Remote OS detection with IPv6

Mathias Morbitzer

m.morbitzer@runbox.com

About me

- 5 years studies in IT security in Austria and the Netherlands
- Worked on translating from IPv4 to IPv6 and back
- Currently working as penetration tester

Why is Remote OS Detection?

- Imagine a 0-day exploit....
  $\rightarrow$ Necessary to determine remote OS
- Creating exploits
- Social engineering
- Inventory
- Finding unauthorized devices

What is Remote OS Detection?

- A lot of different RFCs
- Not everything defined in RFCs
- Not every system follows RFCs

$\rightarrow$ Find differences in behavior

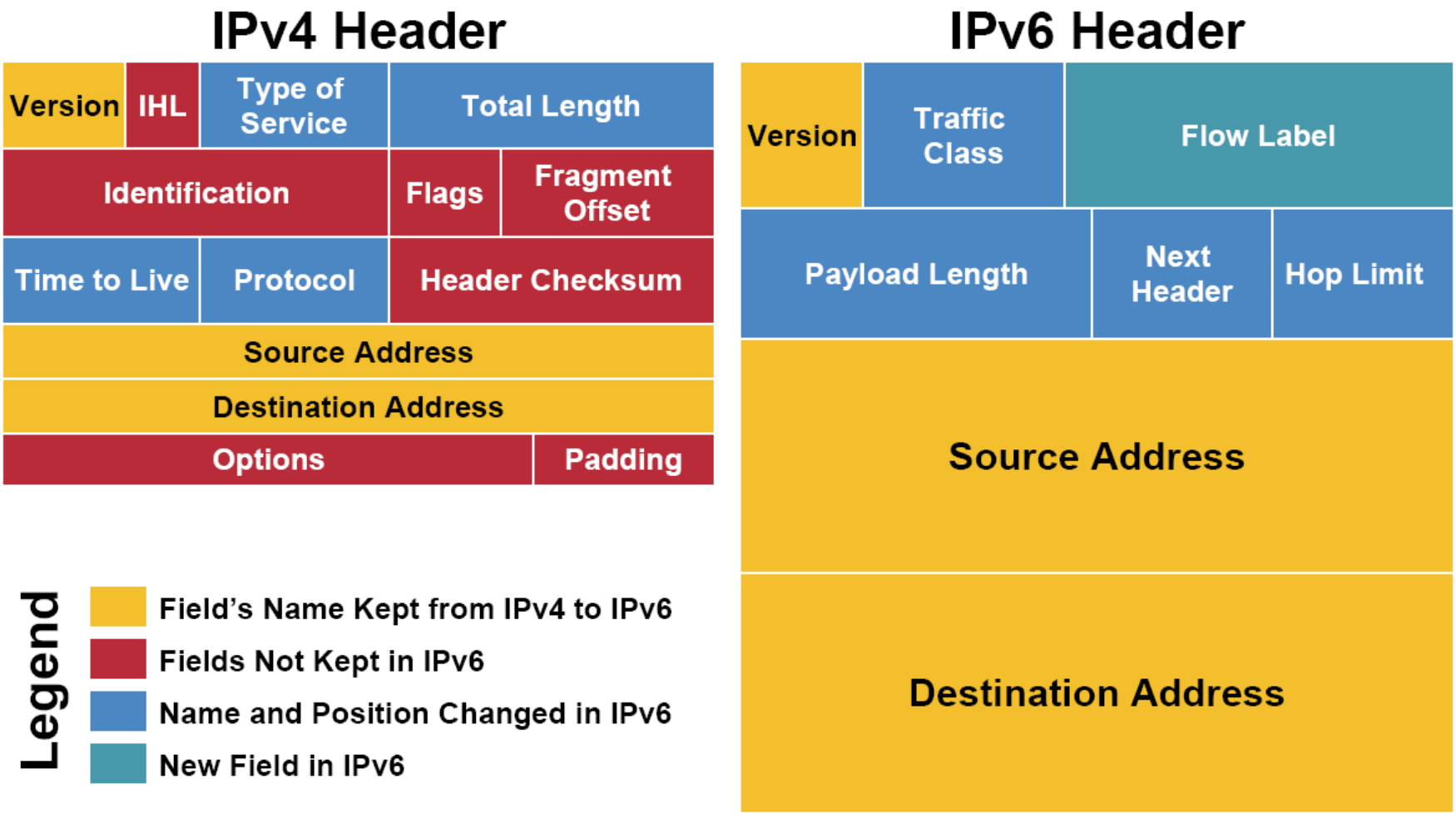
Some existing methods

- IPv4: IPID sequence
- TCP: GCD of ISN
- TCP: predictability of ISN
- TCP: Order of options
- TCP: Window Size
- ICMP: Echo Reply Codes
- ....

Some existing methods

- Tons of methods
  (More at http://nmap.org/book/osdetect-methods.html)
- There is also IPv6!
- IPv6 header has “only” 8 fields

IPv4 vs IPv6


IPv4 Header
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
IPv6 Header
Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address
Legend
Field's Name Kept from IPv4 to IPv6
Fields Not Kept in IPv6
Name and Position Changed in IPv6
New Field in IPv6

Some existing methods

- Tons of methods
  (More at http://nmap.org/book/osdetect-methods.html)
- There is also IPv6!
- IPv6 header has “only” 8 fields
- With all EHs its 38
- Also ICMPv6 and NDP

→ Why not do OS detection in IPv6?

Test setup

- 5 different VMs
  - Windows XP, SP3
  - Linux 3.2.0
  - Windows 8
  - OpenBSD 5.4
  - Solaris 11.1
- Analyze responses in various scenarios
- Only IPv6-related things
  A lot of other good methods, but this is not our focus here

What Nmap does so far I

- Sends a ping
  - Reminder: ICMPv6 type 128, code 0
  - Answer: ICMPv6 type 129, code 0
- 120 byte of data
- Hop by Hop EH (padding only)
- Code 7, not 0

What Nmap does so far I

Responses to ICMPv6 type 128 code 7:

System	Answer
Windows XP SP3	ICMPv6 type 129, code 0
Linux 3.2.0	ICMPv6 type 129, code 7
Windows 8	ICMPv6 type 129, code 0
OpenBSD 5.4	No Response
Solaris 11.1	ICMPv6 type 129, code 7

What Nmap does so far II

- Another ping
- No data
- Code 0
- EH: Hop by Hop, Destination Options, Routing, Hop by Hop

What Nmap does so far II

Responses to EH: Hop by Hop, Destination Options, Routing, Hop by Hop

System	Answer
Windows XP SP3	ICMPv6 unrecognized NH
Linux 3.2.0	ICMPv6 unrecognized NH
Windows 8	ICMPv6 unrecognized NH
OpenBSD 5.4	ICMPv6 Erroneous hdr field
Solaris 11.1	ICMPv6 unrecognized NH

What Nmap does so far III

- RFC 4620: IPv6 Node Information Queries
  Asking for IPv4-, IPv6-Addresses, Hostname
- Type 139 (ICMP Node Information Query)
- qtype: 4 (IPv4 address)

What Nmap does so far III

Responses to NI Query for IPv4 addresses

System	Answer
Windows XP SP3	No Response
Linux 3.2.0	No Response
Windows 8	ICMPv6 Erroneous hdr field
OpenBSD 5.4	Returns IPv4 address(es)
Solaris 11.1	No Response

What Nmap does so far IV

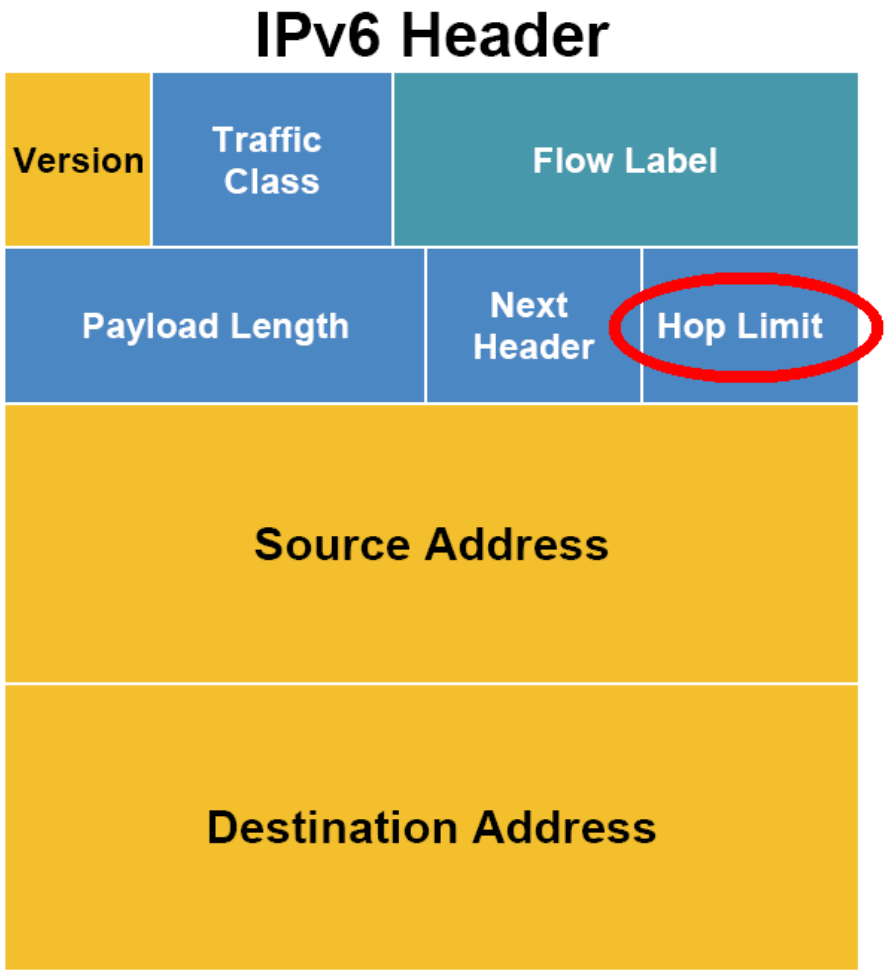
- Neighbor Solicitation
- ICMPv6 type: 135, code: 0
- Flags are all 0
- Only if target on the same subnet

