Transporting Voice by Using IP

National Chi Nan University

Quincy Wu

Email: solomon@ipv6.club.tw

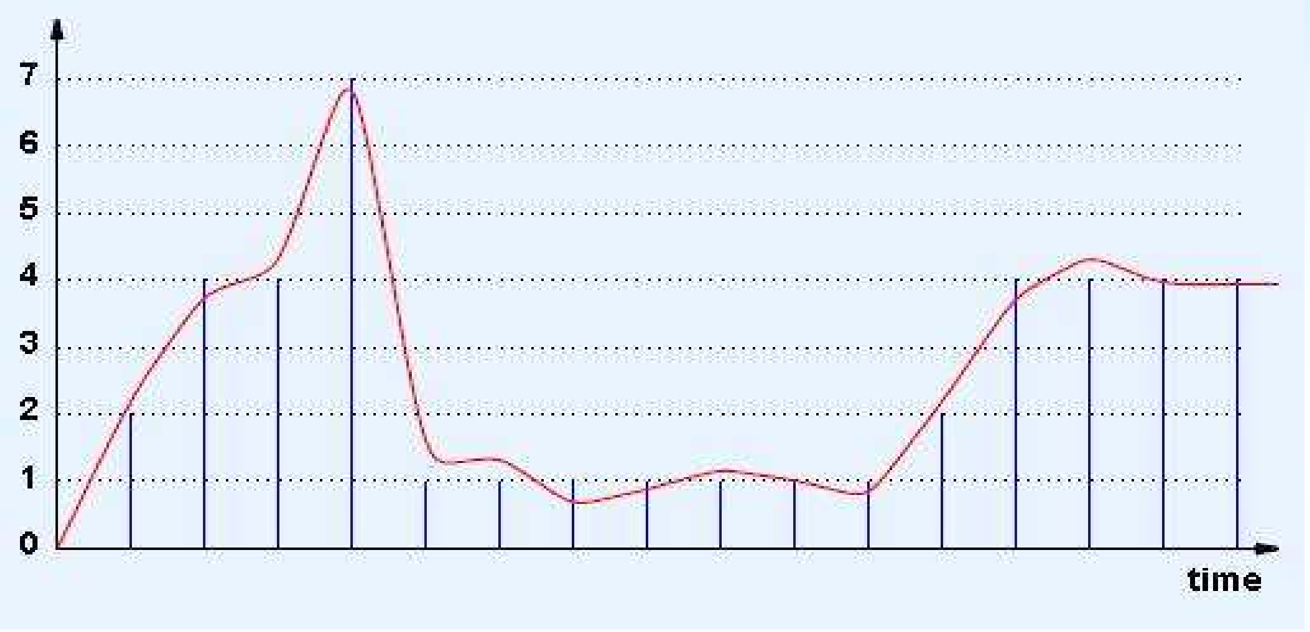
Outline

- Introduction
- Voice over IP
- RTP & SIP
- Conclusion

Digital Circuit Technology

- Developed by telephone companies
- Motivation: *analog signals* degrade as they pass over copper wires
  - Amplifiers distort the signal slightly and introduces noise.
- Designed for use in voice systems since 1962
  - Analog audio from user's telephone converted to digital format
  - Digital format sent across network
  - Digital format converted back to analog audio

Illustration of Digitized Signal



- Pick nearest digital value for each sample
  - 0,2,4,4,7,1,1,1,1,1,1,1,2,4,4,4,4
- Telephone standard known as *Pulse Code Modulation (PCM)*

Transporting Voice over Internet

- Digitized Payload
  - G.711(PCM) u-law / A-law
  - G.729
- Transport-Layer Protocol
  - TCP
  - UDP
- Network-Layer Protocol
  - IP (IPv4 and IPv6)

VoIP

- Transport voice traffic using the Internet Protocol (IP)
- One of the greatest challenges to VoIP is voice quality.
- One of the keys to acceptable voice quality is bandwidth.
- Control and prioritize the access
- Internet: best-effort transfer
  - VoIP != Voice over Internet
  - The next generation Telcos
    - Access and bandwidth are better managed.

Data and Voice

- Data traffic
  - Asynchronous – can be delayed
  - Extremely error sensitive
- Voice traffic
  - Synchronous – the stringent delay requirements
  - More tolerant for errors
- IP is not for voice delivery.
- VoIP must
  - Meet all the requirements for traditional telephony
  - Offer new and attractive capabilities at a lower cost

Lower Bandwidth Requirements

– PSTN
  - G.711 - 64 kbps
  - Human speech frequency < 4K Hz
  - The Nyquist Theorem: 8000 samples per second
  - 8K * 8 bits

– Sophisticated coders
  - 32kbps, 16kbps, 8kbps, 6.3kbps, 5.3kbps
  - GSM – 13kbps
  - Save more bandwidth by silence-detection

– Traditional telephony networks can use coders, too.
  - But it is more difficult.

