CPE 426 Computer Networks

Chapter 4:

Text Chapter 21: IP Address

Text Chapter 22: IP Datagram

TOPICS

- **Chapter 21: IP Address**
  - Addressing Scheme/Prefix&Suffix: 21.1-21.8
  - Subnetting and Mask: 21.9-21.13
  - Special IP Address: 21.14
  - Router IP Addressing/Multi-Homed Host: 21.17-21.18
- **BREAK**
- **Chapter 22: IP Datagram**
  - Header Format: 22.1-22.5
  - Datagram Forwarding: 22.6-22.10
  - Encapsulation: 22.11-22.12
  - MTU and Fragmentation: 22.13-22.17

สรุปบทที่ 20

- **การเชื่อมต่อระหว่าง Network เข้าด้วยกัน จะใช้อุปกรณ์ชื่อ Router ซึ่งทำงานใน Layer 3**
  - แต่ละ Interface ของ Router จะเชื่อมต่อกับแต่ละ Network และต้อง Run Network Protocol ตาม Network ที่มันไปเชื่อมต่ออยู่
- **เราจะต้องมี Global Address ที่ไปขึ้นกับ Address ของแต่ละ Network ที่จะบ่งบอกหมายเลข Host ในการส่งข้อมูลระหว่างสอง Host ที่อยู่คนละ Network**
  - Global Address นี้ต้องสามารถ Mapping กับ Network Address ที่กำหนด Host ในแต่ละ Network ได้ (ต้องมีขบวนการแปลง Address)
  - อย่างไรก็ตาม ในการส่งข้อมูลภายใน Network ยังคงอ้างอิงกับ Address ตาม Protocol ของ Network นั้นๆ ซึ่ง Address นี้จะมี Scope เพียงเฉพาะใน Network หนึ่งๆ เท่านั้น
- **เมื่อมีระบบ Address ใหม่ จำเป็นต้องมี Internet Protocol ที่จะกำหนด Global Address นี้**
  - Protocol นี้ต้อง Run ที่ทั้ง Host และ Router

สรุปบทที่ 20

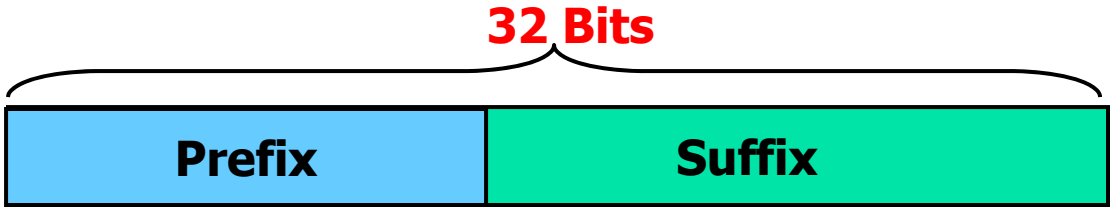
- **Internetworking Protocol ที่นิยมที่สุดคือ IP หรือ Internet Protocol (ใช้ The Internet)**
- **Global Address ที่กำหนดหมายเลข Host คือ IP Address**
- **IP อยู่ใน Layer 3 และต้อง Run ที่ทั้ง Host และ Router**
  - การส่งข้อมูลให้ถึง Host ที่หมาย จะกำหนดด้วย IP Address ปลายทาง แม้ว่าจะมีการส่งข้อมูลผ่านหลาย NW เราไม่จำเป็นต้องรู้ NW Address ของแต่ละ Network ที่ผ่าน เพราะจะมีขบวนการ Mapping โดยอัตโนมัติ (ARP)
  - เรามองเห็นแค่ IP Layer ส่วน Layer ล่างจะ Transparent กับ User เสมือนเราเชื่อมต่อกับ Internet อย่างเดียว
    - เป็น Virtual Network
- **ในการส่งข้อมูลจะแบ่งเป็นสอง Step**
  - 1. ส่ง Packet ไปให้ถึง NW ปลายทางก่อน โดย Router จะดูจากส่วน Network ID ของ IP Address
  - 2. เมื่อ Packet ไปถึง Network ปลายทาง จะเป็นหน้าที่ของ Network Protocol นั้นๆทำการส่งไปให้ถึง Host ใน Network นั้น ในกรณีนี้ ส่วนของ Host ID ใน IP Address จะถูก Map เข้ากับระบบ Address ของ Network นั้น (ใน LAN จะ MAP กับ MAC Address, โดยใช้ ARP)
    - Network Protocol ใน Network นั้นๆจะรับผิดชอบส่ง Packet จาก Router ไปยัง Host ปลายทาง

Chapter 21: IP Addressing

- **IP Address เป็นตัวกำหนด End System (Host) ตลอดทั้งเครือข่าย Internet**
  - ดังนั้นมันจะต้องเป็น Global Address ขณะที่ MAC Address จะหมดอายุเมื่อออกจาก LAN
    - เป็นแค่ Local Address ใน LAN
  - เครื่องสองเครื่องในเครือข่าย จะมีหมายเลขเดียวกันไม่ได้
    - เครื่องที่ต่อใน LAN ออก Internet จะมีทั้ง MAC Address และ IP Address ที่ Match กัน
      - Protocol ARP จะกล่าวในบทที่ 23
- **ปัจจุบันใช้ตามมาตรฐานของ IPv4**
  - Address ขนาด 32 บิต
    - เรียก IP Address, Internet Address หรือ Internet Protocol Address
  - กำหนดหมายเลขเครื่องได้โดยไม่เกี่ยวข้องกับ MAC Address

Ch. 21: 21.4 IP Address Hierarchy

- **แต่ละ IP Address ขนาด 32 บิต จะถูกแบ่งออกเป็นสองส่วน**
  - ส่วนต้น เรียก Prefix เป็นตัวกำหนดหมายเลข Network (Network ID)
  - ส่วนที่เหลือ เรียก Suffix เป็นตัวกำหนดหมายเลข Host ใน Network นั้นๆ
    - หมายเลข Host ใน Network เดียวกันจะซ้ำกันไม่ได้
    - หมายเลข Host ที่อยู่คนละ Network สามารถซ้ำกันได้ เพราะส่วน Prefix (Net ID) นั้นไม่เหมือนกัน



Ch. 21: 21.5 Original Classful IP Addressing

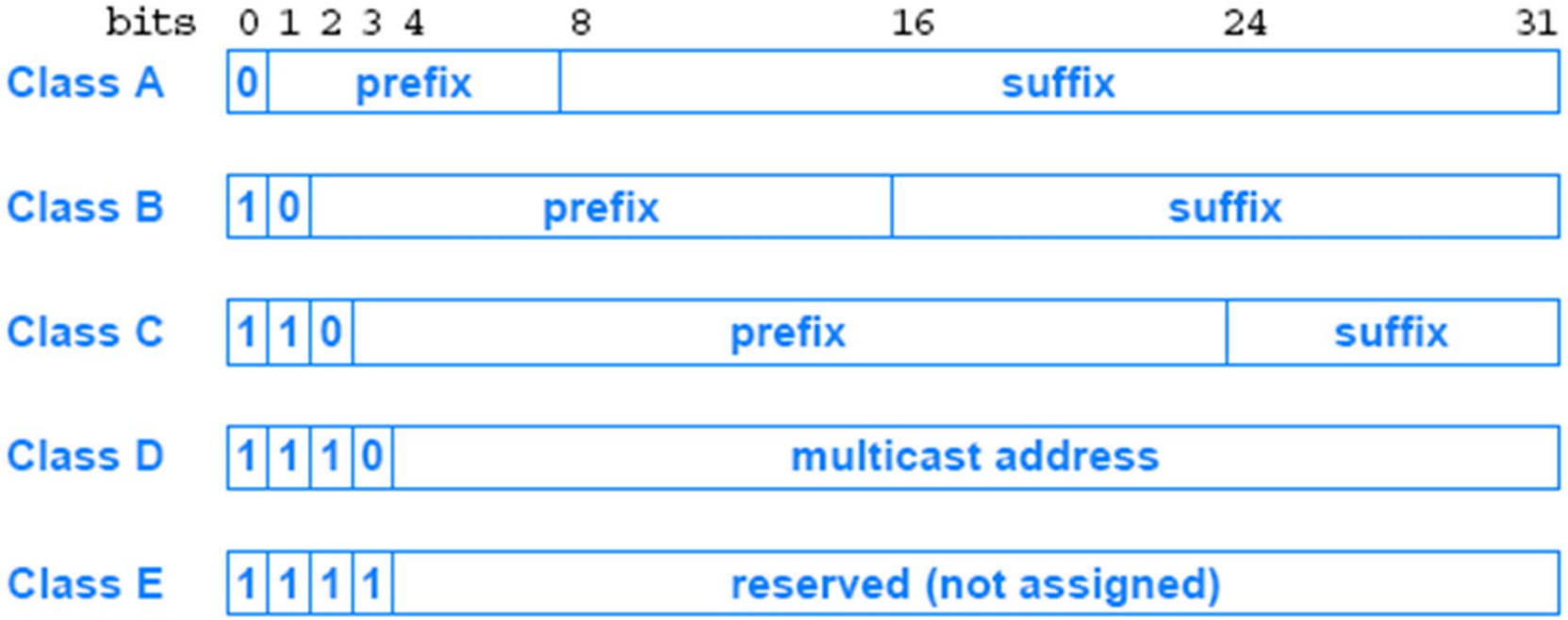


Figure 21.1 The five classes of IP addresses in the original classful scheme.

Ch. 21: 21.6 Dotted Decimal Notation

- **ใน 32 บิต IP Address จะถูกแบ่งเป็นสี่ส่วน ส่วนละ 8 บิต**
  - แต่ละส่วนจะเขียนเป็นเลขฐาน 10 มีค่าได้ระหว่าง 0 – 255
  - แต่ละส่วนจะเขียนต่อกัน ขั้นด้วย “จุด”
- **เรียก Dotted Decimal Notation**
  - Address ต่ำสุดคือ 0.0.0.0
  - Address สูงสุดคือ 255.255.255.255
  - แต่ละ Class สามารถสังเกตุได้จาก Octet แรกของ IP Address

Ch. 21: 21.7 Division of Address Space

- สังเกตุว่า แม้ว่า Class A จะมีแค่ 128 Network แต่มันประกอบด้วยครึ่งหนึ่งของ Address Space ทั้งหมด

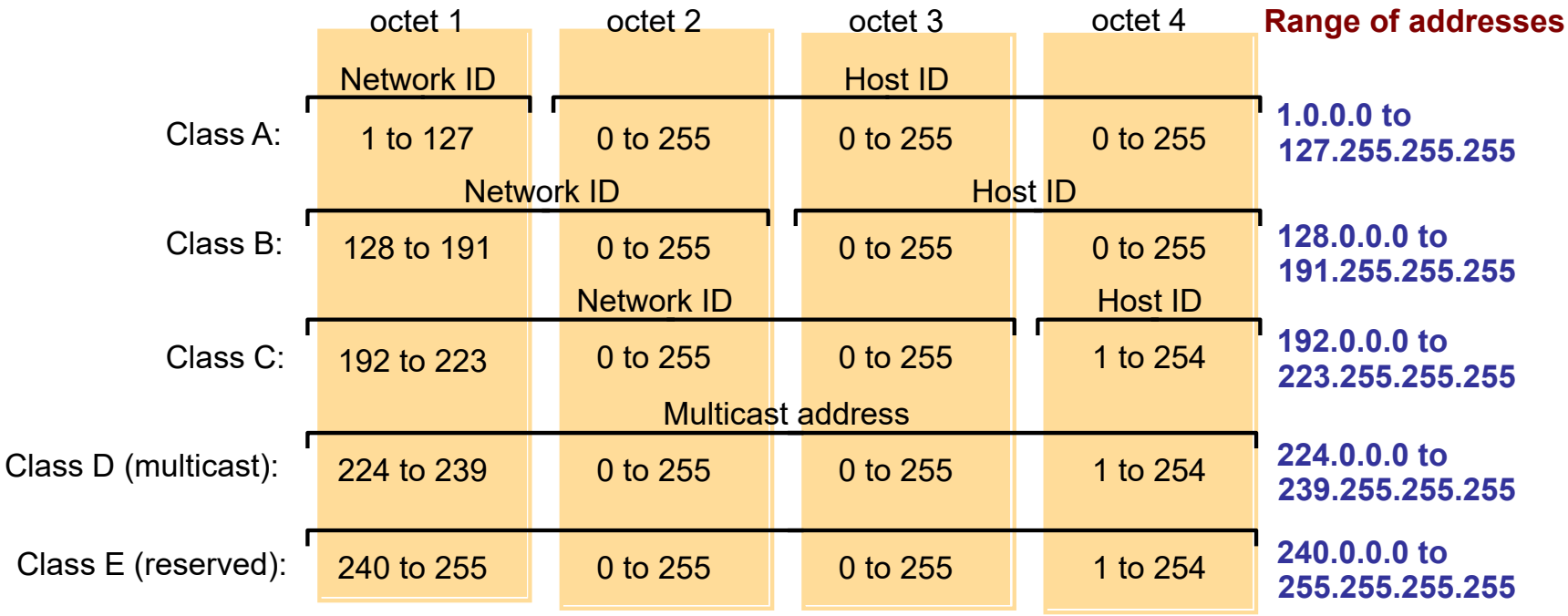
Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Figure 21.3 The number of networks and hosts per network in each of the original three primary IP address classes.

