Digital Communications II

Jon Crowcroft
ack for slides, to Ian Pratt

Michaelmas 2002

Computer Science Tripos Part II

Part1: Course Introduction
and Layering Overview

Structuring Communication Systems

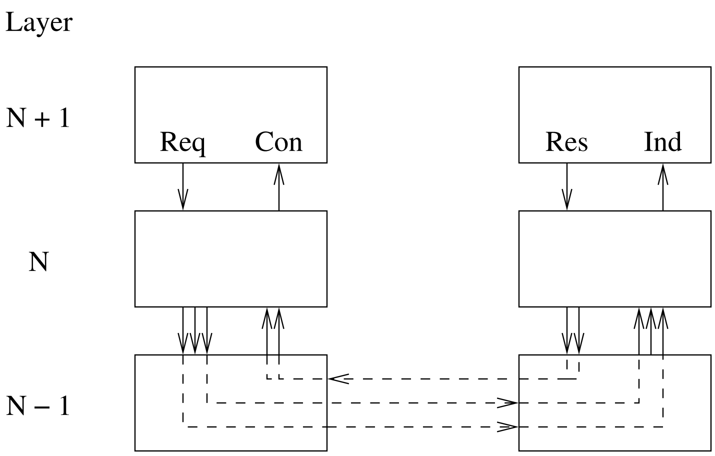
Sometimes we need to partition problems:

- Implementation in hardware or software
- Function the responsibility of the network or host?
- Separate private from public
  – more generally to separate enterprises
- Separate distinct suppliers

Partitioning for comprehension – two elements:

- abstraction
  – define interfaces and semantics
  – allow implementation choices
  – hence allow future replacement
- layering
  – one restricted view of module inteaction

Layering



Abstraction with modules in a stack:

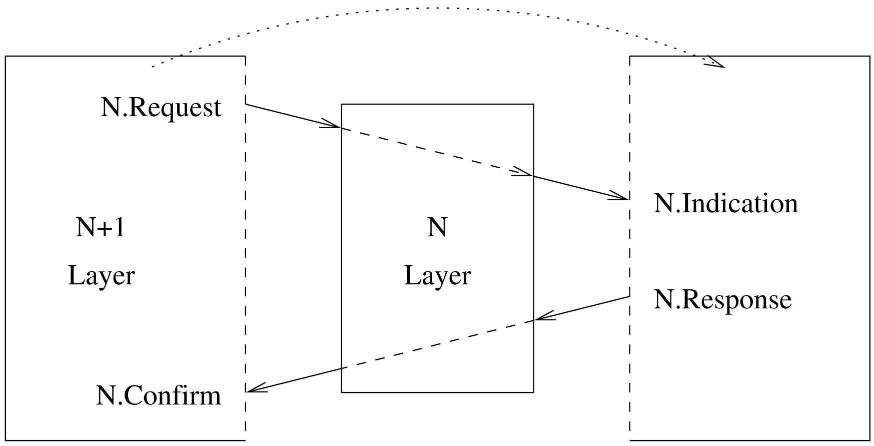
- at the bottom, wire or fibre
- at the top, applications

layer n uses layer n-1 to provide a service to layer n+1

Typically interface described in terms of:

- Request / Confirmation
- Indication / Response
- (implied time relationship)

Service abstraction



Terminology:

- PDU: a layer talks to its peers with *Protocol Data Units*
- SDU: the service primitives take *Service Data Units*
- SAP: you say who you want to talk to by quoting its *Service Access Point*

For exmaple:

```
rc = sendto (sock_addr *sap, void *buf, int len);
```

- An SDU (coded as three arguments),
- Which contains a PDU of length `len` at `buf`
- To be sent to peer SAP defined by `sap`

Standards

Communication requires agreement about representation of information (coding) and about procedures for exchanging information (protocol).

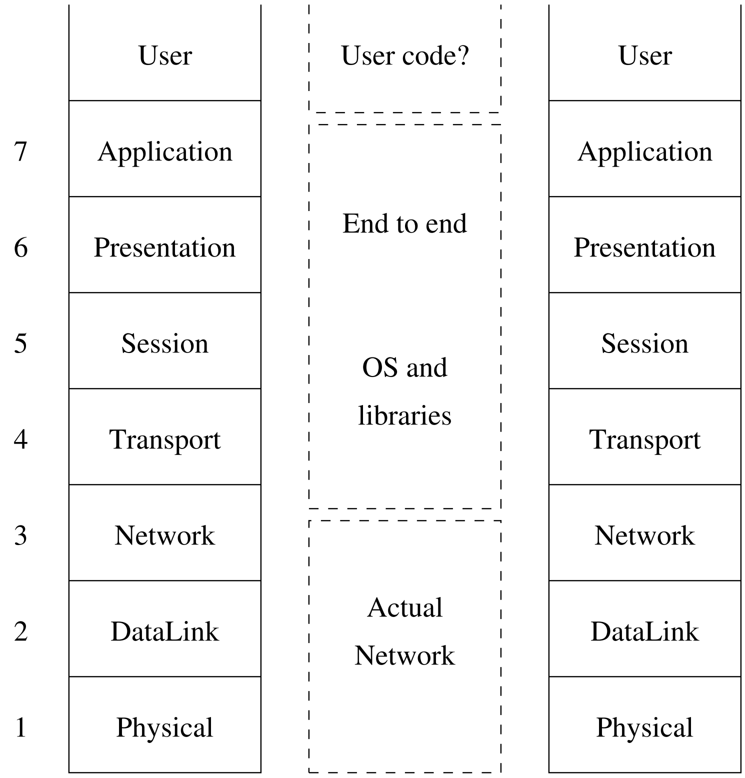
Standards are such agreements amongst suppliers, users, PTTs, etc. Standards organisations are the forum in which these standards are reached.

