Measuring and Characterising Inbound Sessions in Residential DSL Traffic

Shane Alcock
University of Waikato
Hamilton, New Zealand
salcock@cs.waikato.ac.nz

Richard Nelson
University of Waikato
Hamilton, New Zealand
richardn@waikato.ac.nz

***Abstract*—It has been assumed that the need for successful NAT traversal discourages residential DSL users from running services or applications that require them to accept connections from remote hosts. However, there are now numerous approaches for NAT traversal but there have been no studies measuring the prevalence of DSL users accepting inbound sessions. This paper presents an analysis of TCP and UDP flows initiated by remote clients to a group of residential DSL users, using packet traces captured from a New Zealand ISP between 2009 and 2011. Our analysis reveals that over half of all measured users accepted at least one inbound TCP or UDP session. There was no dominant port being used to accept sessions and port usage was spread over thousands of different ports. Skype and BitTorrent were the most popular application protocols observed and had increased in popularity over time. We also report on the use of dual SYNs to perform TCP NAT traversal in our data.**

I. Introduction

The Internet has become an increasingly integral component of our daily lives and, as a result, residential Internet users now contribute a significant portion of Internet traffic. However, the behaviour of residential users and the applications that they use and are not particularly well understood. There have been some broad studies in this area, such as [1] [2] [3] and [4], but many aspects of residential Internet usage have been barely examined by researchers.

One such area is the services and applications run by residential DSL users that require remote hosts to connect to the residential host. It is commonly assumed that few residential DSL users utilise such applications because NAT traversal is needed for them to operate effectively. This may have been true in the past, but the development of NAT traversal techniques and the proliferation of peer-to-peer applications would suggest that this assumption no longer applies. The objective of this work is to determine whether this is the case and to analyse the behaviour of users and applications that accept inbound sessions from remote hosts so that they can be better understood by both researchers and ISPs.

To this end, this paper presents a study of the incoming connections received and accepted by residential DSL users, measured using packet traces captured from a New Zealand ISP between 2009 and 2011. We examined the proportion of users that accepted inbound connections to determine whether this behaviour was common. We investigated the number of NAT holes created by each user and the amount of traffic passing through each NAT hole. We also analysed the range of ports and protocols that were used by the inbound sessions to discover which applications contribute the most inbound sessions.

Our major findings include:

- Over half of all active residential DSL users engaged in at least one TCP or UDP session that was initiated by a remote host. The NAT holes created, sessions per NAT hole and traffic per NAT hole all conform to a power law distribution.
- Port utilisation was spread over a large range of port numbers, with the most common port being used by less than 5% of users.
- Over 70 unique application protocols were identified as involved in inbound sessions in the datasets. The most widely used was Skype, which became increasingly popular over the three years measured. BitTorrent, RDP and STUN were also growing in popularity, while Gnutella, MSNC and SSH declined.
- 12-18% of the TCP NAT holes observed in each dataset resulted from the use of a dual SYN TCP NAT traversal technique. These were primarily due to the MSNC and Manolito applications.

Aside from providing insight into residential DSL user behaviour, these observations are particularly useful for ISPs considering the implementation of IP address sharing schemes such as Large Scale NAT (LSNAT) [5]. As noted in [6], such approaches may greatly disadvantage users that expect to receive incoming connections from remote hosts but, until now, the usage of such applications by residential users had not been accurately measured and documented.

II. Background

Figure 1 depicts a traditional residential DSL user network in New Zealand, with the DSL router acting as the gateway to the Internet. The user's devices are located on a private network using RFC 1918 addresses [7]. The DSL router connects to the ISP, which assigns a single public IP address to the external interface on the router. In many cases, this address is assigned dynamically from an address pool and addresses can be reused as customers disconnect from the ISP network.

The router uses Network Address Translation (NAT) [8] to facilitate communication between the private network and the

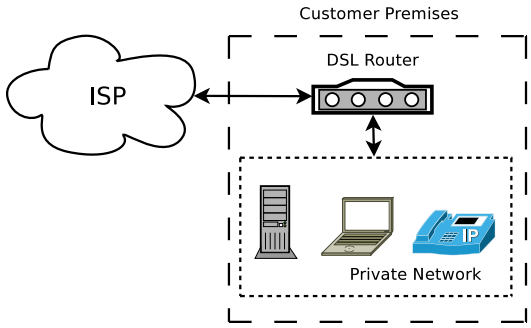


Fig. 1. A typical New Zealand residential DSL user network. The NAT on the DSL router prevents inbound connections from reaching hosts on the private network, unless circumvented by port forwarding or NAT traversal.