Ch. 21: 21.8 Authority for Address

- **องค์กรที่ดูแลจัดการเรื่อง IP Address คือ ICANN**
  - Internet Corporation for Assigned Names and Numbers
- **ปกติ ICANN จะกำหนดให้ Registrar เป็นผู้จัดสรรในแต่ละภูมิภาคอีกทีหนึ่ง**
- **Registrar จะจัดสรร Block ของ IP Address ให้แก่ ISP แต่ละราย**
  - ผู้ใช้งานจะได้รับ IP Address จาก ISP อีกที

Summary of IP Classful

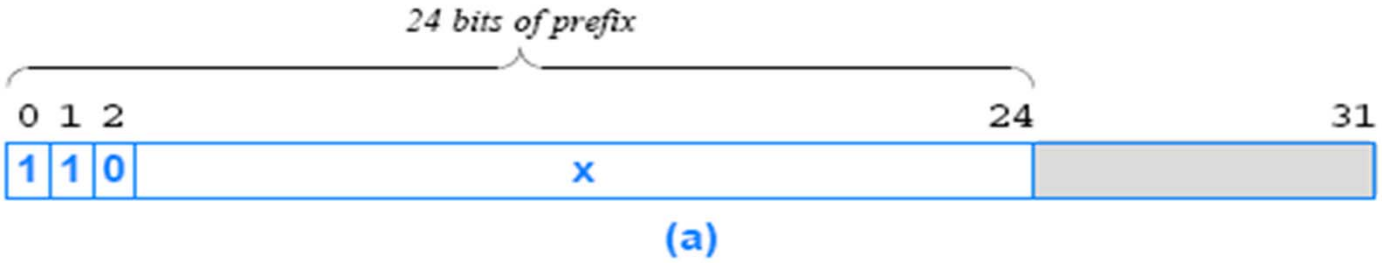


Ch. 21: 21.9 Subnet and Classless Addressing

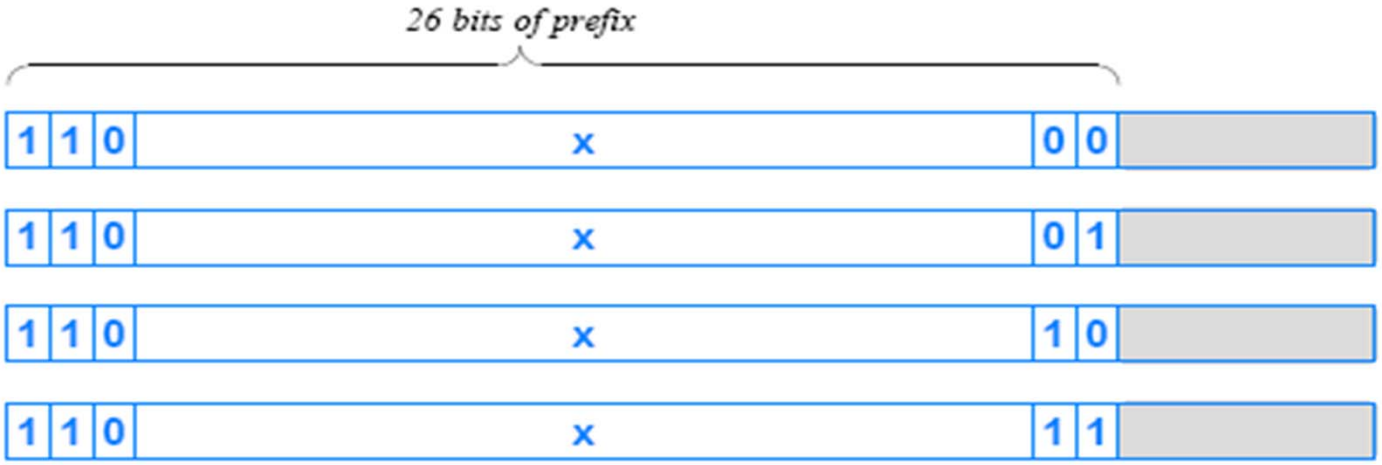
- **การแบ่ง Class ทำให้การใช้งาน IP Address ขาดประสิทธิภาพ**
  - เรามีคอมพิวเตอร์ 15 เครื่อง ดังนั้นต้องใช้ Class C ซึ่งรองรับได้ 256 (-2) เครื่อง แต่ใช้จริงแค่ 15 ที่เหลือ คนอื่นใช้ไม่ได้ เพราะเป็นคนละ Network กับเรา
- **ปัจจุบัน IP Class A และ B ได้ถูกใช้งานหมดแล้ว**
- **เหลือ Class C จำนวนเล็กน้อย และขอได้ยากมาก**
- **เพื่อแก้ปัญหา IP Address ไม่พอใช้ มีสองวิธีที่คิดขึ้นและคล้ายกัน คือ**
  - Subnet Addressing : การแบ่ง Network เป็น NW ย่อย
  - Classless Addressing : การแบ่ง NW โดยกำหนด Prefix เอง
  - กล่าวคือ ไม่มีการกำหนด Class แต่สามารถกำหนดบิตของ Prefix และ Suffix ได้ตามต้องการ

Ch. 21: 21.9 Subnet and Classless Addressing

- ตัวอย่าง Class C เดิม (24 Bit Prefix) แบ่งเป็นสี่ส่วน โดยเพิ่ม Prefix อีก 2 บิต (Suffix ถูกลดลง 2 บิต เหลือ 6 บิต)



(a)



(b)

Ch. 21: 21.10 Address Masks

- **การที่เรากำหนด Prefix เองทำให้ IP Address ไม่มี Class อีกต่อไป**
  - เรียก Classless
- **ในการนี้ เราจะต้องบ่งบอกเองว่าส่วนไหนเป็น Prefix และส่วนใหนเป็น Suffix**
- **กำหนดจาก ค่า 32 Bit เช่นกันเรียก Address Mask (เมื่อก่อนเรียก Subnet Mask)**
  - ส่วนต้น ที่กำหนด Prefix ให้เป็นบิต '1'
    - ต้องเขียนติดต่อกัน
  - ส่วนที่เหลือเป็น Bit '0' กำหนด Suffix เขียนติดกัน
- **การหา Network ID(Prefix) ทำได้โดยทำ Bitwise AND ระหว่าง IP Address กับ Mask**
  - N == (D & M)

Ch. 21: 21.10 Address Masks

Example:

- **Consider the following 32-bit network prefix:**

  10000000 00001010 00000000 00000000 = 128.10.0.0

- **Consider a 32-bit mask:**

  11111111 11111111 00000000 00000000 = 255.255.0.0

- **Consider a 32-bit destination address, which has a**

  10000000 00001010 00000010 00000011 = 128.10.2.3

- **A logical and between the destination address and the address mask extracts the high-order 16-bits**

  10000000 00001010 00000000 00000000 = 128.10.0.0

Ch. 21: 21.11 CIDR Notations

- ปกติ Address Mask เขียนในลักษณะ Dotted Decimal Format เช่นเดียวกันกับ IP Address
  - เช่น 1111 1111 1111 1111 1111 1111 1100 0000 จะเขียนเป็น 255.255.255.192
- เพื่อให้ง่ายในการอ่าน แทนที่เราจะเขียน IP Address คู่กับ Address Mask เราเขียน IP Address ตามด้วยจำนวนบิตที่กำหนด Suffix ในรูป
  - ddd.ddd.ddd.ddd / m
  - เช่น 192.5.48.69 / 26 มีความหมายเดียวกับ IP Address 192.5.48.69 และ Mask 255.255.255.192
- เรียก CIDR Notation
  - Classless Inter-Domain Routing

CIDR

Length (CIDR)	Address Mask	Notes
/0	0 . 0 . 0 . 0	All 0s (equivalent to no mask)
/1	128 . 0 . 0 . 0	
/2	192 . 0 . 0 . 0	
/3	224 . 0 . 0 . 0	
/4	240 . 0 . 0 . 0	
/5	248 . 0 . 0 . 0	
/6	252 . 0 . 0 . 0	
/7	254 . 0 . 0 . 0	
/8	255 . 0 . 0 . 0	Original Class A mask
/9	255 . 128 . 0 . 0	
/10	255 . 192 . 0 . 0	
/11	255 . 224 . 0 . 0	
/12	255 . 240 . 0 . 0	
/13	255 . 248 . 0 . 0	
/14	255 . 252 . 0 . 0	
/15	255 . 254 . 0 . 0	
/16	255 . 255 . 0 . 0	Original Class B mask
/17	255 . 255 . 128 . 0	
/18	255 . 255 . 192 . 0	
/19	255 . 255 . 224 . 0	
/20	255 . 255 . 240 . 0	
/21	255 . 255 . 248 . 0	
/22	255 . 255 . 252 . 0	
/23	255 . 255 . 254 . 0	
/24	255 . 255 . 255 . 0	Original Class C mask
/25	255 . 255 . 255 . 128	
/26	255 . 255 . 255 . 192	
/27	255 . 255 . 255 . 224	
/28	255 . 255 . 255 . 240	
/29	255 . 255 . 255 . 248	
/30	255 . 255 . 255 . 252	
/31	255 . 255 . 255 . 254	
/32	255 . 255 . 255 . 255	All 1s (host specific mask)

Ch. 21: 21.12 CIDR Example

- สมมุติ ISP มี Block ของ IP Address
  - 128.211.0.0 / 16
- ถ้า ISP มีลูกค้าสองราย แต่ละรายต้องการ 12 และ 9 IP Address ซึ่ง ISP สามารถกำหนด Prefix ให้แก่ลูกค้าทั้งสองดังนี้
  - 128.211.0.16 / 28
  - 128.211.0.32 /28
- แม้ว่าลูกค้าทั้งสอง จะใช้ขนาดของ Mask เท่ากัน แต่ Prefix จะต่างกันคือ
  - 10000000 11010011 00000000 0001 0000
  - 10000000 11010011 00000000 0010 0000

Ch. 21: 21.13 CIDR Host Address

- **ในแต่ละละ Block ของ CIDR เราสามารถกำหนด Host Address จากส่วน Suffix**
  - จำนวน Host ในแต่ละ Block เท่ากับ $2^{\text{suffix bit}} - 2$
  - เหตุผลที่ต้องลบสอง คือ Address แรก และ Address สุดท้าย ของ Host จะไม่ใช้
    - Address แรก คือเมื่อ Bit ของ Host เป็นศูนย์ทั้งหมด เนื่องจาก IP Address ของ Host นี้จะเหมือนกับ Network Address และป้องกันปัญหาใน Berkeley Broadcast Address Form
    - Address สุดท้าย คือเมื่อ Bit ของ Host เป็นหนึ่งทั้งหมด จะถูกกันไว้เป็น Address พิเศษสำหรับ Broadcast ใน Network นั้น

Ch. 21: 21.13 CIDR Host Address

- **Example: 28 Bit Prefix**

Ch. 21: 21.14 Special IP Address

- **Network Address**
  - Address ที่ประกอบด้วย Host Bit เป็น '0' ทั้งหมดจะถูก Reserved ไว้สำหรับ Network Address
    - 128.211.0.16/28 หมายถึง Network ไม่ใช่ IP Address
- **Directed Broadcast Address**
  - Address ที่ประกอบด้วย Host Bit เป็น '1' ทั้งหมดจะถูก Reserved ไว้สำหรับส่ง Packet ไปให้กับทุก Host ใน Physical Network นั้น
    - เมื่อ Packet นี้ถูกส่ง จะส่งไป Copy เดียว ผ่าน Internet จนกระทั่งถึง Network ปลายทาง จากนั้น Packet จะถูกส่งต่อไปให้ทุกๆ Host ใน NW ปลายทาง

Ch. 21: 21.14 Special IP Address

- **Limited Broadcast Address**
  - เป็นการ Broadcast กับเครื่องที่ต่อเฉพาะกับ Link นั้น จะใช้ในตอนที่ระบบ Start และยังไม่มี IP Address เป็นของตัวเอง
  - Address คือ 255.255.255.255 (All '1')
  - IP จะ Broadcast Packet ที่ใช้ Address นี้ไปตลอดทั่ว Local Network
- **This Computer Address**
  - กำหนดโดย Address ที่เป็น '0' ทั้งหมด
  - คือ 0.0.0.0
  - ใช้ในการกำหนด Source IP Address สำหรับเครื่องที่ยังไม่รู้ IP Address ของตนเอง
    - ใช้ใน Protocol DHCP เมื่อคอมพิวเตอร์เริ่ม Boot และขอ IP Address จาก DHCP Server

Ch. 21: 21.14 Special IP Address

- **Loopback Address**
  - ใช้สำหรับทดสอบการสื่อสารของสอง Network Application บนเครื่องเดียวกัน โดยไม่ต้องผ่าน Network
  - IP Reserver Network Prefix 127/8 สำหรับใช้กับ Loopback
    - ปกติจะใช้ Host number 1: 127.0.0.1
  - Packet Loopback จะไม่ออกมานอกเครื่อง

Ch. 21: 21.15 Summary Special IP Address

Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127 / 8	any	loopback	testing

Figure 21.7 Summary of the special IP address forms.

