



CPE 426 Computer Networks

Chapter 11:
PART IV in Textbook
Text Chapter 28: NW Performance
(QoS and DiffServ)
Text Chapter 29: IP Telephony





TOPICS

- **Chapter 28 (Only Selected Topics)**
 - 28.2 Performance Measurement
 - 28.3 Latency or Delay
 - 28.4 Throughput, Capacity and Goodput
 - 28.5 Understanding Throughput and Delay
 - 28.6 Jitter
 - 28.7 Relationship Between Throughput and Delay
 - 28.8 Measuring Delay, Throughput and Jitter
 - 28.9 Passive Measurement, Small Packet and Netflow
 - 28.10 QoS
 - 28.11 Fine-Grain and Coarse-Grain QoS
 - 28.12 Implementation of QoS
 - 28.13 Internet QoS Technologies
- **Breaks**
- **Chapter 29 (Only Selected Topics)**
 - Real-Time Transmission
 - Delay and Jitter
 - RTP
 - IP Telephony
 - Signaling
 - IP Telephone System Components
 - Protocol and Layering
 - H.323
 - SIP
 - Telephone Number Mapping



28.2 Performance Measurements

- การใช้คำว่า **Low-Speed Network** หรือ **High-Speed Network** ไม่เพียงพอต่อการบ่งบอกการทำงานของมัน เนื่องจาก **Network Technologies** เปลี่ยนเร็วมาก
 - Low-Speed ปัจจุบัน เป็น High-Speed เมื่อ 2-3 ปีก่อน
- ตัววัด **Performance** หลักๆที่ใช้มีสามตัว
 - Latency(Delay) เป็นตัววัดเวลาที่ต้องใช้ในการส่งข้อมูลผ่าน Network
 - Throughput(Capacity) เป็นตัววัดจำนวนข้อมูลที่ส่งได้ในหนึ่งหน่วยเวลา
 - Jitter (Variability) เป็นค่าการเปลี่ยนแปลงของค่า Delay และระยะเวลาของการเปลี่ยนแปลง



28.3 Latency or Delay

- สามารถแบ่งออกได้เป็น
 - Propagation Delay
 - เวลาที่สัญญาณต้องเดินทางผ่าน Transmission Medium ขึ้นอยู่กับความเร็วของสัญญาณและระยะทางที่ส่ง
 - Access Delay
 - เวลาที่ใช้ในการ Access Medium ยกตัวอย่างเช่นใน LAN ที่ใช้ CSMA/CD หรือใน WLAN ที่ใช้ CSMA/CA
 - Switching Delay
 - เวลาที่ใช้ในการส่งผ่าน Packet ในอุปกรณ์ Switch หรือ Router (เวลาที่อุปกรณ์ต้องใช้ในการ Process Packet)

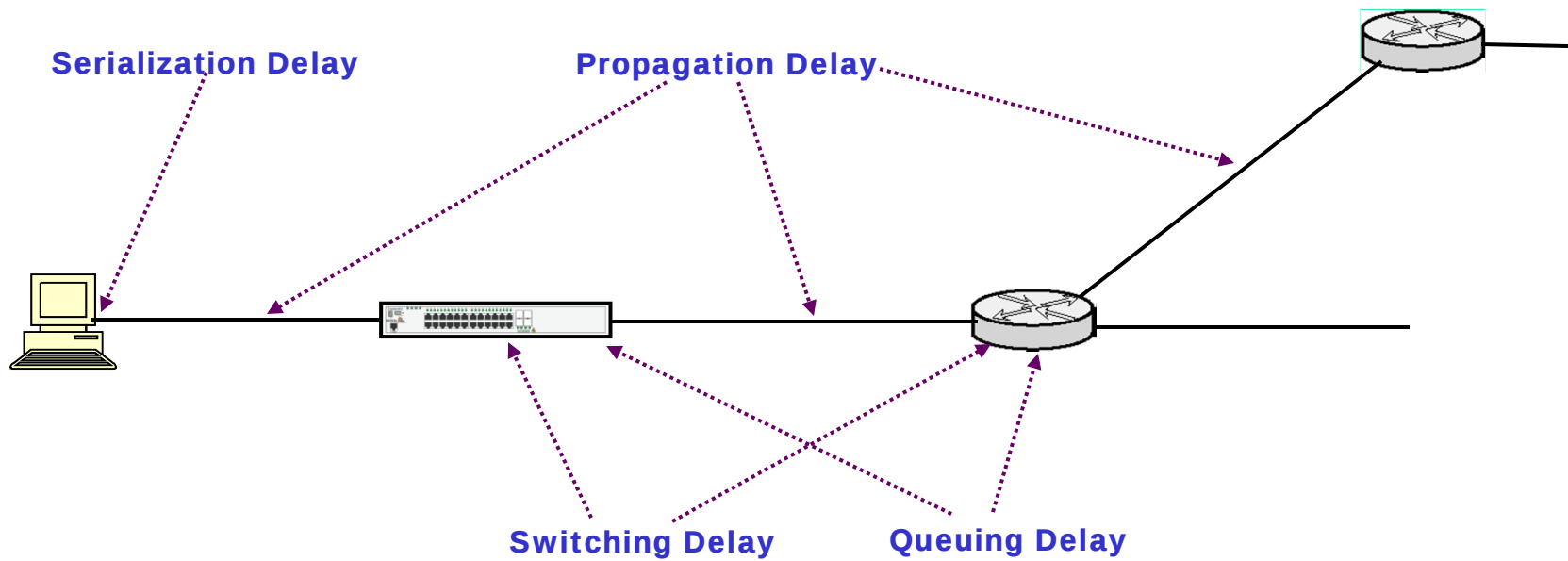


Latency or Delay

- สามารถแบ่งออกได้เป็น(2)
 - Queuing Delay
 - เวลาที่ต้องรอใน Queue ก่อนที่จะถูกส่งออกไป
 - ขึ้นอยู่กับความยาวของ Queue
 - ค่านี้เป็นตัวกำหนดค่า Delay ใน Network
 - Server Delay
 - เวลาที่ Server ใช้ในการตรวจสอบการร้องขอและคำนวณการตอบสนอง
 - Serialization Delay
 - เวลาที่ต้องใช้ในการส่ง Packet ขึ้นกับ Data Rate และความยาวของ Packet



Latency





28.4 Throughput, Capacity and Goodput

■ Capacity

- เป็นค่า Data Rate ของการส่งข้อมูล และแสดงจำนวน Data ที่สามารถส่งได้สูงสุด วัดเป็น bps

■ Throughput

- เป็นจำนวน Data ที่ส่งได้จริงผ่าน Network วัดเป็น bps (ไม่รวม Retransmission)
- บางทีวัดเป็น Percent ของ Capacity เรียก Utilization
- ค่านี้ปกติจะน้อยกว่า Capacity ขึ้นอยู่กับการทำ Flow Control

■ Goodput

- เป็นจำนวน Data จาก Application Layer ที่ส่งได้จริงๆ โดยตัด Overhead เช่นส่วน Header ของแต่ละ Layer ออก และไม่รวมส่วน Control Information, Support Protocol, Handshake, Congestion และ Retransmission
- ต่างจากค่า Throughput ซึ่งจะรวมส่วนของ Overhead ด้วย



