IPMP: IP Measurement Protocol

Matthew Luckie and Anthony J. McGregor

Abstract— **We present IPMP, the IP Measurement Protocol, which has been specifically designed to support active measurements and to address some of the limitations of existing measurement techniques. IPMP supports the measurement of path and delay in a single packet exchange as well as the exchange of information about the clock on hosts participating in the measurement. The IPMP packet formats are router friendly, allowing delay measurement to routers in a method that provides denial of service protection. The protocol is designed to allow elegant kernel-level and line-card timestamping. The protocol is currently used to make measurements in the NLANR AMP system, where it is used to measure delay over approximately 12,000 paths, mostly on the vBNS and Internet2 high performance networks. This paper presents and discusses the key features of IPMP and how they are implemented. This paper also presents some analysis comparing measurements taken with the ICMP echo facilities to those taken with the IPMP echo facilities.**

I. Introduction

As the Internet grows in scale and complexity, the need for measurement increases. One form of active measurement is packet probing. Packet probing involves introducing packets into the network and measuring the way the network handles those packets.

Probe packets are normally encapsulated within protocols such as the Internet Control Message Protocol (ICMP) [1], the User Datagram Protocol (UDP)[2], or the Transmission Control Protocol (TCP) [3]. These protocols are general purpose protocols and were not designed for use in packet-probing measurement. They have limitations when used for this purpose [4].

Current packet probing techniques are not suited to measuring packet delay at the router level. 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 measurement techniques construct measurement traffic that can be difficult to efficiently detect and respond to amongst other network traffic. This type of measurement traffic precludes measuring to a router and makes the task of identifying *where* delay occurs in the network difficult.

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, existing protocols do not provide a mechanism to retrieve this information. The IP Measurement Protocol (IPMP) [5], which is the topic of this paper, is designed specifically for packet delay and path measurement and is intended to address the limitations of existing measurement protocols.

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. 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.

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. It is 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.

The approach taken by IPMP is similar to an ICMP echo request and echo reply packet exchange. As an ICMP echo packet passes through the network, timestamps and router IP addresses may be added to the packet. One of the goals of IPMP is that it be suitable for implementation in a router. IPMP has been designed to be tightly constrained, efficient, and easy to implement.

Deployment in routers is not straight forward. Even if all the technical issues were solved, there are commercial and quasi-political issues to be addressed. IPMP has an 'opt-in' philosophy, and will still operate effectively even if no routers support it on the path between a pair of hosts, so long as the packet is forwarded. Even in this case IPMP has advantages over other measurement protocols.

At present, the protocol is implemented as a patch to the kernel of the FreeBSD and NetBSD operating systems. A Linux kernel implementation is partly completed. The protocol has been deployed in the National Laboratory for Applied Network Research (NLANR) Active Measurement Program (AMP) infrastructure [6] for about a year, where

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

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

it is used to measure delay and loss of approximately 12,000 paths each minute. The paths involve hosts that are connected to to National Science Foundation (NSF) awarded High Performance Connection (HPC) networks such the vBNS and the Internet2. A significant amount of data has been collected in this time. It is the intention of AMP team to replace the ICMP measurements currently collected by AMP with a well engineered implementation of IPMP that uses the capabilities of the protocol to greater effect in order to provide more insight than is currently available with the group's ICMP based measurement infrastructure.

The intention to use IPMP instead of ICMP raises several important questions regarding the impact of protocol change on the results collected. It is established that measurements need to be made in the context of a specific protocol, as outlined in RFC 2330 [7]. Many network administrators view ICMP as a protocol that can be used in denial of service attacks and thus rate-limit or assign a low priority to the ICMP packets that their border-routers process. This paper discusses the impacts that a change in measurement protocol may have on the results collected by the AMP infrastructure.

This paper is organised as follows. A detailed discussion of the IPMP protocol is presented in Section II. Section III presents the method of data collection and preliminary analysis. In Section IV we discuss future work that we plan on conducting with the protocol and other directions of research, and conclude in Section V.

II. The IPMP Protocol

IPMP measurements are based on an echo request and echo reply packet sequence. A measurement system performs a measurement by constructing an echo request packet, sending it to an echo host, and analysing the response packet.

Packet probes have an inherent end-to-end view of a network. The rationale for an end-to-end measurement is that the performance measured is the performance experienced by an end user of the network and thus is a good way to approach delay measurement. One-way measurements take the end-to-end view of a network and divide the network into two segments, the forward path and the reverse path. One-way measurements provide a deeper insight into the performance of the network, although the insight is not sufficient to determine the paths that contribute to the delay measured and experienced. Ideally, one could conduct an end-to-end measurement and quantify *where* delay is occurring. IPMP achieves this by requesting routers to insert path information into IPMP echo packets that identifies the IP address of the interface card the packet was received on, and the time the packet was actually received. The path information that each router is asked to insert is known as an IPMP *Path Record*.

In addition to the echo packet exchange, IPMP contains an information request and reply mechanism. A measurement host may retrieve additional data from each host that inserts path information in an echo packet by composing an information request. The information response contains information about the accuracy of the clock on the host, its relationship to real time, and the overhead incurred in processing an echo packet.

