CS 425: Computer Networks **Jan-Apr 2020**

Lecture 17: Computer Networks – March 4, 2020

Lecturer: Swaprava Nath Scribe(s): Akarsh Goyal, Yash Choudhary, Siddhant Sarkar, Abhishek Kumar

Introduction

In the last lecture, we discussed Fragmentation, in this lecture we will discuss the following topics -

- Path MTU Discovery using ICMP
- IPv6
- Transition from IPv4 to IPv6
- Network Address Translation

17.1 Path MTU (Maximum Transmission Unit) Discovery

- Each IP packet is sent with a header bits indicating that fragmentation is not allowed.
- If a router receives a packet larger than the MTU of the link it is to be forwarded in, it generates an error packet, sends it to the source, and drops the IP packet. If the packet size is smaller, it is sent to the next node.
- If the source receives an error packet, it uses information inside it to refragment packets into smaller pieces and tries sending them again.

Let us take up an example to understand the execution.

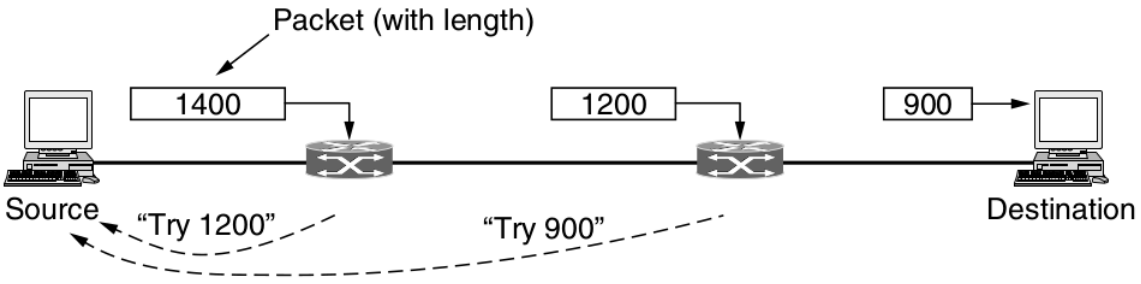

Figure 17.1: Execution of path MTU discovery

1. Packet of size 1400 units is sent by *source* to the first router.

2. As size 1400 units is greater than the MTU of the link it is to be forwarded on, which is 1200, the router sends an error packet "TRY 1200" to *source* and discards the 1400 unit sized packet.

3. *Source* retransmits a 1200 unit sized packet.

4. The packet is forwarded by the first router to the second router.

5. The second router sends an error packet "TRY 900" to *source* and discards the 1200 unit sized packet.

6. *Source* sends a packet of size 900 units this time and it successfully reaches *destination* after 2 hops.

17.1.1 Features of MTU Path Discovery

1. **Message Passing Approach:** Devices collectively send messages to host to decide the right packet size.

2. **Dependence on Path:** Path MTU can change if routing tables change over time. Different paths may have a different minimum allowed packet size.

3. **Implemented using ICMP:** ICMP stands for Internet Control Messaging Protocol. Packet is sent with $DF = 1$ flag. This instructs devices to discard the packet if it is larger and generate an error message.

17.1.2 Error Handling using ICMP

- ICMP is a companion protocol of IP. It sits on top of IP in the protocol stack.

- ICMP packets are carried as IP packets.

Ques: How to identify that ICMP is higher layer protocol and not TCP/UDP?
Ans: Special value Protocol Field is set to 1 in the message.

Structure of ICMP Message

- **Header:** Contains type, code and checksum.

- **Data:** Typically some part of the erroneous message.

<table>
<tr><td>Version Number (4 bits)</td><td>Header Length (4 bits)</td><td>Type of Service (8 bits)</td><td colspan="2">Total Length (16 bits)</td></tr>
<tr><td colspan="3">ID (16 bits)</td><td>Flags (3bits)</td><td>Flag Offset (13 bits)</td></tr>
<tr><td colspan="2">Time To Live (8 bits)</td><td>Protocol (8 bits)</td><td colspan="2">Header Checksum (16 bits)</td></tr>
<tr><td colspan="5">Source (32 bits)</td></tr>
<tr><td colspan="5">Destination (32 bits)</td></tr>
<tr><td colspan="5">Options</td></tr>
</table>

