IT441: Network Servers & Infrastructure

CLASS 11 : 11 Apr 2005
13.30 - 16.15

Last Time

- sample services:
  - mail
  - WWW

This Time

- sample services, cont’d:
  - mail
  - VoIP

Address Services

- what's in a name?
  - user@someusername.node.subdomain.domain
  - where does "user" get its email?
  - what about
    - user@someothernode.subdomain.domain

Address Services

- many organizations now use email gateway
  - node dedicated to receiving, forwarding, sending email
- lets users have email addresses like
  - username@domain
- gateway receives all inbound email
  - either forwards to particular node, or,
    - keeps local and user’s user-agent retrieves the email therefrom

Address Services

- gateway receives all inbound email
  - either forwards to particular node
- gateway needs list of entries associating
  - username@domain with
    - username@node.subdomain.com
  - this association be 1:1?
Address Services

- must this association be 1:1?
- no: can have one name: many names
  - so email gateway can match a listname with > 1 mailboxes and must generate individual copies of the message to each
  - such a gateway is hence known as a *mail-exploder*
  - see Comer fig. 32.5 for example
  - users must have their email address added to list in order to receive mailings

Post Office Protocol (POP)

- example client server interaction in POP3:
  - connect to port 110 on server...
  - OK POP3 server ready
  - USER fred
  - OK
  - PASS d3rf
  - OK user Tagged on
  - LIST
  - *OK 2 messages (120 octets)*
  - 1 120
  - 2 200
  -
Post Office Protocol (POP)

- example client server interaction in POP3, cont’d:

  RETR 1
  <the POP3 server sends the entire message here>
  .
  DELE 1
  +OK message 1 deleted
  DELE 1
  +OK message 1 already deleted
  QUIT
  +OK POP3 server signing off

POP3

- available commands in the POP3 protocol include:

  APOP  PASS  STAT
  DELE  QUIT  TOP
  LIST  RETR  UIDL
  NOOP  RSET  USE

Another Mail Fetching Protocol

- instead of POP, a user agent may use Internet Mail Access Protocol (IMAP) RFC3501
- particular advantage: allows management of messages on a server
  - e.g., can organize messages into folders
  - also provides ability to retrieve only parts of a message
    - e.g., subject lines
    - e.g., specific part of MIME multipart message

Servers

- provide a service
- provide content
  - static
  - dynamic
  - generated on-demand
  - “streamed”
    - on-going flow of data
    - often real-time sensitive
  - may be pre-computed (e.g., mp3)
  - real-time, not pre-computable (e.g., telephony)

Internet Telephony

- real-time
- full duplex
- cannot be pre–computed, pre-compressed
- provide usual telephony services
  - call forwarding
  - call waiting
  - caller ID
  - PBX services
- interoperability with existing PSTN

Telephones

- what is a telephone?
- activation control switch
- audio capture device (microphone)
- audio reproduction device (speaker)
- activity alert device (ringer)
- address selection device (keyboard)
- any combination of hardware and software that performs these functions can be a telephone
**Conventional Phone Network**

![Diagram of a conventional phone network.](image)

**Voice over IP**

- end points of calls:
  - computer (with mic and spkr)
  - conventional telephone
  - a 'phone call' can originate or end at either
  - need to have Internet interoperate with PSTN
- signal standards
- addressing
  - phone numbers
  - IP addresses

**The New Layout**

- need to use telephony stds for PSTN work
- need to use IP stds for Internet work

**Special Gateways**

- media gateway:
  - audio data xlate between IP and PSTN
  - "overall" coordination between IP phones
  - mediates access to callee info in PSTN for IP side
- signaling gateway:
  - signaling operation translation (e.g., between SIP and SS7)
  - xlates call requests from either 'side' into corresponding representation for other 'side'

**The Standards**

- VoIP standards proposed by
  - IETF
  - ITU-T
It Started Off Well…

- ITU-T and IETF agree on the following common ground:
  1. audio is digitized using PCM
     - 8 bits per sample
     - 8000 samples per second 64 kbps
  2. use RTP to move data (Real-Time Transport Protocol) [RFC3550]
     - not a transport-layer protocol
  3. RTP message carried via standard UDP
     - further encapsulated as IP

Why UDP?