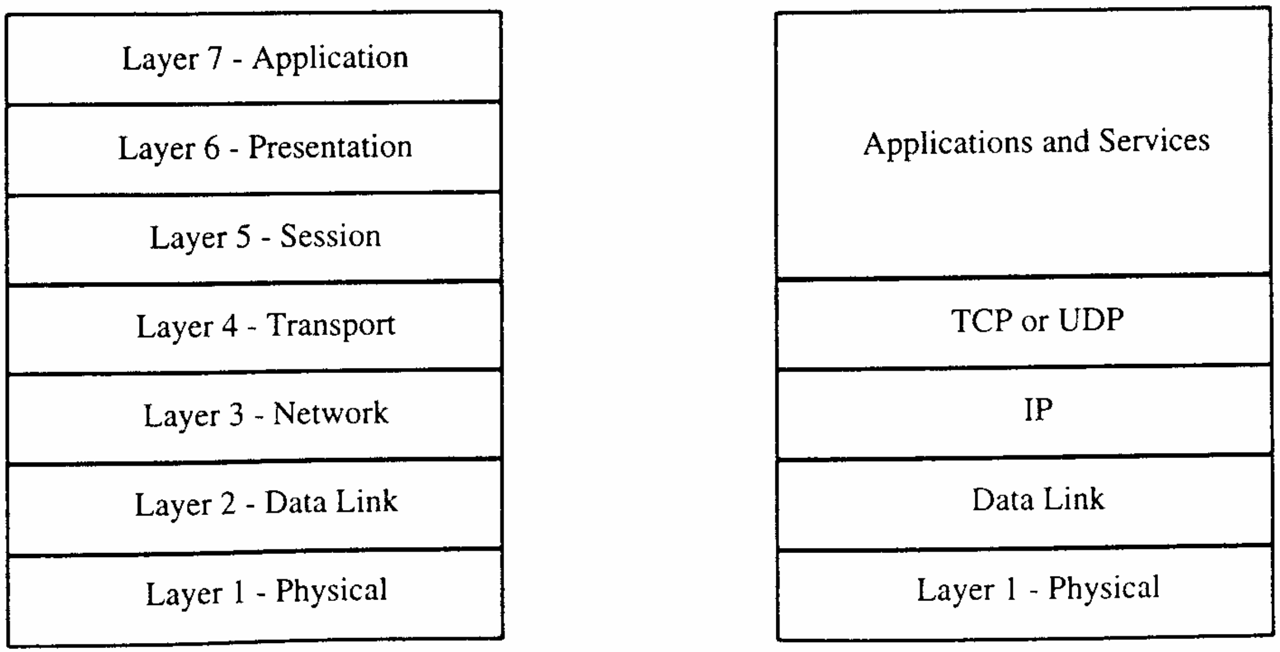
– VoIP – two ends of the call negotiate the coding scheme

Overview of the IP Protocol Suite

- IP
  - A routing protocol for the passing of data packets
  - Must work in cooperation with higher layer protocols and lower-layer transmission systems
- The OSI seven-layer model
  - The top layer: useable information to be passed to the other side
  - The information must be
    - Packaged appropriately
    - Routed correctly
    - And it must traverse some physical medium

The IP suite and the OSI stack

– TCP
  - Reliable, error-free, in-sequence delivery
– UDP
  - No sequencing, no retransmission



