UNIVERSITY OF TWENTE

Faculty of Electrical Engineering, Mathematics and Computer Science (EWI)

Bachelor assignment Telematics

Impact of IPv6 on WiFi-based networks

Robin F. Pronk
s0138746

Committee:
Dr. ir. P.T. de Boer
Dr. ir. G.J. Heijenk

July 4, 2016

UNIVERSITEIT TWENTE.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 The Internet

For many people the internet is a wonderful place and a technology which they use for many hours a day. One of the core protocols of the internet, and the name is quite on the spot, the Internet Protocol. In particular IPv4, version 4, forms a very important component of the internet as we know it nowadays. The Internet Protocol is the basic communication protocol of both almost all local (eg. home or office environment) networks (LAN) and the internet. The protocol basically handles the addressing of data packets so that it can reach the right destination. The Internet Protocol (IP) is tightly coupled with the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). These are placed on top of IP to provide higher layer services like message segmentation and in case of TCP in order delivery guarantees.

The address size used in IPv4 is 32 bit, offering roughly (2 to the power of 32) 4.294.967.296 unique addresses. Although 4 billion unique addresses seems a lot and really did when the protocol was designed in the early 1970s it is not that much if you consider how many people are connected to the internet and often not limited by just one device. The address space of IPv4 gets or even already is exhausted. The next version of the Internet Protocol in line is version 6: IPv6. Instead of 32 bits it uses 128 bits to store addresses. This gives the option for much more unique addresses and thereby more devices connected to the internet. More details will covered in later sections.

Over the past decade(s), the world has become increasingly mobile. Traditional ways of networking have been proven inadequate to meet the challenges posed by our new collective lifestyle. If users must be connected to a network by physical cables, their movement is dramatically reduced. Wireless connectivity, however, poses no such restriction and allows far more freedom of movement on the part of the network user. Not so many years ago you had to use a dial-up modem in order to connect to the internet. Internet connections have evolved extended since the dial-up modem. In this extend most people nowadays associate the word “WiFi” with wireless internet instead of the name it actually is of the most commonly used technology for wireless networking. A wireless network connection that provides always on internet connectivity has become so common for (western) societies that a technology name implies it.

Considering above statements it is not very strange to combine IPv6 with WiFi. However is this as straight forward as it sounds and what are the implications? In this report I will try to assess this.

1.2 Research

The goal of my research and report is to evaluate the impact and possibilities of IPv6 when deploying on a WiFi-based wireless network.
This leads to the main research question:

What is the impact of IPv6 on a WiFi-based network?

This main research question is too broad for the actual research work and needs refinement, for which some ground work and assumptions are required. In order to further specify my research first some background information is needed, which is given in chapter 2. In addition preliminary observations will help to understand the scope of the question and determine potential issues. Combining this information will help to narrow down the research question in chapter 3.

Motivation Since my second year at the University of Twente I have been active in the student association called Studenten Net Twente, which is mainly a association of students fond of computer networking. SNT enjoys and benefits from her close cooperation with ICTS, the IT department of the University of Twente. During my broad range of activities I saw many aspects of the University network and came into contact with interesting people working on and with it. In the IT networking world it has been a known fact for many years now that the need for IPv6 to replace IPv4 will become larger and larger. On the other hand it seems a sidelined technology as for years there hasn't been much commercial activity around it. In 2010 we formed a team of six SNT members and participated in the Dutch IPv6 awards challenge. We pointed our focus onto IPv6 tunnel mechanisms to ease the transitions from IPv4 to its successor Our effort was rewarded with a 15000 euro price.[5] Not the money, which we kept in the association, but the activity itself and feeling around it kept me interested in IPv6 developments. First attempts to implement IPv6 on the campus wireless network failed, as it resulted in a, for both IPv4 and IPv6, painfully slow network. I wanted to know why exactly and what could be done about it. In combination with the growing usage of wireless networks I started this research. During my time at the university and with SNT I gained relevant social connections and combined with the fantastic Campusnet network it helped me in "playing" around with IPv6 and allowed me to perform experiments.

Structure After this introduction chapter I will, as a starting point for this report, first discuss some background information about the most important internet protocols in chapter 2. An introduction to IPv6, including its most important aspects, followed by equal information about WiFi is also covered in this chapter. This will draw the area of interest of this report and gather knowledge to achieve the stated goals.

Chapter 3 will further define my research as more information unraveled. Desired information following from this chapter will be covered in the next two chapters; chapters 4 and 5.

Chapter 6 will describe the experiments I have performed to further achieve my goals. Leading to proposed solutions in chapter 7.

To extend my research two chapters will dive into additional aspects and features, MobileIPv6 in chapter 9 and security aspects and vulnerabilities in chapter 10.

After the conclusions in chapter 11 I will point out some interesting points for future work in chapter 12.

Acknowledgements First I want to thank my supervisor Pieter-Tjerk de Boer for his supervision, including pointing out different perspectives, and his continuous support. I would also like to thank Jeroen van Ingen and Jan Freek Popma, network administrators of the University of Twente. They both have been a sparring partner and supplied me with some testing facilities and information especially in the first sage of my research.

CHAPTER 2

BACKGROUND INFORMATION

2.1 Internet protocols

Most people on the world know about the internet. However it is hard to say what it exactly is or includes as it is complex and ever changing. The public internet is a worldwide computer network, that is, a network that interconnects millions of computing devices throughout the world. Not too long ago, these computing devices were primarily traditional desktop computers and servers. While the internet has grown much since it contains all kind of devices, from mobile computers and cell phones to televisions and even refrigerators.[28]

The internet offers a divers range of services to its users. The technical solution behind can be split into layers, each with its own responsibilities. The OSI model is widely used model for these layers. Figure 2.1 shows the OSI model with a short description of the layers.[26] Networking protocols are always bound to one ore more layers. Especially in the upper layers there are many different protocols. Probably the most known protocol is HTTP, used for serving websites, which is found at the application (top) layer.

This report is primarily about IPv6, found in the transport and network layer, and WiFi, found in the physical and datalink layer. To be more precise: IPv6 consists of IP (network layer) and **T**ransmission **C**ontrol **P**rotocol and **U**ser **U**atagram **P**rotocol (transport layer). WiFi has a physical (PHY) part and splits the datalink layer into its **L**ogical **L**ink **C**ontrol and **M**ultiple **A**ccess **C**ontrol parts.[14] Of course this will be covered in more detail in respectively chapter 2.2 and 2.3.

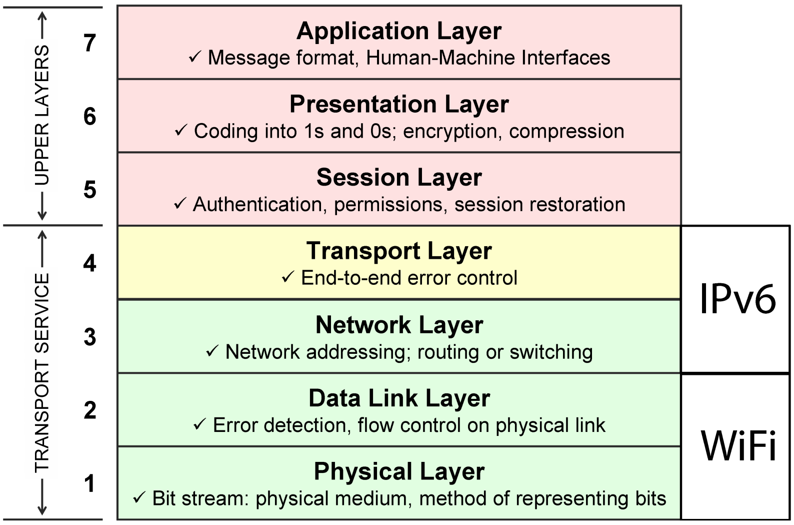

UPPER LAYERS
TRANSPORT SERVICE
7
Application Layer
✓ Message format, Human-Machine Interfaces
6
Presentation Layer
✓ Coding into 1s and 0s; encryption, compression
5
Session Layer
✓ Authentication, permissions, session restoration
4
Transport Layer
✓ End-to-end error control
3
Network Layer
✓ Network addressing; routing or switching
2
Data Link Layer
✓ Error detection, flow control on physical link
1
Physical Layer
✓ Bit stream: physical medium, method of representing bits
IPv6
WiFi

Figure 2.1: The OSI model

2.2 IPv6

2.2.1 History

IPv4 was developed in the early 1970s to facilitate communication between computers of government researchers and academics. At that time the network was closed and had only a number of access locations. Although the developers of IPv4 never worked with the requirement, and probably not even the idea, for general world wide deployment the protocol survived for decades. It has been an integral part of the internet revolution.[17]

The Internet Engineering Task Force began their effort to develop a successor for IPv4 in the early 1990s as they started to see the limitations of version four. IPv6 has been developed based on the rich experience gained by the global adaption of its predecessor. Proven and established mechanisms have been retained and known limitations have been tried to overcome. The main protocol design goals were to handle the growth rate of the internet and to cope with the demanding requirements on services, mobility and end-to-end security. [17]

The first draft standard for IPv6 was submitted back in 1995[6] and was finalized in 1998[7]. That is over 15 years ago and still IPv6 deployment and adoption is far from completion as we will discuss in a next section.

Some people might ask, where is version five? IPv5 was used to define an experimental real-time streaming protocol. To avoid any confusion, it was decided not to use IPv5 and to name the new IP protocol IPv6.

2.2.2 What is IPv6

IPv6 is an evolution of IPv4. It shares many mechanisms and principles and is designed to overcome the limitations of IPv4.[17] To simplify adaption both IPv4 and IPv6 can work side to side in a network. An ideal one day switch to the new protocol would be great, however almost impossible due to the immense wide deployment of IPv4 in software as well hardware.

2.2.3 What is new

Address space The first and probably most obvious change is the extended address space. The amount of bits allocated to store an IP address have been increased from 32 in IPv6 to 128 in IPv6. This drastically expands the address space. Only half of the bits (64) are used for the network subnet and the other half for the interface ID. A basic comparison between IP addresses are shown in table 2.1 and prefix reservations in table 2.2. As seen in the address example there is a short notation in IPv6 which require less characters for an address. Leading zeros can be skipped and "::" completes to the right amount of zeros in between.[31] The current world population exceeds 7 billion people. So even if it were possible to use 100 percent of the IPv4 address space, we would not be able to provide an IP address for everyone on the planet.[17] In contrast the address space of IPv6 is enough to provide every sand grain on the planet with an address.

In general an IPv6 node will have at least two addresses: one with a link local scope, used only within the local network, and one with a global scope. Most desktop operating systems will also enabled Privacy Extensions default. When Privacy Extensions are enabled the node generates additional temporary addresses on the advertised network prefix using stateless autoconfiguration. These ephemeral addresses are used to communicate with remote hosts making it more difficult to track a single device by its IP address.[32]

Table 2.1 IP address comparison

	IPv4	IPv6
Address bits	32	128
Subnet / host bits	8 - 30 / 24 - 2	64 / 64
Unique addresses	2^{32} = 4294967296	2^{128} = 3,402823669 * 10^{38}
	4 billion	a lot!
Example IP	130.89.3.249	full: 2001:067c:2564:a102:0000:0000:0001:0001
(www.utwente.nl)		short: 2001:67c:2564:a102::1:1

Table 2.2 Address space assigned prefixes[17]

Allocation	Prefix binary	Prefix hex	Fraction of address space
Unassigned	0000 0000	::0/8	1/256
Reserved	0000 001		1/128
Global unicast	001	2000::/3	1/8
Link-local unicast	1111 1110 10	FE80::/10	1/1024
Reserved*	1111 1110 11	FEC0::/10	1/1024
Local IPv6 address	1111 110	FC00::/7	1/128
Private administration	1111 1101	FD00::/8	1/256
Multicast	1111 1111	FF00::/8	1/256
Documentation**		2001:db8::/32	
Global unicast***		2000::/3	1/8

* Deprecated, formerly Site-local multicast

** No coincidence to see this subnet so often in documentation

*** Currently being delegated by the IANA, might change in the future[9]

This table does not show all assigned prefixes

Simplification of header format Compared to its predecessor IPv6 uses a simpler header format, see figure 2.2. It uses two times 16 bytes for the source and destination addresses and 8 additional bytes for general header information. Fields like header length, fragmentation, option or checksum have been removed. The simplicity and 40 bytes fixed length allows for faster processing.[2] So called extension headers are introduced and can be used to extend the main header for additional protocol options. Each header, both main and extension, contains a next header field specifying the type of the next header when applicable. The main header has to be understood by all hops on the path for a successful transfer, but an extension header will just be ignored and passed on if an intermediate hop can not parse the extra header information. This principle introduces flexibility while maintaining fast processing on intermediate routers. The base specification includes a set of six extension headers, including headers for routing, Mobile IPv6, and quality of service and security.[17]

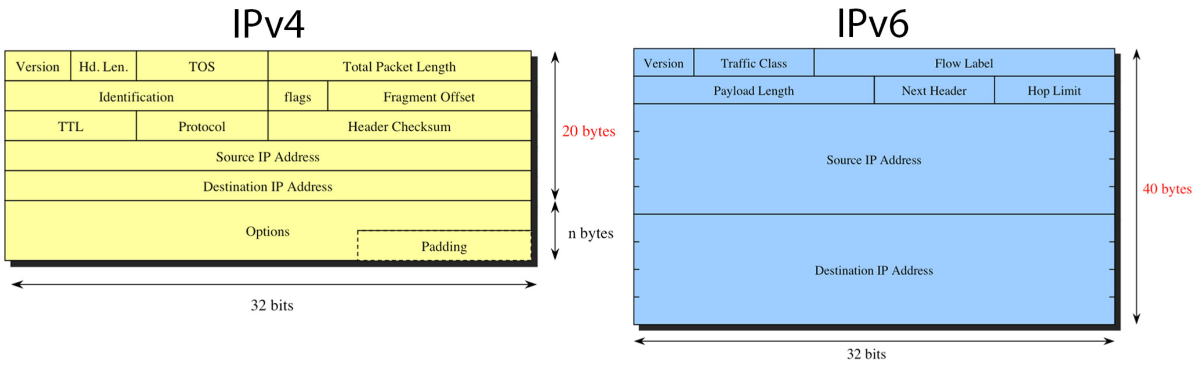


Figure 2.2: Header comparison of IPv4 and IPv6 [31]

Autoconfiguration In the IPv4 world you need to assign a static IP address to a host or get one assigned by a DHCP server. With IPv6 both of these methods, not surprisingly by introducing DHCPv6, are still available. An additional method is also introduced called stateless autoconfiguration where as the DHCPv6 methods gets the name stateful. When connecting a IPv6 node it is able to assign it self a global IP address based upon the network prefix, which is send out by the router, and its own network interface MAC identifier or private random number generated once during operating system installation life cycle.[17] This eliminates a central administration and monitoring of allocated addresses in a network. Especially useful in a fast and dynamic changing network; more often found in wireless networks. More detail about surrounding mechanism will be discussed in section 4.2.

MAC Layer Address Resolution IPv4 uses the Address Resolution Protocol to find the correct link layer (eg. Ethernet) address for an IP address. The combination of autoconfiguration and expanded address space requires IPv6 to implement another solution for this essential mechanism. The solutions is called Neighbor discovery, which be explained in section 4.2.

Multicast IP multicast is an extension for IPv4 but is part of the core in IPv6. Furthermore broadcast no longer exists in IPv6. It has been replaces by multicast addresses with the local network segment as target.

Mobile IPv6 Already mentioned as an standard extension header. Mobile IPv6 attempts to solve mobility problems at the layer three point of attachment to the network, including addressing and routing.[31] I will discuss this in more detail in chapter 9.

IPSec To provide security features, including encryption, IPSec was developed. IPSec was developed years after the IPv4 was standardized making it an optional feature. Retrofitting IPSec in existing IPv4 implementations introduced many inter-operability and performance issues. In contrast the IPv6 specification includes IPSec. Resulting in every standard conforming IPv6 implementation includes IPSec. [17]

NAT Network Address Translation is an opposite case. Using NAT a single external or global IP address can be used for multiple devices. Technically it is still possible with IPv6 but the need has been taken away as the address space now allows an unique global IP address for every device. As many IPv4 networks use a NAT setup this will cause some large changes in network planning, routing, topology etc. Applicable for both home and office networks and by that it is worthy to mention in this list of new features.

2.2.4 Usage statistics

The growth rate of the internet does not lie about the need for IPv6 adoption. Many predicted dates about IPv4 address space exhaustion have long passed and we are still stretching its lifetime.

To promote the adoption of IPv6 the Internet Society has organized the World IPv6 Launch Day event on June 8th 2011[39]. Several of the highest ranked websites by Alexa, including the top three: Google, Facebook and Yahoo, joined the event and activated IPv6 availability for their services on the Launch event day. My current home Internet Service Provider, XS4All[1], participated in this event and is still the only consumer ISP in The Netherlands providing native IPv6 connectivity to their customers. In other countries the situation is equal. Residents of and students on the University Twente campus terrain can also use a IPv6 capable internet connection. For far most consumers speed and price are far more interesting characteristics for a home internet subscription and can we can we can we prove them wrong? There is simply not a (large) commercial or financial incentive for IPv6 deployment. In Asia IPv6 became already more a reality. The high population and accelerated Internet growth rate, combined with the limited IPv4 address space, does not leave any other choices[17]. The Law of the handicap of a head start can be applied to IPv6 adoption.

Figure 2.3: World IPv6 Launch Banner[39]

Total usage measurements over the global internet is probably impossible I took the IPv6 usage statistics from Google to illustrate world wide usage. Figure 2.4 shows the percentage of users connecting to the Google servers using IPv6. Around halfway 2011 the line starts to display an interesting growth. Now, beginning of 2016, we are are still stuck around 8% so there is a long way to go before every internet user has (native) IPv6 connectivity. The adoption rate differs greatly per country as can be seen in figure 2.5.

[1] My intention is not at all to (commercially) advertise, but I am glad with the native IPv6 connectivity they offer me.

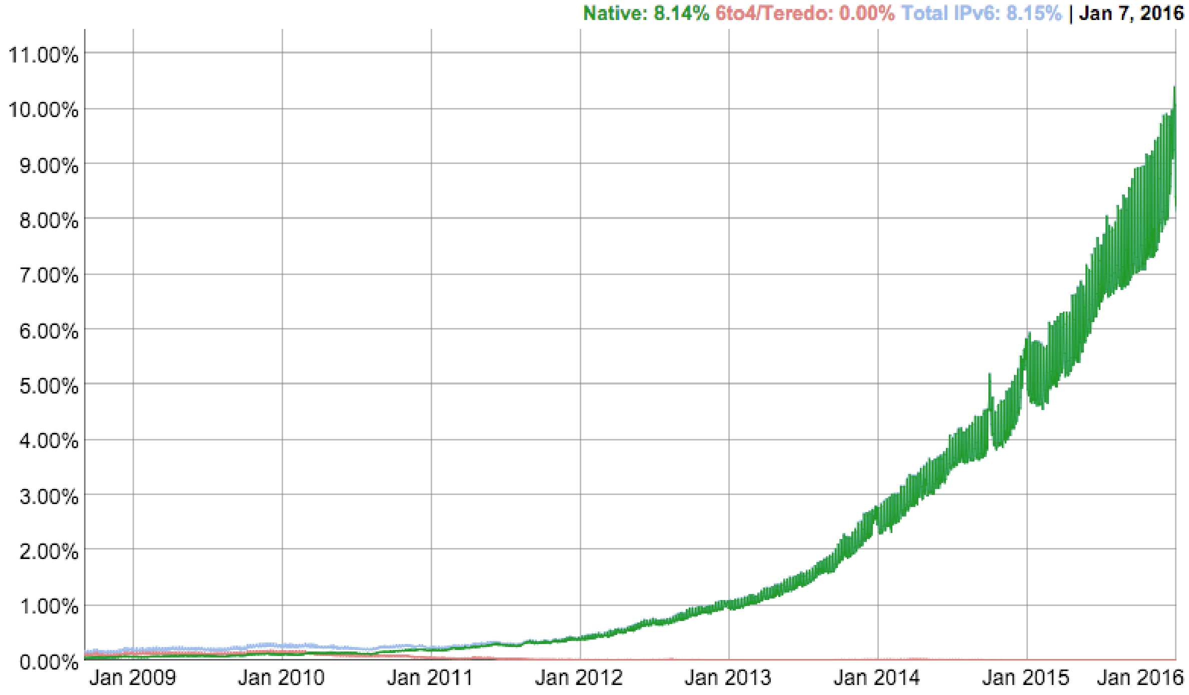


Figure 2.4: IPv6 usage statistics from Google[15]

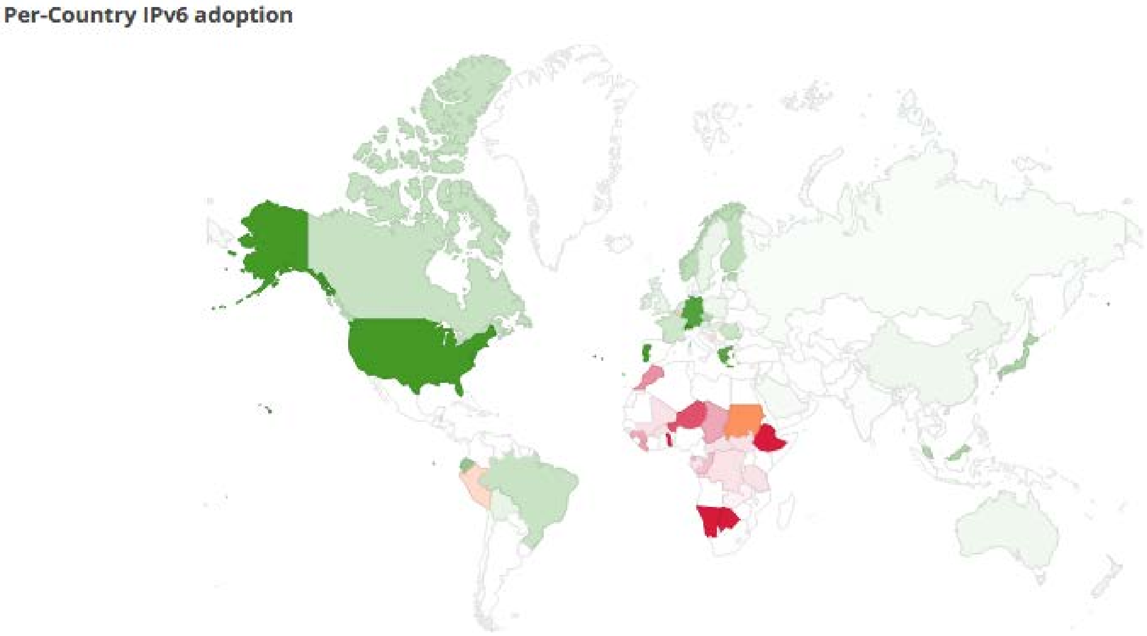


Figure 2.5: IPv6 adoption per country[15]

2.3 WiFi

Wireless networks offer several advantages over wired networks:[14]

Mobility
Connections are not fixated with cables which allows the user to move around with his mobile device without the hassles of cables.

Flexibility
No cables means no re-cabling. Wireless technologies allow networks to adapt rapidly to the available clients and wanted network topology.

Ease, speed and cost of deployment
Many areas are difficult to reach for traditional wired networks. Deploying cables in houses and office buildings is often a time consuming and even expensive task.

2.3.1 What is WiFi?

Wireless networking technologies have been very successful in the past years as it enables people to connect with their networks with far more freedom. By far the most successful wireless networking technology for computer networks has been IEEE802.11. The WiFi Alliance defines WiFi as any Wireless Local Area Network (WLAN) products that are based on the IEEE802.11 standards.

IEEE802 WiFi (IEEE802.11) is superficially similar to Ethernet (IEEE802.3) in its functions. Switching between Ethernet and WiFi should not affect upper layers in the OSI model. For further reference both use MAC addresses to address the nodes on the physical layer. IEEE802.11 is a member of the IEEE802 family, which is a series of specifications for local area network (LAN) technologies. Figure 2.3.1 shows the relationship between the various components of the 802 family and their place in the OSI model. IEEE802 specifications are focused on the two lowest layers of the OSI model because they incorporate both physical and data link components. All 802 networks have both a Medium Access Control (MAC) and a Physical (PHY) component. The MAC is a set of rules to determine how to access the medium and send data, but the details of transmission and reception are left to the PHY. The use of radio waves as a physical layer requires a relatively complex PHY.[14]

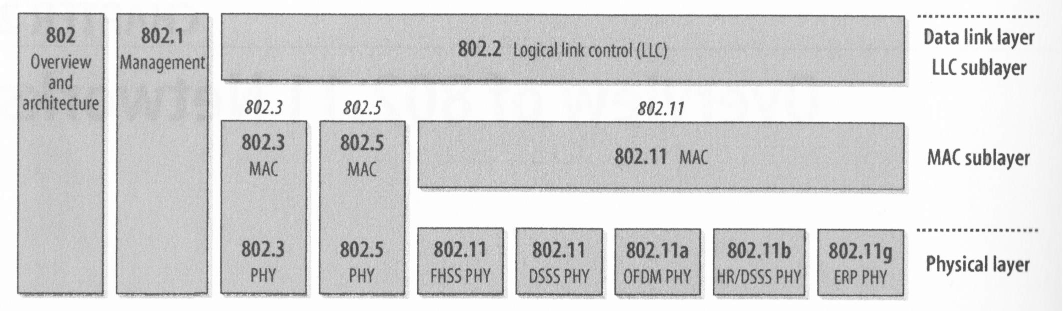


Figure 2.6: The IEEE802 family and its relation to the OSI model[14]

Systems parts WiFi networks consists of four major physical components.[14]

Access points
Access points are devices used to transfer frames between wireless and wired networks In infrastructure networks access points provide the security controls of the wireless network.

Wireless medium
The air between radios is used to move frames from one to the other carried in radio waves.

Nodes

The reason for networking is of course to connect nodes to each other. A WiFi node needs a WiFi Network Interface Controller to participate in the WiFi network.

Distribution System

Although not required most wireless networks will have a (wired) backbone to connect multiple access points to each other and distribute frames across the entire network.

Radio spectrum Wireless devices are constrained to operate in a certain frequency band. Each band has an associated bandwidth, which is simply the amount of frequency space in the band. Bandwidth has acquired a connotation of being a measure of the data capacity of a link. The use of a radio spectrum is rigorously controlled by regulatory authorities through licensing processes. To prevent overlapping uses of the radio waves, frequency is allocated in bands, which are simply ranges of frequencies available to specified applications.[14] IEEE 802.11 technology has been licensed for the so called ISM (Industrial, Scientific and Medical) bands around 2.4 and 5.0 GHz bands. To achieve the most usable bandwidth within these bands different IEEE802.11 standards use increasingly smarter modulation techniques. This however falls out of scope for this report.

Shared medium One of the most important differences with common wired technologies, like Ethernet, is found in the shared nature of the medium wireless technologies use. Instead of copper wires between two devices the air around the clients is used as a medium. You cannot claim and isolate a piece of air around you to use as a medium. All wireless nodes within each others radio range must share the same medium. A medium can only be used by one concurrent transmitter making it very important for wireless technologies to perform decent medium access control to prevent collisions and keep the shared medium usable. The MAC layer of IEEE802.11 uses a Request-To-Send Clear-To-Send mechanism to control access to the shared medium.[14] The basic operation of the RTS/CTS mechanism, as illustrated in figure 2.3.1, contains the following basic steps:

1. Wait for silence on the medium using physical carrier sense
2. Transmit a Request To Send frame to reserve air time and silence other radios
3. If the target node receives the RTS it will respond with a Clear To Send packet
4. Transmit your frames. Optionally the RTS/CTS clearing procedure can be used. in which every frame needs to be acknowledged by the receiver.