What Nmap does so far IV

Responses to NS

System	Answer
Windows XP SP3	Flags: Solicited, Override
	ND Option: Destination LL Address
Linux 3.2.0	Flags: Solicited
Windows 8	Flags: Solicited, Override
	ND Option: Destination LL Address
OpenBSD 5.4	Flags: Solicited
Solaris 11.1	Flags: Solicited, Override
	ND Option: Destination LL Address

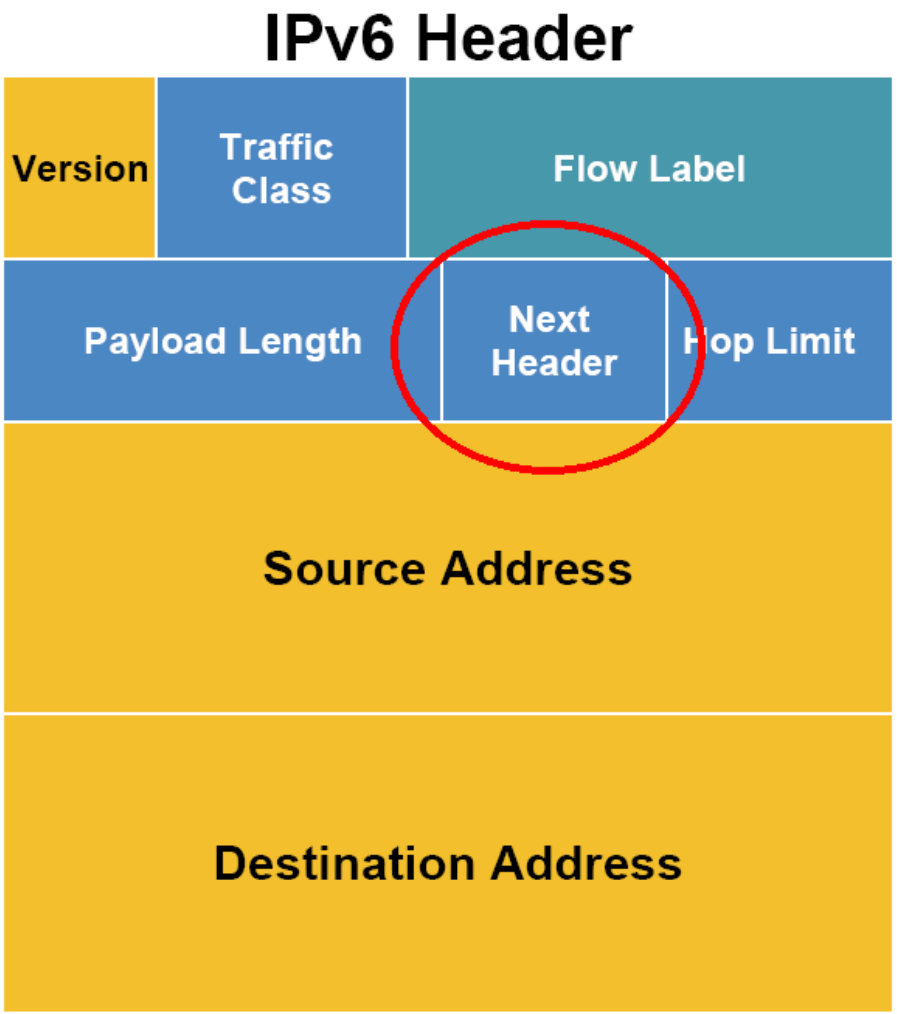
IPv6 Hop Limit



Different Hop Limits

System	Value
Windows XP SP3	128
Linux 3.2.0	64
Windows 8	128
OpenBSD 5.4	64
Solaris 11.1	255

IPv6 Next Header



What if NH set to ...

- ICMP (IPv4)?

Responses to NH=ICMP

System	Answer
Windows XP SP3	ICMPv6 Parameter Problem
Linux 3.2.0	ICMPv6 Parameter Problem
Windows 8	ICMPv6 Parameter Problem
OpenBSD 5.4	No Reply
Solaris 11.1	ICMPv6 Parameter Problem

What if NH set to ...

- ICMP (IPv4)?
- IPv4?

Responses to NH=IPv4

System	Answer
Windows XP SP3	ICMPv6 Parameter Problem
Linux 3.2.0	ICMPv6 Parameter Problem
Windows 8	No reply
OpenBSD 5.4	No reply
Solaris 11.1	ICMPv6 Parameter Problem

What if NH set to ...

- ICMP (IPv4)?
- IPv4?
- AH? (with invalid data)

Responses to NH=AH

System	Answer
Windows XP SP3	No reply
Linux 3.2.0	ICMPv6 Parameter Problem
Windows 8	No reply
OpenBSD 5.4	No reply
Solaris 11.1	No reply

What if NH set to ...

- ICMP (IPv4)?
- IPv4?
- AH?
- Ethernet within IP?

Responses to NH=Ethernet within IP

System	Answer
Windows XP SP3	ICMPv6 Parameter Problem
Linux 3.2.0	ICMPv6 Parameter Problem
Windows 8	ICMPv6 Parameter Problem
OpenBSD 5.4	No reply
Solaris 11.1	ICMPv6 Parameter Problem

New stuff in IPv6

- Node Information Query
- Already seen NI Query for IPv4

Reminder: NI Query for IPv4

System	Answer
Windows XP SP3	No reply
Linux 3.2.0	No reply
Windows 8	ICMPv6 Parameter Problem
OpenBSD 5.4	Returns IPv4 addresses
Solaris 11.1	No reply