Full details of the protocol can be found in the IPMP protocol specification [5]. In this section we introduce the echo request, echo reply, information request, and information reply packets. The fields in each packet are explained and the various quirks of IPMP are discussed. Before doing so, we discuss the implications on IPMP of encapsulation directly inside an IP packet.

A. Encapsulation of IPMP

IPMP is an end-to-end measurement protocol that seeks the cooperation of routers connecting a pair of hosts so that deeper insights can be obtained. The desire to obtain the cooperation of routers in a packet probe requires that IPMP be tightly constrained and easy to implement. In order to make the protocol easy to implement from the router's perspective, measurement traffic must be obvious so that it can be processed in an efficient manner at the router's line card. To facilitate this, IPMP is carried directly inside of an IP header. Encapsulation of the IPMP packet directly inside of an IP packet has several other advantages in addition to the advantage noted.

Firstly, it makes the protocol more suitable for a kernel implementation where timestamps can be recorded that are exempt from potential user-space process switching that may result in a less accurate timestamp [7]. Secondly, the use of a separate protocol number provides scope to allow measurement packets to request to be queued as if they were another type of packet, such as a DNS name lookup packet, in order to understand how the network is likely to treat these packets compared with packets of other types. Thirdly, and related to the second point, the use of a separate protocol number allows for more flexible filtering and may avoid measurement traffic being blocked by administrative policies designed to block denial of service attacks. Administrative filters may rate limit or block ICMP packets, UDP packets, and TCP-SYN packets to limit the impact of a denial of service attack. While administrators may also choose to apply these policies to IPMP, it is not necessary that they do so.

Some are concerned that making measurement traffic visible can lead to avoidance or manipulation of the traffic. The visibility of IPMP measurement traffic in its raw form does not mean that there are not ways to engineer IPMP traffic to be less obvious to allow the protocol to be used where administrative blocks or filters might otherwise prevent doing so. One method is to encapsulate an IPMP packet with IPSec using Transport mode [8] and an Encapsulating Security Payload (ESP) header [9] to disguise the IPMP protocol type. Doing so makes it impossible to collect path information from routers involved in the packet exchange and introduces overhead on the echo hosts involved in the measurement, but still enables one-way delay measurements to be conducted. Another approach is to occasionally validate normal IPMP traffic against measurements conducted with other protocols such as ICMP `ping`, OWAMP [10], or IPMP echo encapsulated with IPSec.

B. The Echo Packet

The IPMP echo service is similar to the ICMP echo service in that the target is expected to respond to the echo request with an echo reply. 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 IPMP echo packet exchange can be used to deduce path length in hops for both the forward and reverse paths and, in the case where synchronised time sources are available on the pair of hosts involved, one-way delay. The mechanics of how IPMP extracts this information from the echo packet exchange is now discussed.

A measurement host constructs an echo request packet by creating an IP packet of type `IPPROTO_IPMP` with enough space for the IPMP packet it is to send. The IP Header's 'Don't Fragment' (DF) bit is set so that routers will not fragment the packet if the size of the packet is too large for the maximum transmission unit (MTU) of the link. The DF bit is necessary as routers are not able to add a path record to a fragmented packet. A host that sends an IPMP packet larger than 576 bytes (the IPv4 MTU) risks having that packet discarded by the network.

The format of the echo packet is shown in Figure 1. The *Version* field is set to identify the version of the IPMP protocol, which is currently zero. The *Protocol Queue* is set to identify how the packet should be prioritised at routers that maintain priority queues. For example, if the protocol queue field is set to `IPPROTO_TCP`, the echo packet requests to be prioritised as a TCP packet for measurement purposes. The *Checksum* field is a standard 16-bit Internet checksum for protocol headers.

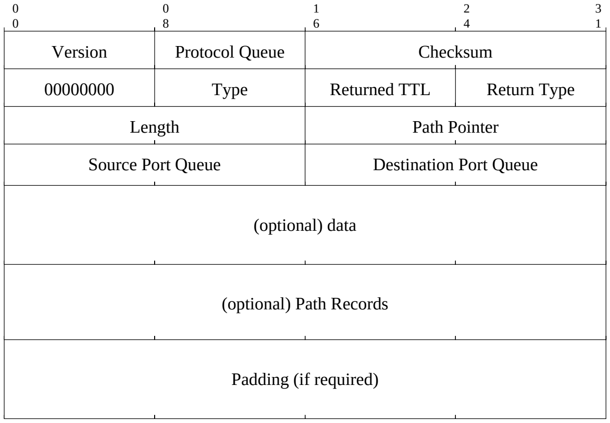


Fig. 1. The Echo Packet

The *Type* field identifies if the echo packet is specifically an echo request packet or an echo response packet as per the types presented in Figure 2. The *Returned TTL* value is set to zero in the echo request packet. The *Return Type* field is set to be the expected value of the type field in a response. For an echo request packet, the return type field is set to the value of an echo response packet as per Figure 2. These 4 bytes (the filler byte, the type, the returned TTL, and the return type) are allocated in such a way that allows the echo request to be transformed into an echo response packet without requiring the IPMP checksum to be recalculated. This transformation occurs by swapping the two 16-bit words in place, which has the effect of swapping the *return type* and *type* values. If the returned TTL value is updated by the target host, the IPMP checksum can be updated incrementally as per [11].

