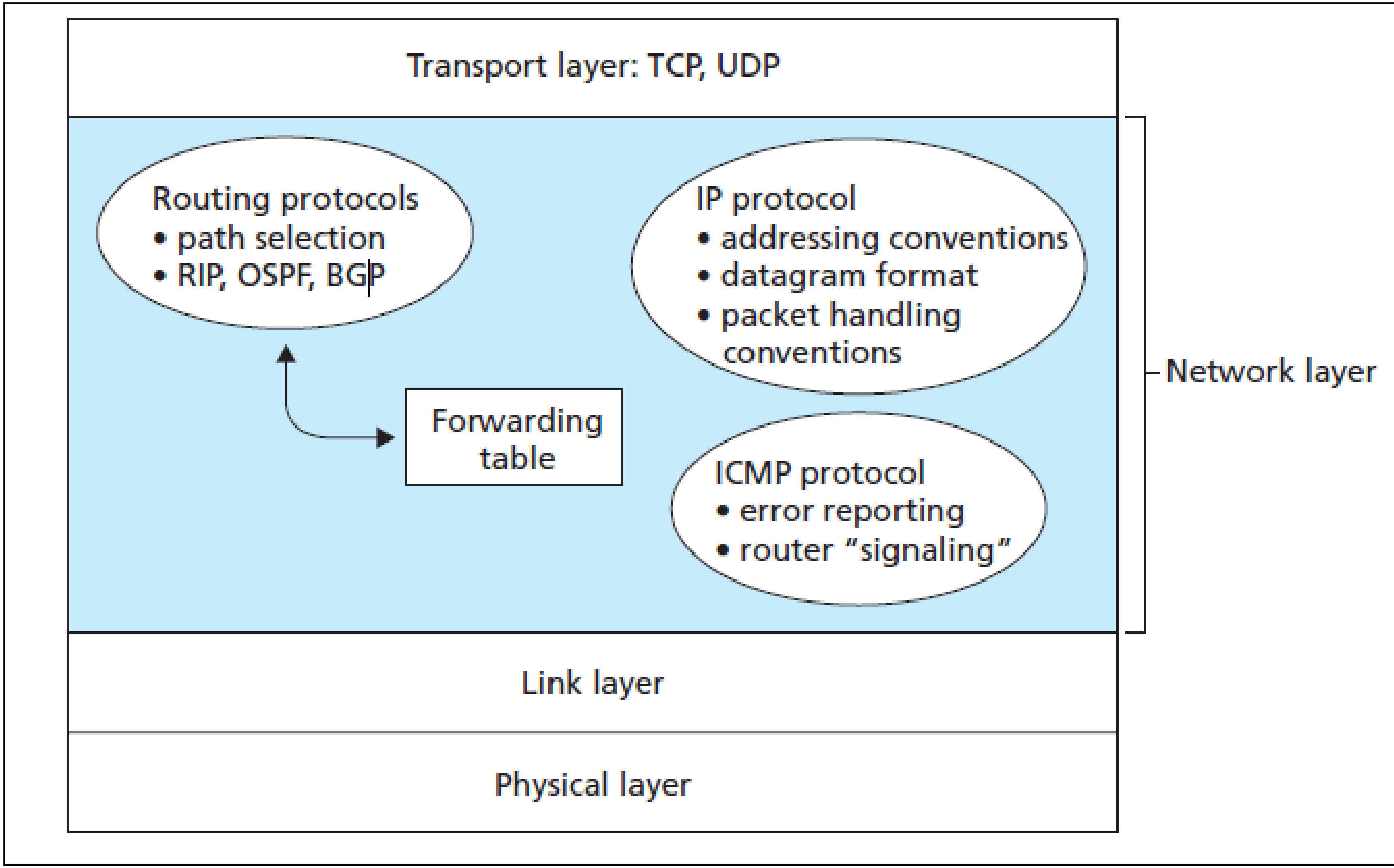
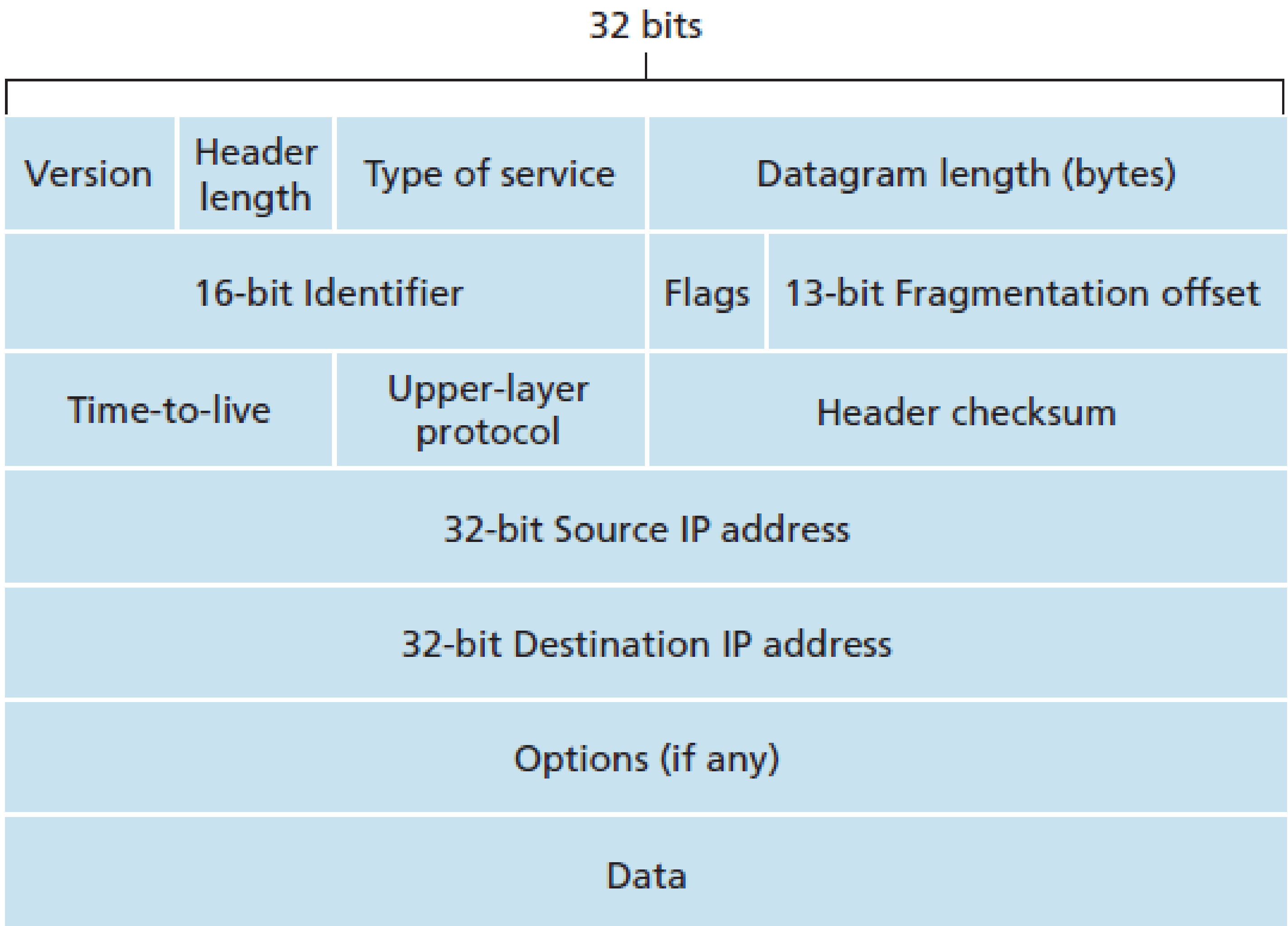
IPV4