This procedure can take up a fair amount of air time, especially for small data transmissions. Tuning of different parameters, like for example the amount of transmission time allowed after a successful RTS/CTS exchange, can improve performance for a specific scenario. A RTS/CTS threshold value can be used to avoid the procedure at all for (very) small transmissions.[24]

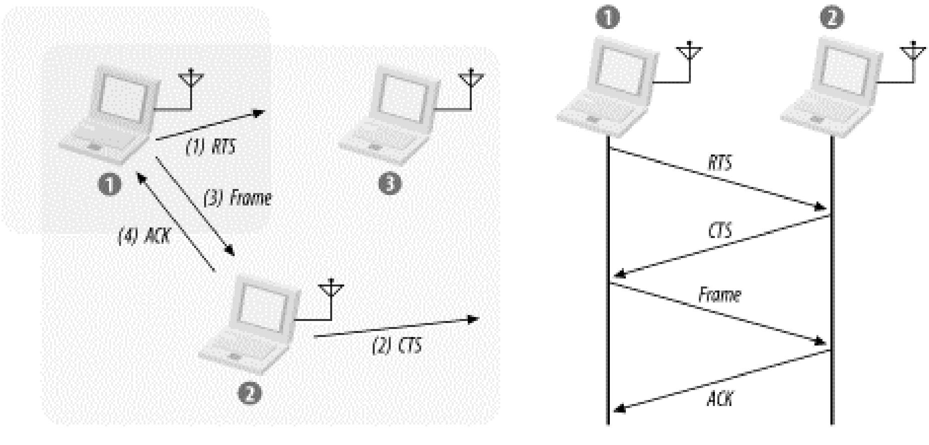


Figure 2.7: RTS/CTS mechanism[14]

2.3.2 Standards

Throughout the years several IEEE802.11 standards have been developed. Each aiming to improve the previous one. Table 2.3.2 gives an impression of the major standards.

Table 2.3 WiFi standards overview

Standard	Release year	Band	Supported data rates	Range*
IEEE802.11	1997	2,4 GHz	1 - 2 Mbit/s	20m
IEEE802.11a	1999	5 GHz	6 - 54 Mbit/s	35m
IEEE802.11b	1999	2,4 GHz	1 - 11 Mbit/s	35m
IEEE802.11g	2003	2,4 GHz	6 - 54 Mbit/s	38m
IEEE802.11n	2009	2,4 & 5 GHz	6 - 900 Mbit/s	70m

* Approximate indoor range for lowest data rate.

It is important to note that the maximum supported data rates are combined for the entire cluster of nodes connected to the same access point. All the active nodes share this available bandwidth. The actual used data rate will depend on the capabilities of both the client and access point combined with the signal to noise ratio between them; of which dynamic properties is primarily influenced by distance, obstacles and noise from foreign transmissions near the same frequency.

Backward compatible All the WiFi standards are designed to be backward compatible. Therefore the header format and size has not been changed forming a disadvantage. In the presence of a older standard newer devices will fall back and become slower causing performance losses in mixed environments.[40]

MIMO An interesting development of the IEEE802.11n standard is the introduction of a smart antenna technology called MIMO. Multiple Input and Multiple Output antennas are used to divide the same output power over multiple antennas giving an array gain. MIMO technology takes advantage of a natural radio-wave phenomenon called multipath whereas single antennas only takes disadvantages from multipath signals.

TXOP The properties of the shared medium (access control) together imply a total amount of Transmit Opportunities (TXOP) over the nodes in the same wireless network or air space independent of the payload size. It is bound by the RTS/CTS algorithm and (management) frame size defined in 1997. IEEE802.11g offers about 1300 TXOP/sec.[40] Many small transmissions can therefore consume all available TXOPs no matter the rising maximum supported data rates offered in the standards. The newer faster standards allow faster and more data to be transferred within a TXOP, however the increase of TXOPs/sec does not follow.

2.3.3 Transmission types

There are four main types/styles of data communication/transmission regarding the destination type. The four types are illustrated in figure 2.3.3.

Unicast The most straight forward and most used; communication from one host to another.

Broadcast It is also possible to transmit a packet to all hosts on the network segment.

Multicast Instead of addressing only one or all nodes it is also possible to address a single packet to multiple addresses.

Anycast For some protocols it does not really matter which nodes responds but instead you want the network to determine your best suitable target.

IPv6 uses multicast communication for various management traffic as described in section 2.2 which could make this an important aspect when deploying IPv6 on a WiFi-based network.

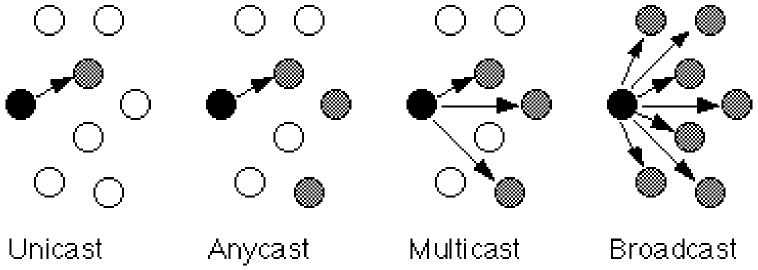


Figure 2.8: Four types of communication

Multicast IEEE802.11 does not offer multicast style communication. To overcome the lack for multicast communication, multicast packets are treated as broadcast packets.

Hidden terminal The range of a wireless node is not static determined by a cable length. The combination of transmission power, antenna gain and obstacles between sender and receiver determines the range. A difference between send and receive range is even possible. Multiple nodes can be connected to the same wireless access points while not in each others range. They cannot sense each other but can distort each others transfers. This is called the hidden terminal problem. See figure 2.3.3 for an illustration. Beside a problem for medium access control it creates the need for the access point to retransmit the packets in order to transfer data between these nodes.[14] In this simple three node setup it takes approximately twice the amount of air time to transmit the same data. As it is not possible for the access point to know if direct communication between two nodes is possible, communication between nodes connected to the same access point will always go through the access point. This is also the case for broadcast transmissions originating from a connected node.

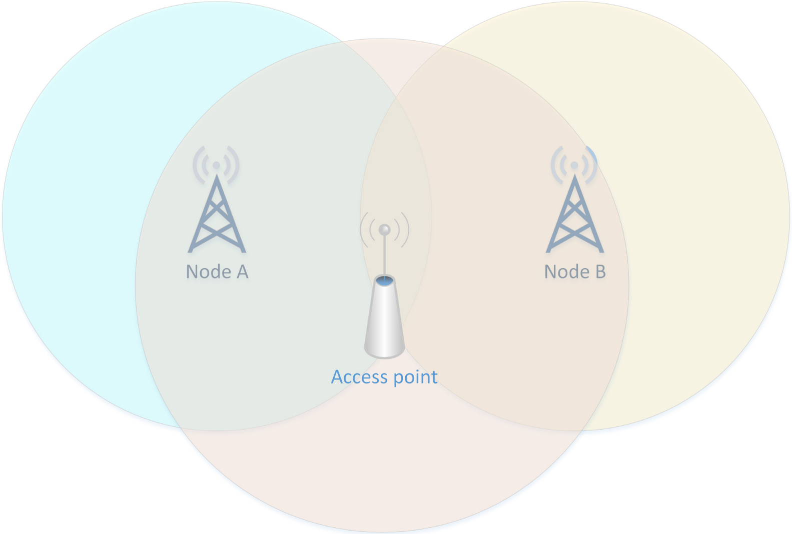
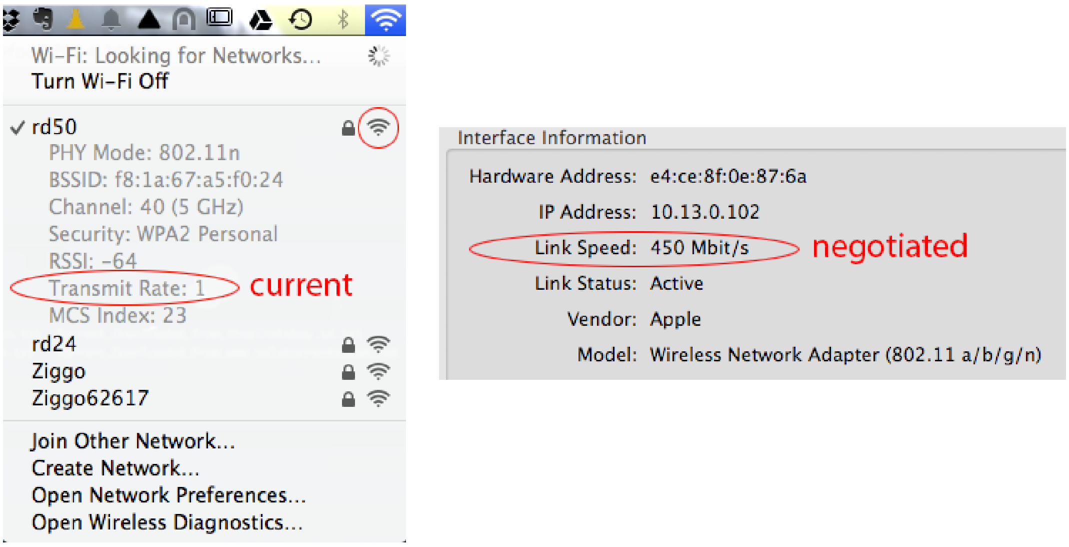


Figure 2.9: The hidden terminal problem

Encryption keys Different security types for WiFi exists, like WEP or WPA/WPA2. Discussing these types in detail falls out of scope of this report. One point is worth mentioning; more advanced security types use unique encryption keys for each node; between client and access point. This offers an important security advantage as nodes are not able to decrypt each others communication. However a broadcast packet should then be send separately to each node demanding a linear increasing amount of air time based on connected nodes. To overcome this problem IEEE802.11 use a shared key per access point for broadcast transmissions.[14] This however introduces a small security problem when multiple network (different SSID connected to different VLANs) are served on a single access point. Broadcast transmissions can be now be decrypted by nodes connected to another SSID as the broadcast key is shared. In practice this will not cause any problems as it will only apply to packets containing basic network information (eg. found in ARP or NDP packets).

Lowest data rate The used data rate between a node and an access point is determined by their capabilities and the signal quality between them. In order to transmit a broadcast packet to all its connected nodes, the access point needs to fall back to the lowest data rate in use on its connections. All the connected nodes will briefly fall back to this lower data rate during the broadcast.[11] When all connected nodes are enjoying a fine signal quality this will not raise a noticeable issue. However when a node is on the edge of the access point range it will use the lowest supported data rate to stay connected. Like seen in section 2.3.2 this can be as low as a single megabit per second. Of course the broadcast transmission itself will be done using a low data rate, but the main issue arises as the entire wireless cluster will step down during in the entire transmission process around the broadcast packet transmission, including the RTS/CTS medium access control mechanism.[25] Only a fair amount of broadcast traffic can hereby be enough to cripple the wireless network bandwidth. Figure 2.3.3 shows the connection state of a machine running Mac OS X; it has the maximum reception indication of four bars and is still forced into the very low data rate of a single megabit per second caused by the poor reception on another node during the many broadcast transmissions of multicast packets. To avoid a crippled wireless network due to broadcast/multicast traffic many wireless access points have the option to block broadcast transmissions, which of course can also block certain kinds of traffic and protocols.



PHY Mode: 802.11n
BSSID: f8:1a:67:a5:f0:24
Channel: 40 (5 GHz)
Security: WPA2 Personal
RSSI: -64
Transmit Rate: 1
MCS Index: 23

Hardware Address: e4:ce:8f:0e:87:6a
IP Address: 10.13.0.102
Link Speed: 450 Mbit/s
Link Status: Active
Vendor: Apple
Model: Wireless Network Adapter (802.11 a/b/g/n)

Figure 2.10: Mac OS X displaying fine WiFi reception however crippled by a very low data rate currently in use on a crowded IPv6 network.

Most access points will provide a configuration parameter to set the lowest offered data rate to reduce this problem. Increasing the lowest supported rate will decrease the offered range as a better signal level is needed to maintain connection. For network administrators it is dilemma between maintaining speed and providing range.

2.4 UT Campusnet

The University of Twente strives to stay ahead on many technical frontiers. One of them is their computer network. Their campus terrain, includes academic buildings, student housing, sport and cultural facilities and much more is unique in The Netherlands. This also presented beautiful opportunities and challenges on their network. The network has two logical parts, UTnet for the academic environment and Campusnet for the student environments, coupled behind the same internet uplinks. Both networks are used extensively and handle a lot of traffic. It gives both students as staff members opportunities to use the network between them, within the network or through the internet, as they see fit for their sometimes very demanding needs. The University of Twente was one the first universities in Europe to deploy a campus wide wireless network in June 2003. Internet access became possible for staff and students anywhere on the campus without needing to plug a cable into a wall socket on the university terrain of about 140 hectare.[10] Years later the eduroam initiative has been joined, allowing foreign students (national or international) to make use of the wireless network using the credentials of their own participating universities.

There are hundreds of wireless access points installed throughout the University campus terrain in the endeavor to achieve campus wide WiFi coverage. Figure 2.4 shows map to illustrate the access point deployment.

Beside a very fast (gigabit internet connectivity in student homes) the University of Twente was the first University in The Netherlands to become ready for IPv6.[35] Early experiments started before the year 2000 but a destructive fire of the main data center in 2002 formed a major setback. Years later the IPv6 internet connectivity was restored and a new start was made by offering native IPv6 connectivity to wired connections on Campusnet in student housing. After earlier failed attempts IPv6 connectivity also became available on the wireless network in April 2014.

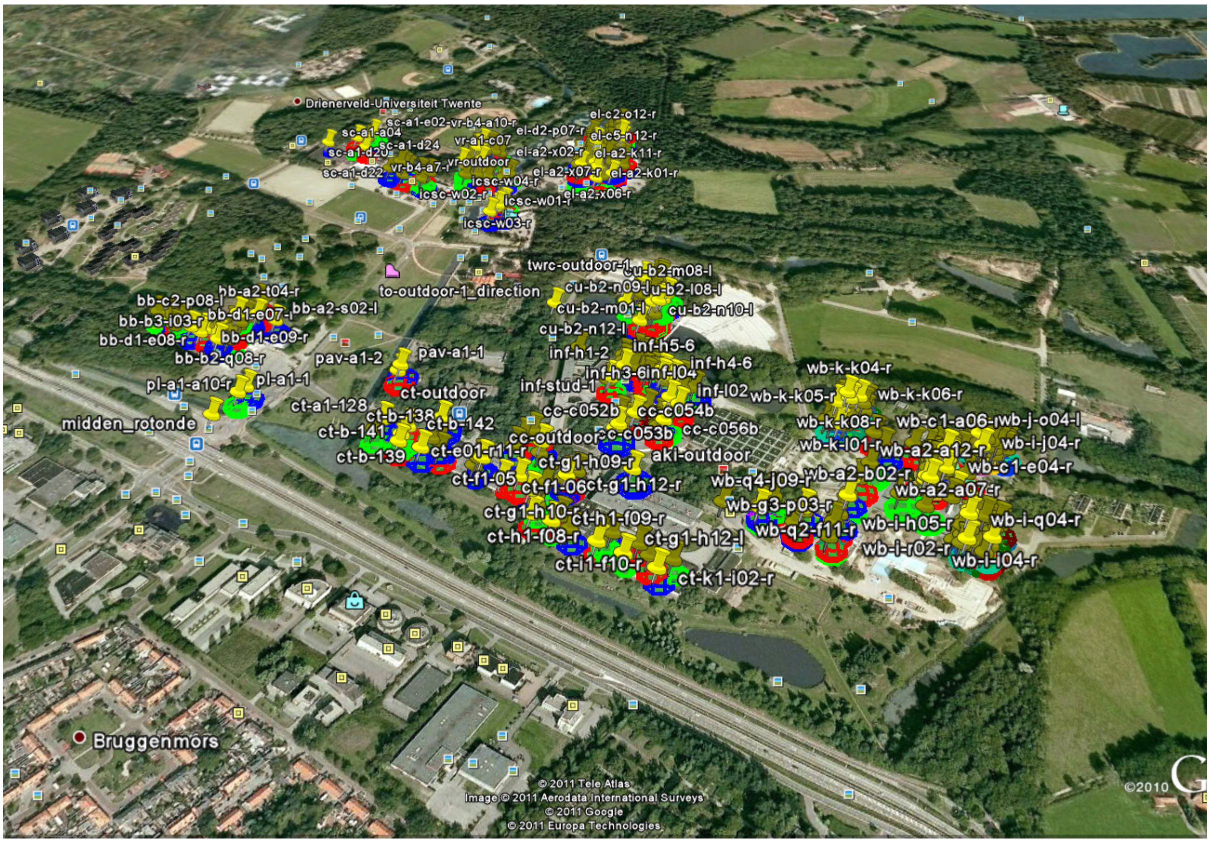


Figure 2.11: University of Twente map of access points

Chapter 3

Research

3.1 Considerations and assumptions

In order to keep this research in bounds and to refine the main research question I will keep the following considerations in mind while stating a few assumptions based upon them.

This research is only about native IPv6 connections and by that does not include tunneling methods like Teredo and 6to4. Tunneling methods will become regular IPv4 traffic between end-host and tunnel server causing the traffic on WiFi related layers to remain IPv4.

In practice an IPv6 enabled network will be dual stack with IPv4 still enabled. All control traffic of IPv4 will thereby remain present. This is not accountable to IPv6 directly but should be kept in mind.

3.1.1 Performance

An important aspect of many technologies is performance. The terminology performance can be described in many ways. In case of (wireless) networking a few metrics to determine performance are: bandwidth individual and/or shared, latency, power usage, wireless coverage range, packet/frame count, maximum supported users, reliability, packet loss etc. For networking technologies the available bandwidth and latency are in general important performance metrics. Many performance aspects of an end-to-end network connection, like range or medium bandwidth, fall out of scope when only altering the IP protocol in the entire stack as they are ascribed to characteristics of other layers in the stack, like link data rate and transmit range, which all stay unchanged. Considering the new features offered by IPv6 no drastic changes in performance should be expected in general networks. The research of Huston[22] confirms this expectation. Huston compares the end-to-end performance of IPv4 and IPv6 using a production website. Huston concludes the round-trip-times are roughly equivalent.

On the internet routing works often on a per protocol basis. Even when peering and transit providers match the resulting paths may differ. Comparing performance of both protocols over the internet is thereby a more complex task and not the purpose of this research as this is not influenced by the wireless part. On a local connections, without routers in the connection path, these considerations do not apply.

Considering above statements I will, in order to determine the impact of IPv6 in contrast to IPv4 on a WiFi-based wireless network, focus on total available bandwidth as first performance metric.

3.1.2 Differences

Chapter 2 describes the main differences between IPv4 and IPv6.

Figure 2.1, the OSI model, illustrates the complete stack used in end-to-end data communication and the parts where IPv6 and WiFi play a role. All other layers will be equal in IPv4 and IPv6 systems. The focus of this research should therefore be placed upon the influenced layers. The actual data to be transferred comes from upper layers and therefore shall not change, not affecting the actual size to be transferred over the connection. The packet header appended by either IPv4 or IPv6 will enlarge the packet and by switching the used version will affect the data for the lower layers. The differences in the

IP packet header between the two versions of the protocol, see section 2.2.3, will only change packet size marginally and therefore I assume this will not make a significant difference in total performance.

Aside of data transport the other aspect of the IP protocol is found in the management and control traffic and activities. The sub-protocols used by the versions for management and control functions do differ on different aspects. Version six of the ICMP control protocol may give an important handle to research the impact of IPv6 on a WiFi-based network. Therefor I will discuss ICMPv6 in greater detail in chapter 4. Packets required for these network management and control tasks performed by these sub-protocols, which are an important part of the IP stack, may influence the total network performance and makes another interesting metric for my research.

3.2 Real world observations

Aside for gathering background information about IPv6 and WiFi I also tried to gain some knowledge from real world networks. The information gathered gave me both a quick and high level as complex and detailed insight in what IPv6 is doing on a wireless network. I used my laptop to capture wireless traffic using Wireshark and by setting my wireless network adapter in monitor mode (described in section 6.2.2). I did these captures solely to get a feeling of what is happening and to spot potential problems and not to gain scientific results. These are not setup experiments which can be repeated in a controlled environment. These captures gave a great insight and helped determining the focus for my research.

3.2.1 Home network

To prevent diving in too deep I started to look at my own home network. This network consists of a handful of devices and has native IPv6 internet connectivity. The capture showed all the expected ICMPv6 packet types. The network connection was perfectly usable for both IPv4 as IPv6 traffic. As a comparison I disabled IPv6 on my router and captured again. The performance felt equal between the two scenarios. No potential problem did arise.

Something noticeable is the amount of packets generated by discovery services, like Bonjour, Netbios and uPnP. These services will periodically scan the surrounding network for available services and can initiate a lot of control traffic to gain information about neighboring IP addresses. These services are most useful in home networks to provide an easy plug and play experience to end-users and are not designed for use in larger (corporate or public) networks. This does not only affect IPv6 however an increase in control traffic can be assigned to these services.

3.2.2 Eduroam at University Twente

The University of Twente already offers native IPv6 connectivity to their users on the wired network for several years. A next step is of course to include the wireless network. There have been multiple experiments enabling IPv6 on the campus wide university network, called Eduroam (the European standard SSID for academic wireless networks). First tests rapidly drastically reduced the network performance to a level it was no longer usable. Later test, with newer equipment like thin access points controller by central wireless controllers, initially promised better results. For a few months the wireless network offered both IPv4 and IPv6 connectivity. However performance was still affected and too often caused problems. That is why until there is a solution IPv6 support is disabled again on the wireless part of the university network. One of the key problems appeared to be caused by nodes that are just in range of an access point and forces the entire wireless cell to drop to a low data rate for broadcast transmissions as described in section 2.3.3.

Capturing During the period that IPv6 was enabled on the wireless network I took the opportunity to investigate the network traffic. I did this in multiple situations, of which the following were most interesting:

1. During a regular weekend in an academic building with practically nobody around me. This gave a situation with only a few connected devices and low network usage.

2. In my student home just before diner time. This gave a situation with a handful connected devices of which a few were quite heavily using the network connection (eg. Youtube, which offers IPv6).

3. Just outside the largest lecture halls filled with students, which were hopefully attending the professor but most of them will carry a smartphone associated with the wireless network. This gave many connected devices with low network usage by the users.

4. During the exam period in the university library with lots of students in the room working on their laptop or tablet. This gave me a situation with many connected devices which were all using the network connection.

The first situation was quite similar to my home network. I did see more ICMPv6 traffic compared to ARP but nothing to see a potential problem in. All the other three situations showed me the same behaviour. The ratio of control packets from the IPv6 stack to that of the IPv4 stack was leaning completely towards IPv6. The amount of router advertisement packets was higher than expected, the routers probably use a quite high interval, but the numbers were not alarming. The amount of neighbor solicitation and advertisement packets was, in all the captures with many devices connected, at least a power of ten higher than ARP packets. I also noticed great fluctuations on the connected rate of my wireless network adapter. This was not the case in the first situation. These fluctuations can be caused by a larger amount of collisions or actively influenced by the access point as formed a problem in earlier tests. Two interesting observations for further research.

3.3 Research questions

As stated before, the goal of my research and report is to evaluate the impact and possibilities of IPv6 when deploying on a WiFi-based network.
Considering above considerations and assumptions this leads me to the following research questions:

1. What is the difference in control traffic between IPv4 and IPv6 and how does it influence network performance?

2. How does the use of multicast by IPv6 influence WiFi-based networks?

3. Are there solutions to solve identified issues for deploying IPv6 on a WiFi-based network?

In addition to my primary research I formed the following secondary research questions to contemplate additional interesting aspects of IPv6:

4. Does IPv6 offer new features or possibilities especially interesting for wireless networks?

5. Does IPv6 introduce security risks and vulnerabilities, especially applicable on wireless networks?

In attempt to answer these questions I will make use of available literature, experience and knowledge of people in the field and by conducting my own experiments.

CHAPTER 4

ICMPv6

The network control functions constitute an important part of the network and transport layer. Therefore this chapter dives in more detail of this part of IPv6 in follow up to the background information given in chapter 2.

People familiar with IPv4 will recognize the Internet Control Message Protocol (ICMP), maybe even as a good debug friend as it gives important information about the health of a network. Not surprisingly ICMPv6 is the version that comes with IPv6. ICMP is used to report errors or information about the status of the network. It also performs diagnostic functions, such as the well known ping command, which uses ICMP Echo Request and Echo Reply messages to test availability of a node. ICMPv6 has extended functionality over its predecessor.

ICMPv6 has extension number 58. A preceding header in a IPv6 packet will mention this number in its next header field. The header format itself is quite small and simply, which can be seen in figure 4.1, and should keep overhead within bounds.

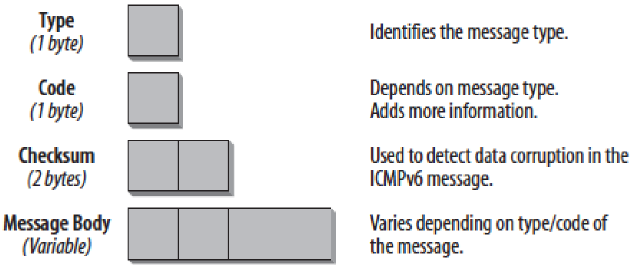

Type
(1 byte)
Identifies the message type.
Code
(1 byte)
Depends on message type.
Adds more information.
Checksum
(2 bytes)
Used to detect data corruption in the ICMPv6 message.
Message Body
(Variable)
Varies depending on type/code of the message.

Figure 4.1: The ICMPv6 header[17]

4.1 Types

There are two classes of ICMP messages:[17]

ICMP error messages
Like the name indicates these messages indicate an error. Error messages have a 0 in the high order bit of their Message Type field (range 0-127).

ICMP informational messages
Likewise holds for informational messages. However there is a 1 in the high order bit of their Message Type field (range 128-255).

The most interesting part of ICMPv6 for this report is found in the neighbor discovery related messages, discussed in the next section.

4.2 Neighbor Discovery

Section 2.1 explains the basic topology of a network stack and its top down layers. Each layer communicates with those above and below them. In IPv4 the **A**ddress **R**esolution **P**rotocol is used to find the corresponding data link layer address to a network link layer, an IP address to in general, for ethernet, a MAC address. When an hosts requires the link layer address it broadcasts an ARP request to all nodes on the network. The host with the corresponding IP address will send a ARP reply containing its link layer address. See figure 4.2 for a flow diagram.

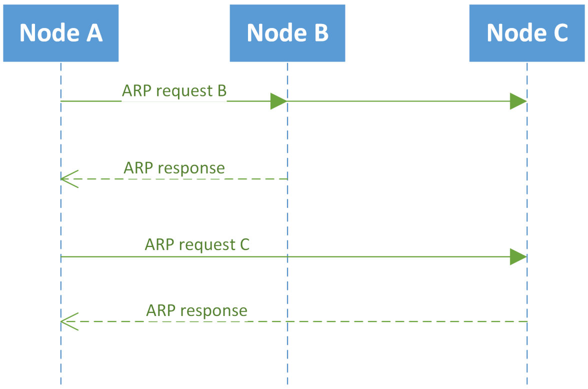


Figure 4.2: ARP sequence diagram

The successor of ARP in IPv6 is called the **N**eighbor **D**iscovery **P**rotocol, often referred to as **N**eighbor **D**iscovery[33].
Neighbor Discovery supersedes ARP and also features additional functionality. The main features of Neighbor Discovery are:

Address resolution
Like ARP it resolves IP addresses to link layer addresses, self evident for IPv6 addresses.

