PART IIA 2005
3F5: Computer and network systems
Principal Assessor: Dr F M Stajano

Datasheet: None

Please note that some of these answers, particularly 3 and 4, are much more verbose than those expected from the candidates in the exam.

1 (a) (i) In order of appearance:

$3 = temporary storage to hold the word being transferred from source to dest
$4 = source address (updated as we go along)
$2 = counter of the number of words transferred
$5 = destination address (updated as we go along)
$0 = reserved register holding the constant 0

(ii)
The counter $3 is not initialized. Should be set to 0 before starting.
The source and destination registers $4 and $5 are incremented by 1 at each step, but the addresses of consecutive *words* (not bytes) differ by 4. Should be 4.

Not required for full marks, but extra credit for discussing and possibly fixing the complications that arise if the source and target area overlap.

(iii)

```
        add     $2,$0,$0      # Initialize copied-words counter
back:   lw      $3,0($4)      # Read next word from source
        addi    $2,$2,1       # Increment copied-words counter
        sw      $3,0($5)      # Write word to dest
        addi    $4,$4,4       # Point to next source word
        addi    $5,$5,4       # Point to next dest word
        bne     $3,$0,back    # Repeat unless word was 0
```

(b)

```
(i)  addi $8, $0, int16

(ii) lui $8, upper_half(int32)
     ori $8, $8, lower_half(int32)
```

(iii)
```
li $at, int32
add $at, $at, $9
lw $8, 0($at)
```

(iv)
```
li $at, int32
add $8, $9, $at
```

(v)
```
li $at, int16
beq $8, $at, L
```

(vi)
```
slt $at, $9, $8
beq $at, $0, L
```

(c) The opcode slt stands for "set if less than". The instruction "slt $8, $9, $10" sets $8 to 1 if $9 is less than $10, otherwise it sets $8 to 0.
To simulate this instruction in its absence, the obvious route would be to perform the subtraction $9-$10 and examine the sign of the result, setting $8 iff negative. However this subtraction could result in overflow (e.g. if we had largePos – largeNeg). So we must resort to a more complex sequence.

If $9 and $10 have the same sign, subtracting one from the other will never overflow, so it's OK to perform the subtraction.

If they have opposite signs, we might have overflow if we subtracted one from the other but we don't have to do that, because we can say straight away that the negative one is smaller.

To test the sign of a register, mask off everything but the MSB with an AND. If what's left is zero, the register was positive; else, negative.

Pseudocode to replace slt is therefore as follows:

```
if (($9 < 0) and ($10 < 0))    or    (($9 > 0) and ($10>0)):
        $at := $9-$10
        if $at < 0:
                $8 := 1
        else:
                $8 := 0
else:
        if $9 < 0:
                $8 := 1
        else:
                $8 := 0
```

2 (a)

(i)

word = a unit of data storage as wide as the registers of the machine (4 bytes for MIPS). Usually also the width of the data bus, at least in well-balanced architectures.

block = a sequence of consecutive words, always loaded together as a unit by the cache. The block size is a power of 2 and blocks are always aligned in memory, ie the addresses of all the words in a block have the block number as a common prefix in their high order bits.

Set = one of the units of storage into which the cache is partitioned. Each set in the cache may contain several blocks. A given block address uniquely determines a set, but within the set the block may freely be stored anywhere.

LRU = least recently used. When deciding where in a set to place a block, the LRU strategy will choose to overwrite (i.e. throw out of the cache) the block that was accessed least recently.

How it works: as hinted at above while defining "set" and "LRU", in a set-associative cache each block maps to a well-defined set in the cache (in our example, blocks with an even block address go to set 0, while blocks with an odd block address go to set 1), but within the set the block doesn't have a fixed place. This allows for more efficient cache usage than a direct-mapped cache, in which the block address uniquely determines the location, because two frequently used blocks might otherwise keep on kicking each other out. A fully associative cache would give total flexibility (any block can go anywhere in the cache) but its implementation would be much more costly, especially in hardware.

