Towards Improving Packet Probing Techniques

Matthew J. Luckie, Anthony J. McGregor, Hans-Werner Braun.

Abstract— **Packet probing is an important Internet measurement technique, supporting the investigation of packet delay, path, and loss. Current packet probing techniques use Internet Protocols such as the Internet Control Message Protocol (ICMP), the User Datagram Protocol (UDP), and the Transmission Control Protocol (TCP). These protocols were not originally designed for measurement purposes. Current packet probing techniques have several limitations that can be avoided. The IP Measurement Protocol (IPMP) is presented as a protocol that addresses several of the limitations discussed.**

Keywords— **Packet Probing Techniques, Packet Delay, Network Path.**

I. Introduction

As the Internet grows in scale and complexity, the need for measurement increases. The underlying need for measurement is to understand why the Internet behaves the way it does in complex conditions. One form of measurement is active measurement; this involves introducing packets into the network and measuring the way the network handles those packets. This paper focuses on packet delay measurements.

Measurement packets can be encapsulated in existing protocols such as the Internet Control Message Protocol (ICMP) [1], the User Datagram Protocol (UDP) [2], and the Transmission Control Protocol (TCP) [3]. Examples of packet probing techniques that are encapsulated in these existing protocols are `ping`, `traceroute`, and the IP Performance Metrics (IPPM) group's One-way Delay Protocol (OWDP) [4]. The protocols used to encapsulate these measurement packets were not designed with measurement as a consideration. There may be serious limitations to measurements encapsulated in these protocols.

Current packet probing techniques are not suited to measuring packet delay at the router level. This makes the task of identifying where delay occurs in the network more difficult. In addition, current approaches to ensuring clock synchronisation where delay measurements include timestamps from more than a single clock have a high cost, normally requiring a dedicated external time receiver be installed on each host or router involved in a measurement. Despite the implicit requirement for information regarding the synchronisation of a clock when multiple independent clocks are represented in a measurement, the authors are not aware of any protocol that provides a mechanism to retrieve this information.

The IP Measurement Protocol (IPMP) [5] is introduced as an example of a protocol that addresses some of the limitations of using existing protocols to encapsulate packet probes. IPMP considers both the packet delay and the path a packet takes in a single packet exchange between the measurement host and the echo host. In the opinions of the authors, the protocol is tightly constrained, efficient, and easy to implement. It is hoped that these characteristics will make IPMP suitable for implementation by router manufacturers. This would enable packet delay measurements to indicate places of delay due to congestion between two hosts as a single packet passes through the network.

This paper is organised as follows. In section II, a discussion of the current measurement techniques is presented. Limitations of these techniques are presented in section III. Section IV presents the IPMP protocol that introduces a new technique that allows path and delay measurements to be combined in a single packet exchange. Section V presents some remaining design issues with the IPMP protocol in its present specification.

II. Discussion of Current Techniques

A. Ping

The most widely used method to investigate network delay is for a measurement host to construct and transmit an ICMP echo request packet to an echo host as outlined in RFC 792 [1]. Round trip time (RTT) is calculated as the difference between the time the echo request is sent and the time a matching response is received by the `ping` application.

A variation of this method is to construct an ICMP

The authors are with the National Laboratory for Applied Network Research (NLANR), Measurement and Network Analysis Group, San Diego Supercomputer Center, University of California, San Diego, La Jolla, USA and the University of Waikato, Hamilton, New Zealand. Email: mjl,tonym,hwb@nlanr.net

This work is funded, in part, by NSF Cooperative Agreement No. ANI-9807479. The U.S. government has certain rights to this material.

timestamp request packet, also outlined in RFC 792 [1]. This packet contains three timestamps - the originate timestamp, the receive timestamp, and the transmit timestamp. If both hosts involved in the timestamp exchange have synchronised clocks, the forward path delay can be calculated using the originate and receive timestamps. The reverse path delay can be calculated using the transmit timestamp contained in the timestamp response packet and the time the response packet arrived back at the transmitter.