Ch. 21: 21.16 The Berkeley Broadcast Address Form

- **ในการพัฒนา TCP/IP บน UNIX ยุคแรกๆ โดย University of California at Berkeley (BSD UNIX) ได้ใช้ Directed Broadcast Address ที่ไม่ได้เป็นมาตรฐาน**
  - กล่าวคือใช้ Host Bit '0' ทั้งหมด กำหนดเป็น Directed Broadcast Address
    - รู้จักกันในนาม 'Berkeley Broadcast'
  - ยังมีอุปกรณ์รุ่นเก่าที่ยังใช้งานอยู่ที่ใช้ Broadcast Address นี้
    - Network Manager จะต้องระวัง และเลือกใช้

Ch. 21: 21.17 Router and IP Address

- **นอกเหนือจากการกำหนด IP Address ให้แก่ Host แล้ว อุปกรณ์ Router ต้องกำหนด IP Address ด้วย**
  - Router ต่อเข้ากับหลาย Physical Network
  - แต่ละ Interface ของ Router ต้องใช้ Prefix ตาม Network ที่มาต่อ
    - แต่ละ Interface ของ Router จะใช้ Host ID (Suffix) ที่ไม่ซ้ำกับ Network ที่มาต่อนั้น

Ch. 21: 21.17 Router and IP Address

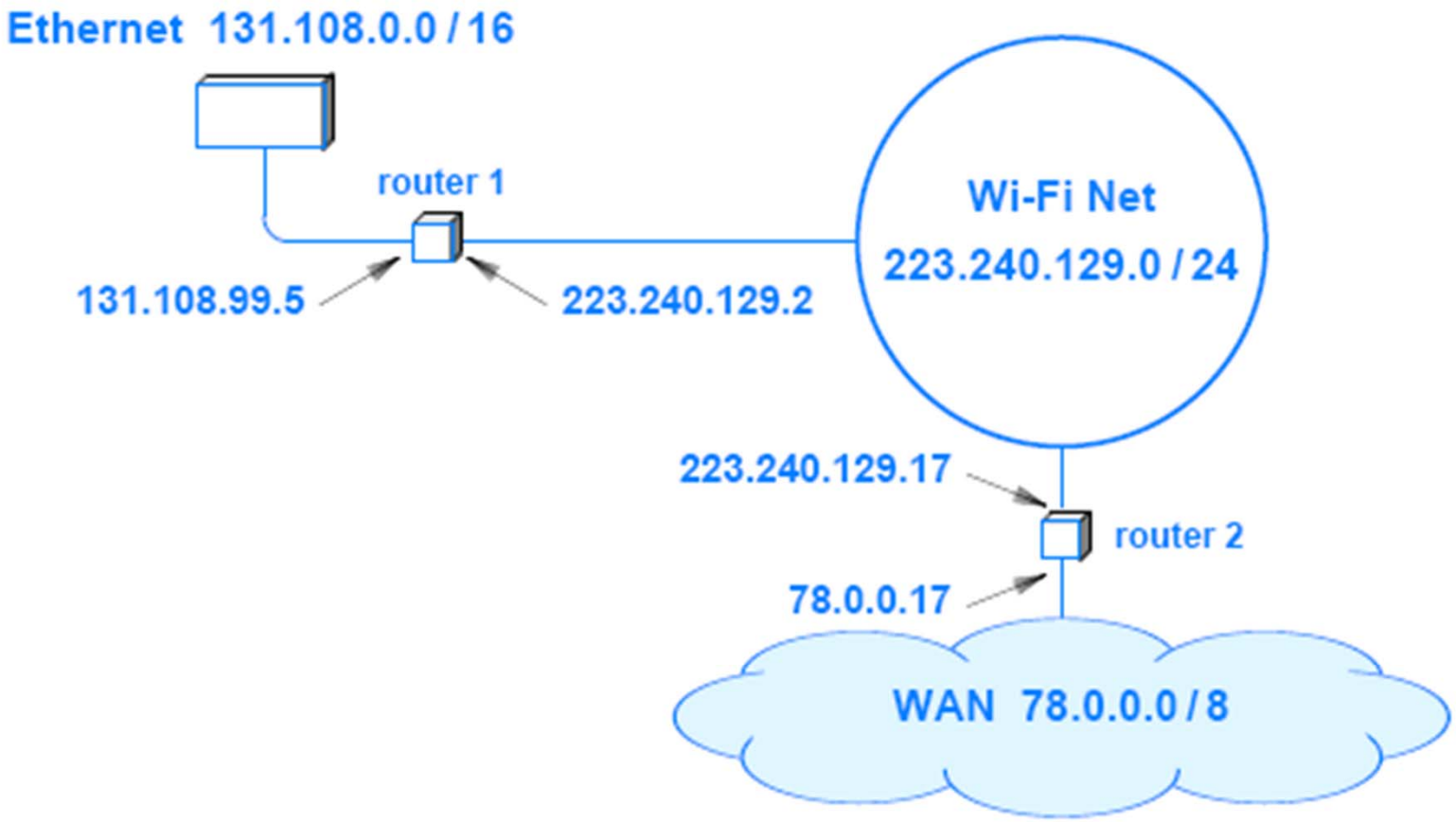
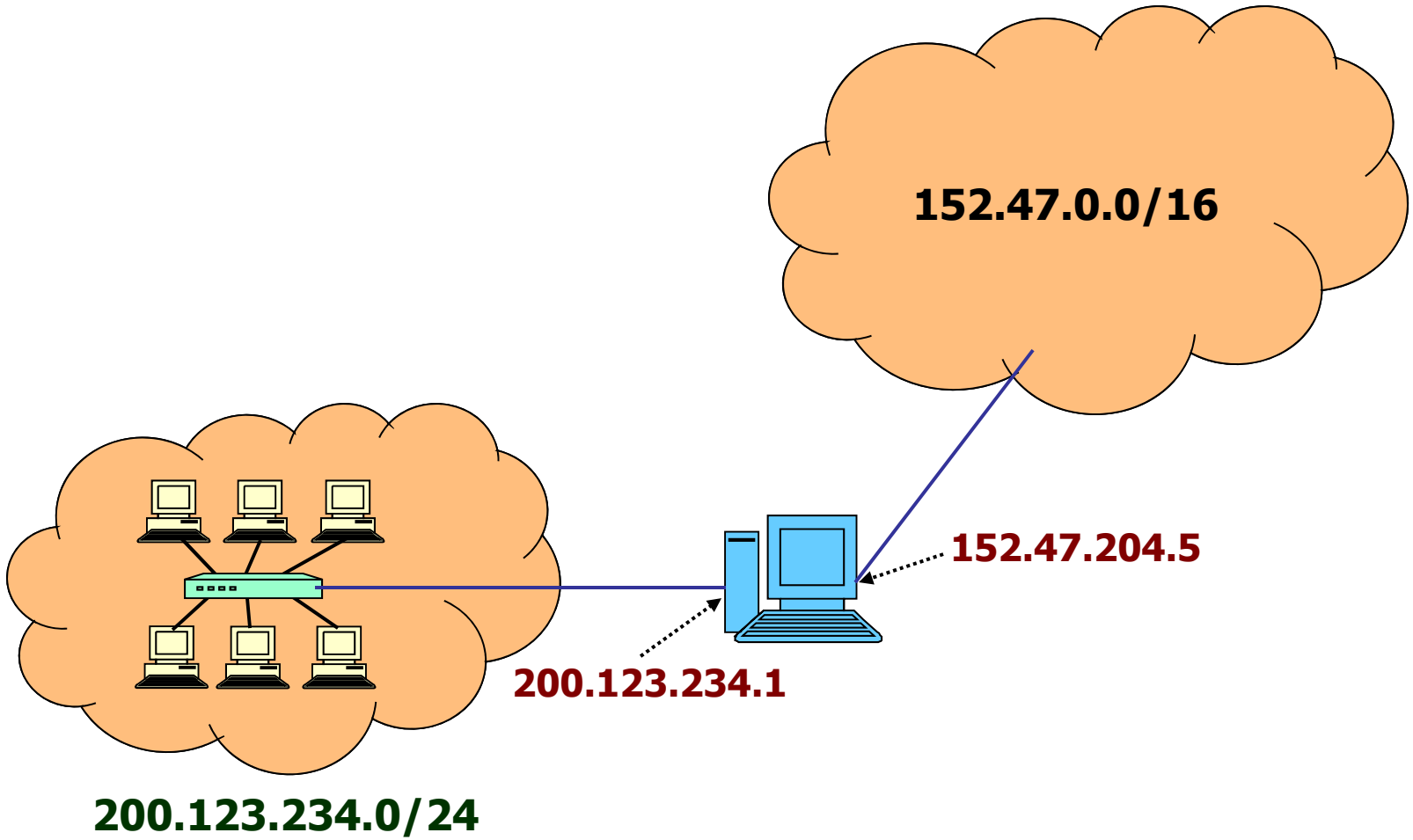


Figure 21.8 An example of IP addresses assigned to two routers.