Internet. By default, the router only forwards packets for flows that are present in its session table and sessions can only be created by hosts within the private network. As a result, end-to-end connectivity is broken and remote hosts cannot initiate connections to any of the hosts in the user network. For users that only utilise applications where they act as the client, such as HTTP or email, this is sufficient. They can still connect to the servers that they need to and have the additional security of being unreachable by those seeking to exploit their machines through insecure or unintentional listening ports.

This network configuration is not ideal for users that want to participate in peer-to-peer exchanges or host services of their own, such as game servers. As a result, it is assumed that most residential DSL users do not run servers or accept connections from remote hosts. However, the problem posed by the NAT can now be circumvented in one of several ways.

First, the service may be run on the same device as the NAT, meaning that incoming connections can be accepted without difficulty. For instance, any applications running on a PC with a PCI DSL modem card will not be behind NAT and thus capable of end-to-end connections. Second, the user may manually configure their DSL router to forward all inbound traffic on specific ports to the desired device within the private network. This process is commonly referred to as "port forwarding". Manual port forwarding is a non-trivial task for most residential users but the UPnP [9] protocol allows port forwards to be configured on a DSL router by the private host, removing the need for manual configuration. However, both the application and the router must support compatible versions of UPnP for this to be successful.

Furthermore, application level gateways (ALGs) that run on the residential router can enable NAT traversal for specific application protocols. They do not require any support from the application itself, as long as the developer has followed the protocol standard. ALGs examine and adjust the packet payload for the supported application protocol to enable packets to pass in and out of the private network successfully. FTP, for example, is commonly supported through an ALG.

Finally, NAT traversal protocols such as STUN [10], TURN [11] and ICE [12] can be used to enable private hosts to receive inbound sessions, typically via a relay with a public IP address.

TABLE I
TRACE SETS USED IN THIS STUDY (AFTER FILTERING)

Name	2009	2010	2011
Start Date	2009/01/07	2010/01/07	2011/01/06
Duration	7 days	8 days	8 days
Total Traffic	3,552 GB	5,366 GB	6,949 GB
Total Flows	225 M	365 M	385 M
Active Local IPs	4018	4693	4424

TABLE II
INBOUND SESSIONS ACCEPTED BY DSL USERS

Dataset	2009	2010	2011
Users accepting TCP only	14.56%	7.54%	6.42%
Users accepting UDP only	17.37%	19.43%	18.76%
Users accepting both	23.20%	25.27%	27.42%
Total accepting users	55.13%	52.25%	52.60%
Mean NAT holes per user	5.64	23.45	8.81
Mean sessions per hole	2644.84	916.98	2265.34
Mean traffic per hole	41.08 MB	12.34 MB	42.75 MB