Figure 16.10: IPv4 packet

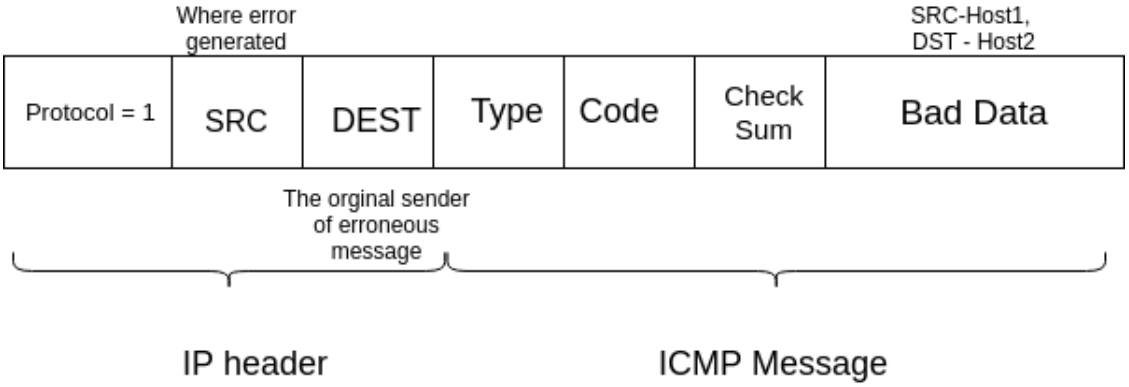

Figure 17.2: Address structure of ICMP message

Type and code determine the kind of error. [IANA] contains a list of types and codes.

Situation	Type	Code	Usage
Data Host unreachable	3	0 or 1	To check connectivity
Data Host (fragmentation)	3	4	Path MTU Discovery
Time exceeded	11	0	traceroute
Echo request/reply	8 or 0	0	Ping(Checks if host is alive or not)

Table 17.1: Some ICMP types and codes

Example of path MTU discovery: For IPv4 packets, Path MTU Discovery works by setting the Don't Fragment (DF) flag bit in the IP headers of outgoing packets. Then, any device along the path whose MTU is smaller than the packet will drop it, and send back an Internet Control Message Protocol (ICMP) Fragmentation Needed (Type 3, Code 4) message containing its MTU, allowing the source host to reduce its Path MTU appropriately. The process is repeated until the MTU is small enough to traverse the entire path without fragmentation.

An example of ICMP message usage - traceroute implementation

Traceroute implementation uses the TTL field. The TTL field is initially set to a very large value. This value decreases each time it passes through a router. When field value reaches 0, an error ICMP message is sent to the host. Loops are also detected this way.

1. Host sends IP packet with TTL=1. The first router sends ICMP message containing source address of the router $\implies$ host knows the first router and its address.
2. Host then sends IP packet with TTL=2. The packet passes through the first router and TTL is decreased to 1. At the next router, TTL is decreased to 0, and it sends back an ICMP message to the source $\implies$ host knows the second router and its address.
3. Host then sends IP packet with TTL=3 and so on. This way, a list of devices and their addresses is created in the implementation of traceroute.

17.2 IPv6

- IPv6 was proposed in 1994 but gained popularity in 2010 due to exhaustion of IPv4 addresses.
- IPv6 uses 128 bits in contrast to 32 bits in IPv4. This gives rise to 2^{128} different addresses. Almost impossible to exhaust in the near future.

- Other minor improvements:
  - Streamlined processing because of introduction of *flow label*. This is used in services like VoIP, where all the packets are equally delayed producing a coherent output on the receiver's side.
  - Better fit with features like mobility, multitasking etc.

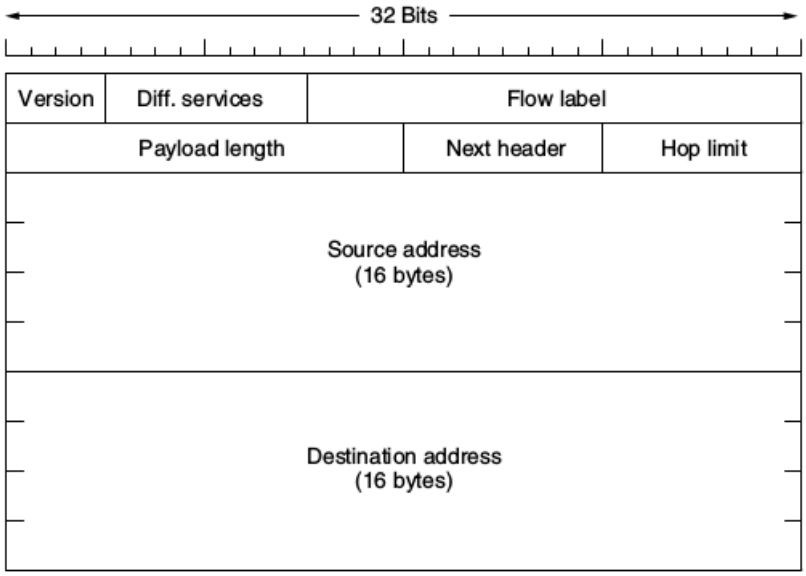

Figure 17.3: Packet structure in IPv6

Address Structure

An IPv6 address is made up of 128 bits divided into 8 groups of 4-digit hexadecimal numbers separated by colon symbols. For example -