Network Layer Overview

IPV4 Datagram Format



IP Datagram Format

- ***Version[4bits]***: Helps router to determine the version of IP protocol.
- ***Header Length[4bits]***: Specifier where in the datagram data actually begins.
- ***Type of service[8bits]***: Allows different types of IP datagrams to be distinguished from one other. Like a IP datagram carrying real-time data from a non real-time traffic.
- ***Datagram Length[16 bits]***: Specifies total length of IP datagram (header+data)

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

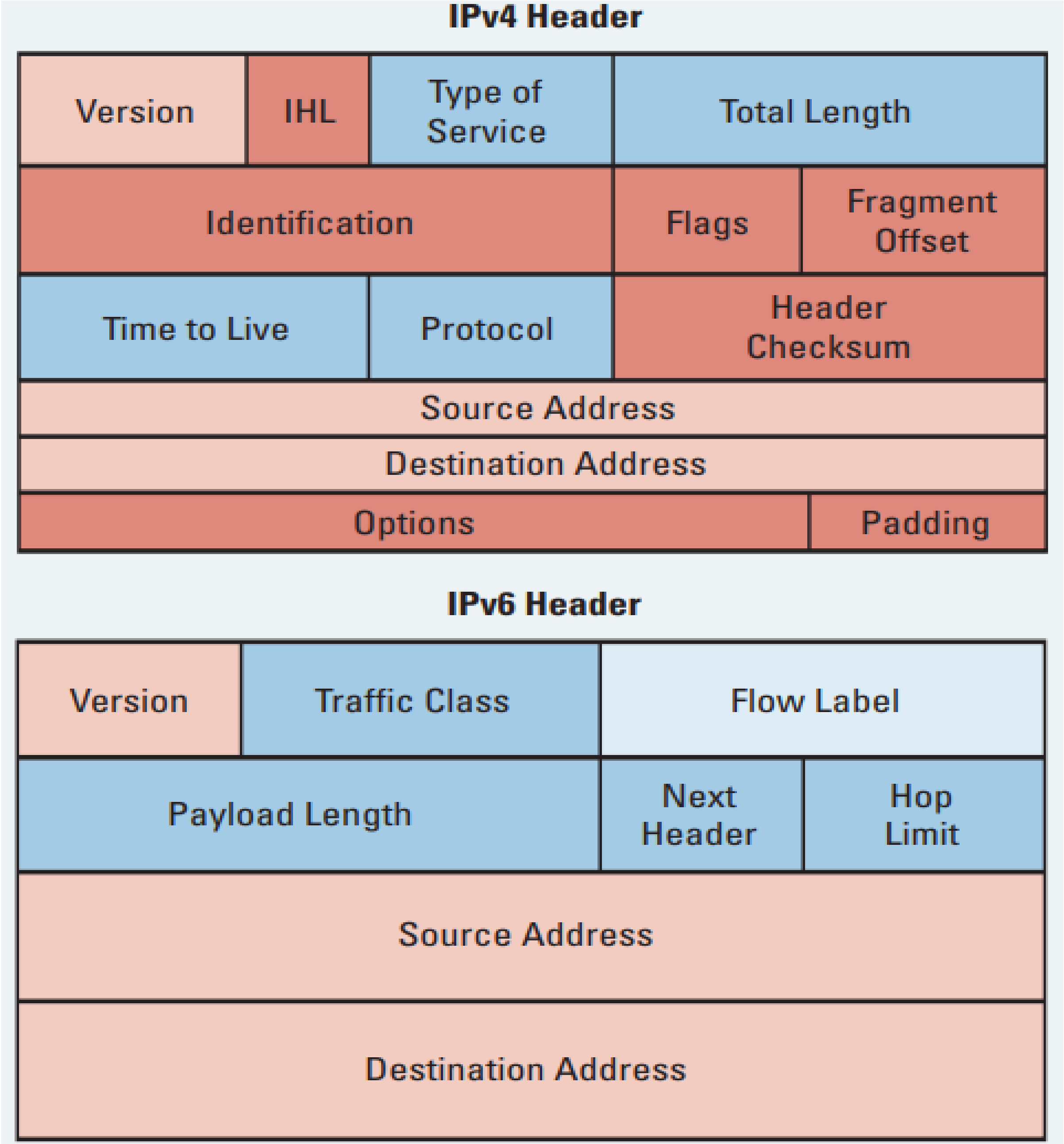
IP Datagram Format

- ***Identifier[16bits]***
  - Uniquely identifies PDU for a particular sender/receiver
  - Needed for re-assembly and error reporting
- ***flags[3bits]***
  - First: Is this data fragmented?
  - Second: Are we allowed to fragment the data?
  - Third: not used
  - Default, NF, MF flags as they are called
- ***fragment offset[13 bits]***:
  - Used to identify the sequence of fragments in the frame
  - It generally indicates a number of data bytes preceding or ahead of the fragment.
- ***Time to live[8bits]***: Ensures that datagram does not circulate forever, each time it is processed by router this field is decremented by 1, when field reaches 0 it is discarded.

IP Datagram Format

- ***Protocol[8bits]***: Specifies the transport layer protocol to which the datagram is to be handed, value of 6 and 17 indicates TCP and UDP respectively.
- ***Header Checksum[16bits]***: Carries checksum and aids router to detect any error in the datagram
- ***Source & Destination IP Address***: Specifies IP addresses of source and final destination.
- ***Options***: Allows IP datagram to be extended.
- ***Data***: Carries the payload

IPV6 Header Format



IPV6 Header Format

- **Version** (4 bits):
  - Specifies the IP version; in this case, the value is
- **Traffic Class** (8 bits):
  - Similar to the Type of Service (ToS) field in IPv4, it allows classification of packets by priority.
  - Helps differentiate types of traffic and can be used for quality of service (QoS) purposes.
- **Flow Label** (20 bits):
  - Used to identify and manage packets that belong to the same flow (a sequence of packets from the same source to the same destination with specific requirements).
  - This field enables faster processing of packets that are part of the same flow.
- **Payload Length** (16 bits):
  - Indicates the size of the payload in bytes, which includes any extension headers and the data.
  - Maximum value is 65,535, but larger payloads can be accommodated with the optional "Jumbo Payload" extension.

IPV6 Header Format

- **Next Header** (8 bits):
  - Specifies the type of the next header that follows the IPv6 header, which can be an extension header (such as a hop-by-hop option) or a higher-layer protocol (e.g., TCP, UDP).
  - Similar to the Protocol field in IPv4, this field enables protocol chaining and extends the capabilities of IPv6 headers.
- **Hop Limit** (8 bits):
  - Specifies the maximum number of hops (routers) the packet can pass through before being discarded.
  - Similar to the Time to Live (TTL) field in IPv4, but renamed to clarify its purpose as a hop counter.
- **Source Address** (128 bits):
  - Contains the IPv6 address of the sender (source) of the packet.
- **Destination Address** (128 bits):
  - Contains the IPv6 address of the recipient (destination) of the packet.

IP Datagram Fragmentation

- Datagrams received by datalink layer need to be encapsulated to enable transmission from one link to other.
- There is a limit as to how much a link layer frame can carry.
- Maximum amount of data a link layer frame can carry is called Maximum Transmission Unit (MTU).
- Problem arises when each link uses a different link layer protocol having different MTU

