Internet, intranet and Web — Lecture I

Technologies and protocols for application communications

Marco Solieri marco.solieri@lipn.univ-paris13.fr
Info et Réseaux en Apprentissage, Sup Galilée, Université Paris 13

November 3rd, 2014

Outline

Contents

1 Computer communication

1.1 Communication and layers

Communication

Definition 1 (Information Technology)**.** Technologies for creation, storage, transmission, and manipulation of information.

Definition 2 (Communication)**.** Act of sending, from a sender to a recipient, a message via a channel.

Channel and time:

synchronous channel is a transmission medium,

asynchrounous channel is a storage medium.

Definition 3 (Network)**.** Collection of addressed nodes (eventually terminals) and links able to communicate.

Layers in computer communication (1of2)

Abstraction:

- derives higher concept from usage of literal concepts,
- loved by computer science and engineering.

Senders and receivers:

- network adapter and its firmware,
- operating system (host),
- process (application).

Layers in computer communication (2of2)

Storage channels:

- physical support (hard disk, optical discs),
- filesystem (FAT32, Ext4, XFS, ISO 9660),
- format (text, images, audio, video)

Transmission channels (ISO/OSI):

- link protocol (802.3, 802.11) for local networks,
- internet protocol (IP) for inter-networks, and transport protocols (UDP, TCP),
- application layer protocols (SMTP, FTP, HTTP).

1.2 End-to-End argument

Transmission channels and layering

Definition 4 (End-to-End argument)**.** Link or transport levels can not and must not offer application-level features.

- first formulated by Saltzer, Reed, and Clark in 1981
- architecture principle of encapsulating layers
- one of the most important and successful design principles of the Internet

Opinions about End-to-End

Pros:

- lowers core network complexity
- increases extendibility of services

Cons:

- prevents core network from meeting the requirements of some application for quality of service
- hampers the application-level targeting of ISPs' offers
- is open to malice that exploits core network's neutrality

Typical answers:

- add features to the core network and partially break the principle,
- establish service domains and preserve the principles.

2 Electronic Mail

2.1 Introduction

Electronic Mail

Definition 5. Asynchronous communication technologies for message exchange realized with client-server and text-based protocols and formats.

Dissection of a message exchange from sender to recipient:

1. sender ⟶ sender's server,
2. sender's server ⟶ recipient's server,
3. recipient's server ⟶ recipient.

Electronic Mail: standards
IETF definitions:

SMTP communication model, protocol for message transfer, error codes, message format, addresses, dates (RFCs 821, 822; 2821, 2822; 5321, 5322)

MIME extension to the message formats (RFCs 2045, 2046, 2047, 2048, 2049)

ESMTP extensions to SMTP (RFCs 1869, 1652, 1870, 1830, 2197, 1891, 1985, 2034, 2487)

POP message access (RFC 1939)

IMAP message access (RFC 2060)

2.2 Message transfer

SMTP: Simple Mail Transfer Protocol
Scopes:

- exchange of messages,
- recipients verification.

Definition 6 (SMTP connection).

- opening
- command dialogue: a sequence of
  - command by client
  - reply by server
- closing

SMTP: commands
Main commands:

MAIL FROM sender identification

RCPT TO recipient identification

DATA message content

SMTP: replies

Definition 7 (SMTP reply). A three-digits numeric code and a human-readable string

Reply types:

2xx Positive Completion: requested action successfully completed,

3xx Positive Intermediate: requested action pending because the server is waiting for some additional information,

4xx Transient Negative Completion: command not accepted and requested action not occurred, because of a temporary condition,

5xx Permanent Negative Completion: command not accepted and requested action not occurred, they will never be.