Address autoconfiguration
Routers can advertise the available network prefix which is used in the autoconfiguration process.

Duplicate IP Address Detection
Duplicate addresses on the network can be detected. Autoconfigured addresses (or those from privacy extensions) already in use by other nodes can be detected and thereby passed.

Find neighboring routers
Routers to other network segments, or of course the internet, can be found without static configuration.

Keep track of reachable neighbors
Reachability of other nodes in the network can be monitored.

Detect changed link layer addresses
Nodes can notify other nodes when it changes its link layer address.

Specify network options
Additional network parameters like the MTU can be configured by ND. Later extensions to specification also allowed other options like DNS servers or the local domain name.

One of the main differences between the two protocols is the layer at which they operate: ARP is a link layer protocol, while NDP operates at the network layer.[29]
Neighbor Discovery defines five different message types to perform its functionality, which are part Internet Control Message Protocol (ICMPv6) packet types. This includes two pairs, Router Solicitation coupled with Router Advertisement and Neighbor Solicitation with Neighbor Advertisement. These different type of messages serve the following purpose:[33]

Router Advertisement
Routers advertise their presence on the network using Router Advertisements. It can include

various link parameters and other network parameters. Routers advertise either periodically, or in response to a Router Solicitation message. Router Advertisements contain prefixes that are used for address autoconfiguration.

Router Solicitation
When a node wishes a Router Advertisement, for example when enabling an interface, hosts can send Router Solicitations. The solicitations request routers to generate Router Advertisements immediately, rather than at their next scheduled time.

Neighbor Solicitation
A node can determine the link layer address of a neighbor using Neighbor Solicitation messages. It is Also sent by a node to verify that a previously known neighbor is still reachable by its cached link layer address. Neighbor Solicitations are also used for Duplicate Address Detection.

Neighbor Advertisement
A response to a Neighbor Solicitation message, which includes the link layer address of the node. Nodes can also send unsolicited Neighbor Advertisements to announce a link layer address change.

Redirect
Used by routers to inform hosts of a better first hop for a destination, or that the destination is on the same local link. As used by routers only this in general will not be seen on wireless links.

Figure 4.3 shows a sequence diagram of the Router Solicitation and Advertisement combination and figure 4.4 a sequence diagram of the Router Solicitation and Advertisement combination. A flow diagram for packet transmissions is shown in figure 4.5.

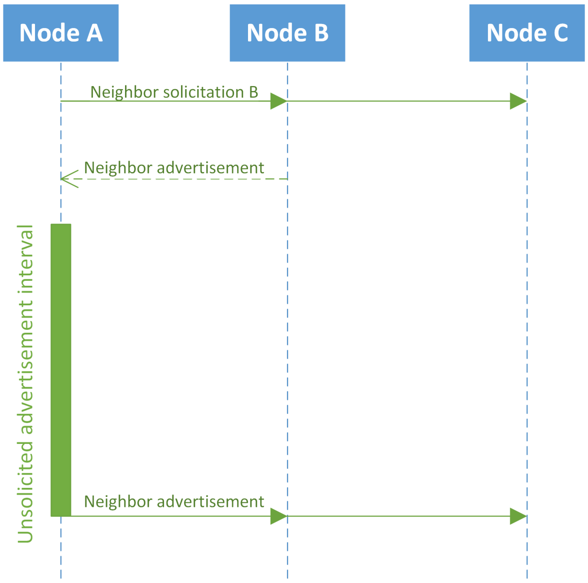


Figure 4.3: Neighbor Solicitation & Discovery sequence diagram

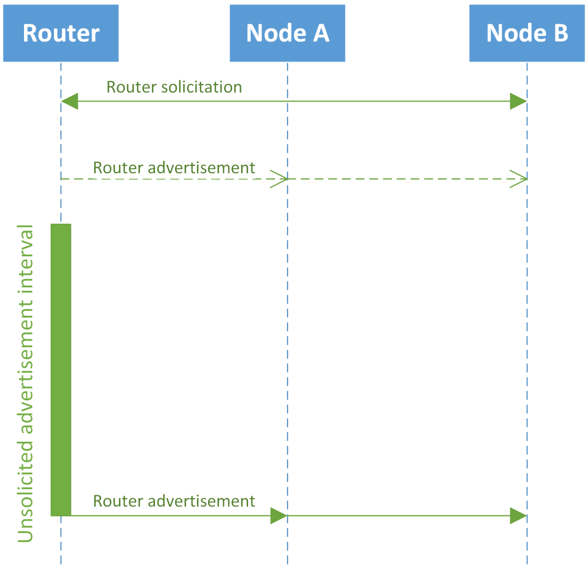


Figure 4.4: Router Solicitation & Advertisement sequence diagram

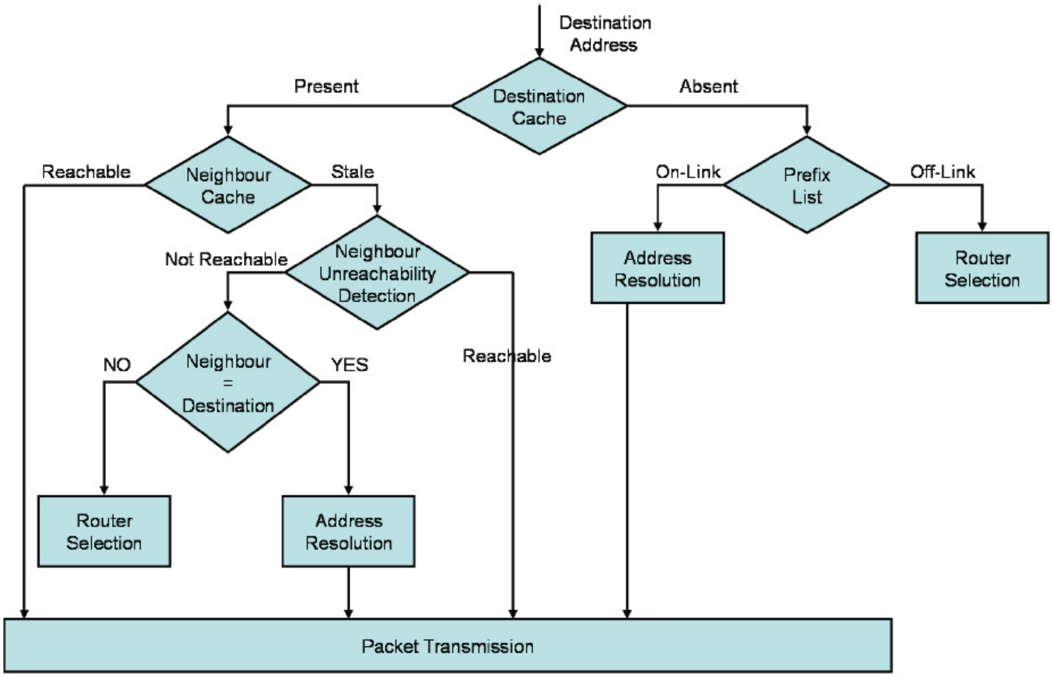


Figure 4.5: Packet transmission flow diagram[21]

4.2.1 Router Advertisement

Router advertisement messages are used to advertise the availability of routers on the network. It can also be used to configure network parameters. This feature enables centralized administration of some important network parameters as they are configured on the router and automatically propagated to all hosts that are attached. The ability to configure clients on the network using router advertisements might lead to possibilities for this research and solutions for encountered problems. Some are already present in the first specification, others are later added and thus may not always be supported.[36]

Router address
The address of the router that is advertising itself. As it is included in the router advertisement message no further neighbor discovery is needed to communicate with the router.

Prefix information
Information about the IPv6 subnet that the router has to offer to the clients. It contains the prefix for autoconfiguration addresses.

Default router preference
Multiple (default) routers can coexists on the network. Using a preference value a preference order can be specified.

Lifetime
The time the information in this router advertisement may be used. If set to zero then the advertised router should not be used as the default router.

MTU
The maximum transmission unit that should be used on the network segment.

Hop limit
The default maximum hop count value for outgoing unicast packets through the advertised router.

Managed flag
Indicates if this subnet is managed by DHCPv6.

MobileIPv6 config
Information about MobileIPv6 support and if available parameters needed for configuration.

Intervals & timeouts
Different types of intervals and timeouts can be configured through router advertisements This includes the intervals for unsolicited router or neighbor solicitations and advertisements.[41]

4.3 Multicast usage

Multicast is used by most messages in the neighbor discovery protocol.

Remember on a WiFi network multicast will become broadcast and broadcast transmissions will be done at the lowest supported data rates of all connected clients, see section 2.3.3. IPv6 does not support broadcast and uses multicast instead. A mismatch arises in the combination of WiFi and IPv6. The difference in nature of multicast and broadcast can result in much more transmissions as every connected client will be targeted by broadcast traffic and not only those part of the multicast audience. On a single access point this will not matter as the nature of the shared medium results in transmissions being received by every client. However when multiple access points form a single network then not every access point might need to transmit the multicast transmission as it has no subscribers associated. In case of broadcast every client is targeted and thus every access point needs to transmit it.

Access points Incoming multicast messages on the wired interface of an access point will be broadcast over its wireless radio interface. As multiple access points can be connected to the same wired network, broadcast messages will transmitted multiple times. This will often be desired behavior as otherwise not all network nodes will be reached. On the other hand can it burden the shared wireless medium with many messages with targets beyond their scope.

4.3.1 Multicast group management

Multicast packets are addressed to a multicast group address and only the hosts that are member of this group will process that packet. Multicast is an integral part of IPv6 in contrast to IPv4.[17] IPv6 defines multiple multicast addresses for different scopes:

- FF02 link local scope
- FF05 site local scope
- FF0E global scope

Multicast group management is done by the **M**ulticast **D**iscovery **P**rotocol, part of ICMPv6. The protocol allows multicast listeners to register for multicast addresses for which they want to receive data, to ensure efficient routing. A node sends listener reports for its multicast memberships using **M**ulticast **L**istener **D**iscovery. This active form of multicast membership is called solicited node multicast. All MLD messages are sent using link local multicast addresses and a hop limit of one to make sure they remain on the local link.

Routers and intelligent switches can monitor the MLD listener reports in order to discover which multicast addresses have listeners on each of its links.[8] Routers basically listen and process traffic that is not directly intended for them and that is why this process is also called MLD snooping. It is similar to ARP snooping for IPv4.[3] Using the acquired knowledge of group members the router or switch only have to send multicast traffic on its links with connected members. Preventing broadcasting the data on all its links and thereby propagating it to unnecessary parts of the network. A wireless access point only has a single wireless link so it is unable to send traffic on selected links only. However when a network consists of multiple wireless access points information about multicast group memberships can be used to know which access point need to transmit the multicast data.

Recall from section 2.3 the lack of multicast support in WiFi, transforming multicast traffic to broadcast traffic. The IPv6 stack will perform MLD on wireless links without achieving the desired effect as it will operate in broadcast form.

4.3.2 Addresses and multicast groups

The main advantage behind using solicited node Multicast IPv6 address is that only the interface which is configured with the particular unicast or anycast address will be listening to the solicited node multicast address. All other interfaces are not disturbed every time a Neighbor Solicitation message is sent to the solicited node multicast address, which belongs to another interface or node. Only the member nodes need to receive the multicast group packets removing the need to broadcast them to the entire network. Proper multicast group management on the network is required to achieve this.

All hosts are required to join the associated solicited node multicast addresses for all unicast and anycast addresses that are assigned to the corresponding interface.[20] The neighbor discovery protocol relies on these memberships for discovery and solicitation messages.

Solicited node multicast addresses are derived from the associated unicast or anycast IPv6 address.[20] Solicited node multicast addresses are defined in the network prefix FF02::1:FF00:0/104, part of the link local scope. The remaining 24 bits are taken from the unicast or anycast address, namely the last 24 bits. The collision space of solicited node multicast addresses is thereby 24 bit large. The chance of having two unicast addresses sharing the same solicited node multicast address is 22̂4 or one in 16 million. Causing the amount of used multicast groups to effectively scale in a linear way with the amount of nodes on the network. Considering each IPv6 enabled node has a least a single global unicast address and a link local address it will require two solicited node multicast addresses. When also considering privacy extensions this will only increase.

The amount of supported multicast groups by network equipment is limited. In general this will not exceed 1000 groups on professional network hardware and far less on consumer grade hardware. In practice this will prevent successful MLD snooping when more groups are in use. The behaviour of the equipment in this case is often not documented and can be either to simply fallback to broadcast mode or dropping those packets[42].

CHAPTER 5

THEORETICAL RATE APPROXIMATION

In this chapter I will formula some basic formulas in order to predict packet rates for the network control protocols and give general feeling about it. The goal is not to create a fully covering and complete mathematical model. Goal is solely to take a look at the packet rates that can be expected from both control protocols.

5.1 IPv4: ARP

IPv4 uses ARP to translate IP addresses to the MAC addresses of the lower (ethernet) layer. A ARP request for a IP address is broadcasted to all hosts on the network and the host with the corresponding IP address will respond with an ARP reply containing its link layer address. A host will keep a cache (ARP table) so it does not need to send ARP requests for every IP packet. To make sure that an IP address is still owned by the same MAC address the cache expires after the so called Reachable Time and a new ARP request will be send.[18] The default Reachable Time for most desktop operating systems (including Linux and Windows) is between 15 and 45 seconds, based upon a base time of 30 seconds and a random generated multiplier. This results in an average of 30 seconds for every connected IP host.[29] λ: packet rate per minute

$$n_{ARP} = n_{ownARPRequests} + n_{otherARPRequests} + n_{ARPreplies}$$

$$\lambda_{ARP} = 2 * n_{Connections_{LAN}}$$

Others hosts will also broadcast their ARP requests.

$$\lambda_{ARP} = n_{Connections_{LAN}}$$

$$\lambda_{ARP} = 2 * n_{Connections_{LAN}}$$

Connections to the Internet (WAN) usually all go through a single router, requiring only a single link layer address, matching the IP address of the router.

$$\lambda_{ARPrequests} = 2 * n_{Connections_{LAN}} + f(n_{Connections_{WAN}})$$

$$\lambda_{ARPreplies} = 2 * n_{Connections_{LAN}} + f(n_{Connections_{WAN}})$$

$$f(x) = \begin{Bmatrix} 0 & x = 0 \\ 2 & x > 0 \end{Bmatrix}$$

Combining above formulas gives:

$$\lambda_{ARP} = 4 * n_{Connections_{LAN}} + f(n_{Connections_{WAN}})$$

$$f(x) = \begin{Bmatrix} 0 & x = 0 \\ 4 & x > 0 \end{Bmatrix}$$

5.2 IPv6: NDP

As mentioned before ARP is superseded by Neighbor Discovery in IPv6. NDP operates in a similar way as ARP. For every connected IP address, now version 6, it needs a link layer address. Again a cache is used to prevent the need for discovery for every IP packet. When the host does not know the link layer address of the IP address it sends out a Neighbor Solicitation message and waits for a Neighbor Advertisement response. Most operation systems use the same standard values for the reachable time as with ARP, an average of 30 seconds.

$$n_{NDP} = n_{NDPsolicitation} + n_{NDPadvertisement}$$

$$\lambda_{NDPsolicitation} = 2 * n_{Connections_{LAN}} + f(n_{Connections_{WAN}})$$

$$\lambda_{NDPadvertisements} = 2 * n_{Connections_{LAN}} + f(n_{Connections_{WAN}})$$

$$f(x) = \begin{Bmatrix} 0 & x = 0 \\ 2 & x > 0 \end{Bmatrix}$$

IPv6 routers will also periodically advertise itself. There is no default interval value stated in the RFC[41]; an interval of 10 seconds is not uncommon in default configurations.

$$\lambda_{RAadvertisements} = 60/10$$

Connecting hosts will also send out router solicitations; however that depends on the rate of disconnecting hosts which in general will not be very high. Thus I will not take that into account here.

NDP will also monitor hosts in its neighbor tables for freshness using Duplicate Address Detection messages.

$$\lambda_{DADrequests} = 2 * n_{Connections_{LAN}}$$

$$\lambda_{DADreplies} = 2 * n_{Connections_{LAN}}$$

Combining above formulas gives:

$$n_{NDP} = n_{Nodes} * (n_{NDPsolicitation} + n_{NDPadvertisement} + n_{DADrequests} + n_{DADreplies}) + n_{RAadvertisments}$$

$$\lambda_{NDP} = Nodes * (8 * n_{Connections_{LAN}} + f(n_{Connections_{WAN}})) + 6$$

$$f(x) = \begin{Bmatrix} 0 & x = 0 \\ 2 & x > 0 \end{Bmatrix}$$

According to these formula's NDP will give and increase of control packets compared to ARP. This is in agreement with the observations made in section 3.2.

CHAPTER 6

EXPERIMENTS

6.1 Tests

In order to answer the research questions stated in section 3.3 I need to analyze the effect of IPv6 on a wireless network. I designed a few experiments to accomplish this. A comparison between IPv4 and IPv6 within the experiments will be required to determine the influence of IPv6.

I have designed and executed the following tests:

ARP vs NDP
How does the amount of ARP packets compare to NDP packets? This will follow up the first research question. Results are gathered using traffic sniffing and analytic tools. Goal: comparison of control packet amount and thereby the control overhead of both IPv4 and IPv6.

Bandwidth
How many bandwidth (maximal achievable throughput) will be available combined on the wireless network in different usage scenarios? Performance testing tools iperf and netperf will be used to measure bandwidth. Goal: measure impact of IP protocol version on available bandwidth. This will help in answering the first two research questions and hopefully provide some directions for number three.

Border node influence
How great is the impact of a single node on the border of reception? While measuring performance, what happens when another node physically moves around and its supported data rates decreases while its distance to the access point increases? Goal: determine influence of bad wireless reception of a node on the network in combination with broadcast and multicast transmissions.

Divide nodes
What is the effect if the same amount of nodes on the same amount of access points are divided over subnets by letting every access point use a different SSID with a different subnet? This test will be combined with the first and second test. A border node will only influence a single access point network so the third tests will not be combined with this last test. Goal: measure influence of a flat against a more distributed network architecture on network performance. This test is a follow up for the third research question and a step ahead to solutions proposed in chapter 7.

For all executed test the following statements hold:

- All tests will be performed using varying predefined amounts of active/online network nodes: (3, 10, 25, 50, 100, 250).
- All tests will be performed during varying predefined scenarios for different network activities and loads, see section 6.2.3.
- All tests will be performed at least five times and the results will be averaged in order to exclude coincidence.
- Results from different test runs are compared and checked if single tests do not differ too much from others to exclude failed tests due to external influences and increase test reliability.
- Every test will run at least for ten minutes.

- If applicable test will be performed over IPv4 and IPv6 so they can be compared.
- During a pure IPv4 test, IPv6 is disabled on the nodes and vice versa for IPv6 tests.

6.2 Test setup

There are two possible methods to perform experiments for this kind of research: using simulation or using real hardware. Using real hardware has the advantage of coming close to real world applications as equal hardware and software is used and environmental influences are real. A simulation can also come very close to the real world however required parameters and behaviour patterns should be first be analyzed and gained from real hardware usage. For my research I have chosen to use setup environment using real hardware.

In order to execute my research I have build multiple setups to perform different experiments and detect differences between them.

Test setup A
A single access point with a single subnet which uses a single subnet. This would be the most common deployed setup in small environments. See figure 6.1.

Test setup B
Three access points on three different wireless channels. All three use the same SSID and subnet forming a single network. Larger wireless networks in general use this setup. See figure 6.2.

Test setup C
Three access points on three different wireless channels. Each one uses a different SSID and a different (adjacent) subnet forming three separate networks with overlapping air and thereby their wireless medium. This setup has the same physical properties as setup B and can be used as reference to determine to impact of a single network against a separated network design. See figure 6.3.

All nodes are within a few meters of the access point and by that well in radio range of each other. During there were no other wireless networks on the same channel in connectable range.

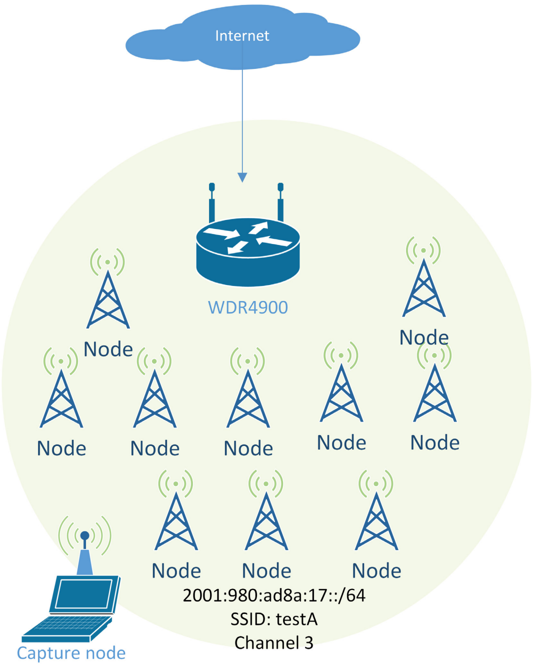


Figure 6.1: Test setup A: a single access point and single SSID

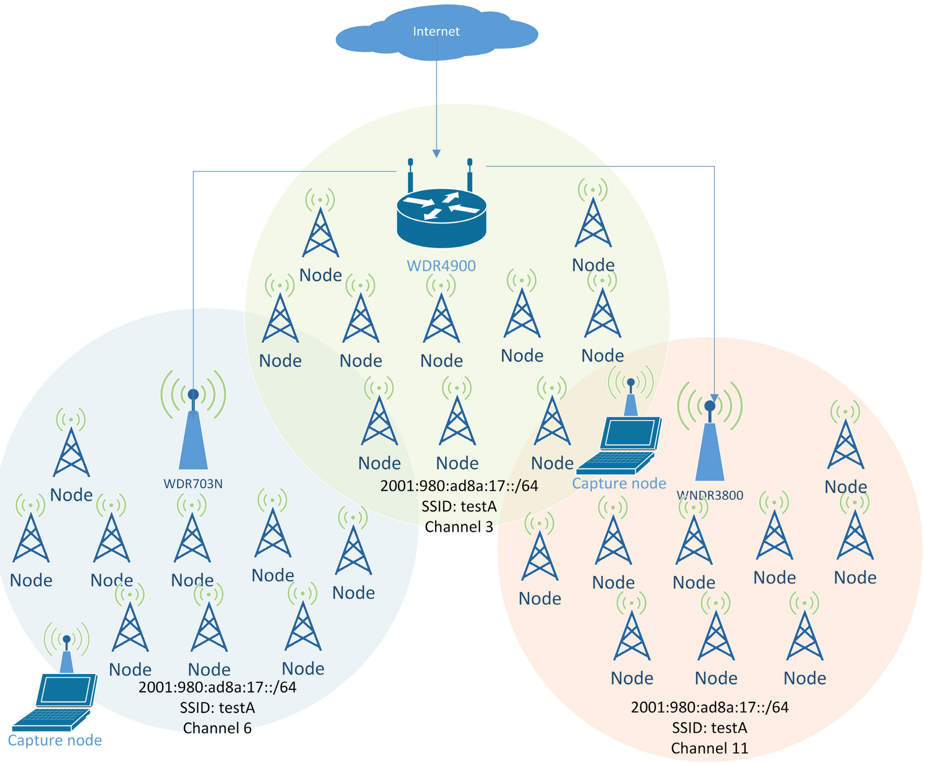


Figure 6.2: Test setup B: three access points and a single SSID

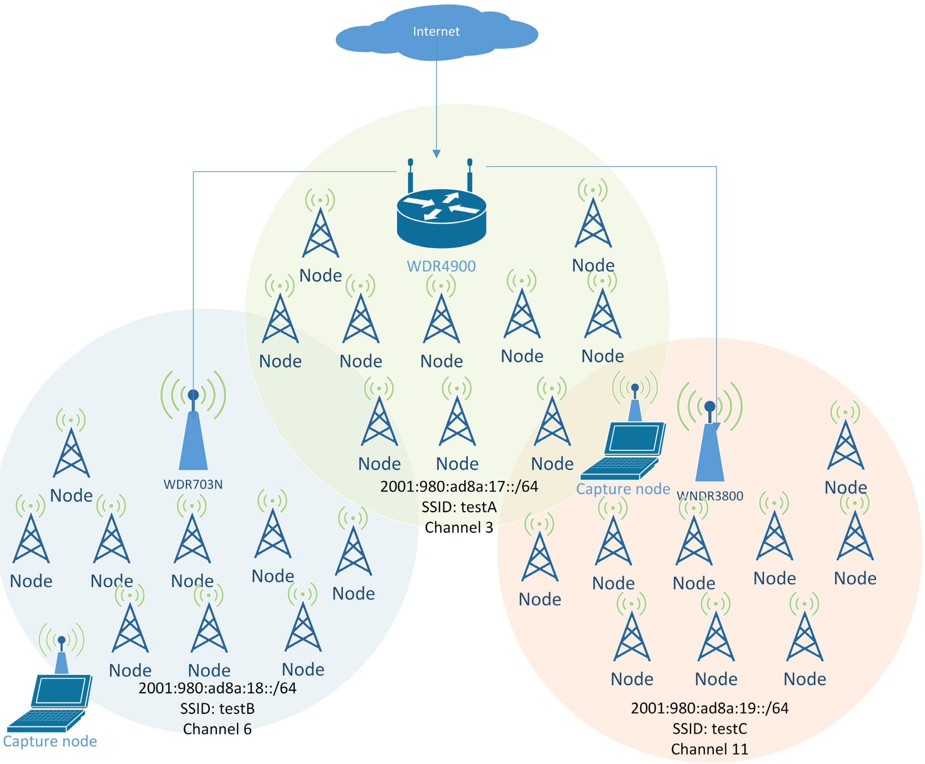


Figure 6.3: Test setup C: three access points using three SSIDs and three subnets

To construct the network setups illustrated in the figure 6.1, 6.2 and 6.3 I needed some hardware. In order to keep the required amount of systems low I used virtualization to achieve a high node count while preventing a high hardware bill. Using combination of a few hypervisor systems, some more USB hubs and a pile of WiFi controllers I created wireless networks with enough wireless nodes needed for my experiments. As I did not have enough WiFi controllers for every virtual machine I had to share them between multiple, up to five, virtual machines when I needed more then fifty nodes. Figure 6.4 sketches the hardware setup as used during the experiments; note this sketch does not include all hardware however includes all the different parts. See appendix A for more details on used hardware.

Servers connected by wired gigabit ethernet connections are used to serve test data files to the wireless nodes. During experiments their loads are monitored to make sure the performance bottleneck will not be these servers.

WiFi adapter sharing A relative simple benchmark have been performed to determine the performance influence of adapter sharing by measuring the usable bandwidth for multiple virtual machines using an adapter per VM (physical next to each other) against sharing a single adapter. This benchmark was performed using 2, 5, 10 and 20 concurrent virtual machines. A throughput performance reduction between 2% and 6% was measured. The WiFi network adapter only forms the lowest two, physical and data link, layers of the OSI stack and will not directly influence the higher layers formed by IPv4 and/or IPv6. The influence of network adapter sharing will be equal for both the IPv4-based and IPv6-based experiments and thereby not affect the comparison. During experiments traffic between virtual machines sharing an adapter is avoided in order to keep the sharing influence to a minimal.

