# Passive Measurement of One-way and Two-way Flow Lifetimes

DongJin Lee, Nevil Brownlee
Department of Computer Science
The University of Auckland
dongjin@cs.auckland.ac.nz, nevil@auckland.ac.nz

## ABSTRACT

Flow based analysis has been considered a simple and effective approach in network analysis. 5-tuple (*unidirectional*) flows are used in many network traffic, however, often these analyses require bidirectional packet matching to observe the interactions. Separating the flows into two categories as *one-way* (packets in one direction only) and *two-way* (packets in both directions) flows can yield further insight. We have examined traces of *Auckland* traffic for 2000, 2003 and 2006, and analyzed their one-way and two-way flows. We observed several behaviors and the changes in flow sizes and their lifetimes over time. In our traces, we observe that one-way flows are mostly malicious, re-transmissions, and some are long-lived. Two-way flows are mostly normal end-to-end transmissions with their lifetimes/RTTs decreasing, their sizes increasing, and many short-lived flows mostly depict errors in TCP. Also, we observe similarity between one-way and two-way flow sizes for their lifetimes.

## Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations – Network monitoring

## General Terms

Measurement, Performance, Design

## Keywords

traffic meter, one-way flow, two-way flow, passive measurement, flow lifetime

## 1. INTRODUCTION

Ever since the introduction of the 'flow' terminology in network traffic analysis, many researchers and interest groups have used their own representations of a flow. Most researchers use 'flow' to mean 5-tuple CPB IP flows[1] [14]. Also, there are several other types of flows that are often used such as NetFlow [1] and CoralReef [2]. The IETF's IPFIX working group [4] defines the term 'flow' slightly different to the typical 5-tuple flows. It considers that a flow can be as simple as matching a single IP header field or can be more complex, for example including MPLS label, providing more universal standardized representations of a flow. All flows discussed so far are regarded overall as unidirectional. Many network analyses are based on these flows mainly due to their simplicity in measurement.

[1] Using source address, destination address, source port, destination port and protocol to group packets into a single flow

Recently, papers such as [16, 18] used 5-tuple flows to characterize, cluster and identify network behaviors and applications, including hard-to-detect P2P and malicious traffic. It is also possible to forecast [22] and model [19] traffic flows using wavelet and principal component analysis. Further, to have a more detailed analysis of the network traffic, mice and elephant flows are studied. Additionally, a flow lifetime analysis [12] can yield important information about today's network. Such information is useful in traffic prediction, traffic engineering to support QoS and also in accounting. The definition of Streams, Flows and Torrents in NeTraMet [13] regards network traffic as being made up of bidirectional flows, known as *streams*. Flows may originate from outside our network (inbound) or inside it (outbound), they are observed inside the network's gateway router.

The authors in [12] summarized streams by lifetime as: *dragonflies* lasting less than 2 seconds, *short streams* lasting up to 15 minutes and *tortoises* lasting more than 15 minutes. They found that most of the streams are dragonflies carrying low total bytes, and a few streams are tortoises carrying high total bytes. Furthermore, streams with six or fewer packets could be ignored to achieve higher performance due to the reduced processing cycles, while ignoring only 2% of the total network traffic [11]. In network communications, most host-to-host interactions follow end-to-end principles, i.e., packets are exchanged in both directions. Thus, a TCP connection by default is a bidirectional flow since it relies on data packets being acknowledged. While UDP and other types of protocol may not follow end-to-end principles, communicating hosts mostly recognize each other by exchanging packets at some point in time. In fact, especially in an edge network where both directions of packets can be observed, bidirectional flow entities are crucial factors to understand network traffic. For instance, a host can suffer from enormous amounts of malicious traffic (e.g., DDoS). That is, attack flows such as port scans may produce significant numbers of flows. Experiments conducted by the authors in [10] showed the characterization of traffic flow anomalies by gathering 5-tuple flows and separating them out as inbound vs outbound flows. Also in [17], a ratio of inbound and outbound packet transmission (Packet Symmetry) is examined to 'curtail' malicious traffic. Thus, studying only one side of flows may easily ignore the end-to-end behavior by not being aware of the opposite host's response.

We are particularly interested in flow *lifetime* distributions and their changes over the years. Here, our early approach with unidirectional flow or stream lifetimes exposed some questions. For instance, consider Figure 1 showing one-day Cumulative Distribution Functions (CDFs) of typical flows, bytes and packets vs their lifetimes in 2006. The overall distributions can be observed, but they give no insight as to how or why such variances occur, such

as the existence of many short-lived flows (e.g., ≤ 10us) or sudden steps in traffic contributions (e.g., > 0.1s). To our knowledge, such issues have not been addressed. For instance, the authors in [24] used various flow analysis techniques (e.g., using T-RAT) to understand flow rates and sizes, but disregarded flows that contain a single packet or had short durations (i.e., < 100ms). Additionally, we found that many of the aforementioned 5-tuple flow analyses *necessitated* the use of bidirectional flows at some point in their studies (e.g., classifications, flow interactions, etc), which often require building *separate* bidirectional flow tables from the already-collected flows. Overall, unidirectional flows alone are inadequate to study end-to-end behaviors. Unfortunately, flow lifetime studies have not considered whether the flows is unidirectional or bidirectional, often assuming that all of the flow lifetimes exhibit end-to-end behavior.

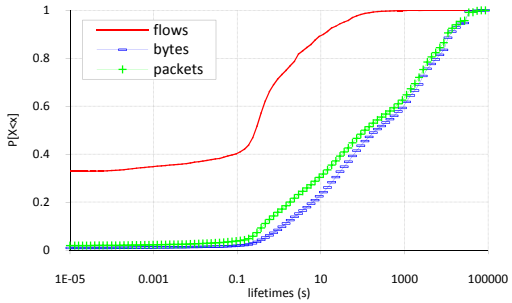



**Figure 1:** CDFs of total flow, byte and packet lifetimes observed for 24 hours at *Auckland* in 2006.

Our initial attempt to analyze our traffic data resulted in two types of flows: unidirectional and bidirectional. As a result, the motivation for our study is to attempt to resolve following two main issues encountered in the analysis of flow lifetimes over the years:

i. Changes of flow sizes in bytes/packets, their lifetimes, and their applications/protocols
ii. Finding an effective way to build flow tables that can make it easy to measure flow attributes such as lifetimes and RTTs.

We do not claim that the using bidirectional flow is an innovative or unusual approach. However, we believe that having a bidirectional flow entities yields a better platform for flow analysis. Here, our term used to distinguish two types of flows is somewhat similar to the IETF's RTFM architecture [8]. However, NeTraMet's approach includes unidirectional flow as a *subset* of a stream, regarding practically any flow to be a stream. For these reasons, our work uses two new kinds of flows: *one-way* and *two-way* flows. We build a table of flows while capturing packets, regarding the unidirectional 5-tuple flow as a one-way flow initially, and when we see a packet for this flow in its opposite direction, we change it to a two-way flow. Thus, our one-way flows can only exist when they contain no opposite direction packets. They become two-way flows when the opposite direction is presented on the fly. Closer to our study, the authors in [14] stated that unidirectional flows often show very significant asymmetries in the two directions, hence manipulating the unidirectional flows first, then transforming them into bidirectional flows during the analysis process is a preferred approach. However, we manipulate both types of flow *at once*.