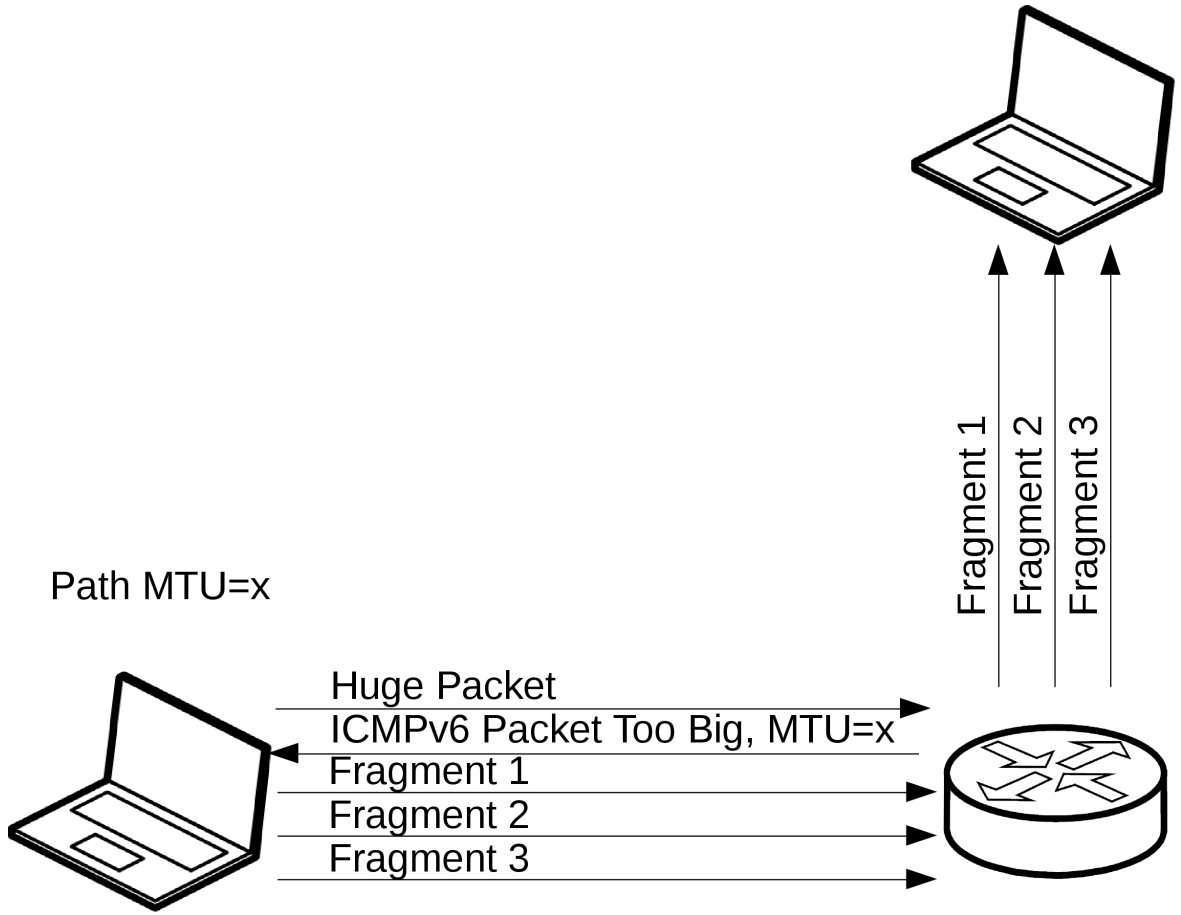
New stuff in IPv6

- Node Information Query
- Already seen NI Query for IPv4
- And now for IPv6

NI Query for IPv6

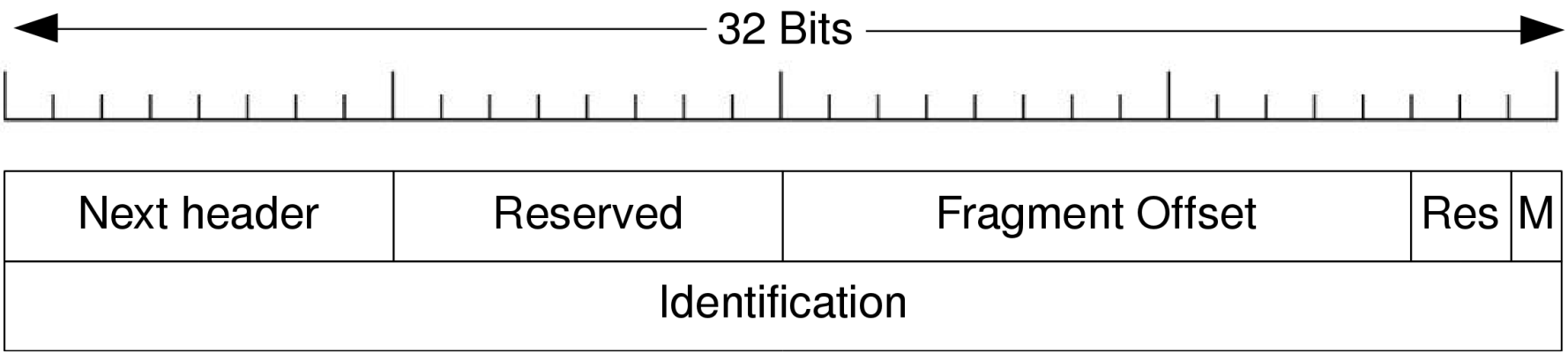
System	Answer
Windows XP SP3	No reply
Linux 3.2.0	No reply
Windows 8	ICMPv6 unrecognized next header
OpenBSD 5.4	No reply
Solaris 11.1	No reply

Fragmentation in IPv6



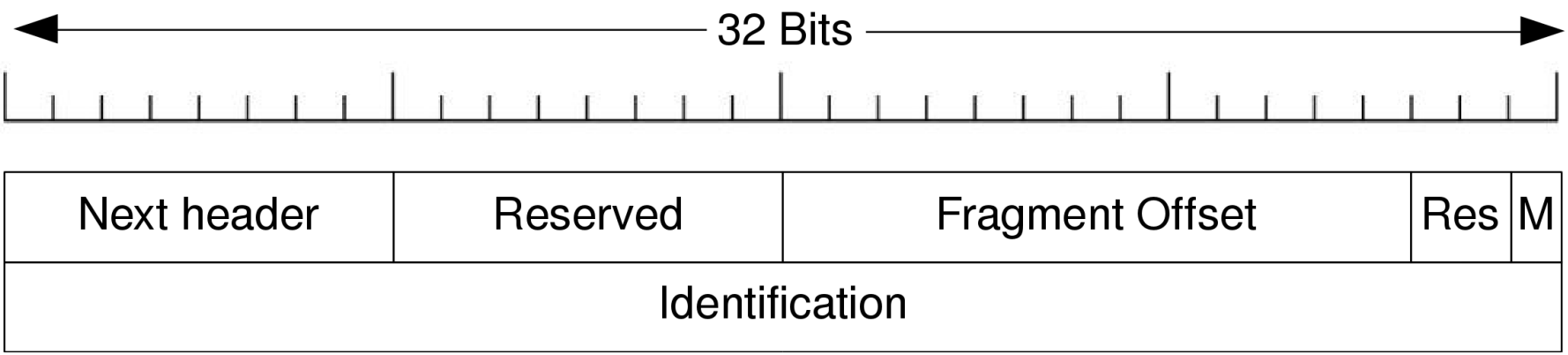
Fragmentation in IPv6

- Extension header used when needed
- Located between IPv6 and TCP header
- Extension header for fragmentation / Fragmentation header:



Fragmentation in IPv6

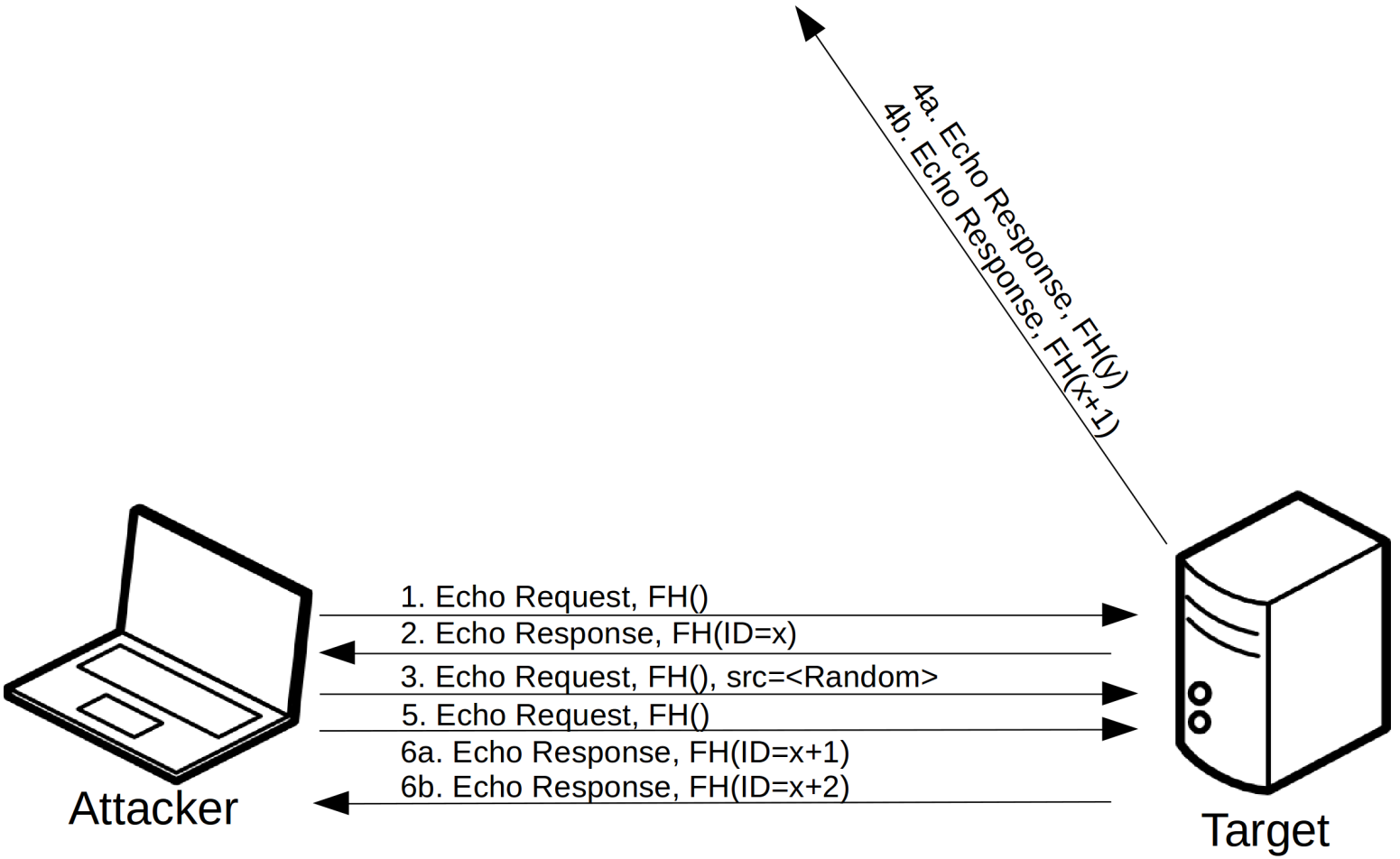
- Assignment of Identification varied already in IPv4
- Send a fragmented Echo Request
  $\rightarrow$ get a fragmented Echo Response