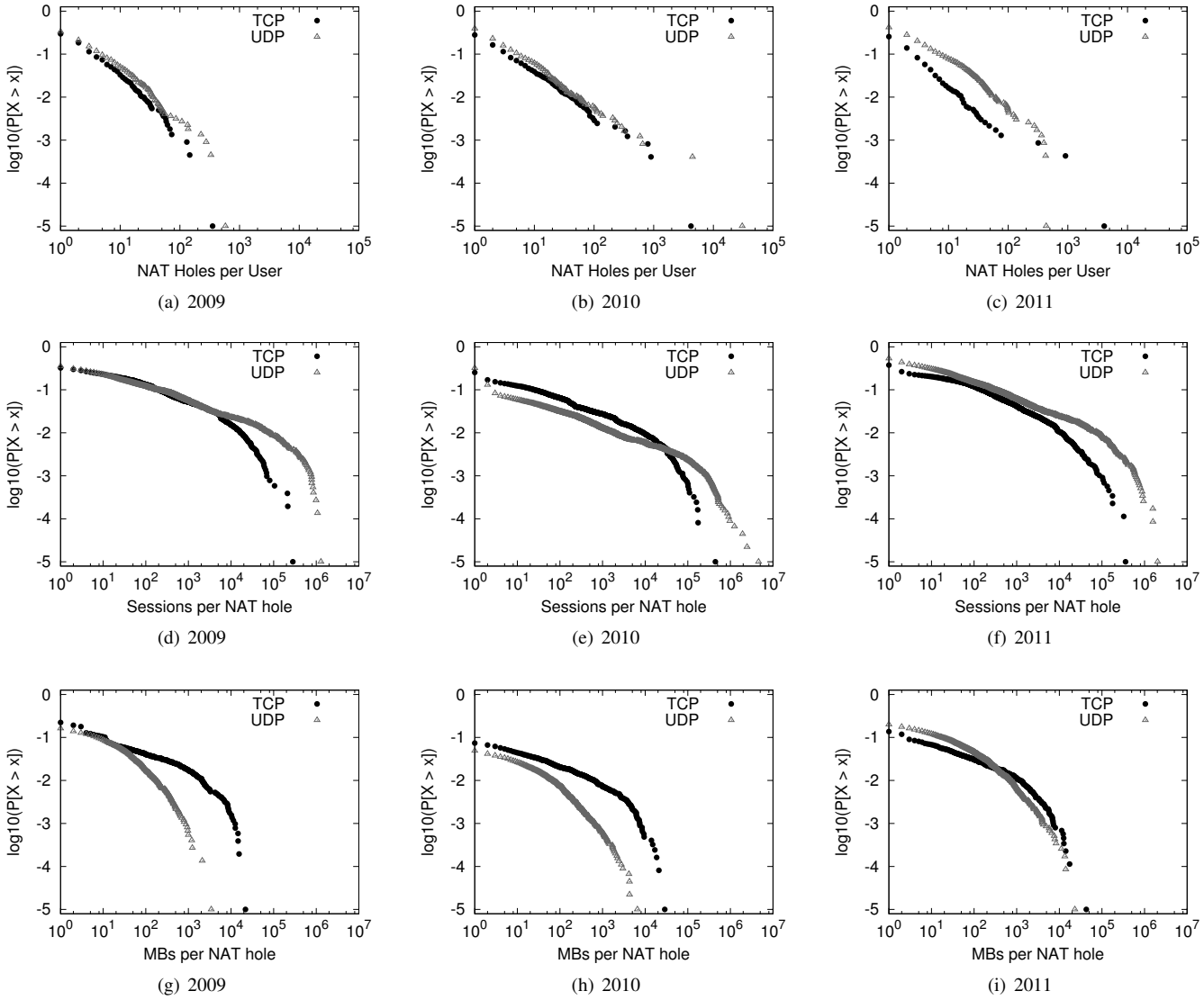
This approach is implemented within the application and does not require any direct support on the router. The best-known example of an application that uses NAT traversal is Skype and most peer-to-peer file sharing programs also implement some form as well. Previous research suggests that NAT traversal is not reliable for TCP connections [13], which may limit the number of TCP applications that implement NAT traversal.

As a result, any residential user can potentially participate in TCP or UDP sessions originating from a remote host, despite the obstacle presented by the NAT used by conventional DSL routers. The aim of this work is to investigate whether this is apparent in measured user behaviour and, if so, to determine what applications are responsible.

III. METHODOLOGY

For this study, we analysed packet header traces captured from a New Zealand ISP in January of 2009, 2010 and 2011. These traces are described in more detail at [14]. The packet traces were filtered to only contain traffic to and from residential DSL customers, using address ranges provided by technicians at the ISP. The ISP assigned a static IP address to each DSL customer, so each observed IP address represented a single residential DSL user. The traces were captured at a location where bidirectional traffic could be observed, i.e. a symmetric link. The ISP did not employ any form of IP address sharing at the time of the captures, so each IP address observed corresponded to one customer only. The trace sets are summarised in Table I.

The number of active customers was determined by counting the number of local IP addresses observed transmitting an outbound packet that is either a TCP SYN ACK or contained payload after the transport header. This prevented inactive IP addresses that were probed or scanned by a remote host from being counted as a customer.



(a) 2009 (b) 2010 (c) 2011

(d) 2009 (e) 2010 (f) 2011

(g) 2009 (h) 2010 (i) 2011

Fig. 2. CCDFs showing the distribution of NAT holes per active user (a, b and c), sessions per NAT hole (d, e and f) and traffic per NAT hole in each of the datasets. The distributions for TCP and UDP NAT holes are plotted separately.

