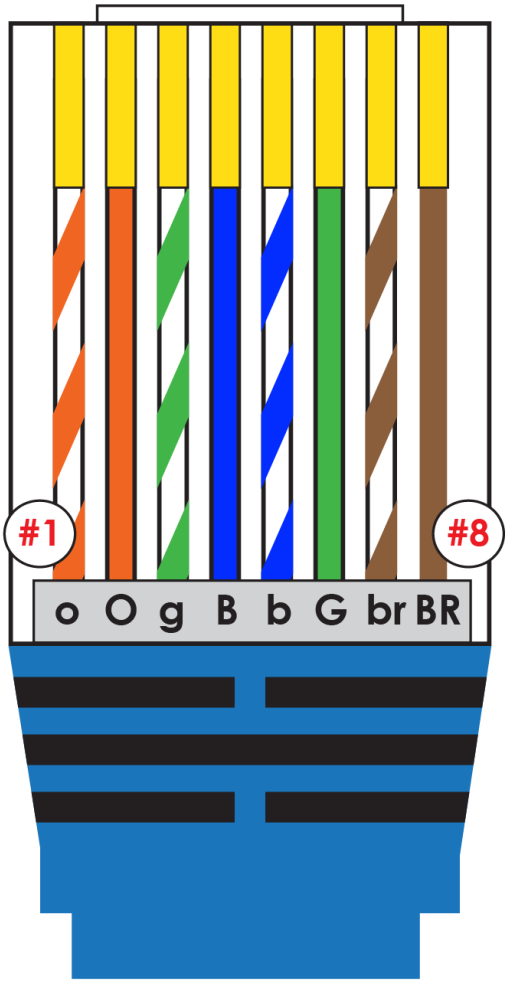
CT3531

NETWORK & DATA COMMUNICATIONS II



T-568B Standard

Andreas Ó hAodha
University of Galway
2023-11-01

Contents

1 Introduction

1.1 Network Classification

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

Figure 1: Classification of Interconnected Processors by Scale

1.2 Reference Models

The **Open Systems Interconnect (OSI) Reference Model** is a network architecture based on a proposal developed by ISO to standardise the protocols used in various layers.

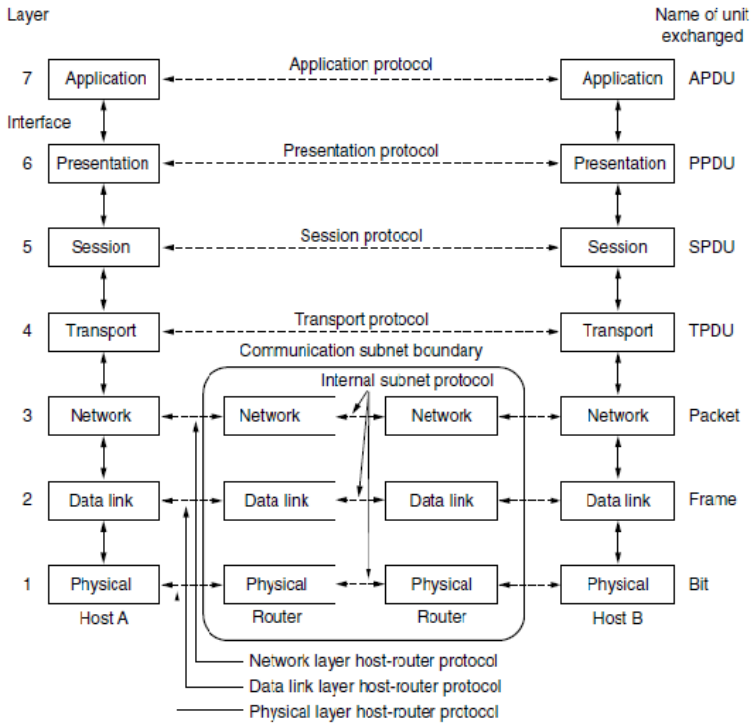


Figure 2: OSI Reference Model

The **TCP/IP Reference Model** is the model used by the Internet, a packet-switching network of networks based on a connectionless internetwork layer.