(ii)

```
           -----set0---     -----set1---
           -- --  -- --     -- --  -- --
12 miss    12 13
13 hit     (   )
14 miss                     14 15
15 hit                      (   )
24 miss           24 25
13 hit     (   )
12 hit     (   )
13 hit     (   )
14 hit                      (   )
20 miss           20 21
22 miss                            22 23
20 hit            (   )
12 hit     (   )
24 miss           24 25
25 hit            (   )
12 hit     (   )
13 hit     (   )
14 hit                      (   )
15 hit                      (   )
20 miss           20 21
```

(iii)

```
Final:     12 13  20 21     14 15  22 23
13 hits,  7 misses
```

(b) In a single-cycle-datapath machine, each instruction takes one clock cycle. The clock period must be as long as required by the slowest instruction in the instruction set.

In a multicycle-datapath machine, each instruction is broken into smaller steps, each of which takes one clock cycle (with a much shorter clock period). This allows a functional unit such as the ALU to be used more than once per instruction, so long as this happens in different clock cycles. Moreover, different instructions may use a different number of small steps.

In a pipelined machine, each instruction is broken into the same number of small steps, each of which takes one clock cycle. The datapath is broken into pipeline stages, all operating concurrently. At each clock cycle, each pipeline stage executes one step of a different instruction, so as a whole the machine is executing (parts of) several instructions at the same time.

(c) (i) Strictly speaking, this instruction does not fit any of the three MIPS formats of R (register), I (immediate) and J (jump). The closest match is the R format. This has space for a 6-bit opcode, three 5-bit register specifications, one 5-bit shift amount and a 6-bit function field. Here, we need to specify 4 registers so we could use the shift amount to hold the 4th register.

This means we can no longer specify a shift value for this instruction, but that's not a problem because in MIPS that field is only used by explicit shift instructions, unlike what happens in other architectures such as ARM where even non-shift instructions can have a shifting modifier. If this had been ARM, with this modification we would have broken the regularity of the instruction set.

(ii) The instruction will require using the ALU twice. A multiplexer should be added to permit the ALU to use the previous result produced by the ALU (stored in ALUOut) as an input for the second addition. Changes will also be needed for register read. The third register source may be read during the cycle in which the first addition is taking place. A multiplexer and a new control signal will be needed for the inputs to one of the register read ports.

3 (a) For successful data communication across a network, appropriate operating procedures must be established. They must be specified in detail and strictly adhered to by the sending data terminal (or computer) and any intervening switching centres. These procedures are called protocols.

Many local area networks (LANs) interconnect data terminals (or computers) from the same manufacturer and operate using proprietary protocols. The need arose for communication between computers and terminals from different manufacturers. Open systems interconnect (OSI), to enable networks to be machine independent. The ISO standards are based on a seven layer protocol known as the ISO reference model for OSI. Conceptually, the layers in the OSI model can be considered as performing one of two overall functions:

- Network dependent functions
- Application oriented functions.