Using libflowmanager [15], the traces were processed to assign each packet to a bidirectional session record, where the session was defined by the 5-tuple (local IP address, remote IP address, local port, remote port and transport protocol) for the flow. All non-inbound sessions were discarded. We defined a session as inbound if the first packet observed for the flow originated from outside the ISP network. For TCP sessions, this first packet also had to have the SYN flag set.

To remove port scans and unsolicited traffic from our analysis, we also discarded sessions that had not been accepted by the local host. To accept a session, the local host must have replied with a payload-bearing packet. A SYN ACK packet was not sufficient for this purpose. An exception was made for TCP sessions where the local host did not transmit a payload-bearing packet but the remote host had sent more than three unique segments. These were classed as accepted to ensure that bulk one-way file transfers, such as FTP data exchanges, were included in our results. Each accepted inbound session was passed through libprotoident [16] to try and identify the application protocol used by that session.

IV. Results

A. Prevalence of Inbound Sessions

We start by examining the number of users that accepted inbound sessions in each of our datasets and the number of "NAT holes" created by each of those users. A NAT hole is defined as a unique 3-tuple (user, transport protocol, local port) on which an inbound session was accepted. The Mean NAT holes per user value was calculated by dividing the total number of NAT holes observed by the number of users that had accepted an inbound connection. The results are summarised in Table II.

Over 50% of the active users in each of the datasets accepted at least one TCP or UDP inbound session. This suggests that the assumption that NAT inherently limits the ability of residential DSL users to accept inbound sessions no longer applies. The proportion of users accepting inbound sessions has remained relatively static over the past three years, although there is a trend towards increased acceptance of UDP sessions. TCP session acceptance was much higher in 2009 than in the subsequent years, although we believe that much of this can be attributed to the Conficker worm, which we will discuss further in §IV-B.

There was a much greater number of UDP NAT holes observed in the 2010 dataset compared to the other years, resulting in the skewed averages in Table II. Using libprotoident, we discovered that 62.4% of the NAT holes observed in the 2010 dataset matched the same payload pattern but the pattern did not belong to a known application protocol[1]. The holes matching the pattern were shared amongst only seven users and contributed just 0.2 % of the total traffic for inbound sessions. If those holes are ignored, the mean NAT holes per user, mean sessions per hole and mean traffic per hole become 8.82, 2436.61 and 31.22 MB respectively, which are more consistent with the results for other years.

Next we examined the distributions of NAT holes per accepting user, the number of flows associated with each NAT hole and the amount of traffic passing in and out of each NAT hole. The distributions for each dataset are shown in Figure III. We observe that all of the distributions shown approximate a Pareto distribution, i.e. they conform to a power law.

Most users accepted inbound sessions on a small number of ports but there was also a small minority of users that utilised hundreds of different TCP and UDP ports for inbound sessions. One user in 2010 accepted connections on 30,415 unique UDP ports – this was due to the "mystery" UDP application noted earlier. That aside, there appears to be an upper bound at approximately 600 UDP NAT holes. Small FTP exchanges were the cause of high TCP port utilisation by a few users. Each FTP session used a different ephemeral port on the client, creating a new NAT hole each time.

Examining the sessions and traffic passing through the NAT holes, we note that the distributions have barely changed for TCP over the past three years. This suggests that there has been little change in the behaviour of the TCP applications and that no "killer-apps" have appeared in that time.

The change in the UDP session per NAT hole distribution in 2010 can again be attributed to the mystery UDP application, which was responsible for a large increase in the number of UDP holes accepting two or three inbound sessions. These holes also generate relatively little traffic, which is apparent in the traffic distribution for 2010. There was a notable increase in the proportion of UDP NAT holes that resulted in more than 100 MB of traffic in 2011. This is due to a growth in UDP BitTorrent traffic, which we demonstrate in Section IV-C.

[1]Libprotoident calls this protocol "Mystery_0D."