Example

Example 8 (SMTP connection, 1 of 2).

```
220 foo.com Simple Mail Transfer Service Ready
> HELO bar.com
250 foo.com says: Nice to meet you bar.com
> MAIL FROM:<smith@foo.com>
250 OK
> RCPT TO:<bob@bar.com>
550 No such user here
> RCPT TO:<alice@bar.com>
250 OK
```

Example

Example 9 (SMTP connection, 2 of 2). > DATA

```
354 Start mail input; end with <CRLF>.<CRLF>
> Buy cialis and viagra
Blah blah blah. . .
blah blah blah. . .
.
250 OK
> QUIT
221 foo.com Service closing transmission channel
```

2.3 Message format

Message format in SMTP

Definition 10. Message *headers* `CRLF CRLF` *body*

Header *field_name*:*field_value* `CRLF`

Body *text_of_the_message*

Problems:

- character set: 7-bit ASCII,
- insertion of a `CRLF` sequence at most every 1000 characters,
- maximum size: 1 MB.

MIME: Multipurpose Internet Mail Extensions

Redefinition of the message format, introducing

- different encoding format:
  - other charsets (e.g. `ISO-8859-1`, `UTF-8`)
  - extensible set of format for non-textual messages,
- multi-part messages:
  - different encoding for different parts (e.g. body and attachments),
  - no size limit.

Backward-compatibility with SMTP channel:

1. sender's user agent encoding from MIME to SMTP format
2. message(s) transmission
3. recipient's user agent decoding from SMTP to MIME

MIME: headers

Added headers:

`Content-Type` defines the data type of the message part

- useful in choosing the best user presentation,
- permits message inclusions;

`Content-Transfer-Encoding` defines the encoding type used for the SMTP channel (e.g. `7bit`, `base64`, `quoted-printable`)

MIME: message

Example 11 (Multipart message: body and attachment). `MIME-Vers`

```
Content-Type: multipart/mixed; boundary="----=NextPart_6E6809

This is a message with multiple parts in MIME format.
------=NextPart_6E680986
Content-Type: text/plain

This is the body of the message.
------=NextPart_6E680986
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFkPgogIDxib2R5PgogICAgPHA
+VGhpcyBpcyB0aGUg Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgP
C9ib2R5Pgo8L2h0bWw+Cg=
------=NextPart_6E680986
```

2.4 Message access

Message access

Recipient's mail server $\xrightarrow{?}$ recipient user

History:

- SMTP doesn't care of it access to the mailbox via filesystem
- POP introduces a simple protocol message retrieval and removal
- IMAP offers an advanced protocol mailbox management and access

POP: Post Office Protocol v3

Definition 12 (POP connection).

- greeting: opening
- authorization: client identification
- transaction: a sequence of
  - command by client
  - reply by server
- update: server perform requested actions

Main commands:

LIST get information about messages: number and size

RETR retrieve a whole message

DELE delete a message

TOP retrieve headers of a message

UIDL get a Unique IDentification of messages

IMAP: Internet Message Access Protocol v4r1

Novelties w.r.t. POP

- multiple mailboxes: managing different folders
- multiple client connection: built-in mechanisms for concurrency handling,
- connected modes of operation: message retrieval on demand, without local storage, server-side search,
- MIME parts access: separated and partial fetch of message parts,
- message state information: message tags, e.g. *read, replied to, important, to do.*

2.5 Communication security

Communication security

Definition 13 (Communication security). **Authentication** the message received comes from the sender.

Confidentiality the message sent goes to the recipient only.

TLS: Transport Layer Security

A transport underlay channel for security

- Authentication: digital signatures, with asymmetric-cryptography (public and private keys/certificates),
- Confidentiality: channel encryption, with symmetric session keys.
- Centralized trust model, with Certification Authorities.
- Application independent: transparent channel
- Open standard: IETF RFC 5246 (was: Secure Sockets Layer (SSL))

TLS and Email

STARTTLS

- protocol extension to:
  - SMTP (RFC 3207), POP and IMAP (RFC 2595)
  - other protocols (NNTP, XMPP . . .)
- channel upgrade to TLS: session security

Cost-effectiveness of TLS

- high for confidentiality: session-key exchange and go,
- low for authentication: need for an authority's certification.

Spread within email services

- high for confidentiality
  - common in user-server (email submission and access),
  - common in server-server (email transfer);
- low for authentication
  - common as server-to-user, less common as server-to-server,
  - negligible as users-to-server.

TLS and email, questions

Consider an email transfer over TLS, where the certificate is not verified as trusted by any CA trust chain.