Using a packet header from the traces, our work highlights some lifetime issues for the flows with both one and two directions. First, we show that a separation of one-way and two-way flows is simple and achievable. For instance, the flow table entries can effectively be reduced to half of the traditional entries provided that each entry stores data for both directions, thereby increasing the speed of the hash lookups. Intuitively, more fine-grained studies can be *readily* available with a two-way flow table. For instance, the entry in the table contains useful information such as inter-arrival time, lifetime and RTT, involving no further processing once the packets are processed into the flow table. All are essential tools in a standard network measurement and monitoring toolkit. Second, we attempt to analyze some of the inconsistent flow sizes in bytes and packets for their lifetimes, over the years. We focus on the behaviors of one-way flows and compare that with two-way flows. For example, applications that are intended to produce two-way flows often produce only one-way flows, and the proportion of one-way flows has increased over the last few years.

The rest of the paper is organized as follows. In Section II, we discuss our methodologies for trace data collection, flow table manipulation and we evaluate our traffic meter performance. Our observations of one-way and two-way flows are discussed in Section III. In Section IV, we compare both types of flows by their protocols and host associations. Section V summarizes our conclusions and suggests further work.

## 2. METHODOLOGY

***Auckland* traces**: We used the Internet edge gateway measurement point of The University of Auckland, via the *Auckland II* (2000) and *VIII* (2003) traces from NLANR PMA [6]. In addition, we captured *Auckland* traffic in 2006 using tcpdump [7]. An interesting aspect of the traces is that they include several busy web-servers in the network. That is, the traces consist of hosts that represented many popular activities in New Zealand. For example, in the middle of 2001, our web-servers were rated as No.1 'most popular/visited' in New Zealand (and they are still in the top 100) [3]. Table 1 shows three traces used in our measurement. All observed datasets are one-day (24 hours) in duration. The particular dates were chosen for our observations since they were not greatly different from similar days within the *Auckland* trace sets. The traces tested were all measuring the main link (both inbound and outbound traffic), at a point between the network's boundary routers and its main firewall. Due to New Zealand's slow broadband deployment, in year 2000 most local (home) users were still accessing our networks using dial-up modems and our network had only 6MB/s of international bandwidth. From 2002 onwards, ADSL lines started to take over, and our network increased its peak international bandwidth to 30MB/s in 2003. As of 2006, the peak bandwidth was further increased to 70MB/s. Our campus network consists of some 14,000 hosts, mainly used by staff and students. Students require Internet credits to be able to access outside network resources, charged per MB. Other than the copyright issues for P2P file-transfer applications, students are also discouraged from using them since the cost of data transmissions is non-trivial. The traces for 2000 and 2003 were captured during semester breaks, and for 2006 were captured during the normal semester. Overall, our network mostly contains HTTP web traffic with very few protocols other than TCP, UDP and ICMP.

**Traffic meter**: Figure 2 illustrates the building of a flow table. Our flow manipulation's basic algorithm is similar to, but simpler than the stream-matching in RFC 2722 [8]. For each packet seen, 5-tuple fields are inspected and the flows are immediately manipulated into a hashtable. If a flow does not exist in the hashtable

**Table 1**. Summary of three *Auckland* traces (24 hours)

| Year | IP packets | | | Application Bytes | | | Total Bytes |
|---|---|---|---|---|---|---|---|
| | TCP | UDP | ICMP | WWW HTTPS | DNS Proxy Mail | Other | |
| 2000-Jan-14 | 20.0M | 4.0M | 0.3M | 63% | 15% | 22% | 8.3 GiB |
| 2003-Dec-04 | 133.9M | 13.9M | 1.7M | 78% | 14% | 8% | 63.6 GiB |
| 2006-Jul-27 | 431.1M | 52.4M | 2.4M | 55% | 30% | 15% | 273.9 GiB |

or an incoming packet, then its directions are swapped to be re-checked in the hashtable, effectively building two-way flows. Thus, whether the flows are one-way or two-way, only one entry is made in the hashtable. To distinguish directions of flow's inbound and outbound packets, each table entry contains two pointers. Here we use IP addresses to identify inbound and outbound traffic. For each one-minute interval, all flow entries are checked against the meter's expire timeout. Packets that are still in the process of flow aggregation are temporarily locked and buffered until the checks have finished. We used 64 seconds [14] as our expire timeout, the maximum time that flows exist in the hashtable without further packets being seen. Expired flows are garbage collected and passed on to the analysis process. At the end of reading traces (i.e., 24 hours of trace time), flows that remain in the hashtable are all set to expire and processed.

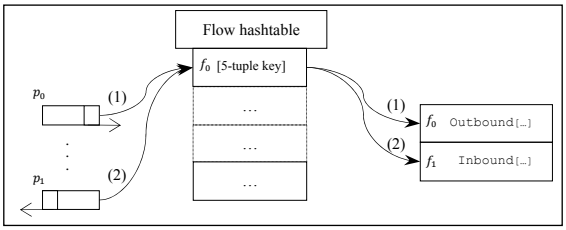



**Figure 2**: An illustration of two-way flow matching. **(1)** outbound packet $p_0$ is considered a flow $f_0$. **(2)** inbound (opposite) packet $p_1$ does not exist in the hashtable, so the direction is *swapped* and matched to the flow $f_0$.

Next, we constructed information from the processed data to perform our experiments. To demonstrate the different behaviors of one-way and two-way flows, and instead of sampling, groups of log-scale bins were created: for instance, two-way flow RTTs and lifetimes. After several experiments in finding appropriate bin sizes, we decided to use 100 bins for 24 hours of flow lifetime ranging from 10us to 100ks and 50 bins for flow RTT ranging from 100us to 100s. To find the average RTT for each two-way TCP flow, we match up the sequence and acknowledgement numbers for the inbound and outbound packets to calculate individual RTT. Then, we aggregate successive RTTs to compute the flow's average RTT. RTT is not calculated for non-TCP flows.

Further, in order to study the behavior of flows, our observations were separated into application types (i.e., well-known ports) and protocols (i.e., TCP and non-TCP). We used known IP addresses and ports to distinguish the groups, so as to estimate their contributions of the total flows, bytes and packets.