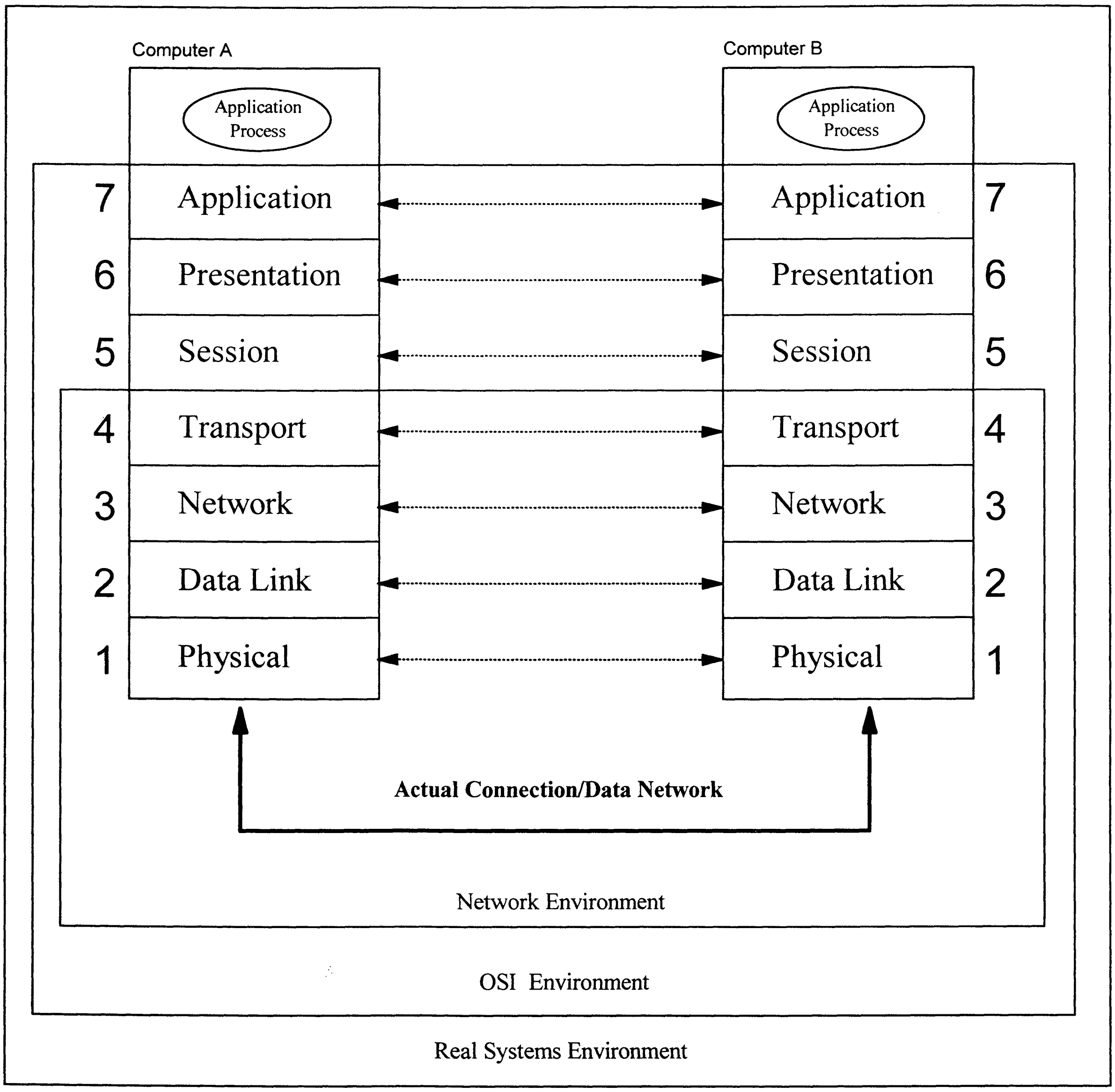
Both the network dependent and application oriented (network independent) components of the OSI model are in turn implemented in the form of a number of protocol layers. The boundaries and processes for each layer have been selected based on experience gained from other standards in the past.

The OSI model can be broken up within this system into 7 separate layers, each with a clearly defined purpose and protocol.

- Each layer is a service user to the layer below and a service provider for the layer above.
- Each layer is specified independently of the other layers;
- Each layer has a defined interface with the layer above and below.

The lowest three layers (1-3) are network dependent and are concerned with the protocols associated with the data communication network being used to link the two computers. In contrast, the upper three layers (5-7) are application oriented and are concerned with the protocols that allow the two end user application processes to interact with each other.

As far as users are concerned, communication appears to take place across each layer. Each data exchange passes down to the bottom layer (the physical layer) at the sending terminal, crosses the network to the receiving terminal and then passes up again. Data communication between layers is implemented through the addition and reading of headers on the data



b) ***Layer 1: The physical layer***. This defines an interface in terms of the connections, voltage levels and data transmission rate, in order for signals to be transmitted bit by bit. The function of the physical layer is to provide a 'virtual bit-pipe' service to the data link layer of the OSI reference model. At the lower interface of the physical layer, we have the physical communications channel. This may be using any of the physical media available for transmission of data.

Whatever the transmission media, the physical layer has to shield the data link layer from it and must adapt the output from the data link layer to a form that best fits the transmission medium.

A **coaxial cable** consists of a stiff copper wire as an inner conductor, inside a solid insulator, that is itself inside a closely woven braided wire mesh that acts as an outer conductor. This is then covered in an insulating protective cover. The solution of Maxwell's equations also shows that coaxial cables have:

- A high frequency cut-off
- A high immunity to external interference and crosstalk.

Coaxial cable is a good transmission medium for high data rates over relatively short distances, however it is rather expensive and is not ideally suited to long distance transmission and has been largely replaced by optical fibre.

It is used as the basis of most 'cable-TV' (CATV) systems as it has a high bandwidth and is much easier to join and terminate that optical fibres. It was also extensively used in Ethernet, bus type systems where many nodes can be tapped into a single coaxial backbone.

