International Core Journal of Engineering Volume 6 Issue 9, 2020
ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

Internet Protocol Version 6 Migration

Bohuai Zhao[1], Chenming Du[2], Zeqi Zhu[3], Xiaoyuan Zhu[4]

[1]College of Software, Jilin University, Changchun, Jilin 130012, China;

[2]School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 200000, China;

[3]School of Computer Science, Wuhan University, Wuhan, Hubei 430072, China;

[4]Nanjing Foreign Language School, Nanjing, Jiangsu 210046, China.

Abstract

As the migration from IPv4 to IPv6 deepens, more and more researches are being done. These researches mainly appear in the progress of the migration, the comparison between IPv6 and IPv4 in various aspects, and the data investigation of IPv6. This paper summarizes and explores four documents in these fields to draw a general conclusion about IPv6.

Keywords

IPv6, measurement, IP networks, Transitioning Mechanisms and Solutions, Fragmentation, Target Generation.

1. Introduction

IPv6 is a popular technology which receives much attention from global scholars and has been expanded to an unpredictable breadth and depth. However, it is likely for a beginner to be puzzled by numerous researches. As a result, a summarized document is highly required. Based on carefully selected and representative documents, this paper aims to meet the requirement by sorting out a concise and readable introduction of IPv6, including the history of IPv6 development, the mechanisms and approaches of transition from IPv4 to IPv6 and two possible directions awaiting future research.

1.1 Paper Structure

The paper introduces the topic of Internet Protocol Version 6 Migration, and mainly focuses on four specific parts in different sections. Section 2 talks about the past progress of the migration from IPv4 to IPv6, the stakeholders who mainly influence IPv6 adoption, and factors that affect their decisions. Section 3 specifies the mechanisms for IPv6 migration and expounds on the technology and economic factors that influence it. Section 5 takes a critical point of view, compares the IPv4 and IPv6 fragmentation technologies, and states the flaws in IPv6 fragmentation. Section 6 introduces a possible algorithmic approach to scan IPv6 address space, which is a technical gap needed to fill.

2. Migration to IPv6

This section concludes the development of IPv6 technology and its adoption among different stakeholders. Based on their respective roles and purposes, the possible factors that affect stakeholders' decisions are summarized and simple relationships that may reflect the decision process of different stakeholders based on external and internal factors are deduced.

Nowadays, with the fast-growing network industry, Internet-connected products are becoming more and more popular. In this situation, the 32 bits IPv4 addresses have now been gradually depleted,

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

which will not only increase the cost of applying for an IPv4 address but also restrain the development of new network technologies such as the Internet of Things. IPv6 is a popular technology aiming to solve the problem of an exhausted IPv4 reserve. However, due to the complexity of the network system and other technology and economic issues, which will be explained next, the adoption of IPv6 is still marginal. Analyzing factors affecting the decision process of different stakeholders is useful for decision-makers to come up with new tactics to accelerate the system's adoption of IPv6.

The internet stakeholders [1] are mainly in 4 categories: Internet technology developers (ITD), Internet service providers (ISP), Internet content providers (ICP), and Internet content consumers (users). ITDs are the main developers of new Internet technologies, including IPv6, and release new IPv6 versions for other stakeholders' further deployment. ISPs provide Internet connectivity to ICPs and users, and purchase infrastructures from ITDs ICPs produce content for users. And Internet content consumers are common users.

2.1 Adoption phases

The history of IPv6 adoption is in 3 phases:

1) Stagnation (1995-2009)

2) Emergence (2009-2011)

3) Acceleration (2011-now).

Stagnation, as the name suggests, had little or even no growth in the adoption of IPv6. In the next phase, Emergence, in which the IPv6 usage gradually increased in a very little but evident scale. Finally, Acceleration, the IPv6 usage in this phase starts to accelerate, indicating a likely complete adoption of IPv6 in the future. There are numerous technologies contributing to the performance of the IPv6 network, and it is necessary to provide an introduction to the development of some representative technologies.

The routers or switches which are able to support IPv6 packets were first introduced between 1998 and 2000. However, the early IPv6 routers had such poor performance that a study [2] in 2007 concluded that these routers were the main factor that made the IPv6 forwarding plane lagging behind the IPv4 counterpart. Nonetheless, the IPv6 routers' forwarding plane performance had become similar to IPv4 routers since 2011. Next, the operating systems seem to have a similar experience as routers; the first IPv6 offerings suffered numerous problems. Nonetheless, in 2011, almost all operating systems had IPv6 functions, and there were very few remaining problems.

From the similar experience of these three technologies, it is easy to conclude that, from ITDs' view, the stability and quality of IPv6 mainly influenced its major adoption, and the poor performance in the early years may be the cause of the original stagnation of IPv6.

Next, for ISPs, the number of IPv6-capable ASs doubled in 2009, 2011, and again in 2013. A similar trend is shown in IPv6 peering links, and these all corresponding to the acceleration phase. It seemed that though the overall adoption of IPv6 is still marginal, the previous barrier between emergence and acceleration phase seemed to have been overcome.

A group [3] started its own measurements on ICPs from 2009, tracking IPv6 accessibility across popular Web sites. From the provided forms, it can be concluded that the penetration of IPv6 among Web Sites also gradually increases with time. A significant turning point was between 2011 and 2012, with the official exhaustion of IANA's IPv4 address pool and the following World IPv6 Day, by ICPs' increasing awareness of the issue.

The focus of the next part is stakeholders' adoption decisions and factors affect them. Since users are passive in this decision making, it seems reasonable to focus on the remaining three stakeholders, who are ITDs, ISPs, and ICPs.

2.2 Factors affecting stakeholders' decisions

ITDs decision is based on the demand for their products. As a result, the stability and quality of IPv6 compared with its IPv4 substitute mainly determine how it will behave. Since they have no

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

responsibility for maintaining and upgrading infrastructures, ITDs need to consider no cost related. ISPs' decision is based on the relation between cost of upgrading infrastructures to IPv6, and the cost of procuring new public IPv4 addresses. Due to the increasing scarcity of public IPv4 addresses, the cost of applying for new IPv4 addresses seems to increase too. This growing cost is the main motivation for ISPs to make up their minds to take such an upgrade. ICPs mainly focus on accessibility and performance. ICPs' revenue is highly related to user experience, which is the connectivity quality. A larger IPv6 user base can encourage ICPs to upgrade their infrastructures, due to the potential performance loss caused by translation, a technology discussed in the next section. So ICP's decision can be considered as a model comparing the revenue gained by upgrading, which increases with the growth of the number of IPv6 users and the cost of upgrading it, which is proportional to the size of ICPs.