Type	Value
Echo reply	0
Echo request	1
Information request	32
Information response	33

Fig. 2. IPMP Packet Types

The *Length* field identifies the preallocated boundary in the echo packet into which the path records are inserted. The *Path Pointer* identifies where in the packet the next path record (explained in Section II-C) should be inserted. For each path record that is inserted, the path pointer is incremented by the amount of space that path record uses. Finally, the *Source Port Queue* and *Destination Port Queue* fields are used if the *protocol queue* is set to `IPPROTO_TCP` or `IPPROTO_UDP` and allows the IPMP echo packet to be queued based on the five-tuple. The five-tuple is used to decide how a packet should be queued or filtered, and is composed of source and destination IP address, IP protocol type, and source and destination port.

C. The Path Record

As described in Section II-A, IPMP seeks the cooperation of routers connecting a pair of hosts that exchange echo packets. IPMP echo packets contain preallocated space for routers connecting a pair of hosts to insert a path record. As shown in Figure 3, the path record structure contains a 64-bit timestamp and an IPv4 address. The *Forwarding Address* represents the IPv4 address of the interface the echo packet was received on. The *Timestamp* is a fixed-point representation of time that follows the conventions of RFC 1305 [12]; the first 32-bits represent the time in seconds since January 1900; the second 32-bits represent the fraction part.

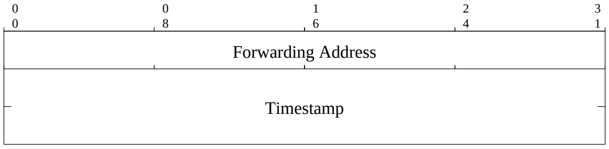


Fig. 3. Path Record Format

Each path record structure requires 12 bytes of data in the IPMP echo packet. A 576 byte IPv4 packet contains sufficient space for 45 path records to be inserted in an echo packet. Larger IP packets are at risk from being discarded by the network, although measurement hosts can gain a degree of confidence in the interconnecting networks through path maximum transmission unit (PMTU) discovery [13], and are free to allocate larger IPMP echo request packets.

In order for a router to make an effective target for IPMP measurement, path records should be inserted at the line card and not passed to the central CPU. Addition of a path record is designed to be a simple and efficient process. Several optimisations are available at the line-card level, such as the incremental update of the IPMP checksum [11]. The packet forwarder must check to ensure there is available preallocated space in the echo packet. The IP address of the line-card is known in advance, and the value of the incremental update to the IPMP checksum as a result of inserting the address can be precalculated due to the preallocated space for path records being initialised to zero. The load and store of a suitable timestamp is perhaps the most challenging operation as the incremental update to the IPMP checksum is not able to be predicted. We estimate that a path record can be inserted in approximately 12 machine instructions.

It is recommended that the measurement host insert an initial path record to the echo request and append a final path record to the echo response. The two path records allow the measurement host to be more certain that the echo packet was sent and subsequently received close to this time and was not subject to process switching which can affect the accuracy of timestamps created in user-space. In addition, this pair of path records allow the identification of the interface the measurement host sent and subsequently received the echo packet on.

D. Echo Packet Dynamics

An IPMP echo request occurs as outlined in Figure 4. A measurement host performs an echo measurement by creating an echo request packet, addressing it to the target host, and sending it onto the network. As the packet is sent, the operating system's kernel may insert a path record as shown in step one of Figure 4. The echo request is then routed through the network. If the packet encounters a router is IPMP capable, the router may insert a path record that signifies the time the echo packet arrived and the interface it arrived on, as shown in step two. If a router is not IPMP capable, as shown in step three, the packet is forwarded without the insertion of a path record.

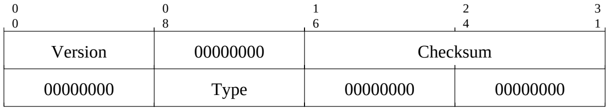
When the packet arrives at the echo host, the echo host may insert a path record and record the current TTL value of the IP packet in the *Returned TTL* field, before it transforms the echo request into an echo response and sends the packet back. Note that the echo host does not reset the TTL value in the IP header of the echo packet. The echo response is then routed along the reverse path, toward the measurement host, where IPMP capable routers may insert path records. When the packet arrives back at the original measurement host, a final path record may be inserted.

At the application layer, path information may be extracted from the response and preliminary results extracted. It is important to note that IPMP has captured path information from both the forward and reverse paths in a *single packet exchange*, which cannot be done with measurement techniques such as `traceroute`. The path information collected indicates the placement of routers on both the forward and reverse paths.

If the measurement involves the calculation of delay between a pair of points that do not share the same clock, the final stage of measurement is to send an information request, as will now be discussed.

E. The Information Request

The information request packet is designed to procure an information response that will allow measurements to be corrected for clock drift and for timestamps that are not supplied in UTC. The information request packet also provides the ability for clocks with known, but limited accuracy, to be used in a one-way or even per-hop measurement. If multiple echo packet exchanges occur over a relatively small timescale, the information exchange may cover more than one echo packet exchange.