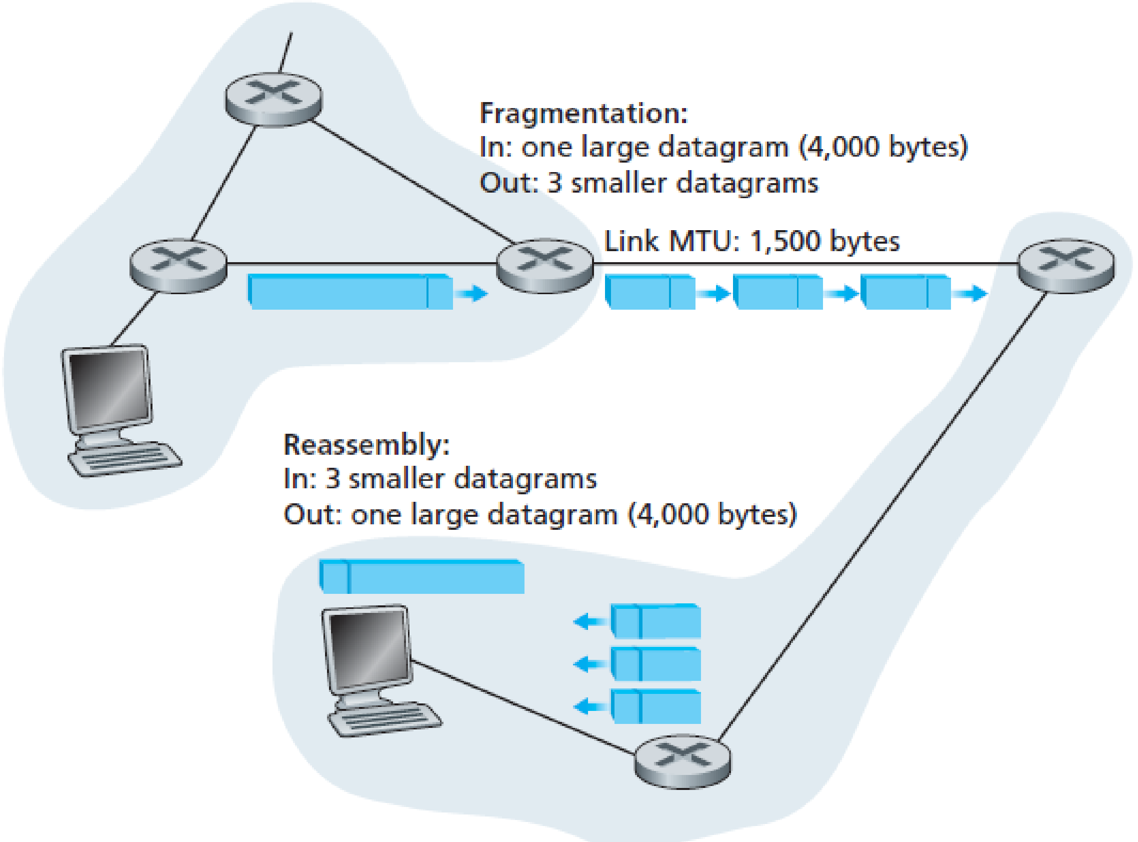
Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

MTU Sizes

IP Datagram Fragmentation

- Link layer frames are broken down into smaller size fragments to enable transmission over the link.
- Fragments are reassembled at the destination by the end systems.
- To enable reassembly of fragments, each fragment has a identification no.
- Sending host increments the identification no. of each outgoing fragment.
- The last fragment has a flag value 0 to denote end and remaining other have a flag value of 1.
- Offset denotes place where the data is to inserted.

IP Datagram Fragmentation



Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that 185 · 8 = 1,480)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= 3,980–1,480–1,480) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that 370 · 8 = 2,960)	flag = 0 (meaning this is the last fragment)