An introduction to the main factors that shape stakeholders' decisions and the extent to which these factors' changes affect them is in the following part.

The first factor is the demand for IPv6 technology, which had the most obvious effect on ITD. Due to their role as technology developers, ITDs were most likely to respond to or anticipate these changes. This may explain the fast development of IPv6 technology in the 1990s, and the following growth of IPv6 enabled devices. The next factor is the cost of the public IPv4 address. Since the official exhaustion of IANA's IPv4 address pool, many institutions like markets have been established for the trading of these addresses, which also implied network stakeholders that they had to pay for the previously free IPv4 resources. The main effect of this change is on the ISPs since they are the institutions that take charge of users' internet addresses. And the cost of IPv4 address is inversely proportional to ISPs' utility of IPv6. Infrastructure Upgrade Costs will be discussed next. The upgrade cost is usually levied on ISPs and ICPs since this cost is inversely proportional to the maturity of IPv6 technology, a relationship between demand for IPv6 and the investment for ITDs to maintain the quality of technology can be obtained. Similarly, for ICPs, the cost of upgrading is related to the size of the infrastructures. The main consideration of ICPs is to deliberate the cost of upgrading and current technology maturity or to delay the upgrade and perform later with a larger user base. Translation cost has marginal effect since both IPv6 and private IPv4 addresses need to be translated to enter the public IPv4 network. However, the translation cost is approximately proportional to the translation traffic. So the decision of ISPs may be affected a little bit. There is no doubt that ITDs contribution to the improvement of IPv6 technology plays an important role. However, after 2013, ITDs' technology improvement is not the only main factor that affects connectivity quality. The end-to-end connectivity is also affected by the end systems and the network. As commented earlier, the IPv6 performance is almost flawless in operating systems. The performance difference is mainly attributed to the network.

IPv6 network performance mainly depends on two factors:

1) the data plane

2) the control plane.

Data plane mostly depends on ITDs' effort, control plane's ability also related to ISPs' adoption. The group [3] measured the different performance with same or different paths destination, which shows that the data plane is not the main contributor to the performance difference (since IPv4 and IPv6 have similar performance when running the same path), instead, the control plane, the decisions making on IPv6 routing path, is the most significant factor. The different routing paths were caused by ISPs' lack of IPv6-enabled infrastructure on the original routes, which force IPv6 routing to detour from the optimized IPv4 paths. In summary, the IPv6 technology's initial lack of maturity and its poor performance contributed to its slow adoption by ISPs in the beginning, and this caused a performance gap in the control plane. Then, as the IPv6 technology reached parity with IPv4, the growth rate of IPv6 adoption seemed to increase too.

The relationship between the factors and the practical results in each phase is introduced next. First, in the stagnant phase, there was no sign of IPv4 exhaustion, the demand for IPv6 was very low, so

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

the initial investment for IPv6 to increase its quality was also low, which led to slow maturation of IPv6. There was no significant event that stimulated a fast development of IPv6, but the government policies and investment for far-sighted ITDs gradually brought IPv6 nearly on par with IPv4, which paved the way for the next emerging phase. Then, in the emerging phase, though IPv6 had reached parity with its IPv4 competitors, the ICPs and ISPs still didn't have the motivation to upgrade to IPv6, due to the minor IPv6 user base and the still available IPv4 addresses pool. However, with the gradually shrinking public IPv4 reserve and improving IPv6 performance, the IPv6 adoption still grew at a low but steady rate. At last, in the accelerating phase, a series of events greatly increased people's awareness to transfer to IPv6, including the exhaustion of IANA's public IPv4 address reserve, the World IPv6 Day, and the World IPv6 Launch. At the same time, ISPs' increasing adoption of IPv6 finally reached a level that brought IPv6 a competitive practical performance, which maintained the users who temporally turned to IPv6. In this virtuous circle, the growth rate of IPv6 finally passed its critical threshold to accelerate.

From the development of IPv6, it can be found that since the stakeholders of the network are generally utilitarian, the government's stimulation and people's rising awareness play a significant role in the penetration and final adoption of new technology. So, for the world's further development, it is reasonable not to ignore these factors.

3. IPv4 to IPv6 Transitioning Mechanisms Introduction

After the introduction of development of IPv6 technology and the adoption among different stakeholders, the main focus is turned to the transitioning mechanisms of IPv6. [4]

Nowadays, with the appearance of IPv6, many proponents regard it as IPv4's improvement instead of just replacing IPv4. However, The deployment of IPv6 must experience a process of evolving and then getting to saturation. The main issue is the deployment of IPv6 on a larger scale. That is to say. It is not clear what form the evolution will take and how the transitioning process will take. Issues surrounding the actual realization of IPv6 is the main discussion of the section.

The content is divided into three parts:

1) Current transition mechanisms and approaches

2) Economic factors affecting IPv4 to IPv6 evolution

3) Technical issues in deploying IPv6 networks

There are three forms of transitioning mechanisms: **Dual stacks, translation, and tunneling**.

3.1 Dual Stack

Dual stack is the main building block for transitioning. IPv6/IPv4 dual stacks maintain both IPv6 and IPv4 stacks, which enable IPv6 and IPv4 applications to use their corresponding stacks to process packets. Nevertheless, applications can use different versions to communicate with each other.

3.2 Translation

Translation refers to the direct conversion of packets and packet headers. It has two features:

1) It can be stateless or stateful.

2) Translation can be finished in end systems and network devices.

3.2.1 Translation Mechanisms In end systems

Stateless IP/ICMP Translation Algorithm (SIIT) is a bidirectional translation algorithm between IPv4 and IPv6 headers, which neglects extension headers in IPv6. There are two end-system translators based on SIIT: **BIS and BIA,** which stands for **Bump-In-The-Stack** and **Bump-In-The-API [5]**.

As shown in figure 1, when comparing BIS with BIA, it can be found that the translator's location is different. The translator of BIA is located in the transport layer, avoiding the conversion of IP packets in BIS. Both translators contain name resolver, the address mapper, and the translator (BIA is function mapper). Name resolver ensures that the node can only support IPv6. The address mapper provides

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

an IPv4 address for the IPv6 peer and caches the result. Translator (Function Mapper) does translation. Both translators can't solve embedded addresses.