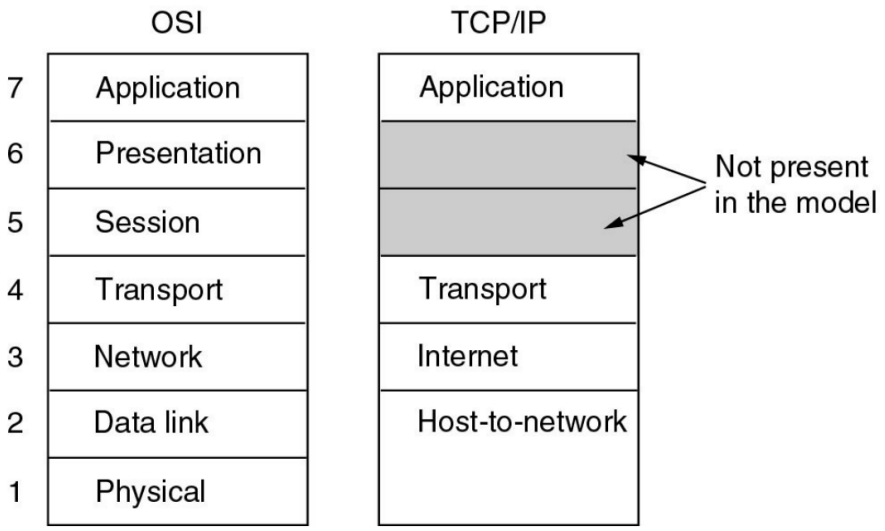


Figure 3: OSI Reference Model

1.3 DNS Name Space

The Internet is divided into over 200 Top-Level Domains (TLDs). Each domain is divided into sub-domains, which are further partitioned, etc. All domains can be represented by a tree: The leaves of the tree represent domains that have no sub-domains (but contain machines). A leaf domain may contain a single host or represent a company and contain thousands of hosts. Top-Level Domains could be generic and country domains.

One DNS server could theoretically service all requests, but in practice would be overloaded. To solve this, the **DNS name space** is divided into non-overlapping zones. Each zone contains some part of the tree & name servers holding zone information. A zone would have a primary DNS which gets information from the disk, and one or more secondary DNS to get information from the primary DNS.

1.4 Fibre Cables

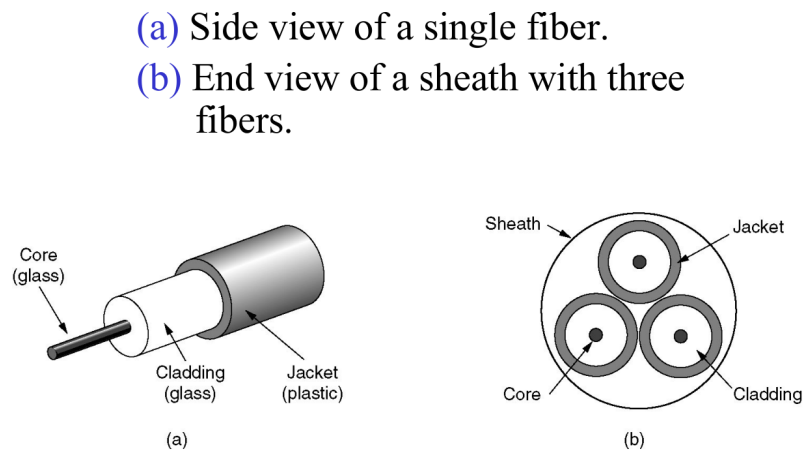


Figure 4: Fibre Cables

Optical networking & Dense Wavelength-Division Multiplexing (DWDM) is rapidly bringing down the cost of networking, and further progress “seems assured”. Butter’s “Law” states that the amount of data coming out of an optical fibre doubles every nine months. Thus, the cost of transmitting a bit over an optical network halves every nine months.

2 Virtual LANs

2.1 Introduction

VLANs logically segment switched networks based on the functions, project teams, or applications of the organisation regardless of the physical location or connections to the network. All workstations & servers used by a particular workgroup share the same VLAN, regardless of the physical connection or location.

Broadcast traffic in LANs is sent to all devices on the LAN, but this can become a problem in large LANs. The traditional solution is to interconnect LANs by IP routers. However, LAN membership of a host is tied to the local switch.

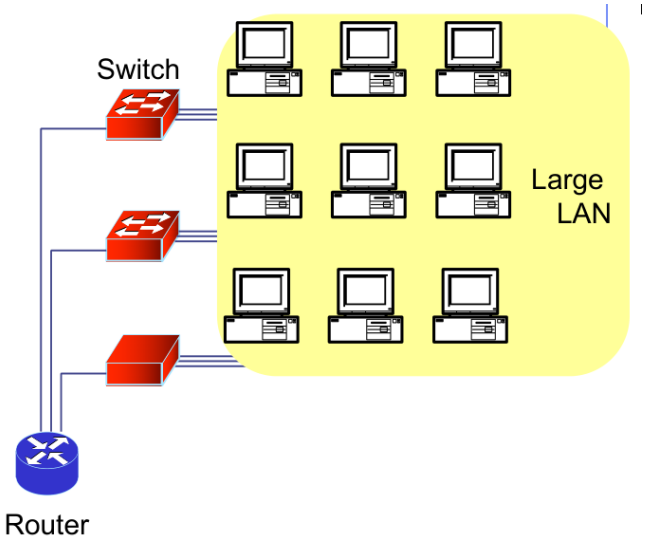


Figure 5: LANs Interconnected by IP Routers

A better solution is VLANs. VLANs separate the broadcast domain from the location of the hosts. This is used to partition large LANs. VLANs are interconnected by IP routers. You can run a separate spanning tree in each VLAN.