DHCP: Dynamic Host Configuration Protocol

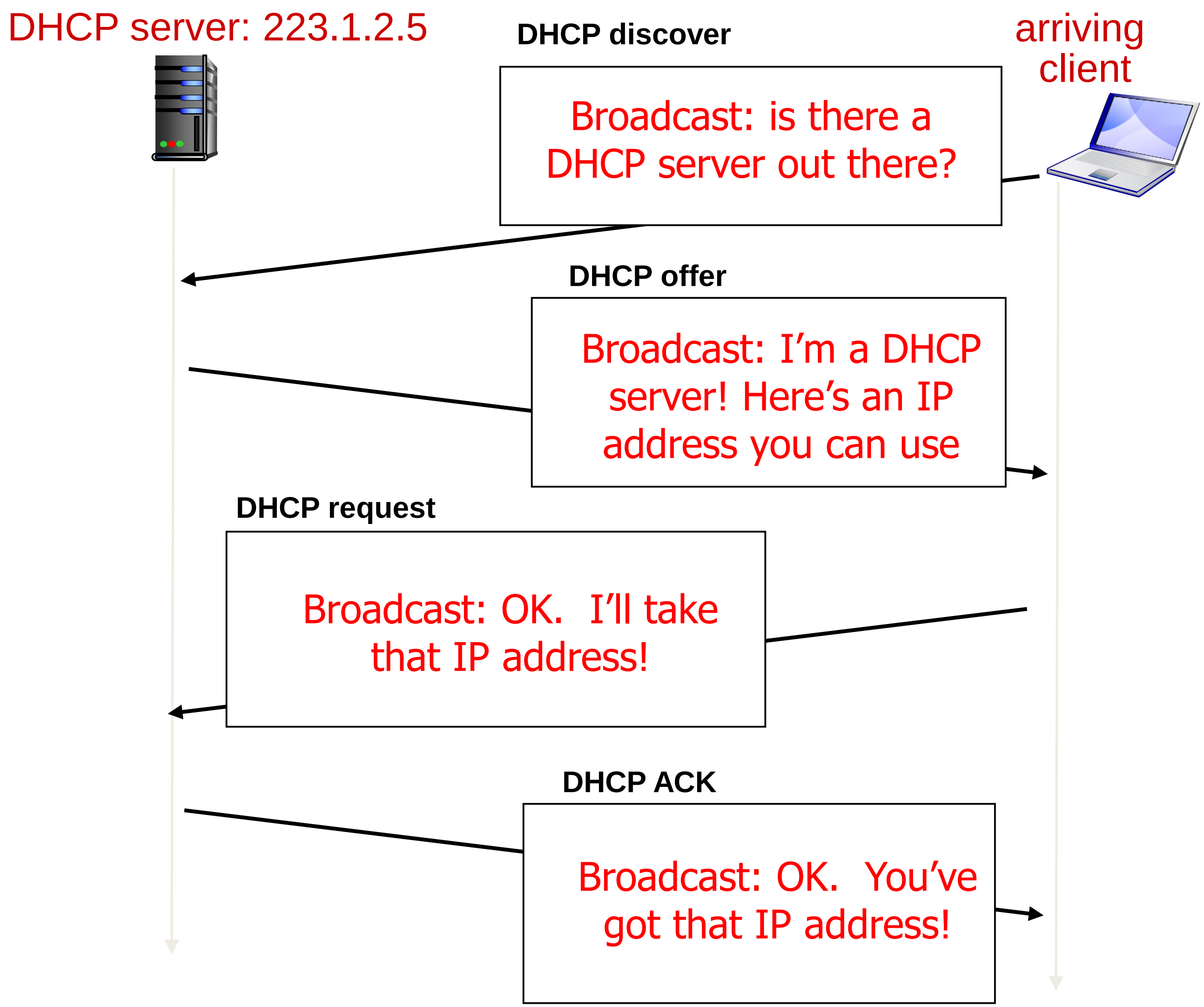
goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)