B. Traceroute

The `traceroute` technique allows a measurement host to deduce the forward path to an echo host by systematically sending a sequence of IP [6] packets with an increasing time-to-live (TTL) value that is initially set to one. The forward path is deduced by extracting the source IP address from the ICMP TTL expired messages that are sent as successive routers discard the IP packets.

A variation of `traceroute` is `mtrace` [7], for investigating the multicast path from the transmitter to a receiver. `mtrace` uses features that are built into multicast routers, and thus `mtrace` does not use ICMP TTL expired messages as its method for tracing a path. `mtrace` measures the reverse path from multicast transmitter to receiver, and requires only a single query packet to be sent to the transmitter for the trace to be conducted for the entire reverse path. The trace is conducted in parallel, with each router sending a trace response to the request in addition to forwarding the request to the next multicast router on the reverse path to the receiver.

C. One-Way Delay Protocol

The IP Performance Metrics (IPPM) group has published several Request for Comment (RFC) documents that define frameworks for measuring the performance of IP networks [8], [9]. The IPPM group is well advanced in the engineering of a One-way Delay Measurement Protocol (OWDP) [4] that will build on a framework designed in RFC 2679 [9].

The OWDP specification provides a mechanism for measuring packet delay with UDP packet probes. In addition, the specification describes a mechanism for controlling a measurement session between two hosts with a TCP connection, for negotiating the UDP port numbers involved in the delay measurement, and for encrypting the data carried in the measurement packets to protect against manipulation by a third party.

III. Limitations of Current Techniques

A. Separation of Path and Delay Measurements

There are often large variations in delay between successive packets following the same route, particularly when a load balancing arrangement is in place or when a network is under high load. The path that successive packets take to the same destination may change during measurement, resulting in tools like `traceroute` possibly reporting a path that does not exist. This makes the task of correlating a `traceroute` measurement to a `ping` measurement difficult. A major problem is that it is more problematic to measure a network under heavy load, precisely when measurement is most valuable. A measurement technique that combines path and delay measurement would allow a measurement to ascertain not only network delay, but where delay occurs.

B. Limited Ability to Measure to a Router

Routers often make bad measurement targets because they are optimised for the relatively simple task of forwarding packets. Routers may process tasks that are resource intensive and therefore an opportunity for a denial of service attack at low priority or not at all. This has implications for path and delay measurements taken with techniques such as `traceroute`.

Some network administrators express concern that if the amount of active measurement activity on their network increases, significant network resources may be consumed handling this activity. In many cases the total traffic added to a network is often a very small fraction of the network's capacity. It is important that packet probes disturb the network as little as possible. In general, this means adding the minimum necessary number of packets to the network. Path measurement using `traceroute` requires many packet probes per path.

Some measurement techniques construct measurement traffic that can be difficult to efficiently detect amongst other network traffic. This type of measurement traffic precludes measuring to a router.

C. Limited Consideration of Protocol Encapsulation

Packet probes encapsulated in general purpose protocols such as ICMP, TCP, and UDP, may be subject to protocol filtering. This may result in delay measurements that not only consider network delay, but protocol filters that may not be appropriate for traffic encapsulated in other protocols. The reason for protocol filtering and rate limiting is often to prevent denial of service attacks.

The ICMP protocol is filtered at many routers, and may be blocked entirely despite RFC 1812 [10] requiring

ICMP. Section 4.3.2.8 of RFC 1812 allows for the router to rate limit ICMP replies to avoid the consumption of bandwidth and the use of router resources. The implication of this is that ICMP is not a reliable protocol for conducting delay and loss measurements.

The encapsulation of packet probes in UDP may be problematic due to the general-purpose nature of the protocol. UDP does not contain the well developed congestion management algorithms inherent in TCP and it is possible that UDP packets will be rate limited during periods of peak UDP usage in order to reduce their impact on TCP flows.

TCP has limitations for delay and loss measurement. Each TCP packet that arrives at an echo host incurs overhead in the TCP stack, matching that packet with a data structure representing that TCP connection. This process is comparatively CPU intensive, and thus delay measurements encapsulated in TCP will include a component of delay introduced by the TCP stack in addition to the network delay.