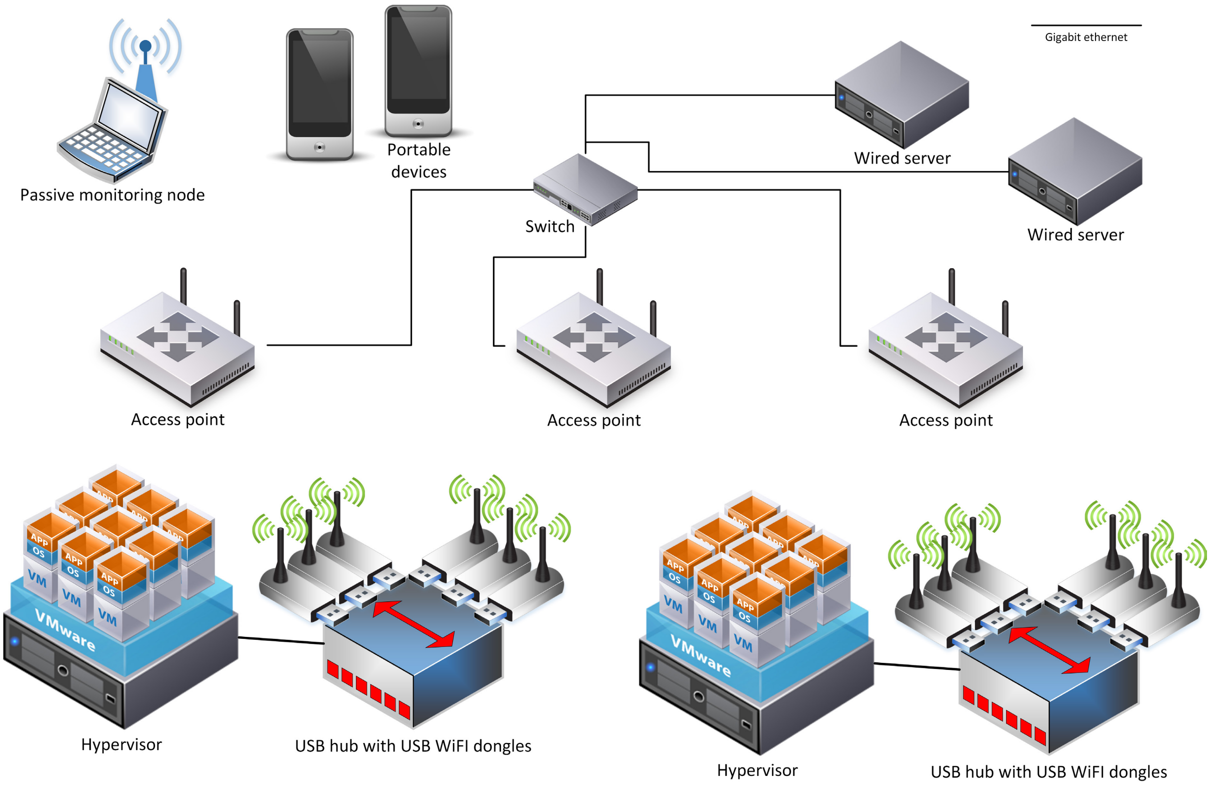


Figure 6.4: Overview of hardware setup for experiments

6.2.1 Measurement metrics

For the experiments I need to gather the following data:

- Packet count, accounting type (control vs data)
- Combined available bandwidth for data communication

6.2.2 Measurement methods

To get the result of an experiment you need to measure metrics which provide information about the outcome of the experiment. Not all my experiments require the same information to be measured. However analyzing the traffic on the wireless network was useful for all of them.

Operating modes During normal operation a network controller, both wired and wireless, only passes on packets of which it is the destination and drops the rest. With Ethernet and WiFi the destination MAC address must match the MAC address of the controller in this mode. In order to analyze traffic on the network all received packets are desired. For that purpose many network controllers offer different operating modes.

Managed mode The normal operation mode. The network controller is connected, associated to an access point in case of wireless, and provides only data targeted to the system.

Promiscuous mode The network controller will pass on every packet it receives. This is very useful for packet sniffing and analyzing purposes. Promiscuous mode is also used for virtualization purposes as virtual machines will use additional MAC addresses, unknown to the network controller.

Monitor mode Wireless controllers can support an operating mode that goes even further in passing data. Not every wireless chipset and/or driver supports this mode and even when supported the offered quality and reliability differs greatly. In monitor mode the wireless controller will pass on every (IEEE802.11) frame it receives. It does not need to be associated with an access point for this mode and will include frames from all SSIDs in range. As the wireless network controller will provide raw frames to the system, packets from a encrypted wireless network will not be decrypted by the wireless network controller.

To capture wireless frames/data I used wireless controllers which offers a well performing monitor mode, see Appendix A for an overview of used WiFi controllers.

The machine used for capturing raw frames was always a passive node in the network and by that did not participate in the network (traffic). It was always physically positioned for optimal reception.

Packet count While running the experiment a passive node captures all traffic its wireless adapter receives in monitor mode. After the run this data set can be analyzed using Wireshark.

Bandwidth Used bandwidth is measured on two places:

Server side The wired servers serve all the test data used in the experiments and thereby are a good measurement point for traffic.

Client side The nodes are the systems generating the real wireless traffic forming an ideal measurement point for traffic. It will be taking into account that traffic between nodes are measured twice and should only be counted once.

During an experiment run these two group entities measure their traffic using kernel statistics. After the run this data is collected and aggregated.

The amount of measured traffic on both measurements points always were very close to each other. To improve accuracy the average value of both measurement points was used. If the difference between both measurements was more than 5%, the data was discarded and the experiment was repeated. The combined bandwidth is calculated by dividing the total amount of traffic by the execution time of the experiment run. Note, the total amount of traffic includes all traffic that flows over the network during the test execution, this includes not only the data generated by the test scripts but also control data from the network stack.

Decryption A wireless network controller operating in managed mode will (in hardware or software) perform decryption of the received packets. Promiscuous mode shares this behaviour. In monitor the wireless network controller does not handle decryption and therefore captured data will still be encrypted. Decrypting packets can be done after capture if the encryption key is known. For wireless networks using WEP the encryption key can be derived from the pre-shared passphrase as every wireless node in the network shares the same encryption key. Each node will have its own encryption key when using WPA/WPA2. For WPA/WPA2 Personal the eapol handshake procedure has to be captured too in order to derive the encryption key for the node. This means the capture nodes should be already capturing when other nodes join the network. WPA/WPA2 Enterprise networks provide encryption keys based upon a secure authentication method, which prevents deriving the key.

Software I have used different software utilities to capture and analyze traffic, see Appendix B for an overview. Capturing data was done using tcpdump, Wireshark or airodump-ng in combination with airdecap-ng. After capture I analyzed the traffic/data using Wireshark.

WiFi is very broadly used and in a city environment its frequency range is often quite crowded. In monitor mode every received frame is passed on, even when the signal is too weak to connect, showing in general a large amount of wireless networks. Filtering on SSID is therefore very useful.

Parameters There are many configurable parameters for both WiFi and IPv6 and for example tuning the intervals and timeouts in the router advertisements[41] can make a difference. However there are many combinations of values and evaluating them would give months of work. Therefore unless required I did not change the default values of the operating systems and other used software (eg. daemons).

6.2.3 Test scenarios

I designed a few scenarios for the network nodes to differentiate the network usage. For all scenarios the available bandwidth is measured as a resulting sum of all test traffic across the participating nodes. Traffic between nodes is only counted once instead of counting both upload and download traffic.

Idle The nodes are online and are all, except one, fully idle (no user space activity). No additional commands are executed on the nodes. To measure the available bandwidth a single node will run iperf against a wired counterpart; thus only one wireless node will be active instead of idle. This scenario will be used to determine the impact of the amount of connected clients, without them performing activities.

Web browsing A script with some simple HTTP GET statements is executed on all the nodes. Sleep statements provide a random pause between them. This scenarios emulates a user browsing the web on each client. Only local HTTP servers are used to prevent external influences. This scenario represent the most common state a network with end users (as common for wireless networks) using the network for daily activities without single users stressing the network.

```
#!/bin/bash
declare -a urls=('1k' '10k' '100k' '500k' '1M' '5M' '10M');
declare -a servers=('thuus.lan' 'cnpi.lan' 'up.lan' 'mon.lan');
MAXSLEEP=30

while true; do
    sleeptime=$RANDOM
    let "sleeptime %= $MAXSLEEP"
    echo Sleeping for $sleeptime seconds
    sleep $sleeptime

    wget -O /dev/null http://${servers[$RANDOM % ${#servers[@]}]}/test/${urls[$RANDOM % ${#urls[@]}]}
done
```

Heavy traffic Like with the light traffic scenario a script is executed on all the nodes. In the high traffic scenario the pause times are lower and larger pages are included for the HTTP GET commands. It also connects other network nodes forming inter node connections. This scenario resembles a network that is very heavily used and constantly under pressure by multiple users and is used to determine what happens if the network is heavily stressed.

```
#!/bin/bash
declare -a urls=('1k' '10k' '100k' '500k' '1M' '5M' '10M' '50M' '100M');
declare -a servers=('thuus.lan' 'cnpi.lan' 'up.lan' 'mon.lan');
declare -a hosts=(list of reachable IPs, v4 or v6)
MAXSLEEP=3

while true; do
    sleeptime=$RANDOM
    let "sleeptime %= $MAXSLEEP"
    echo Sleeping for $sleeptime seconds
    sleep $sleeptime

    if [ $RANDOM -gt 21000 ]; then
        wget -O /dev/null http://${servers[$RANDOM % ${#servers[@]}]}/test/${urls[$RANDOM % $
            {#urls[@]}]}
    else
        IP=${hosts[$RANDOM % ${#hosts[@]}]}
        ping -q -c 1 $IP        # change to ping6 if dealing with IPv6 hosts
    fi
done
```

Network scan The node pings other nodes on the network. This will establish many light connections between nodes and fill up arp or neighbor tables. This can be done using a normal scan or an aggressive (very fast) scan. The behaviour of a network scan resembles background peer scans performed by various background services, for example: Netbios, Bonjour, uPnP Dropbox etc, and will form connections between many hosts within the network. This scenario is used to determine the impact of the amount of connected clients while forming many light connections (only scan, no data transfer) between them. For IPv4 this is done using the commands:

```
nping 10.13.0.0/24
zmap -M icmp_echoscan -B 25M -P 1 -T 100 10.13.0.0/24 -g -c 3   # fast
```

And for IPv6:

```
ping6 -B -I eth0 -I [global address of eth0] ff02::1
ping6 -f -B -I eth0 -I [global address of eth0] ff02::1   # fast
```

The fast methods soon found to be too aggressive to be useful. A single node performing such a scan already heated up all the CPUs let aside performing multiple scans at once.

6.2.4 Test execution method

The following execution sequence was common for all my experiments.

1. Prepare experiment at hand by configuring for example the access points
2. Power up access points
3. Start capture nodes in monitor mode
4. Boot network nodes
5. Wait till boot processes complete and all nodes fade into idle state
6. Connect a shell to all participating nodes (using clusterssh)*
7. Execute test script on all needed nodes*

8. Wait pre-determined period or until all tests are finished
9. End test script*
10. Stop capture and save both capture and results to disk
11. Gather traffic usage from nodes and wired servers
12. Analyze results

** In idle scenario only on a single node (iperf)*

6.3 Results

In this section I will expand my results and discuss them. All the experiments combined gave a lot of data which I tried to present in a clear way in tables and diagrams, see table 6.1 for an overview[1].

Table 6.1 Overview of tables and figures with results

	Numbers	Idle	Light traffic	Heavy traffic	Network scan
ARP vs NDP	6.2	6.3.1	6.3.1	6.3.1	6.3.1
Bandwidth	6.3	6.3.2	6.3.2	6.3.2	6.3.2
Border node	6.4	6.3.3	6.3.3	6.3.3	6.3.3
Divide nodes	6.2, 6.3,	6.3.1, 6.3.2, 6.3.4	6.3.1, 6.3.2, 6.3.4	6.3.1, 6.3.2, 6.3.4	6.3.1, 6.3.2, 6.3.4

6.3.1 ARP vs NDP

Tested in test setup B and C as mentioned in the table header.

[1] The table is in particular useful when viewing a clickable PDF.

Table 6.2 ARP and NDP packets/minute

Scenario	Nodes	ARP B	NDP B	ARP C	NDP C
Idle	3	7	31	7	29
Idle	10	18	42	21	41
Idle	25	26	84	32	72
Idle	50	37	184	44	157
Idle	100	57	264	63	211
Idle	250	89	592	91	380
Light browsing	3	37	61	33	56
Light browsing	10	168	192	156	171
Light browsing	25	276	334	239	279
Light browsing	50	537	684	482	594
Light browsing	100	1057	1264	917	1024
Light browsing	250	2589	4342	2214	3742
Heavy traffic	3	104	248	98	187
Heavy traffic	10	454	1226	436	1005
Heavy traffic	25	718	2070	680	1495
Heavy traffic	50	1396	3866	1321	2781
Heavy traffic	100	2960	7505	2820	5225
Heavy traffic	250	7249	24382	6824	18057
Network scan	3	672	1530	663	1521
Network scan	10	2360	16067	2400	10771
Network scan	25	5700	103542	5925	60754
Network scan	50	11531	*	11550	257394
Network scan	100	22424	*	23801	*
Network scan	250	*	*	59238	*

* too much for the network to handle; nodes started to lose connectivity.

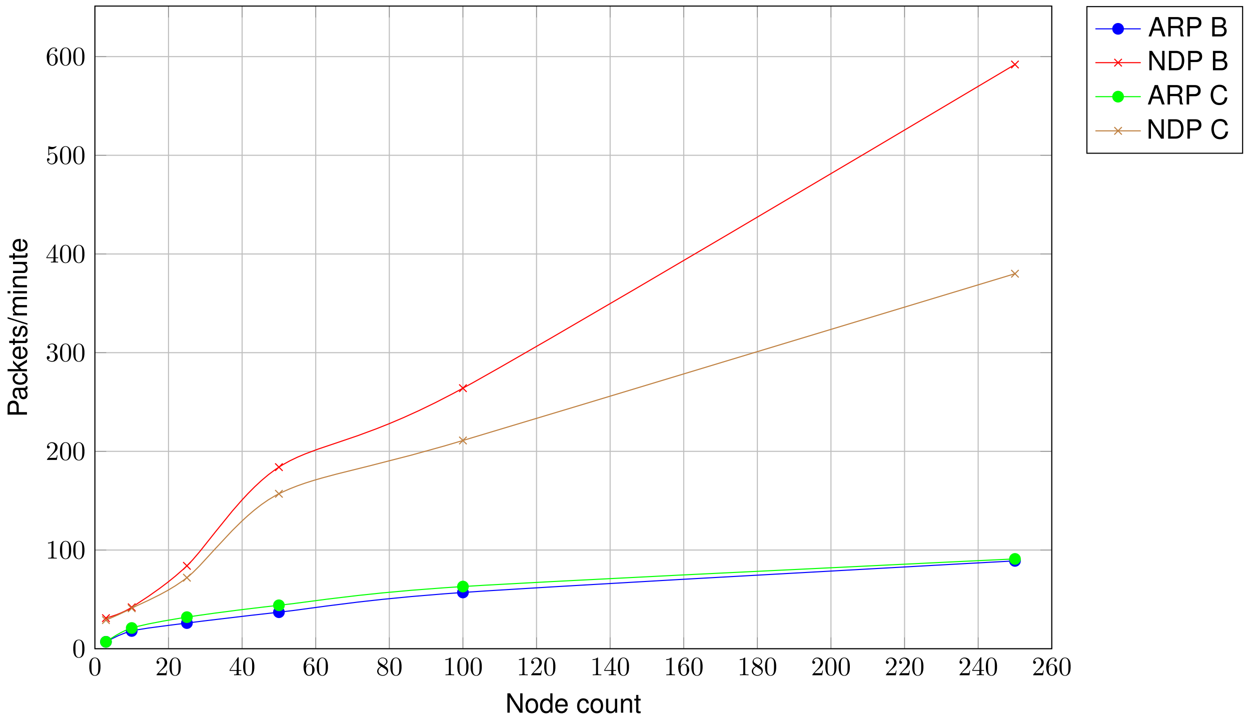


Figure 6.5: Amount of ARP and NDP packets per minute in idle state

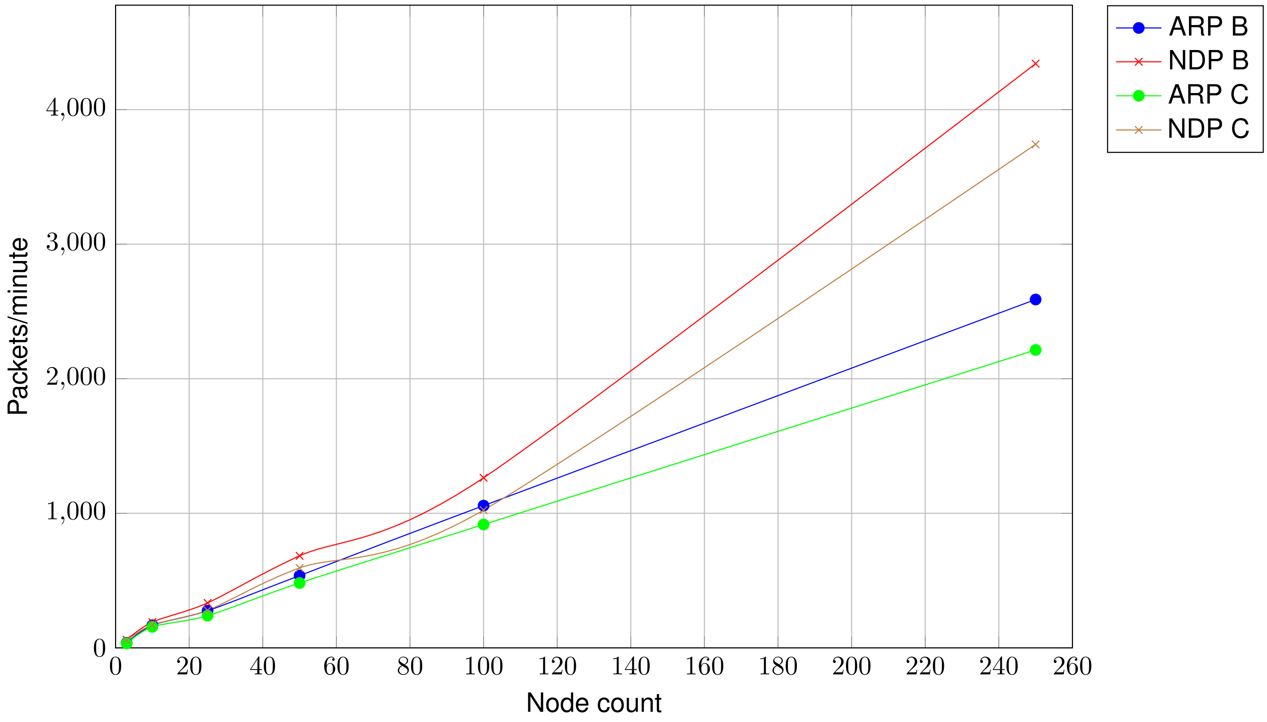


Figure 6.6: Amount of ARP and NDP packets per minute during light traffic (browsing)

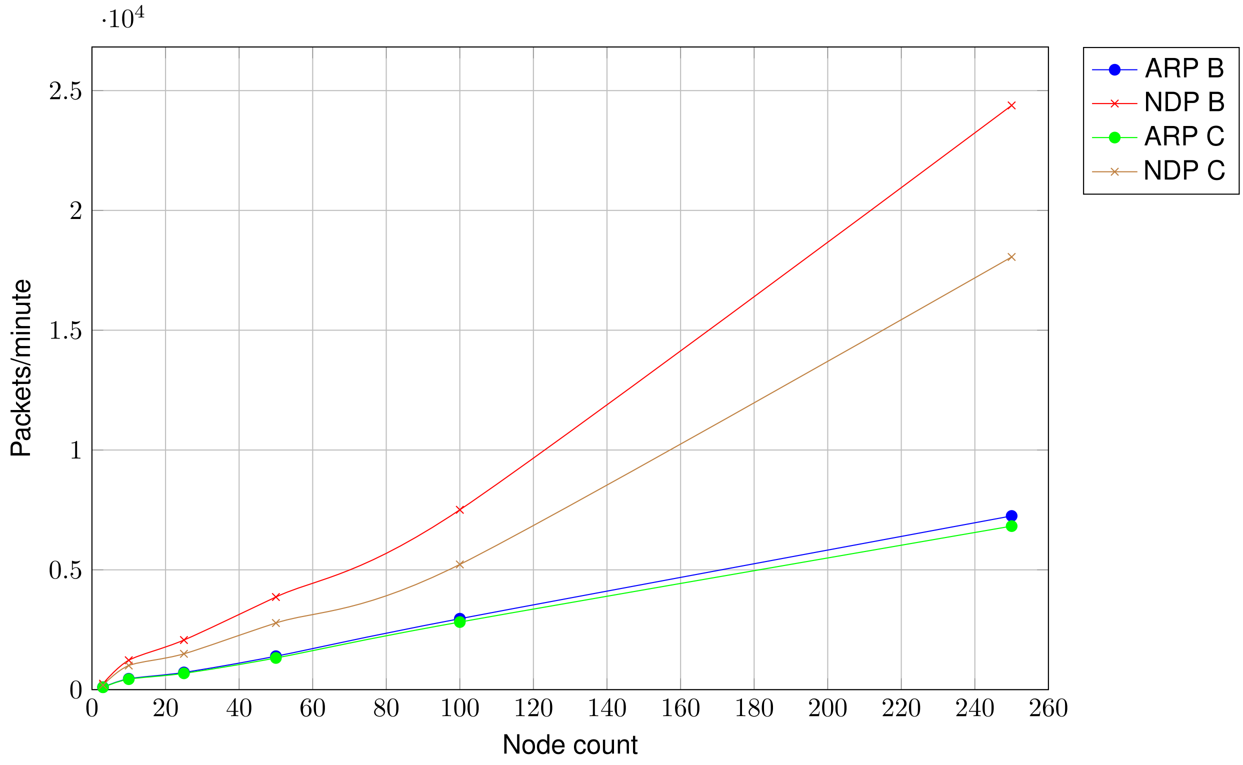


Figure 6.7: Amount of ARP and NDP packets per minute during heavy traffic

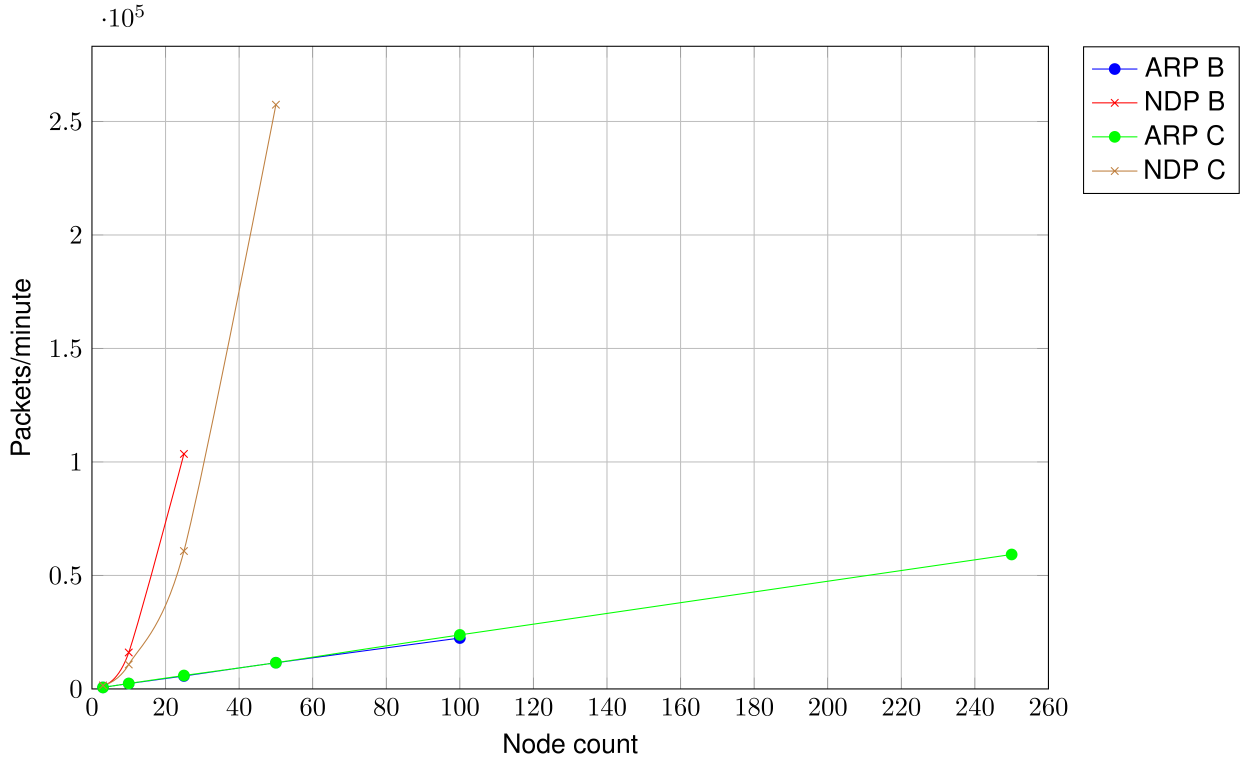


Figure 6.8: Amount of ARP and NDP packets per minute during network scan

Conclusion In general the amount of control packets in idle state is much larger in IPv6 compared to IPv4. However the absolute amounts relative to the node count is not very concerning. The difference in the light traffic test is much smaller, but start to grow with a large node count. Equal result is seen in the heavy traffic test. In idle state the nodes do not really know about the amount of nodes in the network and not many control packets are needed to obtain the required information for their network surroundings. When the amount of connections grow so do the ARP or Neighbor tables. In idle state only background services who scan for their peers will add connections. This causes the IPv6 stack to start performing many DAD checks. All those multicast packets start to flood the wireless network. The huge amount of packets also had a clear impact on the CPU load of the hypervisor systems. The network scan obviously rapidly fills ARP and neighbor tables making IPv6 generate huge amount of control packets (over)flooding the network.

As suspected by the formulas in chapter 5.2 the NDP packet rate is above the ARP packet rate.

6.3.2 Bandwidth

Tested in test setup B and C as mentioned in the table header.

The wireless network should offer 144Mbps bandwidth according to the link speed indicated between the nodes and the access point. A single node can not exceed this value, however in test setup B and C you can potentially reach three times this value over all nodes as three access points are used.

For this experiment the scenarios describe the background activity beside the test scripts. The bandwidth used by the test scripts in the light browsing and heavy traffic scenarios is included in the total amount of bandwidth. See section 6.2.2 for measurement methods.