00	08	16	24 31
Version	00000000	Checksum	
00000000	Type	00000000	00000000

Fig. 5. The Information Request Packet Format

The information request packet is presented in Figure 5. The information request is expected to be answered at a low priority and is designed for user-space implementation. For this reason, the information request is not designed to be transformed into an information response as the echo request was designed to become an echo response. The *Type* of an information request is 32, as outlined in Figure 2.

F. The Information Response

The data contained in an information response packet can be used to correct for clock drift and other timing issues as outlined in Section II-E. The format of the information response packet is presented in Figure 6.

The *Type* field is set to 33 as per Figure 2. The *Precision* field identifies the number of bits in the fractional part of timestamps that are valid. The *Length* field is set to the total length, in bytes, of the IPMP packet. The *Performance Data Pointer* field identifies the position, in bytes from the beginning of the IPMP packet, of any performance data included in the information response packet. The *IP Address* field is a single IP address by which the router identifies itself. This field is used to identify routers in the network that may be represented by multiple path records in an echo packet, each with a different interface address.

The *Accuracy* field signifies the maximum difference between actual real time and the inferred real time of any timestamp generated by the interface. The accuracy field allows the timestamp to be presented with error bounds. The *IPMP Processing Overhead* field signifies the maximum difference between the time taken to process and forward an echo packet and the time taken to forward an IP

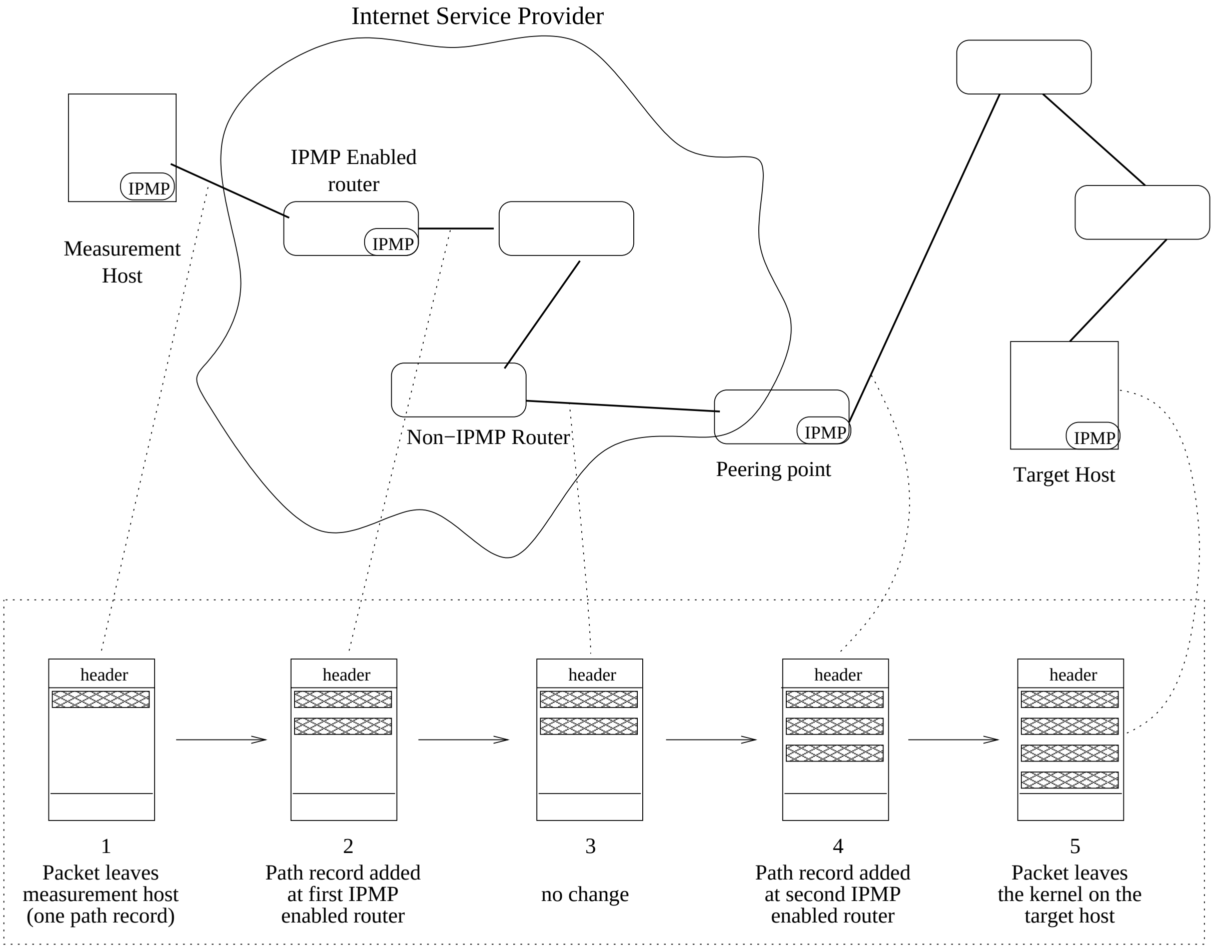

Internet Service Provider
IPMP
Measurement Host
IPMP Enabled router
IPMP
Non−IPMP Router
IPMP
Peering point
IPMP
Target Host
header
header
header
header
header
1
Packet leaves measurement host (one path record)
2
Path record added at first IPMP enabled router
3
no change
4
Path record added at second IPMP enabled router
5
Packet leaves the kernel on the target host

Fig. 4. Progression of an IPMP echo request packet through a network

packet with the same characteristics. The *Real Time References Points* field is a variable length field that includes multiple real time reference points. The format of an individual real time reference point is presented in Figure 7. Finally, the *Performance Data* field allows arbitrary information from the MIB of the system or the interface to be included.