DHCP overview:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario

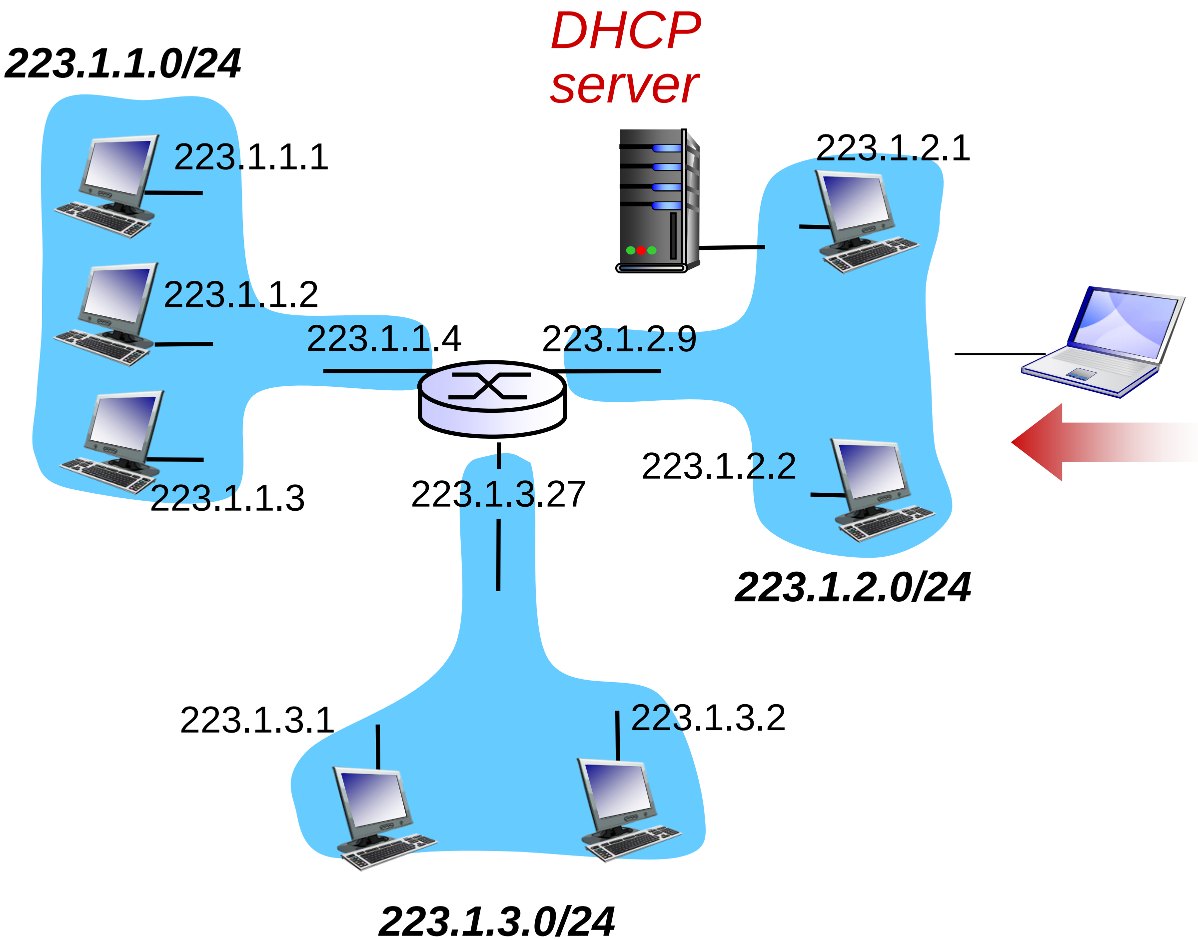


DHCP: more than IP addresses

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

DHCP client-server scenario



arriving DHCP client needs address in this network

NAT: network address translation

implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr

- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

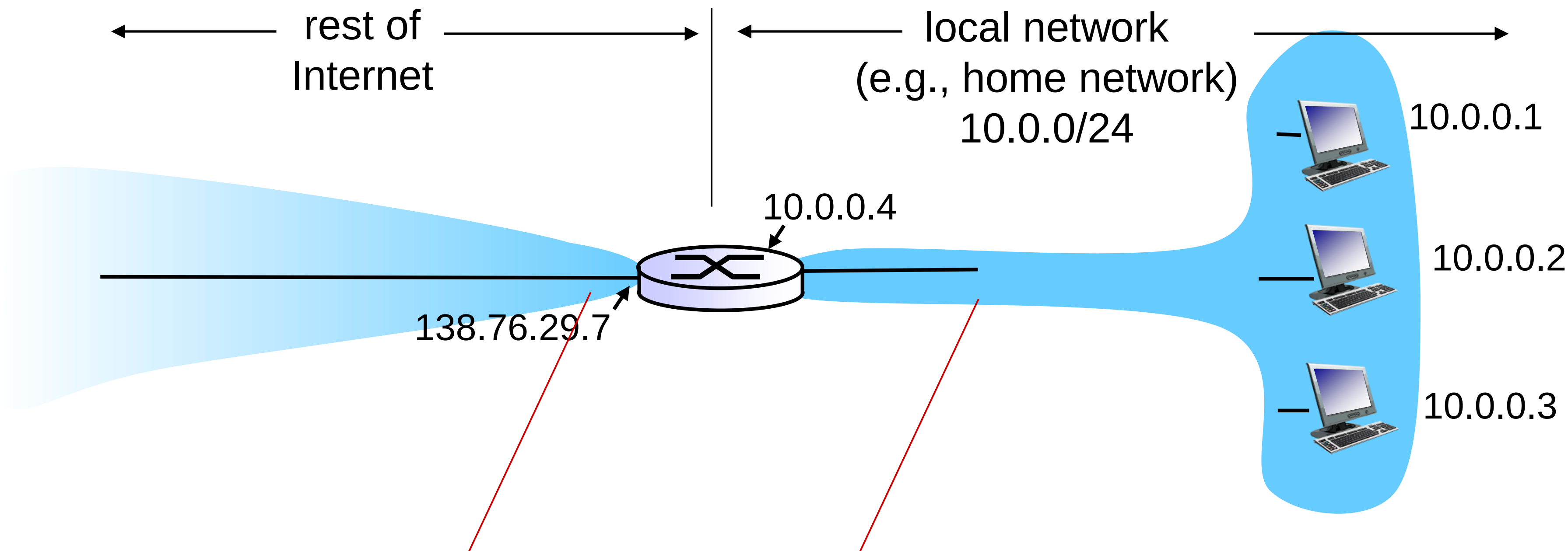