- What security properties are lost?
- About what?
- Why?

So what about the rest?

OpenPGP, Pretty Good Privacy

With TLS, in the whole user-to-user communication:

- incompleteness of security composition,
- therefore insecurity.

OpenPGP: an application underlay channel for security

- Secure, via encryption and signature of messages (asymmetric keys paradigm)
- Application-level security: from sender to recipient.
- Distributed trust model: "web of trust" (IDs signatures and transitivity).
- Proposed standard: IETF RFC 4889

OpenPGP and Email

With OpenPGP in the whole user-to-user communication:

- nobody can tamper a signed message,
- nobody can eavedrop a crypted message,
- therefore user-to-user security.

Security actions (encryption/signature and decryption/verification):

User
1. save of the message/file,
2. invoke the program.

Application
- MIME extensions (proposed standard RFC 1847): `Multipart/Signed` and `Multipart/Encrypted`
- MUA capability of performing security actions (built-in or plug-in)

OpenPGP and MIME

Example 14 (Multipart message with digital signature).

```
Content-Type: multipart/signed; boundary="----=NextPart_6E680986";
  protocol="application/pgp-signature";  micalg=pgp-sha1

------=NextPart_6E680986
Content-Type: Text/Plain

This is the message.
------=NextPart_6E680986
Content-Type: application/pgp-signature; name=signature.asc
Content-Description: This is a digitally signed message part.

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.12 (GNU/Linux)

iEYEABECAAYFAk+e/+IACgkQly3UgJ8i5j28NACeLEZeYND7q1IsvEOoHg5mOGxT
KQoAmgO6yobL3xceQ5Jnu7JkD+s15ALf
=KDTV
-----END PGP SIGNATURE-----
------=NextPart_6E680986
```

3 Widespread message distribution

Electronic mailing list

Single Email address identifying a list of address.

Functioning:

1. user requests for subscription to a list,
2. server add the user's address to the list,
3. user post a message (send to the list address),
4. server forward the message to each subscriber.

Additional features:

Moderation incoming messages need to be approved.

Restricted subscription need to be approved.

Closed only subscribers can post.

Archive messages are stored for later retrieval (i.e. via Web).

Uses: announcement lists, newsletters, public or private discussion lists.

Usenet and NNTP

For public lists the Email load is excessive and unneeded.

Usenet (IETF RFC 1036)

- Idea: distribute not to users, but to servers aggregating users.
- Users post to the local server.
- Users retrieve from its server messages of interest on demand.
- Servers floods messages to "friend" servers.

Protocols:

- UUCP: Unix-to-Unix Copy before the Internet and until 90s, now dead.
- NNTP: Network News Transfer Protocol (IETF RFC 3977)
  - message distribution between servers,
  - message access for user reading and posting,
  - security with TLS.

4 Instant messaging

Instant messaging

Synchrous communication technologies evolution:

1980s real-time text messaging with multiple users support and peer-to-peer architecture (Zephyr Notification Service, Internet Relay Chat, `talk`)

1990s text messaging with audio/video support with centralized architecture, proprietary protocols, GUI clients (OSCAR for AOL IM and ICQ, Rendezvous for MSN)

2000s open standards for messaging and audio/video (eXtensible Messaging and Presence Protocol, Session Initiation Protocol, H323)

XMPP: eXtensible Messaging and Presence Protocol

Project Jabber defined and implemented the first version (1999)

Key features:

Instant messaging text messaging, audio/video call, file transfer, presence information and contact list maintenance;

Open standard IETF RFCs 6121, 6122, 3922, 3923;

Decentralization email-style architecture with message routing;

Extendibility XML based protocol, usable as middleware messaging.

Now used by:

- Google Talk (2005),
- Facebook (2010),
- Microsoft .NET Messenger (2011),
- Nokia Ovi Contacts (2012).

5 File communication

5.1 File transfer

File transfer

Definition 15. Asynchronous communication technologies realized with file storage and file exchange protocols.

Architectures:

Centralized File server offers to clients the service of storage, upload and download of files. Example:

- file system and a transfer protocol (FTP, HTTP).