Table 6.3 Combined bandwidth in Mb/s

Scenario	Nodes	IPv4 B	IPv6 B	IPv4 C	IPv6 C	IPv4 G	IPv6 G
Idle	3	164	178	170	181	4%	2%
Idle	10	192	184	191	194	-1%	5%
Idle	25	49	51	112	108	129%	112%
Idle	50	36	36	42	47	17%	31%
Idle	100	26	26	34	33	31%	27%
Idle	250	20	20	27	26	35%	30%
Light browsing	3	128	114	174	179	36%	57%
Light browsing	10	33	31	72	94	118%	203%
Light browsing	25	13	12	41	43	215%	258%
Light browsing	50	8	9	23	21	188%	133%
Light browsing	100	5	5	9	9	80%	80%
Light browsing*	250	4	3	7	4	75%	33%
Heavy traffic	3	88	79	142	145	61%	84%
Heavy traffic	10	50	48	95	92	90%	92%
Heavy traffic	25	14	15	34	28	143%	87%
Heavy traffic	50	7	7	19	14	171%	100%
Heavy traffic*	100	4	3	12	5	200%***	67%
Heavy traffic*	250	2	1	4	3	100%***	200%***
Network scan	3	68	63	132	138	94%	119%
Network scan	10	32	28	67	61	109%	118%
Network scan	25	28	14	61	56	118%	300%***
Network scan*	50	12	12	23	14	92%	17%
Network scan*	100	4	3	9	5	125%	67%
Network scan*	250	1	0	3	1	**	**

Test results are rounded to Mb/s.

The G columns contains the gain (improvement) of using test setup C over B.

* even when nodes started to lose connection due to heavy network load a combined bandwidth could still be measured.

** The numbers will not give a representative gain percentage.

*** The precision of gathered measurement data (megabits as integer) may give a flattened percentage value. Especially visible on, yet not restricted to, indicated values.

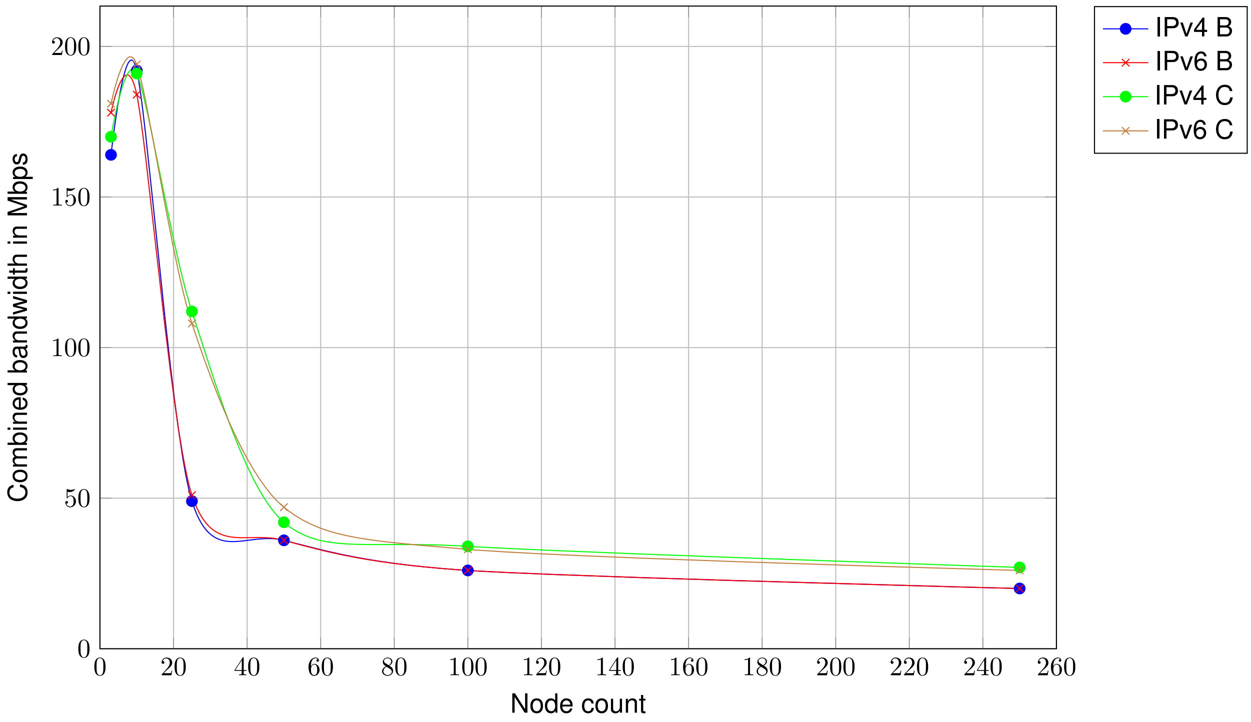


Figure 6.9: Combined bandwidth in Mbps/s in idle state

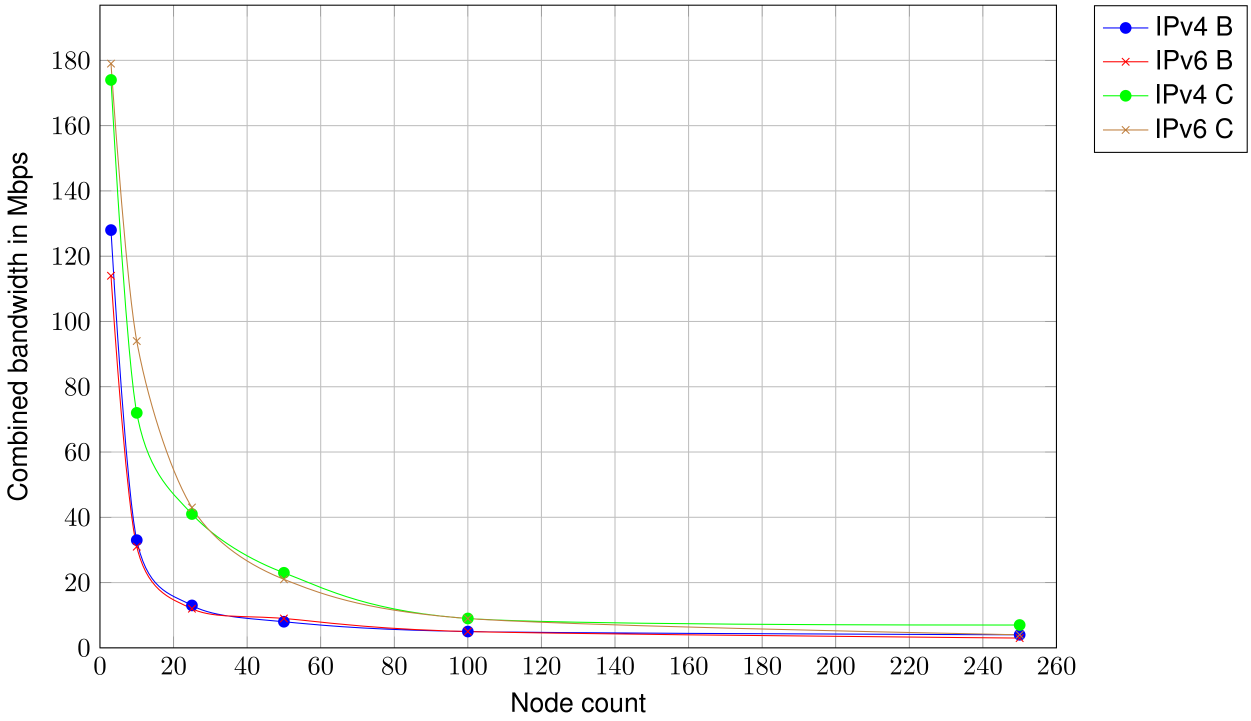


Figure 6.10: Combined bandwidth in Mbps/s during light traffic

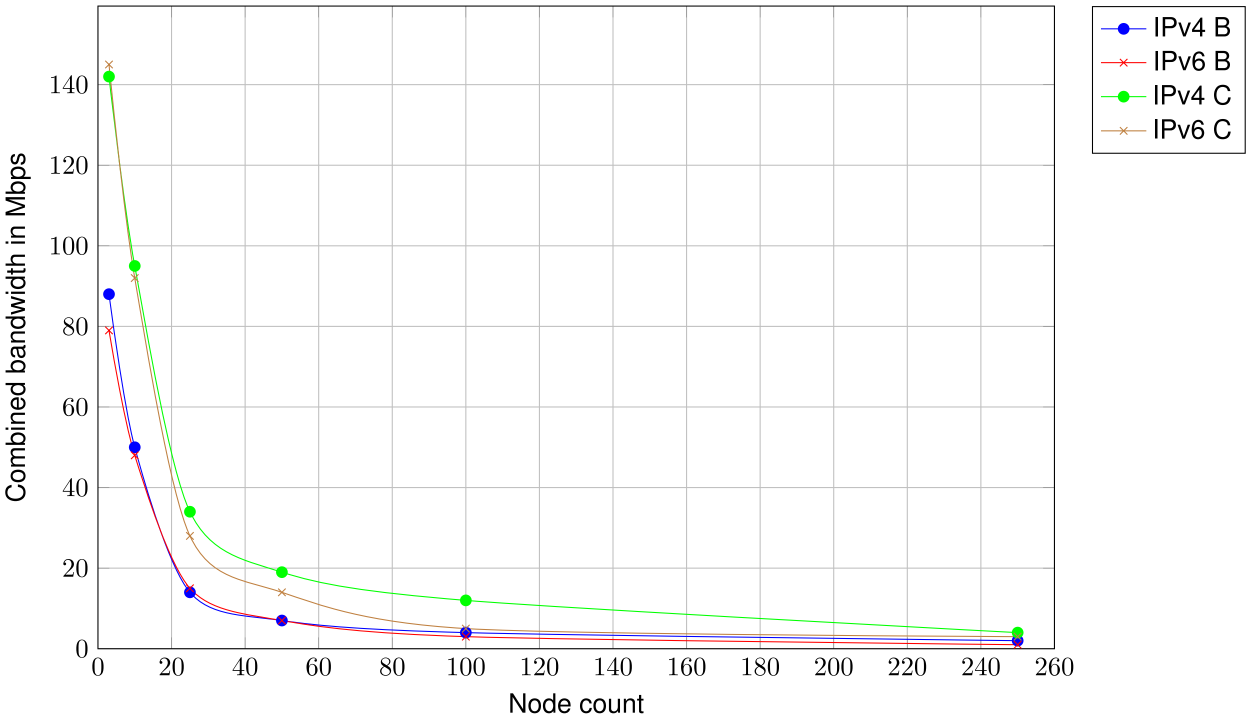


Figure 6.11: Combined bandwidth in Mbps/s during heavy traffic

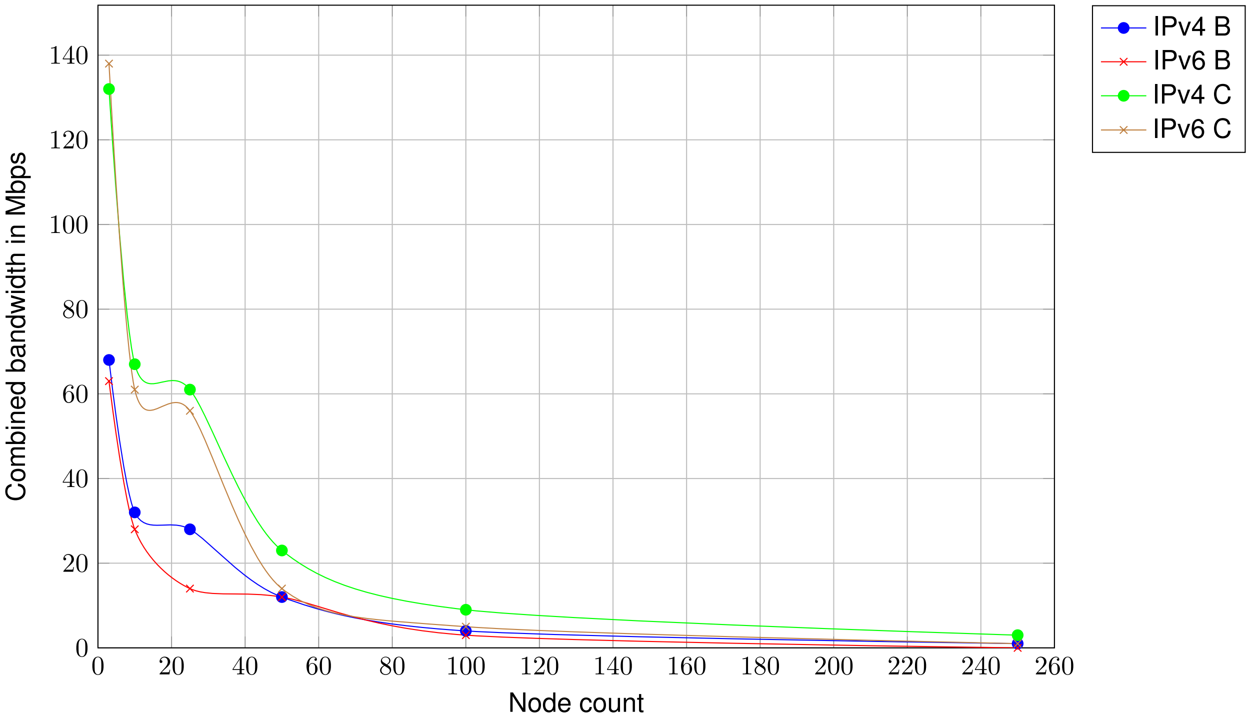


Figure 6.12: Combined bandwidth in Mbps/s during network scan

Conclusion In general the differences between IPv4 and IPv6 performance are small and negligible. IPv4 gains an advantage over IPv6 when the node count grows. The effect of filling ARP and neighbor tables seen in the first experiment can also be seen in this one. The shared medium of the wireless network can not be used very efficiently when so much transmissions 'pollute' the air. The overall bandwidth when many nodes are connected will not provide users a satisfying network experience. Of course in this test all nodes are equally trying to use the network, which in the real world rarely occurs.

6.3.3 Border node influence

In this test one of the nodes was positioned at the border of the wireless range. The low signal quality made only the lowest supported data rate possible for this node, 1Mbps. The previous experiment is repeated in test setup A. All nodes will be sharing the same access point in this test setup, maximizing the influence of the border node.

Connecting 250 nodes was too much for a single access point to handle, so for this experiment the 250 node test is skipped.

Table 6.4 Combined bandwidth in Mb/s with a node at the border

Scenario	Nodes	IPv4 N	IPv6 N	IPv4 B	IPv6 B
Idle	3	105	106	87	89
Idle	10	74	72	68	65
Idle	50	19	20	16	7
Idle	100	8	5	8	1
Light browsing	3	91	90	94	91
Light browsing	10	70	72	57	54
Light browsing	50	20	19	14	8
Light browsing	100	8	11	7	1
Heavy traffic	3	81	79	90	75
Heavy traffic	10	63	58	45	29
Heavy traffic	50	18	17	13	2
Heavy traffic	100	4	12	6	0
Network scan	3	64	68	58	53
Network scan	10	28	24	12	13
Network scan*	50	2	1	1	0
Network scan*	100	0	0	0	0

N all nodes within close range

B one node on the border

* even when nodes started to lose connection due to heavy network load a combined bandwidth could still be measured.

Test results are rounded to Mb/s.

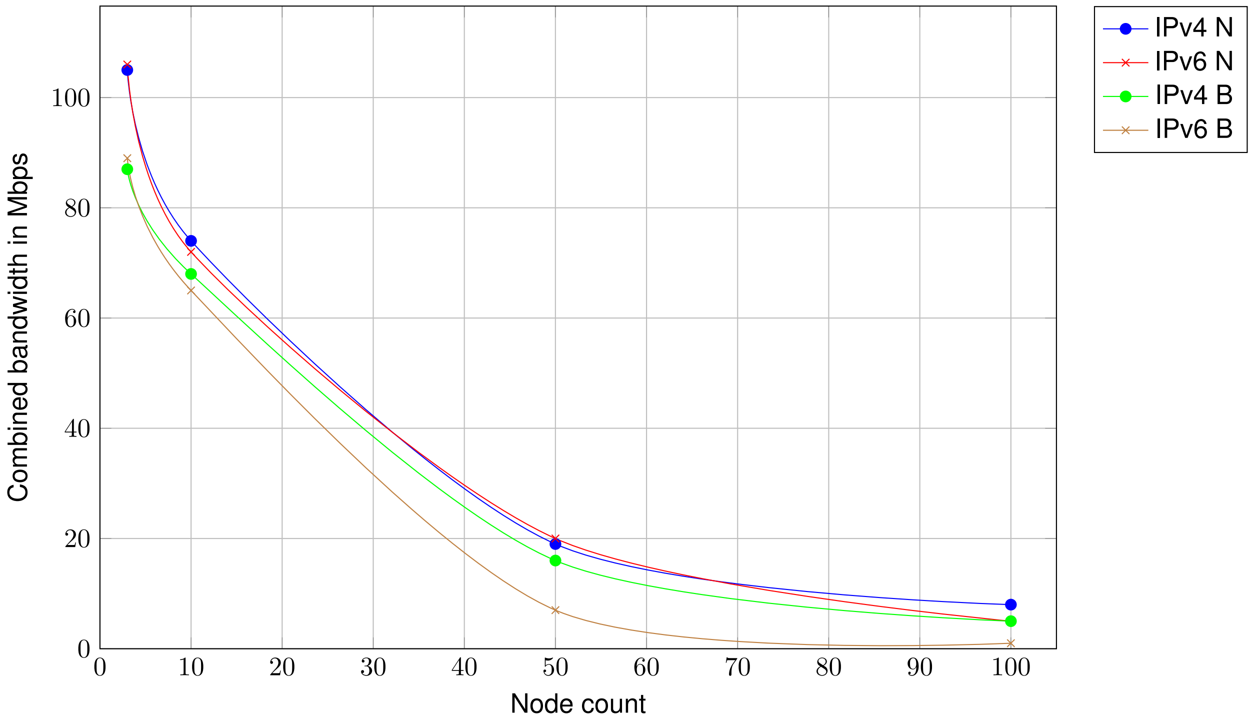


Figure 6.13: Combined bandwidth in Mbps/s with a node at the border in idle state

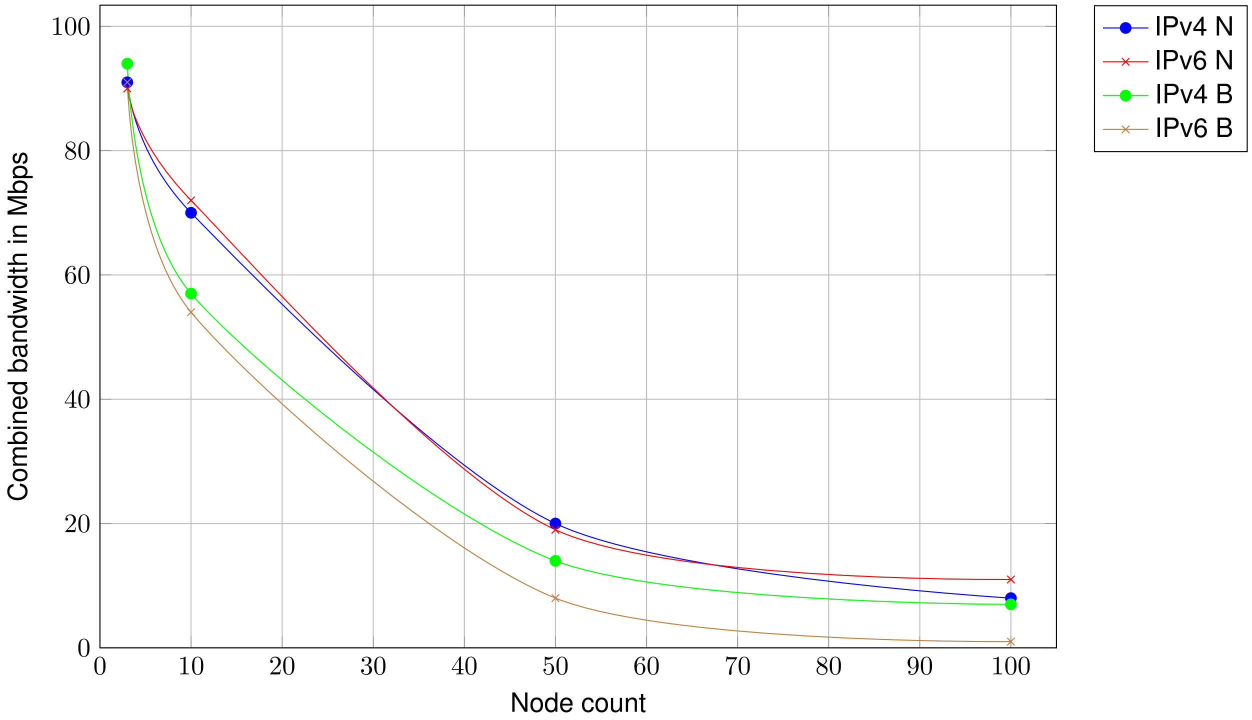


Figure 6.14: Combined bandwidth in Mbps/s with a node at the border during light traffic

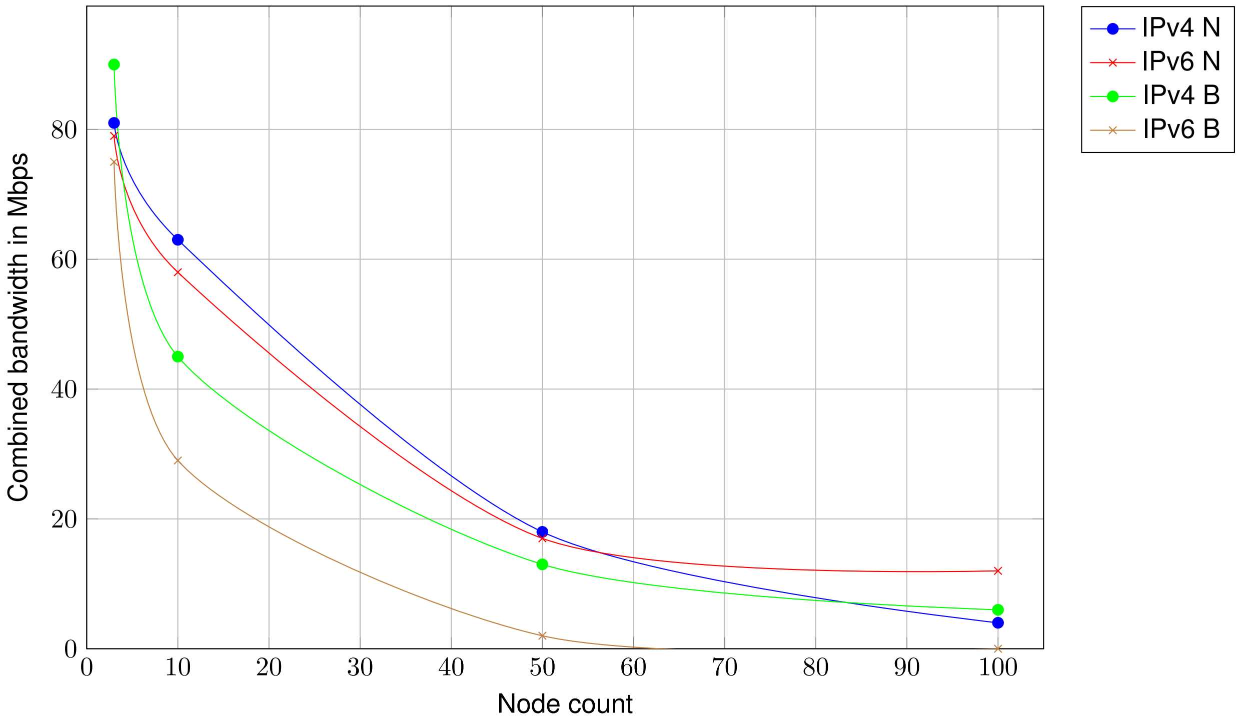


Figure 6.15: Combined bandwidth in Mbps/s with a node at the border during heavy traffic

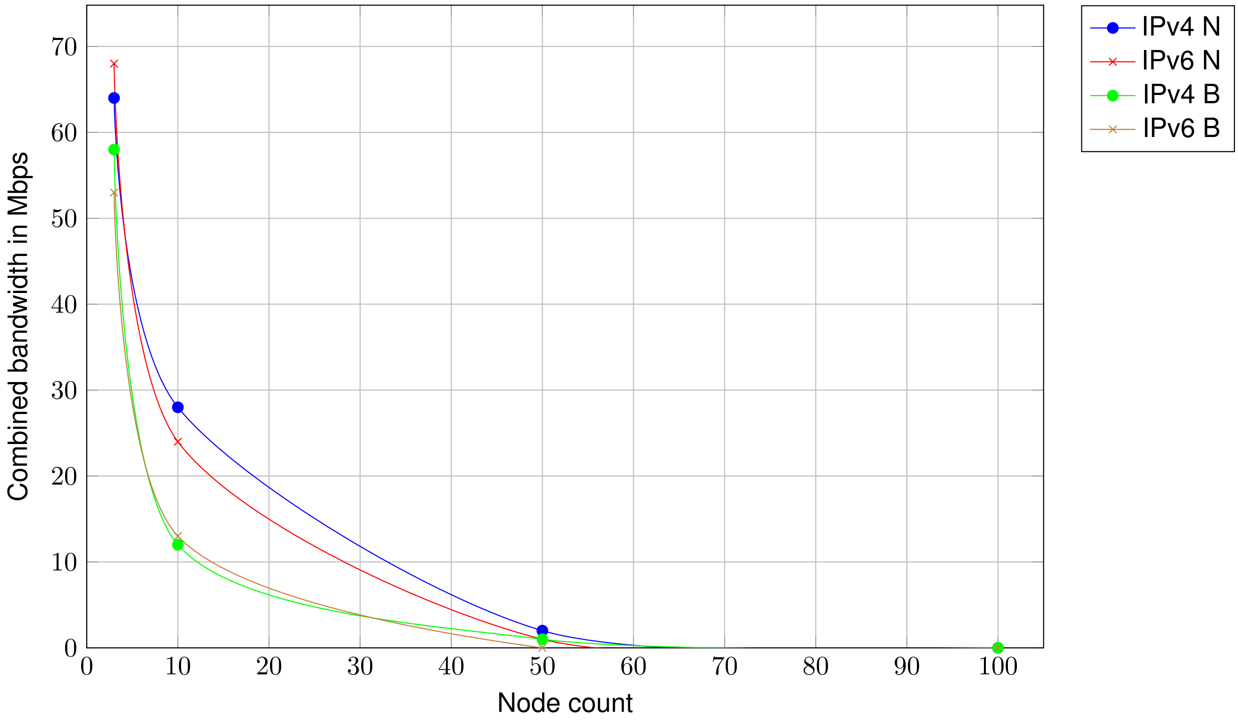


Figure 6.16: Combined bandwidth in Mbps/s with a node at the border during network scan

Conclusion The effect of broadcast messages forcing the entire network to use a slow data rate when a single node requires it has a strong effect on the bandwidth test. As seen in the first experiment the amount of multicast/broadcast control packets is strongly correlated to the node count Especially the multicast control traffic generated by IPv6 cripple the network performance as a great part of available air time is consumed by those multicast packet transmissions. The IPv4 network does not shine in this experiment but has a distinct advantage over IPv6 with a large amount of nodes.

6.3.4 Divide nodes

The ARP vs NDP and Bandwidth tests already included results of test setup B and C. To clarify the effect on the available bandwidth the following diagrams illustrated the gain percentage measured in the bandwidth test.

