

IPv4 TO IPv6 DEPLOYMENT, TRANSITION MECHANISMS AND ISSUES

Richard Macarthy

School of Applied Computing
Faculty of Applied Design and Engineering
Swansea Metropolitan University, UK
Richard.macarthy@smu.ac.uk

ABSTRACT

The Internet Protocol version 4 (IPv4) is a very well established protocol, which has been in use in networks for a vast number of years. IPv6 has been developed as a long-term solution for IPv4 address drain problem. This paper discusses issues about the integration and co-existence of the two protocols and the transition mechanisms when deploying IPv6. Currently IPv4 is the dominant protocol within world networking infrastructures, which means the realistic solution for the transition is to utilize the legacy network and keep the complexity and cost to the minimum. Tunnelling approach offers great advantage over translation and dual stack approaches. Currently the manual tunnelling and 6-to-4 tunnelling have been two favourable techniques, of which the performance will be evaluated in this paper, using a wide area network that is modelled by a popular simulation platform. The statistical results demonstrate the effectiveness of these two techniques.

KEYWORDS

TCP/IP, IPv6, Tunnelling

1. INTRODUCTION

In the 90s approximately 536 million or one eighth of the available IPv4 IP addresses had been used and this number continues to double every five years [1]. At this rate it was predicted that the address space for IPv4 would run out sometime between 2005 and 2011, which is called the exhaustion point. In order to solve this problem, Internet Engineering Task Force (IETF) started on a protocol called the Next Generation Internet Protocol (IPng) which led to the concept of IPv6. The idea was to produce mechanisms that encapsulate IPv6 packets within IPv4 packets and route these packets over existing IPv4 networks. When the final node (exit point) is reached the end point decapsulates the packet (strips

the v4 header) and delivers the packet to the v6 destination [2]. Since migration of IPv4 to Ipv6 will not happen overnight, these mechanisms are very important. They will be in use years to come and are there to provide a smooth transition between both protocols. The goal is to provide a very smooth migration. When looking at migration it is necessary to look at the big picture. It will not only affect professionals within the networking side of Information Technology. It will also affect software programmers who work on Layer 7. They may have to re-write certain aspects of their software to support IPv6. The end users are not bothered about how their data is manipulated and sent across data networks as long as there is little but no change in software that they use.

This paper will only look at two types of tunnelling techniques that have been popularly used for the time being, and will evaluate the metrics gained from different aspects of the network, such as traffic flow and tunnel delay. The two tunnelling techniques studied are 6to4 tunnelling and Manual Configured Tunnelling. 6to4 tunnelling is a point-to-point technique. This means the encapsulation happens within the routers two end points which are normally dual stack routers. Manual Tunnelling is also a point-to-point tunnelling technique.

Section 2 overviews IPv4 and IPv6, and makes theoretical comparisons of these two Internet Protocols, and explains major transition issues and describes the two tunnelling techniques that will be evaluated in this paper. Section 3 demonstrates the designed system that will be used for the performance evaluation. Section 4 shows the statistical results and analysis. Section 5 concludes the paper.

2. IPV6 AND IPV4 TRANSITIONS

2.1. Overview of Internet Protocols

IPv4 was the fourth generation of internet protocol and was also the first version to be widely deployed. The internet protocol sits on the third layer of the OSI network model [3]. The idea was so that any computer on a given network could be uniquely identified with the address it had acquired. The internet protocol also works on a “best effort” service. This does not guarantee that the data packets will be received at a given time, order and may not even be received at all [4]. The 32 bits of IPv4 address scheme is able to provide 2^{32} which is over 4 billion unique addresses. The allocation problem was firstly recognized in the early 90s. IPv6 offers a long-term solution for the problem of IPv4 address drain.

