Evaluation of IPv6 transition mechanisms using QoS service policies

SMITH, Luke, JACOBI, Mark and AL-KHAYATT, Samir

Citation:

SMITH, Luke, JACOBI, Mark and AL-KHAYATT, Samir (2018). Evaluation of IPv6 transition mechanisms using QoS service policies. In: 2018 11th International Symposium on Communication Systems, Networks and Digital Signal Processing,, Budapest Hungary, 18-20 July 2018. IEEE. [Conference or Workshop Item]

Evaluation of IPv6 transition mechanisms using QoS service policies

Luke Smith, Mark Jacobi, Samir Al-Khayatt
Sheffield Hallam University, Faculty of Art, Computing, Engineering and Science, Sheffield, S1 1WB, UK.
Correspondence Email: m.jacobi@shu.ac.uk

***Abstract*—IPv6 networks are rarely fully IPv6 from end to end particularly when interconnected over other providers core data networks, hence the need for IPv6 transition methods or mechanisms. Previous studies have identified some potential impact on performance where transitions occur. This study considered implementations where Quality of Service (QoS) service policies have been applied in the IPv6 network to determine if the transition mechanisms were able to guarantee the same performance and level of service when the IPv6 traffic transitions over a core IPv4 network. Using a test lab, traffic generators and data capture tools the study was able to fully test the transition mechanisms using data rates and link speeds that replicated equipment and traffic levels used in real world implementations. The study showed that the Quality of Service classes applied were maintained across the transition network providing the service guarantees required for a range of traffic classes. Results indicated that the transition implementations on the devices used (Cisco ISR 4351) translate the QoS settings from IPv6 to IPV4 and vice versa were highly effective and with negligible additional impact on performance occurring due to the additional processing required.**

Keywords— IPv4, IPv6, QoS, NAT64, 6to4 tunneling, Cisco, IXIA.

I. INTRODUCTION

The recent exhaustion of IPv4 addresses by IANA has made the requirement for IPv6 greater. But the incompatibilities of the two protocols means at least for now and into the future both protocols need to co-exist while maintaining the inter-communication between both. In most practical implementations, IPv6 islands will be interconnected via IPv4 networks. This is particularly likely over Internet Service Providers (ISPs) and data service providers.

Many transition mechanisms have been proposed that have since been depreciated, and are described in detail by [1]. The three predominant transition technologies that are now widely supported are dual stack implementations, tunneling, and translation. [2].

A. *Overview of Transition methods*

One of the predominant implementations of IPv6 alongside IPv4 is the dual stack implementations, effectively running both protocols at the same time. The limitations however of such an implementation often come down to the increased resource requirements of running both protocols.

Tunneling is used in a variety of scenarios such as VPN tunnels and so forth, however it can be applied specifically to IPv6 as well. Tunneling encapsulates IPv6 packets in IPv4 packets and uses the IPv4 network as a link-layer mechanism. Multiple variations exist with the main ones being Manually configured tunnels, Generic routing encapsulation (GRE), and automatic 6to4 tunnels. [1]

Translation mechanisms require that devices convert the headers from IPv6 to IPv4 and vice versa. The predominant method outlined for translation falls under NAT64 after previous recommendations make NAT-PT depreciated. Both methods fall under what is considered Stateful and Stateless translation mechanisms.

These methods potentially all have some performance impact on the devices used and the data transmitted over them.

B. *Quality of Service*

Implementation of Quality of Service (QoS) mechanisms into data networks to prioritize and/or reserve bandwidth for real-time and other important management traffic has become commonplace. This has become necessary because of native IPv4's inability to guarantee packet delivery when congestion occurs. Mechanisms such as IntServ [2] and DiffServ [3] have been defined to overcome this shortcoming, and IPv6 was defined to be compatible with these QoS methods [4]. Diffserv has become the accepted method for applying QoS due to its scalability and lower impact on router resources.

C. *Related Work*

Previous work has investigated QoS performance of native IPv6 connectivity when utilizing Diffserv architecture in a test network [5]. This work concluded that IPv4 and IPv6 performance guarantees were identical when implemented in Cisco gigabit Ethernet routers but older line cards showed IPv6 performance was worse. It also determined that where switching was implemented in software, performance at high load negatively implemented the control traffic. The authors recommended repeating similar tests when vendors have placed more emphasis on IPv6 in their roadmaps.