Measurement traffic is subject to protocol-based priority queuing policies that may be deployed in the path between a pair of hosts. The choice of a particular protocol type for conducting measurements results in a measurement that not only considers the propagation delay of the packet, but also the effects of queuing policy. The implication of protocol-based priority queuing is that if a measurement shows a change in delay from previous measurements, the change in delay is not necessarily the result of increased network load.

D. Limited Clock Support

Packet probing techniques that measure delay to sections of a network, such as that with one-way delay, require synchronised clocks. Clock behaviour is complex and outside the scope of this paper. A good discussion of the impact of clock behaviour on delay measurements is presented in [11]. The fundamentals of clock theory are that clocks are of limited precision and that they drift at differing rates over time.

One method to address the effects of inaccurate clocks in one-way delay measurements is to insist on echoing hosts using a precise external time source [9], [4], [12] such as those provided by the Global Positioning System (GPS) and the Code Division Multiple Access (CDMA) network. These precise external time sources result in a clock synchronised to real time with an offset of a few hundred nanoseconds. These methods have limitations. GPS time receivers are expensive and there can be logistical difficulties in placing antennae. CDMA networks are not widely available at present.

In the case of one-way delay measurement, an accuracy limitation in the range of around two milliseconds can be acceptable because this limitation represents only a small proportion of delay in the actual network. Existing measurement protocols do not provide a mechanism to retrieve information about clocks located on echo systems, despite echo systems often knowing how far their clock is from real time.

IV. The IP Measurement Protocol

The IPMP protocol is based on an echo request and reply packet exchange for measuring packet delay and associated path metrics, and is similar to the technique that `ping` uses with the ICMP echo capabilities. Full detail of the protocol can be found in [5].

IPMP is carried directly inside of an IP packet in order to make an echo packet obvious to the routers connecting the hosts involved in the measurement. The echo reply packet has been designed so that an echo host can construct an echo reply packet with very few modifications to the echo request packet. The echo protocol packet format is presented in Figure 1.

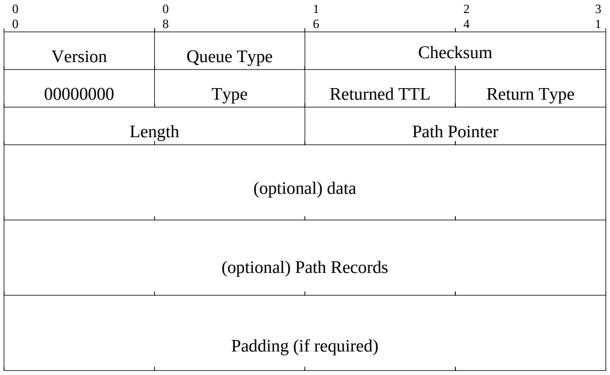

Fig. 1. The IPMP Echo Request Format

The key fields in the echo packet are as follows. The Version field identifies the version of the IPMP protocol being used. The Queue Type field identifies the protocol that the packet should be queued as (e.g., TCP). The Type field identifies the IPMP packet as one of echo request or echo reply. The Returned TTL field identifies the TTL value of the IP packet as it arrives at the destination. The Return Type field becomes the type field for the echo response. The symmetric nature of the second 32-bit word of the packet is intended to make creation of the echo reply packet more efficient. The Length field identifies the amount of data that has been allocated in the IPMP packet to store IPMP Path Records. The Path Pointer field identifies the offset at which the next IPMP path record should be inserted, assuming that there is available space.

A major difference between the echo exchange in IPMP and other echo protocols is the introduction of the IPMP path record, presented in Figure 2. Each path record structure requires twelve bytes of data to be allocated in the IPMP packet. The first four bytes of a path record represent the IPv4 address of the network interface the IPMP packet was received on. If a host inserts a path record to signify the time the packet leaves the kernel, it uses the network interface that the kernel uses to transmit the packet on. The last eight bytes of a path record is a fixed-point representation of time following the conventions of RFC 1305 [13]. The first four bytes is the integer part of the timestamp; the second four bytes is the fraction part. The timestamp represents seconds since January 1900.