Different implementations adhering to the same standard will communicate

Organisations:

- ISO (Open Systems Interconnection – OSI)
  Composed of national and geopolitical entities:
  - ANSI, BSI, ETSI
- ITU-T, was CCITT (eg X and V series)
  Telecommunications suppliers and operators
- IETF (Internet Engineering Task Force
  Internet Drafts, Request For Comments (RFCs)
- IEEE (e.g. 802.x LANs)
- ECMA (manufacturers)

OSI Reference Model



- OSI: Open Systems Interconnection
- seven layer model
- defines functions of different layers
- levels 1 to 3: traditional network levels
- levels 4 to 7: end system components

OSI Reference Model

Important to distinguish OSI reference model from OSI protocol standards

OSI reference model:

- defines a 7 layer abstraction for communications
- aims to collect similar functions in the same layer
- provides a terminology for functions
- provides a framework in which to define abstract interfaces

OSI protocol standards:

- set of specific protocols, defining
  - PDU coding
  - state machine transitions
  - SAP coding

Application

- High-level building blocks for applications
- file transfer and management (FTAM)
- job transfer and manipulation (JTM)
- electronic mail, message handling service (X.400, SMTP)
  - user agent
  - message transfer agent
- directory services (X.500)
- virtual terminal (VT)
- '`fetch_url`', '`render_url`'
- many other applications need such services, so:
  - programs
  - libraries

Presentation

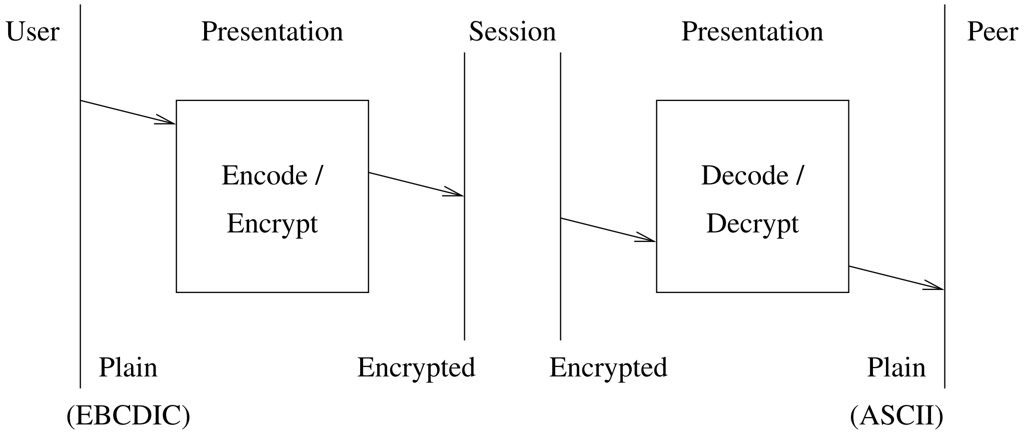
- representation of data and data semantics
- machine / architecture independent
- notation for datatypes (e.g. ASN.1)
  - primitive types: BOOLEAN, INTEGER, BITSTRING, ...
  - constructors: SEQUENCE, SET, CHOICE

```
FILE_OPs ::=
  BEGIN
  PDU ::= CHOICE {
            READ,
            WRITE }
  READ ::= SEQUENCE{
            fileName[0] IA5String,
            offset[1] INTEGER,
            length[2] INTEGER
  ...
```

- various sets of encoding rules, e.g. Basic Encoding Rules (BER):
  - ASN.1 type identifier
  - coding length
  - contents
- Extensible Markup Language (XML)
- provides transformation from application representation to network representation

Other codings

Encryption/decryption is a presentation function:



- Text
  - EBCDIC, ASCII
  - CRLF, LF
- Voice/Audio
  - linear PCM : 12/16 bit, signed/unsigned
  - companded (lossy) to 8 bits
    – μ-law, a-law
  - transform coding
    – ADCPM, RealAudio, MP3, SDMI etc.
- Images/Video
  - Raw: RGB *vs.* YUV colour space
  - Spatial: JPEG, M-JPEG, JPEG2000
  - +Temporal: MPEG 1, MPEG 2
  - +Model based: MPEG 4