One main difference between Ipv4 and Ipv6 is the length of the address. Ipv4 addresses are 32 bits long and have 4 decimal a bit octets separated by dots (x.x.x.x.), An Ipv6 address is 128 bits long and consists of 4 hexadecimal values separated by colons (xxxx:xxxx:xxxx:xxxx) each 4th of the address is 32 bits long. This new address structure allows alot more address allocation globally. The Ipv6 header has also been modified. The new header is 40-bytes in length compared to Ipv4’s 20-bytes and has no option or checksum fields but instead uses Extension Headers. A new field ‘Traffic Class’ replaces the ToS field in the Ipv4 header. This field sets priority on certain data packets. Table 1 shows the major differences between IPv4 and IPv6 protocol.

Table 1. Comparison of IPv4 and IPv6

	IPv4	IPv6
Address	Primary and secondary	Multiple global addresses, link-local addresses
Address auto-configuration	Stateful	Stateless and stateful
Broadcast	Yes	No (replaced by multicast)
Anycast	No	Yes
Structure of the Header	Variable size (to support options)	Fixed base header (one or more optional extension headers)
Flow Labelling	No	Yes

2.2. Transition Approaches and Issues

There are a few options that have been proposed to offer the transition from IPv4 to IPv6 – dual stack, translations, and tunnelling. In the dual stack enabled network systems, both IPv4 and IPv6 running side by side. End systems (clients and servers) and networking devices (routers, etc) have to support both IPv4 and IPv6. The advantage of this transition mechanism is the simplicity. The DNS (*Domain Name Service*) can drive the transition to complete. However, in this scheme, IPv4 addresses are still required, which does not alleviate the pressure of IPv4 address allocation. Besides, the two stack structure may be a resource burden in terms of network management.

The other transition approach is to use translator to convert packets from one protocol to another. This translation scheme supports data communications between IPv4-only and IPv6-only nodes, and the network nodes do not need to support any further extensions. The problem with this approach is the added complexity and poor scalability. The network is engaged with a stateful routing manner which means traffic must be symmetric. On top of that the assigned protocol translator is a single point of failure, and can also be an attractive attack target.

Tunnelling techniques encapsulate IPv6 packets inside IPv4 packets to be transmitted over the Internet, in such a way that the legacy network equipments can be reused and the transition process can be conducted in an incremental and gradual pace. Tunnelling techniques are particularly helpful for edge-to-core transitions. There are a couple of disadvantages of tunnelling techniques. For example, tunnels can create routing inefficiencies which can compromise the network performance; also there may be some potential security problems with certain tunnelling technologies. Currently there is six ways of tunnelling packets which are automatic, 6to4, ISATAP (*Intra-Site Automatic Tunnel Addressing Protocol*), 6over4, Teredo and Manual Tunnelling. Among these techniques, manual tunnelling and 6-to-4 tunnelling have been widely used, for the reason of being easy-to-use.

2.3. Manual Tunnelling

Manual configured tunnels also referred to as 6-in-4 tunnels have to be literally manually configured on both ends of the tunnel. Both gateway routers must know each other’s IPv4 addresses in which to tunnel IPv6 traffic over IPv4. These tunnels are quite reliable, but are not very scalable. Manual configured tunnels would not typically be used in wide area networking. Manual tunnels are said to be the most stable of tunnels due to the control an administrator has over the tunnel end points. Manual tunnels also offer good security if configured

