COMPSCI 314 S1 C

Data Communications Fundamentals

COMPSCI 314 S1C 2006
Data Communications Fundamentals

Lecturers

- Nevil Brownlee – Room 590, n.brownlee@auckland.ac.nz
- Clark Thomborson – Room 593, ctho065@ec.auckland.ac.nz
- Peter Fenwick, p.fenwick@auckland.ac.nz

Tutor

- DongJin Lee – Room 596, dlee064@ec.auckland.ac.nz

Test Date

- Monday 8 May, 6.25 – 7:30 pm

Assignments due

(via the CS DropBox, dates subject to revision)

Wednesday 15 March, Wednesday 5 April,
Wednesday 3 May, Wednesday 24 May

Other matters

- Class representative
- Assignment extensions

  We will consider extensions to the assignment due date only for —
  1. Illness or other unforeseeable emergency
  2. Conflicts with other assignments, but only if the request is made within *one* week of the assignment being distributed

  We will not be sympathetic if told "The 314 assignment is due tomorrow and I have 3 other assignments also due then; can I please have an extension?" The dates have been published weeks ahead; you should have planned your work better or arranged earlier for an extension
- Questions

  Your first contact for questions should be the *tutor*, not the lecturer.
  Also, you could ask on the class forum
- Email

  Email must include the course number (314) and your UPI

Approach to material

- This year we are following the textbook
- The lectures will provide in-depth discussion and comment on the course material. You should read the relevant sections in the textbook!
- The course does *not* cover *all* of the textbook. The sections that are covered are shown on the course outline, as it appears on the *course web page*
- Tutorials are scheduled (in lecture slots 24 and 35) before the terms test and the exam
- *Changes* to the course outline and/or content will be notified on the course web page

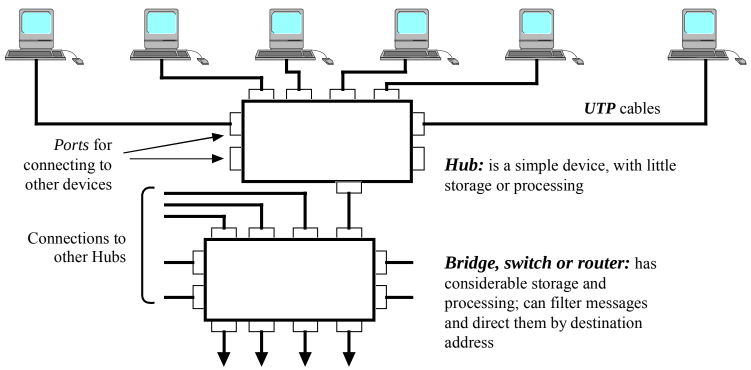
Approximate plan of course

Week starting	Monday	Wednesday	Friday	
27 Feb 2006	1 Introduction	2 Basics	3 Protocols	
6 Mar 2006	4 Protocols	5 Telephone	6 Analog Access	
13 Mar 2006	7 —	8 ISPs, PPP	9 Compression	*Ass 1 due 15 Mar*
20 Mar 2006	10 Compression	11 Error Detect	12 Error Correct	
27 Mar 2006	13 LANs, E/net	14 Lan I/connect	15 VLANs	
3 Apr 2006	16 —	17 Security	18 Security	*Ass 2 due 5 Apr*
10 Apr 2006	19 Security	20 Web Security	– Easter –	
17 Apr 2006	— Mid Semester Break —			
24 Apr 2006	— Mid Semester Break —			
1 May 2006	21 WLANs	22 IP	23 Routing, DV	*Ass 3 due 3 May*
7 May 2006	24 Tutorial (test)	25 Routing, LS	26 Routing, BGP	**TEST: Mon 8 May**
15 May 2006	27 IPv6	28 TCP	29 TCP	
22 May 2006	30 UDP, Streaming	31DNS	32 Email, FTP	*Ass 4 due 24 May*
29 May 2006	33 VoIP	34 Net Mgmt	35 Tutorial (exam)	
5 Jun 2006	*No lectures – just lots of time to study*			