Ch. 21: 21.18 Multi-Homed Host

- **คอมพิวเตอร์ที่มีมากกว่าหนึ่ง Network Card (LAN Card หรือ Network Card อื่น) สามารถเชื่อมต่อได้มากกว่าหนึ่ง Network และสามารถมี IP Address ได้มากกว่าหนึ่งเบอร์**
  - โดยแต่ละเบอร์มี Prefix ตาม Network ที่มาต่อ
  - เรียก Multi-Homed
  - ปกติจะใช้ในการเพิ่มความเชื่อถือได้ของอุปกรณ์ในกรณีที่ Network หนึ่ง Down
    - ยังเป็นการเพิ่มประสิทธิภาพด้วย
  - สามารถใช้งานเป็น Gateway หรือทำ Firewall หรือทำเป็น Router ได้ ถ้ามี Software ที่ถูกต้อง

Ch. 21: 21.18 Multi-Homed Host



TOPICS

- **Chapter 22: IP Datagram**
  - Header Format: 22.1-22.5
  - Datagram Forwarding: 22.6-22.10
  - Encapsulation: 22.11-22.12
  - MTU and Fragmentation: 22.13-22.17

Chapter 22: IP Datagram

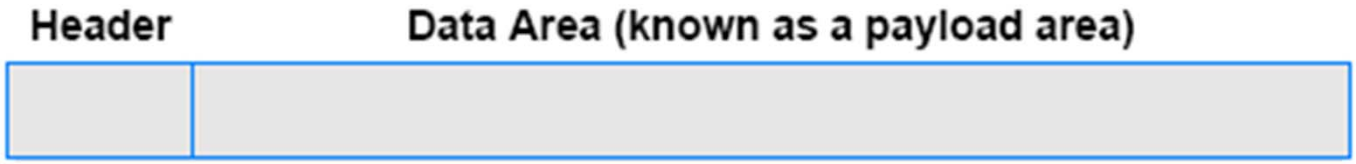
- จุดประสงค์ของ **Internetworking** เพื่อที่จะให้ **Program** ที่ **Run** อยู่บนคอมพิวเตอร์หนึ่ง สามารถส่ง **Packet** สื่อสารได้กับอีก **Program** หนึ่งที่อยู่ต่างคอมพิวเตอร์กัน
- การอกแบบ **Internetworking** ที่ดี การสื่อสารดังกล่าวจะเป็น **Transparent** และไม่ต้องคำนึงถึง **Hardware** เชื่อมต่อที่แตกต่างกัน
- **Internet** ควรจะให้ **Service** แบบใดแก่ **Application Program** เหล่านี้ : **Connectionless** หรือ **Connection-Oriented**
- ผู้ออกแบบ **TCP/IP** ตกลงเลือก **Service** ทั้งสองประเภท
  - Connectionless: User Datagram Protocol(UDP)
  - Connection Oriented: Transport Control Protocol(TCP)
- อย่างไรก็ตาม วิธีการส่ง **Packet** จะใช้แบบ **Connectionless** หรือ **Datagram** โดยถ้าผู้ใช้ต้องการความเชื่อถือในการส่งข้อมูล สามารถผนวกวิธีการของ **Connection Oriented(TCP)** เข้ากับการส่ง **Packet** ที่เป็นแบบ **Datagram (IP)** ได้

Ch22: 22.3 Virtual Packet

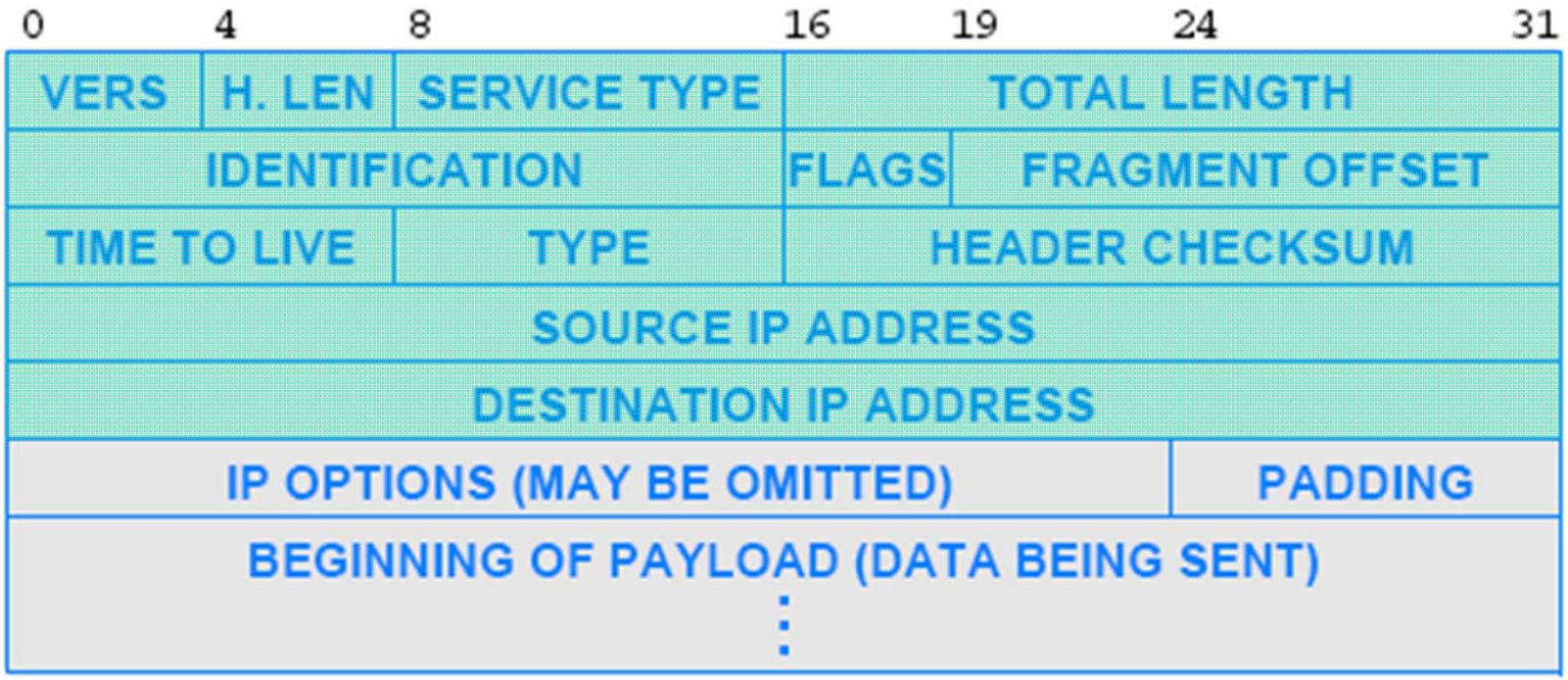
- **การส่งข้อมูลที่เป็นแบบ Datagram หรือ Connectionless ภายใน Network เป็นการปรับปรุงมาจาก Packet Switching Network**
  - โดยผู้ส่งสามารถส่งแต่ละ Packet ผ่าน Internet และแต่ละ Packet จะเดินทางโดยไม่ขึ้นต่อกัน
  - ตัวการที่สำคัญในการส่งผ่าน Packet คือ Router โดยที่ Router จะดู Address ปลายทางและตัดสินใจว่าควรจะส่ง Packet ให้ Router ตัวถัดไปตัวใด
    - Router รับผิดชอบเพียงเท่านี้ คือส่ง Packet ให้ Router ตัวถัดไป (หรือส่งไปให้ Destination ถ้าเป็น Router ตัวสุดท้ายของเส้นทาง)
  - เนื่องจาก Router อาจจะเชื่อมต่อ Network ที่ต่างกันและ Address ของแต่ละ Network ต่างกัน ซึ่ง Router ไม่สามารถที่จะเปลี่ยน Address Format ได้ทุกครั้งที่ผ่านแต่ละ Network เพราะผู้รับปลายทางอาจจะใช้ Address Format ที่ต่างออกไปอีก
    - ในการนี้ Internet จะต้องออกแบบ Packet Format ใหม่ และใช้ Address ที่เป็นหนึ่งเดียว (คือ IP Address) กล่าวคือ Internet Protocol จะต้องวางบน Network Layer ของแต่ละ Network อีกทีหนึ่ง
    - ผลลัพธ์คือ Universal Virtual Packet ที่สามารถส่งผ่าน Network อะไรก็ได้

Ch22: 22.4 IP Datagram

- ใน **TCP/IP** จะใช้คำว่า **IP Datagram** ในการอ้างถึง **Packet** ของ **Internet** ซึ่งประกอบด้วย **Payload** ที่ได้มาจาก **Layer** บน และส่วนหัวที่เรียก **IP Header**
- ขนาดของ **Payload** ไม่ได้กำหนดตายตัว ขึ้นอยู่กับ **Application** ที่ใช้
  - IPv4 สามารถมี Payload ได้ตั้งแต่ 1 Octet จนถึงสูงสุด 64K Octet(65535: รวมส่วนหัวด้วย)
  - ขนาดของ Header ปกติจะคงที่คือ 20 Octet แต่สามารถขยายได้ (ส่วนของ Option)

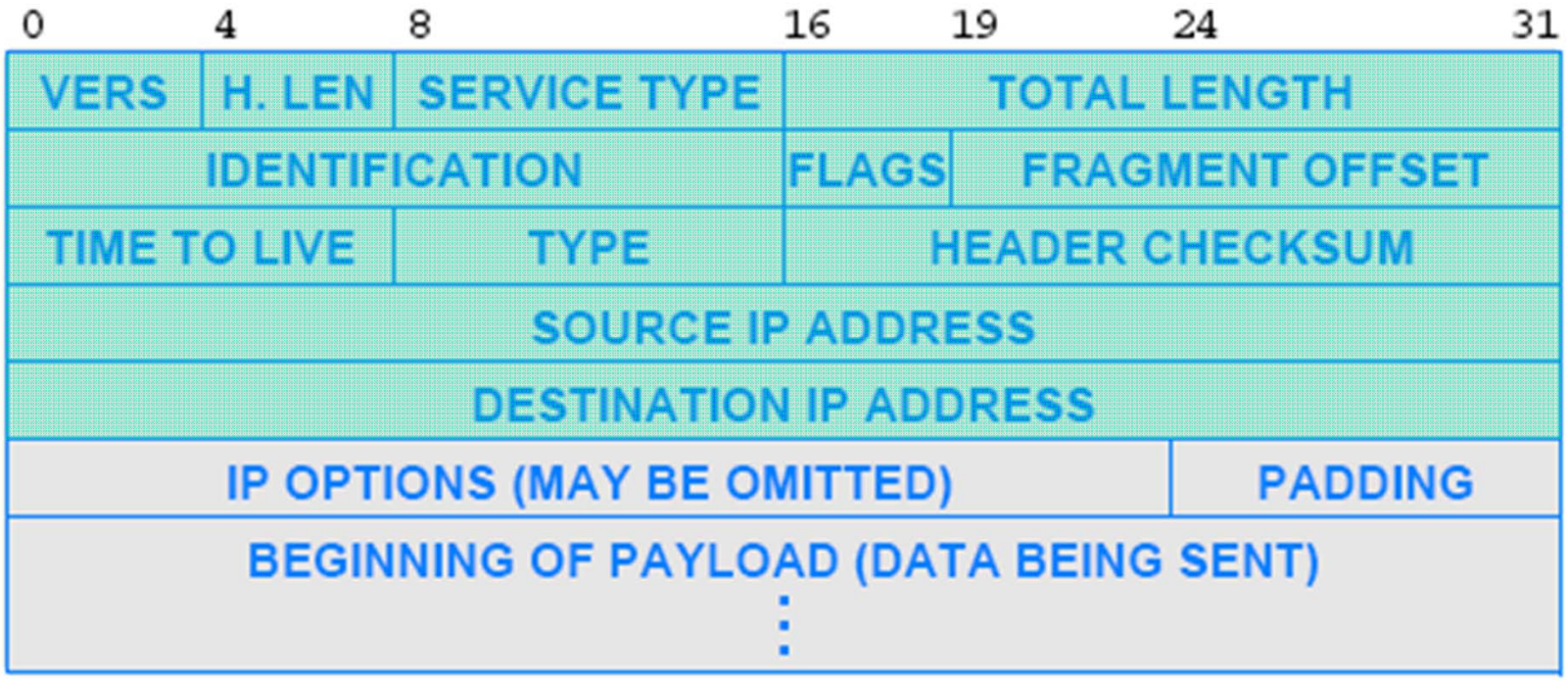


Ch22: 22.5 IP Datagram Header Format (IPv4)