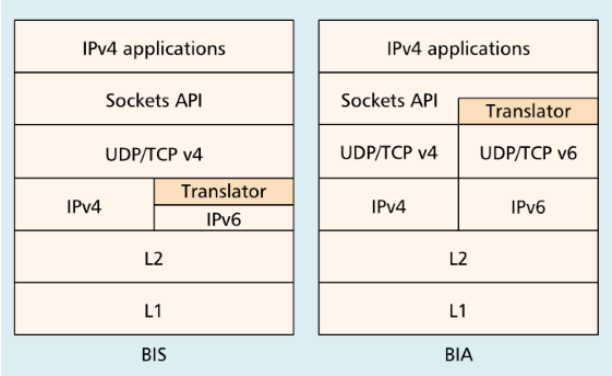

Fig.1 BIS and BIA comparison

3.2.2 Translators Used in Network Devices

The translators can be operated in both the network layer and the transport layer of the network devices. Currently the widely-used **network-layer** translators in network devices is NAT-PT.

Network Address Translation-Protocol Translation (NAT-PT) is a stateful translator that translates **unicast** addresses. It is operated based on SIIT algorithm. NAT-PT allocates an IPv4 address to each IPv6 node and acts as a proxy with IPv4 peers. Because of its statefulness, each session needs to be routed through the same NAT-PT router.

An easy example of the **transport layer** translators applied in network devices is **TRT**.

Transport relay translator [6] translates TCP/UDPv6 sessions into TCP/UDPv4 sessions. The IPv6 side will use a special destination address type to initiate a session. With routing information, the prefix will be routed to a TRT router. After that, the router will stop the IPv6 session and start an IPv4 communication with the destination.

3.3 Tunneling

From the perspective of IPv4/IPv6, tunneling means bridging incompatible networks. It can be regarded as the transfer of a payload protocol by an encapsulating carrier protocol. Tunneling has two features:

1) Both end systems and network devices can act as tunnel endpoints.

2) Tunneling can be implemented both hierarchically and sequentially.

3.3.1 IPv4/IPv6 Tunneling Mechanisms and Solutions

The best example of the mechanism in IPv4/IPv6 tunneling is DSTM, whose operation is shown in the following figure.

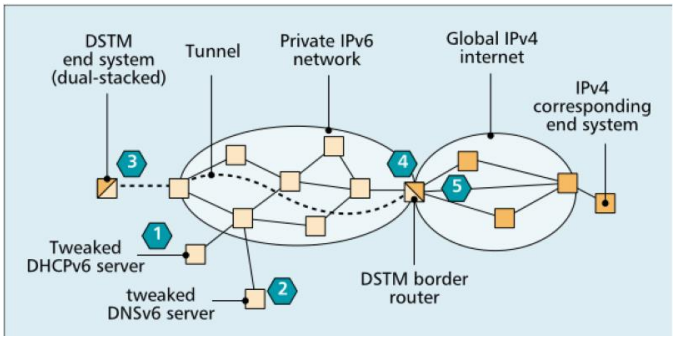

Fig. 2. Dual Stack Transition Mechanism Operation

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

Dual Stack Transition Mechanism (DSTM)[7] enables DSTM dual-stacked systems in IPv6-only networks to communicate with the IPv4 end system of the global IPv4 Internet. As shown in the figure: If the session is initiated from the DSTM end system, the DHCPv6 router will provide an IPv4 address and the border router's address to it. The DNSv6 server is responsible for providing IPv4 addresses to IPv4-only end systems. Thus, IPv4 packets can be encapsulated in IPv6 packets and forwarded to the DSTM border router. The border router can decapsulate and send the packet to the IPv4 corresponding end system of the IPv4 global Internet. Similarly, the incoming packets are tunneled from the border router to the DSTM host.

The most widely used solution in IPv4/IPv6 tunneling is **6to4 Automatic Tunneling**.

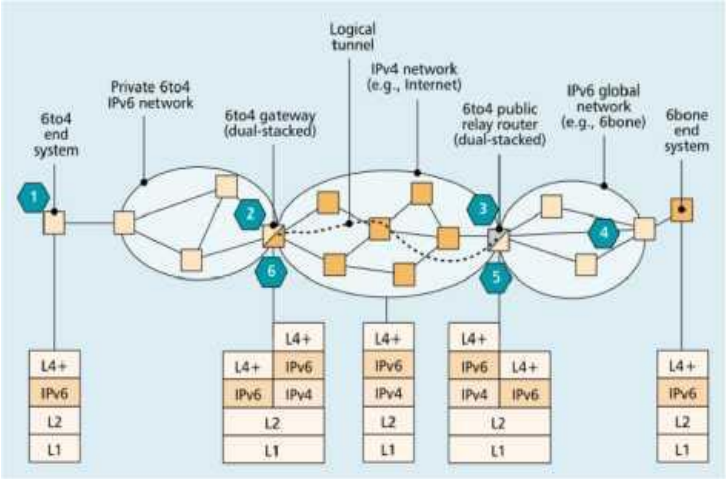

Fig. 3. 6to4 Automatic Tunneling Operation

With 'Automatic,' the configuration of tunnel endpoints doesn't need any administrators' involvement. As shown in the above picture, the 6to4 network with a special prefix is connected to the remaining IPv6 network by a 6to4 gateway and a 6to4 public relay router. The 6to4 gateway tunnels packets to the 6to4 public relay router. And the router will decapsulate the IPv4 packets and do forwarding by 6bone native IPv6 routing. In the reverse direction, the IPv6 packets will first arrive at the relay router and then the router will encapsulate them in IPv4 packets with destination from 6to4 addresses and tunnel them to the gateway using the embedded IPv4 address. Thus, the gateway will do decapsulation and do forwarding.

3.4 Economic Factors Affecting IPv4 To IPv6 Evolution

1) User Demand for IPv6: The most appealing reason for IPv6 is its new features and large address space. For end users, the feature of securing communication between applications deeply attracts people. The second reason is that people can enjoy efficient header processing and automatic configuration of computers. The last reason for its success is the adoption of many international standards and specification bodies.