2001:0000:3238:DFE1:0063:0000:0000:FEFB

IPv6 provides some rules to shorten the address -

- Discard leading zeroes In the example, 5^{th} block has leading zeroes which can be omitted to -

  2001:0000:3238:DFE1:63:0000:0000:FEFB

- If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign '::', such as (6^{th} and 7^{th} block) -

  2001:0000:3238:DFE1:63::FEFB.

  Consecutive blocks of zeroes can be replaced only once by :: so if there are still blocks of zeroes in the address, they can be shrunk down to a single zero, such as (2^{nd} block) -

  2001:0:3238:DFE1:63::FEFB

  Important: see IPv6 addendum at the end

Transition from IPv4 to IPv6

As the change was slow, some fixes were introduced

- **Dual Stack Routers:** These devices are capable of handling both IPv4 and IPv6 traffic. These are expensive.
- **Translators:** Translators dedicated devices at the edge of the network that perform address translation
- **Tunneling:** In tunneling, an IPv6 packet is encapsulated as an IPv4 and at the other end router, reconstructed back from an IPv4 packet to IPv6 packet. This works only when the source and destination hosts are on the same type of network, but there is a different network in between

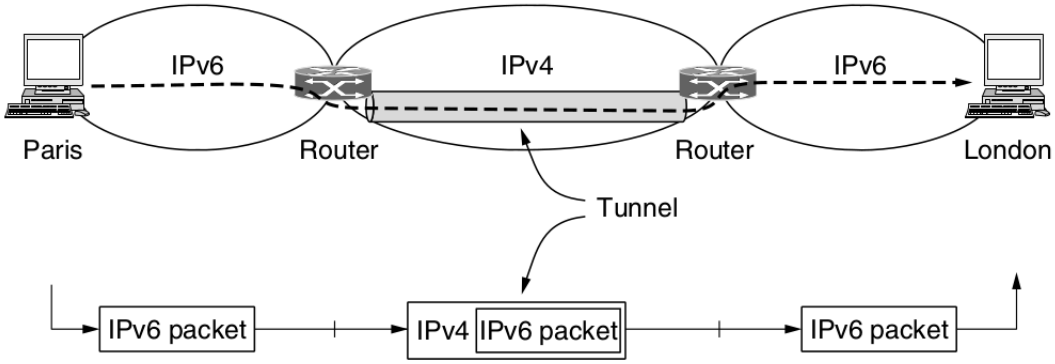

Figure 17.4: Schematic illustration of tunneling

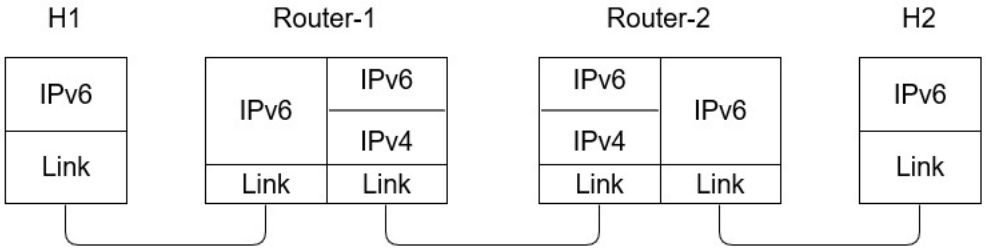

Figure 17.5: Layer Diagram

17.3 Network Address Translation (NAT)

NAT is a technique to counter the lack of new IPv4 addresses.