TCP/IP

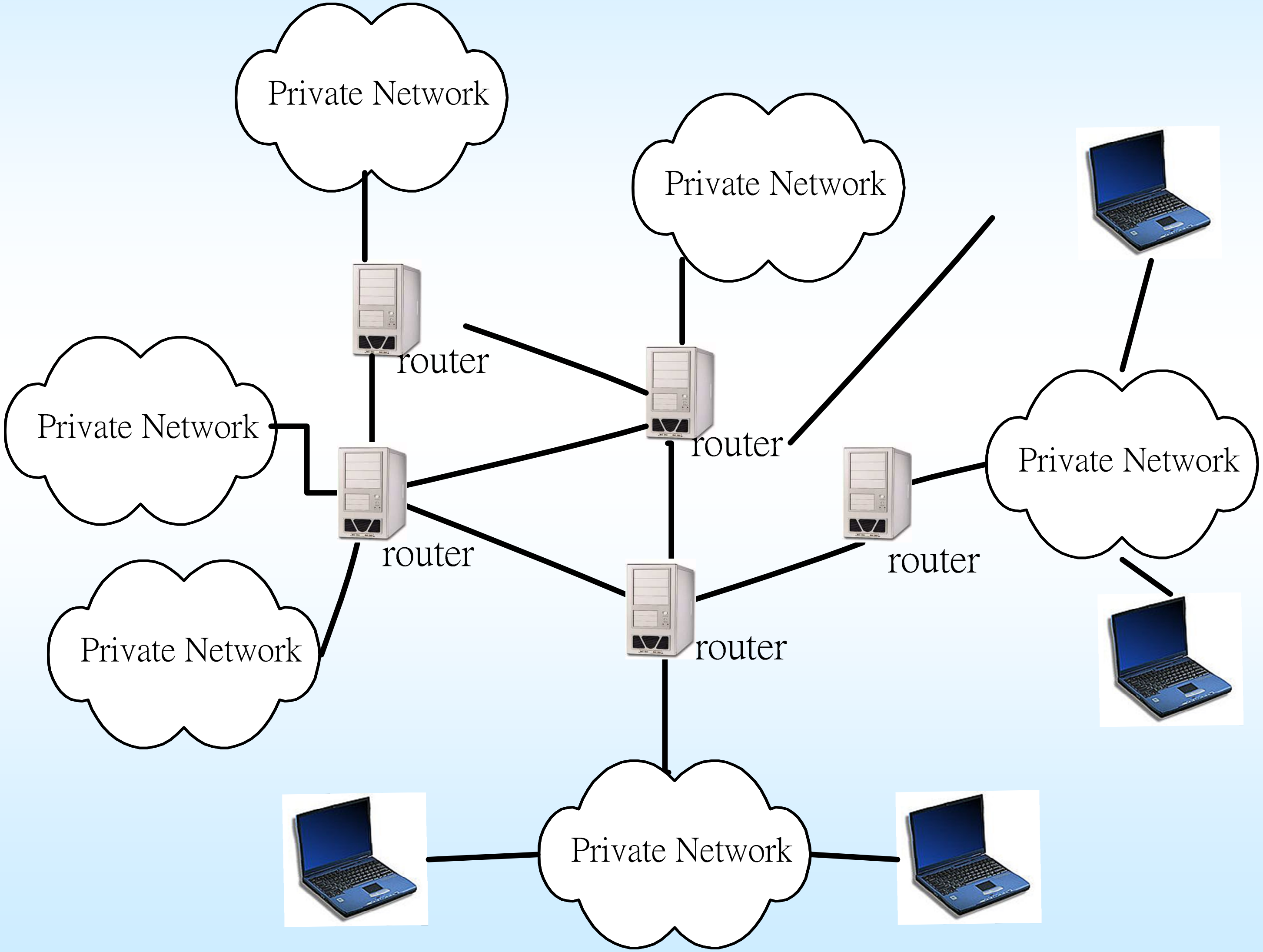
TCP/IP

- IP - A packet-based protocol
  - Routing on a packet-by-packet base
- Packet transfer with no guarantees
  - May not receive in order
  - May be lost or severely delayed
- TCP
  - Retransmission
  - Assemble the packets in order
  - Congestion control
  - Useful for file-transfers and e-mail

Internet Overview

- A collection of networks
  - The private networks
    - LANs, WANs
    - Institutions, corporations, business and government
    - May use various communication protocols
  - The public networks
    - ISP: Internet Service Providers
    - Using Internet Protocol
  - To connect to the Internet
    - Using IP

Interconnecting Networks



IP

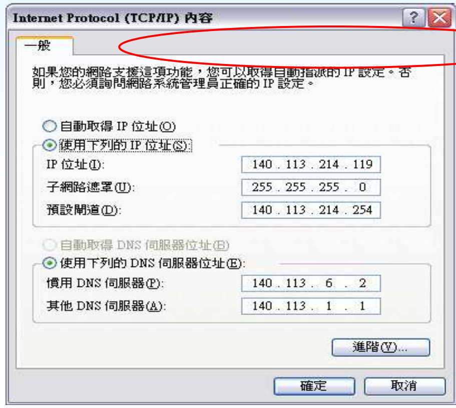
- RFC 791
  - Amendments: RFCs 950, 919, and 920
  - Requirements for Internet hosts: RFCs 1122, 1123
  - Requirements for IP routers: RFC 1812
  - IP datagram
    - Data packet with an IP header
  - Best-effort protocol
    - No guarantee that a given packet will be delivered

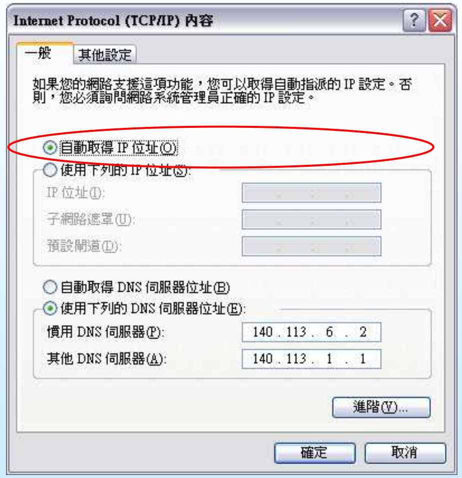
IP addresses: how to get one?

Q: How does *host* get IP address?

- Manually configuration
  - Wintel: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

(more shortly)





IP Header

- Protocol
  - The higher-layer protocol
  - TCP (6); UDP (17)
- Source and Destination IP Addresses

<table>
<tr><td>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="4">Version</td><td colspan="4">Header Length</td><td colspan="8">Type of Service</td><td colspan="16">Total Length</td></tr>
<tr><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td colspan="8">Time to Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td colspan="32">Source IP Address</td></tr>
<tr><td colspan="32">Destination IP Address</td></tr>
<tr><td colspan="32">Options</td></tr>
<tr><td colspan="32">Data</td></tr>
</table>

TCP

- Transmission Control Protocol
  - RFC 793
  - In sequence, without omissions and errors
  - End-to-end confirmation, packet retransmission, flow control, congestion control
  - The source retransmits if no ACK is received within a given period.
  - Applications: HTTP, FTP, TELNET, SMTP

UDP

- User Datagram Protocol
  - Pass individual pieces of data from an application to IP
  - No ACK, inherently unreliable
  - Applications
    - A quick, on-shot transmission of data, request/response
    - DNS (udp port 53)
    - If no response, the application retransmits the request
  - Checksum

<table>
<tr><td>00</td><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">Source Port</td><td colspan="16">Destination Port</td></tr>
<tr><td colspan="16">Length</td><td colspan="16">Checksum</td></tr>
</table>

Voice over UDP, not TCP

- Speech
  - Small packets, 10 – 40 ms
  - Occasional packet loss is not a catastrophe
  - Delay-sensitive
    - TCP: connection set-up, ack, retransmit → delays
  - 5 % packet loss is acceptable if evenly spaced
    - Resource management and reservation techniques
    - A managed IP network
  - In-sequence delivery
    - Mostly yes
- But, UDP was not designed for voice traffic

Real-time Transport Protocol
(RTP)