2) Maintaining Legacy IPv4 Applications: Although currently some applications still need much time to move to support IPv6, even not able to support IPv6, the IPv6 versions of some IPv4 legacy applications will satisfy people's requirements. And the need of IPv6 features will fasten the speed of enabling these applications to move to IPv6.

3) Upgrading Network Infrastructure: Nowadays, IPv4 infrastructure can't handle IPv6 on both the control plane and the data plane. Because of this, the upgradation cannot be easily implemented. However, many popular routers provide IPv6 upgrades for routing software which may lead to performance degradation. Furthermore, the extra memory upgrades caused by software upgrades may increase the cost of hardware upgrades.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

4) Market Availability Of IPV6 Infrastructure: Nowadays, IPv6 support exists in many operating systems, routers, and other devices. In the future, the availability of IPv6 support will happen in programmable devices because of their flexibility of software and their hardware performance.

3.5 Technical Issues In Deploying IPv6 Networks

1) Domain Name Services: Because of the increased address size of IPv6, DNS service becomes very important. The main function of DNS is to translate domain names to IP addresses and keep these records in servers. In the era of IPv4, the way for mapping is **A-records**. When it comes to IPv6, the way becomes **AAAA** which only maps a domain name to its corresponding 128-bit addresses.

2) Routing: Nowadays, routing across the Internet is accomplished by routing protocols which are managed by **Autonomous Systems (ASs)**. Inside the autonomous system, routing tables are kept by interior gateway protocols. Outside the autonomous system, exterior gateway protocols are responsible for exchanging routing information. The routing of IPv4 and IPv6 are kept in parallel to keep IPv6 routing tables efficient. The famous interior protocols supporting IPv6 are OSPF, RIP, and IS-IS, and the popular exterior protocols are BGP4+.

3) DHCP And Address Configuration: The third key issue in deploying IPv6 is the address assignment of the end systems.

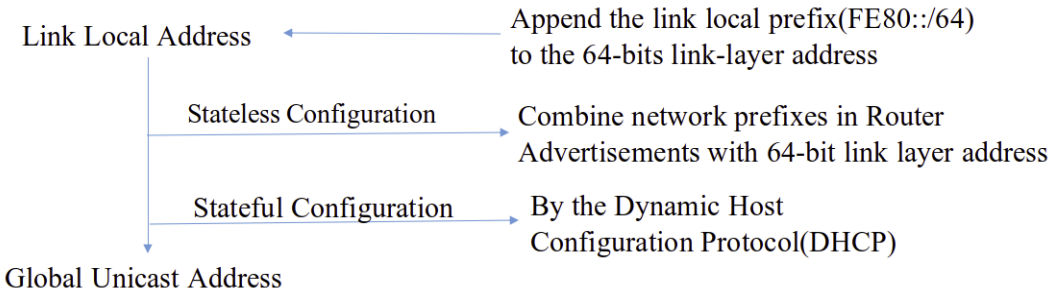

Fig. 4. IPv6 Global Unicast Address Generation

As shown in Fig.4, after obtaining the link-local address, the **allocatable** global unicast address can be got statefully or statelessly.

4) IP Security: In IPv6, it's mandated that the end systems need to support a basic level of security according to the **IPSec Internet Security Architecture** to ensure communication. IPSec provides two services: Authentication Header (AH) and the Encapsulating Security Payload (ESP). It also provides techniques for configuration and parameters security. Nevertheless, in practice, IPSec can only be applied between end systems and translation devices with the cost of additional processing.

5) Achieving Global IPv6 Connectivity: Currently, the Internet is predominantly based on IPv4. But there is no evident single global IPv6 Internet except 6Bone. There are three main approaches to providing wide-area IPv6 connectivity: **6to4 Automatic Tunnels, Configured Tunnels, and Native Connections**.

In 6to4 Automatic tunnels, only one endpoint of 6to4 Automatic Tunnels must be configured. But they are not reliable and they may cause poor QoS because of the uncontrolled load.

In configured tunnels, both endpoints must be configured and the tunnels should be made shorter. However, it can ensure better QoS and it's more strictly controlled. So they are more difficult to manage.

Native connection provides a direct connection to the nearby IPv6 network which often leads to better performance compared with the previous two approaches

All in all, nowadays, because of the new features of IPv6, people's acceptance of IPv6 is rapidly on the increase. And it seems that more and more organizations are now supporting IPv6 versions of

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

their products which only supports IPv4. IETF also provides more and more mechanisms and solutions to help us migrate to IPv6. But the progress in really taking up IPv6 will be much slower than expectation. The reason is that just like what this section has analyzed, the deployment of IPv6 needs to have a deeper consideration. The deployment of IPv6 needs a process of evolving and then achieving global saturation. Until the address space of IPv4 really is not sufficient or the demand for better security and better quality of service becomes very significant, IPv6 technology is still a luxury for people. It needs a phase in order to really migrate to IPv6.

However, there is only one thing need to be kept in mind: The era of IPv6 will eventually come into our lives. And people can enjoy the precious treasures provided by IPv6 technology. After the deployment of IPv6 is deeply arranged and people gain a full understanding of IPv6 technology, it will maybe be very popular and move from dream to reality.

4. Fragmentation

Both IPv6 and IPv4 have their own ways of dealing with packet fragmentation. Also, one of the most important parts in the designation of network protocol is packet fragmentation.

4.1 Usage of Fragmentation

Not all link-layer protocols can carry network layer packets of the same size. Some of them, such as Ethernet frames, are larger while the other such as wide-area links are smaller. That causes differences in the maximum transmission unit (MTU). A datagram can not be transmitted to the end if one or more links in the whole way have a smaller MTU than the sized datagram. And in order to balance the difference, here comes the fragmentation.

4.2 Fragmentation in IPv4

First of all, talk about what the fragmentation likes in IPv4. Routers may sometimes face a problem that datagram length from the former link is larger than the MTU of the next link. The datagram can be divided into several parts in the routers so that each of them can adjust the size limit of the following link. That is called "forward fragmentation". Fig. 5 illustrates how it works.