A number of studies have compared the performance of 6in4 tunneling compared to pure IPv4 or IPv6 [6], [7], [8]. Unsurprisingly they determine that throughput is slightly increased due to the additional header and agree that native IPv6 has lower end-to-end delay than IPv4. Sathu and Shah also compared dual stack and found it only had a slight increase in throughput over IPv6 [7]. Where both IPv4 and IPv6 are operational on each host the additional overhead from effectively duplicating the implementations also identified by Wu et al. as a potential shortfall, will always create a marginal increase [1].

For the tunneling mechanisms themselves Sathu and Shah concluded that 6-to-4 tunneling was slightly better than 6-in-4 [7]. However the variation is quite small indicating neither is necessarily better than the other. This makes sense considering the architectural differences between the two mainly come down to either an automatic tunnel deployment or manual deployment. The actual tunneling process remains the same. But as was shown again like in the previous studies the tunneling caused increased throughput indicating an increased bandwidth requirements for them.

Dual stack implementation in IPv4 and IPv6 was evaluated by Fatah et al. [9]. The study determined that the increased header size also increased the round trip time (RTT), but jitter was better in IPv6 due to the use of the flow label in the header.

D. Rationale for this study

As described above, there is a large body of research providing comparisons of QoS in native IPv4 and IPv6. There are also a number of studies that investigate performance of various transition methods [4], [6], [9]. A reoccurring theme for the various studies investigated is the lack of performance analysis during periods of congestion on the network. Results are oriented for the performance implication of the transition mechanism itself in comparison to the baseline network. More realistic is networks that have a variety of user load on them.

Moreover few studies provide a collaborative study looking at the various methods together, providing a baseline performance test for each and comparing against the results from the transition mechanism. Performing this on a consistent network topology with consistent QoS implementations provides a complete overview of the performance variations of the various transition mechanisms and provides an indication into the effect each has.

The last noticeable gap is the lack of studies utilizing real hardware devices, with the notable exception of [5]. The majority of studies utilize modelling and simulation tools for simplified deployment. This therefore provides a gap for hardware based performance tests for an indication as to how real world equipment compares to those in a simulation environment.

The study described here aimed to fill those gaps providing an overview of performance during congestion with established QoS schemes in place. It addresses the recommendations of [5] as it uses Cisco ISR 4351 routers which are the currently recommended router platforms for branch offices [10]. These routers include faster IPv6 switching and were configured to apply the currently recommended traffic class service policies including CS6 for Network Control traffic [11]. The study aimed to determine if the QoS mechanisms and behaviors implemented in the native IPv6 network are impacted and therefore are less effective where transition mechanisms are used to route the traffic over IPv4 core networks, before transitioning back to IPv6. Through the use of a test lab, traffic generators and data capture tools the study was able to fully test the transition mechanisms using data rates and link speeds that replicated equipment and traffic levels used in real world implementations.

II. Measurement Approach

As previously mentioned, there are three predominant topologies/scenarios that are being utilized in the project, namely dual stack, tunneling, and translation. The logical topologies are provided for these in Figs. 2 to 4 respectively. A base logical topology is also provided in Fig. 1 as a reference point.