**Performance:** We processed the datasets using an object-oriented (Java-based), prototype traffic meter. The authors in NG-MON [15] showed brief performance comparisons of publicly available traffic meters and commented that it is necessary to use the parallel and distributed approach to be able to handle 10Gb/s or more in real-time. However, our traces are from lower-speed links. Thus, we only required a modern PC to process trace files at reasonable speed. We used the *Auckland* 2006 trace to benchmark

**Table 2**. Processing time for average measurements per-interval (60s), *Auckland* 2006

| Process steps | Hashtable manipulation | | Relative Increase/Decrease |
|---|---|---|---|
| | UniDir | OW/TW | |
| Packet capture | 2.8s | 2.8s | NA |
| 1 - Flow manipulation | 1s | 1.8s | Decrease (0.5) |
| 2 - Flow garbage collection | 0.4s | 0.4s | 0% |
| 3 - Flow analysis & output | 3.2s | 0.8s | Increase (4.0) |

using a hardware specification of Intel E6300 with 2GiB of physical memory. For average traffic rates, the traced traffic was 27Mb/s. In busy hours, the traced traffic level was more than 12,000pkt/s and about 65Mb/s. The processing of this 24-hour trace took just over two hours. Also, our implementations were designed to run on the live network. The prototype tests revealed that using the above hardware specifications, we can easily capture and readily process traffic on our gateway access link in real-time.

Table 2 shows three main processing steps of our implemented traffic meter. These steps are executed in series for every one-minute of packet trace time. The comparisons are conducted by measuring the completion time for each step. The memory usage was about 100 to 300MiB. The flow manipulation (step-1) involves aggregating the packets to 5-tuple flows and placing them into the hashtable. The garbage collection (step-2) process involves the expire timeout which removes flow entries to pass onto the flow analysis. Then, these flows are processed (step-3) to produce essential results such as lifetimes, RTTs, host groupings and so on, and plotted. In the traditional approach (UniDir), the steps are followed sequentially, and the memory usage for this approach was slightly less than our approach (OW/TW). In our approach, computation of flow attributes such as RTT is done in step-1.

Even though our approach in the flow direction swapping and rechecking against flow table can be considered as *Big-O constant*, additional variables (and increased memory access) decreased the speed in the flow manipulation stage. However, we observe a considerable performance leap (e.g., 4 times faster) in the flow analysis and output stage. Having just unidirectional flows would require searching the hashtable to find the opposite direction, re-ordering of structures (e.g., re-building hashtable) which takes more time than our implementation. We believe that the relative speed increase in step-3 is significant compared to the decrease in step-1. In other words, our approach hardly processes in step-3 since one-way and two-way flow matching already finds results such as RTTs. Thus, the readily analyzable hashtable structures allowed increased performance in the overall analysis. On average, processing each minute of trace took about 5.8 seconds using our approach compared to 7.4 seconds using the traditional approach.

# 3. FLOW BEHAVIORS

In this section, we split up flows into one-way and two-way flows, and attempt to study their lifetime distributions in terms of flows, bytes and packets. Table 3 shows a summary of our three traces, listing percentages of flows, bytes and packets for one-way (OW) and two-way (TW) flows. Although the total byte and packet contributions of the one-way flows as compared to two-way flows are small (e.g., less than 9%), their contributions to total flows are not. In 2000 and 2006, total flows identified as one-way are 23.4% and 34.6%. Surprisingly, in 2003 more than

**Table 3:** Statistics of one-way and two-way flows

| Year | Flows (%) | | Total Bytes (%) | | Total Packets (%) | |
|---|---|---|---|---|---|---|
| | OW | TW | OW | TW | OW | TW |
| 2000-Jan-14 | 23.4 | 76.6 | 3.2 | 96.8 | 8.4 | 91.6 |
| 2003-Dec-04 | 55.4 | 44.6 | 2.4 | 97.6 | 7.6 | 92.4 |
| 2006-Jul-27 | 34.6 | 65.4 | 1.9 | 98.1 | 2.6 | 97.4 |

| Year | OW Flows (%) | | TW Flows (%) | | OW Bytes (%) | | TW Bytes (%) | |
|---|---|---|---|---|---|---|---|---|
| | TCP | non-TCP | TCP | non-TCP | TCP | non-TCP | TCP | non-TCP |
| 2000-Jan-14 | 25.5 | 74.5 | 85.3 | 14.7 | 12.5 | 87.5 | 96.7 | 3.3 |
| 2003-Dec-04 | 80.9 | 19.1 | 73.3 | 26.7 | 28.1 | 71.9 | 95.2 | 4.8 |
| 2006-Jul-27 | 46.0 | 54.0 | 78.5 | 21.5 | 50.3 | 49.7 | 95.3 | 4.7 |

half of total flows (55.4%) are identified as one-way. Here, we further separate these flows by a protocol (TCP and non-TCP), we observe that the byte and packet contributions of the two-way flows are steady over the years, but we observe inconsistency for the number of one-way flows. Additionally, we observe that the byte contributions of the one-way flows are mostly non-TCP flows (87.5%) in 2000, but this has been reduced (49.7%) in 2006. Thus, we question why such contributions exist particularly when more than 90% of traffic volumes are TCP throughout the years. Here, we observe several behavior differences between one-way and two-way flows.

## 3.1 One-way Flows

Several cases of the one-way flows could be observed. It is worthwhile to note that, for every passive measurement and regardless of traffic meter types, it is inevitable to encounter traffic meter 'start-up' and 'finish-off' effects. In 'start-up' effects, the traffic meter may count an intended two-way flow as one-way because it managed to capture just the very last packet of the two-way flow. In 'finish-off' effects, the traffic meter counts intended two-way flows as one-way because it captured only the very first packet of the two-way flow and then finished reading the trace. Also, the *Auckland* network contains three main Internet Proxies, which service all campus users (excluding the staff) who access the Internet. Students can access outside network resources when their Internet credits are available, but when they reach zero while still accessing resources, the proxy will immediately disconnect the user, causing outbound hosts (and the proxy itself) to fail the connections, possibly giving rise to one-way flows. In rare cases, a network may experience an outage causing many hosts to produce one-way flows. While it is possible to find many cases of one-way flows, we have limited the cases into to three main types. Firstly, many applications use either TCP or UDP to communicate between pairs of hosts. From this aspect, all applications use request/response packets to exchange messages, building up the two-way flow (Figure 3(a)). However, failure to acknowledge messages would prompt re-transmissions or in worst cases, disconnections. For example, if a host or router strikes a busy CPU process, it would drop packets. Also a host may decide not to respond to transmissions when it encounters hard shutdown or network cable being unplugged (Figure 3(b)). Secondly, there are applications that mostly send or receive packets in one direction only. Few or no packets are produced from the other side. DNS or ICMP packets often follow this type of behavior when their destination host does not respond (Figure 3(c)). Lastly, there are packets passing the link (Figure 3(d)) that are blocked by the gateway firewall or by the hosts' software firewalls (e.g., port scans).