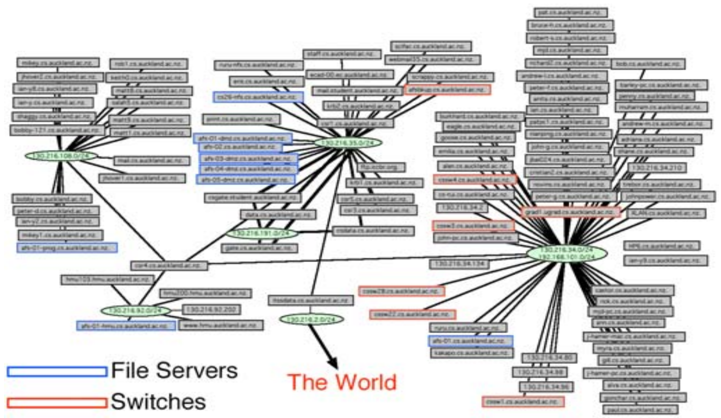
1.1 Overview

- Data communications includes
  - Telephones (PSTN, mobile)
  - Entertainment networks (TV, cable, satellite)
  - Data networks (LANs, WANs, Wireless, Internet)
- We concentrate on the Internet, but will mention the others briefly
- Our focus is on how things work, especially on the underlying protocols – we won't look at 'how to configure a router,' etc.
- We start by looking at the U Auckland campus network …

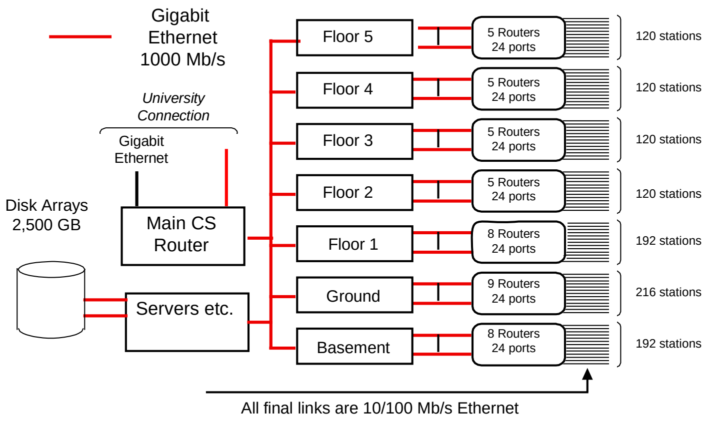
A user connection into a network



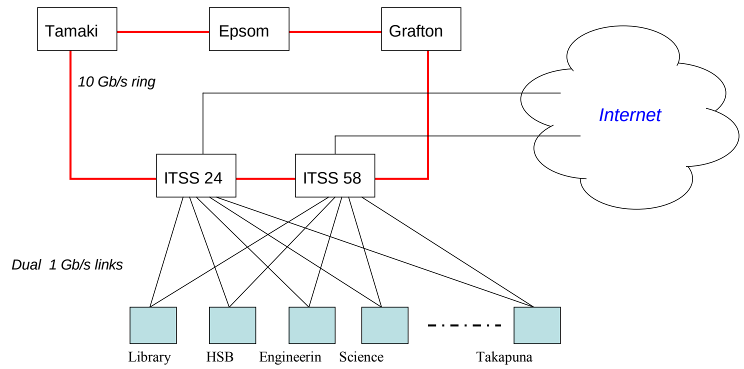
CS Network 2002 (omitting student stations)