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation

motivation: local network uses just one IP address as far as outside world is concerned:

- range of addresses not needed from ISP: just one IP address for all devices
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable, visible by outside world (a security plus)

NAT: network address translation



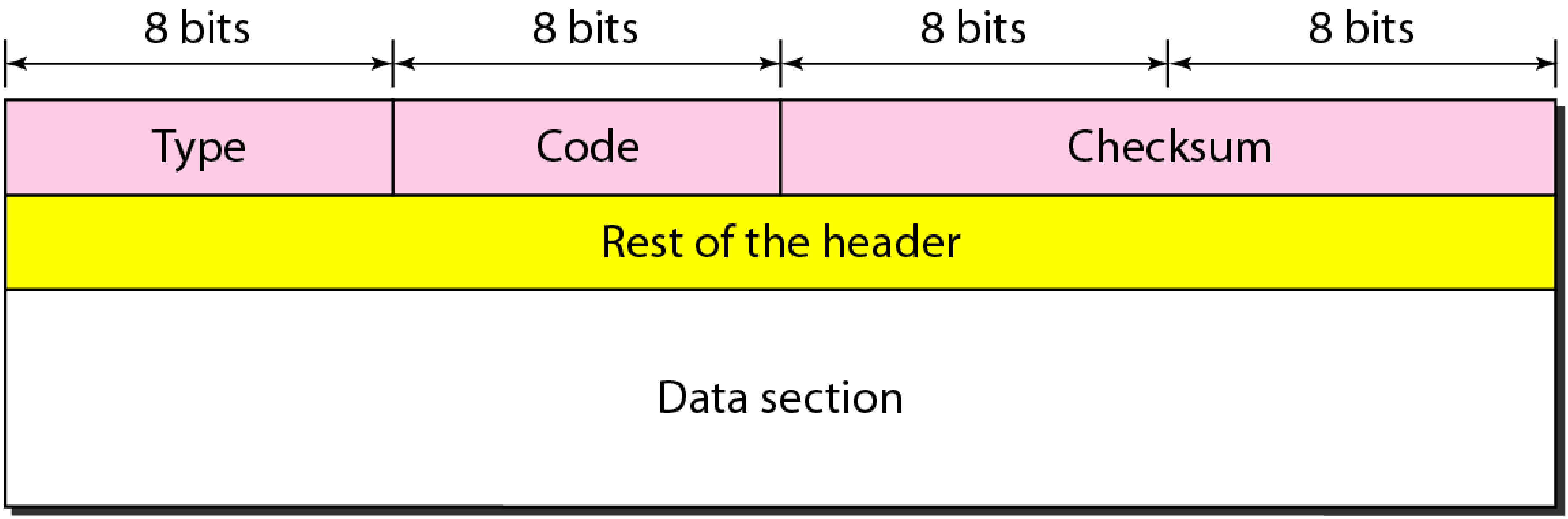
all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

