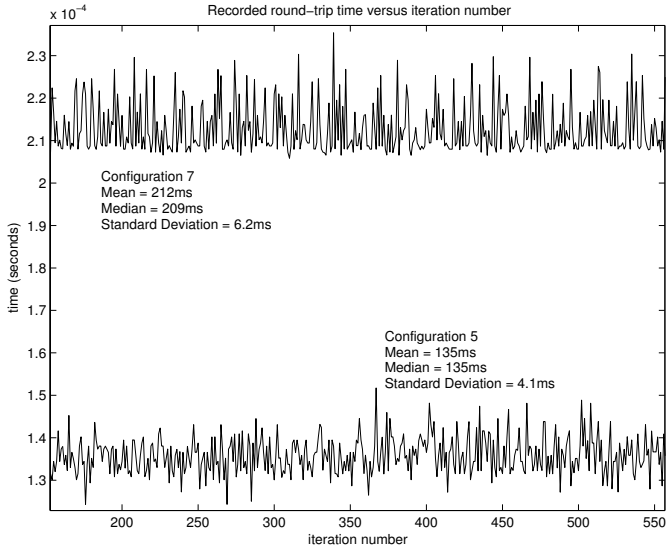

Fig. 2. Typical Measured Round-trip Times for 960 byte Single Packets for Two Network Configurations

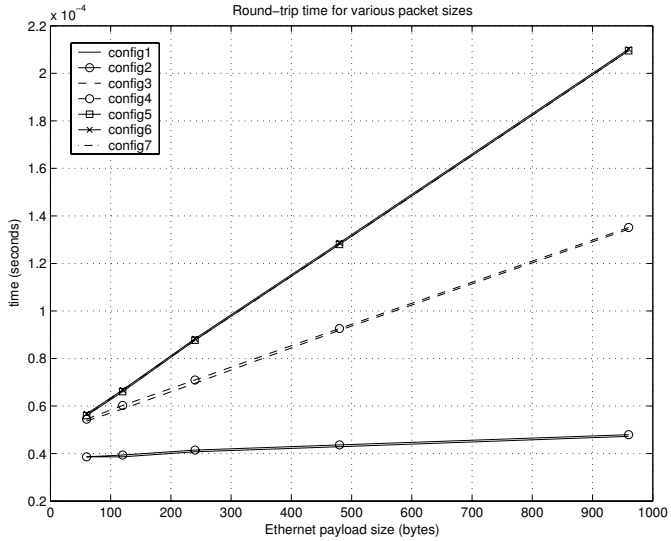

Fig. 3. Round-trip Times for Single Packets with Various Sizes for each of the Network Configurations

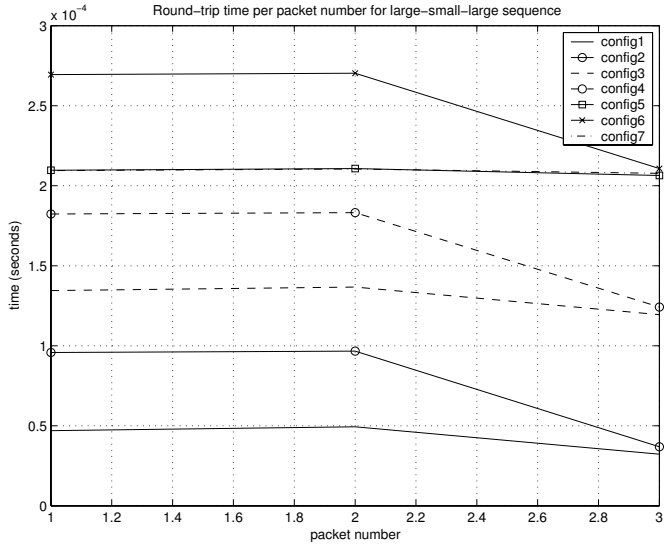

Fig. 4. Round-trip Times for each of the Large, Small, Large Packets for each of the Network Configurations

IV. Experiments and Results

Experiments were conducted using the setup for the probing host as described in Section III. A single target node (a laptop running Linux) was connected to the probing host using several experimental network configurations consisting of 100Mbit hub and switch elements as shown in Figure 1. The experiments had the aim of detecting each of the network elements by measuring the influence it had on the round-trip times of packets sent to the target node.

Initially a set of four experiments were conducted for every network configuration. The ARP request packets (or sequence of packets) sent during each of the experiments consisted of the following:

- Single packets with sizes of 60, 120, 240, 480 and 960 bytes 10 milliseconds apart.
- A pair of packets (both 960 bytes in length) as close together in time as possible.
- A triplet of packets (60, 960 and 60 bytes in length) as close together in time as possible.
- A triplet of packets (960, 60 and 960 bytes in length) as close together in time as possible.

It was hoped that the combinations of small and large packets would reveal the half-duplex or full-duplex status of the target node. It should also be noted that the size of the ARP reply packet generated by the target node remains constant (60 bytes).

The sets of packets were transmitted a 1000 times for every configuration and the median of the round-trip times for every packet was calculated and used in further plots. Figure 2 shows the typical distribution of measured round-trip time values for a group of iterations for the single 960 byte ARP request packets for network configurations 5 and 7. It is expected that the noise in the round-trip time measurements would increase in an uncontrolled environment because of network and target node activity. Figure 3 and Figure 4 show the results for the first and last experiment respectively.