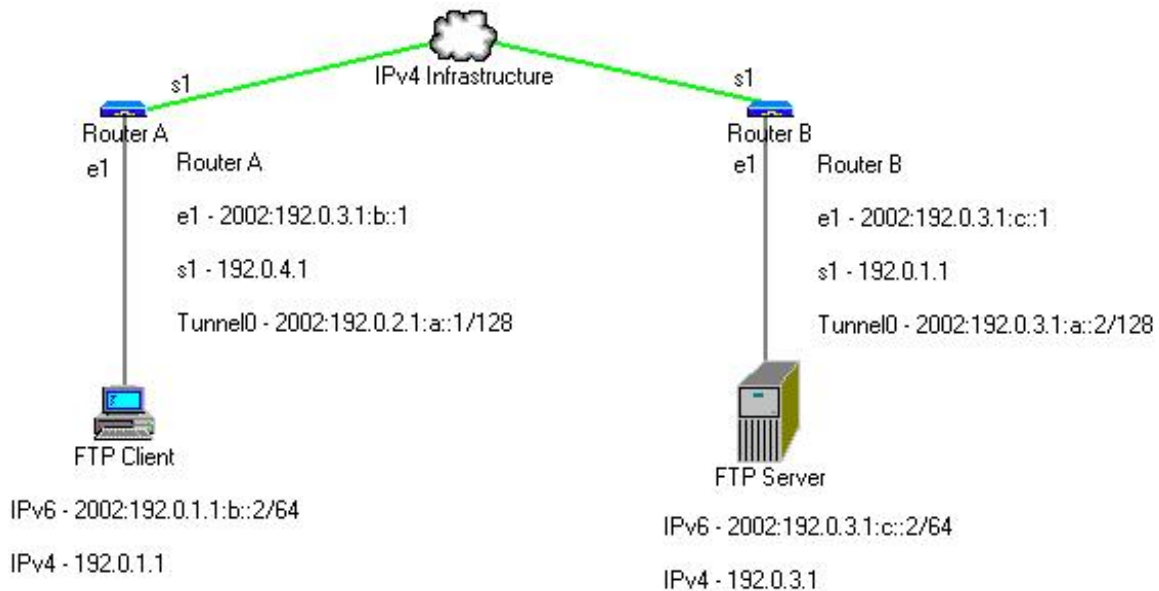


Figure 1. Network Scenario 6to4 Tunnel

properly. One security issue of manual configured tunnels is if the tunnel end point does not check the source of the IPv4 packets entering the network was the one configured by the administrator. A drawback to manual tunnels is the amount work to set them up. The IP addresses have to be hard coded into the router. This means that they are not suitable for networks with dynamic IPv4 addressing. There can be mechanisms to work around

this problem, but may not be worth the trouble. Manual tunnels should be used for small networks where an administrator can configure the tunnels properly, without any possible human error.

2.4. 6-to-4 Tunnelling

6to4 tunnels create dynamic tunnels over IPv4 networks and have a prefix of 2002::/16. 6to4 tunnels do not need IPv4 compatible addresses. They require only one IPv6 router with an IPv4 routable address. The native IPv6 network traffic can then be routed over IPv4 infrastructures using gateway 6to4 routers. A 6to4 router can communicate with any other router so long as the neighbouring router is configured with a 6to4 address. Unlike manual tunnelling where both gateway routers need each other's addresses. 6to4 hosts or routers accept and decapsulate traffic from any source. They encapsulate and decapsulate on a pseudo interface. This means that 6to4 networks are prone to attacks such as buffer overflow attacks on the routers pseudo interface, this could be a problem. Buffer overflow attacks can be solved by filtering traffic from a certain range of IP addresses and prefixes. The 6to4 pseudo interface cannot tell the difference between malicious traffic encapsulated inside IPv4 packets and genuine traffic from 6to4 relays. 6to4 tunnels are also prone to denial of service attacks. The IPv4 traffic has malicious data inside of the packet destined for native IPv6 sites.

This leads to the conclusion that 6to4 traffic forwarded over an administrated router should always be monitored. More so if the router is configured with an IPv4 anycast address for public 6to4 relays. Other than monitoring traffic flow across the routers no other management is needed for 6to4. This tunnelling technique was designed for ease of use and low maintenance. 6to4 is a very promising tunnelling technique and it is very possible that this technique will be the one most widely deployed on the internet as we know it, for migration and integration purposes [5].