Simple view of Computer Science Network, 2003



The University of Auckland Network 2006



Channels and link types

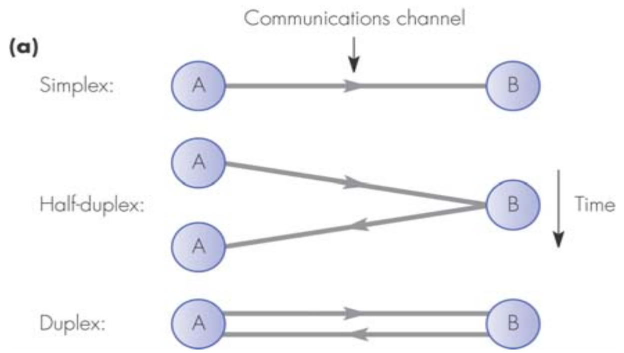


Figure 1.4 Communication modes: (a) unicast

Packet Switching Network

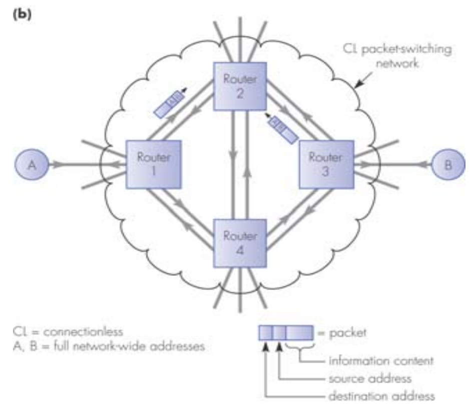


Figure 1.6 Packet-switching network principles: (b) connectionless

Quality of Service (QoS)

- Network
  - *Availability:* what % of time is it 'available?'
  - *Reliability:* what error/loss rate is allowed?
  - *Delay:* what are max, median propagation times?
- Application
  - *Packet loss rate:* what % of packets can be lost without affecting application performance
  - *Delay Variation* (jitter): how much variation is allowed?
  - *Voice Quality:* measured subjectively using *Mean Opinion Scores (MOS)*

1.3 Communications basics

- Data is sent from / received by an *interface* on a device (e.g. a PC)
- It may be sent directly, using *baseband* transmission, or it may be mixed with a carrier signal, i.e. sent using *modulated* transmission
- The time taken to transmit one bit ('0' or '1') is called the *bit cell period*. Within each such period, a receiver must decide whether the incoming bit is '1' or '0'