Assignment of Identification value

#	System	Assignment of Identification
1	Android 4.1 (Linux 3.0.15)	Per host, incremental
2	FreeBSD 7.4	Random
3	FreeBSD 9.1	Random
4	iOS 6.1.2	Random
5	Linux 2.6.32	Per host, incremental
6	Linux 3.2	Per host, incremental
7	Linux 3.8	Per host, incremental
8	OpenBSD 4.6	Random
9	OpenBSD 5.2	Random
10	OS X 10.6.7	Global, incremental
11	OS X 10.8.3	Random
12	Solaris 11	Per host, incremental
13	Windows Server 2003 R2 Standard 64bit, SP2	Global, incremental
14	Windows Server 2008 Standard 32bit, SP1	Global, incremental
15	Windows Server 2008 R2 Standard 64bit, SP1	Global, incremental by 2
16	Windows Server 2012 Standard 64bit	Global, incremental by 2
17	Windows XP Professional 32bit, SP3	Global, incremental
18	Windows Vista Business 64bit, SP1	Global, incremental
19	Windows 7 Home Premium 32bit, SP1	Global, incremental by 2
20	Windows 7 Ultimate 32bit, SP1	Global, incremental by 2
21	Windows 8 Enterprise 32 bit	Global, incremental by 2

How to differ between per host and global assignment



Multiple FHs

- Thats fragmentation in IPv6
- RFC 2460 says most EHs SHOULD only occur once

  $\rightarrow$ How about multiple FHs?

Responses to 2 FHs

System	Answer
Windows XP SP3	Reply (without FH)
Linux 3.2.0	Reply (without FH)
Windows 8	Reply (without FH)
OpenBSD 5.4	No Reply
Solaris 11.1	Reply (without FH)

How about 5 FHs?

System	Answer
Windows XP SP3	Reply (without FH)
Linux 3.2.0	Reply (without FH)
Windows 8	Reply (without FH)
OpenBSD 5.4	No Reply
Solaris 11.1	Reply (without FH)

Playing with the MTU

- How about a ping with 1295 byte?

A 1295 byte ping

System	Answer
Windows XP SP3	Reply (1 Packet)
Linux 3.2.0	Reply (1 Packet)
Windows 8	Reply (1 Packet)
OpenBSD 5.4	Reply (In fragments)
Solaris 11.1	Reply (1 Packet)

Playing with the MTU

- How about a ping with 1295 byte?
- MTU in Ethernet is 1500
- How about sending a 1501 byte packet?

A 1501 byte ping

System	Answer
Windows XP SP3	Reply (1 Packet)
Linux 3.2.0	Reply (1 Packet)
Windows 8	Reply (1 Packet)
OpenBSD 5.4	Reply (In fragments)
Solaris 11.1	Reply (1 Packet)

Playing with the MTU

- How about a ping with 1295 byte?
- MTU in Ethernet is 1500
- How about sending a 1501 byte packet?
- Anything special about 1509 byte?

A 1509 byte ping

System	Answer
Windows XP SP3	Reply (1 Packet)
Linux 3.2.0	Reply (1 Packet)
Windows 8	No Reply
OpenBSD 5.4	Reply (In fragments)
Solaris 11.1	Reply (1 Packet)

Playing with the MTU

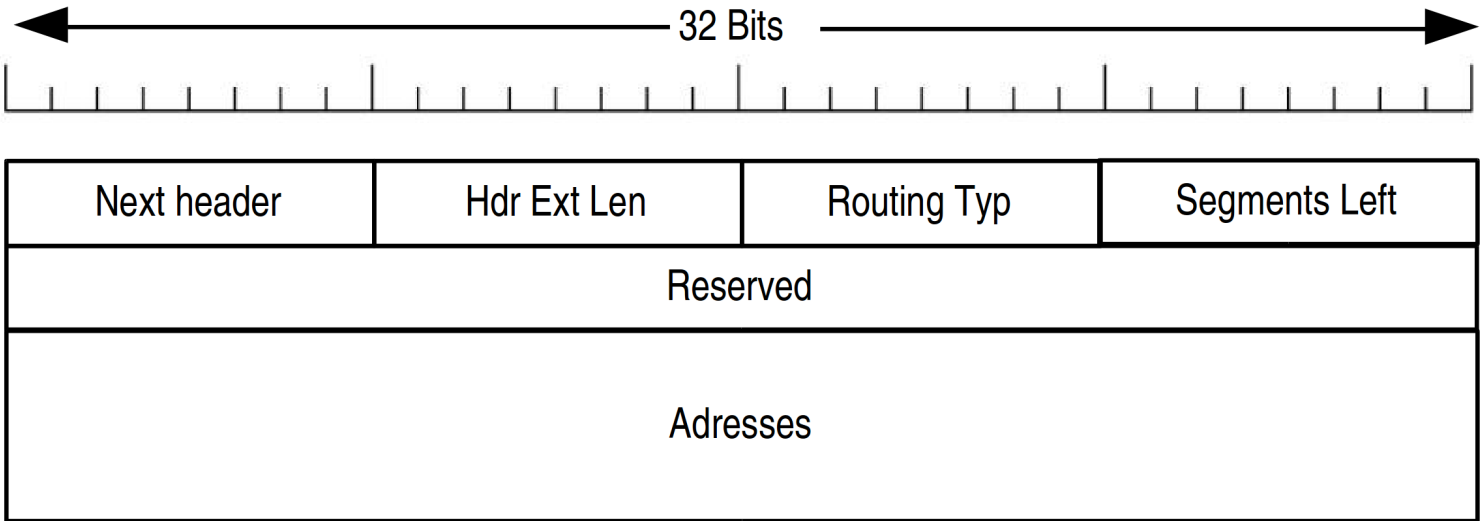
- How about a ping with 1295 byte?
- MTU in Ethernet is 1500
- How about sending a 1501 byte packet?
- Anything special about 1509 byte?
- So what's happening at 1815 byte?

A 1815 byte ping

System	Answer
Windows XP SP3	No Reply
Linux 3.2.0	No Reply
Windows 8	No Reply
OpenBSD 5.4	Reply (In fragments)
Solaris 11.1	No Reply

Routing Header

- List intermediate nodes
- Behavior of destination depends on Segments Left-field



Sending wrong Routing Headers

- Hdr Ext Len set to 0

Responses to Hdr Ext Len set to 0

System	Answer
Windows XP SP3	Replies normally
Linux 3.2.0	Replies normally
Windows 8	Replies normally
OpenBSD 5.4	ICMPv6 Erroneous hdr field
Solaris 11.1	Replies normally

Sending wrong Routing Headers

- Hdr Ext Len set to 0
- Multiple RHs

Responses to multiple RHs

System	Answer
Windows XP SP3	Replies normally
Linux 3.2.0	Replies normally
Windows 8	Replies normally
OpenBSD 5.4	ICMPv6 Erroneous hdr field
Solaris 11.1	Replies normally

Sending wrong Routing Headers

- Hdr Ext Len set to 0
- Multiple RHs
- Segments left = 1 but no addresses

Responses with segments left = 1

System	Answer
Windows XP SP3	Replies normally
Linux 3.2.0	Replies normally
Windows 8	Replies normally
OpenBSD 5.4	Replies normally
Solaris 11.1	No reply

Sending wrong Routing Headers

- Hdr Ext Len set to 0
- Multiple RHs
- Segments left = 1 but no addresses
- Segments left = 1 and one address

Responses with segments left = 1 and one address

System	Answer
Windows XP SP3	Dest. unreachable, Administratively prohibited
Linux 3.2.0	ICMPv6 Erroneous hdr field
Windows 8	ICMPv6 Erroneous hdr field
OpenBSD 5.4	ICMPv6 Erroneous hdr field
Solaris 11.1	No reply

Sending wrong Routing Headers

- Hdr Ext Len set to 0
- Multiple RHs
- Segments left = 1 but no addresses
- Segments left = 1 and one address
- Segments left = 1 and two addresses

Responses with segments left = 1 and two address

System	Answer
Windows XP SP3	Dest. unreachable, Administratively prohibited
Linux 3.2.0	ICMPv6 Erroneous hdr field
Windows 8	ICMPv6 Erroneous hdr field
OpenBSD 5.4	ICMPv6 Erroneous hdr field
Solaris 11.1	No reply