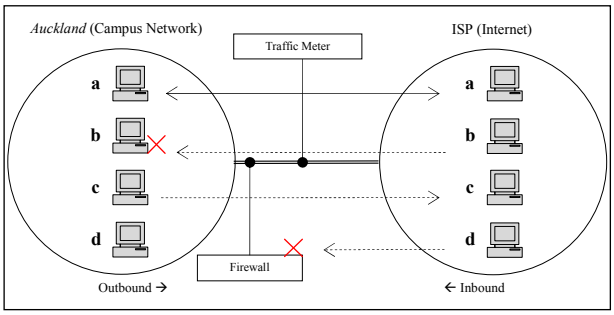
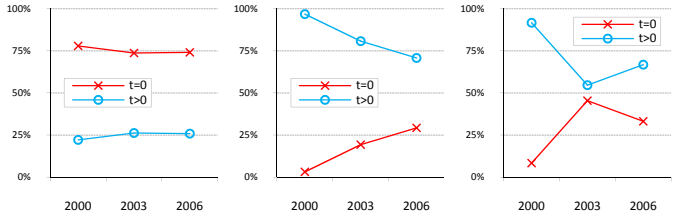



**Figure 3**: A diagram of raw packet transmission forms, showing two-way (a) and one-way flows (b,c,d)

We separated out the one-way flows observed in Table 3 into two different lifetimes as shown in Figure 4. For flows with a single-packet ($t = 0$), we observe relatively steady proportions of flows, and increases in their sizes in bytes and packets over the years. That is, the rest of the flow ($t > 0$) contributions to the bytes and packets have decreased over the years. Our first notion was that steady flows over the last seven years could imply that there are certain numbers of hosts producing one-way flows. However, the increases of these bytes and packets could mean that there are other types of traffic emerging as well.



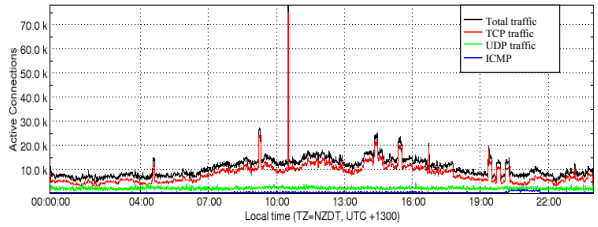

(a) flows (b) bytes (c) packets

**Figure 4**: Three plots of one-way flow percentages in two groups by their lifetimes ($t$): Single-packet flows ($t = 0$) and Multi-packet flows ($t > 0$)

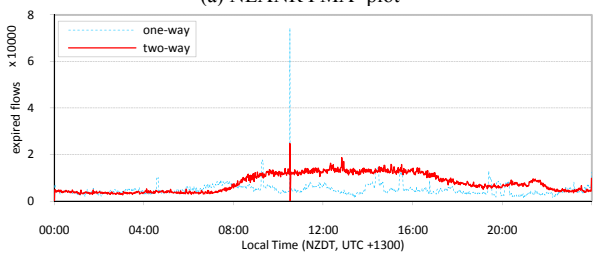
Figure 5(a) shows a traffic plot for an active connection from NLANR PMA 2003, with a high spike of TCP flows observed at about 10:30 NZDT. However, we found that the vertical axis was not actually 'active connection' (as indicated on the NLANR plot), but merely watching (any) flows. Here, the same dataset was plotted using our approach, its expired flow counts of one-way and two-way flows are shown in Figure 5(b). These high spikes lasted for 9 minutes and were caused by both inbound malicious flows and our outbound proxy hosts being shut down. The other smaller spikes were also identified as one-way flows. That is, these spikes caused more than 50% of total flows. Further, about four times as many one-way flows appeared in the inbound traffic as compared to outbound over the years.

**Single-packet flow:** Considering the high proportions of dragonfly flows appearing in the CDFs of Figure 1(a), we found that between 74% and 78% of one-way flows contained exactly a single packet, contributing few bytes and packets overall. Most of the single-packet TCP flows have consistent average size (e.g., 70 bytes) over the years, indicating that the packets have little or no payload. We also found that many of these flows contained a SYN flag, they were 'probes' that failed to initiate connections. There were legitimate single-packet flows using DNS and ICMP protocols, contributing less than 10%.

However, as years pass, we observe more spikes appearing on




(a) NLANR PMA[2] plot




(b) one-way/two-way plot

**Figure 5**: Flows vs time of day, *Auckland* 2003-Dec-04

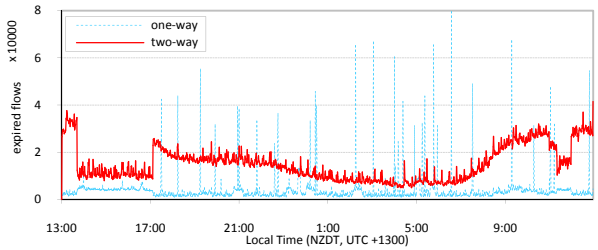



**Figure 6**: Flows vs time of day (one-way/two-way), *Auckland* 2006-Jul-27

our network. Figure 6 shows that an enormous number of single-packet UDP flows appeared in 2006, contributing more than 70% of the total one-way flows. This behavior was observed in [11] as plagues of dragonflies, a sudden rise of short-lived flows. Closer analysis has revealed that all of the high spikes were caused by attack flows, widely known malicious packets [5, 9] targeting UDP port 1026-1029, 135, 445, 4899 and so on (e.g., enumerating various hosts inside our network). These flows size varied between 200 to 1000 bytes. Also, a few attack flows that were using well-known protocols (e.g., port 25, 53 and 80) were discovered.

**Multi-packet flow:** We regard the remaining one-way flows as multi-packet flows, with two typical behaviors.

First, we observed the significance of the flows with two or three packets in several types of application. Their byte contributions are less than 2%, but their lifetimes are extremely short, ranging from 100us to 10ms, most likely bursts of multiple request packets. Because only small flows existed, there were variations in the flow sizes, averaging between 100 and 300 bytes. Similar to the single-packet flow, most TCP flows contained SYN flags. However, there were not many malicious flows, and their contributions varied between the years. Figure 7(a,b) shows the one-way average flow sizes in bytes and packets vs lifetime distributions in 2000 and 2003. 2006 distributions were similar to 2003, and are therefore not shown. Small step rises or drops appearing for short lifetimes (e.g., less than 1s) are mainly caused by dominating applications such as DNS, ICMP and web retries. We observe a steady average flow sizes lasting somewhere from 100us until 40 to 50s for all years, most likely reflecting re-transmissions. In addition, we observe sudden steps between 3s and 10s in the CDF plot shown in Figure 7(c), contributing about 10% to the total one-way flows. Flows with these lifetimes were identified as web flows, with retries being ended by the users or host application (e.g., by refreshing the web pages assigns new source ports). Furthermore, roughly 80% of one-way mice flows lasted less than 2s, and the remaining 10% lasted 10s onwards. However, there were virtually no elephant flows (refer Section IV for their protocol flow size distributions).