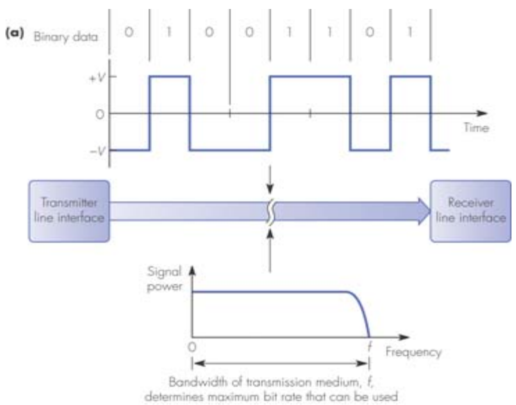


Figure 1.8 Modes of transmission: (a) baseband transmission

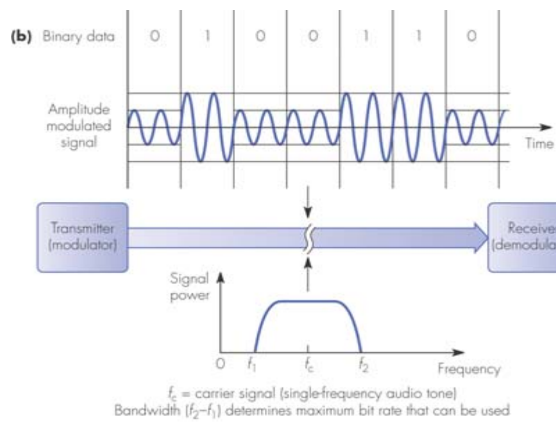


Figure 1.8 Modes of transmission: (b) modulated transmission

Copper media

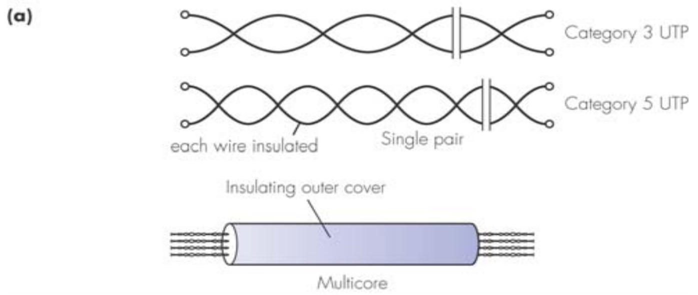


Figure 1.10 Copper wire transmission media: (a) unshielded twisted pair (UTP)

Optical media (fibre)

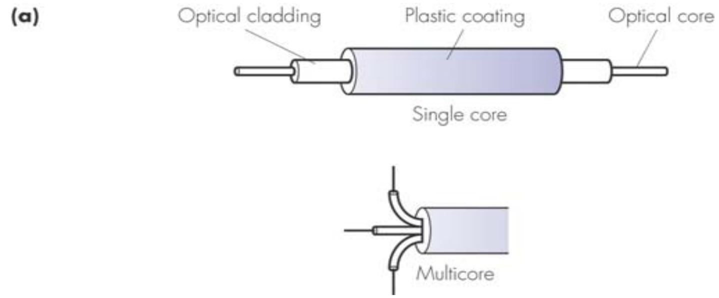


Figure 1.11 Optical fiber transmission media: (a) cable structures

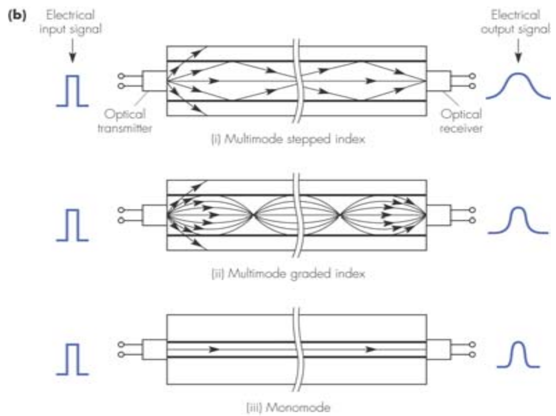


Figure 1.11 Optical fiber transmission media: (b) transmission modes

Important information on transmission of bits

Bits, as electrical signals, always travel at a 'propagation speed' of

- 300,000 km/s in "free space" (radio, satellites, etc) (30cm per nanosecond)
- 200,000 km/s on copper or fibre-optic cables (20cm per nanosecond)

A 'faster' link has the bits arriving *more often* (say 1000 per microsecond, rather than 100 per microsecond),
but they *never travel any faster*.

- The circumference of the Earth is 40,000 km (by the definition of the metre)
- The distance from New Zealand to North America, South America, Japan or Singapore is close to 10,000 km.
- The delay or "latency" from New Zealand to almost anywhere except Australia is at least 1/20 second (50 ms). This delay cannot be reduced!

Distances between bits, on optical fibre

Assume propagation speed of 200,000 km/s in glass fibre
(These distances are nearly correct if the page is printed on A4 paper).

OC-3 (155 Mb/s) 1.33 m — *1.33 m*

OC-12 (622 Mb/s) 32 cm — *32 cm*

1 Gb/s 20 cm — *20 cm*

OC-48 (2.4 Gb/s) 8.3 cm — *8.3 cm*

OC-192 (9.6 Gb/s) 2.1 cm — *2.1 cm*

Asynchronous Transmission

- Transmitter and Receiver clocks must run at (nearly) the same rate
- Character starts when line voltage drops (next slide)