Real-Time Transport Protocol

- Disadvantage of UDP
  - Packets may be lost or out-of-sequence
- RTP: A Transport Protocol for Real-Time Applications
  - RFC 1889; RFC 3550
  - RTP – Real-Time Transport Protocol
  - RTCP – RTP Control Protocol
- RTP over UDP
  - A sequence number
  - A time stamp for synchronized play-out
  - Does not solve the problems; simply provides additional information

RTP Header Format

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V=2		P	X	CC				M	PT							Sequence Number															
Timestamp																															
Synchronization Source (SSRC) Identifier																															
Contributing Source (CSRC) Identifiers (0 to 15 entries)																															

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Profile-specific information																Length															
Header extension																															

The RTP Header

- Sequence number
  - A random number generated by the sender at the beginning of a session
  - Incremented by one for each RTP packet
- Timestamp
  - The receiver
    - Synchronized play-out
    - Calculate the jitter
    - Support silence suppression
    - The initial timestamp is a random number chosen by the sending application.
- Payload Type (PT)
  - In general, a single RTP packet will contain media coded according to only one payload format.
  - RED is an exception.

RTP Payload Formats [1/2]

- RTP carries the actual digitally encoded voice
  - RTP header + a payload of voice/video samples
  - UDP and IP headers are attached
- Many voice- and video-coding standards
  - A payload type identifier in the RTP header
    - Specified in RFC 1890
    - New coding schemes have become available
  - A sender has no idea what coding schemes a receiver could handle.
    - Negotiated by signaling protocols like SIP.

RTP Payload Formats [2/2]

- Separate signaling systems
  - Capability negotiation during the call setup
  - SIP and SDP
  - A dynamic payload type may be used
    - Support new coding scheme in the future
    - The encoding name is also significant.
      - Unambiguously refer to a particular payload specification
      - Should be registered with the IANA
- RED, Redundant payload type
  - Voice samples + previous samples
  - May use different encoding schemes
  - Cope with packet loss

Speech-coding Techniques

- In general, coding techniques are such that speech quality degrades as bandwidth reduces.
  - The relationship is not linear.
- G.711 64kbps 4.3
- G.726 32kbps 4.0
- G.723 (celp) 6.3kbps 3.8
- G.728 16kbps 3.9
- G.729 8kbps 4.0
- GSM 13kbps 3.7
- iLBC 13.3kbps 3.9

RTCP

- A companion protocol
- Exchange messages between session users
- # of lost packets, delay and inter-arrival jitter
- Quality feedback
- RTCP is implicitly open when an RTP session is open
- E.g., RTP/RTCP uses UDP port 5004/5005

Session Initiation Protocol (SIP)

Introduction

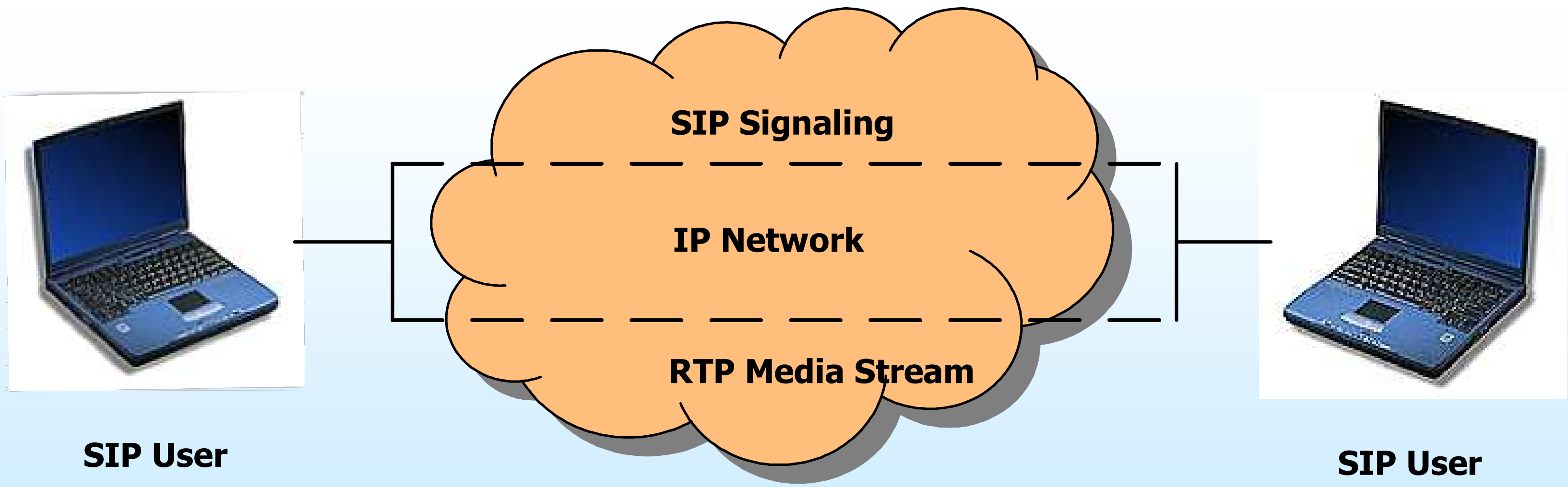
- A powerful alternative to H.323
- More flexible, simpler
- Easier to implement
  - Advanced features
- Better suited to the support of intelligent user devices
- A part of IETF multimedia data and control architecture

The Popularity of SIP

- Originally Developed in the MMUSIC (Multiparty Multimedia Session Control)
  - A separate SIP working group
  - RFC 2543
  - Many developers
  - The latest version: RFC 3261
- SIP + MGCP/MEGACO
  - The VoIP signaling in the future
- “bake-off” and SIPit activities
  - Various vendors come together and test their products against each other
    - to ensure that they have implemented the specification correctly
    - to ensure compatibility with other implementations