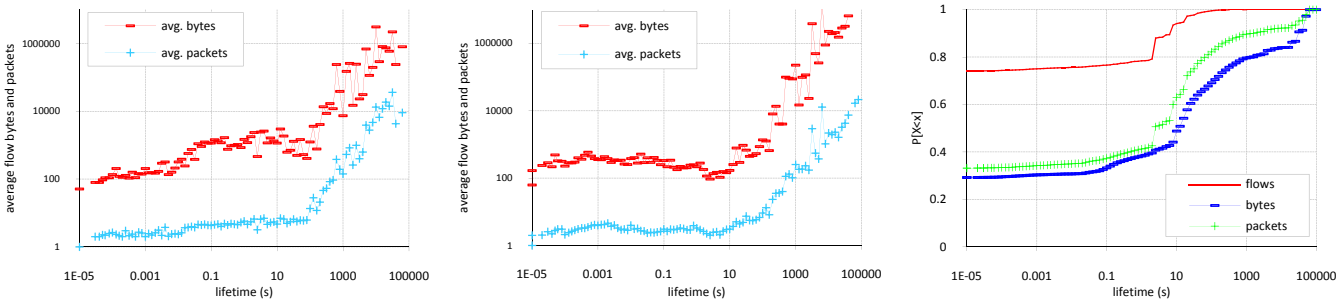
Second, we observe a high rise of averages after 40 to 100s, until the end of 24 hours (Figure 7(a,b)). This behavior appeared in all of our traces. We analyzed these sharp rise behaviors by separating one-way flows into a number of applications. In this, DNS and ICMP mainly caused the sharp rise. This is because DNS servers mostly use UDP packets and often servers may not respond to the request packets. The source and destination port number stays surprisingly constant (e.g., receiving on port 53 and sending on port 32770). This effect reduces the 5-tuple into a 3-tuple flow by excluding the source and destination port, causing flows for a given pair host entities to be accumulated over time resulting in longer lifetimes. Thus, bigger flow sizes in bytes and packets are observed.

Figure 8 shows per-flow average packet inter-arrival times in 2003 and 2006. Very small packet gaps are observed for flows that lasted from 1ms to 100ms, averaging about 0.4ms to 40ms of packet inter-arrival time (again, most likely a burst of request packets). Slightly higher packet gaps are observed from 100ms to 1s, averaging from 100ms to 300ms of inter-arrival time. Further, flows lasting about 10s have an average 2s to 4s of inter-arrival gaps. As flow lifetime increases, we observe a clear increase of inter-arrival time for each application. These plots represent the behaviors of re-transmissions according to the transport-layer algorithms (e.g., exponential back-off) or specific protocol behaviors. Also, we observe more diverse inter-arrival times in 2006.

In order to see the effect of longer flow table expiry times, we conducted an experiment by increasing the flow expire timeout. We set timeout to 128s and 256s to see the changes in flows and inter-arrival times. In this, the one-way flows lasting up to about 40s are similar for all timeouts. However, after 40s, we notice that the longer-running flows of DNS, ICMP and other applications produced longer inter-arrival times (plots not shown). Commonly, this is because there are fewer flows observed for DNS and ICMP with longer timeout. As discussed previously, a longer timeout tends to join smaller flows into larger flows. This is one of the main reasons we see a sharp rise of average bytes and packets in Figure 7(a,b).

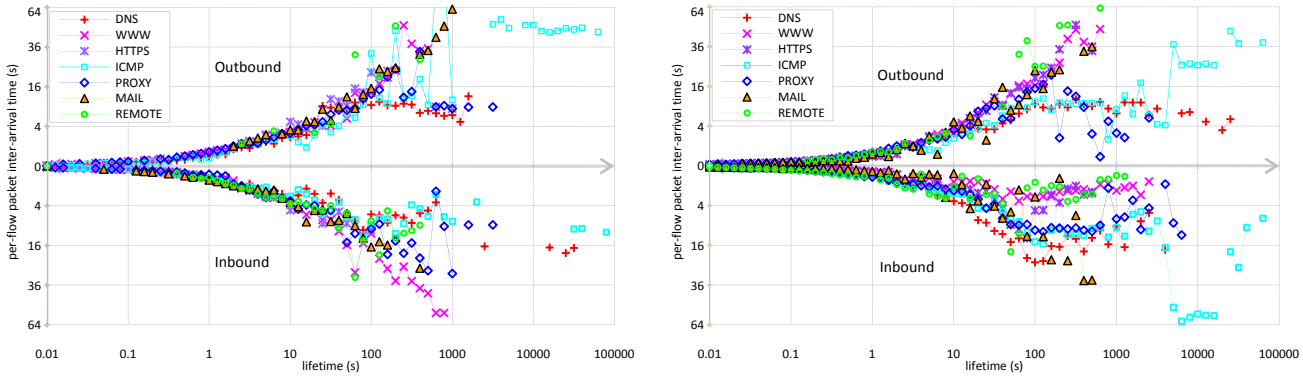
In summary, as of 2006, more than 74% of single-packet flows are attack flows and few are legitimate flows. The rest of the one-way flows contain multiple packets, which lasted as briefly as 100us and as long as 64s or more. Application observations of packet inter-arrival show that many one-way flows that lasted until about 40s are re-transmission flows, and we observe more diverse behaviors of inter-arrivals over the years. In terms of flow contributions, mice flows contribute a large proportion of total one-way flows (e.g., up to 80% flows lasted less than 2s). The longer lasting flows (more than 40s) produce sharp rises in flow size due to flows being joined, especially for DNS and ICMP.

[2] Detailed images available from http://pma.nlanr.net/Special/auck8/20031204.html




(a) 2000 (b) 2003 (c) 2006

**Figure 7:** (a,b) Average one-way flow sizes in bytes and packets vs lifetime distribution, (c) CDF plot of one-way flows, packets and bytes




(a) 2003 (b) 2006

**Figure 8:** One-way per-flow packet inter-arrival time vs lifetime distributions

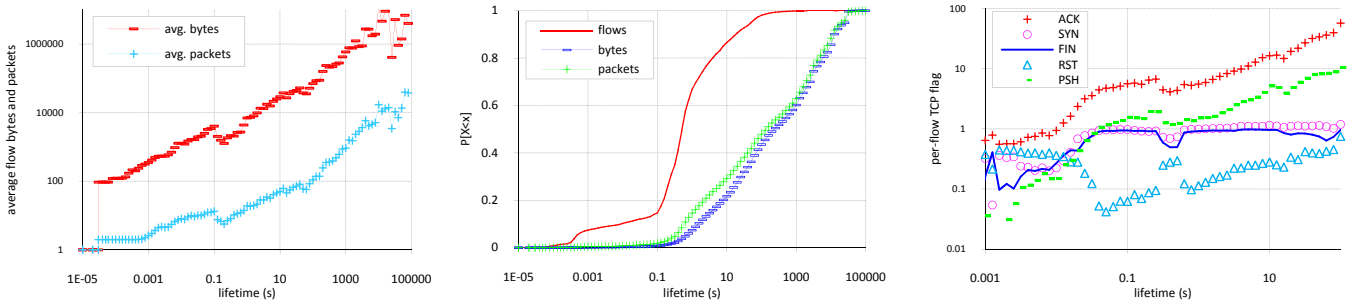
## 3.2 Two-way Flows

As noted earlier, one-way flows become two-way flows when the traffic meter observes packets for them in both directions. In terms of contributions over the years, two-way flows carry more than 95% of the total bytes. We first analyze single-packet exchange and multi-packet exchange flows. We then observe TCP flags that are presented in each two-way flow for their lifetimes. Further, we observe RTTs of the two-way flows.