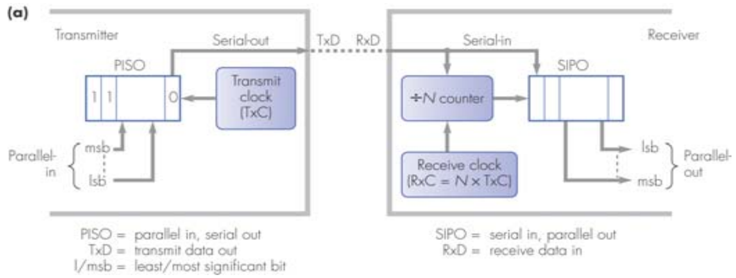


Figure 1.15 Asynchronous transmission: (a) principle of operation

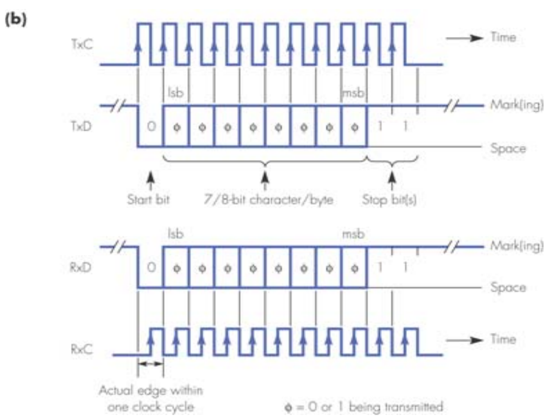


Figure 1.15 Asynchronous transmission: (b) timing principles

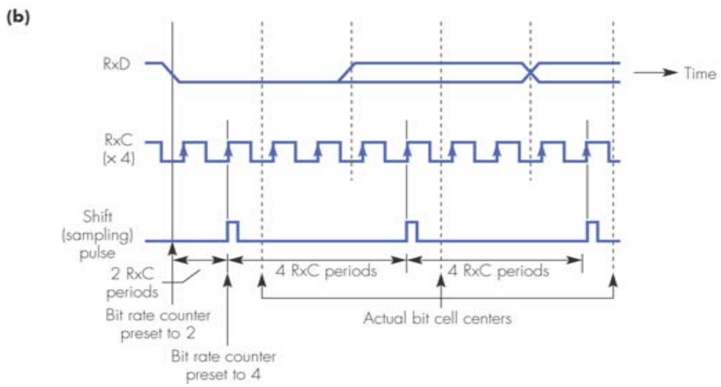


Figure 1.16 Examples of three different receiver clock rate ratios: (b) × 4

Framing (blocks of characters)

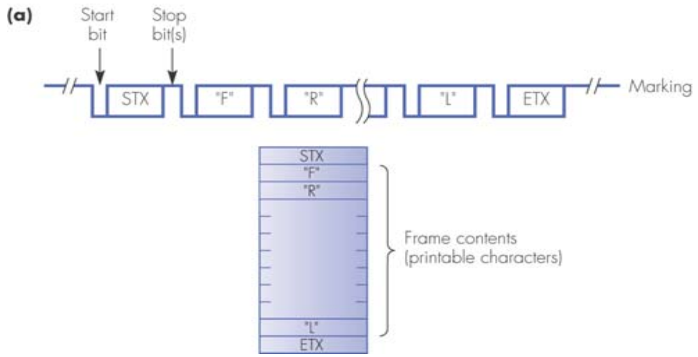


Figure 1.17 Frame synchronization with different frame contents: (a) printable characters

Framing (non-printable characters)

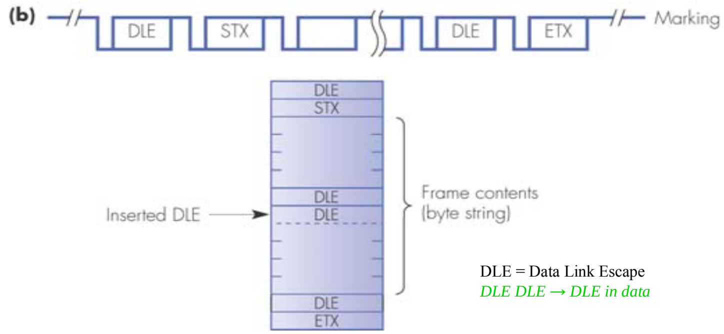


Figure 1.17 Frame synchronization with different frame contents: (b) string of bytes