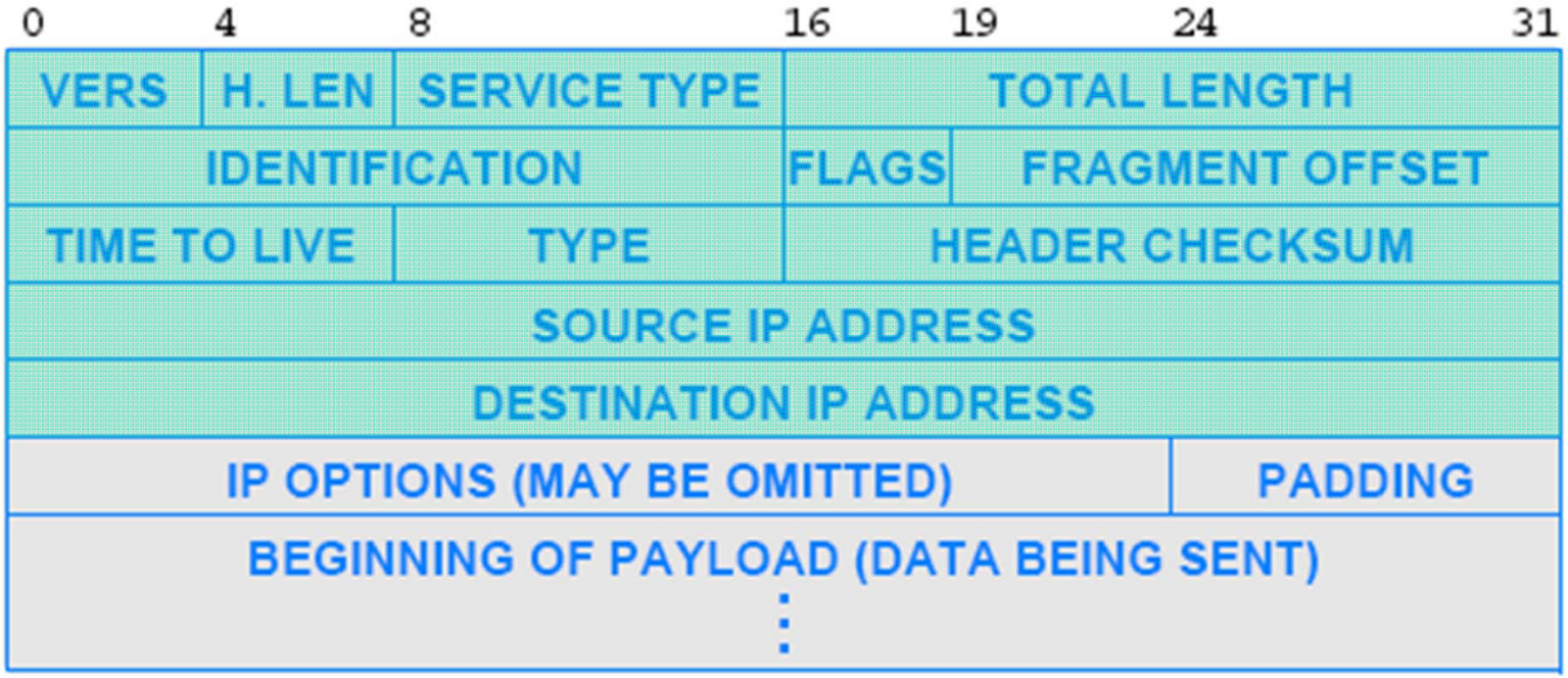
- **VERS.** กำหนด **Version** ในที่นี้คือ **0100 B**
- **H.LEN.** กำหนดความยาวของ **Header** มีหน่วยเป็น **32 Bit Word** ถ้าไม่มี **Option** ค่านี้คือ **0101 B**

Ch22: 22.5 IP Datagram Header Format (IPv4)



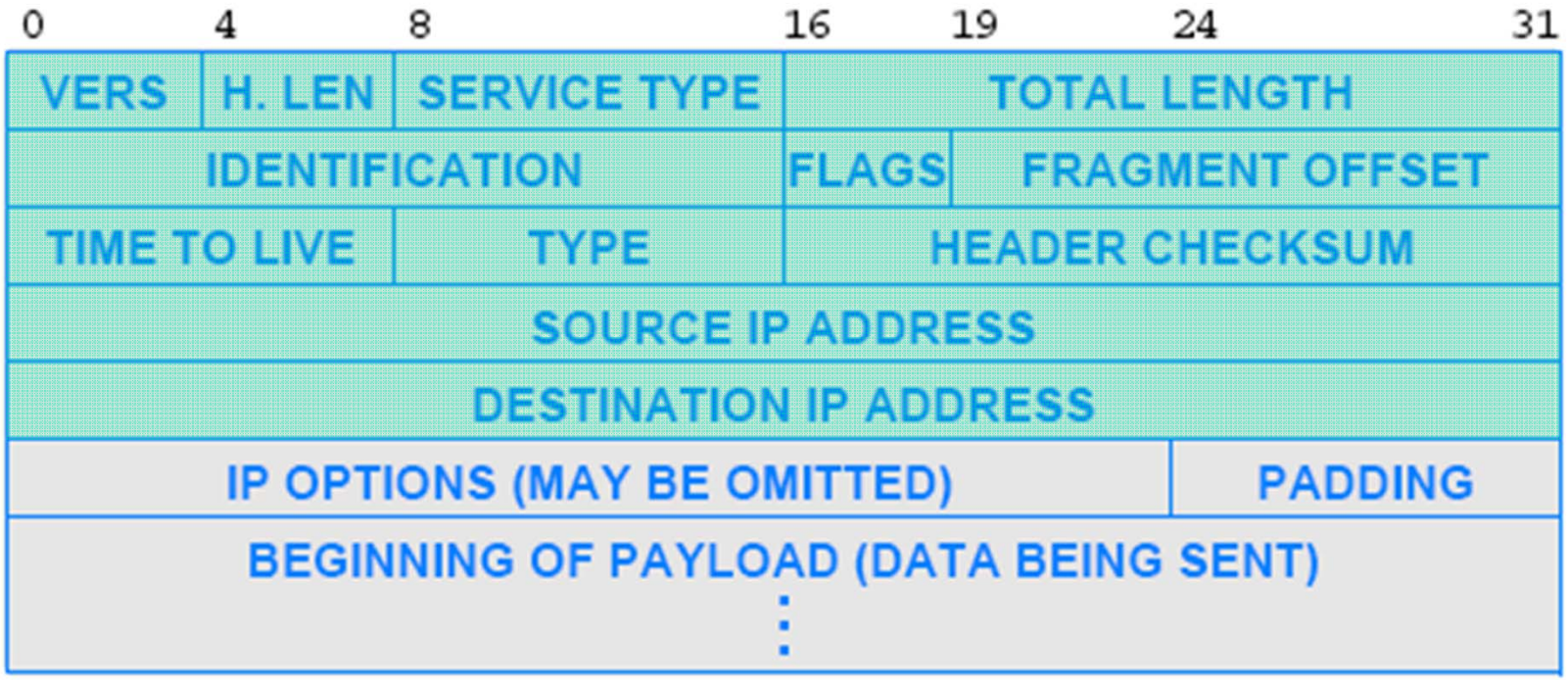
- **SERVICE TYPE.** กำหนด Class ของ Service ปกติไม่ใช้ (จะพูดในเรื่อง QoS)
- **TOTAL LENGTH.** ความยาวเป็น Byte รวมทั้งส่วนหัว

Ch22: 22.5 IP Datagram Header Format (IPv4)



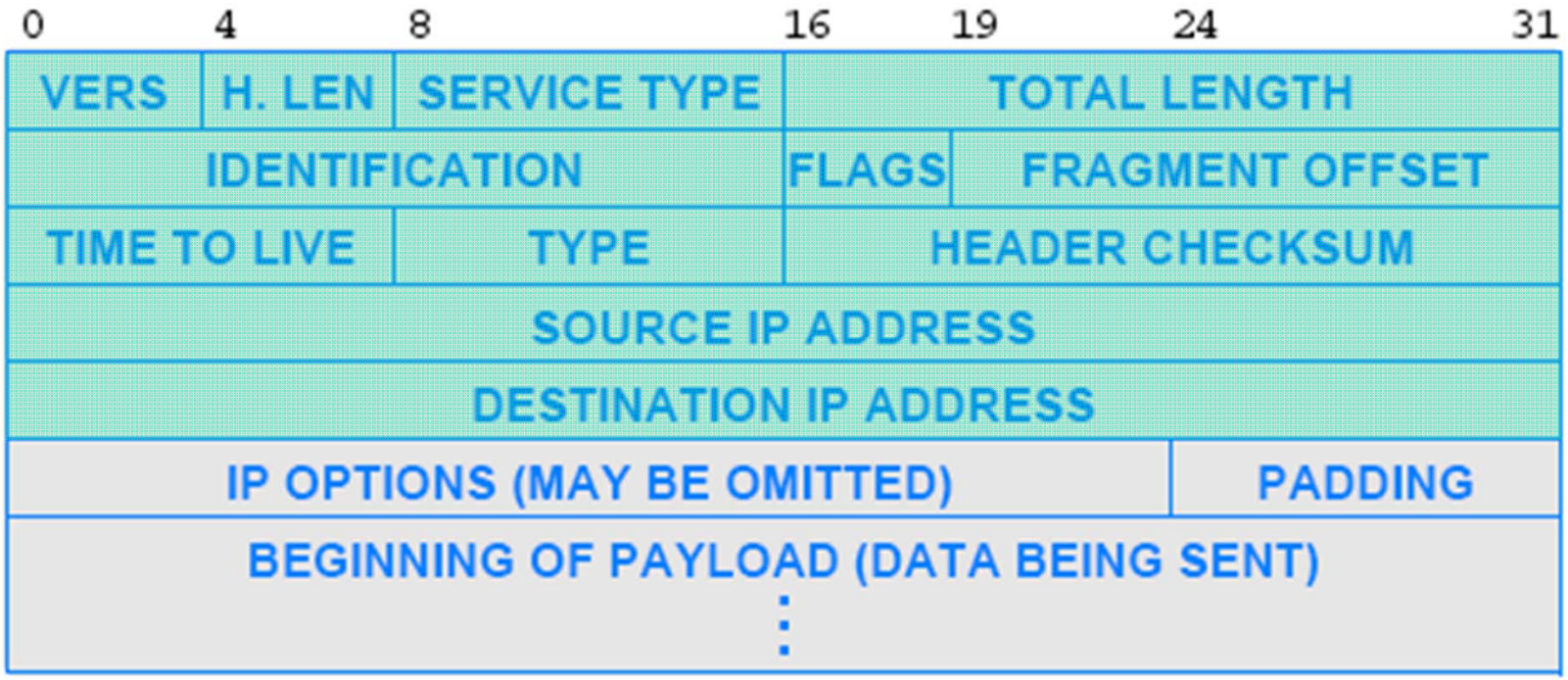
- **IDENTIFICATION.** กำหนด **Sequence Number** ใช้ในกรณีที่มีการ **Fragmentation**
- **FLAGS.** เป็นตัวกำหนดว่าจะยอมให้ทำ **Fragment** ได้หรือไม่ และกำหนด**Fragment** สุดท้าย

Ch22: 22.5 IP Datagram Header Format (IPv4)



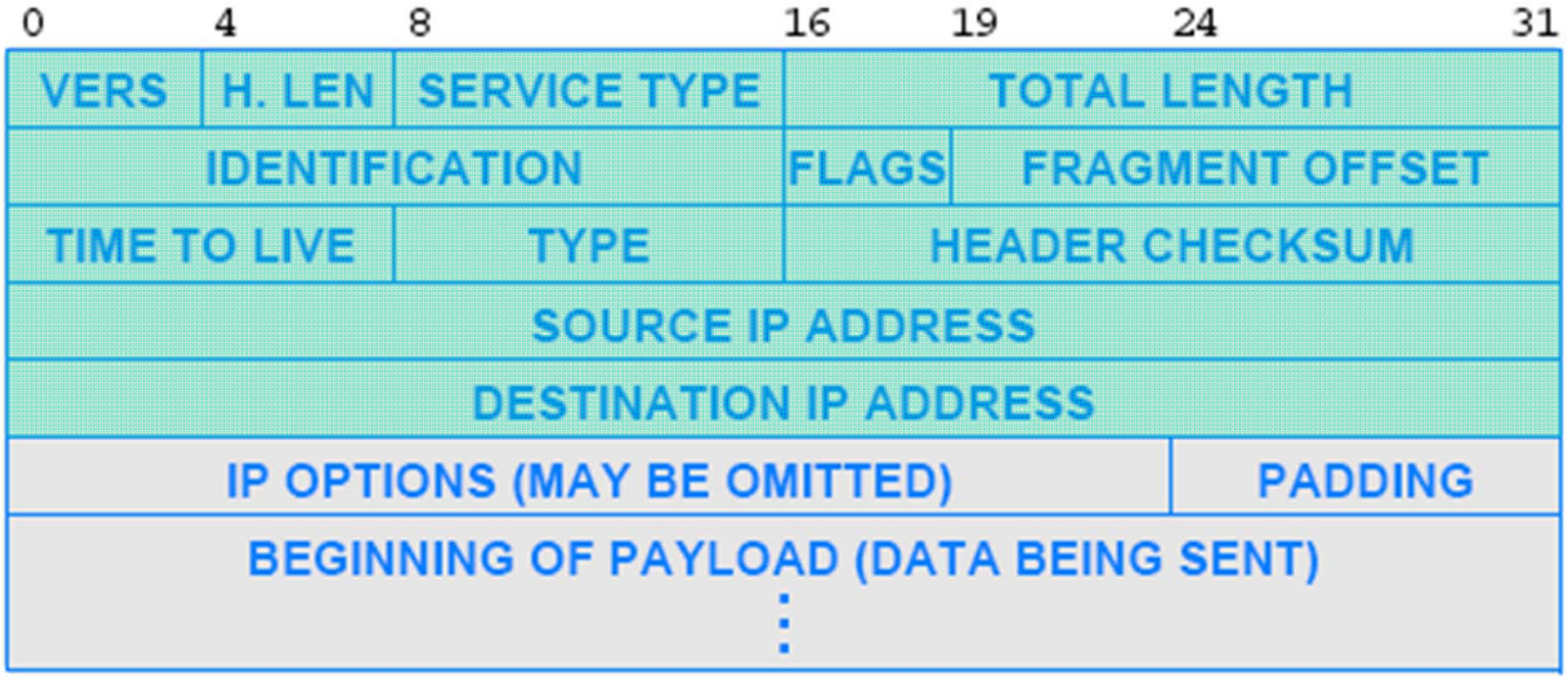
- **FRAGMENT OFFSET.** ค่าจะถูกคูณด้วย 8 เพื่อบ่งบ่งตำแหน่งของ Datagramนี้ใน Datagram ก่อน Fragment
- **TIME TO LIVE.** กำหนดโดยผู้ส่ง โดยจะเป็นตัวกำจัด Packet ในกรณีที่มี Loop ซึ่งค่านี้จะถูกลดลงหนึ่งทุกครั้งที่ผ่าน Router เมื่อค่านี้เป็นศูนย์ Datagram นี้จะถูกโยนทิ้ง และมีการส่ง Error Message กลับไปยังผู้ส่ง