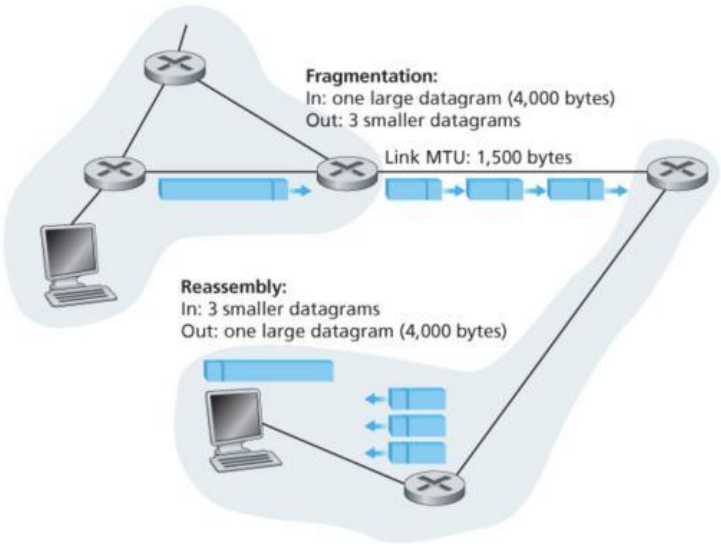

Fig. 5. Fragmentation in IPv4[8]

The fragmentation can be accomplished by routers. The destination host will reassemble them. All of these are done in the network layer. The upper layers even know nothing about the fragments. In the beginning, the upper layers send a whole segment and receive the segment in the end. But, IPv6 changes nearly all of these.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

4.3 Fragmentation in IPv6

The article pointed out that packet fragmentation can do harm to efficiency, security, and pose a cap on maximal delay bandwidth [9]. So, The IPv6 designers removed the fragmentation controls from the common IPv4 packet header and placed them into an 8-octet IPv6 Extension Header. This additional packet header, situated between the IPv6 packet header and the end-to-end transport packet header, was present only in fragmented packets [9].

IPv6 does not allow for fragmentation and reassembly at intermediate routers. These operations can be performed only by the source and destination. If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram and sends a "Packet Too Big" ICMP error message back to the sender. The sender can then resend the data, using a smaller IP datagram size.

Fragmentation and reassembly is a time-consuming operation; removing this functionality from the routers and placing it squarely in the end systems considerably speeds up IP forwarding within the network [8].

4.3.1 TCP-IPv6

If the packet is too large to transmit, the host may be permitted to send an IPv6 packet too big ICMPv6 diagnostic message in order to update the MSS (maximum segment size) to a new smaller one.In that case, fragmentation will be avoided.

4.3.2 UDP-IPv6

And for UDP it is more difficult. It relies on upper level application protocol. That means only UDP can be considered.

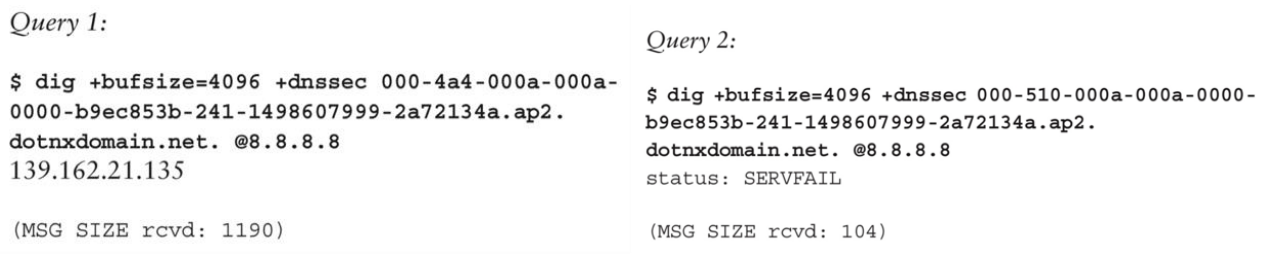

Fig. 6. Query [9]

So, the author found DNS as an example of application protocol, because DNS is the major user of UDP. He tried to send the two DNS queries in Fig.6. Both of them use UDP over IPv6. They are almost the same and are aiming at Google's Public DNS service. But the second response is larger than the maximum packet size. As shown in Fig. 6, the message size received is 104. The packet has been divided into two parts. As a result, the status is SERVFAIL.

4.4 Firewall

Fragmentation can also raise the eyebrows of a firewall. The fragmentation only copies the packet header but not copies the transport protocol. This brings difficulties to firewalls because only the first fragment can be examined for its correct form.

The 8-octet extension header is mentioned. In the first packet of a set of fragmented packets, the Upper-Level Protocol Header is chained off the fragmentation header, at byte offset 48, assuming that there are no often Fragmentation Extension Headers in the packet. Here comes the problem: the device needs time to deal with the chain, and the extension may not be recognized.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

4.5 Dropping packet

4.5.1 UDP

According to the problem, maybe dropping all IPv6 packets that contain extension headers is a good idea. Here is a report about it. "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World". This document reports a 38% packet-drop rate when sending fragmented IPv6 query packets to DNS Name servers in Google's Public DNS resolvers [10]. That proves the packet drop. The author gave an experiment to find whether all of the resolvers are facing the same problem. And the result is 37%, almost 38% in the mentioned report. So, the answer is yes.

4.5.2 TCP

What about it in TCP? In other to find that, an experiment is used to do a fragmentation test. All TCP packets passed across the unit from the back end towards the internet that contains a TCP payload larger than 15 octets are fragmented.

Table 1[9]: Results of Fragmentation Test

	Count	% of Total
Sent Fragmented TCP Packets	1,675,898	
Acknowledged Fragmented TCP Packets	1,324,834	79.03%
Failed to Acknowledge Fragmented TCP Packets	351,514	20.97%

In Table 1, the result was 20% of total failed to acknowledge the fragmented TCP packet. It is much better than 37% in UDP. However, that's not enough. That means there is a packet-drop in every five packets. It's still a serious problem [9].

Teredo and 6to4 are two auto-tunneling techniques. The result in Table 2 shows that most of the teredo endpoints can not handle IPv6 fragmentation successfully. The result for 6to4 was a little better, 6.1%. The author pointed out that "In some cases, it appears that the choice of customer premises equipment, or the configuration of IPv6 firewalls, may be a factor."[9]

Table 2[9]: Results of IPv6 Fragmentation Test for Teredo and 6to4 Prefixes

	Teredo	%	6to4	%
Sent Fragmented TCP Packets	53,780		24,384	
Acknowledged Fragmented TCP Packets	263	0.5%	1,486	6.1%
Failed to Acknowledge Fragmented TCP Packets	53,517	98.5%	22,898	93.9%

4.5.3 Teredo and 6to4

Obviously, the IPv6 fragmentation is not satisfiable: IPv6 fragmentation is just not a viable component of the IPv6 internet and was that needs to adjust the protocols to avoid fragmentation.