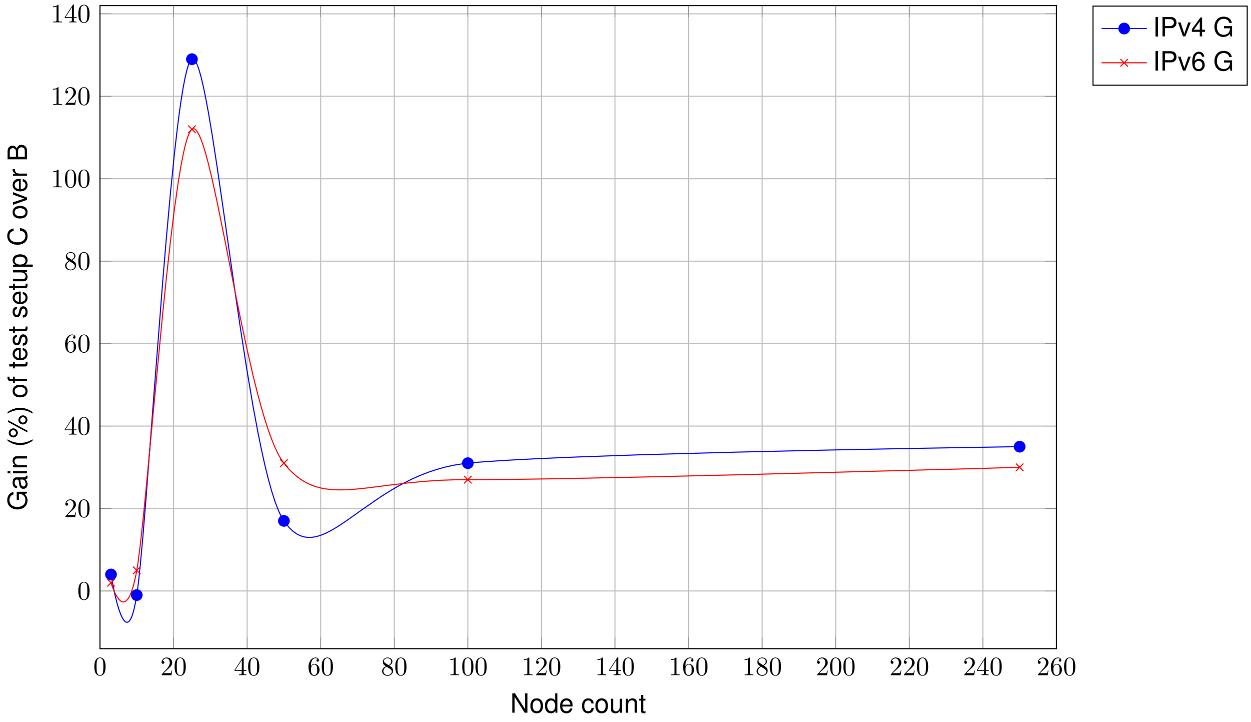


Figure 6.17: Gain (%) of using test setup C over B in idle state

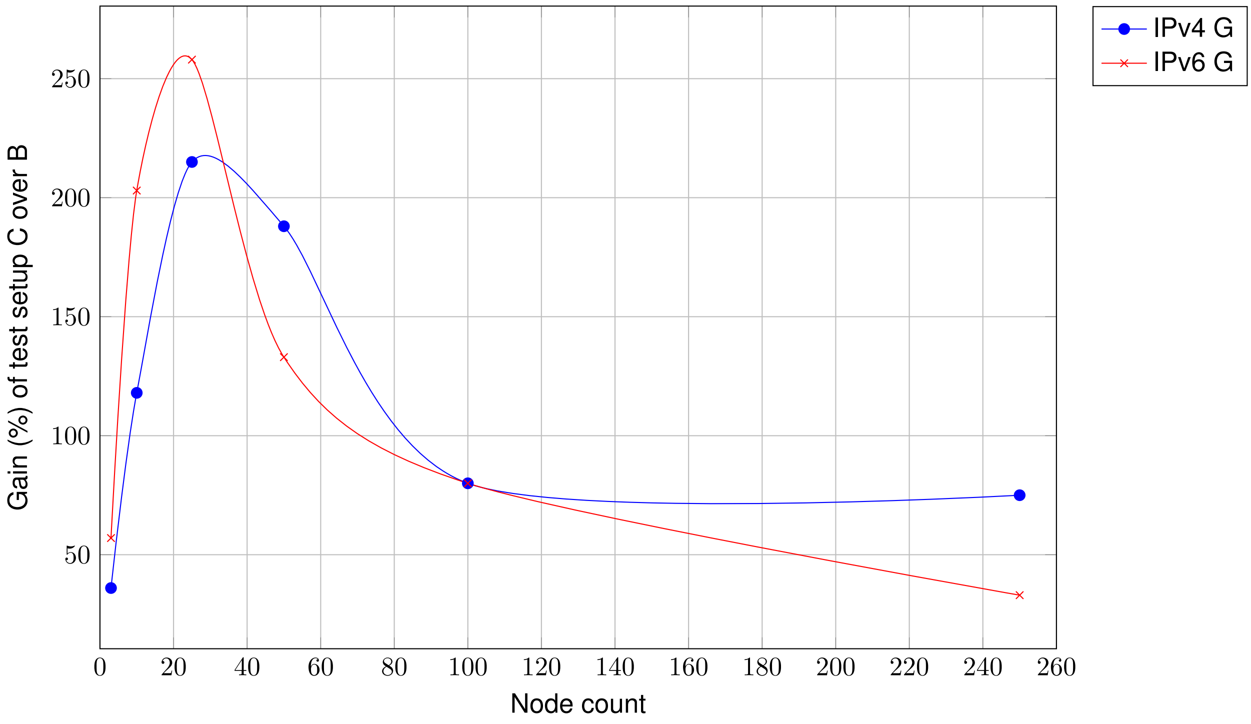


Figure 6.18: Gain (%) of using test setup C over B during light traffic

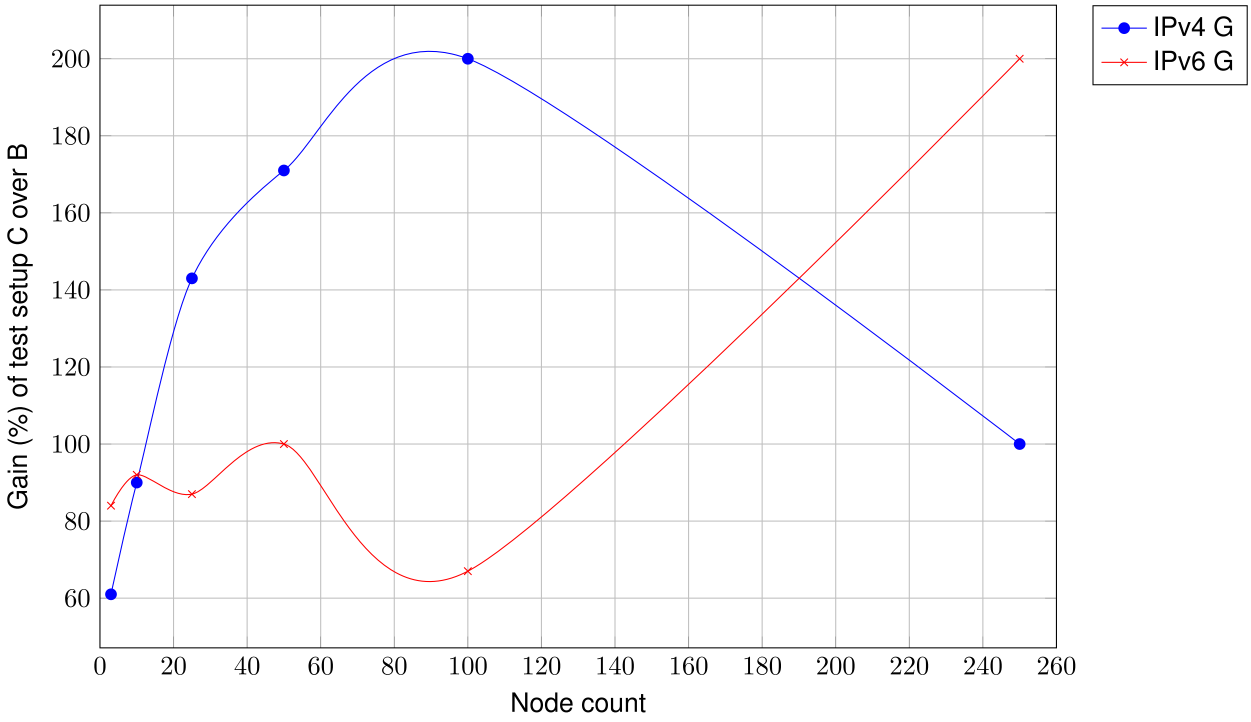


Figure 6.19: Gain (%) of using test setup C over B during heavy traffic

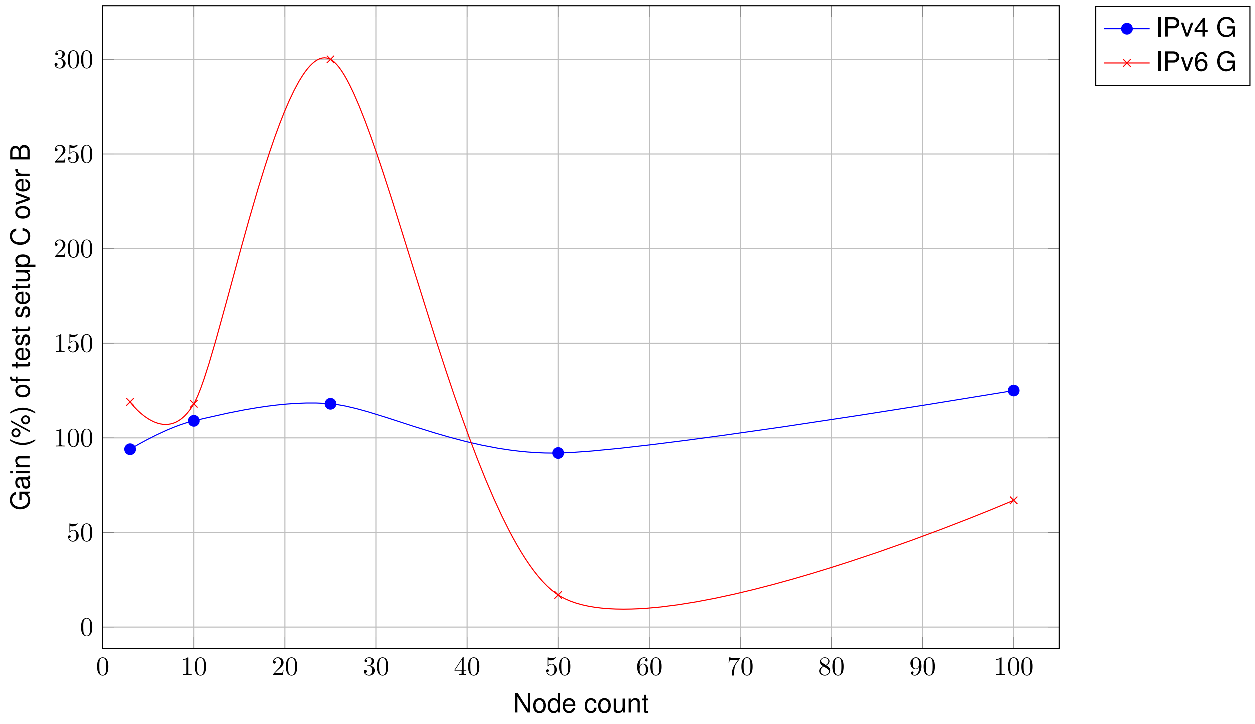


Figure 6.20: Gain (%) of using test setup C over B during network scan

6.3.5 RA parameter tuning

Like mentioned in sections 4.2.1 and 6.2.2 there are many configurable parameters in the system but evaluating them gives tons of extra work. I did experiment a bit with some radvd configuration parameters from the stateless autoconfiguration RFC[41]. Some of them did make a difference in the retransmission rate of router advertisements. Non default values did not always result in a stable IPv6 network; I suspect a bug in the daemon software as it was often restored by rebooting the radvd daemon on my primary router. This could form an interesting point to investigate deeper however I quickly discovered that most operating systems do not respect most of those extra parameters given by the router advertisements and thus will not make a noticeable difference in the entire network.

Conclusion Using multiple access points basically separates the shared medium as they use different, non overlapping, channels. That causes the total available bandwidth to grow and that can clearly be seen in the results. The amount of nodes needed to saturate the network increases. The advantage does not rise from the results if the node count is too small as they do not require the extra bandwidth provided by multiple access points. Around 25 total nodes all the access points, serving a third of the nodes, a balance is found between the bandwidth offered by the access points and the bandwidth consumed by the nodes. Besides dividing the network over multiple channels the network is also divided into multiple subnets. This causes the actual network size as experienced by a node to decrease. As seen in the previous experiments, the control packets of IPv6 grow faster (not linearly) with an increasing node count which causes IPv6 to greater benefit the division into multiple subnets.

6.4 Identified issues

Reviewing the results of the previous sections I can summarize the findings with the following overall issues.

Broadcast lowers data rate

For all broadcast transmissions (including multicast) the lowest data rate of the associated nodes on the access point is used causing a major drop in performance in the entire wireless cell. The impact of this issue became very clean in the 'Border node influence' experiment in section 6.3.3.

Large amount of multicast traffic

The control protocol of IPv6 generates a lot of traffic which is almost all send as multicast (which becomes broadcast) traffic. This rapidly increases with a increasing amount of nodes in the network. Even nodes that are not actively using their network connection contribute in this behaviour. This issue became clear from the 'ARP vs NDP', see section 6.3.1, and 'Bandwidth', see section 6.3.2, experiments.

Note that the impact of the second issue can be amplified by the first issue.

CHAPTER 7

PROPOSED SOLUTIONS

Section 6.4 states the identified issues when deploying IPv6 on a wireless network. In the following sections a few possible improvements are proposed to provide a solution.

I will only propose these solutions as implementing and testing of (some of) them may form a basis for an entire research project.

7.1 Broadcast causing overall low data rate

Lowering the minimal data rate setting on access points will decrease loss of performance due to broadcast transmissions but also reduce wireless coverage and range. This setting becomes more important on a IPv6-enabled network and should not be forgotten. An optimum value should be a weighted decision by the network administrators. Aside from this there is no real solution for this problem.

7.2 Reduce multicast traffic

The excessive use of multicast traffic is largest issue and by that offers most space for improvements. In the first subsection some small improvements will be listed followed by some more complex solutions will in the other subsections.

7.2.1 Small improvements

Some small improvements within the current protocol specifications. These can be considered for every network without thorough changes required.

Reduce interval of unsolicited router advertisements

The maximum allowed value is 9000 seconds.[34], which might be too long for certain networks. However too small values only generate extra multicast traffic. No new information is given by these messages to already connected nodes and connecting nodes can always send a router solicitation message.

Unicast solicited router advertisements

The specification[34] already allows (by using the verb 'may') routers to reply using unicast opposed to the default multicast response. In practice this is not performed by standard network equipment.

Increase advertised reachable value

The default value for Neighbor Unreachability Detection is only 30 seconds[34]. After this timeout the address will be removed from the ND caches and rediscovered on subsequent traffic. Increasing this value to a (much) higher value will decrease the amount of solicited neighbor discovery. A possible disadvantage of a high value is a higher risk of overflowing the ND table on routers.

Implementing all these solutions will reduce the amount of multicast traffic. However in contribute enough to solve the problem. Yet every small step counts.

7.2.2 Infrastructure based solicited-node multicast filtering

Multicast is intended in contrast to broadcast to be transmitted only to the interested nodes. The most common solutions for this is MLD snooping, see section 4.3.2. However like stated before in section 4.3.2 MLD snooping often fails in larger networks. It can also be achieved by using a mapping from a multicast destination on a link layer address as proposed by RFC 6085[16]. This however not yet bought into practice. Using either one of these methods enable the wireless network infrastructure to determine where the interested nodes are and with that information transmit only on the required access points instead of broadcasting it over the entire (wireless) network. It is up to the network vendors to provide a working implementation for this. Wireless network controllers may provide a centralized administration role here.

7.2.3 Advertise on-link bit to zero

This will cause all traffic between wireless node to go through the default gateway. Therefor no longer requiring information about its neighbors on the global scope. The remaining control traffic for the link local scope can be mitigated by blocking direct communications between nodes on the access point.[45] Most of the traffic on a wireless network, such as Eduroam, will already require routing to destinations outside the wireless network as in general there are no servers on the wireless network. Routing to wired parts of network and the off-premise internet is common practice. Traffic between wireless nodes will give extra load for the router. This however will in general a relative small amount of traffic and will not form an issue. Router advertisements contain a list of prefixes used for on-link determination. Flags associated with the prefix specify the intended uses of a particular prefix.[34] Connected nodes use this information to decide whether a packet's destination is on the same link or beyond a router. This solution seems very promising as it will greatly reduce multicast traffic while requiring only a small change in configuration within the standard specifications and a little more processing power on routers.

Configuring this option should be trivial. The common used radvd, see appendix B, daemon has the option 'AdvOnLink' for this. However I did not get this actually working on the primary advertised global prefix was the setting seemed to get ignored on that prefix. Configuring it on an additional prefix did result in the flag to be set. However as the primary prefix was still advertised as on-link this does not have the desired effect. The documentation in the source code states 'The both currently defaulted to enabled but are included here for introductory purposes.'. I will expect future releases to actually support this option.

7.2.4 Unicast responses to neighbor solicitation

Unicast responses from the access point to neighbor solicitation from connected nodes. Let the access points, or their centralized controllers, use information from their ND tables to quickly respond using unicast instead of spreading the request over the network. This will require modification of access point firmware. Section 8.1 describes my attempt on this.

7.2.5 Single multicast group per AP

Let all node associated with an access point share a single multicast group for neighbor discovery. Prevent MLD activities on both the node as the access point as the association with a certain access point defines the group membership. This simplifies the solicited node process. The amount of multicast groups will be decreased, namely dived by the (average) count of associated nodes per access point. In order achieve the protocol needs to be altered. This will require a large amount of work on both client as infrastructure side.

7.2.6 Segment network into subnets

A slight variant on the previous solution. Most multicast traffic is between neighboring nodes on the network. More segmentation of the (wireless) network will reduce the amount of neighbors and should also reduce the amount of multicast traffic. A good start will be a single subnet per access point. Multicast traffic will still be come multicast traffic per access point. However due to the nature of the shared medium this can not be overcome completely anyway. The results of test setup C, see section 6.2, showed the performance gain of solution.

This idea can be extended to the proportion that every node has its own subnet. Section 8.2 describes my attempt on this.

7.2.7 Restrict scanning services

Keep connections between wireless nodes to a minimum to prevent them to require and acquire information about their neighbors.. Prevent the use of discovery services like Bonjour, Netbios and uPnP, see section 3.2.1. It is often not possible to regulate this kind of traffic on all the network clients as network administrator do not control every wireless client. In the network infrastructure it is however possible to rate limit or block certain types of traffic using ACL rules on routers and/or access points.[43]. This can reduce the amount of broadcast traffic generated by clients while in general not reducing network functionality.

CHAPTER 8

FAILED EXPERIMENTS

There are two experiments I wanted to perform but which I unfortunately did not succeed in.

8.1 ND caching on AP

As seen in previous sections neighbor discovery broadcasts can become a burden on the wireless network. I wanted to implement the solution proposed in section 7.2.4.
In a larger network with many access points it could become a great benefit if an access point would not blindly broadcast all received neighbor solicitations and advertisements. Only those applicable for its direct connected nodes should pass. Neighbor solicitations messages could also directly be answered by a unicast response from the access points if it would keep track of an extended neighbor table for the entire network instead of only its own records.
Looking at the increasing amount of control packets seen in table 6.2 this mechanism could greatly reduce broadcast traffic in larger networks. Especially when wireless nodes are primarily connecting through the gateway instead of each other. The actual performance gain is hard to estimate.

The following behaviour steps describe my target solution.

1. A nodes perform regular neighbor discovery, sending out a neighbor solicitation
2. The neighbor solicitation is spread over the network
3. The other initially replies to the neighbor solicitations
4. Access point use snooping to fill its neighbor tables
5. Again a regular neighbor solicitation is send out
6. The access point recognizes the requested address and will not propagate the packet
7. The access point instead replies with a neighbor advertisement using unicast

I underestimated the complexity of the networking stack on Linux based access point software like OpenWRT. Modifying its behavior to include caching for neighbor discovery was not straight forward. It would have taken far too long to implement this and can be major project on its own. Furthermore it would be better to implement this in a centralized way, wireless access point controllers would be the ideal place for it.

8.2 Tiny subnets for clients

To implement the extended version of the proposed solutions of section 7.2.6 I again tried to modify access point firmware. If every wireless node resides in its own tiny subnet all the neighbor discovery stays between a single node, the access point and the default router. You would create a point to point like setup for every wireless node and the negative effects of scaling the wireless network could be partially circumvented. It is important for this to assume most traffic is between the wired network, including the routers, and the wireless node and not between wireless nodes near each other as this setup would then be highly inefficient.

Implementing this on the access point side proved also to be quite difficult as their firmware has become very large and complex. My modified firmware often prevented wireless connections at all or even crashed the entire access point. When connections were offered the internal administration of the connected nodes and the routing between everything was problematic and stable IP connections were never achieved. Another problem was found in the client network stacks, who started to behave strange with tiny subnets; dropping IP connections and disconnecting from the wireless network. Diagnosing the source of the problem, access point of client network stack, was difficult and sometimes even confusing.

CHAPTER 9

MOBILE IPV6

9.1 Research

A wireless connection gives the user the ability to physically move around with a networked system. Transitioning between WiFi access point will traditionally cause disconnects on established connections. Depending on the setup the system can even receive a different IP address. Can the mandatory IPv6 part MobileIPv6 offer an advantage in such a scenario? Will this functionality give IPv6 an advantage for the end user over IPv4 on a WiFi network? Above the advantage for the user can it support roaming between multiple access points when dividing the network into multiple divided subnets as proposed in section 7.2.6? This subject will be discussed as I find the most promising feature and research question four as stated in section 3.3.

9.2 Description

The past few years notebooks, tablets and smart phones are gaining a lot of ground in the world internet connected devices The same can be said for wireless connections in contrast to their more traditional wired counterparts. Devices connected to the internet are getting more portable. We expect our device to remain connected when we move around and even change our point of attachment to the network. The same behavior we are accustomed to when moving around with a mobile phone, roaming from one cell to another without noticed by the user.

For example, in the morning at home your smart phones is connected to your home WiFi network, then you leave your house and on the street your device switches to a UMTS data connection and when arrived at the office switches to another WiFi network. You would not want a changing phone number on your trip to the office. Suppose your IP address would not change either and all the applications running on your devices stay connected as they were. This is where Mobile IP comes into play. Without Mobile IP most applications will keep working as they only fetch data and basically constantly use different connections. Especially continuous or corporate applications may benefit from a static address with respect to manageability and security and of course connections to your portable device.

As a solution to overcome the problem of IP address changes during roaming between access points, I will take a look at Mobile IPv6. In a practical point of view, changing from access points can be seen as completely changing networks.

Mobile IP is not new in IPv6[13]. It has previously been developed as an extension for IPv4[12]. However Mobile IPv4 is not often deployed and VPN solutions are commonly preferred. Mobile IPv6 offers some advantages and maybe in combination with the deprecated need for Network Address Translation in networks it can offer a suitable and preferable solution.

9.3 How it works

In order to describe Mobile IPv6 we need some definitions to work with.

Mobile Node

A network node that uses Mobile IPv6 which can be connected to either its home or foreign link.

Home Address

A global unicast address assigned to a Mobile Node which is used as the permanent address for the node.

Care-of Address
The address of a Mobile Node while on a Foreign Link.

Home Link
The link on which the Home Address is defined.

Foreign Link
Another link to which the Mobile Node can be connected.

Home Agent
A router on the Home Link which takes care of the Home Address if the Mobile Node is away.

Binding
The association of the Home Address to a Care-of Address for a Mobile Node

Correspondent Node
A external node communicating with the Mobile Node.

As long as the Mobile Node is connected to the Home Link it receives packets through regular IP routing mechanisms and behaves like any other regular host.[17] Things start change when the Mobile Node is away from home and connected to a Foreign Link. It now has received a Care-of Address on the Foreign Link using regular IPv6 mechanisms such as Stateless autoconfiguration. There is no configuration necessary on the Foreign Link for Mobile IPv6. Noted that a firewall on the Foreign Link can disturb matters. While away from its Home Link the Mobile Node registers its care-of address with its Home Agent by sending a binding update to the Home Agent which will respond with a binding acknowledgement.

There are two operating methods:

Bidirectional Tunneling
Packets from the Correspondent Node are sent to the Home Agent, which encapsulates them and sends them to the Care-of Address of the Mobile Node. Packets from the Mobile Node are sent back to the Home Agent, which forwards them to the Correspondent Node through regular routing. The Mobile Node sends Bindings Updates to the Home Agent to maintain its relation. During operation the Home Agent intercepts the traffic on the Home Link in order to redirect them to the Care-of Address of the Mobile Node. A flow scheme with basic header information is illustrated in figure 9.2. This mode does not require any Mobile IPv6 support on the Correspondent Node.

Route Optimization
Communication between a Mobile Node and Correspondent Node is direct without routing the traffic through the Home Agent. This is a large advantage of Mobile IPv6 over its predecessor. Route Optimization requires Mobile IPv6 support at the Correspondent Node. It requires an additional setup step called Correspondent Registration, in which the Mobile Node registers its Care-of Address to the Correspondent Node. The IP stack on the Correspondent Node will know about the mobility of the Mobile Node and will manage the combination of the Home Address and Care-of Address. A flow scheme with basic header information is illustrated in figure 9.3.

When the Correspondent Node also supports Mobile IPv6 Route Optimization should be used for faster connection using the shortest path, otherwise Bidirectional Tunneling offers a solutions. Figure 9.1 illustrates both operating methods. IPSec can be used on the links, especially between Mobile Node and Home Agent, to secure data transfer for the Mobile Node on a, possible untrusted, Foreign Link.

In contrast to MobileIPv4 neither modes requires a Foreign Agent forming an advantage for the new version. Other advantages are, Home Agent discovery and extended mobility options in the headers.[17] Please refer to documentation for more information.