Synchronous transmission

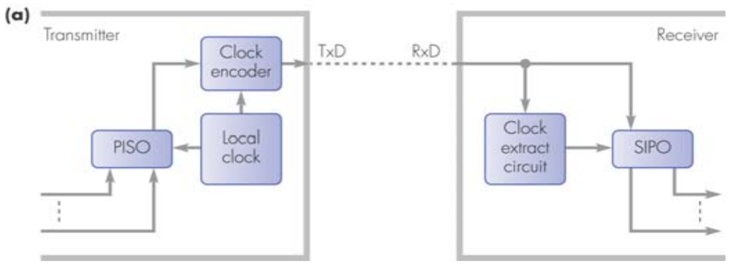


Figure 1.18 Alternative bit/clock synchronization methods with synchronous transmission: (a) clock encoding

Manchester encodings

- Manchester
  - Transition in middle of every bit cell
  - Low-high → 1, high-low → 0
- Differential Manchester
  - Transition at start of cell → next bit is a 0

- Manchester encodings are *balanced*
  - Long-term average value is 0, i.e. no DC component
  - Signals can be AC coupled, simplifies the electronics

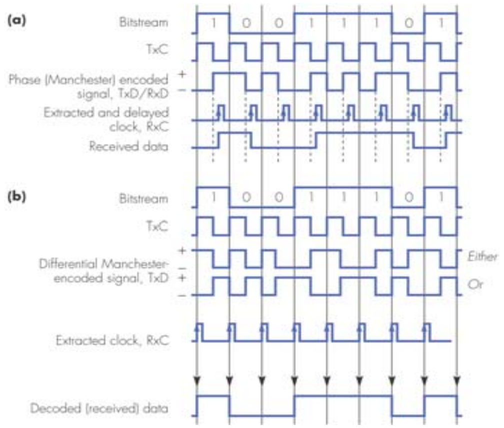


Figure 1.19 Synchronous transmission clock encoding methods: (a) Manchester; (b) differential Manchester

Synchronous *character* framing

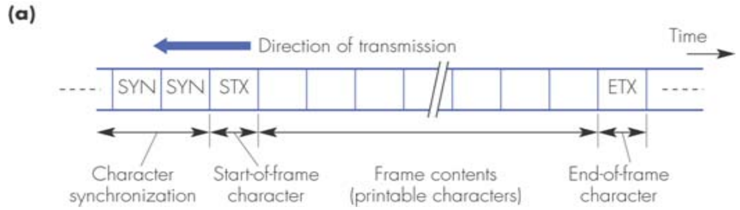


Figure 1.21 Character-oriented synchronous transmission: (a) frame format

Synchronous *character* framing

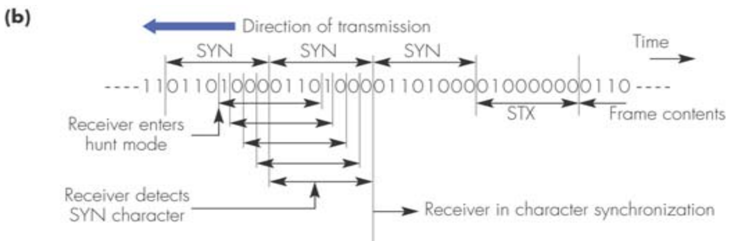


Figure 1.21 Character-oriented synchronous transmission: (b) character synchronization