Ch22: 22.5 IP Datagram Header Format (IPv4)



- **TYPE. กำหนดชนิด(Protocol) ของ Payload ที่อยู่ข้างใน (ที่อยู่ Layer บนถัดไป)**
- **HEADER CHECKSUM. ผลบวกของ 16 Bit Word และ Complement**

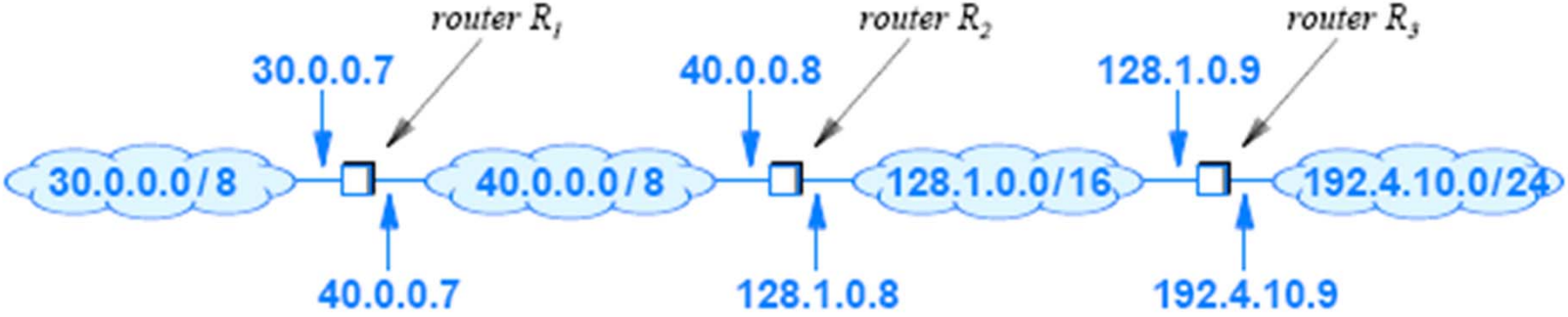
Ch22: 22.5 IP Datagram Header Format (IPv4)



- **SOURCE & DESTINATION IP ADDRESS.** กำหนดเครื่องต้นทางและปลายทาง ไม่ใช่ **Address** ของ **Router**
- **IP OPTIONS.** ใช้ในการควบคุมการทำ **Routing** ปกติจะไม่ใช้
- **PADDING.** ด้วยศูนย์เพื่อให้ครบ **32 Bit Word**

Ch22: 22.6 IP Datagram Forwarding

- **IP Router จะใช้ Forwarding Table (Routing Table) ซึ่งสามารถจะ Update ได้เป็นระยะ**
  - จะเป็นลักษณะของ Next-hop Forwarding



(a)

Router 2

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

(b)

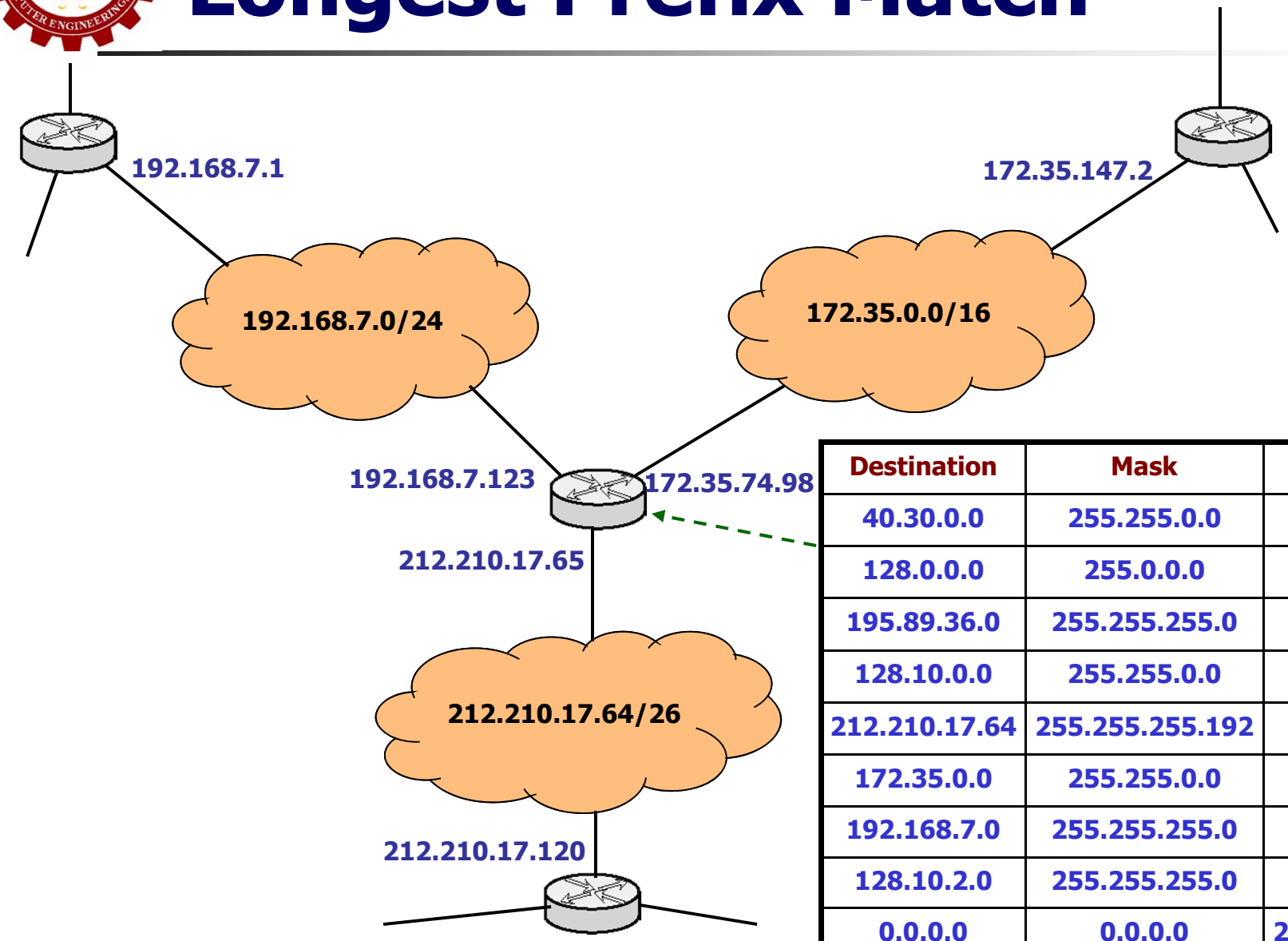
Ch22: 22.7 Network Prefix Extraction

- **จากที่กล่าวแล้ว Router จะดูแค่ส่วน Prefix ของ IP Address(NW ID) จากนั้นจะเปรียบเทียบกับตาราง Routing Table เพื่อหาเส้นทางในการส่งข้อมูลต่อไป ดังนี้**
  - ในแต่ละแถวของตาราง Routing Table ตัว Router จะนำ Mask จากแถวนั้นมาทำการ Bitwise AND กับ IP Address ปลายทางที่ได้จากส่วน Header ของ IP Datagram และเปรียบเทียบกับค่าในตาราง ถ้าค่า Prefix ที่ได้ตรงกันกับค่าจากแถวนั้นในตาราง ถือว่า Match และข้อมูลจะถูกส่งไปในทิศทางนั้น
    - ยกตัวอย่าง มี Datagram ต้องการส่งไปที่ 192.4.10.3 ส่งมาที่ Router R2
      - ในแถวแรกของตาราง เมื่อทำการ AND เราได้ 255.0.0.0 & 192.4.10.3 = 192.0.0.0 ซึ่งไม่ตรงกับ 30.0.0.0
      - แถวที่สองจะได้ 192.0.0.0 เช่นกันและไม่ตรงกับ 40.0.0.0
      - แถวที่สามจะได้ 192.4.0.0 และไม่ตรงกับ 128.1.0.0
      - แถวที่สี่จะได้ 192.4.10.0 ซึ่งตรงหรือ Match ดังนั้นข้อมูลจุถูกส่งออกไปที่ 128.1.0.9 คือ IP Address ของ Interface ของ R3 ที่ต่อกัน เรียกว่า Next Hop Address
    - การส่งให้ Router ตัวถัดไปนั้น จะไม่มีการเปลี่ยนแปลง IP Address ของ Datagram แต่จะส่งโดยใช้ Local Network Address แทน เนื่องจาก Interface ของ Router ที่ต่อตัวถัดไป จะอยู่ใน NW เดียวกันกับ Router ตัวที่ส่ง
    - ถ้าเป็น Direct Connect ตัว Router สามารถส่งข้อมูลให้กับ Host ได้โดยตรง ไม่ต้องส่งให้กับ Router ตัวถัดไป

Ch22: 22.8 Longest Prefix Match

- **ในทางปฏิบัติ ตาราง Routing Table อาจจะมีขนาดใหญ่ และ Routing Algorithm ค่อนข้างสลับซับซ้อน การส่งผ่านข้อมูลจะมีข้อกำหนดเพิ่มเติมที่สำคัญดังนี้**
  - ถ้าไม่มี Prefix ใด Match เลยในตาราง ข้อมูลจะส่งไม่ได้ ดังนั้นแถวสุดท้ายมักจะกำหนด 'Default Route' โดยใช้ Address 0.0.0.0 และ Mask เป็น 0.0.0.0 (Always Match)
  - ผู้ส่งสามารถกำหนด Route เองได้ โดยกำหนดเส้นทางส่งข้อมูลผ่าน Interface ต่างๆในส่วน Option ของ Header
  - ในกรณีที่มีการ Match Prefix มากกว่าหนึ่งอัน ตัว Router จะส่งข้อมูลในทิศทางที่มีการ Match Prefix ที่ยาวที่สุด
    - ดังนั้นในตาราง Routing Table, Software จะทำการเรียงตารางใหม่เริ่มจากแถวที่มี Prefix มากที่สุดก่อน
    - เช่นในตารางมีสอง Prefix คือ 128.10.0.0/16 และ 128.10.2.0/24 ถ้ามี Datagram ที่ต้องการส่งไปที่ 128.10.2.3 เข้ามา มันจะถูกส่งไปตามแถวที่กำหนดจาก 128.10.2.0/24 เพราะมีการ Match ยาวกว่า

Longest Prefix Match



Destination	Mask	Next Hop
40.30.0.0	255.255.0.0	172.35.147.2
128.0.0.0	255.0.0.0	192.168.7.1
195.89.36.0	255.255.255.0	192.168.7.1
128.10.0.0	255.255.0.0	172.35.147.2
212.210.17.64	255.255.255.192	Directed
172.35.0.0	255.255.0.0	Directed
192.168.7.0	255.255.255.0	Directed
128.10.2.0	255.255.255.0	192.168.7.1
0.0.0.0	0.0.0.0	212.210.17.120