**Single-packet exchange flow:** From our analysis of two-way flow lifetimes, the very short-lived flows are all characterized as single-packet exchange flows, identified as 2-packet back-to-back flow in [24] (sending one packet in each direction). These flows often contained SYN/RST flags and contributed less than 10% of total two-way flows over the years. Figure 9(a) shows lifetime distributions for two-way flow sizes in 2003. These single-packet exchange flows lasted from 0.5ms to 10ms in 2000, 50us to 800us in 2003, and 20us to 500us in 2006. Indeed, these flows have the shortest RTT intuitively. The average size of these flows was 120 bytes and those that did not contain RST flags were mostly DNS traffic.

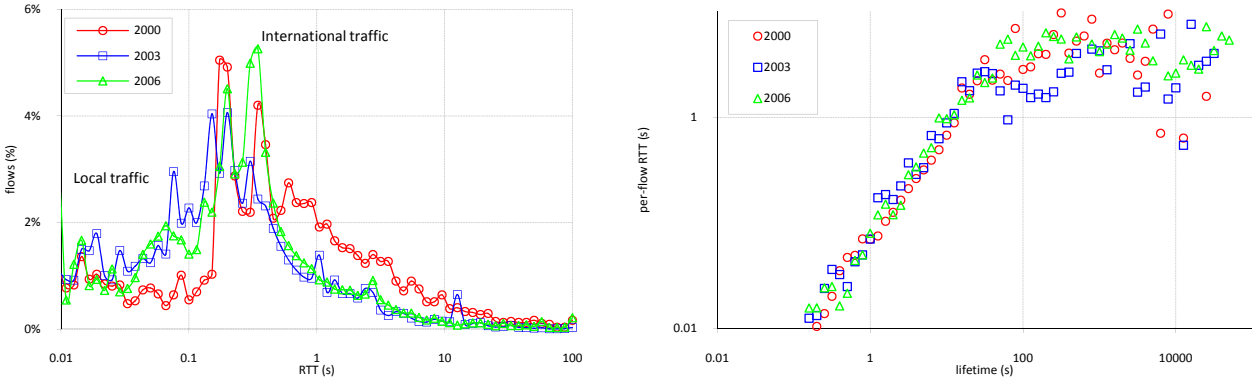
**Multi-packet exchange flow:** We regard the two-way flows containing more than a single packet in either direction as multi-packet exchange flows. These flows contribute most to the total bytes: the data transmission. Looking again at Figure 9(a), the longer lasting flows (e.g., from 10000s) are in distorted lines, mainly due to fewer log bins and only a few flows observed. The distortions appeared less in 2006, reflecting the increasing number of tortoise flows. Also, we observe increases in flow sizes and decreases in their lifetimes over the years, depicting higher network speed. A slight decrease in average bytes (3kB) and packets (12pkts) at about 0.3s to 0.6s is most likely due to web traffic dominating the contributions (port 80) and the inclusion of the connection reset (RST). Compared to one-way flows, two-way flow size distributions are approximately power law (linear log-log distribution). Considering the two-way flow lifetime as a Zipf's law rank (taking account of largest average bytes as 1st rank), our two-way flow datasets show a straight line. Recalling from a previous study conducted in [23], their rank plot (in Fig.1) was an approximate straight line with a rather heavier down-slope in the lower rank (mice). This could well be because [23] included one-way flows (i.e., attack flows) in their dataset. Furthermore, unlike one-way behavior, we observe fewer mice flows showing that the proportion of two-way flows is significantly less than the proportion of one-way flows. (Figure 9(b)). On average over the years, less than 20% of two-way mice flows lasted less than 1s as compared to 70% of one-way mice flows. Also, about 60% of two-way flows lasted between 0.1 to 10s, and less than 0.1% lasted more than 1000s (e.g., elephant flows).

**TCP flags:** Figure 9(c) shows per-flow average TCP flags (i.e., flags presented inside each two-way flow) in 2000. Other years were not shown because they are similar. For each two-way flow, we count the number of TCP flags. Thus, longer-lived flows are likely to contain more ACK flags since most of the packets acknowledge a previous packet's sequence number. In this, ACK/PSH flags are frequent (90%), presenting similar distributions to the flow sizes in bytes and packets shown in Figure 9(a). Further, we observe SYN and FIN flags appearing at least once for most of the flows, depicting the correct TCP transmission procedure. RST flags occur rarely (1.5%), but appear mostly at the lower lifetimes (also shown in Figure 9(a)),




(a) 2003: average flow sizes in bytes and packets (b) 2006: CDF plot of flows, bytes and packets (c) 2000: per-flow TCP flags

**Figure 9:** Various two-way flow lifetime distributions




(a) two-way TCP Round Trip Times (b) per-flow average RTT vs lifetime distribution

**Figure 10**: RTT distributions of two-way TCP flows

suggesting that most short-lived flows containing ACK/RST flags are probably reset by their destination host (e.g., signaling errors occurred by TCP). For one-way TCP flows (not shown), SYN/ACK flags mostly dominated the plots and they were much less regular, probably due to the non-responding hosts.

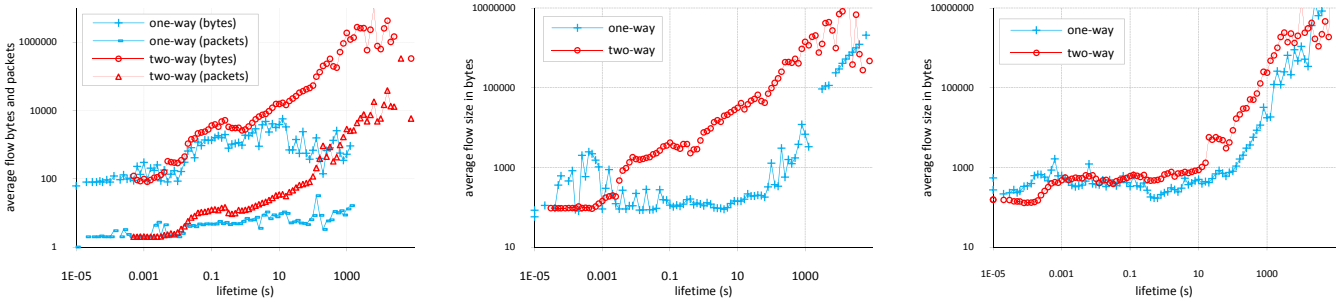
**Two-way flow RTTs**: Figure 10(a) shows the distribution of two-way TCP flow RTTs. Overall, the shape of the plot depicts similar views to those studied at another New Zealand edge network, University of Waikato [21]. Their studies show that the low RTT depicts the main outbound servers such as DNS, Proxy and Mail, and higher RTT reveals the Pacific Ocean hops. Due to the similar network setup at Waikato, our university observations are also similar. Thus, small proportions of flows have very short and very long RTTs. In other words, the majority of RTT (more than 70%) for flows were between 10ms and 1s. Evidently, we observe shifting behaviors of RTTs from 2000 to 2006, giving rise to the decreases in RTTs. For international traffic, we observe that many flows had long RTTs in 2000 (e.g., ≥ 300ms), while most flows in 2003 and 2006 had shorter RTTs. As mentioned previously, our network had a limited international bandwidth (especially in 2000) with ATM 'traffic shaping' (*Auckland* case studied in [20]). The improvement of network devices and increased bandwidth in our own network reduced the overall RTTs.