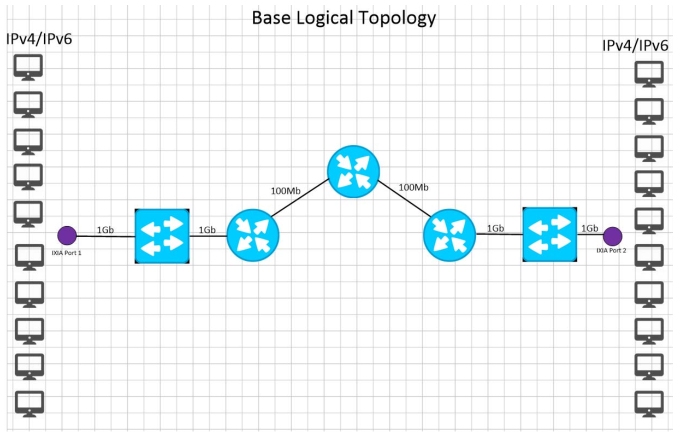


Fig. 1. Base Logical Topology

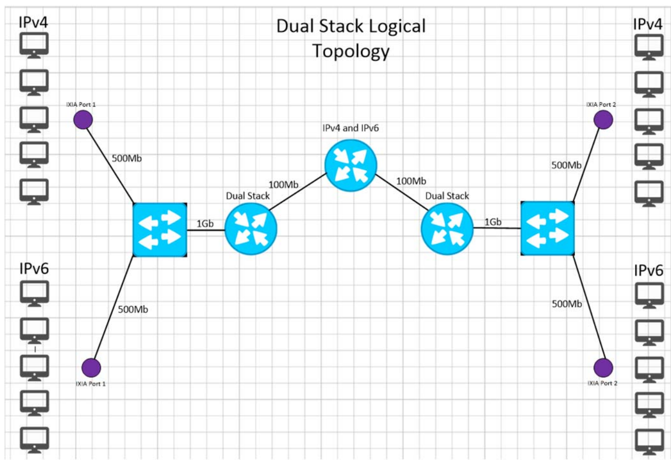


Fig. 2. Dual Stack Logical Topology

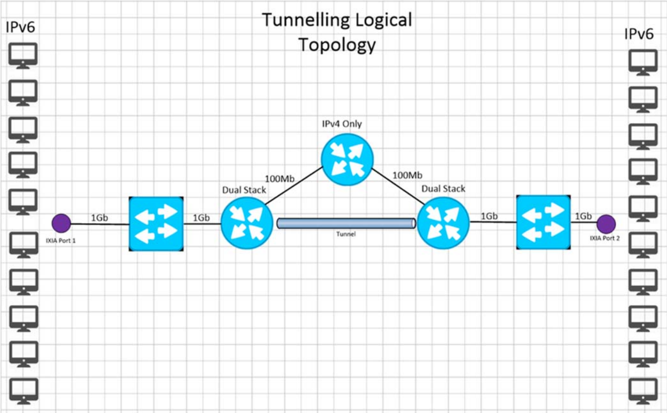


Fig. 3. Tunneling Logical Topology

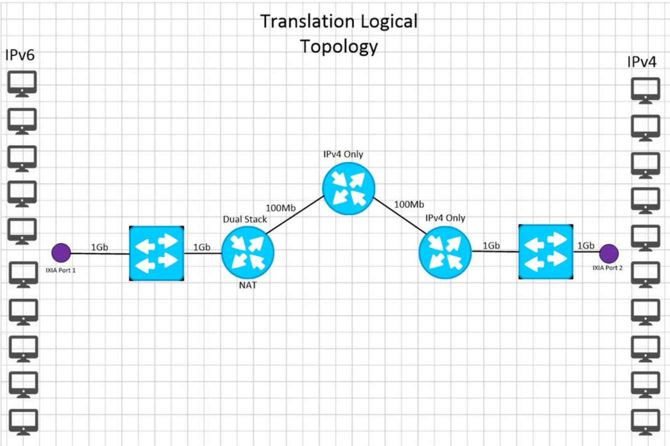


Fig. 4. NAT Translation Logical Topology

A. *Network Equipment*

The overall topologies emulated a central and remote branch site interconnected through a WAN link. Cisco's ISR 4351's were used for the edge devices on both sides of the topology. The reasoning behind the choice is that these devices are the recommended Cisco router range for medium branch sites and as such are applicable to the emulated topology requirements.

Cisco 3850 switches were utilized as distribution switches. Again, these are recommended switches for a medium branch for the backbone therefore making them applicable to the emulated topology requirements and match well with the ISR4351.

B. *QoS Implementation*

DiffServ was used as the QoS architecture to use for the topology. DiffServ is predominantly used due to the lower overhead and scalability that it provides [4].