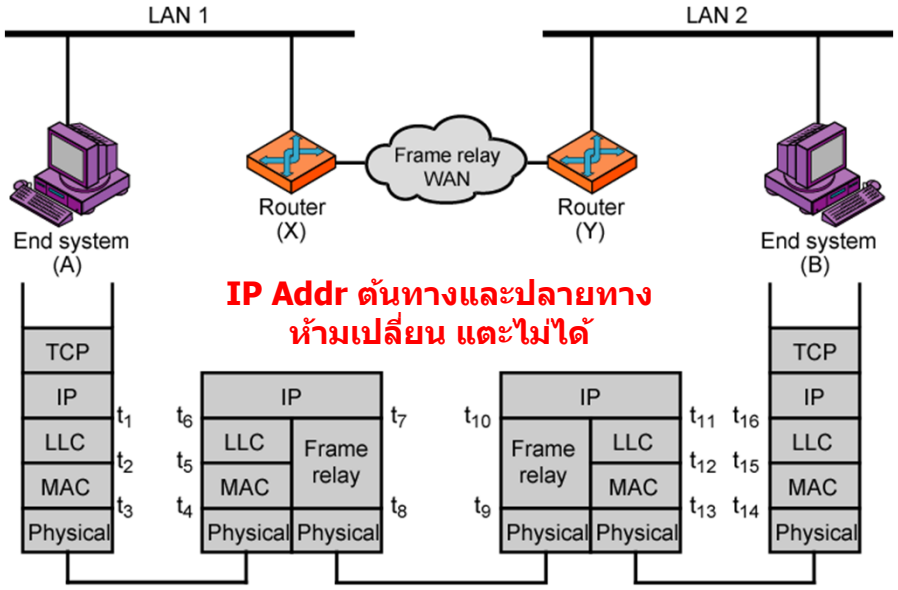
Ch22: 22.10 Best Effort Delivery

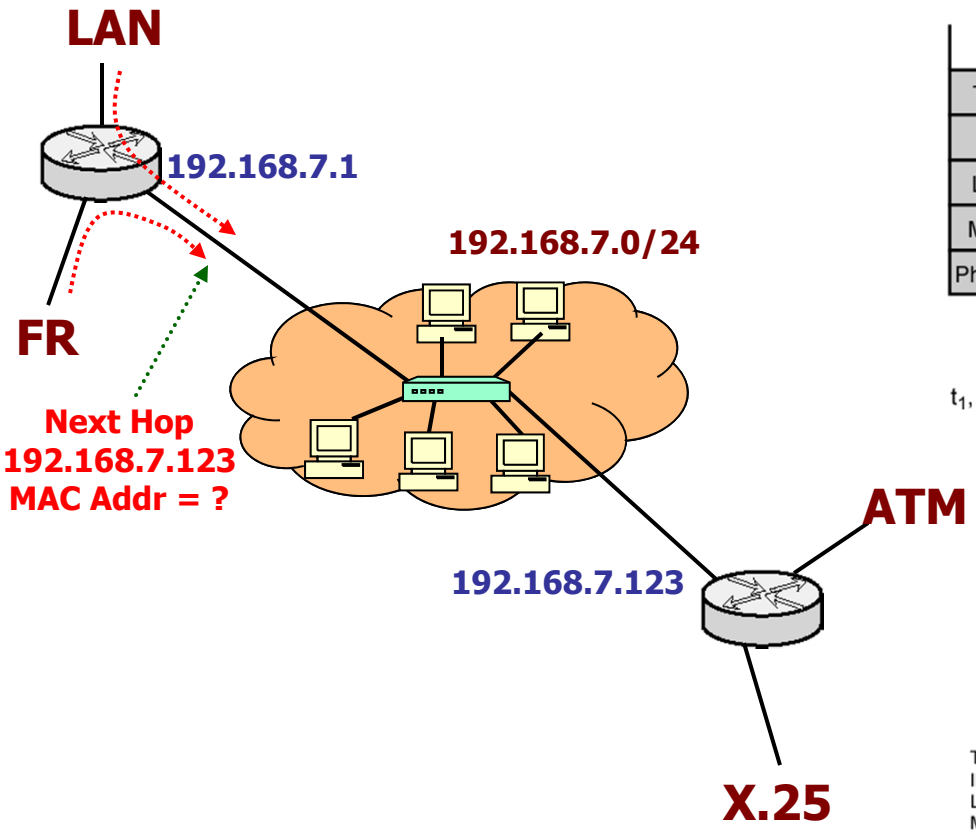
- **การส่ง Datagram ใน IP Network จะเป็นลักษณะ Best-Effort Delivery คือส่งเท่าที่จะสามารถส่งได้ อย่างไรก็ตามสิ่งต่างๆเหล่านี้สามารถเกิดขึ้นได้**
  - Datagram Duplication
  - Delayed หรือ Out-of-Order Delivery
  - Data Corruption
  - Datagram Loss
- **กล่าวคือ จะมี Error เกิดขึ้นได้เสมอในการส่งข้อมูลผ่าน Internet**
- **ในการนี้ มาตรฐาน TCP/IP ได้ออกแบบ Protocol อีก Layer หนึ่งเพื่อที่จะมาจัดการกับเรื่องเหล่านี้ (TCP)**
  - ขึ้นอยู่กับผู้ใช้งาน จะเลือกแค่ Best-Effort Delivery หรือต้องการ Protocol เสริมที่จะช่วยจัดการกับ Error
    - UDP/IP หรือ TCP/IP

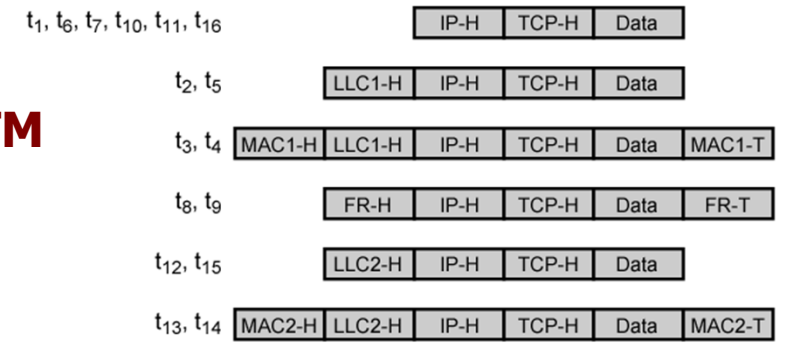
Ch22: 22.11 IP Encapsulation

- **ในการส่ง Datagram ผ่านแต่ละ Network จำเป็นที่จะต้องบรรจุ Datagram ลงใน Network Protocol ที่กำลังผ่าน**
  - เรียกว่าการทำ Encapsulation
- **เมื่อ Encapsulated Datagram ผ่านจาก Network หนึ่งไปยังอีก Network หนึ่ง (ผ่าน Router) มันจะถูก Decapsulated และถูก Encapsulate ใหม่ตาม Protocol ของ Network ถัดไป**
  - การ Decapsulate นั้น จะต้องรู้ว่ามันเป็น IP Datagram Encapsulate มิฉะนั้นแล้วการทำงานจะผิดพลาด เนื่องจากแต่ละ Network อาจจะมีการ Encapsulate หลาย Protocol
  - ดังนั้นที่ส่วนหัวของ Network Protocol ที่ทำการ Encapsulate จะต้องมี Code กำกับบอกชนิดของ Payload
    - ใน Ethernet Frame ถ้าเป็น IP Datagram บรรจุอยู่ใน MAC Frame จะใช้ Ethertype 0x0800
  - นอกจากนี้แล้ว การ Encapsulate ยังต้องการการกำหนด Network Address ของ Router ที่ Datagram จะถูกส่งออกไปยัง NW ถัดไป (หรือ NW Address ของ Host ในกรณีที่ Datagram ไปถึงที่หมาย)
    - อย่าลืมว่า Router กำหนดแค่ IP Address ของ Next-Hop Router ไม่ได้กำหนด Network Address ของ Next-Hop Router
    - เราจะใช้ Protocol ชื่อ ARP มาทำการหา Network Address จาก IP Address ของ Next-Hop Router (Interface)
    - กรณีของ Ethernet Network, ARP จะใช้ในการหา MAC Address จาก IP Address ที่กำหนด รายละเอียดจะอยู่ในบทถัดไป

Ch22: 22.11 IP Encapsulation







TCP-H = TCP header
IP-H = IP header
LLCi-H = LLC header
MACi-H = MAC header
MACi-T = MAC trailer
FR-H = Frame relay header
FR-T = Frame relay trailer

Ch22: 22.12 Transmission Across An Internet

- การส่ง Datagram ใน Internet จะถูก Encapsulate-Decapsulate เป็นทอดๆ ตาม Protocol ในแต่ละ Network ที่ผ่าน
  - มีการกำหนด Code ของ Payload ที่ส่วนหัวของแต่ละ Network Protocol
  - มีการหา Network Address ของ Router ตัวถัดไป ที่ตรงกับ IP Address โดยใช้ ARP