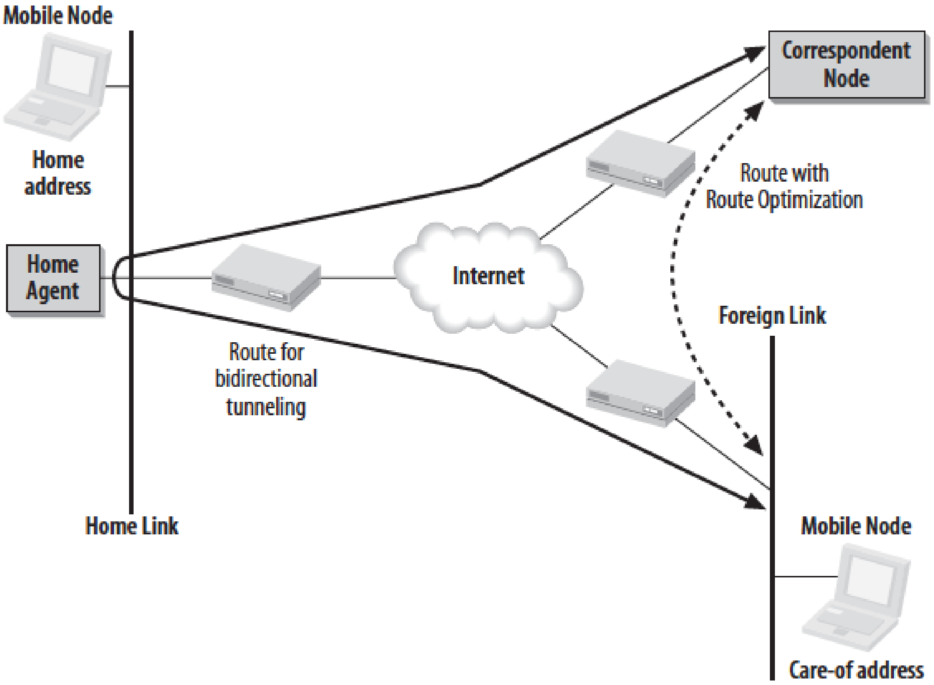
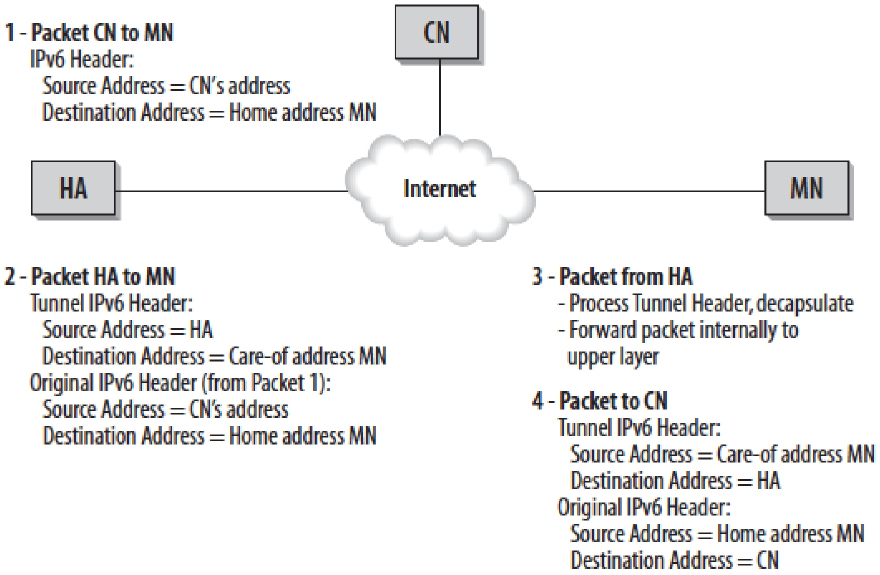


Figure 9.1: Routing modes[17]



1 - Packet CN to MN
IPv6 Header:
Source Address = CN's address
Destination Address = Home address MN

2 - Packet HA to MN
Tunnel IPv6 Header:
Source Address = HA
Destination Address = Care-of address MN
Original IPv6 Header (from Packet 1):
Source Address = CN's address
Destination Address = Home address MN

3 - Packet from HA
- Process Tunnel Header, decapsulate
- Forward packet internally to upper layer

4 - Packet to CN
Tunnel IPv6 Header:
Source Address = Care-of address MN
Destination Address = HA
Original IPv6 Header:
Source Address = Home address MN
Destination Address = CN

Figure 9.2: Bidirectional Tunneling[17]

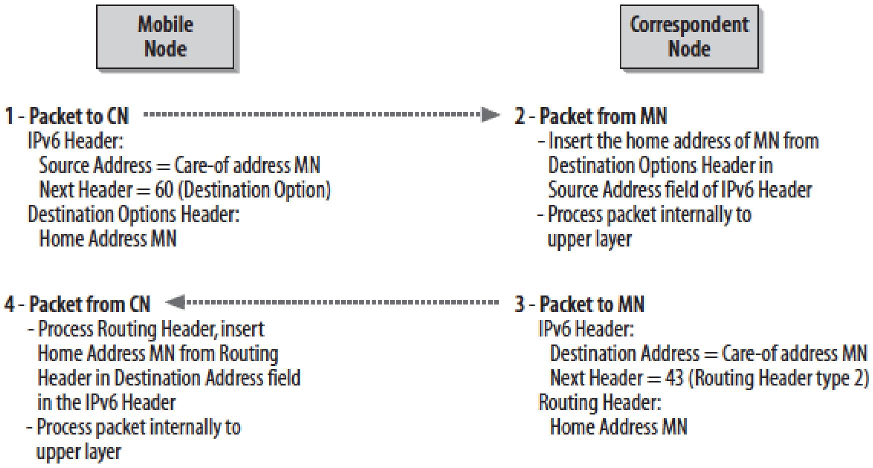

Mobile Node
Correspondent Node
1 - Packet to CN
IPv6 Header:
Source Address = Care-of address MN
Next Header = 60 (Destination Option)
Destination Options Header:
Home Address MN
2 - Packet from MN
- Insert the home address of MN from Destination Options Header in Source Address field of IPv6 Header
- Process packet internally to upper layer
3 - Packet to MN
IPv6 Header:
Destination Address = Care-of address MN
Next Header = 43 (Routing Header type 2)
Routing Header:
Home Address MN
4 - Packet from CN
- Process Routing Header, insert Home Address MN from Routing Header in Destination Address field in the IPv6 Header
- Process packet internally to upper layer

Figure 9.3: Route Optimization[17]

9.4 Test setup

In order to test Mobile IPv6 and evaluate its performance and usability a test setup is required.

Mobile IPv6 is not by far a mainstream enough feature to be included in standard operating system network utilities. Therefore manual configuration is needed. For Linux environments there are a few projects that provide the user space configuration utilities to support Mobile IPv6. I used UMIP, see appendix B, to configure my test nodes. I choose not to use IPSec in my tests as it is not required to test Mobile IP core functionality, gets thing more complex especially when using Route Optimization and makes debugging (eg. packet inspection) a lot more difficult.

The following list describes the used entities.

Mobile Node
I used a virtual machine running Debian. The virtual machine runs on my laptop to make it portable.

Home Link
I created a /64 subnet on which a Home Agent could act. I also put a separate wireless access point on it so the Home Link network was accessible both wired and wireless.

Home Agent
Another virtual machine was deployed to act as the Home Agent. It has a wired connection to the Home Link.

Nearby Link
My normal home network (/64 subnet) acted as a Foreign Link, which I called Nearby Link. It is within the same /48 subnet as the Home Link but has a router between it and a different address space. The term Nearby applies to both physical, I can switch links without physically moving the Mobile Node, as network technical, a short and fast path between them, properties. A nearby link can be seen as an adjacent access point with its own subnet.

Foreign Link
As Campusnet[1] provides a fine IPv6 connection I used it as a Foreign Link. Physical travel time between the Home Link and this Foreign Link is about 20 minutes.

Correspondent Node I
My desktop machine at home acted as a Correspondent Node. It is connected to the Nearby Link. I added the letter 'I' to indicate this node is intelligent as it is Mobile IPv6 aware.

[1] The student connectable part of the University of Twente network

Correspondent Node D

My secondary laptop acted as a Correspondent Node. It is connected to the Nearby Link. In contrast to the 'I' Node this one is Dump, without any Mobile IPv6 configuration.

External Node

An External Node which is completely independent of others entities and is in a separate network. It also acts as an VoIP server.

See appendix A for more details on used hardware.

I installed and configured UMIP on the Home Agent, Mobile Node and Correspondent Node I. On the main router I configured the required subnets and routes. In order to give an impression of the needed configuration I will briefly show the configuration on the involved nodes.
The main Home Agent UMIP configuration is as follows:

```
# act as Home Agent
NodeConfig HA;
# give me lots of debug information
DebugLevel 10;
Interface "eth0";
# allow MN to bind
BindingAclPolicy 2001:980:ad8a:42::50 allow;
# deny all others
DefaultBindingAclPolicy deny;
# do use IPSec between MN and HA
UseMnHaIPsec disabled;
KeyMngMobCapability disabled;
```

To perform Router Advertisements I used the radvd daemon. It allows a few extra configuration lines to advertise Mobile IPv6 support on the advertised subnet.

```
AdvHomeAgentFlag on;
AdvHomeAgentInfo on;
HomeAgentLifetime 3600;
HomeAgentPreference 10;
```

The Mobile Node uses the following UMIP configuration:

```
# act as Mobile Node
NodeConfig MN;
# give me lots of debug information
DebugLevel 10;
# wait for binding ack before using tunnel
OptimisticHandoff disabled;
# let a binding be valid for 60s
MnMaxHaBindingLife 60;
# do not use IPsec
UseMnHaIPsec disabled;
KeyMngMobCapability disabled;
# define used interface
Interface "wlan0" {
    MnIfPreference 10;
}
# define Home Link and HA address
MnHomeLink "wlan0" {
    HomeAgentAddress 2001:980:ad8a:42::100;
    HomeAddress 2001:980:ad8a:42:/64;
}
# enable when testing Route Optimization
#DoRouteOptimizationMN enabled;
# disable when testing Bidirectional Tunneling
DoRouteOptimizationMN disabled;
```

And finally the UMIP configuration for the Correspondent Node I. Correspondent Node D does not need any configuration.

```
# act as a Correspondent Node
NodeConfig CN;
```

```
# give me lots of debug information
DebugLevel 10;
# do not use IPsec
KeyMngMobCapability disabled;
# allow Route Optimization
DoRouteOptimizationCN enabled;
# Simply use above statement and allow all mobile nodes to bind
CnBindingPolicySet {}
# let a binding be valid for 60s
MnMaxCnBindingLife 60
```

My test setup knows three situations:

Home The Mobile Node is in the Home Link, see Figure 9.4

Nearby The Mobile Node is in the Nearby Link, see Figure 9.5

Foreign The Mobile Node is in the Foreign Link, see Figure 9.6

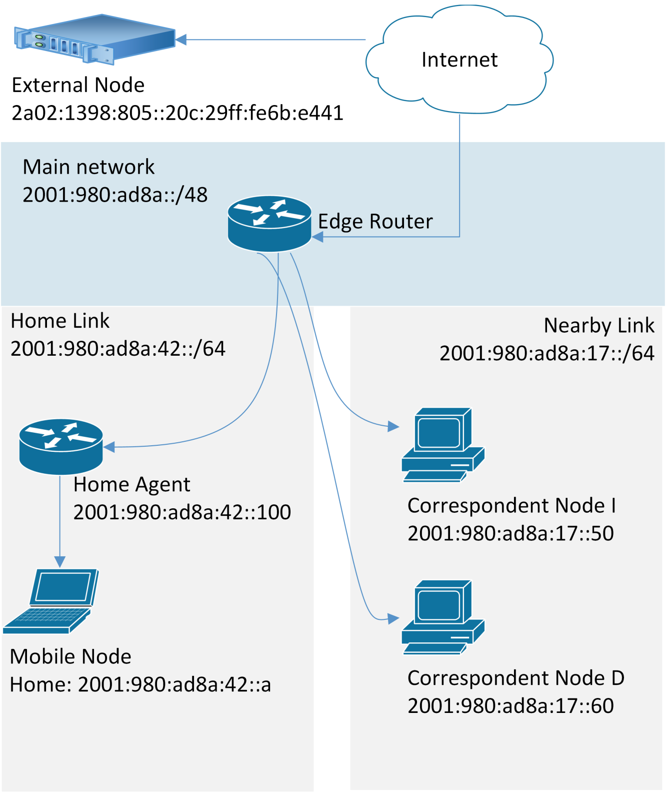


Figure 9.4: Mobile Node on its Home Link

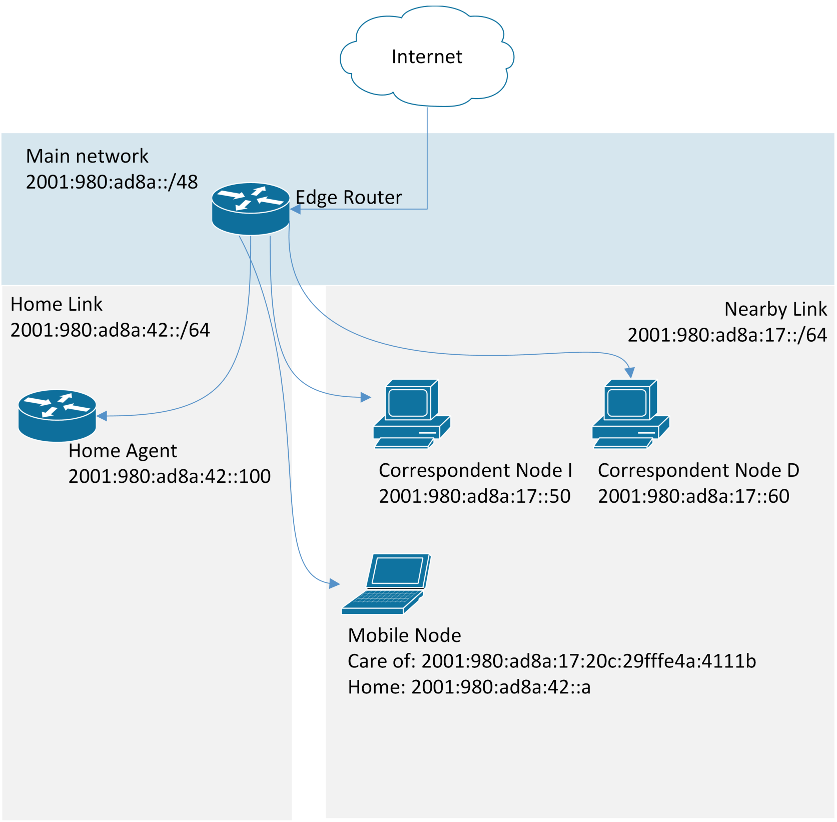


Figure 9.5: Mobile Node on the Nearby Link

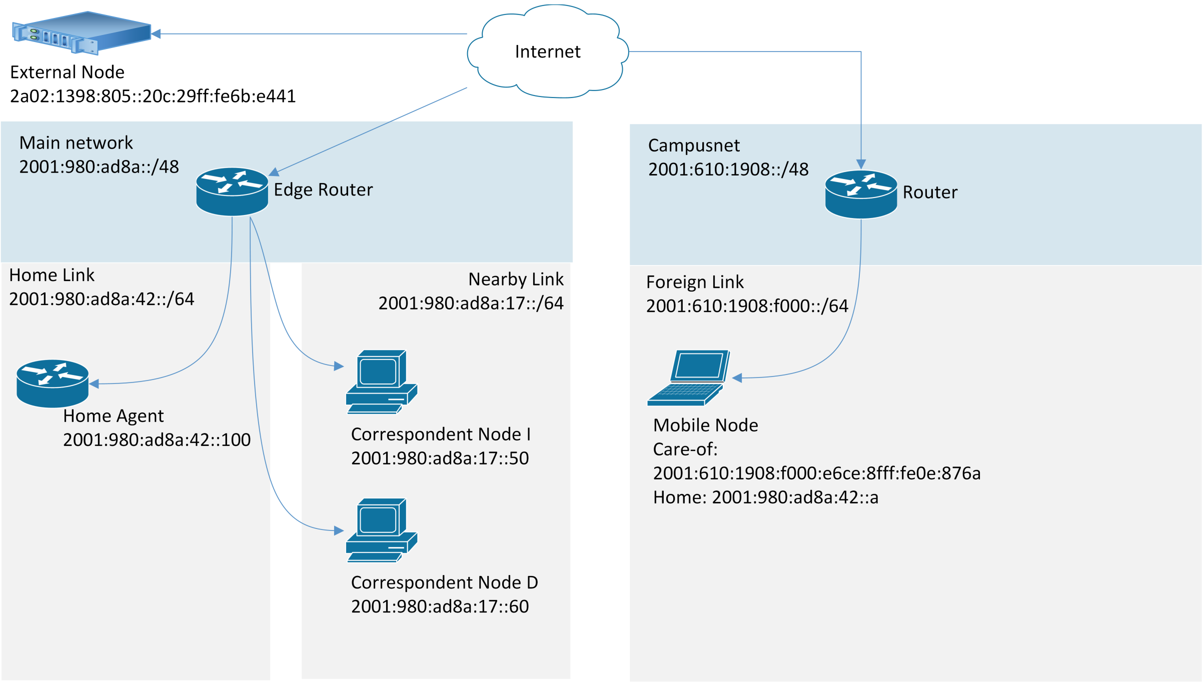


Figure 9.6: Mobile Node on Campusnet as a Foreign Link

9.5 Results

In order to test my setup and the basic performance of Mobile IPv6 I will perform the following tests in the three situations I described in the previous section: Home (figure 9.4), Nearby (figure 9.5) and Foreign (figure 9.6). Unfortunately I do not have possession of a IPv6 capable cellular data connection to test handovers outside on the move.

Ping
A simple continuous running ping from both the Correspondent Nodes to the Mobile Node in both directions.

SSH
A normal SSH connection is made from the Correspondent Nodes to the Mobile Node. It is continuously outputting some text. Will the session survive situation transitions?

Streaming
A video stream will be displayed on a Correspondent Node hosted on the Mobile Node via HTTP. Standard video player (VLC) buffering will be active as would be on normal usage.

VoIP
Calling using a VoIP client on the Mobile Node connected to the VoIP server on the External Node. During this test both parties will count aloud so you can measure the amount of time you are not connected.

Transfer
A measurement of available bandwidth using iperf. Not performed on transitional situations.

In the following subsection I will briefly discuss the results. Discussing all the results will greatly expand the size of this report and will not provide more value. All the results are summarized in section 9.5.6. Moving from Home/Nearby (at home) to Foreign takes me 20 minutes to travel. I subtracted the time between disconnecting the Mobile Node at home and reconnecting on the University campus.

9.5.1 Ping

The ping test did not show surprises. In a static situation the performance was always great. This demonstrates my test setup worked in every situation.

Example output when moving from Home to Nearby. Changing WiFi network took about two seconds to complete. The test missed a total of four replies and thus the Mobile IP handover took about two seconds.

```
16 bytes from 2001:980:ad8a:17:20c:29ff:fe4a:411b, icmp_seq=40 hlim=63 time=0.250 ms
16 bytes from 2001:980:ad8a:17:20c:29ff:fe4a:411b, icmp_seq=41 hlim=63 time=0.297 ms
16 bytes from 2001:980:ad8a:17:20c:29ff:fe4a:411b, icmp_seq=42 hlim=63 time=0.322 ms
16 bytes from 2001:980:ad8a:17:20c:29ff:fe4a:411b, icmp_seq=43 hlim=63 time=0.346 ms
16 bytes from 2001:980:ad8a:17:20c:29ff:fe4a:411b, icmp_seq=48 hlim=63 time=0.362 ms
16 bytes from 2001:980:ad8a:17:20c:29ff:fe4a:411b, icmp_seq=49 hlim=63 time=0.332 ms
16 bytes from 2001:980:ad8a:17:20c:29ff:fe4a:411b, icmp_seq=50 hlim=63 time=0.284 ms
```

The handover to/from Foreign took one second longer. Resulting ping times were equal to ping times using the Care-of Address.

9.5.2 SSH

After transitions between Home and Nearby the session survived with a small hitch but without losing output. Transitioning between Foreign took too long and disconnected the session; travel time is to blame here.

9.5.3 Streaming

While transitioning with Foreign I paused the video player during travel time in order to exclude it.
The video kept playing without a hitch in all tests. Buffering in the video player prevented any discomfort.

9.5.4 VoIP

A phone call is known to be very sensitive to connection interruptions. I regularly use my smart phone for VoIP over my mobile data connection. When my phone switches between different network types (4G/3G/2G/WiFi) the call is over.
Calling during a transition between Home and Nearby actually kept the call alive! There was a silence of about eight seconds, but it did not disconnect. Transitioning between Foreign took too long and disconnected the call, again travel time is to blame.

9.5.5 Transfer

To perform this test I used iperf to measure available bandwidth between the Correspondent Nodes and Mobile Node in both directions. For the Mobile Node I tested using both the Home Address and Care-of Address, which bypasses Mobile IPv6.

As can be seen in table 9.1 there were no significant differences between a regular connection and both modes of Mobile IPv6. Only in the last test Route Optimization showed a slight performance improvement to Bidirectional Tunneling. This can be explained by the slightly lower upload capacity of the Home Link compared to the download capacity of the External Node in combination with the over capacity of the Foreign Link. Like noted before, a network topology with longer paths including lower bandwidth sections will probably give the advantage to Routing Optimization.

Table 9.1 Transfer test results in Mbps

Situation	Direct	Bidirectional Tunneling	Route Optimization
Home	162.4	-	-
Nearby	161.8	161.5	161.6
Foreign	92.6	93.1	92.8
Foreign ⇒ External	98.6	92.7	96.5

9.5.6 Summary

To give an overview of the test results I summarized the results in table 9.2. The different test cannot be expressed in a single concrete unit of measurement. As the goal of this chapter is not to measure actual performance on this network in in particular, instead the usability and added value of this functionality in IPv6 is of importance here. Plus and minus signs are used to indicate how well different parts went both looking at actual performance as user experience.

Overall I was very satisfied by the results and if client software usability improve Mobile IPv6 can provide a better user experience on IPv6 enabled wireless networks when moving around a building or even campus.
I hardly noticed performance differences between Bidirectional Tunneling and Route Optimization. When network paths between the different nodes are longer or have lower bandwidth sections this will probably provide an advantage in performance. This was not the case in my setup explaining the lack of performance differences.

Table 9.2 The results summarized

Situation	Ping	SSH	Streaming	VoIP	Transfer
Home	+++	+++	+++	+++	+++
Nearby	+++	+++	+++	+++	+++
Foreign	+++	+++	+++	+++	+++
Home ⇒ Nearby	++	++	+++	+	+++
Home ⇒ Foreign*	++	- **	+++	- **	+++
Nearby ⇒ Home	++	++	+++	+	+++
Nearby ⇒ Foreign*	++	- **	+++	- **	+++
Foreign ⇒ Home*	++	- **	+++	- **	+++
Foreign ⇒ Nearby*	++	- **	+++	- **	+++

+++ Excellent; not noticeable to user
++ Fine; barely noticeable to user
+ Did work, however not very satisfying
- Not usable in practice
* I excluded the travel time in the results
** Did not succeed due to travel time

CHAPTER 10

VULNERABILITIES

The main subject is not to evaluate the (security) vulnerabilities of IPv6 in combination with WiFi and therefore I can not dive too deep into this matter. However it is an important present-day aspect of computer networking and that is why this chapter is included in this report. Many of the mentioned vulnerabilities of IPv6 will not be solitary applicable to wireless networks; yet can be more effective or easier to deploy in contrast to wired networks.

This chapter will not cover all the vulnerabilities in IPv6 and/or WiFi, neither is it a selection of only the most dangerous ones. The selection is purely a result of my literature research and my own experiments. My goal is to discover interesting vulnerabilities that IPv6 can introduce with focus on wireless networks and to point out that such a new protocol can introduce risks, even when native connectivity is not offered on the network.

Firewall One of most obvious security risks after deploying IPv6 is the lack of firewalling. Firewalls are often defined at the edge router of the network. A IPv4 network using Network Address Translation to provide internet access to the internal hosts will often have only one or a few public internet, world wide available, IP addresses and a firewall can be defined upon the edge of them. Deploying IPv6 will introduce public/global IP addresses to the end nodes. Firewalling IPv6 nodes is no rocket science but easy to forget or oversee as it was not really needed in a IPv4 only network.[30]

10.1 Attacks

The effect of vulnerabilities are often best tested by attempting an attack (mis)using them. This will expose the risk and impact of the vulnerability. In the following subsections I will discuss and test different ones.

10.1.1 Rogue gateway / tunnel server

Send out router advertisements and/or offer DHCPv6 to let clients on the network use your machine as their default IPv6 router. Traffic flows will be redirected through your node, see figure 10.1. You will need IPv6 connectivity on your node which can be obtained by using tunnel techniques. It will enable you to perform several attacks on their traffic. Passive attacks include sniffing their traffic and active attacks include modifying their traffic. Of course this is especially applicable to non-encrypted traffic and is harder, however not impossible, on for example HTTPS traffic. You can even run a modified DNS server to promote AAAA records (do not return A records) and thereby route more traffic over your malicious router.

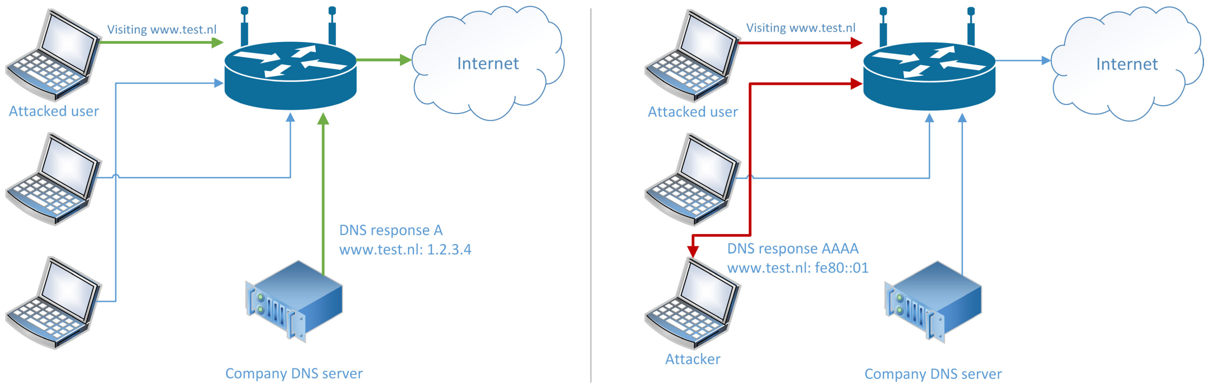


Figure 10.1: Abuse autoconfiguration. Left: normal, right: compromised.

To capture traffic from other users in the network I created a node with two network interfaces, one in the network to attack without IPv6 connectivity and the other in a network which offers native IPv6 connectivity. The node advertised it self as a IPv6 router on the first network using router advertisements. After a few seconds I started to see IPv6 traffic of unsuspecting nodes flow through my rogue gateway node. Using wireshark I was able to inspect plain HTTP traffic. Google, the most popular search engine in the Netherlands, uses HTTPS as default, otherwise you would be able to see exactly which topics people are working on. As it did not offered more value to this experiment I refrained from modifying traffic, however this is certainly an option using this principle.

The same principle can be applied when offering a rogue tunnel server instead of a “native” gateway.

Interesting to keep in mind that this attack can even be more effective when the network does not offer IPv6 support as you can fill in a gap as many clients will already support IPv6 and will happily accept you malicious offer.

10.1.2 Cross SSID injection

In section 2.3.3 it is stated that a wireless access point uses a shared broadcast key for all its clients. On one hand this is very efficient but on the other it made me wondering if this could be a vulnerabilities. Many access points will only offer a single SSID and others offer multiple over the same radio interfaces. For example open guest networks are increasingly offered aside the regular, secured, wireless network. As everyone can connect to the open guest SSID you will be able to obtain the shared broadcast key and misuse it for the other available SSIDs from the same access point. Using the obtained broadcast key you can eavesdrop the broadcast traffic and also transmit broadcast packets into the protected network, see figure 10.2.