28.5 Throughput and Delay

- **Propagation Delay** เป็นตัวกำหนดเวลาที่แต่ละ Bit จะเดินทางใน Network จากต้นทางไปถึงปลายทาง
- **Throughput** กำหนดจำนวนบิตที่สามารถวิ่งผ่าน Network ได้ในแต่ละเวลา
- เราสามารถซื้อ Throughput เพิ่มขึ้นได้ แต่ไม่สามารถจ่ายเงินเพิ่มเพื่อลดค่า Delay ได้
- ส่วน **Queuing Delay** เป็นเวลาที่ต้องรอใน Queue ของ Switch/Router ก่อนที่สามารถจะส่งข้อมูลได้



28.6 Jitter

- ค่านี้มีความสำคัญในกรณีที่เราต้องส่ง **Real-Time Voice หรือ Video**
- คือการเปลี่ยนแปลง หรือค่า **Variance** ของค่า **Delay**
- ปกติการส่ง **Real-Time Data** จะมีตัว **Buffer** ชื่อ **Jitter Buffer** ป้องกันการผันแปรของเวลาที่ข้อมูลมาถึง
 - ถ้าค่า Jitter สูงจะทำให้ Buffer ว่าง หรือ **Overflow** และเสียงจะขาดหายหรือภาพจะกระตุก
 - วิธีแก้คือใช้ Buffer ขนาดใหญ่ขึ้น แต่มีข้อเสียคือจะเกิด **Delay** ใน Buffer สูง และมันจะไม่เป็น **Real-Time** อย่างแท้จริง



Jitter

- **Real time Application เช่น Audio หรือ Video ต้องการรับข้อมูลอย่างต่อเนื่องและตามกำหนดเวลา**
 - Jitter ทำให้ข้อมูลบางส่วนมาถึงในเวลาที่ย่ำเกินกำหนดที่จะต้องแสดงผล ทำให้คุณภาพลดลง
 - แม้ว่า Network จะมี Delay ต่ำ แต่ค่า Jitter สูง จะส่งผล Network ที่มี Delay ปานกลาง แต่ Jitter ต่ำไม่ได้ เพราะ Delay แสดงเพียงค่าเฉลี่ยแต่ Jitter แสดง Variance (Standard Deviation)
- **การจัดการกับ Jitter**
 - ใช้ Isochronous Network (Circuit Switching Network)
 - ใช้ Protocol ในการช่วยจัดการกับ Jitter
 - Real-Time Protocol + Jitter Buffer



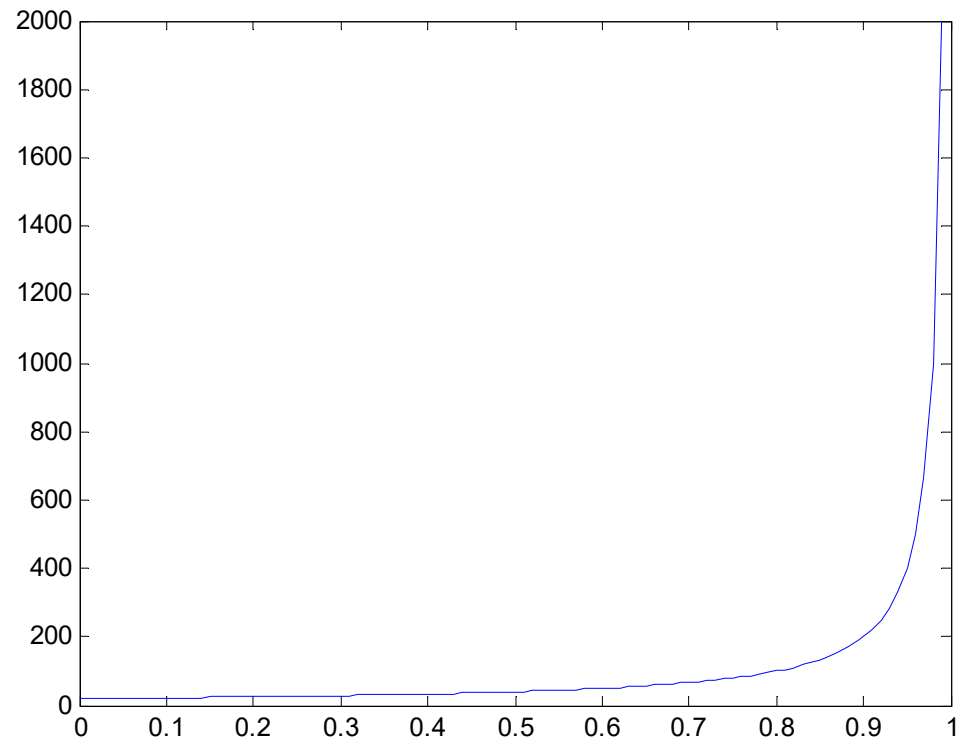
28.7 ความสัมพันธ์ระหว่าง Delay และ Throughput

- สามารถอธิบายโดย Queuing Theory (CPE332)
- ถ้าให้ u เป็นค่า Utilization เราจะได้ค่า Delay, D ดังนี้

$$D = \frac{D_0}{1-U}$$

$$U = 0.5 \Rightarrow D = 2D_0$$

D_0 คือ Delay ในกรณีที่
Network Idle





ความสัมพันธ์ระหว่าง Delay และ Throughput

- ค่า Delay-Throughput Product
- เป็นจำนวนของ Bit ที่วิ่งผ่าน Network ในเวลาใดเวลาหนึ่ง
 - Bits in Network = $D \times T$ (sec. x bit/sec.)



28.8 การวัดค่า Delay, Throughput และ Jitter

- การวัดค่า Throughput และค่า Jitter ทำได้ไม่ยาก โดยการส่ง Packet ติดต่อกัน และจับเวลาว่าส่งข้อมูลได้เท่าไร และแต่ละ Packet ที่ไปถึงห่างกันเท่าไร
 - อาจจะใช้ Protocol Analyzer ช่วยในการคำนวณ เช่น Sniffer, Ethereal หรือ Wireshark
- Delay จะวัดยากกว่า เพราะต้องจับเวลาที่ส่ง และที่รับ
 - หมายถึงเครื่องส่งและเครื่องรับต้อง Synchronization กัน
 - Protocol Analyzer ต้องมีอุปกรณ์เสริมช่วย บางตัวจะทำได้
 - เราอาจจะวัดค่า Round Trip Time แทน เช่นใช้การ 'Ping'
- ค่าที่วัดได้เหล่านี้ ค่อนข้างจะซับซ้อน
 - Network เป็น Asymmetric
 - Network มีการเปลี่ยนสถานะค่อนข้างเร็ว
 - การวัดค่าจะ Load Network และทำให้ค่าเปลี่ยน
 - Traffic ใน Network ปกติจะมีลักษณะ Burst



28.9 Passive Measurement, Small Packet and Netflow

- การวัดแบบ **Active** เราต้อง **Inject Traffic** ลงใน **Network** อาจจะทำให้ค่าเปลี่ยน
- เราสามารถเลือกใช้วิธีการวัดแบบ **Passive** คือ **Monitor Network** และนับจำนวน **Packet** ที่ผ่าน
 - เทคนิคที่นิยมคือ 'Netflow'
 - Router จะสุ่มตัวอย่าง Packet จากนั้นจะวิเคราะห์ข้อมูลจากตัวอย่างนั้น



Quality of Service (QoS)