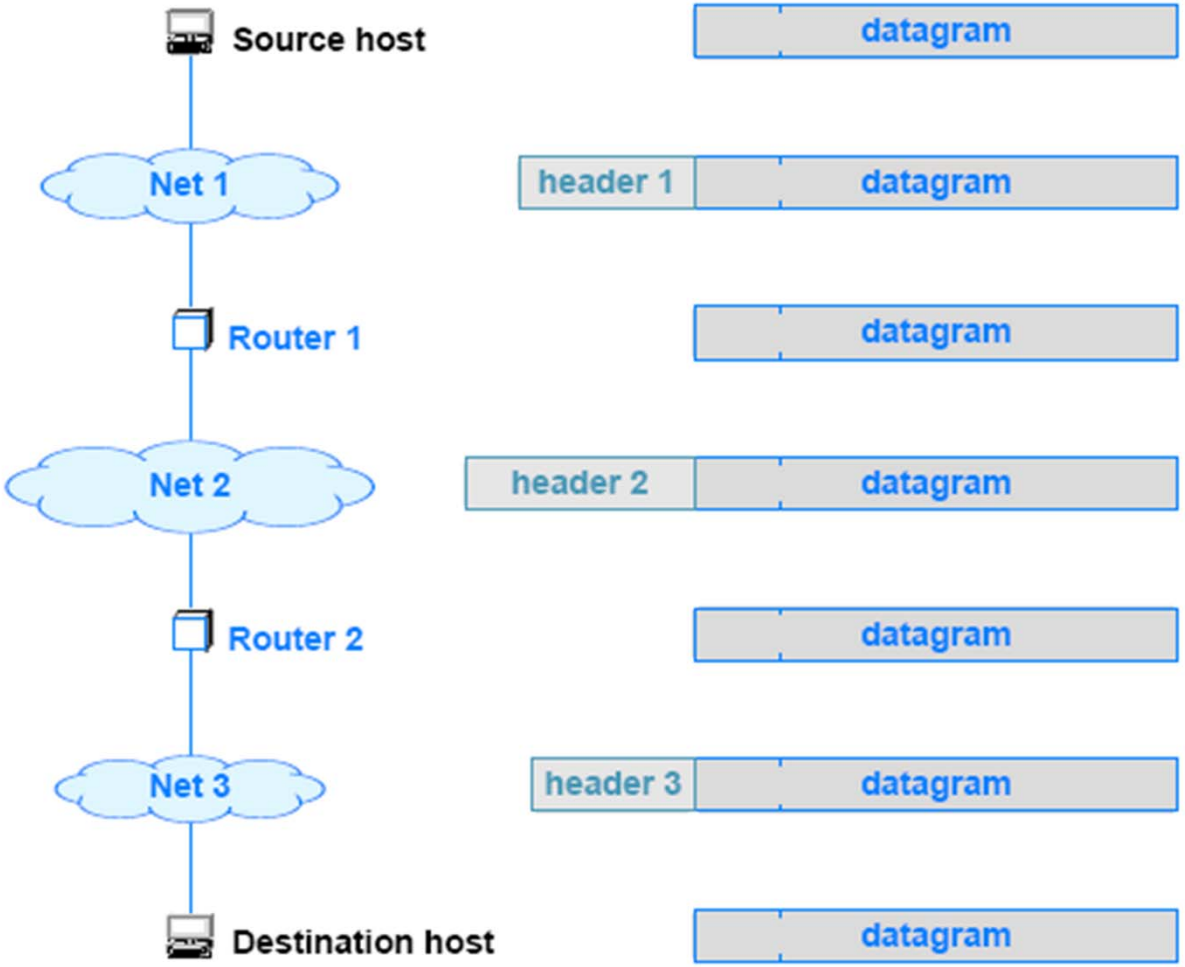


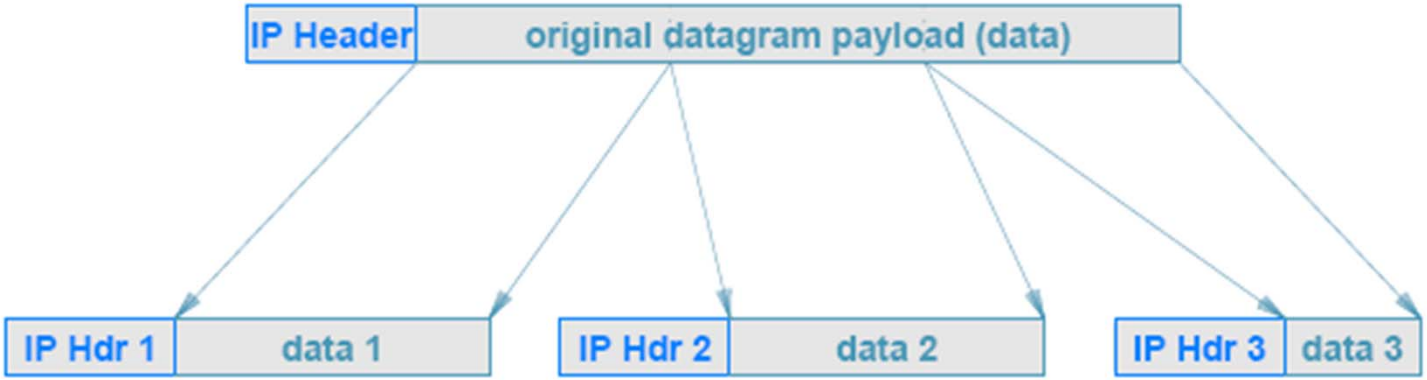
Figure 22.5 An IP datagram as it travels across the Internet.

Ch22: 22.13-14 MTU and Datagram Fragmentation/Reassembly

- **ขนาดของ Data ที่จะใส่ใน Frame สูงสุดที่สามารถส่งผ่านได้เรียก MTU: Maximum Transfer Unit**
  - แต่ละ Network จะกำหนดค่า MTU ต่างกัน
  - เป็นไปได้ว่า Datagram ที่ส่ง อาจจะมีขนาดใหญ่กว่าค่า MTU ของ NW นั้น
    - จะต้องมีการแตก Datagram เป็นส่วนสั้นๆ เรียก Fragment เพื่อจะสามารถส่งผ่านไปได้ ขบวนการนี้เรียก Fragmentation

Figure 22.6 Illustration of a router that connects two networks with different MTUs.

Ch22: 22.13-14 MTU and Datagram Fragmentation/Reassembly



- **ในแต่ละ Fragment ของ Datagram จะมี Format เหมือน Datagram ปกติ ส่วน Header ของ Fragment จะกำหนด IP ต้นทางและปลายทางเหมือน Header เดิม ในส่วน Field 'FLAG' จะบ่งบอกว่านี่คือ Fragment ไม่ใช่ IP Datagram เต็มๆ และส่วนใน 'FRAGMENT OFFSET' จะบ่งบอกว่า Fragment นี้อยู่ที่ส่วนใหนของ Datagram เดิม**
  - การประกอบคืน (Reassembly) กำหนดให้กระทำที่ Host ปลายทาง
  - เพราะ Host ปลายทางสามารถรู้ได้ว่า Fragment ได้ครบหรือไม่
  - ปกติ Router จะไม่สนใจว่า Datagram ที่ส่งเป็น Fragment หรือไม่

IP Header Flags

- **จากที่กล่าวมาแล้ว ส่วนของ Header จะประกอบด้วย FLAG ขนาด 3 บิต ใช้ในกรณีที่ทำ Fragment ดังนี้**
  - Bit 0: จะถูก Reserve และจะต้องมีค่าเป็น 0
  - Bit 1: Don't Fragment(DF) ถ้ามีการ Set จะไม่มีการทำ Fragment ที่ Datagram นี้
    - เมื่อ Datagram ต้องผ่าน NW ที่กำหนดค่า MTU น้อยกว่าขนาดของ Datagram จะส่งผ่านไม่ได้ และจะต้องโยนทิ้ง
  - Bit 2: More Fragment(MF) จะ Set ถ้า Datagram นี้เป็น Fragment ยกเว้น Fragment สุดท้าย
    - ถ้าไม่มีการทำ Fragment ส่วน Bit 2 นี้จะไม่ถูก Set เช่นกัน ซึ่งมีความหมายเดียวกับเป็น Fragment สุดท้าย

Ch22: 22.15 Collecting The Datagram Fragments

- **IP จะไม่ Guarantee การส่งข้อมูล**
  - Fragment อาจหาย มี Error หรือถูกส่งไปในทิศทางที่ต่างกัน และมาถึงปลายทางอย่างไม่เป็นลำดับ
- **เราจะรู้ได้อย่างไรว่า Fragment ของแต่ละ Datagram ได้รับครบแล้ว พร้อมจะประกอบเป็น Datagram เดิมคืนมา**
  - Bit ในส่วนของ 'FLAG' Field จะบอกว่านี่เป็น Fragment ให้รอ Fragment อื่นจนครบ และประกอบเป็น Datagram เดิมก่อนที่จะส่งให้ Layer บนต่อไป
  - ปลายทางจะใช้ส่วนของ IDENTIFICATION Field รวมกับ Source IP Address เป็นตัวกำหนดว่า Fragment ที่ได้อยู่ใน Datagram เดิมอะไร โดยที่ Fragment ของ Datagram เดียวกันจะมีค่า 'IDENTIFICATION' เหมือนกัน ส่วน Source IP Address จะบ่งบอกอีกทีให้แน่ใจ เพราะแต่ละผู้ส่งอาจจะใช้ Identification เหมือนกัน
  - ส่วนของ 'FRAGMENT OFFSET' Field จะบ่งบอกว่า Fragment นี้อยู่ส่วนไหนของ Datagram เดิม

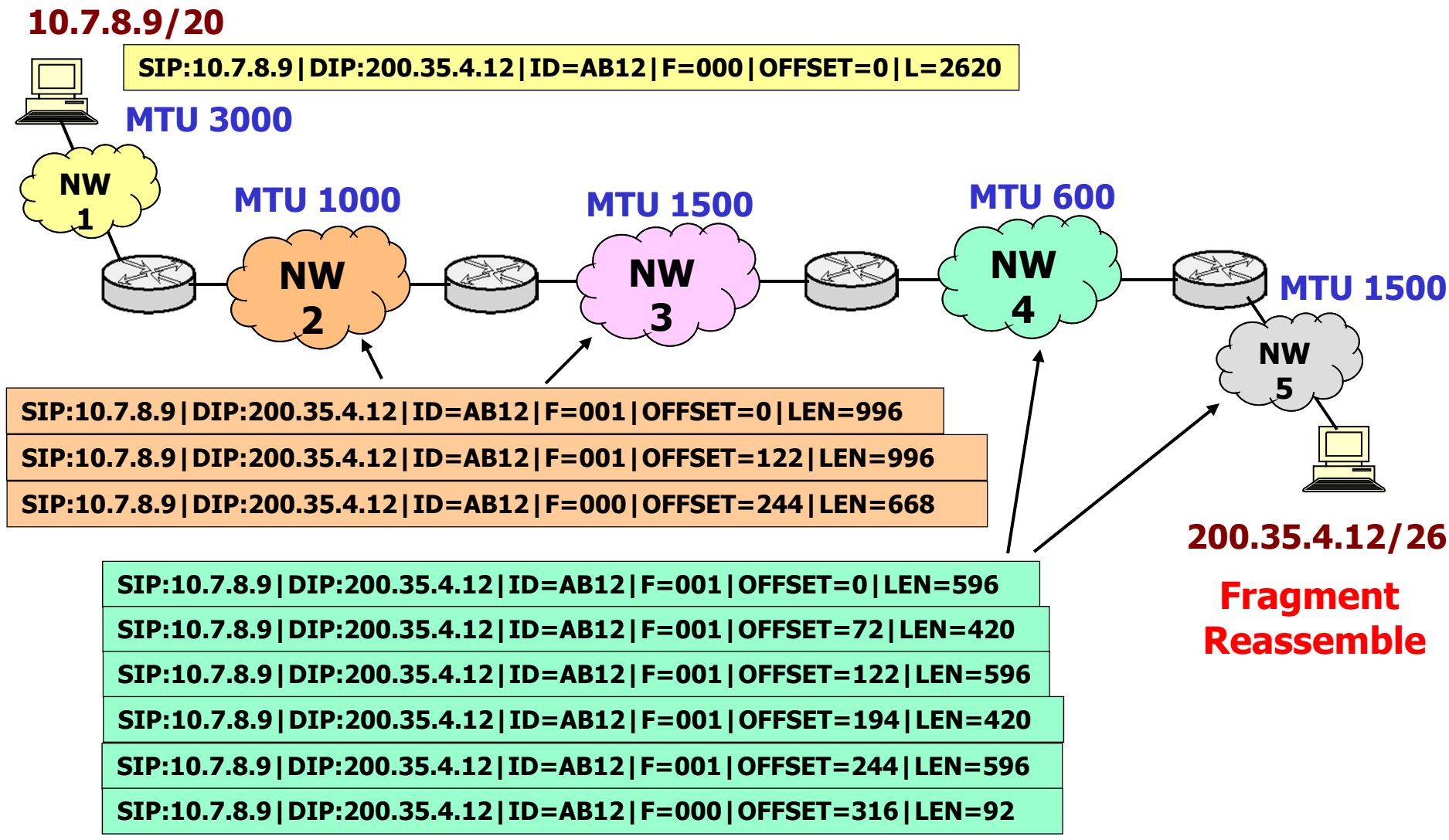
Ch22: 22.16 Fragment Loss

- **Fragment ต้องมาครบจึงจะประกอบได้ ดังนั้นถ้ามี Fragment Loss ปลายทางจะต้องเก็บ Fragment ที่เหลือไว้จนกว่าจะครบ ซึ่งจะเกิด Delay**
- **การเก็บ Fragment จะใช้ Memory และไม่สามารถเก็บได้ตลอดไป มิฉะนั้น Resource ของ Host ปลายทางจะไม่เหลือให้ทำอย่างอื่น**
- **IP กำหนด Timer ชื่อ 'Reassembly Timer' ซึ่งถ้า Timer Expire(หมดอายุ) ทุกๆ Fragment ในชุดของ Datagram นั้นจะถูกโยนทิ้ง**
  - IP ไม่มี Mechanism สำหรับ Fragment Retransmission เนื่องจาก Fragment กระทำที่กลางทาง
    - ที่จริงจะกระทำไม่ได้ เพราะผู้ส่งไม่รู้ว่า Fragment มันแบ่งอย่างไร ที่ส่วนใหนของ Datagram เดิม
  - ดังนั้นถ้า Fragment Loss จะทำให้ Timer Expire และจะลงเอยด้วยการส่งทั้ง Datagram นั้นมาใหม่ (All-or-Nothing)

Ch22: 22.17 Fragmenting A Fragment

- **เป็นไปได้ที่ Router ตัวหนึ่งทำการ Fragment Datagram ส่งไปใน Internet และ Fragment นั้นต้องผ่าน Network ที่มีค่า MTU ต่ำกว่าขนาดของ Fragment**
  - เราสามารถทำ Fragment ของ Fragment กี่ระดับก็ได้
  - Router จะมองไม่ออกว่านี่เป็น Fragment ของ Fragment
  - การประกอบจะกระทำครั้งเดียวที่ปลายทาง
    - ไม่มีการประกอบจาก Sub-fragment เป็น Fragment ก่อน แล้วจึงค่อยประกอบ Datagram

IP Fragmentation



End of Chapter 21-22

- **Homework II Due Next Week**
  - Download HW2 (Week 4)

End of Week 4

- **Next, Week V: Chapter 23**
  - Chapter 23: Supporting Protocol and Technologies
    - ARP
    - ICMP
    - DHCP
    - NAT