Sending wrong Routing Headers

- Hdr Ext Len set to 0
- Multiple RHs
- Segments left = 1 but no addresses
- Segments left = 1 and one address
- Segments left = 1 and two addresses
- Segments left = 0 and one address

Responses with segments left = 0 and one address

System	Answer
Windows XP SP3	Replies normally
Linux 3.2.0	Replies normally
Windows 8	Replies normally
OpenBSD 5.4	ICMPv6 Erroneous hdr field
Solaris 11.1	Replies normally

Os detection with firewalls

- NDP still works :)

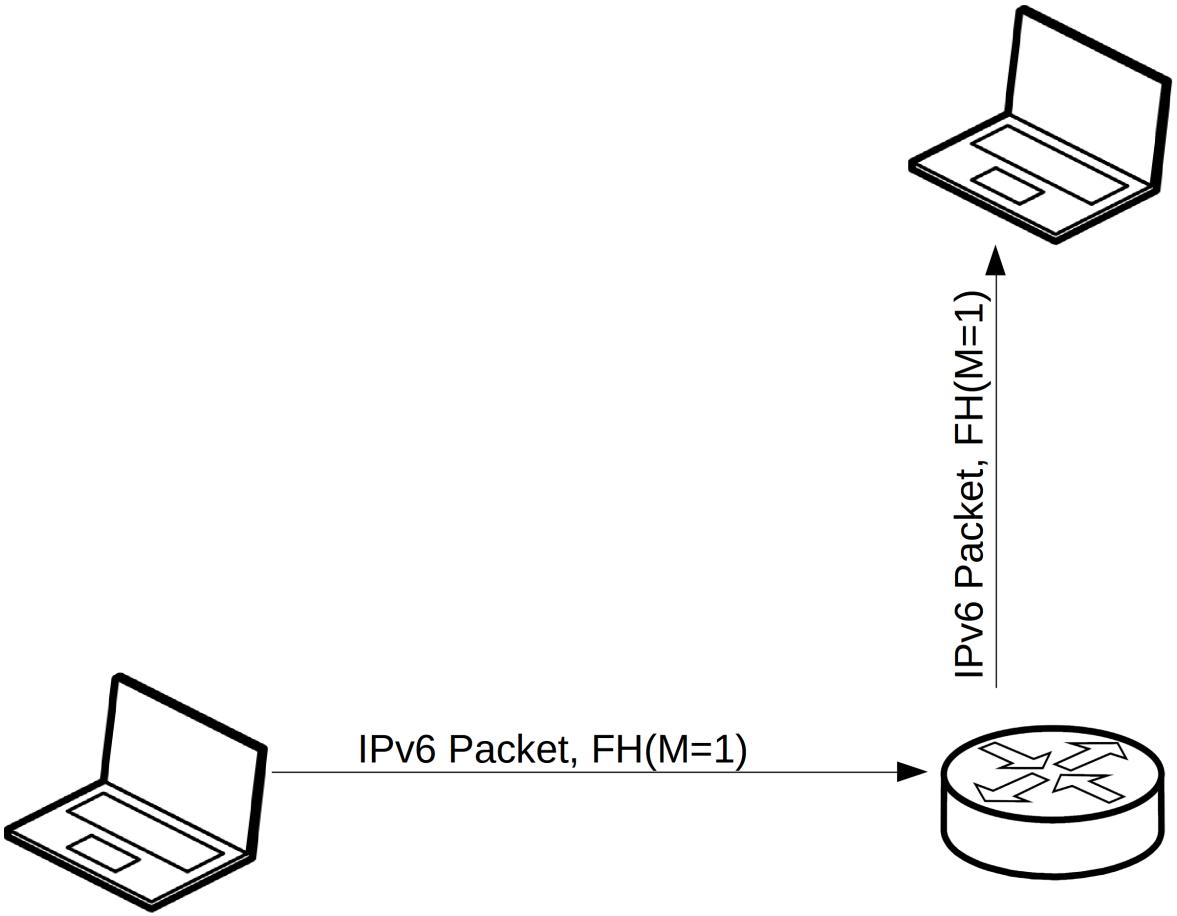
Reminder: Responses to NS

System	Answer
Windows XP SP3	Flags: Solicited, Override
	ND Option: Destination LL Address
Linux 3.2.0	Flags: Solicited
Windows 8	Flags: Solicited, Override
	ND Option: Destination LL Address
OpenBSD 5.4	Flags: Solicited
Solaris 11.1	Flags: Solicited, Override
	ND Option: Destination LL Address

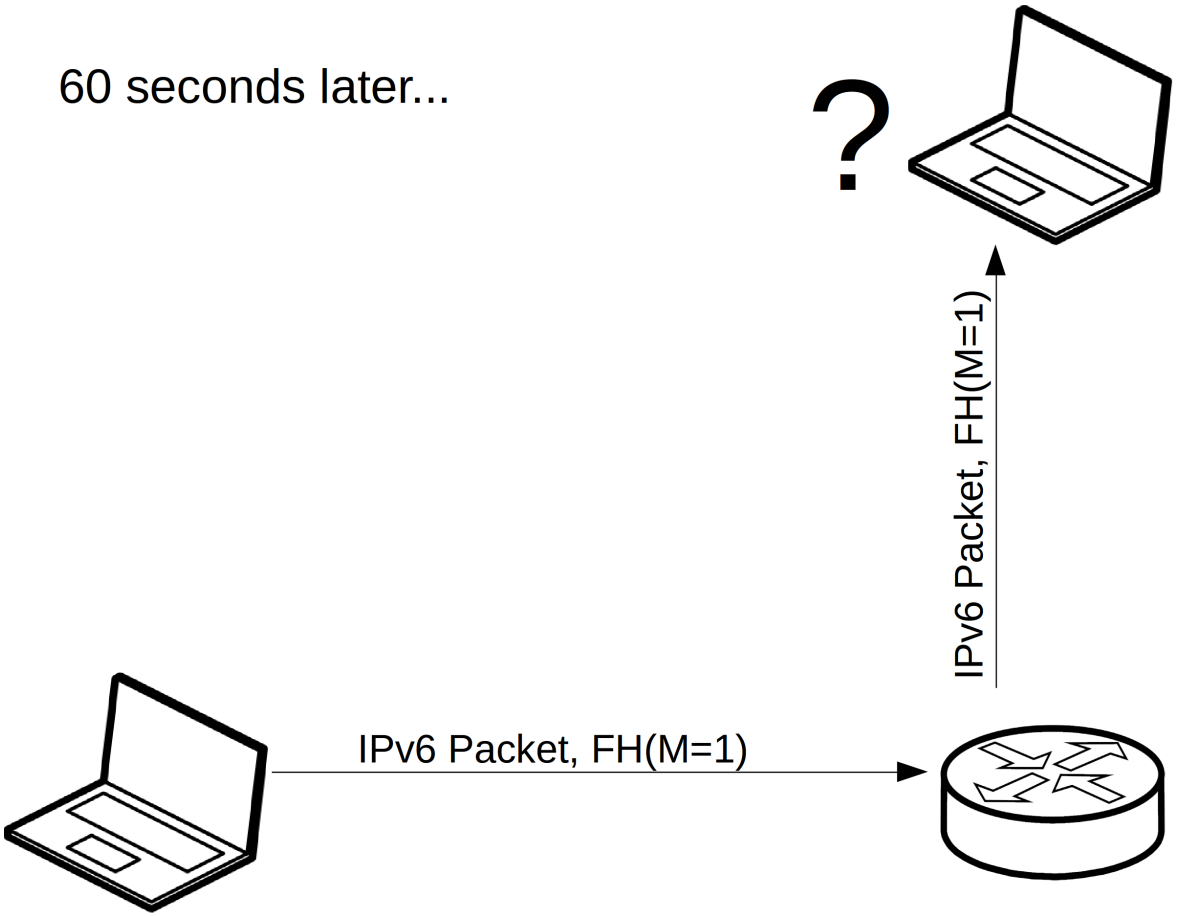
OS detection with firewalls

- NDP still works :)
  → no unique identification possible
- Rest does not :(
- Fragmentation of NDP?
  → Computer says no!
  (RFC 6980: Security Implications of IPv6 Fragmentation with IPv6 ND)
- Approach for targets in other networks?