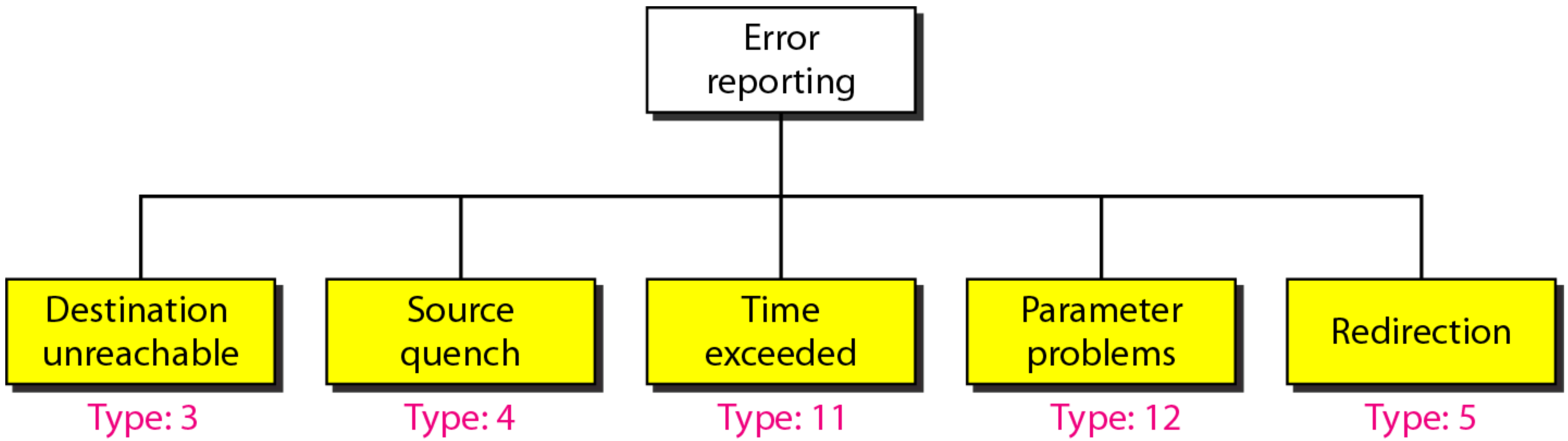
datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

ICMP

- IP protocol has no error-reporting or error-correcting mechanism
- IP protocol also lacks a mechanism for host and management queries
- Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies.
- ICMP always reports error messages to the original source.

Format of ICMP messages





Destination Unreachable Message

- ICMP Fields: Type 3
- Code:

  0 = net unreachable

  1 = host unreachable

  2 = protocol unreachable

  3 = port unreachable

  4 = fragmentation needed

  5 = source route failed

Source Quench Message

- ICMP Fields:
  - Type 4
  - Code 0
- A gateway may discard internet datagrams if it does not have the buffer space needed to queue the datagrams for output to the next network on the route to the destination network

Time Exceeded Message

- ICMP Fields:
  - Type 11
- Code
  - 0 = time to live exceeded in transit;
  - 1 = fragment reassembly time exceeded.
- Gateway processing a datagram finds the time to live field is zero it must discard the datagram
- If a host reassembling a fragmented datagram cannot complete the reassembly due to missing fragments within its time limit it discards the datagram

Parameter Problem Message

- ICMP Fields:
  - Type 12
  - Code
    - 0 = pointer indicates the error.
- The gateway or host processing a datagram finds a problem with the header parameters such that it cannot complete processing the datagram