To provide a performance baseline a QoS scheme was utilized for the network. The same QoS scheme was applied to both IPv4 and IPv6 packet types. After extensive review from [12] the eight-class QoS model was utilized with minor adjustments to utilize the classes that are applicable to the topology. Table 1 shows the classes defined, their associated DSCP value, the associated bandwidth utilization of each class, and the applicable Per Hop Behavior (PHB). The IXIA traffic generator pre-marked the packets so no classification is done on the devices themselves.

TABLE I. BASELINE QOS VALUES

Class	DSCP	Bandwidth	PHB	WRED
Voice	46	10%	EF	No
Video	34	23%	AF41	No
Network Control	48	7%	CS6	Yes
Signaling	24	4%	CS3	No
Transactional Data	18	26%	AF21	Yes
Best Effort	0	27%	DF	Yes
Scavenger	8	3%	CS1	No

Class Based Weighted Fair Queuing (CBWFQ) was used to match the requirements of the traffic passing over the network along with Low Latency Queueing (LLQ) for real time traffic in accordance with [12].

For congestion avoidance WRED was used in certain scenarios, dependent on the PHB.

C. *Background Traffic Rates*

In order to provide as close to realistic as possible background traffic variations in bandwidth were utilized for the different classes according to Table 2. Traffic rates were per direction (unidirectional), so should therefore be doubled for bi-directional traffic, e.g. a concurrent phone call at 2Mbps will be a total of 4Mbps in both directions.

TABLE II. CLASS TRAFFIC PROFILES

Traffic Class	Traffic Rate	Equivalent Example
EF (Voice)	2Mbps	20 concurrent VoIP Calls
CS3 (Signaling)	100Kbps	150 IP Phones for SCCP control traffic
AF41 (Video)	35Mbps	2 Cisco Telepresence concurrent 1080p video calls
DF (Best Effort)	40Mbps	N/A
AF21 (Transactional Data)	20Mbps	Oracle Data Guard with 2Mbps of Redo traffic

CS6 (Network Control)	2Mbps	20,000 Objects within Solarwinds for Network Management
CS1 (Scavenger)	150Mbps	20 Netflix/YouTube High Definition Streams

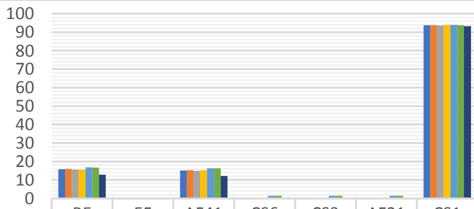
Total traffic rates equate to approximately 249Mbps, which is over double the prescribed bandwidth amount of the WAN link used (100Mbps). This provides the platform for congestion and consequent enforcement of QoS policies.

D. Traffic Generator and Data Capture tools

The traffic was generated using IXIA traffic generator hardware. IXIA's IXNetwork software was used to define stateless traffic classes and because it utilizes two data ports, was also used to capture the data statistics.

III. Results

The results in Fig. 5 demonstrate the effectiveness of the QoS behaviors implemented. Loss for the EF (Voice class), CS6 (Network control), CS3 (Signaling) and AF21 (Transactional data) classes is minimal, while for low priority traffic such as CS1 (scavenger), and AF41 (video) is higher as expected due to the limited bandwidth guarantee for this class. The uniformity across transition methods for each class indicates that all methods implement the QoS behaviors effectively.



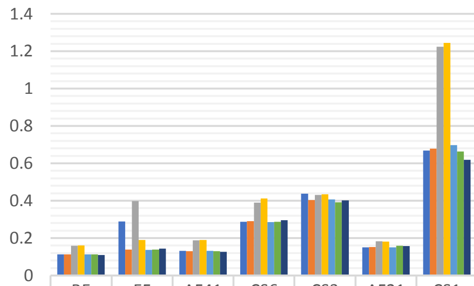
	BE	EF	AF41	CS6	CS3	AF21	CS1
Base IPv4	15.852	0	15.103	0	0	0	93.691
Base IPv6	15.915	0	15.259	0	0	0	93.741
Dual Stack v4	15.659	0	14.993	0	0	0	93.575
Dual Stack v6	15.58	0	15.213	0	0	0	93.875
Manual Tunnel	16.844	0	16.36	1.393	1.383	1.4	93.91
Automatic 6-to-4 Tunnel	16.71	0	16.218	1.396	1.469	1.393	93.738
Stateful NAT64	12.797	0.002	12.102	0	0	0	93.302