TABLE III
PORTS USED TO ACCEPT INBOUND SESSIONS

2009

Port	Users (%)	Sessions (%)	Bytes (%)
TCP 445 (SMB)	6.55	<0.01	<0.01
TCP 80 (HTTP)	4.50	0.29	0.32
TCP 22 (SSH)	3.50	1.33	0.15
TCP 21 (FTP)	3.09	0.01	0.01
TCP 25 (SMTP)	2.26	0.57	0.36
TCP 3389 (RDP)	2.00	<0.01	0.16
TCP 443 (HTTPS)	1.34	0.72	0.15
UDP 3074 (Xbox)	1.25	0.07	0.92
TCP 23 (Telnet)	1.12	0.01	<0.01
UDP 5060 (SIP)	1.10	<0.01	<0.01

2010

Port	Users (%)	Sessions (%)	Bytes (%)
TCP 80 (HTTP)	3.51	0.21	0.37
TCP 22 (SSH)	2.49	1.26	0.19
TCP 3389 (RDP)	2.28	0.02	0.33
TCP 25 (SMTP)	2.28	0.79	0.56
UDP 3074 (Xbox)	2.00	0.22	1.07
TCP 21 (FTP)	1.96	0.06	0.02
TCP 443 (HTTPS)	1.64	0.11	0.36
UDP 53 (DNS)	1.32	<0.01	<0.01
UDP 5060 (SIP)	1.00	<0.01	<0.01
TCP 23 (Telnet)	0.98	<0.01	<0.01

2011

Port	Users (%)	Sessions (%)	Bytes (%)
TCP 80 (HTTP)	3.62	0.15	0.21
UDP 3074 (Xbox)	2.89	0.24	1.76
TCP 3389 (RDP)	2.69	0.66	1.24
TCP 443 (HTTPS)	2.35	0.18	0.33
TCP 21 (FTP)	2.06	<0.01	<0.01
TCP 25 (SMTP)	2.03	0.38	0.31
TCP 22 (SSH)	1.94	0.03	0.08
TCP 23 (Telnet)	1.94	0.72	<0.01
UDP 5060 (SIP)	1.42	<0.01	0.07
UDP 53 (DNS)	1.40	0.01	<0.01

B. Port Utilisation

Next, we analysed the ports that the local host was listening on when an inbound session was accepted. The ten most popular ports by user counts for each of the datasets are shown in Table III. While each port observed in the tables can be associated with a particular application, there is no obvious dominant port. For instance, the ten ports listed for the 2011 dataset combined only accounted for 2.37% of the accepted inbound sessions and 4% of the resulting traffic.

The most popular port observed was TCP port 445 in 2009. This is almost certainly due to the Conficker worm, which was widespread at the time [17], especially given that the port does not appear in the tables for either 2010 or 2011. In those years, the most popular port was TCP port 80, i.e. HTTP.

However, the sessions on that port may not all be instances of residential users running web servers. Many DSL modems are configurable using a web browser and some may have been accepting HTTP connections on the external interface.

There was an increase in the proportion of users accepting sessions on the ports for RDP and Xbox over the past three years, presumably reflecting a growth in remote assistance and online gaming. Overall, though, it is hard to draw any significant conclusions using port-based analysis due to the variety of ports utilised, which has also increased over the three years that we analysed. In the 2009 dataset, accepted inbound sessions were observed on 3675 unique TCP ports and 6250 unique UDP ports. For 2011, our analysis reported 7081 unique TCP ports and 9643 unique UDP ports.

C. Application Protocols

We conducted a similar analysis using the application protocols reported by libprotoident, which we show in Table IV. Skype was easily the most popular protocol and increased in popularity each year. This was not surprising, as Skype has developed a successful NAT traversal protocol for UDP and is an application with broad appeal for residential users.

BitTorrent was the biggest contributor in terms of both sessions (via UDP) and bytes (via TCP). In this context, BitTorrent UDP represents the UDP-based systems employed by BitTorrent implementations which are used to maintain the peer swarms, such as distributed hash tables [18] and peer exchanges. BitTorrent TCP is the protocol by which files are exchanged between peers and it is therefore not surprising that this protocol contributes the most traffic.

More users accept inbound BitTorrent sessions over UDP than TCP. This presumably reflects that only a subset of BitTorrent users are sharing content with other users, which would require a TCP session. However, the UDP DHT maintenance occurs regardless of whether the user is actively sharing files. In 2011, BitTorrent UDP sessions accounted for a much greater proportion of inbound session traffic than previously. Prior to this, DHT sessions were numerous but seldom produced much traffic, but this appears to no longer be the case. We have manually verified the largest BitTorrent UDP flows in the 2011 dataset and confirmed that they had not been misclassified. The largest of the flows contributed 1.35 GB of traffic, almost all of which was sent by the local host. This may indicate that μTP [19] is starting to see wider use by BitTorrent applications.