This section is mainly aiming at UDP. According to the ICMPv6, TCP is not an important problem. However, as for UDP, the combination of DNSSEC UDP and IPv6 is really not going to work well.

In the end, a quick UDP internet connections protocol is mentioned. The protocol uses a maximum packet size of 1350 octets. The size is considered to be a conservative choice. And if DNS over IPv6 uses a packet size that is similar to this, many of the packet-loss problems can be avoided.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

5. Target Generation

With the continuous development of network speed and computer hardware, people can already scan IPv4 networks with certain tools in a short time, and ensure very high accuracy. For example, the ZMap[11] tool uses the density and limited size of the IPv4 space to traverse all IPv4 networks through high-speed scanning. On a typical desktop computer with a gigabit Ethernet connection, ZMap is capable of scanning the entire public IPv4 address space in under 45 minutes. However, IPv6 has brought the Emergence of new problems. Since IPv6 has 128 bits, compared to 32-bit IPv4, the size of IPv6 is 2^{96} times that of IPv4, which makes brute force traversing the IPv6 address space seems unrealistic now. Therefore, some effective methods to explore the IPv6 address space is required.

Now, one of the more mainstream methods to explore the IPv6 address space is to use the IPv6 seed set to generate scan targets and is called the target generation algorithm. This algorithm mainly guesses the potential target address outside the seed set based on the input seed set.

First of all, it is necessary to know some simple preliminary knowledge.

5.1 IPv6 address representation

IPv6 has 128 bits, and it is usually divided into eight groups. Besides, each group has 16 bits, which are divided into four parts, and each part has 4 bits, which represent a hexadecimal number. Colon notation (":") is often used to divide each group, and each hexadecimal number referred to as nybble. For example, in IPv6 address 2001 : 0acb: 0000: 0000: 0000: 0000: 0054: 2137,7 is a nybble, and there are four nybbles between each colon. In addition, since IPv6 addresses usually contain many nybbles with zeros, the leading zeros is usually omitted and the longest sequence of all zeros with double colon notation ('::') is replaced: this is called as a compressed representation. Therefore, the previous IPv6 address can be compressed with 2001: acb::54: 2137, and the middle four groups of 0 are all replaced by double colon notation. What's more, wildcards ("?") can be used instead of a nybble to represent a dynamic address range, for example, 2 001: acb::54: 213? Represents 16 addresses, from 2001: acb:: 54: 2130 to 2001: acb:: 54: 213f.

5.2 Seed selection and analysis

TGA usually requires a few known address seeds which are used for excavation, thus revealing how the IPv6 address is assigned. With the results, the addresses of other hosts to be scanned can be predicted. In addition, a specific port can be scanned to detect a certain protocol through a wide range of network scans to find a specific type of host address for analysis. For example, we can scan the host that responds on TCP / 443, and then test the characteristics of HTTPS. Using these specific additional data, TGA can generate target addresses for specific protocols. For example, target addresses that support TCP / 443 and TCP / 80 can be generated and the IP of hosts that support TCP can be better predicted.

The intention of TGA is to generate additional candidate addresses by analyzing the structure of known addresses. Generally speaking, an address model is required to be built first, and there are two types of address models, independent and dependent. The independent model means that each address seed in the model is an independent and identically distributed random sample. Observing a specific seed will not affect other address seeds. In this model, the area with the highest active address density in the address space is positively correlated with the seed density, which means that the higher the active address density, the higher the seed density. The dependent model means that in this model, there is an explicit or invisible dependency relationship between each seed address. Observing a specific seed may affect the probability that other seed addresses are active. In this model, TGA can use the dependencies between these seeds to find more reliable seeds.

The dependent seed model can be imagined as a pattern recognition method. Modeling the dependent seeds can lead to a more effective TGA, but it also has its limitations because it requires that the data set must have some inherent Dependence. Otherwise, the effect will become very poor. In contrast,

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

although the independent seed model leads to a less systematic approach, it also has independence and simplicity, which make it can predict in any case. Besides, in terms of calculation, as it does not need pattern recognition and other methods, it also does not require any learning process so that it can improve the efficiency of calculation.

In order to conduct reasonable data modeling, specific situations need to be analyzed. Because the seed set is just a tiny part of the actual address, it can be discussed in two cases: if the seed set collected is an aggregated part of a certain address, we can build a dependency model and use the pattern recognition method to carry out the learning process; if the seed set is a discrete part of the actual address, it may be better for us to build an independent model because each part of the seed set does not have too many dependencies which mean it is difficult to use a learning model to fit inference.

5.3 6Gen algorithm

Murdock et al. [12] developed an algorithm called 6Gen to identify dense areas of similar seeds. They assume an independent seed model that has no dependencies between each seed, so although it does not have a learning process, with the flexibility and simplicity. The 6Gen algorithm utilizes the pattern recognition method (like clustering algorithm) to generate the target address. It firstly separates each seed into sets, then identifies the most similar seeds and assigns the seeds to a specific seed set. Ultimately, it iterates until the size of the aggregation area exceeds the given scan budget or a cluster that meets the requirements. However, 6Gen is not entirely focused on seed density because it first considers the similarity between various sub-units rather than the density region. The reason for this consideration is to save the budget and improve the efficiency of the algorithm.

In order to aggregate similar addresses, a standard is needed to define the similarity of addresses. They use the Hamming distance [13] to define the similarity. The criterion is to compare the number of different nybble positions between two addresses, and if it appears Wildcard ('?') between the corresponding position of two addresses, the distance is 0. For example, the distance between 2001: acb::54: 2137 and 2001: acb:: 54: 213? is 0 ; the distance between 2001: acb:: 54: 2137 and 2001: acb:: 54: 2134 is 1; the distance between 2001: acb:: 54: 2137 and 2011 : acb:: 34: 2137 is 2. In turn, a group of seed addresses can also be represented with the same characteristics as a range, as shown in Figure.7. For example, 2001:acb::5?:213? can be used to represent the range of 2001:acb::54:2137, 2001:acb::54:2135 and 2001:acb::51:2136.