Almost every home contains **twisted pair** cabling as normally used with a telephone connection. The bandwidth of this cabling is severely limited by its length and the way in which it is used within the telecommunications networks. Twisted pair cables are classified in two types, Unshielded twisted pair (UTP), Shielded twisted pair (STP).

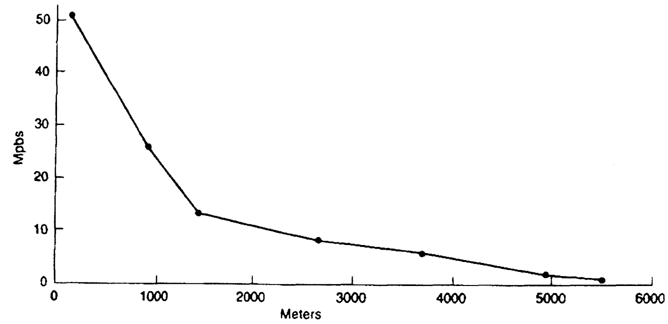
Twisted pair consists of two insulated copper wires twisted together. Any pair of wires held close together will have an associated capacitance which is susceptible to outside interference or crosstalk. By twisting the pair of wires, we introduce mutual inductance, that balances out the effect of the capacitance. By adding more twists per unit length, we can improve the quality of the cable, hence data grade cable has more twists and higher bandwidth than voice grade cable, but is more expensive. The quality of twisted pair cable is divided into 5 categories for UTP and STP. The higher the category of twisted pair wire, the higher the quality and cost.

The available bandwidth is also a function of the wiring length, with 100Mbit/s in category 5 cable only being transmitted up to 100M in new high-speed local area networks (LANs). Care must also be taken at these data rates as the wiring starts to resemble aerials and can pick up other channels along with Q103 FM.

The main limitation with using existing telephone twisted pair cabling: The length of the cable and The filtering applied at the exchange. With modern digital telephone networks, the analogue signal is sampled and converted to a digital bit stream. To avoid aliasing, a low pass filter set at 3.4kHz is applied. Even without the filters, the data rate is restricted to 4kHz as the sampling rate of the digital system is 8kHz.

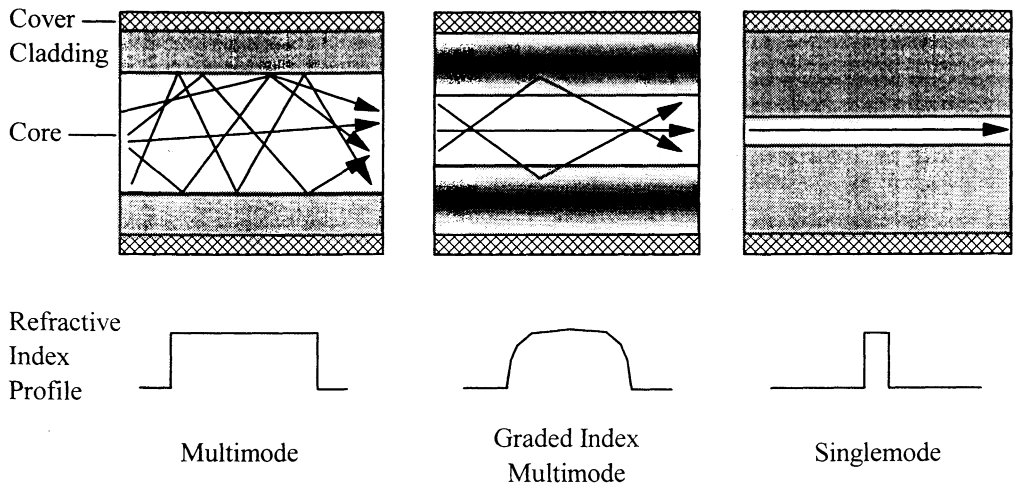
These restrictions can be avoided in two ways: A modem allows a data rate that exceeds the 3.4kHz data rate set by twisted pair wiring and digital telephone network. This is done using clever modulation schemes other than direct binary amplitude or phase down the modem channel.