- It was realized that for large networks, using unique IPv4 addresses is not meaningful.
- NAT assigns *one* IPv4 address for multiple devices in a network (home, business, etc.). Within this network, every computer gets a unique IP address which is used for intramural traffic.
- Whenever devices send a packet outside the network, it passes through a **NAT box** that converts the internal IP source address to the actual outside IP address

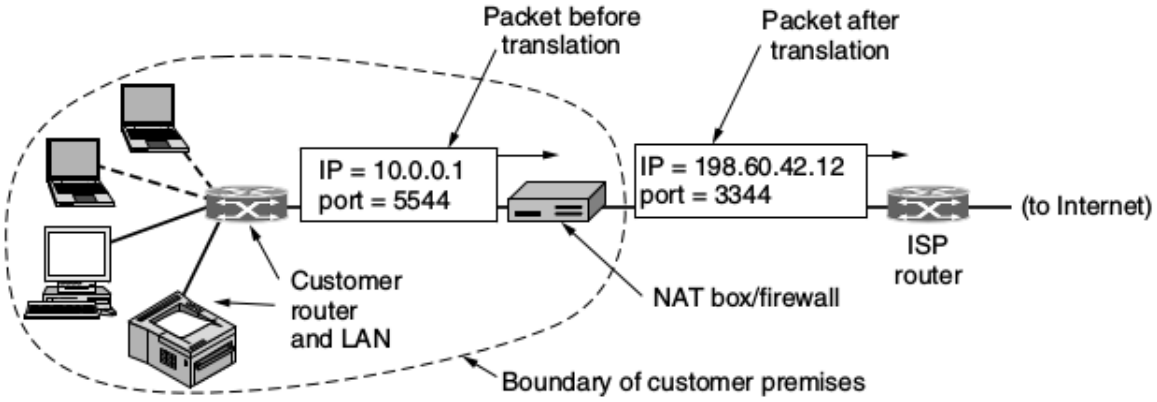

Figure 17.5: Conversion of address by NAT box

Ques: Whenever a response comes from outside the network, how does NAT box know to which device should it be sent?
Ans: [TW] IP packets carry either TCP/UDP payload. Both of these have headers containing a 16-bit source port and a 16-bit destination port. NAT makes use of these ports to maintain the mapping between the inside world and outside world.

Internal IP : port	External IP : port
10.0.0.1:5544	198.60.42.12:3344
10.0.1.2:1234	198.60.42.12:3350
10.0.2.3:1234	198.60.42.12:3354

Table 17.2: Example of NAT mapping

Example

A node H_1 from an internal network wants to send a message to an external node X.
Internal address of H_1 - 10.0.0.1:5544
External address of H_1 - 198.60.42.12:3344
Address of X - 201.3.83.132:1234

H_1 sends an IP packet through the NAT box. The IP packet before translation through NAT has -
$SRC = 10.0.0.1 : 5544$
$DST = 201.3.83.132 : 1234$

The IP packet after translation through NAT has -
$SRC = 198.60.42.12 : 3344$
$DST = 201.3.83.132 : 1234$

If X wants to send a packet to H_1 then the IP packet before translation through NAT has -
$SRC = 201.3.83.132 : 1234$
$DST = 198.60.42.12 : 3344$

The IP packet after translation through NAT has -
$SRC = 201.3.83.132 : 1234$
$DST = 10.0.0.1 : 5544$

Advantages of NAT

- IP addresses conserved.
- Enhances security by for private networks by keeping internal addressing private from the external network

Disadvantages of NAT

- Running servers inside a NAT managed network is difficult - external devices cannot reach to an internal server as there is no mapping in the NAT table.
- Does not work well with connectionless services. eg - UDP

References

[IANA] https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

[TW] AS Tanenbaum, DJ Wetherall, Computer Networks, 5^{th} Ed., Prentice-Hall, 2010

IPv6 addendum:
Only a single contiguous string of all-0s segments can be represented with a double colon; otherwise, the address would be ambiguous, as shown in this example:

Incorrect address using two double colons:

2001::abcd::1234

Possible ambiguous choices:

2001:0000:0000:0000:0000:abcd:0000:1234
2001:0000:0000:0000:abcd:0000:0000:1234
2001:0000:0000:abcd:0000:0000:0000:1234
2001:0000:abcd:0000:0000:0000:0000:1234

As you can see, if two double colons are used, there are multiple possible interpretations, and you don't know which address is the correct one.

What happens if you have an address with more than one contiguous string of all-0s hextets—for example, 2001:0db8:0000:0000:abcd:0000:0000:1234? In that case, where should you use the single double colon (::)?

RFC 5952 states that the double colon should represent:

1. The longest string of all-0s hextets.
2. If the strings are of equal length, the first string should use the double colon (::) notation.

Therefore, 2001:0db8:0000:0000:abcd:0000:0000:1234 would be written 2001:0db8:: abcd:0000:0000:1234.
Applying both Rules 1 and 2, the address would be written 2001:db8::abcd:0:0:1234.