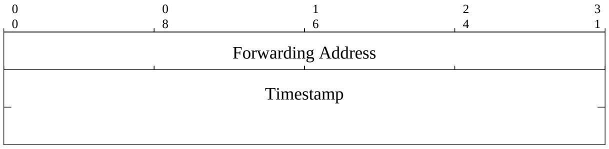

Fig. 2. The IPMP Path Record Format

The minimum packet size that must be supported by an IP network, 576 bytes, allows for 45 path records to be stored in an echo request packet. The size of the path record would increase to 24 bytes in an IPv6 environment due to the storage requirements of a 16 byte address. In an IPv6 environment, the Minimum Transmission Unit (MTU) of 1280 bytes [14] allows for 50 path records to be inserted in an echo packet. Research from CAIDA's Skitter project [15] indicates that the average distance between the F root server and a customer of that server is 13 hops, while less than 0.5% of paths are longer than 22 hops. For a path connecting a pair of hosts that is longer than can be measured with the MTU-sized packet, a measurement host may restart the measurement from one of the last IP addresses in the path record by sending an echo request to an address that will answer the request (if there is such an address), or by sending a larger packet if the underlying network supports the increased size of the packet.

In addition to the ability to infer end-to-end delay by subtracting the time that a measurement packet was sent from the time when the packet returned, the echo request packet can be used to deduce path length in hops for each direction, and one-way delay if the echo host inserts a path record when it responds to the echo request. Provided that adequate space has been allocated in the echo request packet, each router or host that handles the packet is able to insert a path record into the echo packet that signifies the IP address of the router and the time at which the packet was processed.

In addition to the echo packet exchange, the IPMP protocol contains an information packet exchange. This facility is used to retrieve information regarding precision and accuracy limitations of a clock represented in a delay measurement. In addition, the drift of the clock can be inferred with linear interpolation if multiple IPMP Real Time Reference Point (RTRP) objects, shown in Figure 3, are included in the response. The RTRP format presents a point that can be used to map between a reported time and the actual real time of a clock.

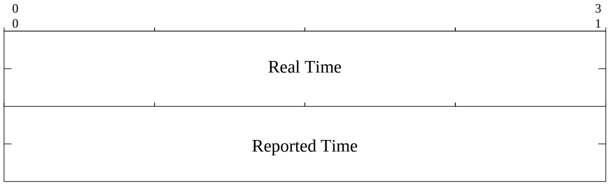

Fig. 3. The IPMP Real Time Reference Point

A discussion of how IPMP addresses the limitations identified in Section III is now presented.

A. Combines Path and Delay Aspects of Measurement

IPMP combines path and delay aspects of measurement by providing an echo packet with predetermined space allocated for routers to insert path records. This provides a measurement host with an ability to identify paths that are congested or have an otherwise lengthy propagation delay, for both the forward and reverse paths. The ability to measure to a router is important as the reverse path from the router is often different from the forward path, as is the case with hot-potato style routing policy [16].

An ISP or other network that wants to be able to discover and demonstrate the degree of delay introduced by their network can deploy IPMP path record enabled routers at the boundaries of their network. IPMP echo packets would have timestamped path records inserted as they pass through border routers in the forward and reverse path.

B. Designed for Efficient Handling by Routers

The protocol is designed so that it can be manipulated in an efficient manner. Path records are placed in allocated space in the packet. Simple word manipulations allow the router to transform an echo request packet into an echo reply packet, removing the need for the IPMP checksum to be recalculated over this portion of the packet.

This makes the protocol more suitable for a kernel implementation where timestamps can be recorded that avoid potential user-space process switching that can result in a less accurate timestamp [8]. The encapsulation of a packet probe with a separate protocol type allows for more flexible filtering and may avoid measurement activities from being blocked due to administrative policies designed to block other packets. Administrative filters may rate limit or block ICMP packets, UDP packets, and even TCP-SYN packets in order to limit the impact of a denial of service attack.