The main limitation on the bandwidth of a Cat 3 UTP telephone line is filter placed at the exchange end before the voice signal is digitised. This limitation means that a normal modem must squeeze all of the data through a 3.7kHz wide band. The Cat 3 UTP is in fact capable of transmitting data over short distances at frequencies in excess of 1.1MHz, which is a property exploited by digital subscriber line modems which bypass the filter in the line card at the exchange and access all of the available line bandwidth.



A further enhancement on modem technology is the digital subscriber line (DSL) which allows the transmission digital data at a very high rate over standard twisted pair cables up to a distance of around 5km. This is often referred to as a 'broadband' service by Telcos such as NTL and BT. There are a variety of standards such as VDSL, HDSL and xDSL, but one of the more common is the asymmetric digital subscriber line (ADSL) which typically offers a 512bit/sec downstream and 64kbit/sec upstream or 1Mbit/sec downstream, 256kbits/sec upstream for premium services (eg BT and NTL).

Aside from being small and fragile, ***single mode optical fibre*** is the perfect transmission medium offering an attenuation of less than 1dB/km over a wavelength range of 700nm. This means that we could modulate an optical carrier at data rates beyond 100Tbits/sec. Optical fibres are immune to external interference and do not produce any radiation that could cause interference.



Single mode (SM) fibre is extensively used for long haul telecommunications. Almost all of the digital telephone networks in the UK uses silica based single mode optical fibre. Silica SM fibre has very low attenuation per unit length, especially at the two main comms wavelengths, 1310 and 1550nm.

It also allows the use of all optical (Erbium doped) fibre amplifiers instead of regenerators. The modulation rate is limited by frequency chirp in the laser and dispersion in the optical fibre leading to inter-symbol interference. Single mode fibre is not used over short distances in LANs, due to the expense of the optical modulators (lasers) and amplifiers and the expense and difficulty of connecting and terminating the fibre. Single mode optical fibre transmission will take advantage of the high bandwidth of the optical fibre by using multiple wavelengths down the fibre and wavelength division multiplexing (WDM).

c) Layer 3: Network layer. This is concerned with the operation of the network between the terminals. It is responsible for establishing the correct connections between the appropriate network nodes, including network routing (addressing) and in some instances, flow control across the computer to network interface.

<table>
<tr><td colspan="2">Version</td><td colspan="2">Internet header length (IHL)</td><td colspan="2">Type of service</td><td>Total length</td></tr>
<tr><td colspan="3">Identification</td><td colspan="2">Flags</td><td colspan="2">Fragment offset</td></tr>
<tr><td colspan="2">Time to Live</td><td colspan="3">Protocol</td><td colspan="2">Header checksum</td></tr>
<tr><td colspan="7">Source address
Destination address</td></tr>
<tr><td colspan="6">Options</td><td>Padding</td></tr>
<tr><td colspan="7">Data</td></tr>
</table>

- The **version** indicates the format of the IP header, specifically which version of the protocol is in use (now version 4, soon version 6). This is non-OSI layer 3 as it infers a knowledge of higher layer functions.
- The **internet header length** (IHL) indicates the length of the header in 4 byte (32 bit) words. This is OSI layer 3
- The **type of service** indicates how the datagram will be switched and otherwise treated during its transit through the network. Very non-OSI layer 3 as it infers a knowledge of higher layer functions. Could be simplified to aid routing and nothing else.
- The **total length** indicates the total length of the datagram, including both header and data. This is not strictly OSI layer 3, more suited in layer 2
- The **flags** are three bits indicating whether fragmentation of the datagram is allowed or not. This is not OSI layer 3, more suited in layer 2 or layer 4 (depending on the sequencing system used).
- The **fragment offset** indicates the position of the associated datagram fragment in the overall message. This is not OSI layer 3, more suited in layer 4
- The **time to live** is the remaining time that the datagram is allowed to exist within the internet. When time is up, the datagram is destroyed. This is OSI layer 3
- The **protocol** indication determines whether TCP or UDP protocol is used in the next higher layer, and therefore determines to which protocol agent, the datagram should be delivered. is not OSI layer 3, more suited in layer .
- The **header checksum** is akin to a cyclic redundancy check (CRC) or frame check sequence (FCS). It is only applied to the IP header and is recomputed each time the header is amended during passage through the network. This is not strictly OSI layer 3, more suited in layer 2.
- The **source** and **destination addresses** are both 32 bits. It is normal to write these addresses as four octets separated by dots (eg. 169.129.24.88). Each of the four decimal values may only have a value between 0 and 255. This is OSI layer 3
- A number of **options** may be included in the header including: Most will aid layer 3, except the first one)

    Security (encryption)

    Loose source and record route (the source provides information to aid the routing across the network).