Redirect Message

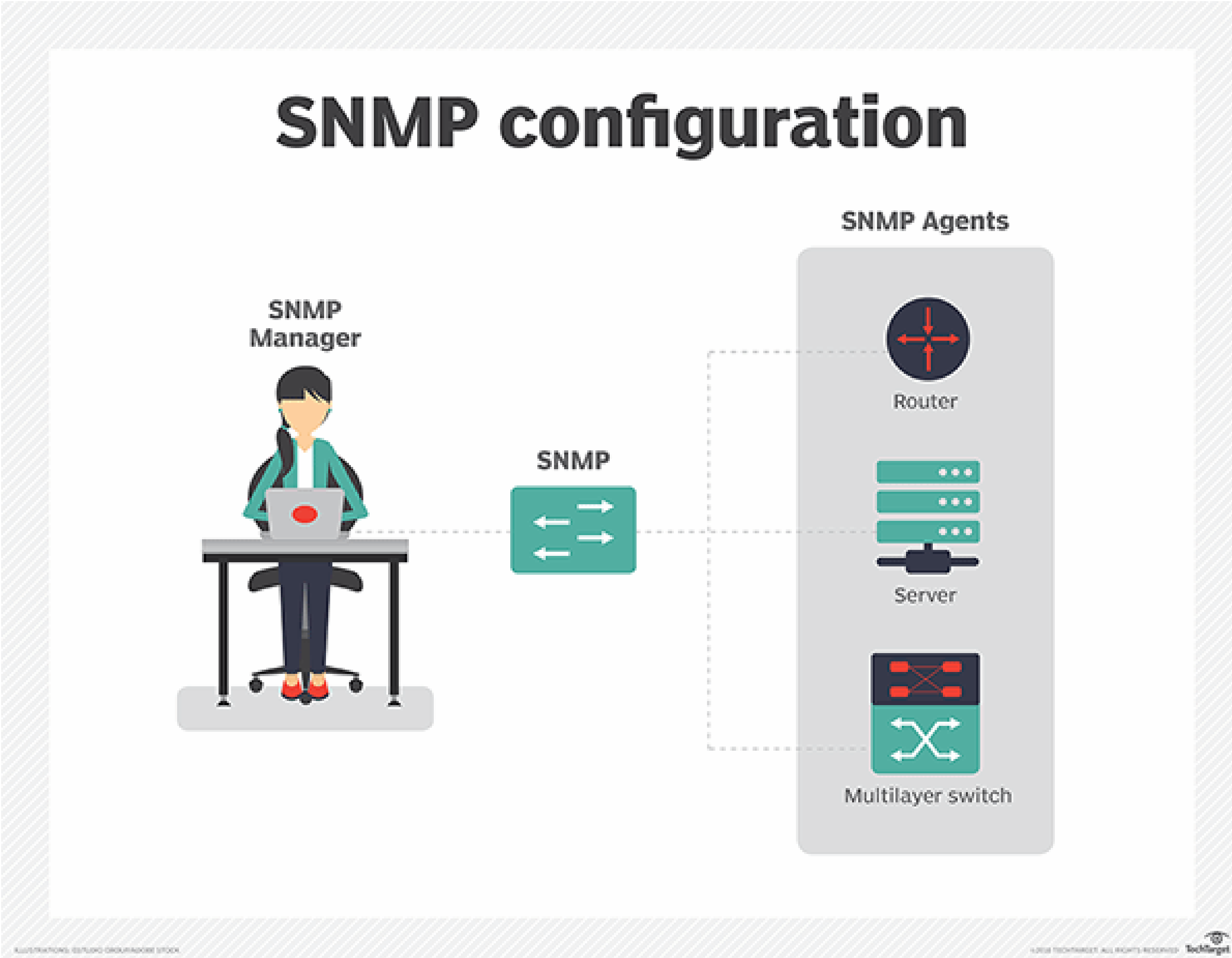
- ICMP Fields:
  - Type 5
  - Code
    - 0 = Redirect datagrams for the Network.
    - 1 = Redirect datagrams for the Host.
    - 2 = Redirect datagrams for the Type of Service and Network.
    - 3 = Redirect datagrams for the Type of Service and Host.

Simple Network Management Protocol (SNMP)

- SNMP is a framework for managing devices in an internet using the TCP/IP protocol suite.
- SNMP provides a common language for network devices to relay management information in a LAN or wide area network WAN.
- SNMP is supported on an extensive range of hardware: Routers, Switches and Wireless Access Points.

Simple Network Management Protocol (SNMP)

- SNMP software agents on these devices and services communicate with a network management system also called an SNMP manager.
- Agents relay status information and configuration changes.
- Using SNMP with an NMS enables a network administrator to manage and monitor network devices from a single interface


SNMP configuration
SNMP Agents
SNMP
Manager
SNMP
Router
Server
Multilayer switch

SNMP Components

- SNMP
  - Agent: software runs on the hardware or service being monitored.
  - collects data about disk space, bandwidth use and other important network performance metrics.
  - When queried by SNMP manager sends the requested information.
- SNMP-managed network nodes
  - These are the network devices and services upon which the agents run.

SNMP Components

- SNMP Manager
  - Network management system (NMS) is a software platform that functions as a centralized console to which agents feed information.
  - NMS actively requests agents to send updates at regular intervals.
- Management information base (MIB)
  - This database is a text file (.mib) that itemizes and describes all objects on a particular device that can be queried or controlled using SNMP

Remote Network Monitoring(RMON)

- Remote Network Monitoring, or RMON, is an extension of the Simple Network Management Protocol (SNMP)
- Allows detailed monitoring of network statistics for Ethernet networks.
- An SNMP-manageable device such as a hub or router needs additional software installed on it only to provide RMON functionality and turn it into a probe.