In order for a measurement host to compensate for clock drift at a particular point in time, a pair of real time reference points must be available. The IPMP information exchange allows a router or a host with a free-running clock to be used as a target for one-way delay measurement if it has access to an external real-time source and can report clock offset as required. There is also the potential to use the information exchange facility for hosts that are synchronised with NTP which may have traditionally made poor one-way delay measurement targets without this information available.

Studies such as [14] and [15] have presented algorithms for detecting clock adjustments for clocks involved in a measurement. An IPMP information response packet allows measurement hosts to ascertain, in a simple and elegant manner, the stability of a clock over a period of time. It allows efficient adjustments to be made to the reported time in a path record with error and accuracy bounds.

G. Implementation Details

The minimum requirements for IPMP, that is the implementation of the echo packet types, have been implemented in the kernel of the FreeBSD operating system and deployed in NLANR's Network Analysis Infrastructure (NAI) as of February 2001.

The reference implementation consists of a modification to the kernel's IP protocol switch table, one loadable kernel module, and a user-space application similar to the well-known ICMP-based `ping`. The protocol is designed so that all time critical sections of code will run in kernel-space, while less critical sections will be passed to user-space for more efficient processing. In practise, this required three modifications to the FreeBSD kernel for the protocol to operate effectively.

Firstly, an entry for IPMP in the IP Protocol switch table (`in_proto.c`) was created so that IPMP packets are passed to the relevant input routine (`ipmp_input`). Secondly, a modification to the `ip_forward` function was made so that IPMP path records are inserted when forwarding an IPMP packet. These first two modifications are required to be statically compiled into the kernel. The last modification, a system call that creates and sends an IPMP echo request with an initial path record, was achieved by creating the loadable kernel module.

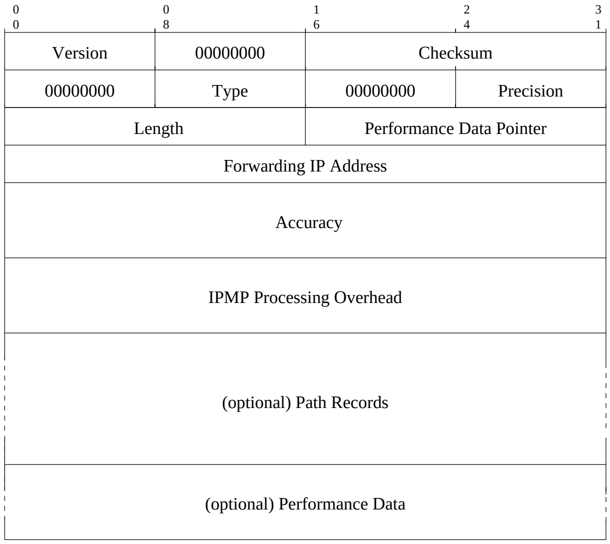


Fig. 6. The Information Response Packet Format

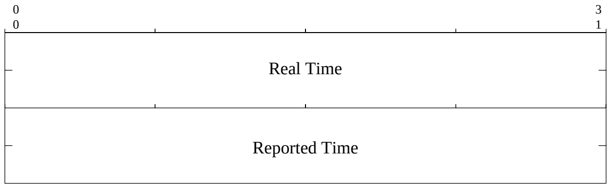


Fig. 7. The Real-Time Reference Point

The application program, `ipmp_ping`, opens a raw IPMP socket, uses the system call described previously to send an IPMP packet, and then waits for a response on the socket. In order to test and to develop IPMP, an IP protocol number that is not currently assigned was chosen. It is interesting to note that the protocol number used, 169, has been noticed by other researchers in passive traces and the like, and has resulted in a small number of enquiries regarding the nature of the traffic.

III. Preliminary Results

IPMP has been in use in the NLANR AMP system for one year where it is used to measure delay and loss across approximately 12,000 paths each minute. The paths are between hosts connected to National Science Foundation (NSF) awarded High Performance Connection (HPC) networks such the vBNS and the Internet2. It is the intention of AMP researchers to replace the ICMP delay and loss measurements that are currently collected with IPMP measurements, and to use the capabilities of IPMP to greater effect.

This section compares the differences seen with delay and loss measured with the IPMP echo protocol compared with the ICMP echo protocol. The data used in this paper was collected on Tuesday, 22nd of January 2002 PST. Each minute, an ICMP echo request was sent to all other AMP machines, followed approximately thirty seconds later by an IPMP echo request.