The increasing popularity of RDP and Xbox Live that we noted when examining port usage can also be observed in Table IV. Other protocols that saw more users were STUN and IPv6 over UDP. At the same time, three protocols declined in popularity over the past three years. MSNC (the MSN file transfer protocol) usage decreased the most, possibly due to the rise of social networking sites which have all the same features but are unaffected by NAT. Gnutella and SSH also declined, which may have been due to users migrating to the BitTorrent and RDP protocols respectively.

TABLE IV
INBOUND SESSIONS BY APPLICATION

2009

Application	Users (%)	Sessions (%)	Bytes (%)
Skype	20.46	1.05	2.14
MSNC	8.44	0.01	0.04
Gnutella (UDP)	8.26	6.63	1.82
BitTorrent (UDP)	7.17	78.47	3.62
HTTP	4.53	0.29	0.31
SSH	3.56	1.33	0.18
SSL/TLS	2.94	0.75	6.85
BitTorrent (TCP)	2.31	2.35	28.55
FTP Control	2.26	0.01	0.01
SMTP	2.24	0.57	0.36
Unknown TCP	18.42	4.06	45.40
Unknown UDP	11.22	0.41	1.2

2010

Application	Users (%)	Sessions (%)	Bytes (%)
Skype	25.06	0.81	5.75
BitTorrent (UDP)	9.10	81.51	7.39
Gnutella (UDP)	7.29	4.28	3.00
MSNC	7.18	0.01	0.05
HTTP	3.58	0.28	1.02
IPv6 over UDP	3.56	0.14	1.36
STUN	2.75	0.02	0.65
SSH	2.64	1.26	0.19
BitTorrent (TCP)	2.58	2.03	29.74
RDP	2.56	0.02	0.50
Unknown TCP	17.20	3.68	34.71
Unknown UDP	9.14	0.96	3.16

2011

Application	Users (%)	Sessions (%)	Bytes (%)
Skype	30.29	1.09	7.86
BitTorrent (UDP)	12.73	85.64	30.46
BitTorrent (TCP)	5.02	7.29	45.86
IPv6 over UDP	4.72	0.42	0.65
HTTP	4.36	0.28	0.75
STUN	3.87	0.06	0.70
RDP	2.90	0.06	1.40
XboxLive	2.76	0.66	1.65
Gnutella (UDP)	2.55	0.51	0.69
MSNC	2.24	<0.01	0.01
Unknown TCP	19.62	0.29	0.53
Unknown UDP	6.93	0.44	4.46

The amount of traffic that was classed as "Unknown TCP" decreased greatly in 2011. We are unsure why libprotoident was better at identifying applications in the 2011 data compared with the preceding years. One theory is that file sharing protocols that libprotoident cannot identify are much less prevalent in 2011, probably as a result of increased BitTorrent usage.

D. TCP NAT Traversal

Finally, we examined the number of TCP NAT holes for which we observed a TCP handshake with two SYN packets, one in each direction, prior to a SYN ACK. This indicates that the NAT hole was created using a technique where both endpoints simultaneously try to connect to one another, thus creating a hole in their local NAT for the other. Other TCP NAT traversal techniques exist (using a public relay, for example), but these were difficult to detect using packet header traces alone.

TABLE V
TCP NAT HOLES USING DUAL SYNS BY APPLICATION

	2009	2010	2011
MSNC	82.6%	92.6%	11.5%
Manolito	12.0%	4.9%	8.3%
Unknown	5.4%	2.5%	80.2%

For the three years of data, we found that 17.8%, 12.2% and 15.2% of TCP NAT holes, respectively, featured at least one instance of a dual SYN TCP session. Using libprotoident, we found that only two identifiable applications were involved in those sessions, as shown in Table V: MSNC and Manolito. However, the proportion of MSNC decreased massively in 2011 while the amount of unidentified NAT holes increased by a similar amount. It is possible that the MSNC protocol changed in 2011 and thus the libprotoident rules may not be identifying it correctly.

V. CONCLUSION

In this paper, we have studied the inbound TCP and UDP sessions accepted by a group of residential DSL users in New Zealand, using packet traces captured from a single New Zealand ISP over the past three years. Our results challenge the common perception that NAT is an obstacle preventing users from utilising applications that accept connections from remote hosts. We also provide insight into the end-to-end services that are being run by residential DSL users on their home computers.