Figure 10(b) shows the log-log scaled average per-flow RTT vs lifetime distributions. That is, given the flow lifetimes, we observe their average RTT. This produced similar plots for all traces, and we observe a linear plot for lifetimes ranging from 10ms to 20s. That is, both flow lifetimes and their RTTs increase/decrease steadily. We think this effect presumably arises because flows with very short lifetimes (dragonflies) have shorter RTT (e.g., DNS, RST flag flows), and with longer lifetimes (tortoises) have longer RTT (e.g., ICMP, infrequent packet exchange). We also observe that flows lasting more than 20s produce RTTs between 1s and 100s. Further, we observe some variations for RTT distributions of tortoise flows mainly due to the fact that there are fewer flows observed. For instance, we observe a few tortoise flows have small RTTs in 2000 as they were the only flows existing for their lifetimes (a file transfer). We also observe a few elephant flows having long RTTs in 2003 (infrequent packet transmission).

To summarize, volumes of two-way flows contributed more than 90% throughout the years reflecting (mostly) the normal end-to-end communications. Most of the single-packet exchange flows lasted for very short times, reflecting their short RTT, containing mostly SYN/RST flags, and with decreasing lifetimes over the years. The majority of RTTs of flows are clustered between 10ms and 100ms (local traffic), and between 100m and 1s (international traffic). Also, they seemed to decrease as increased network capacity become available. Unlike one-way flows, we observe only small proportions of dragonfly flows. In addition, flows lasting from 10ms to 20s have their average RTTs following power laws (e.g., one tenth of their lifetimes). Flows lasting more than 20s do not linearly increase RTTs, but are steady between 1s and 100s over the years.

## 4. PROTOCOL/HOST FLOW ANALYSIS

So far, we have observed several behaviors of one-way and two-way flows; we think it is interesting to identify which protocols or hosts influence the overall shapes of the lifetime plots. Thus, we further distinguish one-way and two-way flow sizes by




a) 2000: two-way TCP flow bytes and packets (b) 2003: one-way and two-way TCP flows (c) 2006: one-way and two-way non-TCP flows

**Figure 11:** Three plots showing flow sizes vs their lifetime

their protocols and hosts. We first separate flows into TCP vs non-TCP. We then observe how one-way and two-way flows are related to each other and with their individual hosts.

## 4.1 Protocol Distributions and Correlations

In this section, we observe both one-way and two-way flow size distributions in terms of their protocols. To observe whether these flow lifetimes are similar, we compute a correlation coefficient (CC) $\rho(X, Y)$ between two variables $X$ and $Y$ as our series of bins (ranging from 1ms to 10000s) for bytes vs packets (Figure 11(a)), one-way vs two-way TCP flow sizes (Figure 11(b)), and non-TCP (Figure11(c)). CC for other ranges (i.e., lifetimes below 1ms or above 1000s) were not computed since fewer flows existed. Our datasets over the years showed some evidence of correlations between one-way and two-way flow sizes.

**Flow size:** we found that whether flows are TCP (Figure 11(a)) or non-TCP (not shown), sizes in packets and bytes are highly correlated (i.e., CC: 0.99). Rather obviously, this reflects the appropriate transmissions of typical packet size (e.g., 60 bytes ACK, 1500 bytes data).

**TCP flows**: Figure 11(b) compares one-way and two-way TCP flow sizes in 2003. For two-way flows, these distributions have remained constant. Their size in bytes and packets has increased slightly over the years, depicting the increases in transmission rates for their lifetimes. For one-way flows, we observe a few transmission failed flows that lasted from 250us to 1ms. Flows lasting up to 100s have relatively a few bytes due to the retransmission (of ACK) packets. We initially expected that the one-way and two-way flow sizes are uncorrelated since one-way flows are most likely attempt-failed connections producing different lifetime distributions. However, from 2003 onwards, we also observe that there exist a longer-lived (lasting from 100s to 10000s) one-way flows whose sizes are enormous and actually followed a very similar distribution to two-way flow sizes (e.g., result in coefficient value was 0.84). This behavior showed that such long-lived one-way flows existed as a part of persistent traffic as well. For instance, Mail, Proxy, DNS and malicious caused this persistency: some delivered packets at a slow rate, but others delivered packets at a high rate.

**Non-TCP flows**: Figure 11(c) shows similar comparisons of one-way and two-way non-TCP flow sizes in 2006. The immediate difference between the TCP and non-TCP flows is that for non-TCP flows, one-way and two-way flow lifetime distributions are more highly correlated than they are for TCP flows. That is, the distributions of one-way and two-way flow sizes look very similar. Two-way flow sizes were slightly bigger than one-way flows, most likely reflecting that the two-way flows are carrying bigger packets and transmitting at a faster rate. As discussed in the previous section (Figure 7(a,b)), we observed a high rise of flow size in bytes and packets from about 40 to 50s, reflecting the UDP dominations from DNS and ICMP.

## 4.2 Overall Host Traffic Analysis

This section discusses the host behaviors in relation to one-way and two-way flows. That is, we attempt to observe host distributions of one-way or two-way flows. To distinguish how hosts behave in terms of percentages of two-way flows, we find the 'two-way percentages', by computing the number of two-way flows over the total flows for each IP host, and observe their distributions, shown in the CDF plot in Figure 12. In other words, if a host contains only two-way flows, then it produces 100% (and 0% if a host contains only one-way flows).