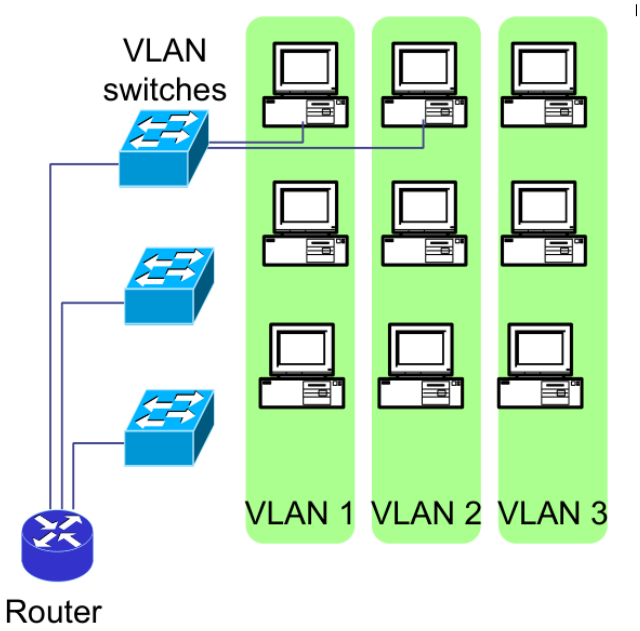


Figure 6: VLANs: The Better Solution

VLANs function by logically segmenting the network into different broadcast domains so that packets are only switched

between ports that are designated for the same VLAN. Routers in VLAN topologies provide broadcast filtering, security, & traffic flow management. VLANs address scalability, security, & network management. Switches may not bridge any traffic between VLANs, as this would violate the integrity of the VLAN broadcast domain. Traffic should only be routed between VLANs.

2.2 Broadcast Domains with VLANs & Routers

A VLAN is a broadcast domain created by one or more switches. A switch creates a broadcast domain, and VLANs help to manage broadcast domains. VLANs can be defined on pot groups, users, or protocols. LAN switches & network management software provide a mechanism to create VLANs.

Layer 3 routing allows the router to send packets to the three different broadcast domains in this example.

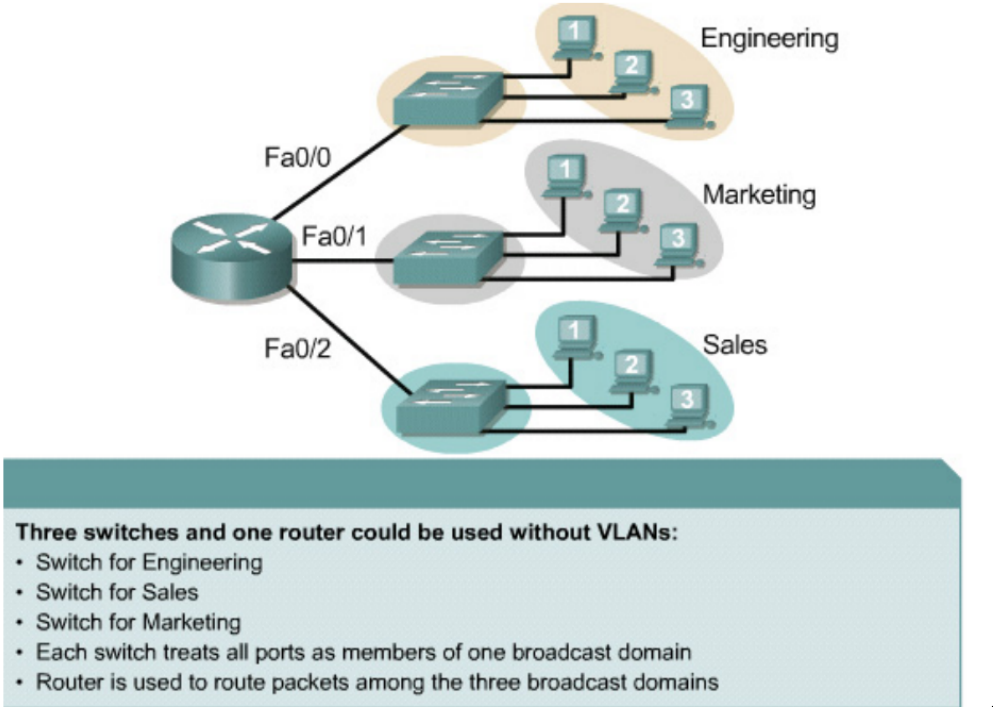


Figure 7: Broadcast Domains with VLANs & Routers

Implementing VLANs on a switch causes the following to occur:

- The switch maintains a separate bridging table for each VLAN.
- If the frame comes in on a port in VLAN 1, the switch searches the bridging table for VLAN 1.
- When the frame is received, the switch adds the source address to the bridging table if it is currently unknown.
- The destination is checked so that a forwarding decision can be made.
- For learning & forwarding, the search is made against the address table for that VLAN only.

2.3 VLAN Operation

Each switch port could be assigned to a different VLAN. Ports assigned to the same VLAN share broadcasts. Ports that do not belong to that VLAN do not share these broadcasts.

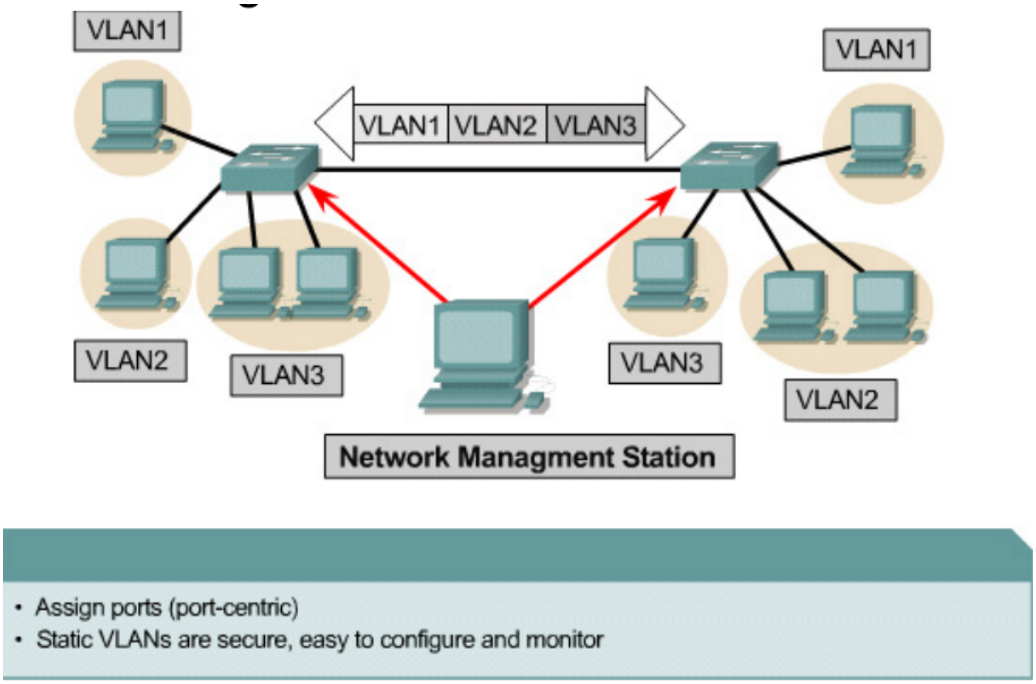


Figure 8: VLAN Operation

Users attached to the same shared segment share the bandwidth of that segment. Each additional user attached to the shared medium means less bandwidth and deterioration of network performance. VLANs offer more bandwidth to users than a shared network. The default VLAN for every port in the switch is the **management VLAN**. The management VLAN is always VLAN 1 and cannot be deleted. All other ports on the switch may be re-assigned to alternate VLANs.

Dynamic VLANs allow for membership based on the MAC address of the device connected to the switch port. As a device enters the network, it queries a database within the switch for a VLAN membership.