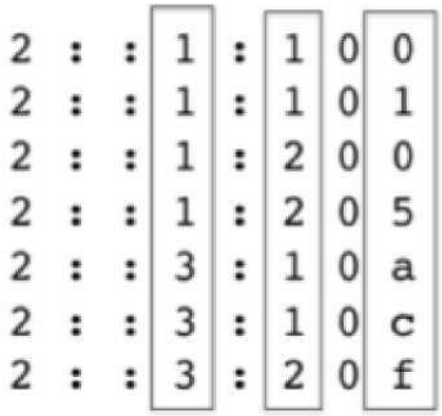

Fig. 7. Dynamic nybbles for a cluster of 7seeds (the cluster's seed set). The cluster has three dynamic nybbles (the other 29 nybble indices have identical values for all addresses) and a range of 2::?:?0?

5.4 Annotation of 6Gen

Function InitClusters is used to initialize clusters, generate a cluster for each seed separately, and then add all clusters to the cluster set.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

```
function InitClusters(seedList)
    for seed in seedList do
        cluster = new Cluster()
        cluster.addSeedUpdateRange(seed)
        clusterList.add(cluster)

function FindCandidateSeeds(cluster, seedList) ▷ Computes the minimum Hamming distance between cluster.range and all seeds in seedList not already in cluster, and returns the list of seeds that are this minimum distance away.

function GrowCluster(seedList)  ▷ Consider growing all clusters by candidate seeds, and select the growth resulting in the highest seed density and smallest cluster range size.
    maxDensity, maxIndex, maxRangeSize = 0, 0, Infinity
    maxCluster = None
    for index in [0, ..., clusterList.length() − 1] do
        cluster = clusterList[index]
        candidateSeeds = FindCandidateSeeds(cluster, seedList)
        for seed in candidateSeeds do
            tmpCluster = cluster.copy()
            tmpCluster.addSeedUpdateRange(seed)
            for otherSeed in candidateSeeds do
                if otherSeed in tmpCluster.range then
                    tmpCluster.addSeedUpdateRange(otherSeed)
▷ Does not further change the range.
            newDensity = tmpCluster.seedSet.size() / tmpCluster.range.size()
            if (newDensity > maxDensity) or (newDensity == maxDensity and tmpCluster.range.size() < maxRangeSize) then
                maxDensity, maxIndex = newDensity, index
                maxRangeSize = tmpCluster.range.size()
                maxCluster = tmpCluster
    return (maxIndex, maxCluster)

function 6Gen(seedList, budgetLimit)  ▷ Grow clusters until the sum of cluster range sizes exceeds the budget. For simplicity, we elide here details about handling cluster overlap and final cluster growth sampling to use up the budget exactly.
    InitCluster(seedList)
    budgetUsed = 0
    while True do
        grownIndex, grownCluster = GrowCluster(seedList)
        oldRangeSize = clusterList[grownIndex].range.size()
        newRangeSize = grownCluster.range.size()
        budgetCost = newRangeSize − oldRangeSize
        budgetUsed = budgetUsed + budgetCost
        if (budgetUsed ≤ budgetLimit) and (seedList.size() > grownCluster.seedSet.size()) then
            clusterList[grownIndex] = grownCluster
        else
            return clusterList
```

Fig.8: 6Gen pseudocode, simplified to illustrate conceptual steps [12]

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

Function FindCandidateSeeds is used to calculate the minimum Hamming distance from each seed to each cluster, and then return the seeds of the minimum Hamming distance.

Function GrowCluster firstly traverses all the clusters, and then add the satisfying seed set returned by function FindCandidateSeeds to the currently taken clusters, finally calculating the new density size. If certain conditions are met, the maximum density, maximum index, maximum range, and maximum cluster are generated, and finally, return the maximum index and maximum cluster.

Function 6Gen first, call the function InitClusters to initialize. Then when the budget is met, continuously call the function GrowCluster to generate clusters until the budget or seed set exceeds the given size, and eventually returns to the cluster set.

Although this algorithm will cause some clusters to overlap, (that is, a seed belongs to both A cluster and B cluster), Murdock et al. do not try to simply merge these partially overlapping clusters because this may cause the cluster density greatly reduced. Instead, they allow a partial overlap of clusters but delete clusters that are completely contained and belong to a subset of other clusters. The whole pseudocode is shown in **Fig.8**.

5.5 Evaluation of 6Gen

Murdock et al. use a set of 2.96 M seeds on a Linux server to evaluate 6gen, and the server has dual 10-core Intel Xeon E5-2650 (2.30 GHz) CPU and 256GB of memory. They grouped the seeds by routing network prefixes and ran 6Gen on each seed to test 10038 prefixes. Figure.9 shows the median running time of different seed numbers. Wall Clock time represents the actual running time of 6Gen. It can be found that as the number of seeds increases, both running times become longer, from 10 to 10000 Within this seed number interval, the growth of running time is relatively slow, but when it reaches the order of 1M, the time overhead becomes relatively large.

5.6 Seed sensitivity

As discussed before, for TGA, the selection of seeds has an important impact on the algorithm. Next, the influence of seed type and seed quantity on 6Gen is considered. Murdock et al. performed only on a DNS name server seed by running 6Gen to detect whether it may find the corresponding host of other types of seeds (Web and SMTP, etc.). They chose 61,000 DNS seed, and scanned the