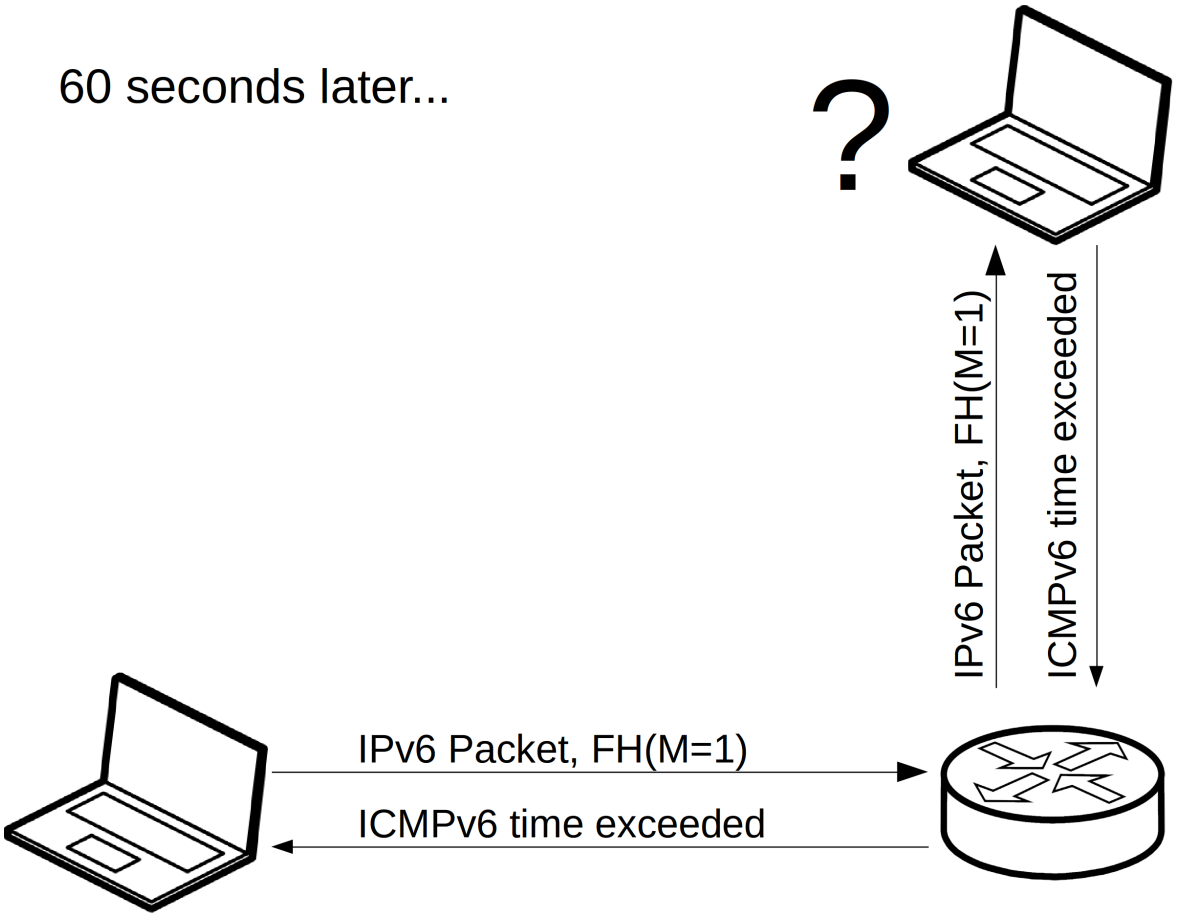
The solution: Time exceeded messages



The solution: Time exceeded messages



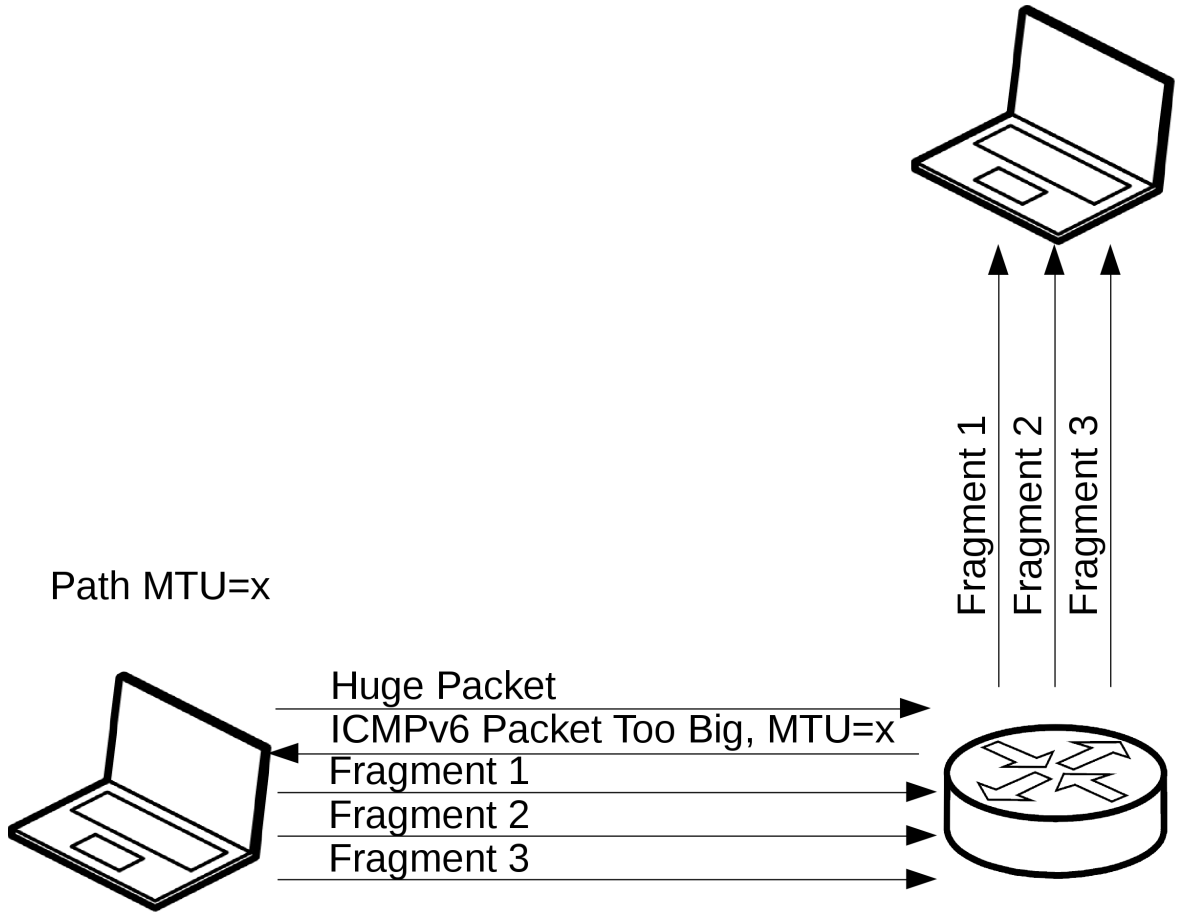
The solution: Time exceeded messages



We have an answer!

- HL allows differentiation between
  - Windows
  - Linux
  - OpenBSD
  - Solaris
- No differentiation between different versions
- Why not try to force the FH in the reply?