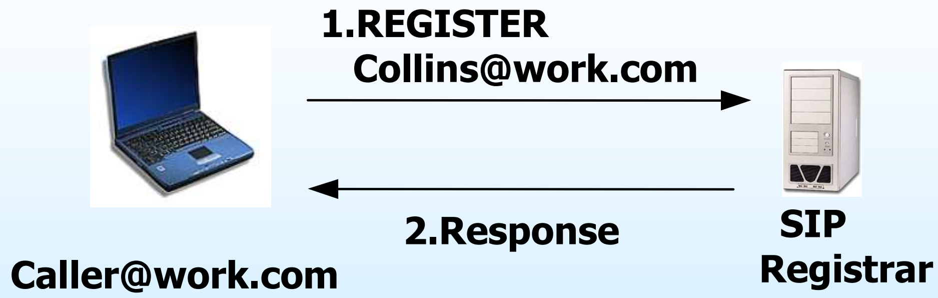
SIP Architecture

- A signaling protocol
  - The setup, modification, and tear-down of multimedia sessions
- SIP + SDP
  - Describe the session characteristics
- Separate signaling and media streams



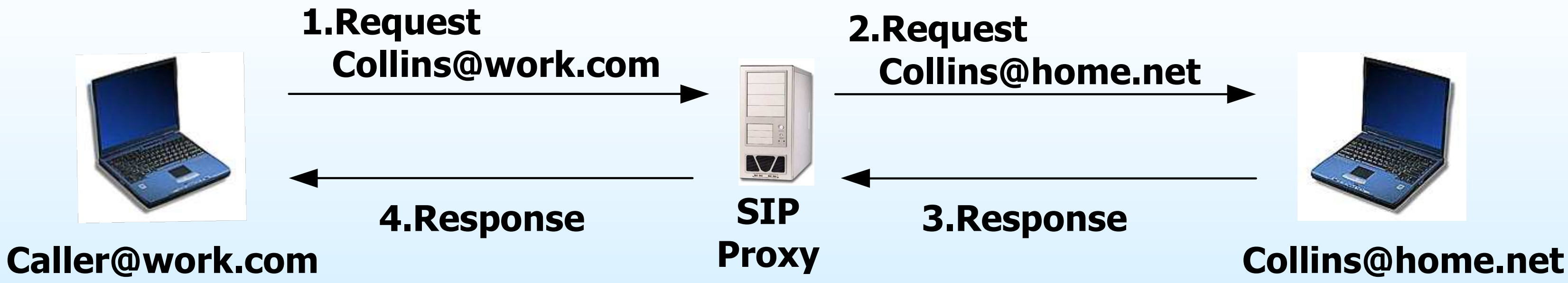
SIP Servers [1/3]

– Registrar
  - Accepts SIP REGISTER requests
    – Indicating that the user is at a particular address
    – Personal mobility
  - Typically combined with a proxy or redirect server



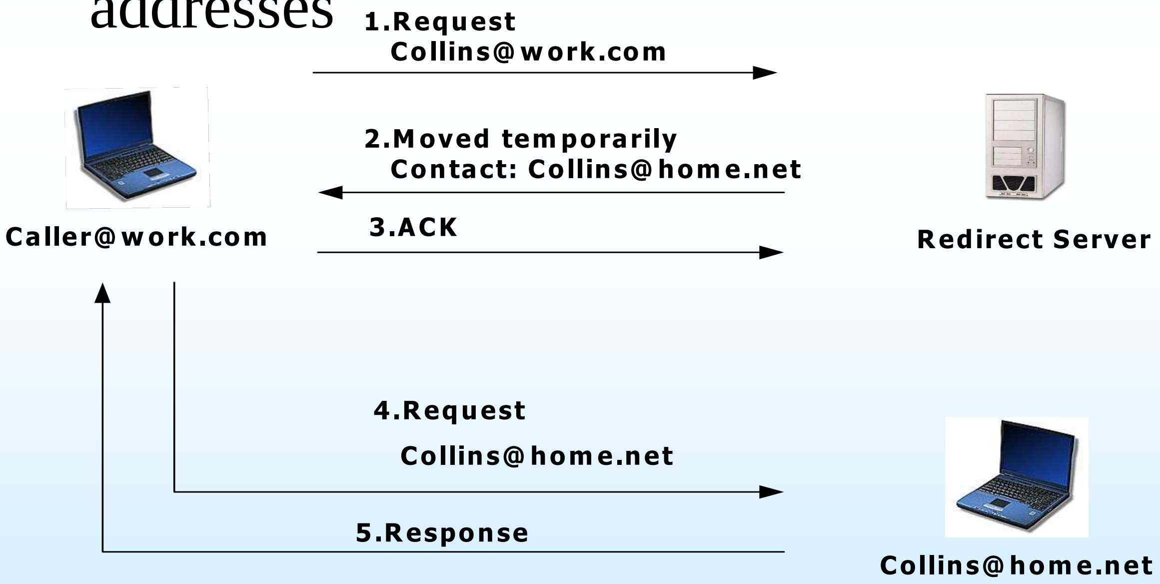
SIP Servers [2/3]

– Proxy servers
  - Handle requests or forward requests to other servers
  - Can be used for call forwarding, time-of-day routing, or follow-me services