- UDP provides
  - out of order datagram delivery
  - duplicates, missing data
  - little or possibly no checking on data
  - overhead of TCP too high for real-time needs
  - better to have a gap from dropped or error pkt than delay waiting for re-xmit
  - introduce real-time clock values into datagrams
    - can eliminate duplicates
    - can reassemble in correct sequence

Time Sensitive (Multimedia) Data

- voice data is carried in UDP datagrams
- needs (at least) timestamps
  - may need other information
  - need for a protocol that can carry time sensitive data via UDP
- what other requirements do we have?

RTP

- Real-time Transport Protocol (RTP) [RFC3550]
  - really a transport protocol?
- Real-time Transport Control Protocol (RTCP)
- supports wide range of ‘real-time’ data
  - audio, video
  - defines, for each class of real-time data:
    - profile: guides understanding of header data
    - 1 format: how data after hdr is represented
RTP Message

0 1 2 3 4 5 6 7

8 9 10 11 12 13 14 15

+----------------+-----------------+
|                   |                  |
+----------------+-----------------+
| v | padding length |
+----------------+-----------------+
| 0 1 2 3 4 5 6 7 |

version = 2
synchronization source (SSRC) identifier
sequence number
timestamp
sequence number
synchronization source (SSRC) identifier
sequence number
synchronization source (SSRC) identifier

RTP: an example of ALF

- ALF: application layer framing
- use to design new protocols for multimedia content
- notion: only the application really best knows its requirements

RTCP

- main functions:
  - app performance, network feedback
    - for adaptive rate applications
  - synchronize different media streams from same sender
    - e.g., audio and video may be sent separately
    - streams may have different clocks, different drift
    - represents identity of sender for display in UI

Moving Datagrams Is The Easy Part

- once endpoints are known, moving datagrams from src to dest is 'easy'
- signaling: work done to manage infrastructure:
  - set-up a call
  - tear-down a call
- current PSTN uses "Signaling System 7" (SS7)
- any IP elements in VoIP must interoperate with SS7 to use any PSTN elements
Standards, Again

- IETF proposes:
  - Session Initiation Protocol (SIP)
- ITU-T proposes:
  - H.323 group
- jointly have proposed:
  - Megaco, MGCP

SIP Terminology

- user agent: entity originating or receiving a call
  - user agent client: originates a call (outgoing)
  - user agent server: receives a call (incoming)
- location server: provides user-specific info, e.g.,
  - services user subscribes to
  - user preferences
- proxy server: handle routing, enforce policy
- redirect server: forwarding, toll-free connections
- registrar server: applies authenticated updates to db used by location server

The SIP Method(s)

- SIP defines methods (message types):
  - INVITE: endpoint invited to join a session
  - ACK: acknowledge INVITE request
  - BYE: terminate session: end call
  - CANCEL: cancel pending request
  - REGISTER: URL where user can be reached
  - OPTIONS: query callee’s capabilities

Sample SIP Session

Who You Gonna Call?

- IP targets identified by IP addresses
- telephones ‘identified’ by telephone numbers
- ITU-T E.164 telephone number standard
- different formats in different places
- how do you identify callee?
  - by phone number?
  - by IP address?

Mapping Phone Numbers

- ENUM protocol [RFC 2916]
  - reverse string holding phone number
  - treat each digit as segment of domain name
  - in special domain e164.arpa
  - e.g., phone number 1–800–555–1244 becomes:
    4.4.2.1.5.5.5.0.0.8.1.e164.arpa
  - is 1:many mapping...DNS server replies with:
    - list of hosts with that phone number
    - protocol needed to reach each
Finding How to Reach Users

- TRIP: Telephone Routing over IP [RFC 3219]
  - location servers use to advertise known "routes" to each other

H.323 Terminology

- terminal: provides IP telephone function
  - for voice
  - for video and other data as well
- gatekeeper: location and signaling functions + coordinates gateway access to PSTN
- gateway: interconnect IP to PSTN networks
- MCU: services including multipoint conferencing

H.323 Family

- H.323 is collection of protocols that collectively perform all telephone operation
- uses both TCP and UDP for transport
- not ASCII-based; uses BER to represent ASN.1 binary messages