3. SYSTEM DESIGN

OPNET Modeler is used as the network modelling and simulation tool [6]. It is a commercial network simulation software package which provides a platform for modelling and simulating network applications. Using OPNET a simple wide area network topology is deployed, which consists of one FTP server, 2 routers, 1 IPv4 Network Infrastructure cloud and one FTP client Ethernet workstation (see Fig. 1).

The IPv4 Cloud consists of only IPv4 traffic. Whereas the router addresses and its subnets will be configured with IPv6 addresses, the routers tunnelling addresses will be configured accordingly. The FTP server has an inter-request time of uniform 1,120. And a file size of 500000 bytes. This means that a 500kb file will be generated and sent across the network on an average of once every minute. Encapsulation happens at Router B then IPv4 packets are forwarded over the IPv4 infrastructure, decapsulated at Router A and forwarded to the IPv6 FTP Client. Address assignment is shown in Fig 1 for 6to4 and Fig 2 for Manual. Comparing these two tunnels against each other will lead to a conclusion of

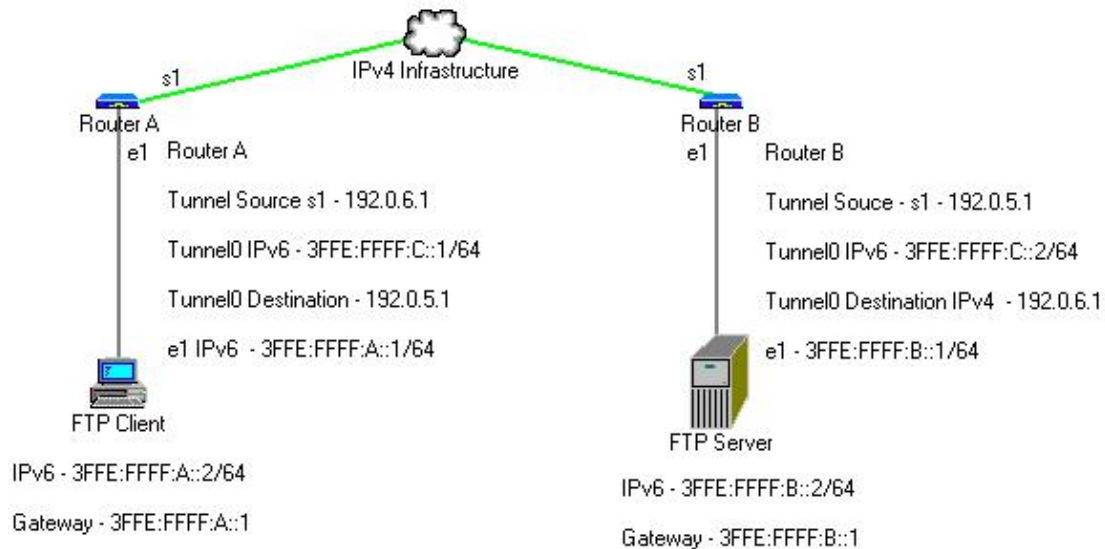


Figure 2. Network Scenario Manual Tunnel

which type of tunnel an administrator would like to use within a network. This test does not take security into consideration. It is purely for statistical analysis and testing.

4. PERFORMANCE EVALUATION

The simulation was run for a virtual time of three days in OPNET. The statistics gained are as follows. Figure 3 shows the TCP delays, which is the delay (in seconds) of packets received by the TCP layers in the complete network, for all connections. It is measured from the time an application data packet is sent from the source TCP layer to the time it is completely received by the TCP layer in the destination node. The result shows that the delay on both tunnels is rather minimal. It shows an average delay of 0.096 seconds for the 6to4 tunnel and a delay of 0.103 for the Manual tunnel. The difference in delay is therefore 0.007 seconds in favour of the 6to4 tunnel. Both results stay rather stable through network stabilisation. This result might indicate that it takes longer to work out the destination address from the source when entering a manual tunnel. The IPv4 addresses in manual tunnels are hard copied into the tunnel points the routing tables.