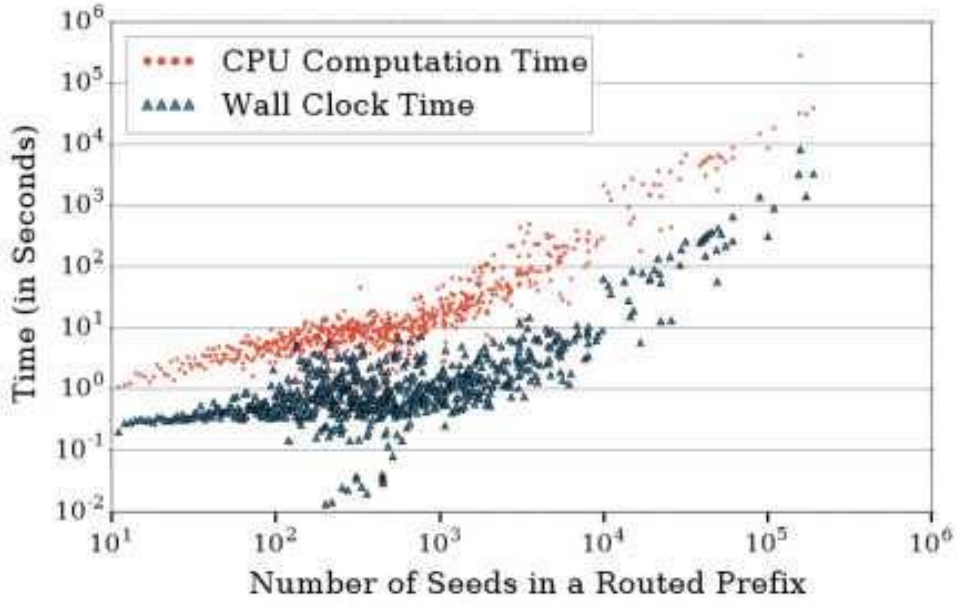

Fig. 9. Median execution time of running 6Gen on routed prefixes with differing numbers of seeds. Routed prefixes with fewer than 10 seeds are omitted as they all required less than one second to process.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

Table 3. Downsampling level in different cases

Downsampling level	W/o Dealiasing		W/ Dealiasing	
	Num. Hits	% vs AII	Num. Hits	% vs AII
1%	758K	1.3%	225K	22.5%
10%	13.3M	23.5%	713K	71.3%
25%	27.3M	48.2%	825K	82.5%
100%	56.7M	100.0%	1.0M	100.0%

In this table, for each downsampling level, both before and after dealiasing, we list the number of hits 6Gen discovered and what percentage that number is compared to the number of hits 6Gen found using all seeds. In all experiments, the budget was 1M probes per routed prefix.

TCP / 80 outputs of prediction. They eventually found 1.2 M target addresses, which indicates that a single type of seed can be used to detect other types of addresses. Then, they studied the detection situation of 6Gen algorithm in part of the data set, which can improve performance. As shown in Table.3, they tested running 6Gen on 1%, 10%, and 25% of the completed data set. It can be seen from the table that in the case of low data volume, 6Gen can hit almost double the data set; more surprisingly, after de-aliasing, 10% of seed can hit 71.3% of the whole set, which shows that 6Gen in low data volume case can also have good efficiency.

The 6Gen algorithm has great potential. It can detect many target addresses with a small amount of data, and it can also find other types of addresses with a single data source. However, 6Gen also has some improvements, such as further improving the algorithm's efficiency, becoming more intelligent, and strengthening the processing of aliasing regions. However, no matter how the algorithm progresses, a high-quality address seed set is an essential prerequisite for IPv6 address discovery. Only in this way can the target address set generated be rich and accurate. Of course, the target address generated should be scanned and collected repeatedly. In short, IPv6 address mapping is difficult to do once and for all, and only through the continuous operation of the address set algorithm can more and more live addresses be accumulated.

6. Conclusion

In this work, the migration process of IPv6 from the background requirements of the migration to the actual application plan is explored.

In addition, how to make the migration from IPv4 to IPv6, working through three phases, and exploring the actual cost are described. Due to the different structure and size of IPv4 and IPv6, the fragmentation method of packets is also different, and there are many problems in that region. What's more, the differences also caused us to be unable to treat IPv6 with some of the methods and tools previously used to treat IPv4. In the address scanning section, some new TGA algorithms are introduced to try to solve the problems on IPv6.

In a word, the transition from IPv4 to IPv6 is still an intermediate process that is not over yet, and there are still many problems to be solved. But it is believed that as technology advances, IPv6 will one day replace the position of IPv4.

Acknowledgments

We would like to express gratitude to our supervisor, Bill Nace, for his constance guidance and inspiring advice.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202009_6(9).0022

Also, all of the teammates have put considerable time and effort on the work. Chenming Du works hard and conscientiously, and finishes the task in the shortest time without delay. Zeqi Zhu has an active mind and always comes up with great ideas, which makes the team think outside the box. Xiaoyuan Zhu likes to try new things and is never afraid of challenges.

References

[1] C. Kalogiros et al., “Final report on economic future Internet coordination activities,” Tech. Rep. D2.2-v2.0.doc, 2012.

[2] X. Zhou, M. Jacobsson, H. Uijterwaal, and P. Van Mieghem, “IPv6 delay and loss performance evolution,” Int. J. Commun. Sys., vol. 21, no. 6, pp. 643-663, 2008.

[3] M. Nikkhah and R. Guerin. “Migrating the Internet to IPv6: An Exploration of the When and Why". In: IEEE/ACM Transactions on Networking 24.4 (Aug. 2016), pp. 2291-2304. doi: 10.1109/TNET.2015. 2453338.

[4] D. G. Waddington and Fangzhe Chang. “Realizing the transition to IPv6". In: IEEE Communications Magazine 40.6 (June 2002), pp. 138-148. doi: 10.1109/MCOM. 2002.1007420.

[5] S. Lee et al., “Dual Stack Hosts Using Bump-in-the-API(BIA),” IETF draft, draft-sylee-bia-00.txt, Feb. 2001.

[6] J. Hagino and K. Yamamoto, “An IPv6-to-IPv4 Transport Relay Translator (TRT),” IETF draft, draft-ietf-ngtranstcpudp-relay-03.txt, Apr. 2001.

[7] J. Bound et al., “Dual Stack Transition Mechanism (DSTM),” IETF draft, draft-ietf-ngtrans-dstm-04.txt, Feb. 2001.

[8] Keith W. Ross James F. Kurose, Computer Networking: a top-down approach/ James F.Kurose, University of Massachusetts, Amherst, Keith W. Ross, NYU and NYU Shanghai, PEARSON, 2017. ISBN: 0133594149.

[9] Geoff Huston. “IPv6 and Packet Fragmentation". In: Internet Protocol Journal 21.1 (Apr. 2018), pp. 13-23.

[10] Fernando Gont, J. Linkova, and Tim Chown, “Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World,” RFC 7872, June 2016.

[11] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-Wide Scanning and Its Security Applications. InUsenix Security,2013

[12] Austin Murdock et al. “Target Generation for Internet wide IPv6 Scanning". In: IMC '17 (2017), pp. 242-253. DOI: 10.1145/3131365.3131405. URL: http://doi.acm.org/10.1145/3131365.3131405.

[13] R.W .Hamming. Error Detecting and Error Correcting Codes. Bell Labs Technical Journal,1950.