- เป็นการออกแบบ **Network** ที่สามารถให้ระดับของการให้บริการแก่ผู้ใช้
 - ผู้ใช้สามารถเลือกบริการตามความเหมาะสม
 - Real-Time Voice/Video
 - Web Service
 - E-Mail
- **Internet IPv4 เป็น Best Effort ไม่มี QoS** เนื่องจาก **IP Network เป็น Datagram**
 - QoS Mechanism ได้ถูกนำมาใช้กับ IPv6 ก่อน ต่อมาภายหลังเมื่อ IPv6 ยังไม่ได้รับความนิยม จึงมีการนำ Mechanism เหล่านี้มาใช้กับ IPv4



- ผู้ให้บริการยอมให้ลูกค้ากำหนดความต้องการ QoS ที่เฉพาะเจาะจง เช่นกำหนด Maximum Delay, Minimum Data Rate หรืออื่นๆ

- ผู้ให้บริการจัดแบ่ง Class ของการให้บริการ แต่ละ Class จะมีค่า QoS (BW, Delay, Jitter) ต่างกัน จากนั้นลูกค้าเลือกจะใช้ Class ใดที่เหมาะสมกับตัวเอง



28.11.1 Fine-Grain QoS and Flow

- **Network** แรกที่มีการให้ QoS แบบ Fine Grain ตั้งแต่เริ่มออกแบบ Protocol คือ ATM
- ปัจจุบัน ATM ตายไปแล้ว แต่คำศัพท์เกี่ยวกับ QoS ที่ ATM ให้บริการยังใช้อยู่
 - CBR Constant Bit Rate สามารถส่งข้อมูลได้ในอัตราที่คงที่
 - VBR Variable Bit Rate สามารถส่งข้อมูลได้ในอัตราเฉลี่ย และอยู่ในช่วงที่กำหนด
 - กำหนด SBR (Sustain Bit Rate), PBR (Peak Bit Rate),
 - กำหนด SBS (Sustain Burst Size), PBS (Peak Burst Size)
 - ABR Available Bit Rate สามารถส่งข้อมูลได้ตาม Bandwidth ที่เหลือ
 - UBR Unspecified Bit Rate ไม่กำหนดอัตราการส่งข้อมูล ส่งได้แบบ Best-Effort
- **Mechanism** ที่ Internet นำมาใช้ในการให้บริการแบบ Fine-Grain เรียก Integrated Service (IntServ)



28.11.12 Coarse-Grain QoS and Class of Service

- ทางเลือกอีกทางหนึ่งในการทำ QoS คือใช้วิธีของ **Coarse-Grain**
 - Traffic จะถูกแบ่งออกเป็น Class และกำหนด QoS Parameter ให้กับแต่ละ Class
- เนื่องจากใน Internet นั้นการทำ Fine-Grain จะทำได้ยาก เพราะ Router จะต้องจดจำ Flow และ State ของ Flow ซึ่งที่ Core Router ประกอบด้วยหลายล้าน Flow
 - นอกจากนี้ User ปกติจะไม่เข้าใจ QoS เพียงพอในการเลือก Performance Parameter
- **Coarse-Grain** สามารถกระทำได้ในทางปฏิบัติ



28.12 Implementation of QoS

- Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step

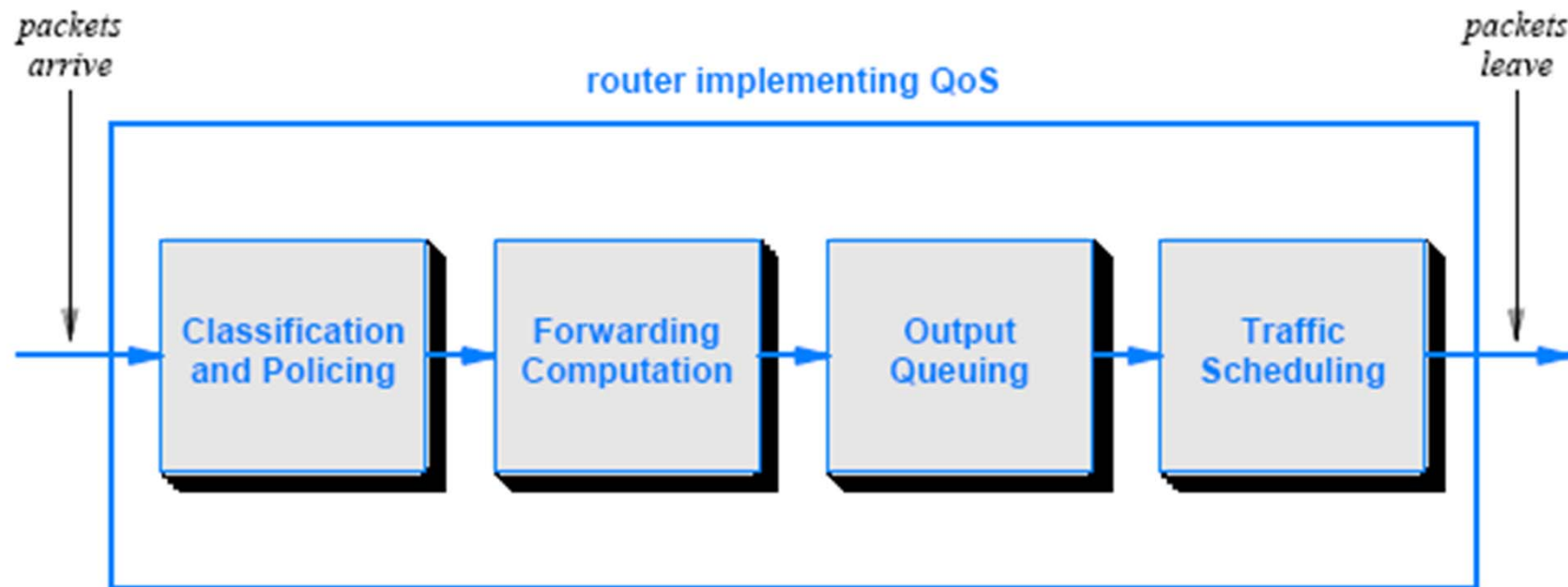


Figure 28.5 The four key steps used to implement QoS.



Implementation of QoS

- **Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step**
 - **1. Classification and Policing**
 - Router จะแบ่ง Class ของ Traffic โดยกำหนด Flow ID ให้แก่ Packet(Fine-Grain) หรือ กำหนด Class(Coarse-Grain) จากนั้น Router จะตรวจสอบ Policy ของ Class หรือ Flow นั้น ถ้าลูกค้าไม่กระทำตาม Policy ที่กำหนด จะมีการโยน Packet ทิ้งโดยใช้ค่า Probability



Implementation of QoS

- **Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step**
 - **2. Forwarding Computation**
 - คำนวณหา Next-Hop ของ Packet จากค่า Flow-ID ซึ่งอาจจะกำหนด Path ของการส่งข้อมูลสำหรับบาง Flow หรืออาจจะดูจาก IP ปลายทางและใช้ตาราง Routing Table
 - **3. Output Queuing**
 - Router ที่ทำ QoS จะมีหลาย Output Queue (ผิดกับ Best-Effort Router ที่มี FIFO Queue เดียวและเป็น M/M/1) โดยส่วนนี้ Packet จะถูกนำมาแยกใส่ Queue ที่เหมาะสมสำหรับ Flow ของมัน
 - Coarse Grain ปกติจะใช้ 1 Queue ต่อหนึ่ง Class
 - Fine-Grain ปกติจะใช้ 1 Queue ต่อ Connection