Distributed Peers exchange and store files with other peers. Examples:

- file system and file sharing protocol (eDonkey, BitTorrent),
- distributed file storage (FreeNet).

5.2 File Transfer Protocol

FTP: File Transfer Protocol

History:

- first drafted for use on ARPANET (1971)
- standard RFC 959

Dissection of a FTP connection:

1. Client connection to the server: opening of control connection.
2. Client login with username and password (fake identification for anonymous services)
3. Transactions, a sequence of:
   (a) Command from client. If transfer is requested:
      i. opening of the data connection by client or server (active/passive mode)
      ii. file transfer on data connection,
      iii. closing of the data connection.
   (b) Reply by server.
4. Client disconnection: closing of control connection.

Representation of data

Four data representations (`TYPE` command):

ASCII Extended ASCII 8-bit character encoding, used for plain text files only.

Image Byte per byte, used for binary files in general.

Local Machine-dependent format, used for some proprietary formats.

EBCDIC 8-bit IBM's character encoding, (once) used for text files (by dinosaurs).

Mode of transfer

Three modes (`MODE` command):

Stream Continuous stream of data End with:

- end of underlying TCP connection
- EOF or EOR characters.

Block Segmentation of data into blocks with header:

- block size
- description

End with a special descriptor. Easy resume of transfer: good for large files.

Compressed Compression of data, typically run-length encoding.

FTP: client commands

RETR transfer a copy of file

STOR store a copy of file

RNTO rename file

DELE delete file

PWD print current working directory

MKD make directory

RMD delete directory

FTP: server replies

Definition 16 (FTP reply)**.** A three-digits numeric code and a human-readable string

1xx Positive preliminary: requested action initiated, wait for completion,

2xx Positive completion: requested action completed,

3xx Positive intermediate: requested action pending because the server is waiting for some additional information,

4xx Transient Negative Completion: command not accepted and requested action not occurred, because of a temporary condition,

5xx Permanent Negative Completion: command not accepted and requested action not occurred, they will never be.

FTPS: FTP Secure

Architecture:

File transfer FTP

Confidentiality SSL/TLS channel

Authentication SSL/TLS channel or FTP

Two security modes:

Implicit communications over TLS assumed (deprecated).

Explicit channel upgrade to TLS (similar to STARTTLS)

Standard: IETF RFCs 2228, 4217.

SFTP: Secure File Transfer Protocol

Architecture:

File transfer New advanced file protocol, providing

- transfer: retrieval and storage;
- management: moving, renaming, directories tree;
- access: ACL permissions.

Security Generic underlay channel (i.e. SSH)

Standard:

- IETF abandoned Draft by SecShell working group: latest version 6 in draft 13, July 2006.
- Widely implemented: OpenSSH

5.3 Version control systems

Version control systems

Definition 17 (Version control system)**.** Asynchronous communication technologies to manage complexity of changes to data.

Use cases:

- documents revisions
- web publishing
- software engineering process: development, maintenance, updates
- system administration: configuration

VCS architectures

Service location:

Application embedded capabilities into the software, metadata into the application file format.

Simple management of a line of changes, for office applications, wikis, CMSs.

Stand-alone dedicated software relying on centralized/distributed metadata storage.

Advanced management of a graph of changes, for complex change processes.

VCS actions

Linear changes:

Checkout initialize a working copy,

Commit record a new version: store changes,

Update sync to another version: get changes.

Graph changes:

Branch start a fork: new changes are now separated,

Merge end a fork: apply its changes to another branch,

Tag give a name to a version.

VCS: communication architecture

Centralized:

- one repository and many working copies,
- all actions performed remotely,
- file lock,
- small local storage,
- prominent system: Subversion.

Distributed:

- many “working repositories”
- usual actions performed locally, merge from/to remotely,
- data redundancy,
- big local storage,
- prominent system: Git.

Suitability depending on the development model architecture.

Conclusions

Conclusions

Technologies and protocols for application-level communication share some design principles:

- rigorous roles of communication (server and client / peers),
- text command and replies,
- standardization,
- KISS principle: “everything should be made as simple as possible, but no simpler”.