From Figure 3 it can be seen that the round-trip time is mostly influenced by the amount of store-and-forward elements (switches in this case) forming the link between the probing host and the target node and that the hub is essentially invisible to this experiment. It can also be seen that the round-trip times increase linearly as the ARP request packet size is increased.

Figure 4 shows the round-trip times for each of the triplet of packets in the large-small-large packet experiment. The influence of the hub on the round-trip times for the first and second packet can now be seen for network configurations 2, 4 and 6. The extra delay in the round-trip time of the first and

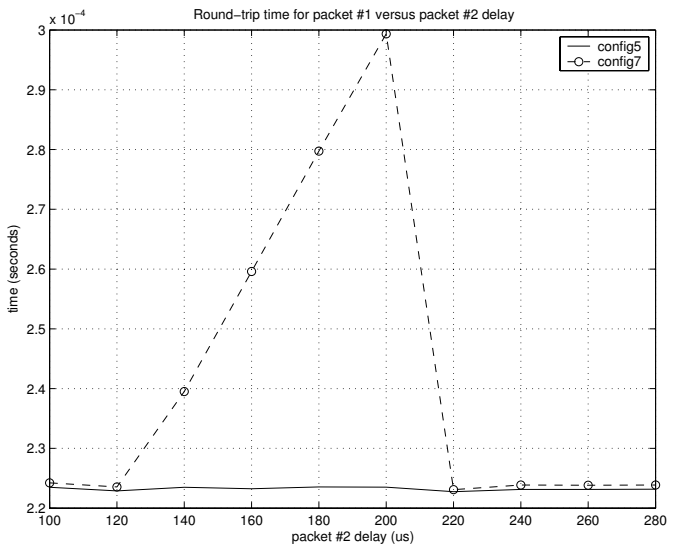

Fig. 5. Round-trip Times for the First of Two Large Packets with Various Delays for the Second Packet for Network Configurations 5 and 7

second packet is caused by the half-duplex nature of the final connection to the target node. The Ethernet carrier-sense [10] circuitry in the target node forces it to wait until the third packet has been completely received before it can reply to the first and second packets.

The results of the large-small-large packet experiment showed that the presence of the hub could be detected by sending multiple packets, provided the packets attempt to utilise the half-duplex link at the same time instant. A final experiment was conducted in an attempt to differentiate between network configuration 5 and 7. Two large packets (both 960 bytes in length) were again sent to the target node, but the transmission of the second packet was delayed by a specific amount of time.

Figure 5 shows the round-trip time for the first packet of the pair of packets for network configurations 5 and 7. It can be seen that for delays between 120 and 220 microseconds, the round-trip time for the first packet increases for network configuration 7. During this time interval, the switch closest to the target node has to wait for the delayed second packet to be completely transmitted through the hub, before the reply to the first packet can be delivered. It can also be noted that the maximum delay of around 75 microseconds correspond to the transmission time of the 960 byte second packet.

V. Conclusion and Future Work

The results of the experiments show that it is possible to detect even uncooperative Ethernet elements like hubs and unmanaged switches using accurate round-trip time measurements. The current experimental configuration could possibly be used as is to detect physical topology changes to an Ethernet LAN, provided a snapshot of the round-trip times for the LAN is stored in advance.

Possible future work include experiments on live networks with a variety of targets, link bandwidth determination using multiple packets and the possible inference of physical topology using data collected from a variety of network vantage points.

References

[1] H.-C. Lin, H.-L. Lai, and S.-C. Lai, "Automatic link layer topology discovery of IP networks," in *Communications, 1999 IEEE International Conference on*.

[2] Y. Breitbart, M. Garofalakis, B. Jai, C. Martin, R. Rastogi, and A. Silberschatz, "Topology discovery in heterogeneous IP networks: the NetInventory system," *Networking, IEEE/ACM Transactions on*, June 2004.

[3] B. Lowekamp, D. O'Hallaron, and T. Gross, "Topology discovery for large Ethernet networks," in *ACM SIGCOMM Computer Communication Review, Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*.

[4] B. Metcalfe, "What is this thing they call Ethernet?" December 2002, Electronic Design Technology Report.

[5] D. T. Stott, "Layer-2 path discovery using spanning tree MIBs," Avaya Laboratories, Tech. Rep., March 2002.

[6] D. C. Plummer, "RFC826 - An Ethernet Address Resolution Protocol," Internet Request For Comments, November 1982.

[7] S. Keshav, *An Engineering Approach to Computer Networking*. Addison-Wesley, 2001, ch. 10, p. 283.

[8] "RTLinux/GPL web site," http://www.rtlinux-gpl.org/.

[9] "RTLinux/GPL Ethernet Device Drivers project page," http://redd.sourceforge.net/.

[10] S. Keshav, *An Engineering Approach to Computer Networking*. Addison-Wesley, 2001, ch. 7, p. 139.

Biographies

J.P. Delport was born on 11 May 1975 in Pretoria. Having completed electronic engineering and computer science degrees at the University of Potchefstroom, he joined CSIR Defencetek in 1999. He is currently completing a part-time masters degree in computer science at the Information and Computer Security Architectures (ICSA) research group of the University of Pretoria.