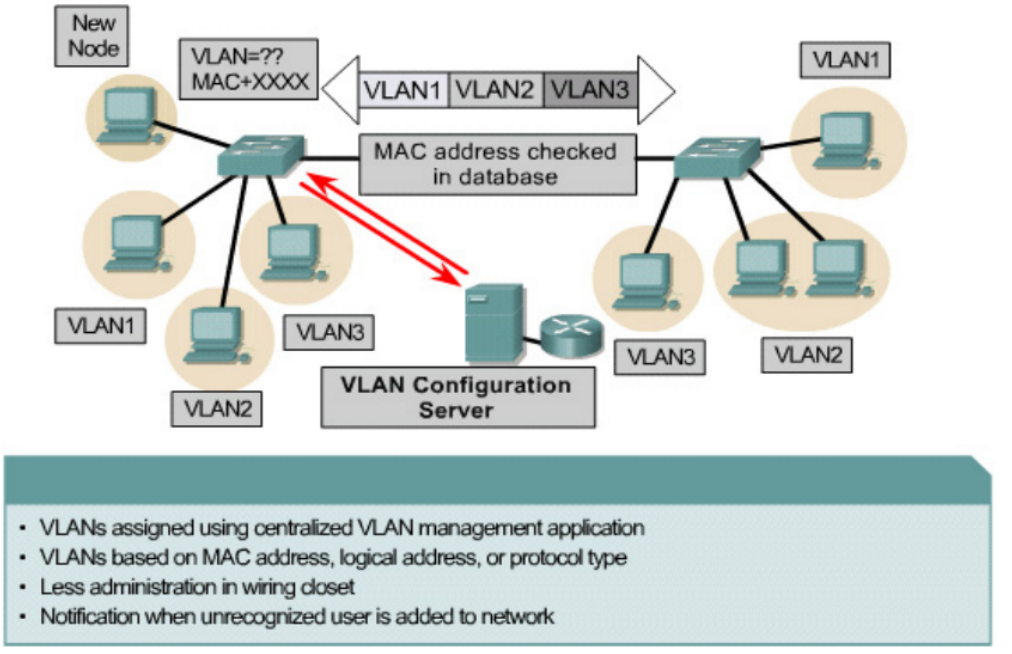


Figure 9: VLAN Operation

In port-based or port-centric VLAN membership, the port is assigned to a specific VLAN membership, independent

of the user or the system attached to the port. All users of the same port must be in the same VLAN.

Network administrators are responsible for configuring VLANs both statically & dynamically. Configuring VLANs **statically** involves the network administrators configuring it port-by-port. Each port is associate with a specific VLAN. The network administrator is responsible for keying in the mappings between the ports & VLANs. Configuring VLANs **dynamically** means that the ports are able to dynamically work out their VLAN configuration. This uses a software database of MAC addresses to VLAN mappings which the network administrator must set up first.

2.4 Benefits of VLANs

The key benefit of VLANs is that they permit the network administrator to organise the LAN *logically* instead of physically.

2.5 VLAN Types

There are three basic VLAN memberships for determining & controlling how a packet gets assigned:

- Port-based.
- MAC address-based.
- Protocol-based.

The frame headers are encapsulated or modified to reflect a VLAN ID before the frame is sent over the link between switches. The frame header is changed back to the original format before forwarding to the destination device.

The number of VLANs in a switch vary depending on several factors, including:

- Traffic patterns.
- Types of applications.
- Network management needs.
- Group commonality.

An important consideration in defining the size of the switch and the number of VLANs is the **IP addressing scheme**. Because a one-to-one correspondence between VLANs & IP subnets is strongly recommended, there can be no more than 254 devices in any one VLAN. It is further recommended that VLANs should not extend outside of the Layer 2 domain of the distribution switch.

2.5.1 Membership by Port

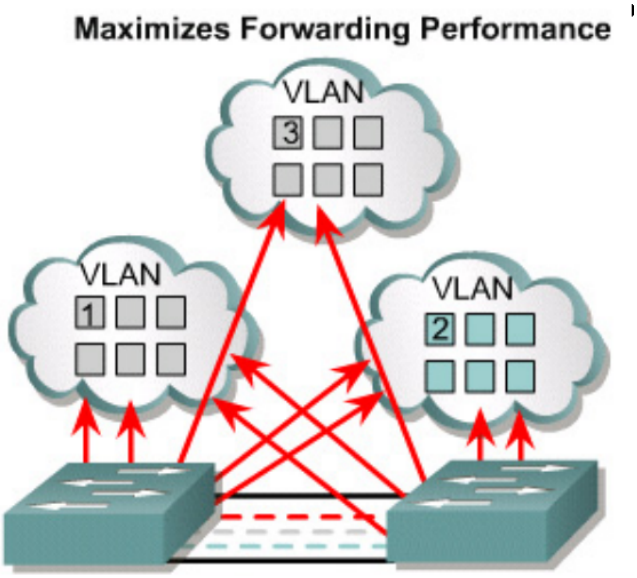


Figure 10: Membership by Port

- User assigned by port association.
- Requires no lookup if done in ASICs.
- Easily administered via GUIs.
- Packets do not “leak” into other domains.
- Easily controlled across network.