A. Delay Comparison

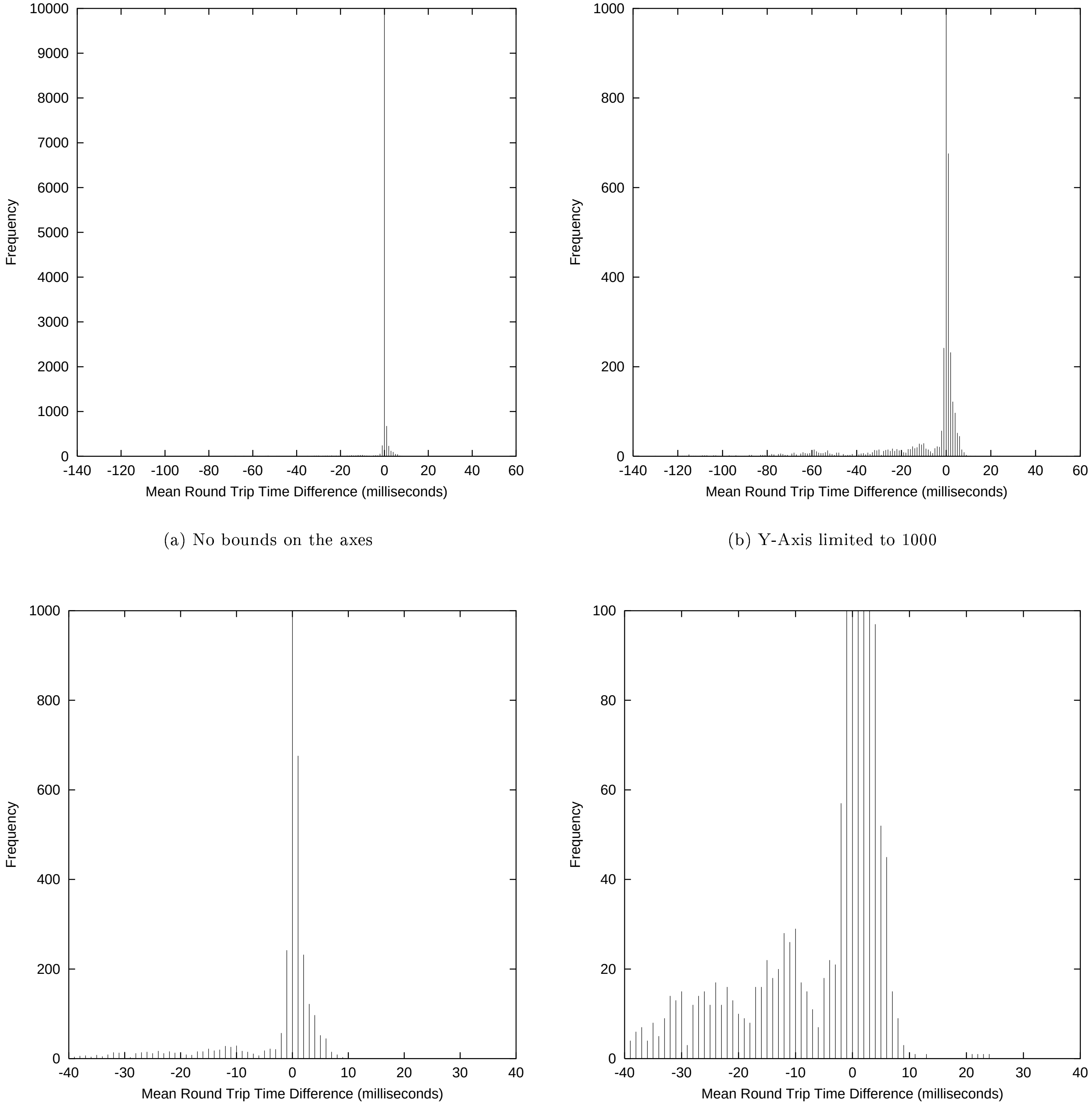
Figure 8 gives an overview of how, for each path, measured delay with the IPMP echo packet compares to measured delay with the ICMP echo packet. If the average measured delay difference is not zero, the path is assumed to potentially show some degree of bias. There are 12309 samples in Figure 8, with 9961 (81%) paths showing no bias. Of the paths that showed some apparent bias, 1092 (8%) returned smaller average delays when measured with IPMP echo packets, while 1259 (10%) paths returned larger average delays when measured with IPMP echo packets. It is not entirely clear why there are slightly more paths showing higher average delay values measured with IPMP compared to delay values measured with ICMP. It is possible that the difference is random noise, as the difference of 167 paths is less than would be affected by a single AMP machine having it's IPMP packets rate limited.

An examination of Figure 8 reveals significant asymmetry. For example, 682 paths (5.5%) show at least a 10msec smaller average delay with IPMP echo packets than with ICMP echo packets, while only 8 paths (0.06%) show at least a 10msec larger average delay with IPMP echo packets than with ICMP echo packets. There is a long tail to the left of zero, where IPMP measurements are more likely to return lower delay values in the AMP environment. This indicates that there is a significant ICMP rate limiting among the paths that AMP measures. The presence of ICMP rate limiting places doubt over the value of conducting ICMP measurements over these paths with significant rate limiting.

We have found no evidence of IPMP measurements being rate limited more severely than ICMP measurements, although we have not compared either ICMP measurements or IPMP measurements to measurements encapsulated in UDP. All AMP machines were deployed with the understanding that ICMP packet exchanges with the machine would be permitted. We have, as Figure 8 would indicate, found several sites that appear to rate limit ICMP traffic as a result of comparing ICMP delay measurements with IPMP delay measurements.