Implementation of QoS

- Router ที่สามารถทำ QoS จะประกอบไปด้วย 4 Step
 - 4. Traffic Scheduling
 - เป็นการเลือกที่จะนำ Packet ของแต่ละ Queue ส่งไปอย่างไร Queue ไหนส่งก่อน และส่งได้ครั้งละกี่ Packet
 - Traffic Management มีหลายวิธี ที่สำคัญมีดังนี้
 - Leaky Bucket: ยอมให้แต่ละ Queue ส่งได้ในอัตราคงที่ โดยการนับและควบคุม Packet ที่ส่ง ถ้าส่งข้อมูลต่ำกว่าที่ยอมให้ในบางเวลา จะสามารถส่งข้อมูลเพิ่มได้บ้างในโอกาสต่อไป โดยใช้ Counter ในการควบคุมการส่ง
 - Token Bucket: ยอมให้ Queue ส่งข้อมูลได้คงที่เช่นกัน แต่จะนับเป็น Byte ที่ส่ง สามารถยอมให้ส่งข้อมูลแบบ Burst ได้บ้าง แต่ค่าเฉลี่ยต้องไม่เกินค่าที่กำหนด โดยมีการกำหนด Token ให้กับผู้ส่งเป็นระยะ



Implementation of QoS

■ ต่อ Traffic Management

- Traffic Management มีหลายวิธี ที่สำคัญมีดังนี้
 - Leaky Bucket
 - Token Bucket
 - WRR-Weight Round Robin: กำหนดจำนวน Packet ที่ส่งในแต่ละ Queue ในแต่ละครั้ง โดยการกำหนดน้ำหนักให้กับแต่ละ Queue และทำ Round Robin วน
 - DRR-Deficit Round Robin: กำหนดจำนวน Byte ที่แต่ละ Queue จะส่งได้ในแต่ละครั้ง และทำ Round Robin



Leaky Bucket

Water can be
added
intermittently

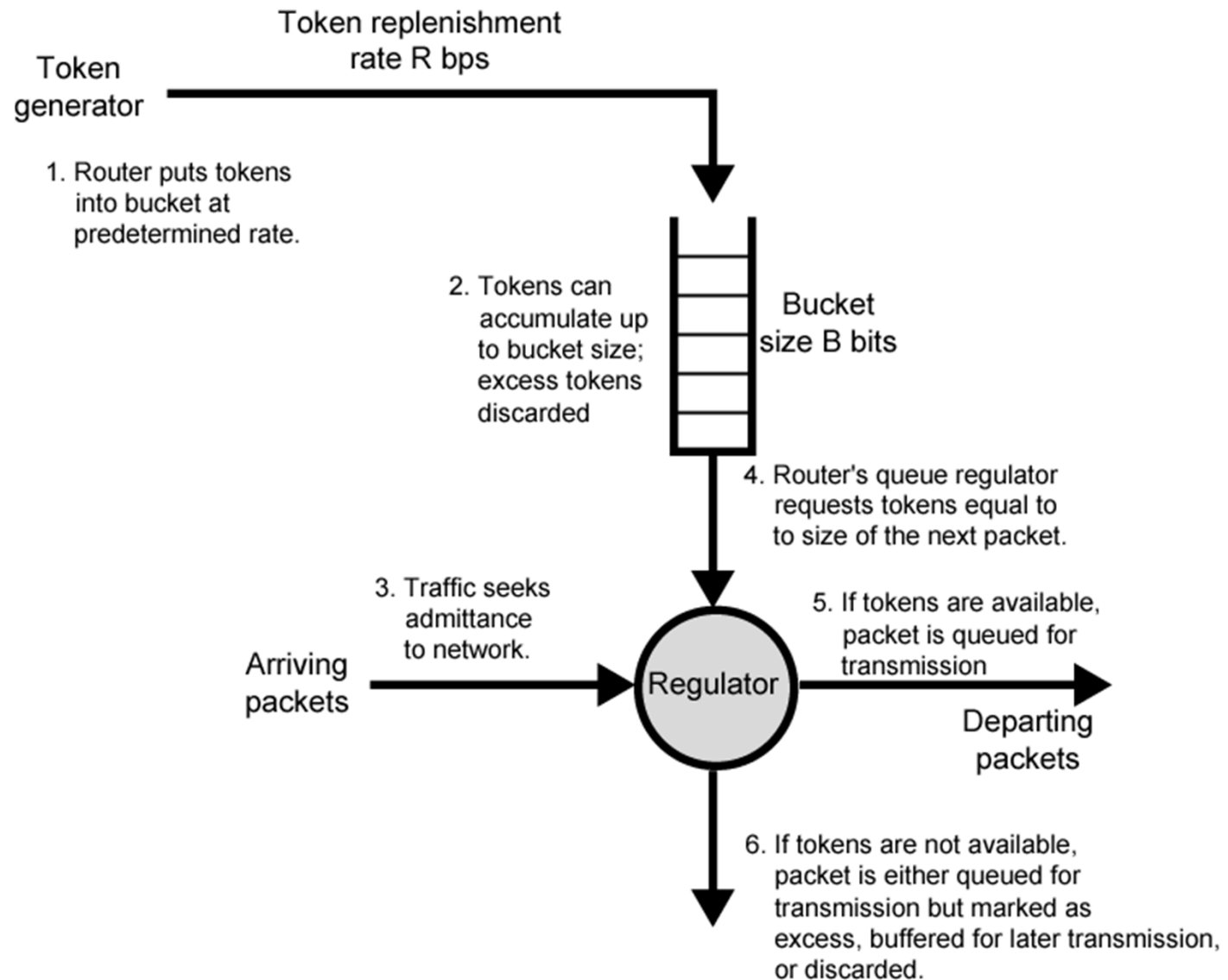
Overflows
when full

Leaks out at a
constant rate
until empty





Token Bucket





Internet QoS Technologies

■ RSVP และ COPS

- RSVP: Resource Reservation Protocol เป็น Fine-Grain โดยต้องกำหนดในแต่ละ Session ของ TCP/UDP ซึ่ง Application จะส่ง Request มาก่อน และ Request จะถูกส่งผ่าน Router ซึ่งจะมีการจอง Resource เอาไว้จนกระทั่งถึงปลายทาง ถ้าทุกๆ Hop ต้อนรับในการให้ Request และมีการจอง Resource จึงจะมีการกำหนด Flow ID และส่งข้อมูลได้
- COPS: Common Open Policy Service เป็น Protocol ที่ใช้ร่วมกับ RSVP ที่จะควบคุม Policy
- RSVP มักจะไม่ค่อยเห็นใช้งาน เพราะเป็น Fine-Grain ซึ่งมีการทำ QoS ในระดับ Flow



Internet QoS Technologies

■ DiffServ: Differentiated Service

- เป็น Coarse-Grain QoS โดยมีการกำหนดการแบ่ง Class ในส่วนของ Field 'Type-of-Service' ใน IPv4 และ 'Traffic Class' ใน IPv6
- ผู้ใช้ส่งข้อมูลโดยกำหนด Class ที่ตัวเองต้องการในส่วนนี้
- ชนิดของ Class จะเป็นตัวกำหนด Traffic Management ที่ตัว Router อีกที
- ยังไม่เป็นที่แพร่หลายมากนัก