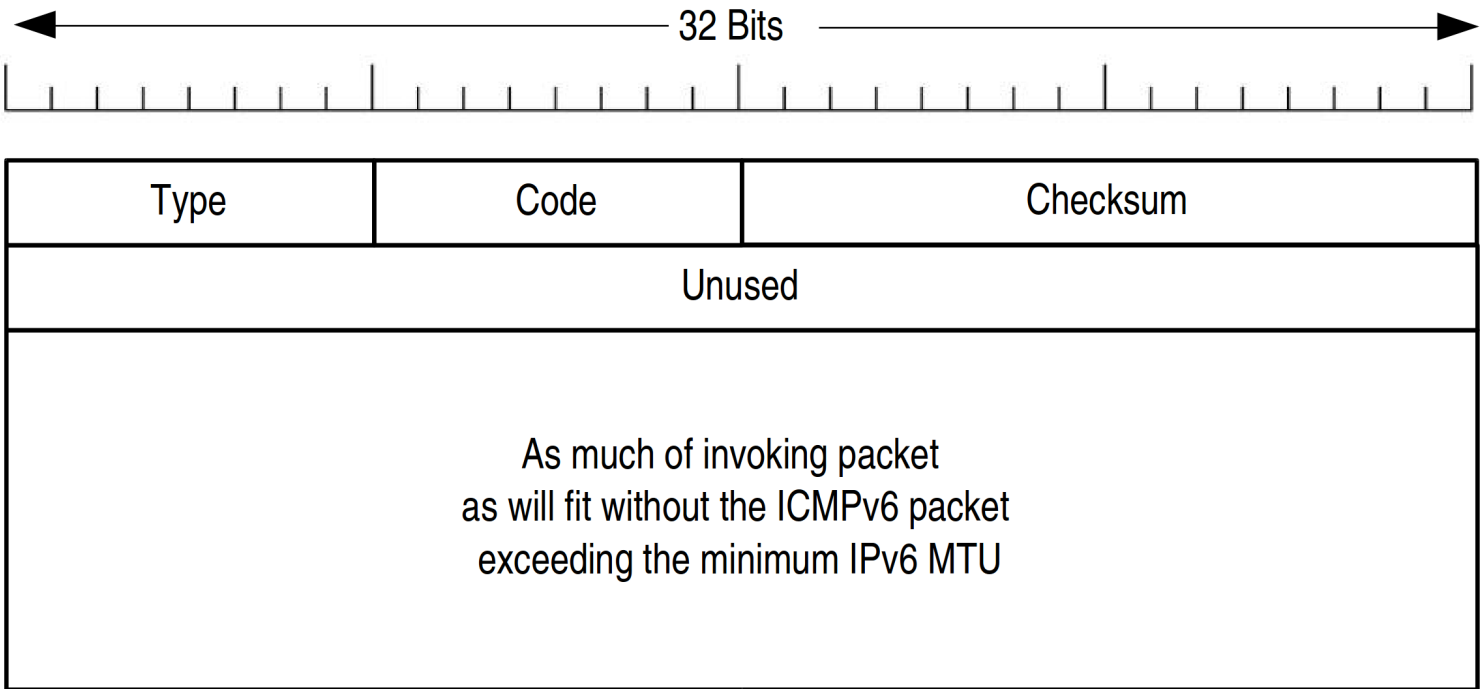
Reminder: Fragmentation in IPv6



Get the FH into ICMPv6 time exceeded

- So we can manipulate another host's Path MTU!
- Minimum IPv6 MTU is 1280
  $\rightarrow$ create a time exceeded message $>$ 1280 byte

ICMPv6 time exceeded message



Get the FH into ICMPv6 time exceeded

- So we can manipulate another host's Path MTU!
- Minimum IPv6 MTU is 1280
  $\rightarrow$ create a time exceeded message $>$ 1280 byte
- ICMPv6 will not be fragmented

Get the FH into ICMPv6 time exceeded

- So we can manipulate another host's Path MTU!
- Minimum IPv6 MTU is 1280
  $\rightarrow$ create a time exceeded message $>$ 1280 byte
- ICMPv6 will not be fragmented
- Let's have a look at RFC 1981

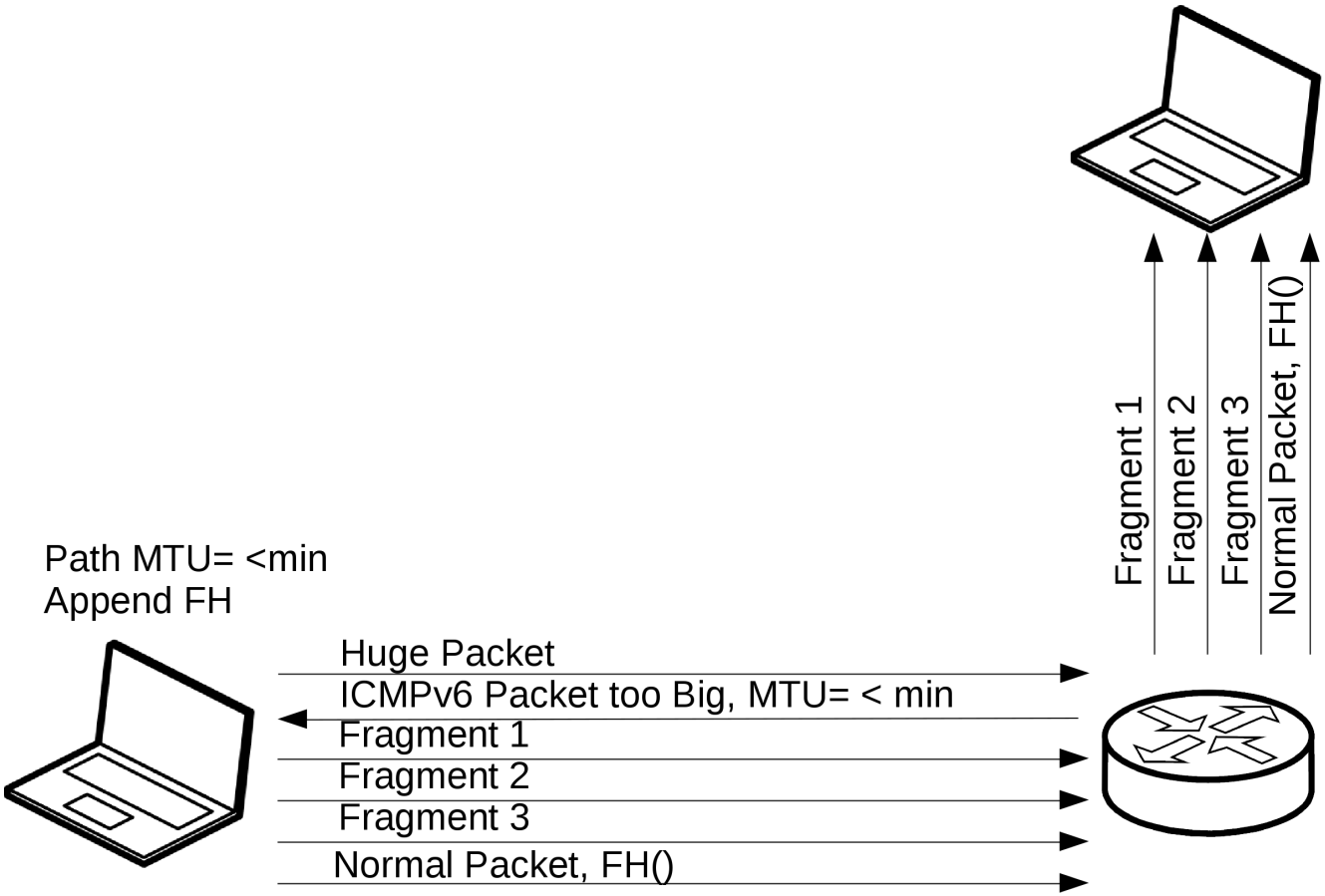
Get the FH into ICMPv6 time exceeded

When a node receives a Packet Too Big message, it MUST reduce its estimate of the PMTU for the relevant path, based on the value of the MTU field in the message

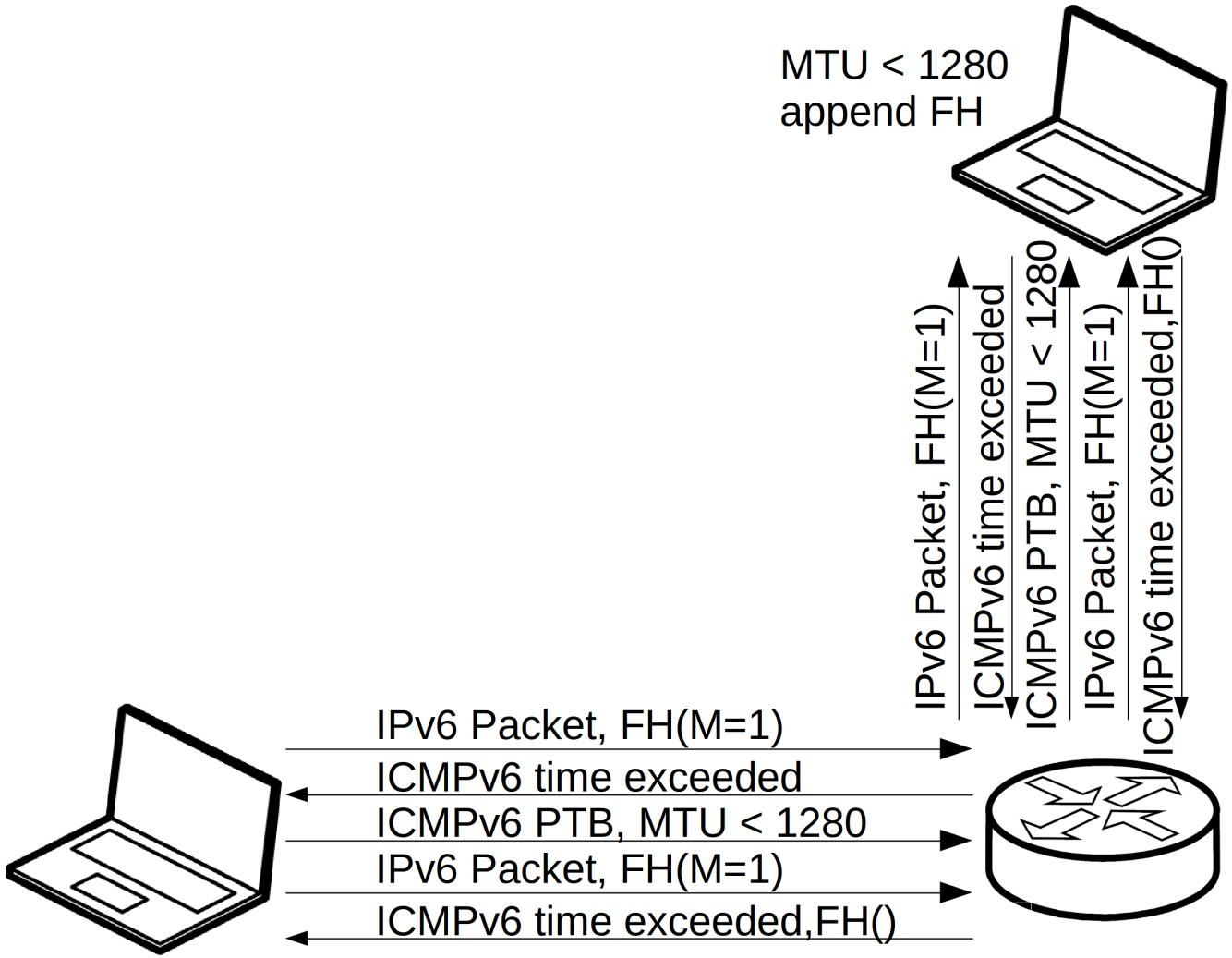
A node MUST NOT reduce its estimate of the Path MTU below the IPv6 minimum link MTU. Note: A node may receive a Packet Too Big message reporting a next-hop MTU that is less than the IPv6 minimum link MTU. In that case, the node is not required to reduce the size of subsequent packets sent on the path to less than the IPv6 minimum link MTU, but rather must include a Fragment header in those packets