Synchronous *character* framing

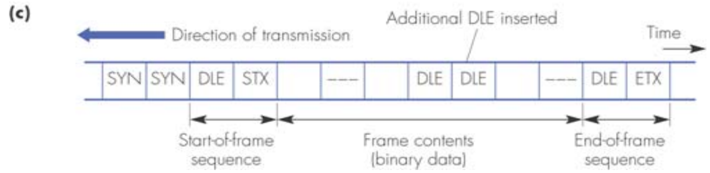
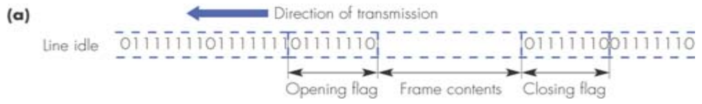


Figure 1.21 Character-oriented synchronous transmission: (c) data transparency (character stuffing)

Synchronous *bit* framing



- Flag has 6 consecutive 1 bits, other characters have 5 or less
- Add 0 bits after 5 consecutive 1s *within frames*

Figure 1.22 Bit-oriented synchronous transmission: (a) framing structure

Synchronous *bit* stuffing

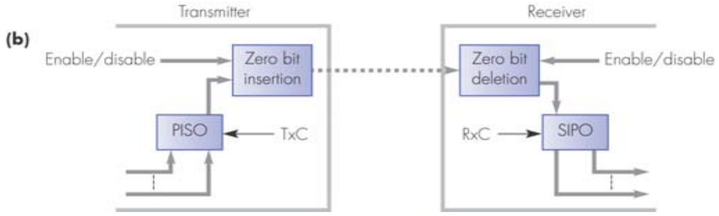


Figure 1.22 Bit-oriented synchronous transmission: (b) zero bit insertion circuit location

1.4 Protocol basics

- Messages (frames) are sent out on the network
- They may suffer transmission errors, or may be lost in the network
- An error control protocol describes how the sender and receiver must co-operate in order to transmit the messages reliably
- The simplest type of error control protocol is *Automatic Repeat Request* (ARQ)
- We begin with Idle RQ …

Idle RQ protocol

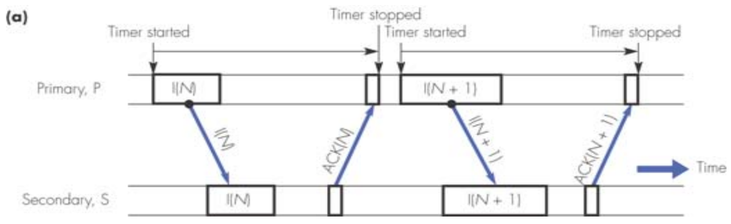


Figure 1.23 ARQ error control scheme: (a) error free

Idle RQ protocol (2)

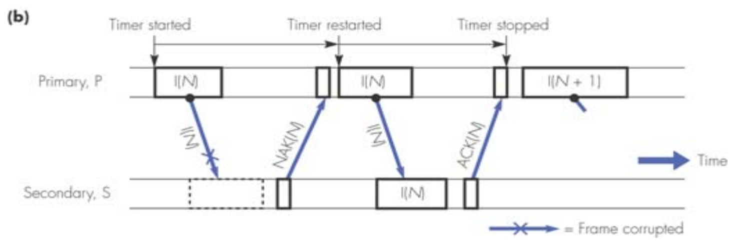


Figure 1.23 ARQ error control scheme: (b) corrupted I-frame

Idle RQ protocol (3)

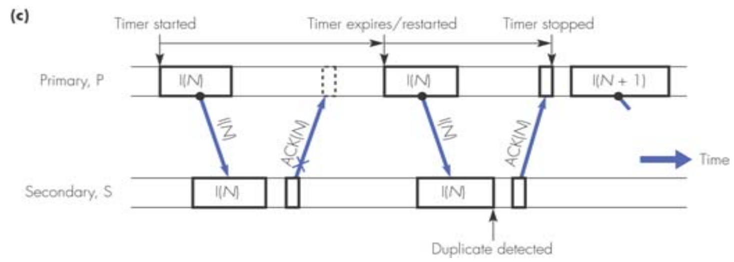


Figure 1.23 ARQ error control scheme: (c) corrupted ACK-frame

Idle RQ protocol (4)