H.323 Family

- Comer, Fig. 33.4 shows the family:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Signaling</th>
<th>Registration</th>
<th>Audio</th>
<th>Video</th>
<th>Data</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>H.225</td>
<td>H.225.9-RAS</td>
<td>0.711</td>
<td>0.295</td>
<td>0.773</td>
<td>0.776</td>
</tr>
<tr>
<td>8</td>
<td>H.225.9-Q.931</td>
<td>H.225.9-Q.931</td>
<td>0.711</td>
<td>0.295</td>
<td>0.773</td>
<td>0.776</td>
</tr>
<tr>
<td>7</td>
<td>H.225.9-AES</td>
<td>H.225.9-AES</td>
<td>0.711</td>
<td>0.295</td>
<td>0.773</td>
<td>0.776</td>
</tr>
<tr>
<td>6</td>
<td>RTP, RTCP</td>
<td>TCP, UDP</td>
<td>UDP</td>
<td>TCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>IP, RSVP, and ISUP</td>
<td>TCP, UDP</td>
<td>UDP</td>
<td>TCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TCP, UDP</td>
<td>UDP</td>
<td>UDP</td>
<td>TCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TCP, UDP</td>
<td>UDP</td>
<td>TCP, UDP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

International Packet Communications Consortium

- formerly ISC: International Softswitch Consortium
- industry-formed group
- define standards that can be implemented and put into service
- attempt to embrace all situations...

IPCC View

- MCC-F: keeps state info in endpoints, provides call logic and call control
- CA-F: maintains call state (e.g., SIP, H.323, Q.931)
- RG-F: signaling between different nets (e.g., SS7 -- SIP)
- RF/AE: Routing of calls/Accounting info re. calls
- SG-F: signaling IP network --> PSTN
- AGS-F: signaling IP network --> circuit-sw access net (e.g., ISDN)
- MG-F: digital audio xlate
IPCC View

- AS-F: application services (e.g., voicemail)
- SC-F: invoked when AS-F controls a service
- MS-F: handles media pkt stream for AS-F app
- MC-F: digital audio xlate
  - may also recognize DTMF tones and phone device events (like OH)

Distributing Services

- client–server model:
  - server provides some service useful to ≥ 1 client
  - server operator’s concerns:
    - how much work does server need to do?
    - how many users can be supported?
    - can server-side infrastructure grow seamlessly?
    - connectivity
    - does server have to do all the work?
    - off-load some to client, e.g., apptets, Javascript
    - database(s) may run on other machines

Distributing Services

- does server have to do all the work?
  - off-load some to client, e.g., apptets, Javascript
  - database(s) may run on other machines
  - 'internal' network traffic: performance issues

Client Server Model

- a familiar example: an HTTP server
  - listens on port 80 for web-client requests
  - in response to a request, HTTP server delivers a text-file in HTML format
  - processing burden on server?
    - light: is just delivering files from disk
    - load depends on frequency and number of clients requesting service

Evolution of a Service

- web pages may be:
  - static: contents created through some application program and file stored in server’s file system
  - dynamic: some or all parts of the page result from computation performed at the time the page is requested
Evolution of a Service

- Dynamic pages place greater burden on server
  - Move some of the work to the client
    - e.g., JavaScript or Java applets for interactive menus
  - May not be able to make client do more of the work
    - e.g., data to be provided may only be available to server and may not be on server

Middleware

- Intermediate layer of software between clients and servers
- Mediates useful communication between clients and servers
- Often plays role of broker between client and some set of servers that may be able to provide answer
- Ideally hiding details from client (at least from user)

Distributed Objects

- Some requests will be for objects and/or methods that operate on them
- Java has built in support for this capability
  - Requiring every distributed object to itself be written in Java
  - What if we want to use objects in other/different environments?
    - Like C++, or Ada
    - No single common representation for the objects or their methods
**Which Broker?**

- Common Object Request Broker Architecture (CORBA) a popular standard for these object brokers

**CORBA**

- A "distributed object" describes services it can provide via Interface Definition Language (IDL)
- A generic language for describing services
- Object Request Broker, ORB, 'connects' clients (requesters) with servers
  - Interaction is between ORBS: one on server, one on client
  - Client-side ORB locates a distributed object sought by client, handles server responses
  - Server-side ORB registers services, handles requests
- [http://www.omg.org](http://www.omg.org)

**Object Brokers**

- CORBA is not alone
  - But is supported by Apple, IBM, Sun, and others
- Microsoft has different 'standard': Common Object Model (COM)
  - Interoperates with CORBA
  - Analogous middleware used for software agents to locate and use services
  - E.g., CMU's RETSINA
- Hot topic: pervasive computing depends (in part) on locating services and delivering results

**Basic Distributed Communication**

- Mediated by message passing
  - Sender
  - Receiver
- Delivery class:
  - Unreliable
  - Reliable