First, we found that over half of the active residential DSL users accepted at least one TCP or UDP session during the period that we measured. This suggests that on-premises NAT should no longer be regarded as an obstacle for residential users who wish to run services or peer-to-peer applications. There are now sufficient mechanisms enabling users to create NAT holes for any application that needs them. We also found that the distribution of NAT holes per user and flows and sessions per NAT hole were long-tailed. Most users utilised less than 10 NAT holes for TCP and UDP, but there were instances where a single IP created over 10,000 holes.

We studied the ports and applications that were being used for accepted inbound sessions. Port usage was spread over thousands of different ports, primarily due to the popularity of peer-to-peer applications that do not use a specific port. Our port analysis also highlighted the effect of the Conficker worm in 2009, which successfully connected with 6.55% of the users over TCP port 445.

The application analysis revealed that Skype and BitTorrent were the most popular applications and have increased significantly in usage over the three years we examined. In particular, the proportion of UDP BitTorrent traffic quadrupled between 2010 and 2011 and the proportion of users accepting Skype connections grew by 50% between 2009 and 2011. STUN, RDP, Xbox Live and IPv6 over UDP were also applications that increased in popularity, while Gnutella, SSH and MSN file transfers decreased.

In the future, we hope to extend this study to analyse traces taken from other ISPs, particularly in other countries, to confirm that the trends reported in this paper are not specific to the ISP that we measured. Another possible study is to compare our results with a similar analysis conducted using traffic from an ISP that uses Large Scale NAT, which may affect the mix of applications that can successfully connect to the ISP's customers.

REFERENCES

[1] K. Cho, K. Fukuda, H. Esaki, and A. Kato, "The Impact and Implications of the Growth of Residential User-to-User Traffic," in *Proceedings of SIGCOMM'06, Sept. 11-15 2006, Pisa, Italy*, 2006.

[2] ——, "Observing Slow Crustal Movement in Residential User Traffic," in *Proceedings of the 2008 ACM CoNEXT Conference*, 2008.

[3] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On Dominant Characteristics of Residential Broadband Internet Traffic," in *IMC '09: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*. New York, NY, USA: ACM, 2009, pp. 90–102.

[4] L. Plissonneau, J.-L. Costeux, and P. Brown, "Analysis of Peer-to-Peer Traffic on ADSL," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science, C. Dovrolis, Ed., vol. 3431. Springer Berlin / Heidelberg, 2005, pp. 69–82.

[5] I. Yamagata, S. Miyakawa, A. Nakagawa, and H. Ashida, "Common Requirements for IP Address Sharing Schemes," http://tools.ietf.org/html/draft-nishitani-cgn-05.

[6] M. Ford, A. Durand, P. Roberts, and P. Levis, "Address Sharing - Coming to a Network near You," *IETF Journal*, vol. 5, no. 1, 2009.

[7] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "RFC 1918 - Address Allocation for Private Internets," 1996.

[8] K. Egevang and P. Francis, "RFC 1631 - The IP Network Address Translator (NAT)," 1994.

[9] "UPnP Forum," http://www.upnp.org/.

[10] J. Rosenberg, J. Weiberger, C. Huitema, and R. Mahy, "RFC 3489 - STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," 2003.

[11] R. Mahy, P. Matthews, and J. Rosenberg, "RFC 5766 - Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)," 2010.

[12] J. Rosenberg, "RFC 5425 - Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," 2010.

[13] S. Guha and P. Francis, "Characterization and Measurement of TCP Traversal through NATs and Firewalls," in *Proceedings of the Internet Measurement Conference, (Berkeley, CA, October 2005)*, 2005.

[14] WAND Network Research Group, "WITS: Waikato Internet Traffic Storage," http://www.wand.net.nz/wits/index.php.

[15] ——, "Libflowmanager," http://research.wand.net.nz/software/libflowmanager.php.

[16] ——, "Libprotoident," http://research.wand.net.nz/software/libprotoident.php.

[17] P. Porras, H. Saidi, and V. Yegneswaran, "An Analysis of Conficker's Logic and Rendezvous Points," http://mtc.sri.com/Conficker/.

[18] A. Loewenstern, "BEP 5 - DHT Protocol," http://www.bittorrent.org/beps/bep_0005.html.

[19] μTorrent, "μTP," http://www.utorrent.com/help/documentation/utp.