The design decision to make IPMP easy to implement for router manufacturers requires IPMP traffic to be obvious so that routers may participate in measurements in an efficient manner. This is not to say that there are no ways to engineer IPMP traffic to be less obvious, and to allow the protocol to be used where administrative blocks or filters might otherwise prevent doing so. One possible method is to encapsulate an IPMP packet with IPSec using Transport mode [17] and an Encapsulating Security Payload (ESP) header [18] to hide the IPMP protocol type. Doing so makes it impossible to collect path information in addition to the packet delay measurement and precludes a router from being a measurement target. The protocol remains useful for measurement of one-way delay between two authenticated hosts, although decrypting an echo packet introduces overhead onto the measurement hosts.

C. Provides a Mechanism for Priority-based Queuing

The IPMP echo protocol format has a queue type field as shown in Figure 1. The purpose of this field is to allow a router to queue a probe packet as if it were another protocol such as TCP, UDP, or ICMP. In effect, it creates a number of performance profiles depending on the protocol in use.

In reality, some routers may queue packets based on five values: the source and destination IP address, the source and destination port numbers, and the protocol type. At present, an echo packet contains the protocol type field and the source and destination IP addresses. The echo packet requires the addition of two 16-bit words that represent the source and destination ports to allow routers to queue IPMP echo packets based on the five-tuple.

In addition, consideration may need to be given to the effect of the Differentiated Services Field in the IPv4 header as outlined in RFC 2474 [19]. This field is a substitute for the Type of Service (TOS) field in the IPv4, and thus measurements taken with IPMP may need to consider the effects of this field in addition to the five-tuple.

A measurement needs to be made in the context of a specific protocol, as outlined in RFC 2330 [8]. IPMP provides the potential to support this by allowing the measurement to request that the measurement traffic be treated as if it were another protocol for the purpose of filtering the packet.

D. Provides a Mechanism for Exchanging Information about Clocks

The IPMP information packet provides a mechanism for a measurement host to collect information from hosts and routers that include time information in delay measurements. The information packet can be used to establish the accuracy of individual clocks that are represented in an echo packet.

In addition, this mechanism allows a measurement host to conduct measurements with echo hosts that do not have external precision time sources, but do have a mechanism to retrieve clock offset data with known accuracy limitations. The separation of timestamp generation and correction to real time reduces the effort required by a router to implement IPMP and allows more sophisticated analysis by the measurement system. An IPMP information packet allows measurement hosts to ascertain, in a simple manner, the stability of a clock over a period of time and to make efficient adjustments to the reported time with a degree of certainty.

The need for a precise pair of clocks for conducting one-way delay measurements in the Internet has been noted [8]. Indeed, some papers state that the use of NTP has a detrimental effect on one-way delay measurements [12]. That simply installing an NTP daemon on a measurement host will lead to invalid one-way delay measurements is not disputed. It is well established that an NTP daemon should not synchronise to hosts over network paths that are a significant component of a path that is being measured for asymmetrical delay properties [20].

Studies such as [11], [21] have presented algorithms for detecting clock adjustments between two hosts involved in one-way delay measurement. An experimental approach is to monitor system calls to ntp_adjtime by the NTP daemon and to make this information available to IPMP information requests. The default action of the NTP daemon is to amortise the offset from real time at a constant rate if it is less than 128ms from what it understands to be real time.

The information passed to the ntp_adjtime system call can give useful information regarding the clock's offset from real time, and an estimation of how accurate this offset is. The estimation of the accuracy of the offset value is the most interesting piece of information for measurement purposes, as a measurement host may compensate for the change in the offset over a period of time with linear interpolation if the accuracy limitation of the offset is acceptable. A measurement host may present the forward and reverse path delay with an estimated accuracy limitation, balancing the cost and utility of delay measurements [22].

V. Remaining Issues

A. Per-Hop TTL Information

When an IPMP echo packet passes through a router or host, an IPMP path record may be inserted if there is sufficient pre-allocated space available. As shown in Figure 2, the path record contains an eight byte representation of time on that router. If the seconds portion of the timestamp was restricted to be relative to midnight UTC, the timestamp would require 17 bits to represent the seconds portion of the timestamp from the current allocation of 32 at present. The first 8 bits of the remaining 15 bits available could store the TTL of the packet as it enters a router. This information would allow analysis of the echo response packet to consider the placement of a router between two points on a network without relying on all preceding routers inserting a path record.