**Basic Distributed Communication**

- Mediated by message passing
  - Sender
  - Receiver
- Delivery class:
  - Unreliable
  - Reliable
- Process action:
  - Blocking
  - Non-blocking
More Than Just Data

- an application may want to invoke some processing service on a remote system
- use Remote Procedure Call (RPC)
- should behave like a local procedure call
- like a local procedure, has a standard interface with defined type
- can check type errors during compile
- can make client and/or server code more portable
- only re-write ‘new’ stuff; use same RPC

RPC Issues

- parameter passing
  - call by value is ‘straightforward’
  - e.g., sqrt(25.0)

RPC Issues

- parameter passing
  - call by value is ‘straightforward’
  - call by reference very awkward
  - e.g., sqrt(&x) or strlen(s)

RPC Issues

- parameter passing
- parameter representation
  - big-endian vs. little-endian
  - ASCII vs EBCDIC vs UNICODE
  - may be handled by underlying network, may not

RPC Issues

- parameter passing
- parameter representation
- persistence of binding
- timing action:
  - synchronous
  - caller waits for remote procedure to return
  - analogous to blocked send in message passing
  - may prevent efficient exploitation of parallelism
**RPC Issues**

- parameter passing
- parameter representation
- persistence of binding
- timing action:
  - synchronous
  - asynchronous
    - caller does not wait for remote procedure to return
    - receives replies whenever they are delivered
    - requires synchronization mechanism

**Some Other Names**

- MSRPC, MSRPC2
  - Microsoft’s remote procedure call standard
  - latter often referred to as ORPC because provides support for objects

**Getting Service**

- so far have focused on clients/servers accessed over ‘wired’ connections
  - requires user (client) to be at ‘fixed’ location
  - fastest growing sector of network services is targeted at services accessible by portable (wireless) devices
  - non-voice services for cell phones
  - network access at public locations (internet café)
  - services for network-capable PDAs

**Some Other Names**

- MSRPC, MSRPC2
- SOAP: Simple Object Access Protocol
- Jini:
  - from Sun Microsystems
  - Java-based
  - allows a ‘computation’ to be performed by collection of resources distributed over many networked computers
  - allows a process to migrate from one computer to another
  - clients/servers can use any protocol of their choosing
**Wireless Standards**

- network access via RF radio
  - susceptible to problems of radio: noise, interference, etc.
  - tradeoff: bit rate ↔ power
  - lower power ↔ less distance

---

**IEEE 802.11**

- frequency ranges:

<table>
<thead>
<tr>
<th>Region</th>
<th>Frequency Range MHz.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>2.4000 - 2.4835</td>
</tr>
<tr>
<td>France</td>
<td>2.465 - 2.4835</td>
</tr>
<tr>
<td>Japan</td>
<td>2.4710 - 2.4970</td>
</tr>
<tr>
<td>USA</td>
<td>2.4000 - 2.4835</td>
</tr>
</tbody>
</table>

---

**Deploying 802.11**

- usually run in infrastructure mode
- have wired access point serving area
- multi-channel allows adjacent APs to operate independently
- 802.11b delivers 11 Mbps service
- newer version 802.11g:
  - delivers (theoretically) 54 Mbps
  - competes for spectrum with 802.11b
  - can restrict access or allow 'open' access

---

**802.11 Wireless**

- operates in two modes:
  - infrastructure
  - ad-hoc

  - Uses access point which coordinates transmission by wireless units for efficiency
  - AP usually connected to wired LAN

  - Lets 802.11 stations work without AP

---

**802.11 Wireless, cont’d**

- data rates 1, 2 and 11 Mbps
- frequency: 2.4 GHz (ISM band) worldwide
- several transmission methods used:
  1. baseband infrared
  2. frequency hop spread spectrum (FHSS)
  3. direct sequence spread spectrum (DSSS)
  4. orthogonal frequency division multiplexing (OFDM)
  5. high-rate direct sequence spread spectrum (HR-DSSS)

  - Many units now available at low cost (~ $100/sht)
  - But not all may interoperate depending on options supported

---

**Emerging Standard**

- current wireless services not fast enough for new services to be offered, e.g.,
  - images,
  - videos,
  - audio

  - Proposed, but not yet standardized: 802.11n
  - 100 Mbits/sec (may be able to do up to 630)
  - IEEE expected to adopt in 2006 or 2007