Fig. 5. Traffic class Loss (%) comparison

Figs. 6 and 7 show similar results for other critical performance metrics. Comparison of the transition methods with a purely IPv6 network show negligible variations, no significant differences in packet loss, latency or delay variation were measured. Critically for the EF (Voice) category, performance levels for jitter and latency were well within the recommended guidelines of 30ms and 150ms respectively [12]. Predominant loss was seen with CS1 being the Scavenger class, with the lowest bandwidth allocation resulting in higher drops on output queues. This displayed the worst jitter and latency as well.

The IPv6 scenario had an identical QoS implementation, with results producing near identical metrics for loss, jitter, and latency. Variations were sub millisecond, although the tunneling and translation scenarios showed higher latency for almost all traffic classes due to the additional overhead caused by the tunneling and transition processes. This was also as expected.

Dual stack scenarios showed increased delay variation in scavenger CS1 class. This can be explained by the high bandwidth of data in this class requiring significant processor resources. Wu et al. suggested that increased resource requirements were a limitation of dual stack implementations [1].

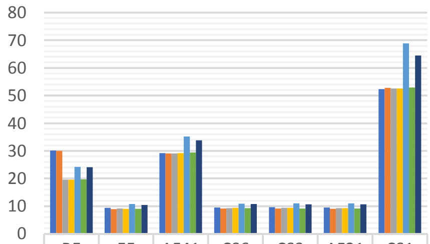


	BE	EF	AF41	CS6	CS3	AF21	CS1
Base IPv4	0.113	0.289	0.131	0.288	0.438	0.151	0.669
Base IPv6	0.113	0.139	0.13	0.29	0.404	0.153	0.679
Dual Stack v4	0.159	0.399	0.188	0.39	0.431	0.183	1.224
Dual Stack v6	0.16	0.19	0.189	0.412	0.434	0.182	1.244
Manual Tunnel	0.113	0.136	0.131	0.285	0.407	0.151	0.697
Automatic 6-to-4 Tunnel	0.112	0.139	0.13	0.288	0.392	0.159	0.663
Stateful NAT64	0.11	0.143	0.127	0.295	0.401	0.157	0.618

Fig. 6. Traffic class average delay variation (msec)

Manual tunneling showed a slight increase in latency across all classes. This was expected due to the operation of the tunnel process and confirms results found in [6]. The reduction in latency for the automatic tunneling suggests more efficient tunnel establishment than for manual tunneling confirming findings from [13].

The same metrics were measured in both directions across the circuit although only those from SW1 to SW2 have been shown here. The results from SW2 to SW1 were identical as expected, other than the increase in Stateful NAT64 latency seen in Fig. 7 was not seen in traffic flows in the opposite direction. This is explained by traffic originating in the IPv6 network requiring the creation of the translation in the NAT table, which is already in place for traffic returning in the opposite direction.



	BE	EF	AF41	CS6	CS3	AF21	CS1
Base IPv4	30.116	9.375	29.248	9.575	9.629	9.504	52.341
Base IPv6	29.988	8.924	29.132	9.124	9.178	9.094	52.779
Dual Stack v4	19.583	9.147	29.133	9.322	9.365	9.288	52.608
Dual Stack v6	19.589	9.109	29.14	9.387	9.38	9.297	52.592
Manual Tunnel	24.222	10.8	35.26	10.904	10.966	11.028	68.87
Automatic 6-to-4 Tunnel	19.696	9.022	29.452	9.281	9.201	9.197	52.890
Stateful NAT64	24.12	10.416	33.815	10.788	10.624	10.699	64.512

Fig. 7. Traffic average class latency (msec)

IV. Conclusions and Further Work

The work and results described here filled a gap in existing research studies by comparing the main IPv6 to IPv4 transition mechanisms while implementing industry standard QoS class profiles. Tests were completed using a range of traffic types including voice and video at loads that created congestion on links forcing traffic shaping and dropping mechanisms to operate. The results from this study demonstrate that the small variations in performance metrics result predominantly from the mechanism of implementation. In the case of manual tunneling the additional processing required, and translation within the mechanism of translation added expected delay. Beyond increases in latency, variations in the metrics are fairly insubstantial when considering traffic amounts and unlikely to be experienced within the applications. The QoS schemes continued to operate optimally providing near 0% packet loss in all occurrences for EF traffic class used for Voice over IP. It can therefore be deduced that QoS implementations existing for current IPv4 networks require little adaption, if any, when implementing various transition scenarios when moving to IPv6.