2.5.2 Membership by MAC-Addresses

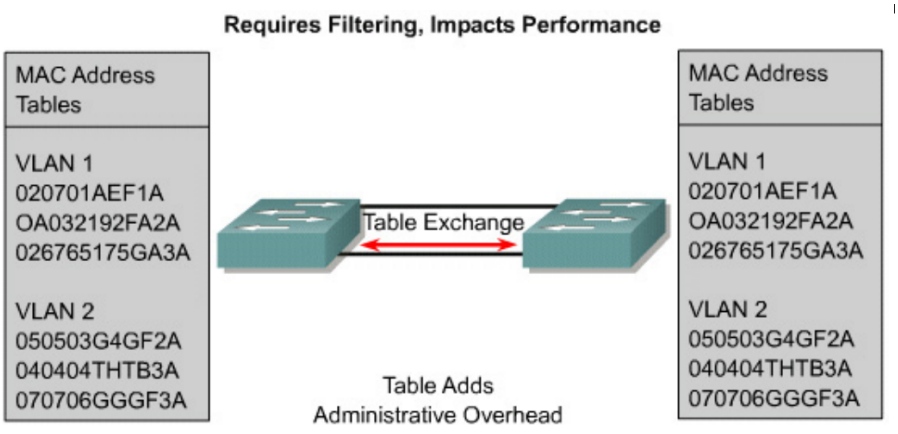


Figure 11: Membership by MAC-Address

- User assigned based on MAC addresses.
- Offers flexibility, yet adds overhead.
- Impacts performance, scalability, & administration.
- Offers similar process for higher layers.

2.6 VLANs Across Multiple Switches

If VLANs span multiple switches, then the traffic between the switches belong to different VLANs. Switches need to be able to demultiplex traffic from different VLANs. **VLAN tags** are used for this purpose.

2.6.1 IEEE 802.1Q: VLAN Tagging

For VLAN traffic between LAN switches, add a tag to Ethernet frames that identifies the LAN. The tag can be transparent to endsystems, by stripping off the VLAN tag.

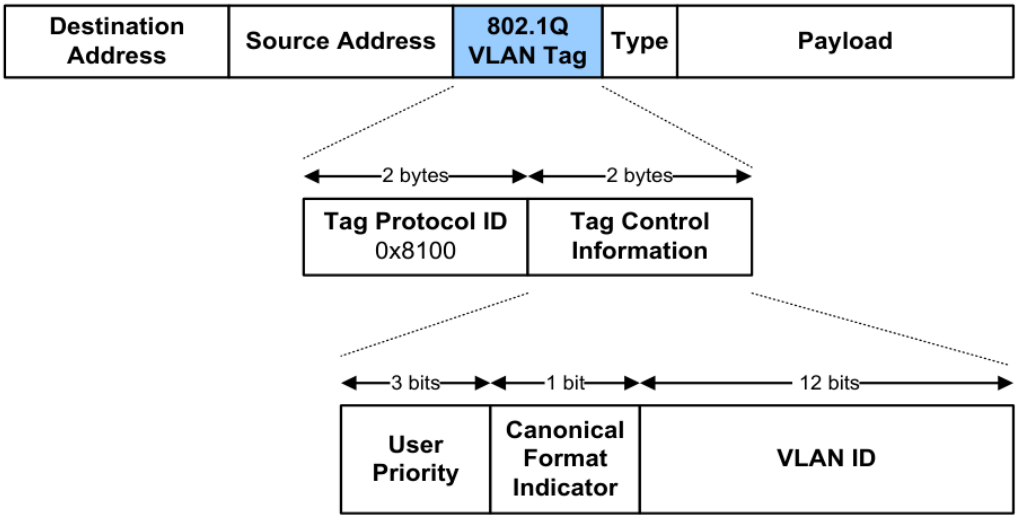


Figure 12: VLAN Tagging

The 802.1Q Tag Fields are:

- **Tag Protocol Identifier:** Value `0x8100` identifies 802.1Q tag.
- **User Priority:** Can be used by the sender to prioritise different types of traffic (e.g., voice, data). 0 is the lowest priority.
- **Canonical Format Indicator:** Used for compatibility between different types of MAC protocols.
- **VLAN Identifier (VID):** Specifies the VLAN (1-4094). `0x000` indicates that the frame does not belong to a VLAN. `0xfff` is reserved.

The normal operation of VLAN tags is as follows:

- Sender sends frame.
- First switch adds tag.
- Last switch removes tag.