Internet QoS Technologies

■ MPLS: MultiProtocol Label Switching

- เป็น Mechanism แบบ Connection-Oriented ที่สร้างขึ้นมาสวมบน IP อีกทีหนึ่ง
- ในการใช้งาน ผู้ดูแลจะกำหนดเส้นทางส่งข้อมูลผ่าน Router ที่ทำ MPLS ได้
- จากนั้นตัว Datagram ที่ส่งจะถูกแปะด้วย MPLS Header และส่งไปตามเส้นทาง เมื่อถึงปลายทาง ส่วน MPLS Header จะถูกนำออก
- ในแต่ละเส้นทางที่ส่ง จะมีการกำหนดค่า QoS Parameter ต่างกัน ดังนั้น Datagram จะใช้เส้นทางตามความเหมาะสมที่จัดตั้งโดย ISP และใช้ Label ที่เหมาะสม
- MPLS Packet จะถูก Switch ในระดับ Layer 2 และจะเร็วกว่า
- เป็นวิธีที่นิยมสำหรับ ISP ที่จะให้บริการแก่ลูกค้าในปัจจุบัน



BREAKS

- **After Break: Chapter 29**
 - IP Telephony



Chapter 29: Multimedia and IP Telephony (VoIP)

- ในการส่งข้อมูลแบบ **Real-time** จะต้องคำนึงถึงค่า **Jitter** ใน **Network**
 - อาจจะใช้ **Isochronous Infrastructure**
 - อาจจะใช้การทำงานของ **Protocol**
- เราจะกล่าวถึงการส่งข้อมูลแบบ **Multimedia** ผ่าน **Best-Effort Network** ได้อย่างไร
- จากนั้นจะกล่าวถึง **Technology** ของ **VoIP**
 - จะเน้นเฉพาะเรื่องของ **SIP** เพราะมีการใช้งานแพร่หลายมากกว่า



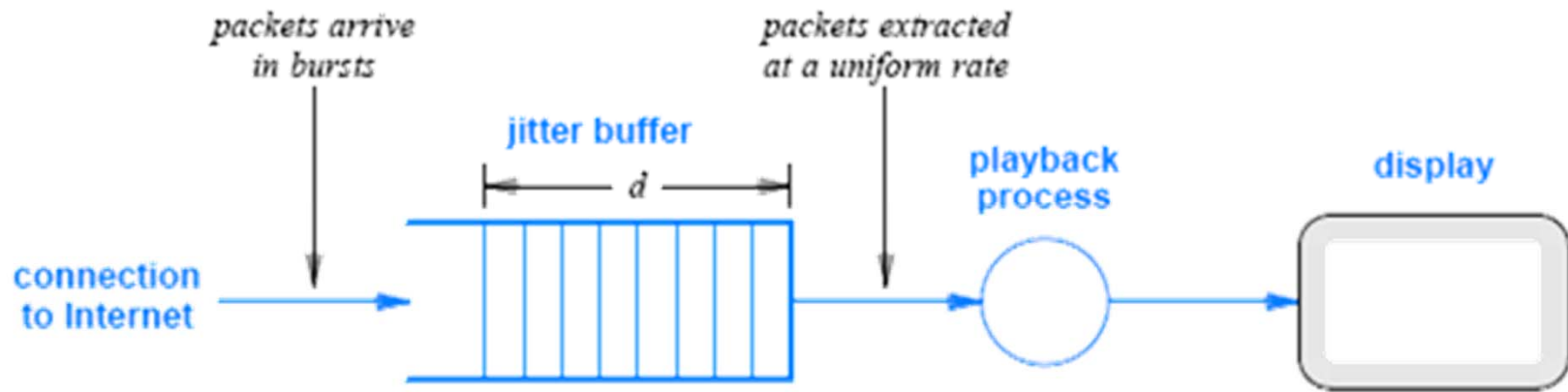
Real-Time Data Transmission and Best-Effort Delivery

- **Multimedia** หมายถึงข้อมูลที่ประกอบด้วยทั้ง **Voice** และ **Video**
- คำว่า **Real-Time Multimedia** หมายถึงข้อมูล **Multimedia** ที่จะต้องแสดงในอัตราที่เท่ากันกับอัตราของการส่งข้อมูล (หรืออัตราที่ข้อมูลถูกบันทึก)
- **Internet** เป็น **Best-Effort Delivery** ที่มีทั้ง **Lost, Delay, Out-of-Order** จะสามารถส่งข้อมูลพวกนี้ได้
อย่างไร
 - การ **Retransmission** จะใช้ไม่ได้ เพราะมันจะไปถึงช้าเกิน
- ใน **Internet** จะกระทำการส่งโดยใช้ **Protocol Support**



Delayed Playback and Jitter Buffer

- ในการจัดการกับ Jitter และเพื่อให้การแสดงผลราบเรียบ จะใช้สองเทคนิคดังนี้
 - Timestamps: ผู้ส่งจะประทับเวลาสำหรับข้อมูลแต่ละชิ้นที่ส่ง ผู้รับจะใช้ค่า Timestamp นั้นจัดการกับเรื่อง Out-of-Order Packet และแสดงข้อมูลตามเวลาที่กำหนด
 - Jitter Buffer: ในการจัดการกับ Jitter จะนำข้อมูลที่ได้รับมาใส่ใน Buffer ก่อน และจะมีการหน่วงเวลาในการแสดงผล





Real-Time Transport Protocol

- **RTP เป็น Mechanism ที่ถูกใช้สำหรับส่ง Real-Time Data ผ่าน Internet**
 - จริงๆแล้วไม่ใช่ Transport Protocol เพราะมันจะวางอยู่บน Transport Protocol อีกทีหนึ่ง (ปกติจะเป็น UDP)
- **RTP ไม่ได้ Guarantee เรื่องการส่งข้อมูลที่เป็นไปตามเวลา และตัว Protocol ไม่มีการ Implement Jitter Buffer แต่มันช่วยให้ผู้รับสามารถสร้าง Jitter Buffer ได้ โดยให้ข้อมูลสามอย่าง**
 - Sequence Number: เพื่อตรวจจับ Packet ที่สูญหาย
 - Timestamp: เพื่อให้ผู้รับแสดงผลได้ตามเวลาที่ถูกต้อง
 - ชุดของ Source Identifier: บ่งบอก Source ของข้อมูลแก่ผู้รับ