The study also shows that the implementation of these transition methods by the devices used for these experiments, in this case Cisco; successfully perform the transitions with negligible impact on performance. This however may be vendor specific, so further research which uses alternative vendors would determine if the transition standards are robust enough to operate across all vendor implementations. Furthermore, traffic generated during the tests was stateless in nature and pre-marked. Therefore further testing utilizing stateful application traffic for each class would determine whether mechanisms and location for classification require adjustment when considering the various transition scenarios.

The tunneling methods applied here were effectively point-to-point tunneling as are all the sources referred to. However there are circumstances where point-to-multipoint tunnels may be more appropriate. Existing devices can only apply QoS classes to physical interfaces which preclude creating per tunnel class policies. Additional mechanisms such as Cisco's Dynamic Multipoint VPN (DMVPN) feature would be required to apply QoS on a per-tunnel instance [12]. This could also be used to tunnel across a range of IPv4 and IPv6 networks but may also introduce additional performance impact and is worth further investigation.

References

[1] P. Wu, Y. Cui, J. Wu, J. Liu and C. Metz, "Transition from IPv4 to IPv6: A State-of-the-Art Survey," in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1407-1424, Third Quarter 2013.

[2] Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J., and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, November 2000, https://www.rfc-editor.org/info/rfc2998.txt.

[3] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998, https://www.rfc-editor.org/info/rfc2475.txt.

[4] H. Al-Fayyadh and M. Koyuncu, "Comparison of QoS architectures for VoIP traffics in IPv4 and IPv6", *IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)*, Baku, 2016, pp. 1-5.

[5] A. Liakopoulos, D. Kalogeras, V. Maglaris, D. Primpas, and C. Bouras, "QoS experiences in native IPv6 networks", *International Journal of Network Management*, vol. 19, issue 2, pp 119-137, March/April 2009.

[6] J. L. Shah and J. Parvez, "Performance evaluation of applications in manual 6in4 tuneling and native IPv6/IPv4 environments," *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kanyakumari, 2014, pp. 782-786.

[7] H. Sathu, M. A. Shah, "Performance Analysis of Video Protocols over IP Transition Mechanisms", in: *A. Abraham, J.L. Mauri, J.F. Buford, J. Suzuki, S. M. (eds) Advances in Computing and Communications. ACC Communications in Computer and Information Science*, vol 193. Springer, Berlin, Heidelberg, 2011.

[8] S. Sasanus and K. Kaemarungsi, "Differences in bandwidth requirements of various applications due to IPv6 migration," *The International Conference on Information Network 2012*, Bali, 2012, pp. 462-467.

[9] F. N. Fatah, A.Suhendra, M. A. Marwan, and H. Firdaus, (2013). "Performance Measurements Analysis of Dual Stack IPv4-IPv6", in *Proc. of the Second Intl. Conference on Advances in Information Technology*, 2013.

[10] Cisco Systems Inc., "Cisco 4000 Series Integrated Services Routers", C78-732542-18 08/17 Data Sheet, 2017.

[11] R. Geib, Ed., and D. Black, "Diffserv-Interconnection Classes and Practice", RFC 8100, March 2017, https://tools.ietf.org/html/rfc8100

[12] T. Szigeti, C. Hattingh, R. Barton, and K. Briley Jr, (2013). *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*. 2nd ed. Cisco Press, 2013.

[13] A. Quintero, F. Sans and E. Gamess, "Performance Evaluation of IPv4/IPv6 Transition Mechanisms," *International Journal of Computer Network and Information Security*, vol. 8, (2), pp. 1, 2016.