We present four graphs in Figure 9 that include both ICMP delay measurements and IPMP delay measurements for a single path for a single day. These four paths were chosen as interesting due to the difference in the average delay value for that day. Figures 9(a), 9(b), and 9(c) show significant variation in ICMP delay measured compared to delay measured with IPMP.

Figure 9(a) shows what appears to be consistently unstable ICMP delay values over the day for that path, while the IPMP delay measured is relatively consistent. Figures 9(b) and 9(c) indicate that measured ICMP delay is similar to the delay measured by IPMP for significant periods of time over the day for that path. These two figures show that for some periods of time, delay measured by exchanging ICMP echo packets is significantly higher than the delay measured by exchanging IPMP echo packets. Figure 9(d) is highlighted as one site that showed an average 2msec higher delay measured with IPMP than with ICMP.



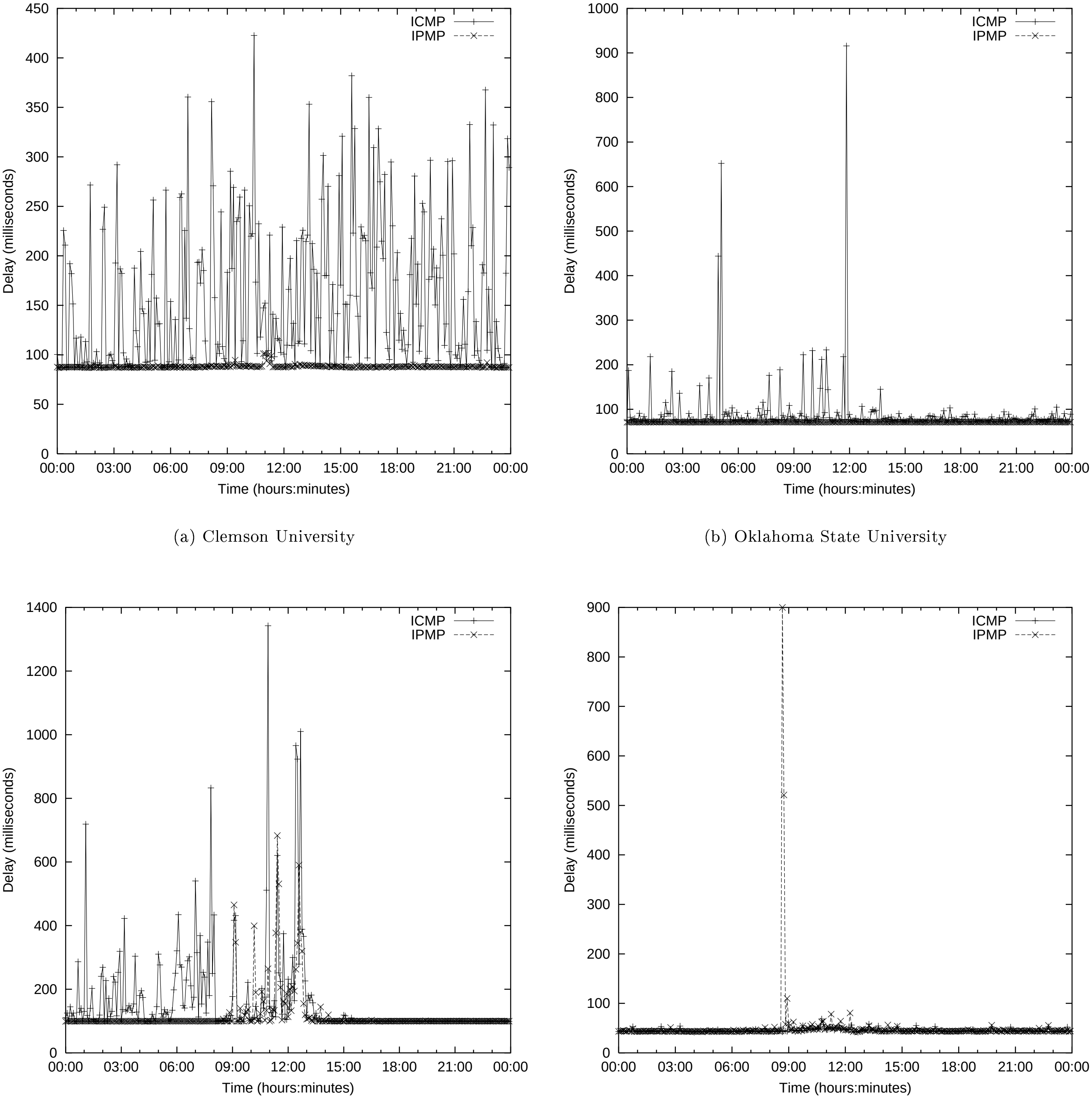
(a) No bounds on the axes

(b) Y-Axis limited to 1000

(c) X-Axis limited to 40, Y-Axis limited to 1000

(d) X-Axis limited to 40, Y-Axis limited to 100

Fig. 8. Average round-trip-time per-path, measured with both IPMP and ICMP on Tuesday, 22 January 2002. Each point signifies the average delay for a five minute period. Measurements were taken every minute for both protocols, although IPMP and ICMP probes were separated by 30 seconds. Bars to the left of zero signify that the IPMP measurement showed, on average, a smaller RTT than the ICMP measurement for this day.