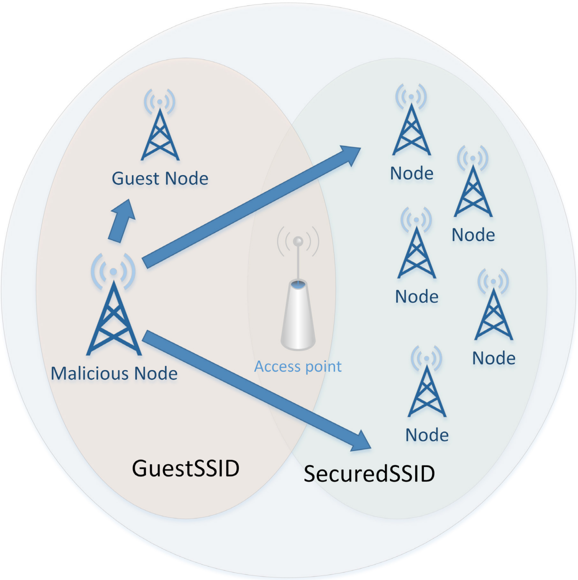


Figure 10.2: Cross SSID broadcast injection

I performed the following steps to successfully test this idea using only Wireshark and airpwm, see appendix B.

1. Connect to guest SSID
2. Send a broadcast message
3. Record shared broadcast key
4. Disconnect guest SSID and switch to monitor mode
5. Transmit packet using recorded broadcast key tagged with SecuredSSID ID

Using this technique you are able to inject broadcast messages and for example inject rogue router advertisements. Eavesdropping regular router advertisements can give you some information to tailor your rogue messages. Multiple of the vulnerabilities mentioned in this chapter can be applied in combination with this one. The broadcast key has only a short lifetime which hardens the operation. Using two wireless interfaces simultaneously could possibly provide a stream of up to date keys, but I haven't tested this as my point was already proven.

10.1.3 THC6 toolkit

During my research I found all kinds of vulnerabilities. In many cases getting to understand the general principle was not too hard. However testing them would take a fair amount of time as an implementation requires intervening in different levels and layers of the network stack. Someday I stumbled upon the THC6 toolkit and it made my day. The THC6 toolkit, see Appendix B, offers on one hand a brilliant toolbox and on the other hand a bit frightening set of simple to use tools to exploit IPv6 vulnerabilities. For this report it offers an interesting set of vulnerabilities related to IPv6 and many of them very effective on wireless networks.

Features Some of the included tools:[19]

parasite6 icmp neighbor solitication/advertisement spoofer, puts you as man-in-the-middle, same as ARP mitm (and parasite)

alive6 an effective alive scanning, which will detect all systems listening to this address

dnsdict6 parallized dns ipv6 dictionary bruteforcer

fake_router6 announce yourself as a router on the network, with the highest priority

redir6 redirect traffic to you intelligently (man-in-the-middle) with a clever icmp6 redirect spoofer

toobig6 mtu decreaser with the same intelligence as redir6

detect-new-ip6 detect new ip6 devices which join the network, you can run a script to automatically scan these systems etc.

dos-new-ip6 detect new ip6 devices and tell them that their chosen IP collides on the network (DOS).

trace6 very fast traceroute6 with supports ICMP6 echo request and TCP-SYN

flood_router6 flood a target with random router advertisements

flood_advertise6 flood a target with random neighbor advertisements

exploit6 known ipv6 vulnerabilities to test against a target

denial6 a collection of denial-of-service tests againsts a target

fuzz_ip6 fuzzer for ipv6

implementation6 performs various implementation checks on ipv6

implementation6d listen daemon for implementation6 to check behind a fw

fake_mld6 announce yourself in a multicast group of your choice on the net

fake_mld26 same but for MLDv2

fake_mldrouter6 fake MLD router messages

fake_mipv6 steal a mobile IP to yours if IPSEC is not needed for authentication

fake_advertiser6 announce yourself on the network

smurf6 local smurfer

rsmurf6 remote smurfer, known to work only against linux at the moment

sendpees6 a tool by willdamn(ad)gmail.com, which generates a neighbor solicitation requests with a lot of CGAs (crypto stuff ;-) to keep the CPU busy. nice.

thcping6 sends a hand crafted ping6 packet

In the next subsections I will briefly discuss some of them. I tested almost all of them and the this piece of open source software keeps it's promises.

10.1.4 DAD Attack

Stateless autoconfiguration generates addresses in the advertised prefix. Before the generated addresses will be used it will be checked for uniqueness in the network using the Duplicate Address Detection feature of ICMPv6.
This process can be tormented by a malicious node performing a DAD attack, which is described by the following simple steps.

1. Autoconfigurating node: Can I use address AA:BB?
2. Malicious node: No, already in use.
3. Autoconfigurating node: Can I use address CC:DD?

4. Malicious node: No, already in use.

Simply reply to all DAD messages claiming unfairly to already have the newly generated address in use. The THC6 toolkit offers a implementation for this and makes the attack very simple. Just performing the command “dos-new-ipv6 wlan0” is enough!
As all generated autoconfiguration address are denied the newly connected client cannot claim an address and thus will not receive IPv6 connectivity.

I tested the construct using the command mentioned above and it works just like it promises.

Windows Tries five times and gives up until reconnecting to the network.

OS X Tries ten times and gives up, after ten minutes it starts to try again.

Linux depends on distro. Some ignore DADs and just uses the generated address, others give up after ten tries.

10.1.5 RA Flood

Router advertisements flooding is a form of a Denial of Service attack. Flooding hosts with (rogue) router advertisements can make nodes on the network “crazy”. Dedicating a system to transmit as many router advertisements as possible not only saturates a network link. Transmitting a static router advertisements packet is much less work for the system then to process a received one.

Again the THC6 toolkit offers a utility to test this. “flood_router6 wlan0”

Windows Entire system will freeze! I expect the reason in a completely overloaded and/or confused network stack; also affecting the loopback device.

Linux & OS X Increased CPU load and crippled IPv6 connectivity

Android Battery gets eaten alive, crippled IPv6 connectivity and WiFi resets

10.2 Countermeasures

The world is not a perfect place and no matter in which field there will be offenders; nothing new in the networking or IT security world. Having said that, let me discuss some countermeasures against these threats.

10.2.1 New in IPv6?

Of course the question rises; are these security problems new in IPv6? The answer is definitely “No, they are not”. Most of the mentioned problems have a known IPv4 counterpart. However they are longer known and many solutions have been implemented against them. The vulnerabilities in IPv6 are relatively new and for most of them protective measures still needs to be developed.[37] Furthermore some more advanced features of IPv6 also allow more advanced threats.

First hop security Protecting against threats is often most powerful when done as close as possible from the source. First hop security plays an important role against many attack types. Intelligent network equipment can offer first hop security on their access/edge ports. For example DHCP snooping will prevent rogue DHCP servers on unauthorized edge ports. Multiple guard methods for edge ports exists to protect against different type of threats; often given a vendor specific name.

10.2.2 SeND

Secure Neighbor Discovery is an extension on the standard Neighbor Discovery protocol that enhances it with three additional capabilities.[23] It is specified in IEEE RFC 3971.

Address ownership proof
Based upon Cryptographically Generated Addresses it protects against stealing IPv6 addresses.

Message protection
Offers message integrity protection, protecting against replay and man in the middle attacks.

Router authorization
Authorizes routers to act as default gateways or a specific prefix.

SeND will solve many threats surrounding the normal neighbor discovery protocol and assist in creating a more secure IPv6 network. A major disadvantage is found in the support requirement for SeND in the entire network to allow operation Native support is present-day still missing in all major operating systems.

10.2.3 RA-Guard

A guarding mechanism against rogue router advertisements and rogue DHCPv6 servers is comparable with the protection offered by DHCP snooping against rogue DHCPv4 servers. Network equipment should block outgoing router advertisements and DHCPv6 offers from unauthorized edge ports. Current implementations can be easily avoided using fragmentation and extension headers.[37] Incorrect configurations will be probably all be blocked using RA-Guards. Malicious users will still be able to abuse router advertisement packets. I believe without a doubt that network vendors will improve their implementations in the, hopefully near, future.

10.2.4 SAVI

The IP protocol, both IPv4 as IPv6, does not verify the source address of packets making it vulnerable to forge source addresses. A vulnerability abused in for example SYN flooding attacks. The Source Address Validation Improvement protocol aims to deal with this problem. SAVI is a combination of multiple coexisting and cooperating mechanisms to validate source addresses in a network to prevent address spoofing in the same network segment. Snooping control packet interactions in the network and creating dynamic bindings between MAC address, IP address and switch port is the basic principle of SAVI. It is most effectively deployed on network switches. Filtering based upon the recorded bindings prevent rapidly spoofing different addresses and stealing addresses of other nodes. It can not prevent address spoofing entirely but it does quite effectively prevent address stealing.[44]

CHAPTER 11

CONCLUSION

During my research I performed various experiments and saw different results. I had a lot of fun but also many struggles as I often over demanded hard- and software and even my programming skills when working with access point firmware and Linux kernels. The restrictions of wireless networks were often shown as did the lack of maturity of some IPv6 implementations.

1: What is the difference in control traffic between IPv4 and IPv6 and how does it influence network performance? The two protocol versions differ in techniques for network control and management. Theoretically ICMPv6 offers some interesting new features. However in practice, years after the engineers designed it, they do not offer real advantages. In small (<25 nodes) the difference in performance is negligible. However with the amount of nodes in the network, which do not have to be associated to same access point, increasing the control traffic of IPv6 will become an issue. The combination of the multicast usage and the characteristics of WiFi lead to seriously degrading performance.

2: How does the use of multicast by IPv6 influence WiFi-based networks? The answer to this question lies in combination with the first one. IPv6 is designed with extensive multicast usage in mind. The usage of multicast itself does not directly imply a problem on WiFi. However in combination with the ratio of ICMPv6 control packets and their flooding over the network the initial small issue becomes expanded. The engineers could at the time not have foreseen the impact of this on wireless networking, which became widely spread.

3: Are there solutions to solve identified issues for deploying IPv6 on a WiFi-based network? Unfortunately there are no complete off-the-shelf solutions available for all the encountered issue. There are however some easy to implement solutions to improve the situation and some promising possible solutions came to light that might solve the issues. Correctly segmenting the network into IPv6 subnets is possible using standard equipment and software and provides a solution to a certain level. Especially, the still to be proven solutions, infrastructure based multicast filtering and advertising off-link prefixes promise great results.

4: Does IPv6 offer new features or possibilities especially interesting for wireless networks? The most promising new feature of IPv6 for wireless networks is Mobile IPv6. It can provide a solutions when a fixed IP address is required during roaming between access points. Especially when every access point has its own subnet then roaming will become more frequent and the need for a persistent global IP address will increase. However quite an effort is needed in order to get Mobile IPv6 active and working properly. This is even the case for end-user client systems (Mobile Nodes). Standard operating system configuration utilities should be adapted to support Mobile IPv6 before regular users will be able to benefit from this technology.

5: Does IPv6 introduce security risks and vulnerabilities, especially applicable on wireless networks? The importance of computer network security keeps growing and IPv6 introduces some new and some revised vulnerabilities which should not be taken lightly. Some IPv6 related vulnerabilities can also be exploited when the network itself does not offer IPv6 connectivity. As the network administrators did not yet implement IPv6 their network will probably also lack any protection against its vulnerabilities.

Closing Without a doubt the world wide deployment of IPv6 should continue as the internet keeps growing and the workarounds to keep IPv4 sufficient will not be adequate for long. Deploying IPv6 on WiFi is not always straight forward however promising solutions are worthwhile consider and might solve the encountered issues for your deployment. The perfect off-the-shelf solution is not yet available. This fact should stimulate further research on this matter. As with many new technologies people should not blindly adopt them without thinking it through.

CHAPTER 12

FUTURE WORK

There is always so much to do and explore and you can not do everything yourself. During my research I came along some points and ideas which are worth mentioning as someone could pursue them for further interesting work.

Solutions Chapter 7 contains a list of proposed solutions. I was able to handle some of them but not all. The chapter contains some interesting possible solutions on which future research might bring promising results.

802.11ac The latest standard, which was too new to include in this report, called IEEE802.11ac includes a different solution for broadcast key sharing. It may offer a solution for the security implications of a shared broadcast key. What can this latest standard offer besides extreme fast transfer rates and does can IPv6 deployments gain along side?

Cellular data networks The amount of internet connected (smart) phones has been rapidly rising in the past few years, demanding more and more IP addresses on those cellular data networks. At least in The Netherlands telecom providers often use NAT for their mobile data connections offered over 3G and 4G networks to circumvent the problem of the exhausting IPv4 address space. It would be very interesting to see IPv6 deployed here and see how it will affect mobile cellular data connections. Removing the NAT equipment and giving a global IP to the end devices can give a performance gain, especially for the high end smart phones of today. MobileIPv6 has the potential for demanding users to supply a fixed IP address all over the world, which would be available on both cellular and WiFi connections.

Countermeasures In chapter 10 many vulnerabilities are discussed and with it challenges are filed for countermeasures, monitoring and guarding techniques.

Net neutrality “Netneutraliteit” or net/internet neutrality has become an hot item in The Netherlands and other countries. The primary fuss has been around certain types of traffic, like VoIP and Skype, being blocked by telecom providers on their cellular data connections. It has led to new regulation and more open cellular data connections. More recently some enthusiastic have taken a different approach to it. They stated that the lack of IPv6 connectivity on a consumer internet connection, either mobile or fixed at home, is a violation of the net neutrality laws.[4] At the time of writing it have been a hot topic on IPv6-related mailing lists. I am very curious if this can contribute to the adoption rate of IPv6 at commercial internet providers.

Mobile IPv6 GUI For general deployment and usage, getting Mobile IPv6 to work is much too hard. Standard network configuration utilities should include support for Mobile IPv6 in order to make it attractive for the more average user.

Major vendor solutions The commercial major vendors are also starting to provide decent support for IPv6 in their wireless network product ranges. However their commercial nature prevent openness about their solutions for the problems. It can be very interesting to test and explore their solutions. This however takes more time, money and probably good work relationships with the vendor or its partners.

APPENDIX A

HARDWARE USED FOR EXPERIMENTS

In order to perform my experiments I of course needed some hardware. The following tables list the used equipment with relevant specifications:

- Table A.1 Different WiFi cards or dongles (USB) used as wireless network adapters
- Table A.2 Used wireless access points / routers (functionalities are combined in single devices)
- Table A.3 Hardware systems used to connect, generate traffic, analyze traffic and other tasks

Table A.1 WiFi cards

Amount	Manufacturer	Model	Chipset	Interface	Type	Supports 2,4/5GHz	Monitor mode
1	Apple	AirPort Extreme	Broadcom BCM4331	PCI-e (mini)	a/b/g/n	300 / 450 Mbps	yes*
3	Intel	Pro Wireless 3945	Intel 3945	PCI-e (mini)	a/b/g	54 / 54 Mbps	yes**
1	Intel	Centrino Advanced-N 6200	Intel 6200	PCI-e (mini)	a/b/g/n	300 / 300 Mbps	yes**
10	TP-Link	TL-WN821N	Atheros AR9001U-2NG	USB2.0	b/g/n	300 / - Mbps	yes
34	TP-Link	TL-WN722N	Atheros AR9002U	USB2.0	b/g/n	150 / - Mbps	poor
2	Cisco	WUSB600N	Ralink RT3572	USB2.0	a/b/g/n	300 / 300 Mbps	poor
1	TP-Link	TL-WN8200ND	Realtek RU8192CU	USB2.0	b/g/n	300 / - Mbps	no
2	Nameless	Wi Fi 150M	Realtek RU8188CUS	USB2.0	b/g/n	150 / - Mbps	no
# 54							

* The Broadcom chipset forming the Apple Airport Extreme provides monitoring and injection capabilities. Software, including driver, support is available in Mac OS X version 10.5 and onwards.

** The Intel iwl3945 Linux driver supports monitor mode and injection of raw frames into the air. The Pro controller versions even support multiple interfaces on a single device allowing simultaneous usage of monitor and managed mode.

Table A.2 Access Points / Wireless routers

Manufacturer	Model	Wireless chipset	Supports 2,4/5GHz	Software
TP-Link	TL-WDR4900*	Atheros AR9381-AL1A Atheros AR9580-AR1A	450 / 450 Mbps	OpenWRT Barrier Breaker
TP-Link	TL-WR703N	Atheros AR9331	150 / - Mbps	OpenWRT Barrier Breaker
Netgear	WNDR3800 Premium	Atheros 9220 Atheros 9223	300 / 300 Mbps	OpenWRT Barrier Breaker CeroWRT trunk

* Main router, connected to my internet service provider. Formed the basis for all experiments.
All above wireless chipsets support monitor mode.

Table A.3 Systems

System	CPU	RAM	OS	Software	Purpose
Apple Macbook Pro 13”	Intel i7 2x 2,7GHz	16GB	Mac OS X 10.9	VMWare Fusion 5	Main workhorse*
MSI Megabook S262	Intel T7200 2x 2,0GHz	4GB	Ubuntu 14.04	VMware Workstation 10	Network (hypervisor)
Desktop	Intel i7 4x 3,8GHz	36GB	Ubuntu 14.04 ESXi 5.1		Second workhorse*
Home server	Intel G3220 2x 3,0GHz	8GB	Ubuntu 14.04	UMIP 1.0	Multicast devil and Home Agent
Raspberry Pi	ARM 700Mhz	496MB	Raspian Wheezy		Network node (with limited CPU power)
Compaq Presario CQ62	Intel T4500 2x 2,3GHz	3 GB	Windows 7	VMware Workstation 10	Network (hypervisor) node
LG Nexus 5	Snapdragon 800	2GB	Android 4.4.2		Network node (with limited antenna**)
Asus Nexus 7	Tegra 3	1GB	Android 4.4.2		Network node with limited antenna
Motorola Moto G	Snapdragon 400	1GB	Android 4.4.2		Network node with limited antenna
Test VM	VMWare vCPU	4GB	Windows 7		Windows 7 test node
Test VM	VMWare vCPU	4GB	Windows 8.1		Windows 8 test node
Test VMs (up to 250x)	VMWare vCPU	96MB	Debian 7		Scalable (stress test) nodes
HP 2510-24G					Network gigabit switch

* Workhorses formed my primary test systems. They functioned as a test platform, Virtual Machine hypervisor, traffic analyzer and many other things.

** Although the Nexus 5 is a small mobile device its WiFi performance is quite powerful.

Appendix B

Software used for experiments

For my experiments I used many different utilities and programs. Table B.1 summarizes the most important and notable ones.

Table B.1 Software

Name	Purpose	Website
OpenWRT	Powerful open source router platform	`openwrt.org`
CeroWRT	Fork of OpenWRT development of new advanced wireless features & improvements	`bufferbloat.net/projects/cerowrt`
Wireshark	Traffic analyzing	`www.wireshark.org`
airpwm	802.11 traffic injection	`airpwn.sourceforge.net`
aircrack-ng	Wireless cracking tools, including capture and decryption	`www.aircrack-ng.org`
tcpdump	Traffic dumping	`www.tcpdump.org`
iperf	Performance testing	`iperf.fr`
netperf	Performance testing	`www.netpef.org`
fprobe	Netflow probe	`sourceforge.net/projects/fprobe`
ntopng	Netflow probe/analyzer	`www.ntop.org/products/ntop`
nping	Network packet generation tool	`nmap.org/nping`
zmap	Fast port scanner	`zmap.io`
radvd	Sending valid and malicious router advertisements	`www.litech.org/radvd/`
thc-ipv6	IPv6 and ICMPv6 attack toolkit	`www.thc.org/thc-ipv6/`
mcsender	Sending multicast messages	`github.com/troglobit/smcroute`
UMIP	Mobile IPv6 user space utilities	`www.umip.org`
vim	My favourite all time editor ☺	`www.vim.org`

References

[1] Marc Blanchet. *Migrating to IPv6. A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks*. Wiley, first edition edition, December 2005.

[2] Rob Blokzijl. Ipv4 header vs ipv6 header, February 2009. URL http://www.ripe.net/ripe/meetings/roundtable/february-2009/RobBlokzijlroundtable2009Rob2.pdf.

[3] M. Christensen, K. Kimball, and F. Solensky. Considerations for internet group management protocol (igmp) and multicast listener discovery (mld) snooping switches. RFC 4541, RFC Editor, May 2006. URL https://tools.ietf.org/html/rfc4541.

[4] M. Davids. Beleidsregel netneutraliteit, May 2014. URL http://www.internetconsultatie.nl/netneutraliteit/reactie/77100d73-4571-471e-b9ef-2ad5d4b9608a.

[5] Paul de Kuyper. Campusnet klaar voor nieuw internetprotocol. *UT Nieuws*, December 2010. URL http://www.utnieuws.nl/sites/default/files/pdf/UT-Nieuws-10-12-02.pdf.

[6] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, RFC Editor, 1995. URL http://www.ietf.org/rfc/rfc1883.txt.

[7] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, RFC Editor, 1998. URL http://www.ietf.org/rfc/rfc2460.txt.

[8] S. Deering, W. Fenner, and B. Haberman. Multicast listener discovery (mld) for ipv6. RFC 2710, RFC Editor, October 1999. URL https://tools.ietf.org/html/rfc2710.

[9] S. Deering, R. Hinden, and E. Nordmark. IPv6 Global Unicast Address Format. RFC 3587, RFC Editor, 2003. URL http://www.ietf.org/rfc/rfc2587.txt.

[10] M. Diepenhuis. Draadloos internet op campus komt eraan. *UT Nieuws*, January 2003.

[11] Ed. E. Vyncke, P. Thubert, E. Levy-Abegnoli, and A. Yourtchenko. Why Network-Layer Multicast is Not Always Efficient At Datalink Layer. RFC, RFC Editor, August 2014. URL https://tools.ietf.org/html/draft-vyncke-6man-mcast-not-efficient-01.

[12] C. Perkins Ed. IP Mobility Support for IPv4. RFC 3344, RFC Editor, 2002. URL http://www.ietf.org/rfc/rfc3344.txt.

[13] C. Perkins Ed. Mobility Support in IPv6. RFC 6275, RFC Editor, 2002. URL http://www.ietf.org/rfc/rfc6275.txt.

[14] Matthew S. Gast. *802.11 Wireless Networks, The Definitive Guide*. O'Reilly, second edition edition, 2005.

[15] Google. Ipv6 statistics, May 2014. URL https://www.google.com/intl/en/ipv6/statistics.html.

[16] S. Gundavelli, M. Townsley, O. Troan, and W. Dec. Address mapping of ipv6 multicast packets on ethernet. RFC 6085, RFC Editor, January 2011. URL https://tools.ietf.org/html/rfc6085.

[17] Silvia Hagen. *IPv6 Essentials*. O'Reilly, second edition edition, 2006.

[18] Eric Hall. *Internet Core Protocols*. O'Reilly, first edition edition, 2000.

[19] Van Hauser. Thc-ipv6, June 2014. URL https://www.thc.org/thc-ipv6/.

[20] R. Hinden and S. Deering. Ip version 6 addressing architecture. RFC 4291, RFC Editor, February 2006. URL http://www.ietf.org/rfc/rfc4291.txt.

[21] Andrew Hines. Neighbour discovery in ipv6. *No: 6225220*, August 2004. URL http://www2.cs.uni-paderborn.de/cs/ag-madh/WWW/Teaching/2004SS/AlgInternet/Submissions/17-neighbour-discovery-protocol-in-IPv6.pdf.

[22] Geoff Huston. Ipv6 performance, November 2015. URL http://www.potaroo.net/ispcol/2015-11/v6perf.html.

[23] Cisco Systems Inc. Ipv6 first hop security - protecting your ipv6 access network. Technical report, Cisco Systems Inc., April 2010. URL http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/whitepaper_c11-602135.pdf.

[24] Cisco Systems Inc. Wireless lan ipv6 client deployment guide, February 2012. URL http://www.cisco.com/c/en/us/support/docs/wireless/5500-series-wireless-controllers/113427-cuwn-ipv6-guide-00.html.

[25] Cisco Systems Inc. *Enterprise Mobility 7.3 Design Guide*. Cisco Systems Inc, revised edition

edition, September 2013.

[26] Cisco Systems Inc. What is osi model & the overall explanation of its 7 layers, March 2013. URL http://www.cisco1900router.com/what-is-ios-model-the-overall-explanation-of-ios-7-layers.html.

[27] Slawomir Kuklinski, Pawel Radziszewski, and Jacek Wytrebowicz. Ipv6 in wireless networks - selected issues. *Journal of Telecommunications and Information Technology*, February 2011. URL http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-article-BATA-0013-0047/c/httpwww_itl_waw_plczasopismajtit2011224.pdf.

[28] James F. Kurose and Keith W. Ross. *Computer Networking*. Pearson Education International, third edition edition, 2005.

[29] B.J. Meijerink. Ipv6 neighbor discovery protocol and its effects on flow monitoring equipment. *16th Twente Student Conference on IT*, January 2012. URL http://referaat.cs.utwente.nl/conference/16/paper/7294/ipv6-neighbor-discovery-protocol-and-its-effects-on-flow-monitoring-equipment.pdf.

[30] Daniel Minoli and Jake Kouns. *Security in an IPv6 Environment*. CRC Press, Auerbach, first edition edition, 2009.

[31] Niall Richard Murphy and David Malone. *IPv6 Network Administration*. O'Reilly, first edition edition, 2005.

[32] T. Narten and R. Draves. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 3041, RFC Editor, 2001. URL http://www.ietf.org/rfc/rfc3041.txt.

[33] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, RFC Editor, 1998. URL http://www.ietf.org/rfc/rfc2461.txt.

[34] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor discovery for ip version 6 (ipv6). RFC 4861, RFC Editor, September 2007. URL https://tools.ietf.org/html/rfc4861.

[35] UT Nieuws. Ut als eerste nederlandse universiteit helemaal klaar voor ipv6. *UT Nieuws*, June 2012. URL http://www.utwente.nl/nieuwsevents/2012/6/260017/ut-als-eerste-nederlandse-universiteit-helemaal-klaar-voor-ipv6.

[36] Oracle. Ipv6 administration guide, 2010. URL http://docs.oracle.com/cd/E19683-01/817-0573/index.html.

[37] Tomas Podermanski. Security challenges in ipv6 from the campus perspective. *NORDUnet 2012 Conference*, September 2012. URL https://events.nordu.net/plugins/servlet/conference-attachment/talks/23/166.

[38] Karl A. Siil. *IPv6 Mandates: Choosing a Transition Strategy, Preparing Transition Plans, and Executing the Migration of a Network to IPv6*. Wiley, first edition edition, March 2008.

[39] Internet Society. World ipv6 launch, May 2014. URL http://www.worldipv6launch.org.

[40] Dave Tath. What's wrong with wifi?, February 2014. URL https://www.youtube.com/watch?v=Wksh2DPHCDI. MIT talk.

[41] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862, RFC Editor, 2007. URL http://www.ietf.org/rfc/rfc4862.txt.

[42] Jeff Wheeler. Layer-2 multicast state problems caused by ipv6 neighbor discovery (nd), June 2013. URL https://www.nanog.org/sites/default/files/tues.general.wheeler.neighbor.12.pdf.

[43] Ruckus Wireless. Deploying veryhigh density wi-fi, 2012. URL http://www.potaroo.net/ispcol/2015-11/v6perf.html.

[44] Zhihui Yan, Gengsheng Deng, and Junyun Wu. Savi-based ipv6 source address validation implementation of the access network. *Computer Science and Service System International Conference 2011*, June 2011. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5974125.

[45] A. Yourtchenko and L. Colitti. Reducing Multicast in IPv6 Neighbor Discovery. RFC, RFC Editor, February 2014. URL https://tools.ietf.org/html/draft-yourtchenko-colitti-nd-reduce-multicast-00.