B. The Role of Router Policy and IPMP

It is difficult to engineer a measurement packet that routers can understand and process almost as efficiently as any other IP packet without making the packet very obvious amongst other packets. This is an important facet of the IPMP echo packet, due to the desire for routers to insert IPMP path records so that per-hop delay can be considered.

A serious prospect is the issue of an administrator blocking IPMP measurement traffic entirely, rendering the protocol less useful for measuring the path between two hosts. As discussed in IV-B it is still possible to consider one-way delay by using mechanisms provided by IPSec.

VI. Conclusion

The limitations of existing packet probing techniques were discussed. The IPMP protocol was presented as a framework for discussing the issues of current packet probing techniques. The IPMP protocol addresses some of the limitations of existing protocols by providing a protocol that combines path and delay measurements, a mechanism to profile the handling of a particular protocol type, and a mechanism to retrieve echo host clock information. These mechanisms are provided in a protocol that is efficient and designed for a plausible implementation by router manufacturers.

Acknowledgements

The authors are thankful for the valuable advice and effort expended by our shepherd, Vern Paxson, and to the editorial assistance provided by Maureen C. Curran.

References

[1] J. Postel, "Internet control message protocol," RFC 792, IETF, 1981.

[2] J. Postel, "User datagram protocol," RFC 768, IETF, 1980.

[3] J. Postel, "Transmission control protocol," RFC 793, IETF, 1981.

[4] S. Shalunov, B. Teitelbaum, and M. Zekauskas, "A one-way delay measurement protocol," IPPM work in progress, IETF, 2001.

[5] A.J. McGregor, "The IP measurement protocol," http://moat.nlanr.net/AMP/AMP/IPMP/.

[6] J. Postel, "Internet protocol," RFC 791, IETF, 1981.

[7] S. Casner, "The mtrace (8) manual page," http://ftp.parc.xerox.com/pub/net-research/ipmulti/.

[8] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," RFC 2330, IETF, 1998.

[9] G. Almes, S. Kalidindi, and M. Zekauskas, "A one-way delay metric for IPPM," RFC 2679, IETF, 1999.

[10] F. Baker, "Requirements for IP version 4 routers," RFC 1812, IETF, 1995.

[11] V. Paxson, "On calibrating measurements of packet transit times," in *Proceedings of ACM SIGMETRICS*, Madison, WI, June 1998, pp. 11–21.

[12] A. Pasztor and D. Veitch, "A precision infrastructure for active probing," in *Proceedings of the PAM2001 workshop on Passive and Active Measurements*, Amsterdam, Apr. 2001.

[13] D.L. Mills, "Network time protocol (version 3): Specification, implementation and analysis," RFC 1305, IETF, 1992.

[14] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification," RFC 2460, IETF, 1998.

[15] Cooperative Association for Internet Data Analyis (CAIDA), "Hop count distribution," http://www.caida.org/tools/measurement/skitter/RSSAC/.

[16] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 601–615, 1997.

[17] S. Kent and R. Atkinson, "Security architecture for the Internet protocol," RFC 2401, IETF, 1998.

[18] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC 2406, IETF, 1998.

[19] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services field (DS field) in the IPv4 and IPv6 headers," RFC 2474, IETF, 1998.

[20] K.C. Claffy, G.C. Polyzos, and H-W. Braun, "Measurement considerations for assessing unidirectional latencies," *Internetworking: Research and Experience*, vol. 4, no. 3, pp. 121–132, 1993.

[21] S.B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *Proceedings of 1999 IEEE INFOCOM*, New York, NY, Mar. 1999.

[22] A.J. McGregor and H-W. Braun, "Balancing cost and utility in active monitoring: The AMP example," in *Proceedings of INET 2000*, Tokyo, Japan, July 2000.