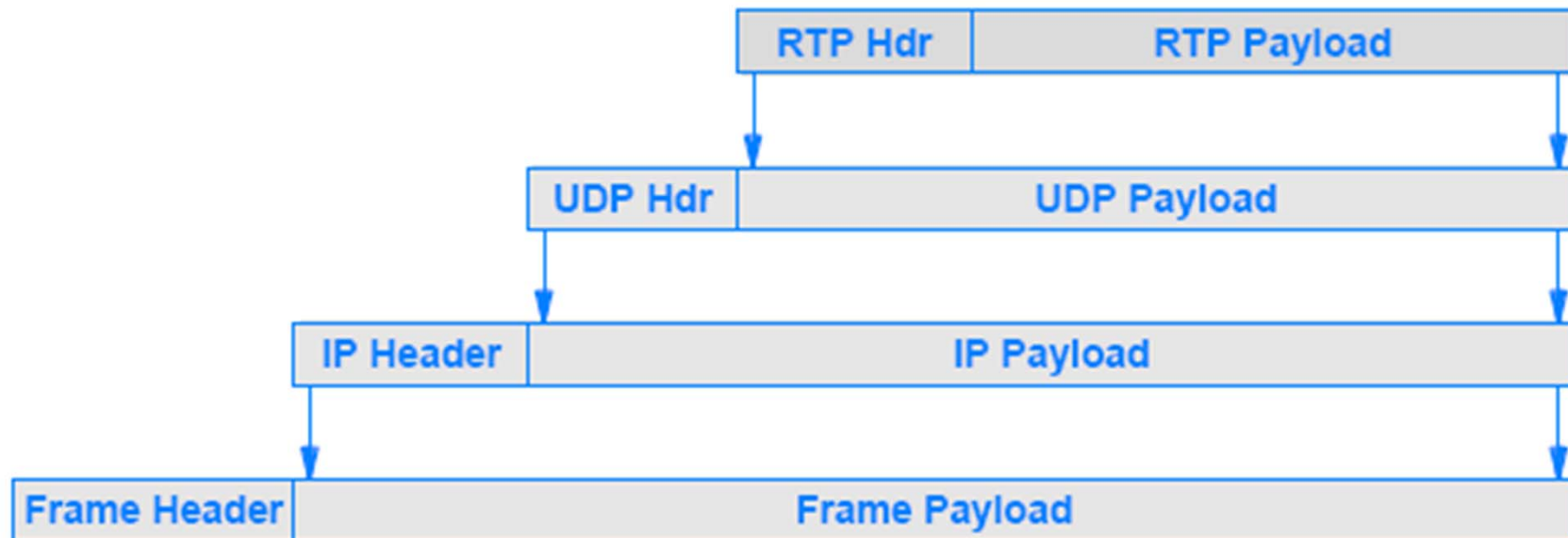
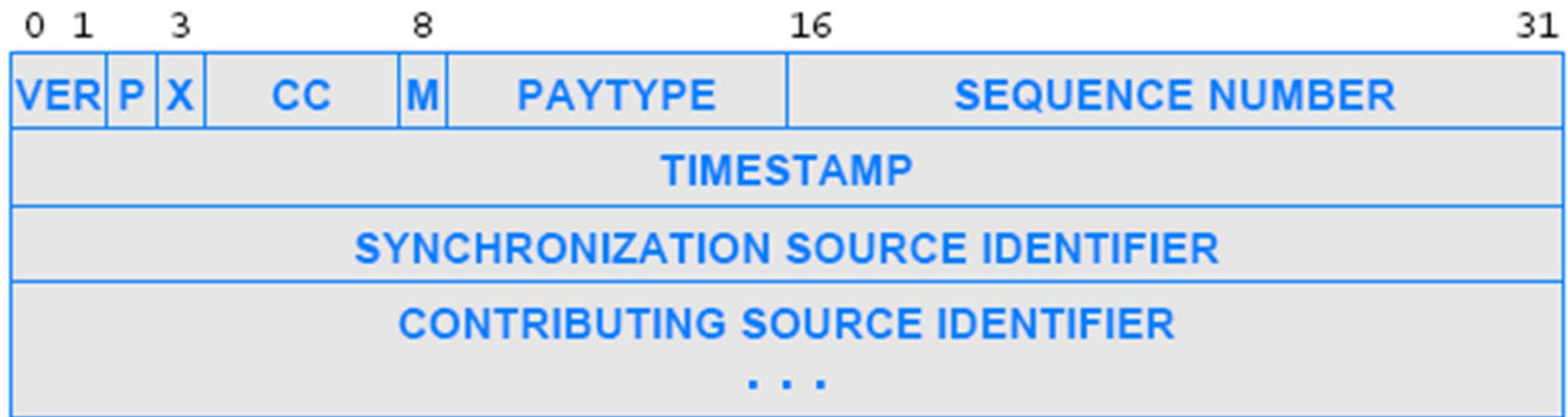
Real-Time Transport Protocol

■ RTP Basic Header

- Ver ปัจจุบันคือ 2
 - P=มี Zero Padded in Payload หรือไม่, X=Extension Header Exist, CC = Source Count, M = Mark Bit สำหรับ Mark บาง Frame
 - PAYTYPE: Payload Type
 - Sequence ใช้ Random Number และจะ Increment ที่ละหนึ่งในแต่ละ Packet
 - Timestamp จะเริ่มจาก Random Time เช่นกัน และจะไม่กำหนดหน่วยเวลา ขึ้นอยู่กับ Payload Type
 - Synchronization Source Identifier และ Contributing Source Identifier จะบ่งบอก Source ของข้อมูล
- ## ■ ปกติ RTP จะถูกบรรจุใน UDP และอาจจะถูกส่งแบบ Broadcast หรือ Multicast (หรือ Unicast)



Real-Time Transport Protocol





IP Telephony

- **IP Telephony หรือ Voice over IP (VoIP) เป็น Application ที่เกี่ยวข้องกับ Multimedia ที่สำคัญที่สุดตัวหนึ่ง**
- **ปัจจุบัน การสื่อสารโทรศัพท์เริ่มจะเปลี่ยนจากการใช้ Telephone Switch มาเป็น IP Router**
- **หลักการคือ เปลี่ยนเสียงพูดที่เป็น Analog ให้เป็น Digital จากนั้นส่ง Digital Stream ผ่าน IP Network และที่ปลายทางเปลี่ยนสัญญาณ Digital กลับเป็น Analog ตามเดิม**



IP Telephony

- รายละเอียดของ IP Telephony หรือ Voice over IP (VoIP) นั้นค่อนข้างจะซับซ้อน
 - ปัญหา Delay และ Jitter ใน Network
 - การจัดการเกี่ยวกับ Call Setup
 - การแปลง Telephone Number เป็น IP Address
 - การค้นหาตำแหน่งของผู้รับ
 - สัญญาณควบคุมต่างๆ เช่นการเชื่อมต่อ การส่ง Ringing Signal การทำ Call Forwarding การบันทึกการใช้งาน หรือการจบการสื่อสาร
 - ส่วนที่ซับซ้อนที่สุดคือ ระบบ VoIP จะต้อง Backward Compatible กับระบบโทรศัพท์เดิม (Public Switching Telephone Network, PSTN)
 - โทรศัพท์ที่ต่อกับ IP ต้องสามารถโทรหรือรับสายจากโทรศัพท์ในเครือข่าย PSTN รวมถึงการรองรับ Feature อื่นๆ เช่น Call Forwarding, Call Waiting, Conference Call และ Caller ID
 - นอกจากนี้แล้ว องค์กรที่มีชุมสายเป็นของตัวเอง (PBX) อาจจะต้องการ IP Phone ที่ให้ Service เหมือนกับระบบที่ตนเองมีอยู่