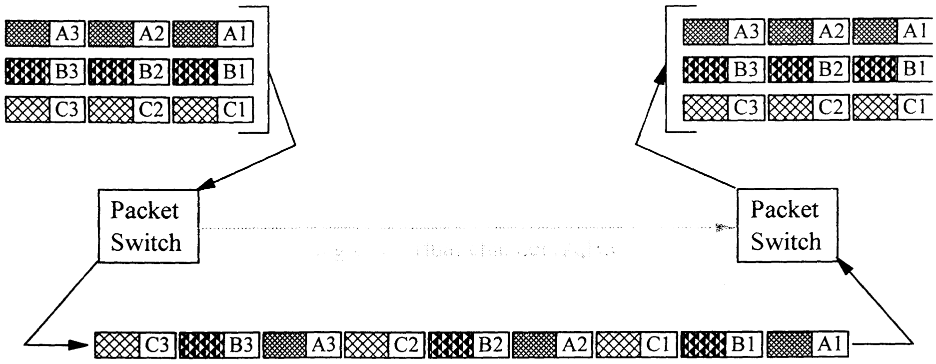
    Strict source and record route (the source is able to give to steer the route taken by the datagram across the network).

    Record route (a record of the route taken through the network)

    Timestamp (the time an date when a datagram was handled by a router)

4 (a) Packet switching is so called because the user's overall message is broken up into a number of smaller packets, each of which is sent separately. Each packet is labelled to identify its intended destination and protocol control information (PCI) is added

The receiving end re-assembles the packets in their proper order, with the aid of sequence numbers and the other PCI fields. Each packet is carried across the network in a store and forward fashion, taking the most efficient route available at the time.



Physical connection

Packet switching is form of statistical multiplexing. The entire available bandwidth is grouped as a single high bitrate transmission pipe, and all packets share the pipe. In this way, the entire bandwidth can be used momentarily by any of the logical channels or virtual circuits (VC) sharing the data connection.

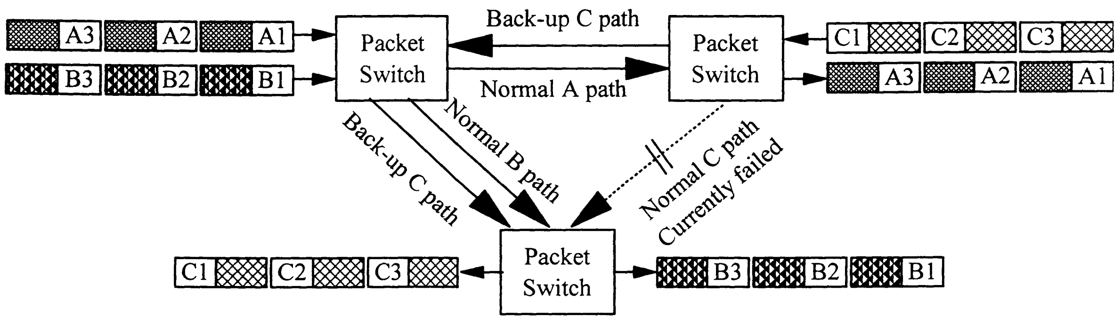
Individual packets are transported more quickly and bursts in transmission can be accommodated. Problems arise when more than one or all transmitters try to send packets at once. This is accommodated by buffers at each end of the connection. These delay some of the simultaneous packets in a first in first out (FIFO) system until the line becomes free. By using buffers, it us possible to run transmission links at close to 100% utilisation. Packet switching is capable of carrying logical channels of almost any bitrate.

When using the trunks in a packet switched network at very close to full utilisation, very large buffers are required for each of the logical channels. It is crucial to ensure that the delays do not become to long. The chance of a long delay is much greater when close to 100% utilisation of the line is expected.

A certain amount of queuing delay caused by buffering is not noticeable to computer users (1/4 of a second is a very long queuing delay). If the average delay becomes much longer, then computer work may become frustrating. There is an entire statistical science (queuing theory) used to estimate queuing delays. The most important formula is the Erlang call waiting formula. The unacceptability of long queuing delays is such that trunks are not utilised at 100% capacity. A typical acceptable utilisation is around 50%. Despite this, packet switched networks are more efficient at carrying data traffic than circuit switched networks.