The average Tunnel End-to-End Delay (seconds) is the delay experienced by a packet coming through a tunnel, i.e. the difference between the time at which the packet is sent in on the tunnel and the time at which it is received at the opposite end, in seconds. This includes the encapsulation and decapsulation delays. Figure 4 demonstrates the end-to-end delay of Tunnel0 on Router A. It indicates that the delays again are very minimal. The Manual tunnel end-to-end delay shows an average time of 0.0015 seconds and gets lower as the network stabilises around a day and a half through. Over the 3 days the delay drops to ~ 0.00147 seconds. The 6to4 tunnel out performs the

manual tunnel again in this test as the delay starts at ~ 0.00143 seconds and quickly drops after around half a day and stays stable at ~ 0.00137 seconds. The 6to4 tunnel is superior to the manual tunnel in favour of 0.00010 seconds after stabilisation. This is a very minimal delay time. This result is not surprising since both 6to4 and Manual tunnelling techniques are both router-to-router this is where encapsulation takes place. The small overhead of the Manual tunnel may be to do with the static addressing needed to be checked before forwarding. 6to4 tunnels forward their traffic to the nearest 6to4 relay.

This result is taken from Router A receiving the IPv4 packet and stripping it off. On the contrary Router B needs to perform the opposite to this task encapsulating the IPv6 packets inside IPv4 packets. So generally the results gained here need to be added to that to give an overall time of how long this process takes, for the entire network.

Figure 5 show the FTP traffic received globally within the network scenario. Average FTP Traffic Received (bytes/sec) measures the average bytes per second forwarded to all File Transfer Protocols (FTP) applications by the transport layers in the network. The Blue line indicates the Manual tunnel scenario. The results show that when the network is stabilising the manual tunnel isn't performing very well and it only receiving around 7,600 bytes per second. But when stable it manages to receive rather steadily at 8000 bytes per second. The pink line indicates the 6to4 tunnel scenario. When the network simulation is started the manual tunnels FTP traffic revived actually rises for a short amount of time then quickly falls dramatically level with the Manual tunnel scenario. It then continues to drop at a slow rate until it is virtually level with the Manual tunnel. This shows that after a period of about three days the two tunnelling techniques actually work on a similar level

when TCP traffic is concerned. After network stabilisation there is little difference between both tunnel methods. This result reflects the end-to-end tunnel delay results. So the increase of performance on the Manual tunnel and decrease of performance in the 6to4 tunnel is expected.