- **ITU (International Telecommunication Union)**
ผู้รับผิดชอบมาตรฐานโทรศัพท์ ได้ออกมาตรฐานของ IP Phone ที่สามารถใช้กับระบบโทรศัพท์ได้
- **IETF (Internet Engineering Task Force)**
ผู้รับผิดชอบมาตรฐาน TCP/IP ได้ออกมาตรฐานออกมาเช่นกัน
- ทั้งสองมาตรฐานแตกต่างกัน แต่มีที่เหมือนกันคือ
 - Audio จะใช้การ Encode ด้วย PCM (Pulse Coded Modulation)
 - RTP จะถูกใช้ในการส่ง Digitized Audio
- ปัญหาที่ทำให้ VoIP ช้าขึ้นคือเรื่องของการทำ Call Setup และ Call Management ที่เรียก Signaling
 - Signaling มาตรฐานปัจจุบันคือ Signaling System 7 (SS7)
 - IETF ออก Session Initiation Protocol (SIP) และ Media Gateway Control Protocol (MGCP)
 - ITU ออก H.323 ออกมา
 - นอกจากนี้ยังมี H.248 (Megaco=Media Gateway Control Protocol) ซึ่งเป็น Protocol รวมของทั้งสองกลุ่ม



ส่วนประกอบของ IP Telephone System

- **IP Telephone**

- ทำงานเหมือนกับโทรศัพท์ทั่วไป แต่เชื่อมต่อกับ Internet และส่ง Digitized Voice

- **Media Gateway Controller (Gatekeeper, Softswitch)**

- ทำการควบคุมและประสานงานระหว่าง IP Telephone สำหรับ Service ต่างๆเช่น Call Setup, Call Termination, Call Forwarding และการหาตำแหน่ง
- ควบคุมการทำงานของ Media Gateway และ Signaling Gateway

- **Media Gateway**

- ทำหน้าที่เชื่อมต่อระหว่างสอง Network ที่ใช้การ Encoding ที่ไม่เหมือนกัน โดยการทำให้ Translation Audio Encoding ระหว่าง NW

- **Signaling Gateway**

- ดูแลการเชื่อมต่อระหว่างสอง Network ที่ใช้ Signaling ต่างกัน และทำการ Translate Call Management



ส่วนประกอบของ IP Telephone System

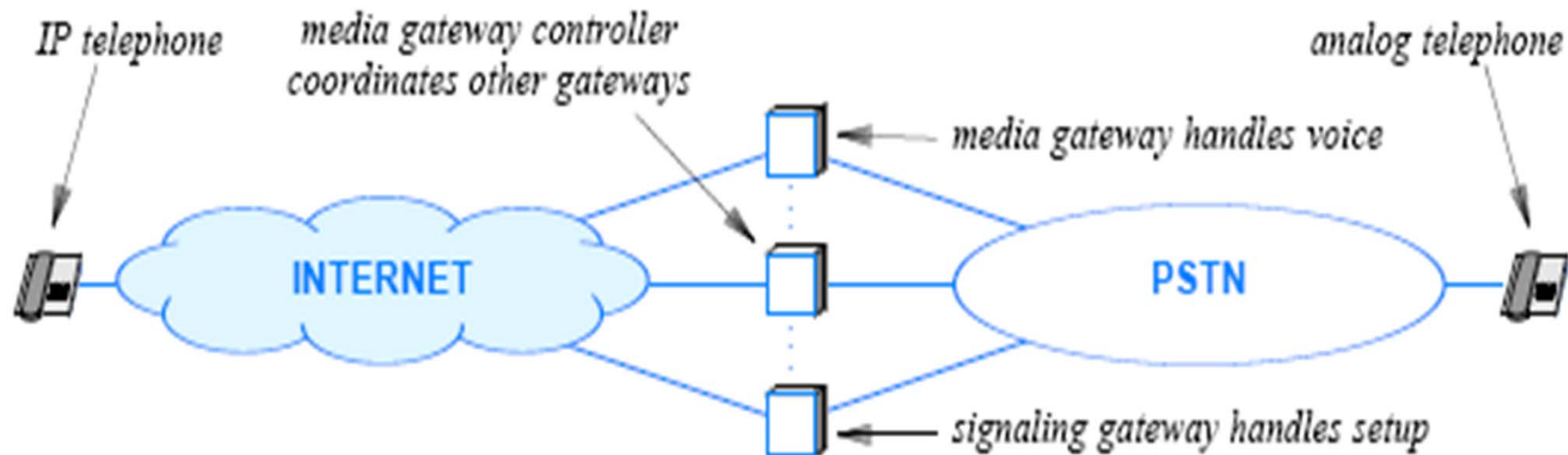


Figure 29.5 Connections among IP telephone components.



H.323 Terminology and Concept

- **มาตรฐานของ H.323 กว้างมาก ประกอบเป็นชุดของ Protocol ที่สามารถรองรับได้ทั้ง Voice และ Video**
 - **Terminal**
 - H.323 Terminal จะให้ Function ของ IP Telephone ซึ่งอาจจะรวมถึงอุปกรณ์ที่ใช้ในการส่ง Video
 - **Gatekeeper**
 - H.323 Gatekeeper จะทำหน้าที่หาตำแหน่ง และทำ Signaling และประสานงานกับ Gateway ที่เชื่อมต่อกับ PSTN
 - **Gateway**
 - H.323 ใช้เพียง Gateway เดียวซึ่งจะดูแลทั้ง Signaling และ Media Translation
 - **Multipoint Control Unit (MCU)**
 - ให้ Service เช่นใน Multipoint Conferencing (Video/Audio)



H.323 Layering

Layer	Signaling	Registration	Audio	Video	Data	Security
5	H.225.0-Q.931 H.250-Annex G H.245 H.250	H.225.9-RAS	G.711 H.263 G.722 G.723 G.728	H.261 H.323	T.120	H.235
			RTP, RTCP			
4	TCP, UDP	UDP			TCP	TCP, UDP
3	IP, RSVP, and IGMP					



SIP Terminology and Concept

- หลักการของ SIP คือพยายามใช้ Protocol ที่มีอยู่แล้วให้มากที่สุด เช่น การใช้ DNS ในการ MAP ระหว่าง หมายเลขโทรศัพท์และ IP Address
- SIP กำหนด Element สามส่วน
 - User Agent
 - หมายถึงอุปกรณ์ที่ Initiate หรือ Terminate Phone Call อาจจะเป็น IP Phone, Laptop, Computer หรือ PSTN Gateway
 - User Agent ประกอบด้วยสองส่วน
 - User Agent Client คือผู้ทำหน้าที่โทรออก
 - User Agent Server จะทำหน้าที่รับสายเข้า



SIP Terminology and Concept

- **SIP กำหนด Element ใน VoIP แบ่งเป็นสามส่วน**
 - User Agent
 - Location Server
 - ทำหน้าที่ในการจัดการฐานข้อมูลเกี่ยวกับผู้ใช้ เช่น IP Address, Service ต่างๆที่ผู้ใช้สมัคร โดยข้อมูลนี้จะถูกส่งเมื่อทำ Call Setup
 - Support Servers
 - Proxy Server; เป็น Proxy สำหรับการส่ง Request ของ User ไปยัง Location อื่น รวมถึง Optimum Routing ไปยัง Location นั้นๆ และกำดูแลเรื่อง Policy
 - Redirect Server; ดูแลจัดการเรื่องการทำ Call Forwarding และหมายเลข 1-800 โดยมันจะส่งค่า Alternate Location กลับไปให้ User
 - Registrar Server; ใช้ดูแลจัดการเรื่องการลงทะเบียนของผู้ใช้ รวมถึงการทำ Authentication และ Update ฐานข้อมูลของ Location Server