Packets are routed across individual paths with two techniques, path oriented routing or datagram routing. The route(s) chosen are usually controlled by the (OSI layer 3) software of the packet switch, together with the information pre-set by the network operator.

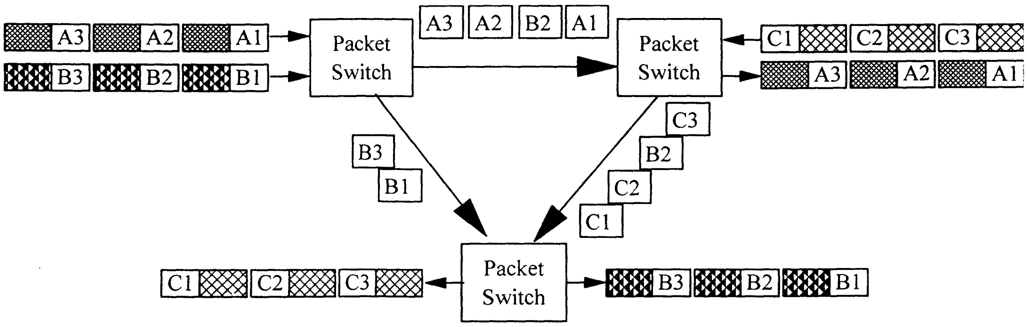
Path oriented routing: A fixed path is chosen for a given logical channel (virtual circuit) at the time of call set-up. The path itself is chosen based on the current loading of the network and the available topology. Should any link in the path become unavailable during the course of the call (say, because of transmission failure), then an alternative path is sought, without breaking the connection. Packets are stored until a new path is found.



The advantage of path oriented routing is that the packets pertaining to a given local connection all take the same path, all suffering about the same amount of queuing delay in the buffers and arrive pretty much in the same order as they were sent (allowing for lost packets along the way). This makes the job of re-sequencing the packets at the receiving end much easier, as well as the job directing the packets through the network. It also leads to more predictable delay performance for the end user or computer application. The packet switched network components can therefore be relatively simple and cheap.

The disadvantage of path oriented routing is inflexibility at high traffic rates. Path oriented switching is a connected system and is used in Frame relay and X.25

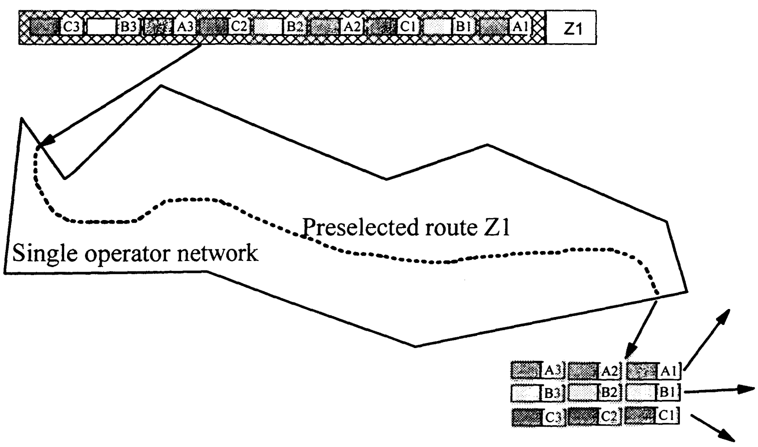
Datagram routing: This allows for more dynamic routing of individual packets and thus has potential for better overall network efficiency. But the technique requires more sophisticated equipment and powerful switch processors capable of determining routes for individual packets. It is used in the Internet and some forms of ATM.



Datagram switching is a connectionless system.

b) Multi-label protocol systems (MPLS). The process of routing packets cannot easily guarantee delay (latency) across the network. This is a problem for voice services across packet switched networks (such as VoIP).

One way to fix this is to group packets through a common network together with a common global header (or label or tag) which gets the group of packets to the other side of the network with minimum delay. This puts a lot of pressure on the routers at the edge of the network to find suitable packet groups.



The individual packets will be routed as per their native protocol outside of the MPLS network, however across the MPLS network, the routing will be path oriented as this minimises delay at each node in the MPLS network. Routing in therefore path oriented.