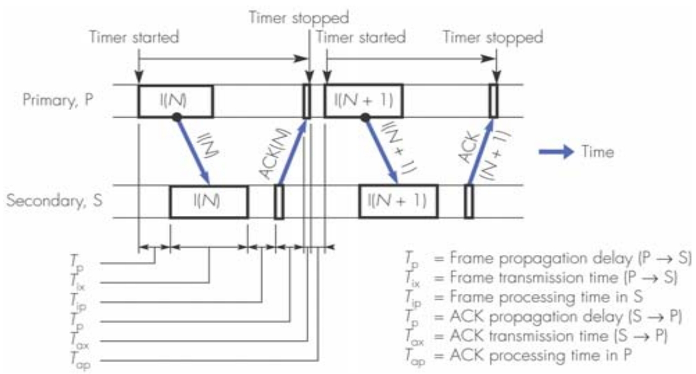


Figure 1.24 Idle RQ link utilization

Idle RQ link utilisation

- Frame transmission time is proportional to the length of the total data transmitted (user + reply), including overheads (header + trailer) and is inversely proportional to the bit-rate on the link
  - For long messages it contributes a time $(U+A)/R$
  - For very short user messages the overheads $2(H+T)$ may become important
  - Choosing a 'faster link' affects only this time, allowing the data to be put into the link at more bits per second
  - A faster link NEVER affects the transmission velocity or link latency; this is something which we just have to live with
- The *frame propagation delay* or *link latency* is dependent only on the distance or link length, D
  - The transmission velocity V is almost always
    300,000,000 m/s (3×10^8 m/s, the speed of light c) for radio
    200,000,000 m/s (2×10^8 m/s, about $2/3c$) for cables (either copper or optical)
  - For transoceanic links the latency can be a very important value indeed

Idle RQ link utilisation example

Assume:

User data size U = 1000 bytes = 8000 bits
Header/trailer overhead (H+T) = 30 octets = 240 bits
Reply message size = 30 octets = 240 bits
End to end cable length D = 1000 metres
Signalling data rate R = 10 Mb/s (10^7 b/s)
Signal velocity in cable V = 2×10^8 m/s

Then

End-to-end latency = 5 µs
Time to send message = $8 \times 1030/10^7$ = 824 µs
Time to send reply = $8 \times 60/10^7$ = 48.0 µs
Total time to send 1000 user bytes
(send + outward + reply + reverse) = 882.0 µs,
Effective user data rate = 1.134 byte/µs

Compare with naive prediction: $10^7/8$ = 1.25 byte/µs (≈ **9.3% reduction**)

Link utilisation - comments

- A longer link, faster signalling rate and smaller packet can give a marked reduction in performance compared with the raw link speed, for ARQ protocols
- Extreme precision is seldom needed or even appropriate in these examples. Packet or data sizes vary widely and cable velocities are seldom known to better than ±1% anyway
- What's important is that you can give a good estimate of the effects of latency, packet overheads, etc

- *The Earth's circumference is 40,000 km (original definition of the metre)*
- *The distance from New Zealand to North America, South America, Japan or Singapore is close to 10,000 km*
- *The delay or 'latency' from New Zealand to almost anywhere except Australia is at least 1/20 second (10,000 / 200,000 = 50 ms)*

Things to do in a network

1. Transmit bits from one place to another (Physical)
2. Assemble bits into bytes and messages, check for reliable transmission (Link)
3. Send messages between end-nodes in mesh-type network (Network)
4. In a mesh network, handle lost packets, broken links etc (Transport)
5. Handle extended connections between endpoints, LANs, etc.
6. Resolve differences between data representation in different computers
7. Do something useful (User application)

These are the seven layers of the "Open Systems Interconnection" (OSI) communications model.

TCP/IP (Internet) combines layers 5-7, into a single Application layer

We discuss only layers 1-4