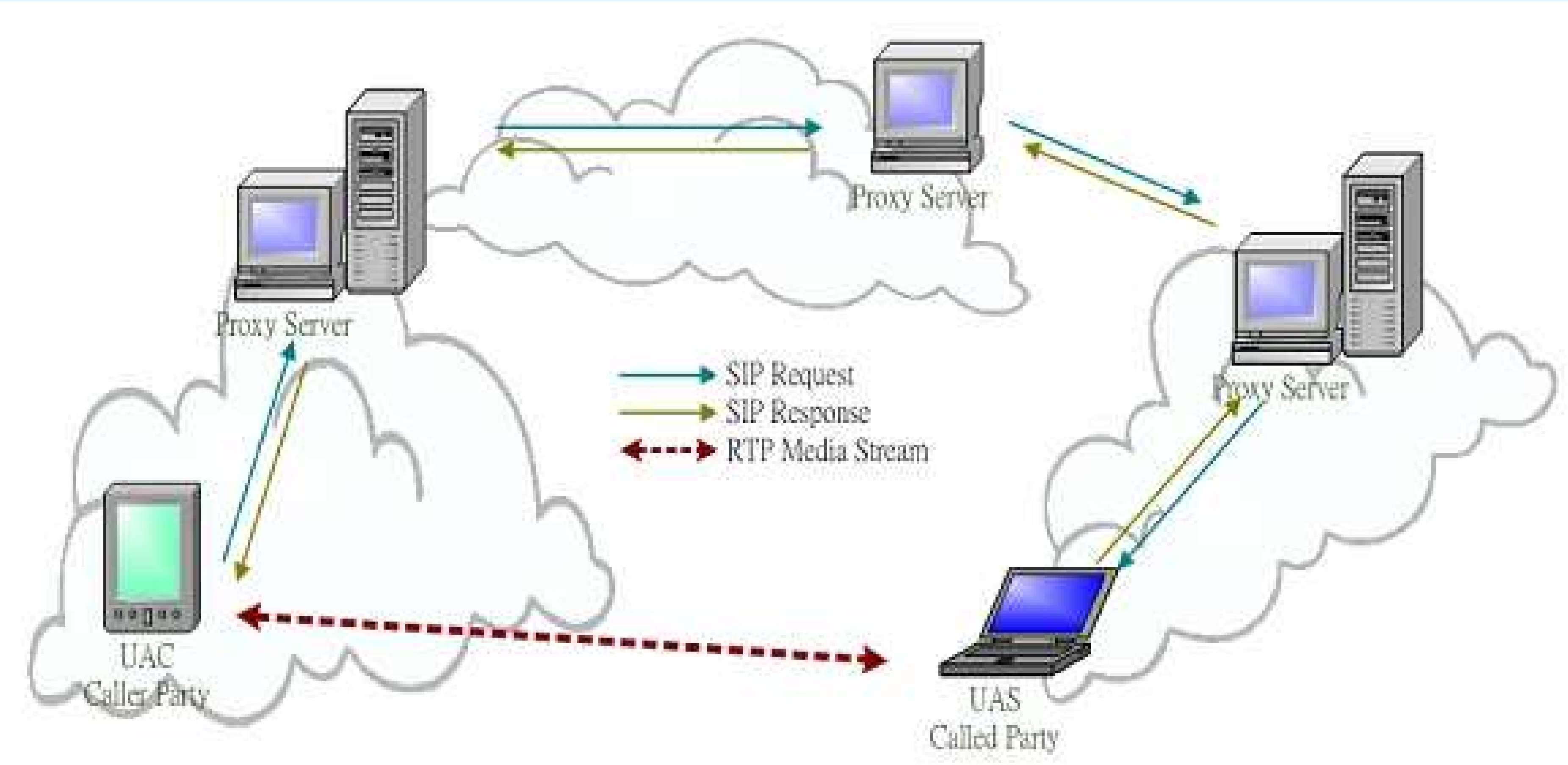


SIP Servers [3/3]

– Redirect servers
  - Map the destination address to zero or more new addresses

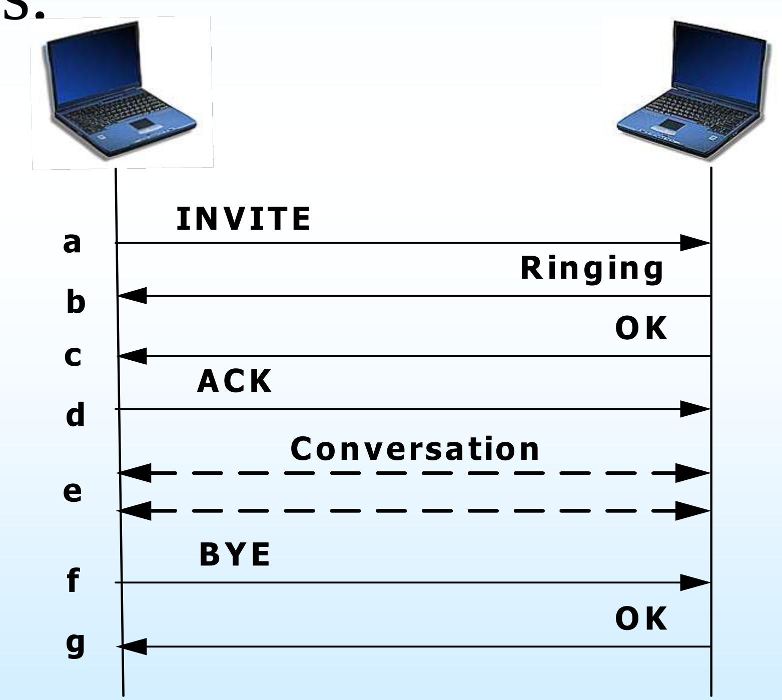


SIP Call Establishment [1/2]