SIP Characteristics and Methods

- **คุณสมบัติของ SIP ประกอบด้วย**
 - จะ Run ในระดับ Application Layer
 - รวมหน้าที่ของ Signaling ต่างๆเข้าไว้ด้วยกัน
 - ให้บริการเสริมอื่นๆเช่น Call Forwarding
 - ใช้วิธีของ Multicasting ในการทำ Conference Call
 - ยอมให้ผู้ใช้ต้นทางและปลายทางสามารถ Negotiate และเลือก Parameter ในการเชื่อมต่อที่เหมาะสม
- **แต่ละ User จะอ้างถึงด้วย SIP URI (Uniform Resource Identification) ประกอบด้วยชื่อ และ Domain Name**
 - เช่น sip:smith@somecompany.com

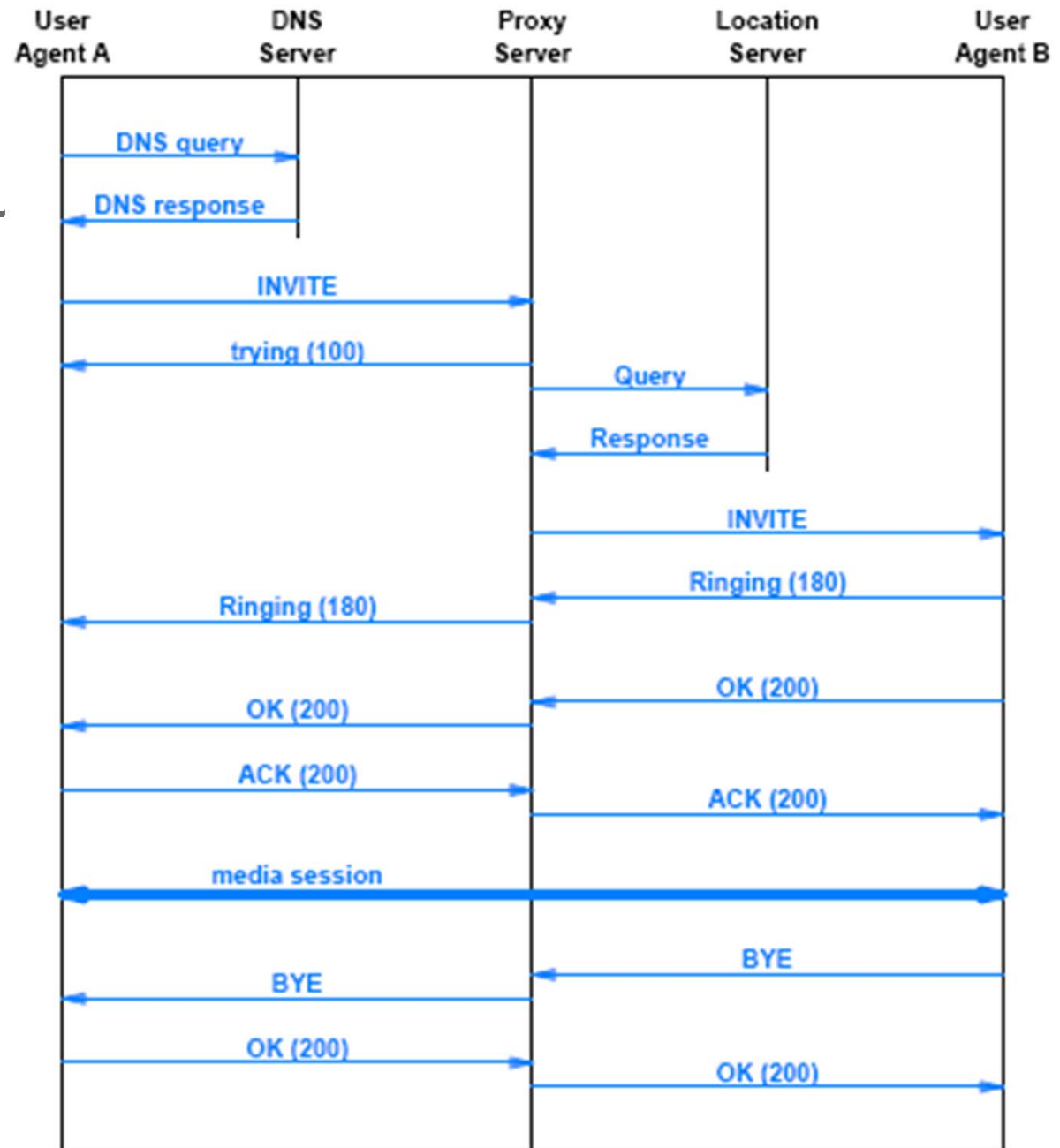


SIP Basic Message Types

- **ประกอบด้วย 6 Basic Message Types**
 - Basic Message Type นี้เรียก Method
 - **Invite:** ใช้ในการสร้าง Session โดยทำส่ง Invite กับ End Point ให้เข้าร่วมใน Session
 - **ACK:** Acknowledge Response สำหรับ Invite
 - **BYE:** จบ Session และจบ Call
 - **CANCEL:** ในทำการ Cancel Request ที่ยังค้างอยู่
 - **REGISTER:** ใช้สำหรับการขึ้นทะเบียนตำแหน่งของผู้ใช้
 - **OPTIONS:** ใช้ร้องขอข้อมูลของอีกฝั่งหนึ่งว่ามีความสามารถอะไรบ้าง



ตัวอย่าง ของ SIP Session





Telephone Number Mapping and Routing

- เราจะกำหนดชื่อ และหาตำแหน่ง IP Phone แต่ละเครื่องได้อย่างไร
- Telephone Number ใน PSTN จะใช้มาตรฐานของ E.164
- SIP จะใช้ IP Address
- IETF เสนอสอง Protocol ในการ Mapping
 - ENUM (E.164 NUMbers) ใช้เพื่อการ Convert หมายเลขโทรศัพท์ให้อยู่ในรูป URI
 - TRIP (Telephone Routing Over IP) เป็น Protocol ที่ใช้หาตำแหน่ง User ใน Network รวม (PSTN+IP)



Telephone Number Mapping and Routing

- **IETF เสนอสอง Protocol ในการ Mapping**
 - ENUM (E.164 NUMbers) โดยทำการ Convert หมายเลขโทรศัพท์ให้อยู่ในรูป URI
 - ENUM จะใช้ Domain Name System ในการเก็บ Mapping โดยใช้ Special Domain 'e164.arpa'
 - Conversion จะมองหมายเลขโทรศัพท์ว่าเป็น String จากนั้นจะทำการ Reverse String
 - เช่นหมายเลข 1-800-555-1234 จะมี URI เป็น 4.3.2.1.5.5.5.0.0.8.1.e164.arpa
 - การ Map อาจจะเป็น 1-to-1 หรือ 1-to-Many
 - TRIP (Telephone Routing Over IP) เป็น Protocol ที่ใช้หาตำแหน่ง User ใน Network รวม (PSTN+IP)
 - ใช้สำหรับ Location Server หรืออุปกรณ์อื่นทำการ Advertise Route ที่ตัวเองรู้ออกไป
 - TRIP ใช้ Concept ของการแบ่งกลุ่มผู้ใช้ทั้งหมดออกเป็น ITAD (IP Telephone Administration Domain)



End of Chapter 28-29 (Week 15)

- **HW 9 Download**