(a) Clemson University

(b) Oklahoma State University

(c) University of Connecticut

(d) Oregon State University

Fig. 9. IPMP and ICMP delay measurements between various hosts and the AMP monitor located at San Diego Supercomputer Center (SDSC) on Tuesday, 22 January 2002. Each point represents the average delay seen for a five minute interval with each protocol.

B. Loss Comparison

While we did not find examples of IPMP packets being rate limited as severely as was observed for ICMP packets, we found five sites that filter IPMP packets (as of 22 January 2002). Excluding the sites that filter IPMP, we have not found that IPMP packets are either more or less likely to be lost than ICMP packets, as illustrated by Figure 10.

It is not clear how similar the results would be if data were collected using IPMP in a setting outside of AMP, such as the commercial Internet. All AMP sites were installed with the understanding that ICMP traffic would not be filtered. Some sites that initially filtered IPMP have allowed the protocol to communicate with the AMP machine after discussion with the site administrator.

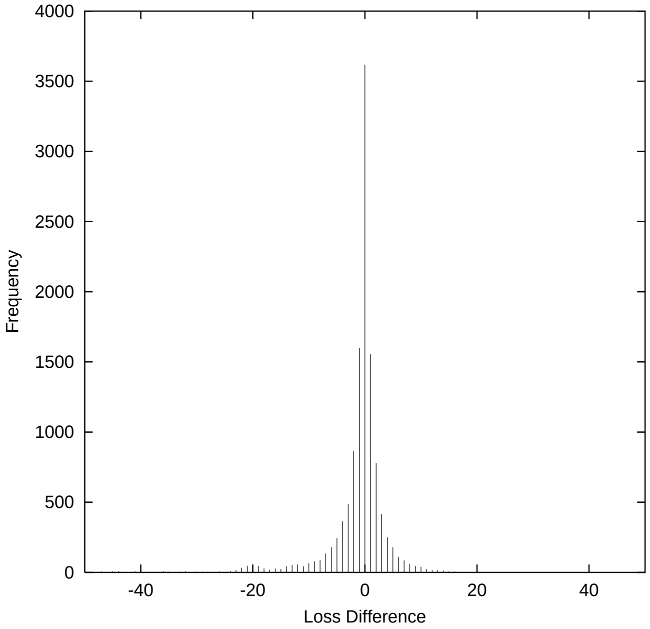


Fig. 10. Differences in Loss Rates between IPMP and ICMP measurements. Each point signifies the difference the number of packets lost for either protocol, if any were lost. Bars to the left of zero signify that less IPMP packets were lost than their ICMP counterparts for these paths.

IV. Future Work

The AMP system is not currently collecting all of the data that is potentially available with the IPMP protocol, such as the actual timestamps recorded in echo packets and the forward and reverse path lengths in hops. At the time of writing, the AMP system is undergoing a transition from the FreeBSD 3.0-RELEASE operating system to FreeBSD 4.5-RELEASE. After the AMP system has completed this transition, work will shift towards collecting much more detailed data with IPMP than is currently collected.

To support the capture of detailed data capable with the IPMP protocol, a specialised measurement daemon has been designed and partly implemented. The daemon's design is focused towards capturing one-way delay measurements with the IPMP protocol using the AMP architecture. There is a desire to use the existing AMP systems, none of which have precise external time sources directly attached, to conduct one-way delay measurements with known accuracy limitations.

V. Conclusion

The work conducted with IPMP so far has demonstrated that the protocol provides utility over existing protocols. IPMP can capture path information, with router manufacturer support, from both the forward and reverse paths in a single packet exchange. The IPMP echo packet can be modified in an efficient manner, making for a plausible implementation by router manufacturers. Even if no routers in the path support the protocol, IPMP provides advantages over existing protocols by capturing information such as path length in hops for both the forward and the reverse paths. The work conducted with IPMP so far has demonstrated that the protocol is deployable and usable in environments similar to AMP.

IPMP allows the echo packet to request to be treated as if it were another packet type for the purposes of rate limiting the packet. IPMP also provides a mechanism to retrieve information about the synchronisation of a time source used in the measurement to real-time, allowing clocks with known accuracy limitations to be used in one-way delay measurements.

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] M.J. Luckie, A.J. McGregor, and H-W. Braun, "Towards improving packet probing techniques," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001, pp. 145–151.

[5] A.J. McGregor, "The IP Measurement Protocol," Available online at `http://moat.nlanr.net/AMP/AMP/IPMP/`, 1998.

[6] 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.

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

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

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

[10] S. Shalunov, B. Teitelbaum, and M. Zekauskas, "A One-way Active Measurement Protocol," IPPM work in progress, IETF, 2001.

[11] Rijsinghani A., "Computation of the Internet Checksum via incremental update," RFC 1624, IETF, 1994.

[12] D.L. Mills, "Network Time Protocol (version 3): Specification, implementation and analysis," RFC 1305, IETF, 1992.

[13] J.C. Mogul and S.E. Deering, "Path MTU Discovery," RFC 1191, IETF, 1990.

[14] 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.

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