Session

- Co-ordinating tasks that require multiple transport level streams
- Synchronization
  - Tokens enforce data phases and controlled termination
  - Tokens as synchronization points
- e.g. FTAM for lots of files:
  - Major token per file
  - Controlled release for safe termination
- Choose from:
  - Negotiated release
  - Half duplex
  - Synchronization
  - Activity management
  - Exception reporting
- Having well-defined subsets of these to make life easier for application programmer
- multi-media synchronization
  – audio/video/animation

Transport service

- ISO TP4 service:
  - end-to-end
  - connection oriented
  - sequenced packets
  - error free data
  - flow control
- TCP as above, but byte-stream oriented
- datagram services e.g. UDP
- Other useful variants possible e.g.
  - no retransmissions for real-time data
  - no packet sequencing or partial order
- implementation depends on network and datalink level, may have to do:
  - multiplexing
  - error detection
  - error recovery
  - flow control
  - retransmission

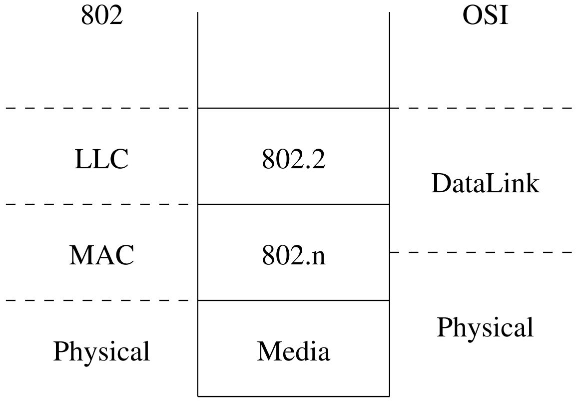
Network level

- Inter-networking: Unifying underlying network technologies
- E.g. Internet Protocol, (ATM)
  - Connection oriented (CONS) *vs.* connectionless (CLNS)
  - Global (hierarchical) addressing
  - Forwarding
  - Segmentation and reassembly
  - QoS specification
  - (congestion notification)
- In ISO model, 3 sub layers:
  - Subnet independent convergence protocol
  - Subnet dependent convergence protocol
  - Subnet dependent access protocol

Network Routers operate at L3

DataLink

Consists of two IEEE 802 layers:



- Logical Link Control: delineating frames
- Media Access Control: sharing the media
- (Mostly implemented in hardware)
- LLC example : 802.2 (Ethernet, FDDI, etc)
  – preamble
  – 48 bit src & dest address, 16 bit type, data
  – CRC, postamble (no length)
- MAC examples:
  – CSMA/CD (shared media Ethernet: IPG, backoff)
  – Token bus (timeliness, e.g. process control)
  – Token ring (efficiency at high load)
- Other DataLink examples: SLIP, PPP
  – IP encoding over "serial lines"
  – header compression

Network 'switches' and 'bridges' operate at L2

Physical

Sockets / pins / volts / wires / optics etc.
generate and recover clocked bitstream:

- RS232, V.24
  - asynchronous (two end clocks can run free)
  - main signals: TxD, RxD, RTS, CTS
- V modem standards, e.g. V.90 (56kb/s digital)
  - negotiation from lower standards e.g. (V.32)
  - echo cancellation, error correction and comp.
- 802.3 lower levels: 10base-T 10base-2 10base-5
  - Manchester encoding over different media
- 100base-TX, 100base-FX, 1000base-FX
  - 8B10B coding for DC balance and framing
- Plesiochronous and Synch. Digital Hierarchy
  - Scramblers
- 1000base-TX, Cable Modems, xDSL, HomePNA
  - exotic coding e.g. QAM-16, OFDM
- 802.11 wireless
  - Direct Sequence *vs.* Frequency Hopping

Network 'hubs' operate at L1

OSI Model Summary

Application – application specific services eg terminal, file system, mail

Presentation – application layer and architecture dependent coding

Session – synchronisation control, common control for application above and beyond “reliable data transfer”

Transport – end-to-end reliable delivery of information

Network – getting the bits across the network, reliably or not

Link – getting the bits across a link, reliably or not, including media access (eg an Ethernet is a link)

Physical – bit encoding, clock recovery, connectors, media

Remote Procedure Call and the OSI Model

- RPC does application layer and architecture dependent coding (marshalling and unmarshalling)
- RPC does recovery from failures over and above reliable transfer of information (at most once, at least once, exactly once semantics)
- operates at (at least) presentation and session layers
- look at an implementation — is the presentation / session split apparent?
- where do you encrypt in RPC?

OSI Model Conclusions (mine)

- has delivered a useful vocabulary, ways to compare systems
- perhaps a bit restrictive: often one runs a network layer over another network layer; model doesn't really allow this
- useful tool, not a master
- don't be concerned with things that don't fit the model e.g.:
  - application level gateways
  - firewalls
  - per-flow queueing routers
  - layer-7 switches
  - transparent HTTP proxying