SIP Call Establishment [2/2]

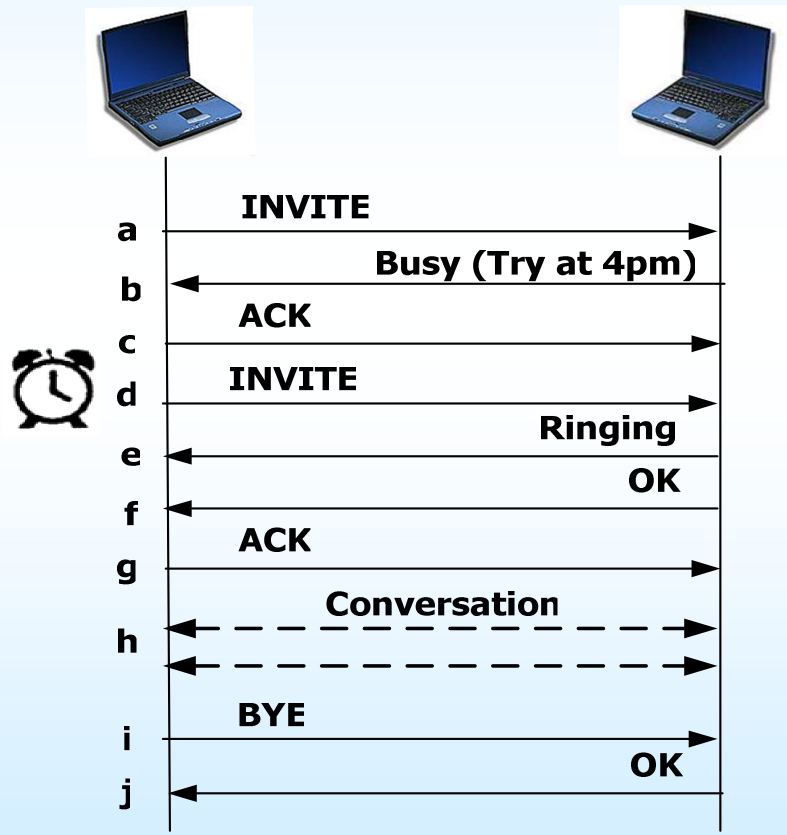
- It is simple, which contains a number of interim responses.