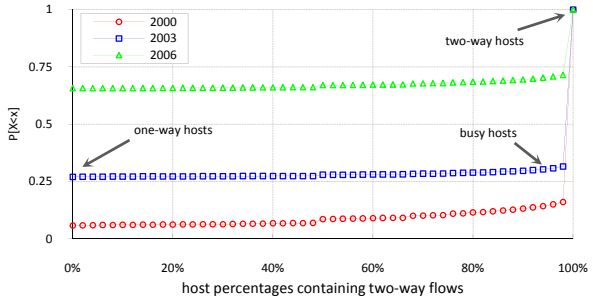



**Figure 12**: One-day CDF plot of host distributions: each host is represented in percentages by the number of two-way over the total flows

As expected, nearly all hosts have either 100% or 0% of two-way flows: hosts contain either all pair communication (two-way flows) or none (one-way flows). That is, the majority of hosts involved with one-way flows are the single-packet flows. Further, about 90% to 99% of two-way hosts are mainly busy hosts including DNS, Mail, proxies, producing occasional one-way flows (e.g., inbound hosts unresponsive). For instance, we observe our outbound DNS and Proxy hosts containing small proportions of one-way flows. Over the last seven years, the number of hosts involved with one-way flows has increased dramatically. In 2006, more than 65% of total hosts contained one-way flows only, as compared to a mere 5% in 2000. Closer inspections revealed that these hosts are mostly from outside our network, sending out malicious packets to various hosts inside our network.

In summary, one-way TCP flows are mostly attempt-failed connections (and malicious). Also, persistent flows are observed, changing the lifetimes of one-way flows to last longer,

and produce bigger flow sizes. Two-way TCP flows on the other hand, show close similarities, indicating that their flow sizes are consistent over the years, reflecting appropriate data transmissions. In addition, one-way and two-way non-TCP flows are highly correlated for all years, producing very similar flow size vs lifetime plots. Generally, we find that overall flow sizes and lifetimes are highly independent of one another, dependent on which applications are present and how much they contribute to the overall traffic. Throughout the years, we observe an increasing number of hosts that produce one-way flows only, mostly malicious flows. In this, we found that hosts could be categorized by their flows. First, hosts that only produce one-way flows appear to be actively using systematic, automated tools to send out malicious packets to various networks. Second, hosts that produce two-way flows only mostly indicate appropriate data transmission. Third, hosts that produce many two-way flows and some occasional one-way flows are most likely busy hosts, such as DNS, Proxy hosts and web servers.

# 5. CONCLUSION

In this paper, we have demonstrated our two initial difficulties. First, we observe changing flow lifetimes over the years, leading us to capture and aggregate packets into both unidirectional and bidirectional flows on the fly. We implemented this by building a traffic meter to gather into one-way and two-ways flows, which effectively reduced the flow table to a half of its traditional size. Second, that separation easily allowed us to examine flows in a more detailed way. We observed and discussed one-way and two-way flow sizes in bytes and packets. Using empirical datasets over the past seven years showed changes in one-way and two-way flow behaviors.

Our study unfortunately left some questions unaddressed, such as how some one-way TCP flows are persistently long-lived, yet producing large flow sizes, or exactly what is causing two-way flows to contain RST flags for certain lifetimes. We consider that such changes are caused by the changes of the application and protocol mix through the years. In addition, we emphasize the issues of the application domination and the inclusion of malicious flows in the lifetime analysis. Such issues can change the overall shape of the plots. The process of analyzing individual flows over the years not only takes time, but also requires knowledge of past network setups, host addresses, application trends and so on, to match up and compare the results. Our work mainly discussed the understandings of one-way and two-way flows from the *Auckland* traces, which logically lead to further studies. We think there are still many variations of these flows in terms of their sizes and lifetimes. Additionally, we would like to study traces other than *Auckland* to examine different behaviors of one-way and two-way flows that may occur.

# 6. ACKOWLEDGEMENTS

The authors are thankful to NLANR PMA [6] and Perry Lorier from The University of Waikato [21] for helping to process the traces. The authors also wish to thank the anonymous reviewers for their valuable suggestions for the improvement of the paper.

# 7. REFERENCES

[1] "Cisco IOS NetFlow," http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

[2] "CoralReef," http://www.caida.org/tools/measurement/coralreef/.

[3] "hitwise.co.nz," http://www.hitwise.com/news/nz200503.html.

[4] "IP Flow Information Export (ipfix) Charter," http://www.ietf.org/html.charters/ipfix-charter.html.

[5] "Link Logger - Common Scans," http://www.linklogger.com/commonscans.htm.

[6] "NLANR PMA - Passive Measurement and Analysis," http://pma.nlanr.net/.

[7] "Tcpdump," http://www.tcpdump.org.

[8] "Traffic Flow Measurement: Architecture - RFC 2722," http://www.ietf.org/rfc/rfc2722.txt.

[9] "Viruslist.com," http://www.viruslist.com/en/index.html.

[10] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement* San Francisco, California, USA: ACM Press, 2001.

[11] N. Brownlee, "Some Observations of Internet Stream Lifetimes," *PAM 2005,* pp. 265-277, 2005.

[12] N. Brownlee and K.C. Claffy, "Understanding Internet traffic streams: dragonflies and tortoises," *Communications Magazine, IEEE,* vol. 40, pp. 110-117, 2002.

[13] N. Brownlee and M. Murray, "Streams, Flows and Torrents," *PAM workshop 2001,* 2001.

[14] K.C. Claffy, H.W. Braun, and G.C. Polyzos, "A parameterizable methodology for Internet traffic flow profiling," *Selected Areas in Communications, IEEE Journal on,* vol. 13, pp. 1481-1494, 1995.

[15] S.H. Han, M.S. Kim, H.T. Ju, and J.W.K. Hong, "The Architecture of NG-MON: A Passive Network Monitoring System for High-Speed IP Networks," *Proceeding of 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management,* pp. 16-17, 2002.

[16] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications* Philadelphia, Pennsylvania, USA: ACM Press, 2005.

[17] C. Kreibich, A. Warfield, J. Crowcroft, S. Hand, and I. Pratt, "Using Packet Symmetry to Curtail Malicious Traffic," *Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets-IV),* 2005.

[18] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement* Taormina, Sicily, Italy: ACM Press, 2004.

[19] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E.D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proceedings of the joint international conference on Measurement and modeling of computer systems* New York, NY, USA: ACM Press, 2004.

[20] K. Mochalski, J. Micheel, and S. Donnelly, "Packet Delay and Loss at the Auckland Internet Access Path," *PAM 2002,* pp. 46-55, 2002.

[21] R. Nelson, D. Lawson, and P. Lorier, "Analysis of long duration traces," *SIGCOMM Comput. Commun. Rev.,* vol. 35, pp. 45-52, 2005.

[22] K. Papagiannaki, N. Taft, Z. Zhi-Li, and C. Diot, "Long-term forecasting of Internet backbone traffic," *Neural Networks, IEEE Transactions on,* vol. 16, pp. 1110-1124, 2005.

[23] J. Wallerich, H. Dreger, A. Feldmann, B. Krishnamurthy, and W. Willinger, "A methodology for studying persistency aspects of internet flows," *SIGCOMM Comput. Commun. Rev.,* vol. 35, pp. 23-36, 2005.

[24] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the characteristics and origins of internet flow rates," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications* Pittsburgh, Pennsylvania, USA: ACM Press, 2002.