(RFC 1981, Path MTU Discovery for IP version 6)
AKA “atomic fragments”, credits to Fernando Gont

Get the FH into ICMPv6 time exceeded



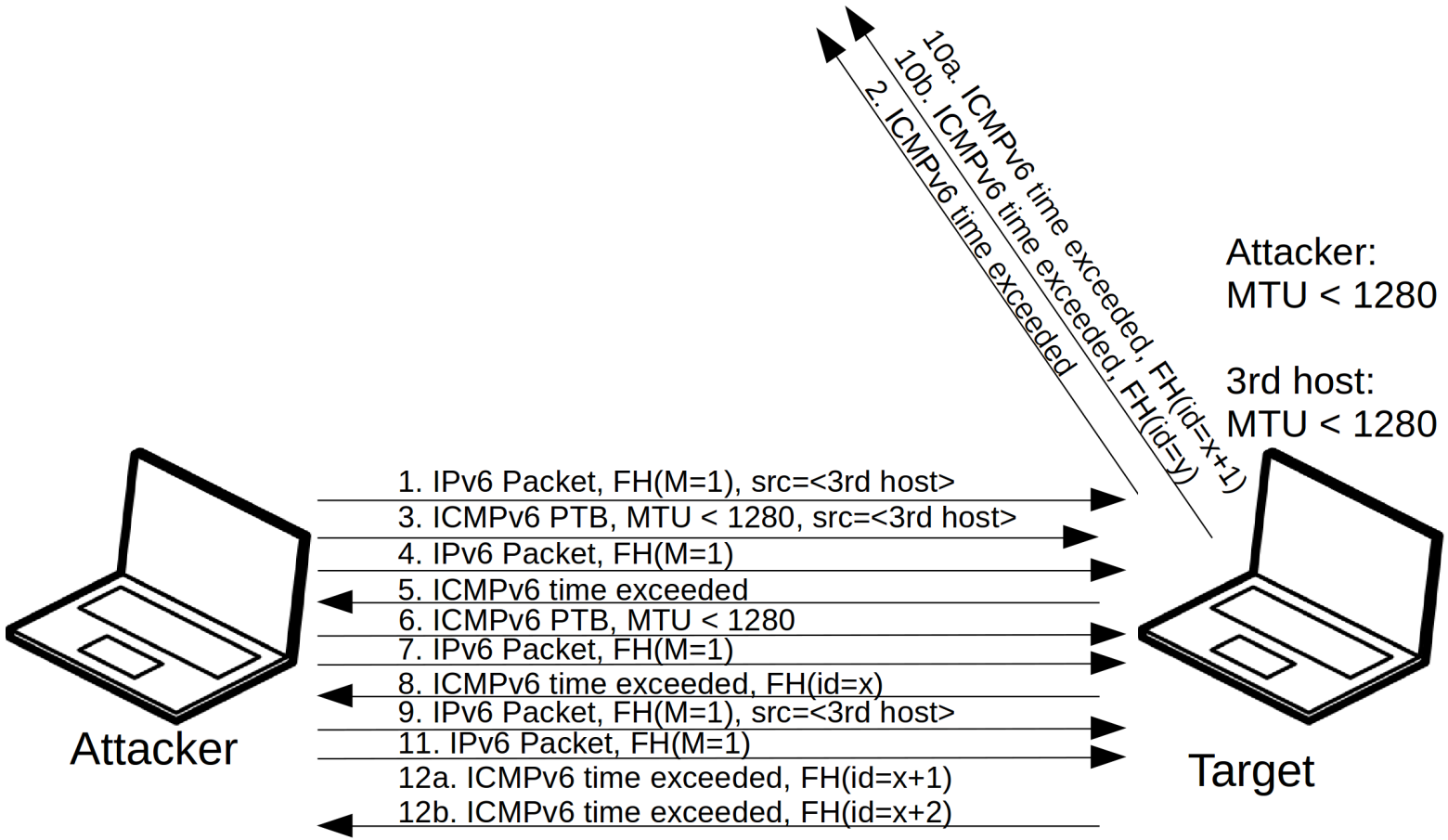
Get the FH into ICMPv6 time exceeded



Now we have a FH with an ID!

#	System	Assignment of Identification
1	Android 4.1 (Linux 3.0.15)	Per host, incremental
2	FreeBSD 7.4	Random
3	FreeBSD 9.1	Random
4	iOS 6.1.2	Random
5	Linux 2.6.32	Per host, incremental
6	Linux 3.2	Per host, incremental
7	Linux 3.8	Per host, incremental
8	OpenBSD 4.6	Random
9	OpenBSD 5.2	Random
10	OS X 10.6.7	Global, incremental
11	OS X 10.8.3	Random
12	Solaris 11	Per host, incremental
13	Windows Server 2003 R2 Standard 64bit, SP2	Global, incremental
14	Windows Server 2008 Standard 32bit, SP1	Global, incremental
15	Windows Server 2008 R2 Standard 64bit, SP1	Global, incremental by 2
16	Windows Server 2012 Standard 64bit	Global, incremental by 2
17	Windows XP Professional 32bit, SP3	Global, incremental
18	Windows Vista Business 64bit, SP1	Global, incremental
19	Windows 7 Home Premium 32bit, SP1	Global, incremental by 2
20	Windows 7 Ultimate 32bit, SP1	Global, incremental by 2
21	Windows 8 Enterprise 32 bit	Global, incremental by 2

Differentiation between per host and global assignment still possible



Acceptance of PTB messages without prev. traffic

System	Answer
Windows XP SP3	Yes
Linux 3.2.0	Yes
Windows 8	No
OpenBSD 5.4	Yes
Solaris 11.1	Yes

Funny stuff with PTB and PMTU < 1280

- Linux 2.6.32: No packets sent on route
- Linux 3.0.15 & Linux 3.2: Calculate wrong TCP checksum
- OS X 10.6.7: PTB with PMTU < 1280 is dropped

This is becoming pretty accurate

#	System	Assignment of Identification
1	Android 4.1 (Linux 3.0.15)	Per host, incremental (1)
2	FreeBSD 7.4	Random
3	FreeBSD 9.1	Random
4	iOS 6.1.2	Random
5	Linux 2.6.32	Per host, incremental (2)
6	Linux 3.2	Per host, incremental (1)
7	Linux 3.8	Per host, incremental
8	OpenBSD 4.6	Random
9	OpenBSD 5.2	Random
10	OS X 10.6.7	Global, incremental (3)
11	OS X 10.8.3	Random
12	Solaris 11	Per host, incremental
13	Windows Server 2003 R2 Standard 64bit, SP2	Global, incremental
14	Windows Server 2008 Standard 32bit, SP1	Global, incremental
15	Windows Server 2008 R2 Standard 64bit, SP1	Global, incremental by 2
16	Windows Server 2012 Standard 64bit	Global, incremental by 2
17	Windows XP Professional 32bit, SP3	Global, incremental
18	Windows Vista Business 64bit, SP1	Global, incremental
19	Windows 7 Home Premium 32bit, SP1	Global, incremental by 2
20	Windows 7 Ultimate 32bit, SP1	Global, incremental by 2
21	Windows 8 Enterprise 32 bit	Global, incremental by 2 (4)

(1) Hosts calculates wrong TCP checksum for routes with PMTU $<$1280
(2) PMTU $<$1280 results in DoS
(3) Does not accept PMTU $<$1280
(4) Requires previous too big message piggybagged in ICMPv6 PTB
(Not tested with all systems)

Conclusion

- A lot of new possibilities for OS detection in IPv6
  Can be combined with others (TCP, ...)
- More to discover (like Inverse NS)
- Also firewalls can't stop us
- OpenBSD always behaves different ;)