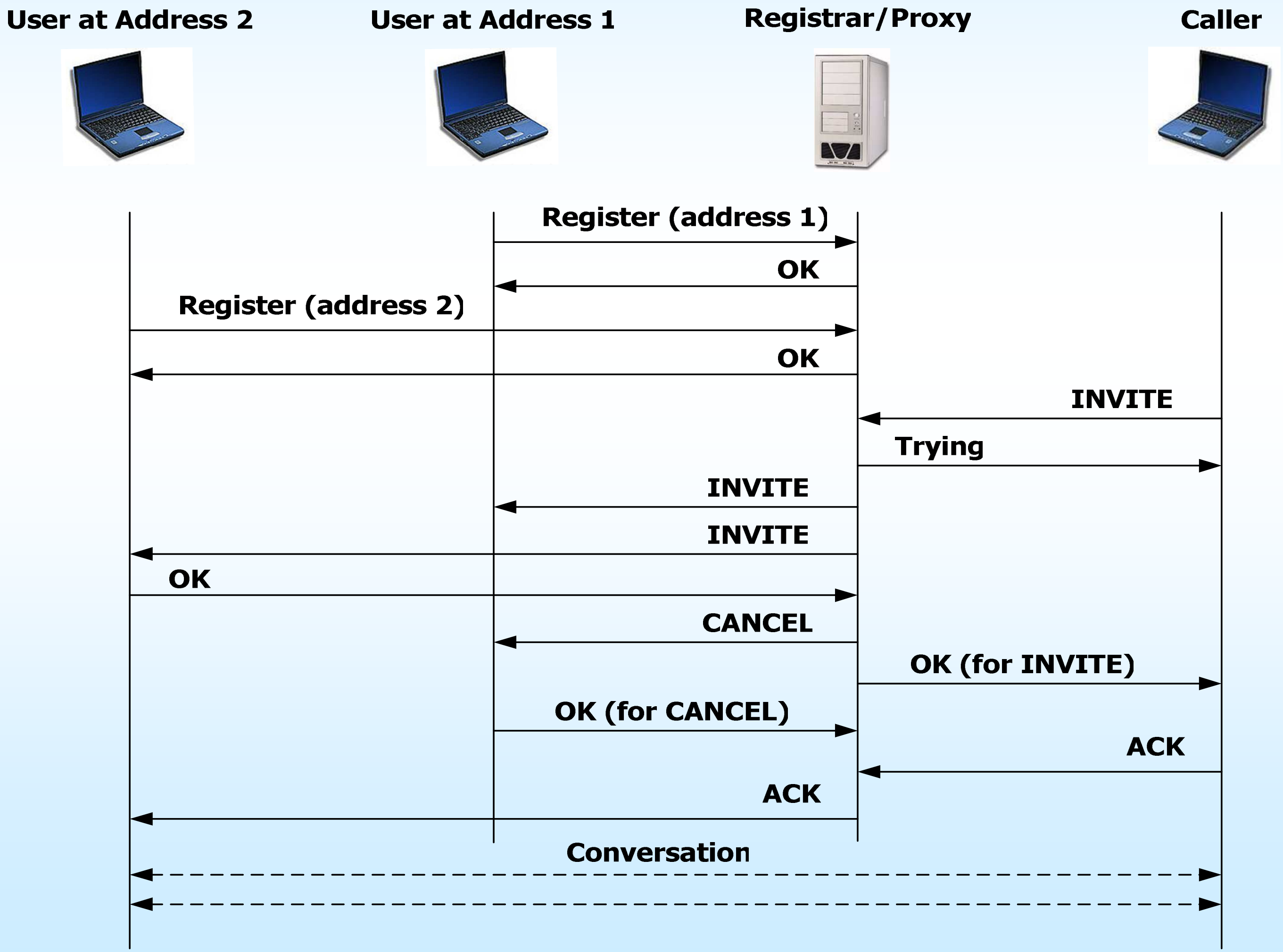
SIP Advantages

- Attempt to keep the signaling as simple as possible
- Offer a great deal of flexibility
  - Does not care what type of media is to be exchanged during a session or the type of transport to be used for the media
- Various pieces of information can be included within the messages
  - Including non-standard information
  - Enable the users to make intelligent decisions
    - The control of the intelligent features is placed in the hands of the customer, not the network operator.
  - E.g., SUBJECT header

Call Completion to A Busy User



“One number” service



Overview of SIP Messaging Syntax

- Text-based
  - Similar to HTTP
  - Disadvantage – more bandwidth consumption
- SIP messages
  - message = start-line
    *message-header CRLF
    [message-body]
  - start-line = request-line | status-line
- Request-line specifies the type of request
- The response line indicates the success or failure of a given request.

SIP Requests [1/2]

- Method SP Request-URI SP SIP-version CRLF
- Request-URI
  - The address of the destination
- Methods
  - INVITE, ACK, OPTIONS, BYE, CANCLE, REGISTER
  - INVITE
    - Initiate a session
    - Information of the calling and called parties
    - The type of media
    - ～IAM (initial address message) of ISUP
    - ACK only when receiving the final response

SIP Requests [2/2]

- BYE
  - Terminate a session
  - Can be issued by either the calling or called party
- OPTIONS
  - Query a server as to its capabilities
    - A particular type of media
- CANCEL
  - Terminate a pending request
  - E.g., an INVITE did not receive a final response
- REGISTER
  - Log in and register the address with a SIP server
  - “all SIP servers” – multicast address 224.0.1.75)
  - Can register with multiple servers
  - Can have several registrations with one server

SIP Responses

- SIP Version SP Status Code SP Reason-Phrase CRLF
- Reason-Phrase
  - A textual description of the outcome
  - Could be presented to the user
- status code
  - A three-digit number
  - 1XX Informational
  - 2XX Success (only code 200 is defined)
  - 3XX Redirection
  - 4XX Request Failure
  - 5XX Server Failure
  - 6XX Global Failure
  - All responses, except for 1XX, are considered final
    - Should be ACKed

SIP Addressing

- SIP URI (Uniform Resource Identifier)
  - sip:user@host:port
  - sip:collins@home.net:5060
  - sip:3344556789@telco.net

SIP Message Body

- Message body
  - Describe the type of session
  - The most common structure for the message body is SDP (Session Description Protocol).
  - Could include an ISDN User Part message
  - Examined only at the two ends

The Session Description Protocol

- The Most Common Message Body
  - Be session information describing the media to be exchanged between the parties
  - SDP, RFC 2327 (initial publication)
- SIP uses SDP in an answer/offer mode.
  - An agreement between the two parties as to the types of media they are willing to share
  - RFC 3264 (An Offer/Answer Model with SDP)
    - To describe how SDP and SIP should be used together

Daniel<sip:Collins@station1.work.com>

Boss<sip:Manager@station2.work.com>

a

INVITE sip:Manager@station2.work.com SIP/2.0
From: Daniel<sip:Collins@station1.work.com>; tag = abcd1234
To: Boss<sip:Manager@station2.work.com>
CSeq: 1 INVITE
Content-Length: 234
Content-Type: application/sdp
Content-Disposition: session

v=0
o=collins 123456 001 IN IP4 station1.work.com
s=
c=IN IP4 station1.work.com
t=0 0
m=audio 9000 RTP/AVP 4 3 0 8 18
a=rtpmap:4 G723/8000/1
a=rtpmap:3 GSM/8000/1
a=rtpmap:0 PCMA/8000/1
a=rtpmap:8 PCMA/8000/1
a=rtpmap:18 G729/8000/1

b

SIP/2.0 200 OK
...

Daniel<sip:Collins@station1.work.com>

Boss<sip:Manager@station2.work.com>

b

SIP/2.0 200 OK
From: Daniel<sip:Collins@station1.work.com>; tag = abcd1234
To: Boss<sip:Manager@station2.work.com>; tag = xyz789
CSeq: 1 INVITE
Content-Length: 163
Content-Type: application/sdp
Content-Disposition: session

v=0
o=Manager 45678 001 IN IP4 station2.work.com
s=
c=IN IP4 station2.work.com
t=0 0
m=audio 6666 RTP/AVP 18
a=rtpmap 18 G729/8000/1

c

ACK sip:manager@station2.work.com SIP/2.0
From: Daniel<sip:Collins@station1.work.com>; tag = abcd1234
To: Boss<sip:Manager@station2.work.com>; tag = xyz789
CSeq: 1 ACK
Content-Length: 0

d

Conversation

Conclusion

- With better engineered network design, VoIP can be provided with good quality. Many ITSPs are doing that now.
- SIP is a flexible protocol which enables rich applications to be integrated with VoIP services.