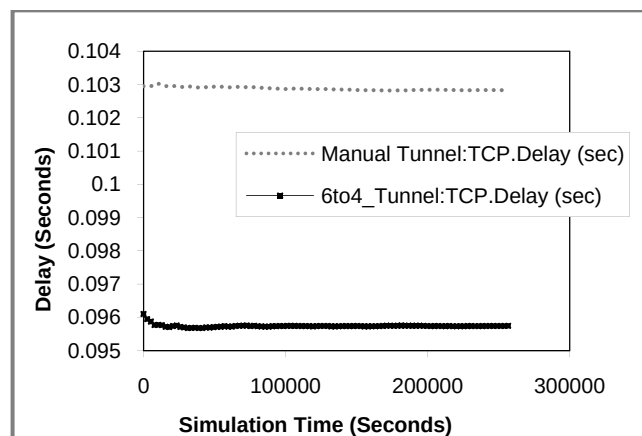


Figure 3. TCP Delay

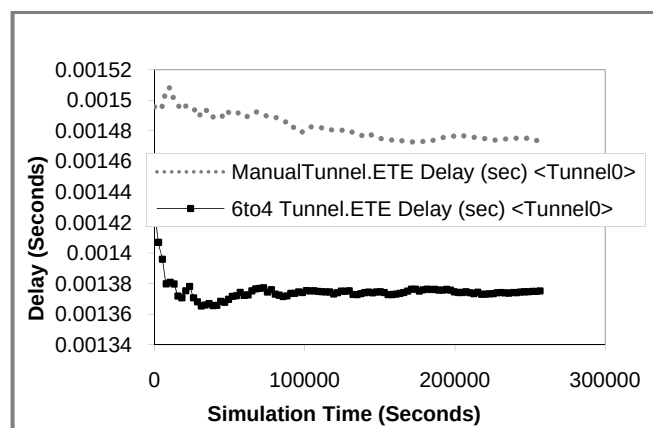


Figure 4. Tunnel ETE Delays

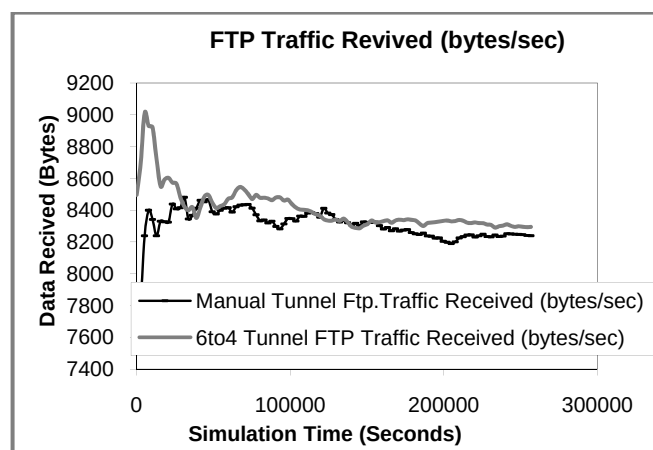


Figure 5 FTP Traffic Received (bytes/sec)

5. CONCLUSIONS

Two network scenarios have been created to evaluate the performance of the tunnelling techniques. One scenario used a manual configured tunnel and the other a 6to4 dynamic tunnel. The results showed that the 6to4 tunnel was the most efficient tunnel overall. The tunnel TCP and Tunnel End-to-End delay on the 6to4 tunnel was less than that of the Manual tunnel. Though the 6to4 tunnel experienced less delay the delay it experienced was not great in difference to the Manual tunnel. From looking at the FTP received results, it shows that there was little difference in traffic revived when the network had stabilised. If the simulation was run for longer than three days maybe the two FTP received results would have met in the middle. It seems to show inverse correlation where the Manual tunnel is increasing and the 6to4 tunnel is decreasing. Since both tunnels are router-to-router tunnels, the results were quite similar, this was no surprise.

With regards to what tunnel a network administrator should use when choosing a migration plan, from these small results it would be feasible to choose either one of these two techniques. The real decision lies within the size of the network that the administrator decides to implement. As mentioned before, manual tunnels need to literally be configured manually. This does not suit well for large scalability, but it very good for security purposes. On the other hand there is the 6to4 tunnels, they are dynamic tunnels designed for ease of use and low maintenance. Security could be a small issue, but looking at the results 6to4 seems to perform better within this scenario. Choosing a 6to4 tunnel would be the ideal way to introduce tunnelling on a large scale network such a MAN or a WAN. But for LAN's a manual tunnel would be just as sufficient if configured correctly.

REFERENCES

- [1] LLjitsch van Beijnum, Running IPv6 2006, CA: USA, Apress, 2006, pp. 2.
- [2] LLjitsch van Beijnum, Running IPv6, CA: USA, Apress, 2006, pp. 33.
- [3] Martin P.Clark, Data Networks, IP and the Internet Protocols Design and Operation, London, Wiley, 2003, pp. 70.
- [4] James F.Kurose, Keith W.Ross, Computer Networking: A Top-down Approach Featuring the Internet, Third Edition, MA: USA, Addison-Wesley, 2001, pp. 284.
- [5] 2000Smith, A.B. & Jones, C.D., Getting it Right, *ABC Transactions on XYZ*, Vol. 10, No. 5, Aug 2004.
- [6] OPNET Modeler, <http://www.opnet.com>