2.7 Spanning-Tree Protocols

Bridges & switches use the **Spanning-Tree Protocol (STP)** to avoid loops. Bridges (switches) running STP participate with other bridges in the election of a single bridge as the **Root Bridge**. They calculate the distance of the shortest path to the Root Bridge and choose a port (known as the **Root Port**) that provides the shortest path to the Root Bridge. For each LAN segment, they elect a **Designated Bridge** and a **Designated Port** on that bridge. The Designated Port is a port on the LAN segment that is closed to the Root Bridge. All ports on the Root Bridge are Designated Ports). The bridge ports to be included in the spanning tree are selected. The ports selected are the Root Ports & Designated Ports. These ports forward traffic, while the other ports block traffic.

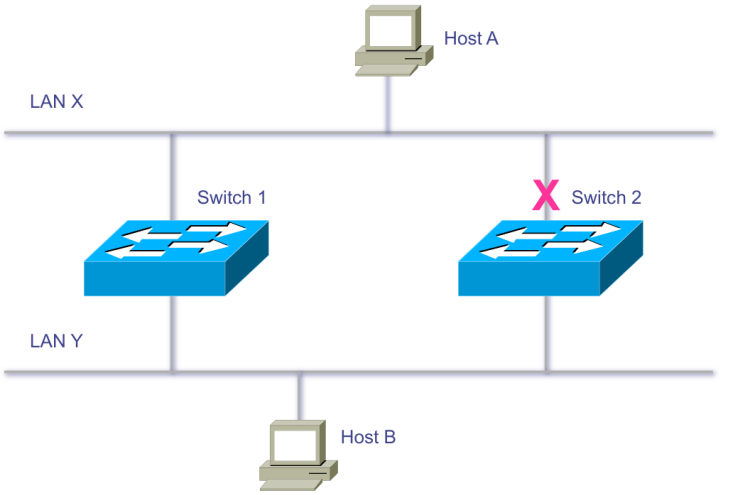


Figure 13: Spanning-Tree Protocols

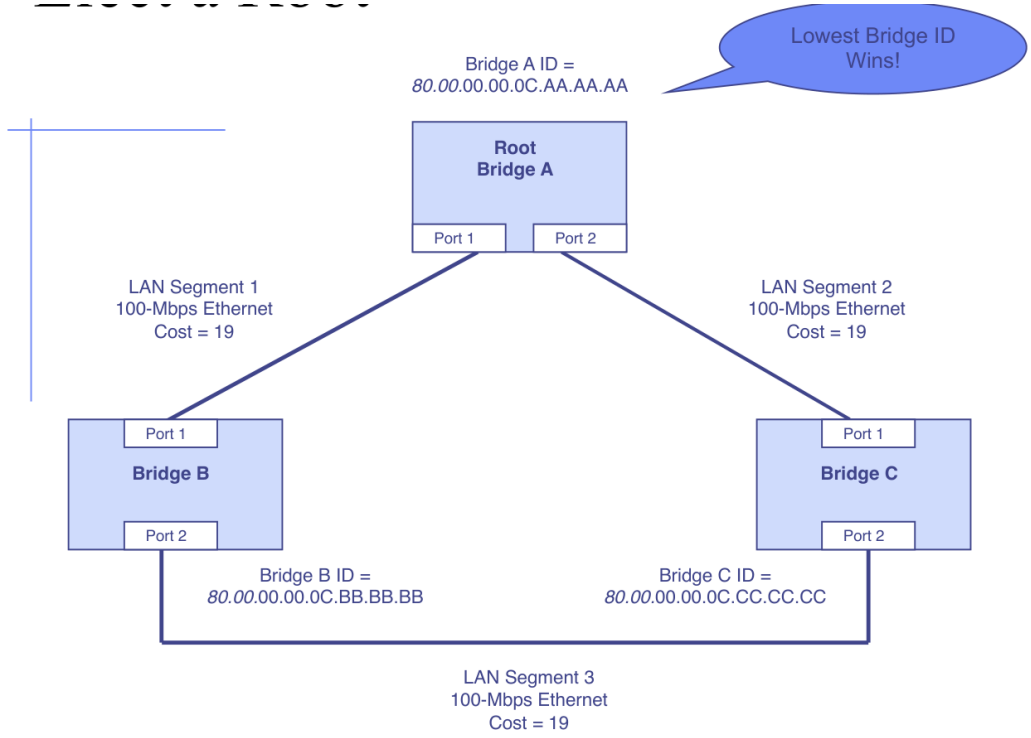


Figure 14: Electing a Root Bridge

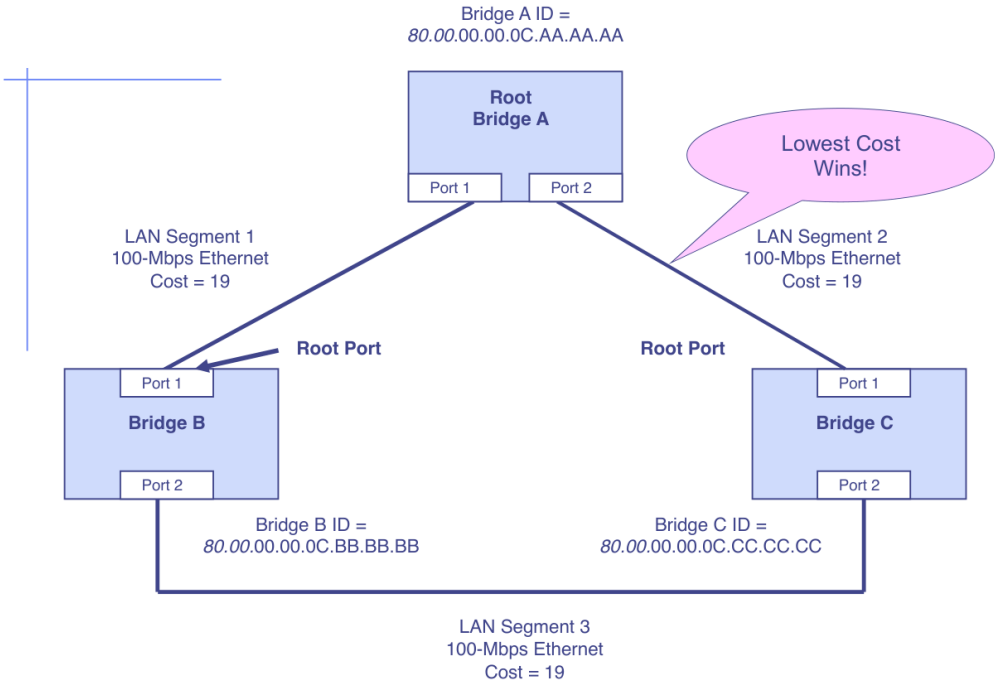


Figure 15: Determining Root Ports

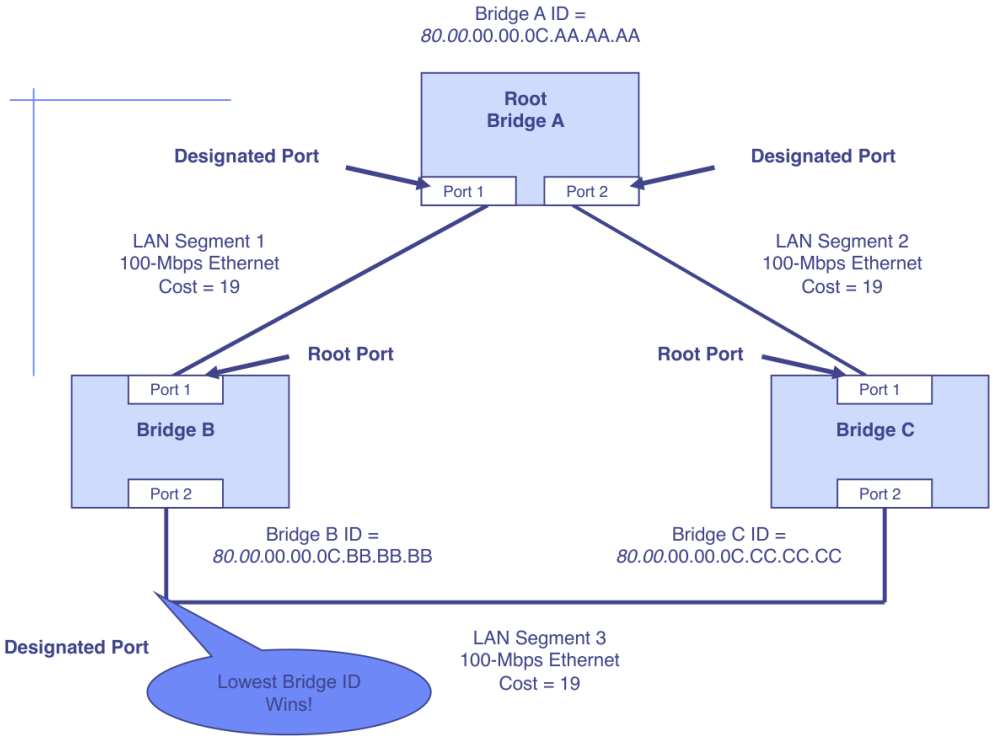


Figure 16: Determining Designated Ports

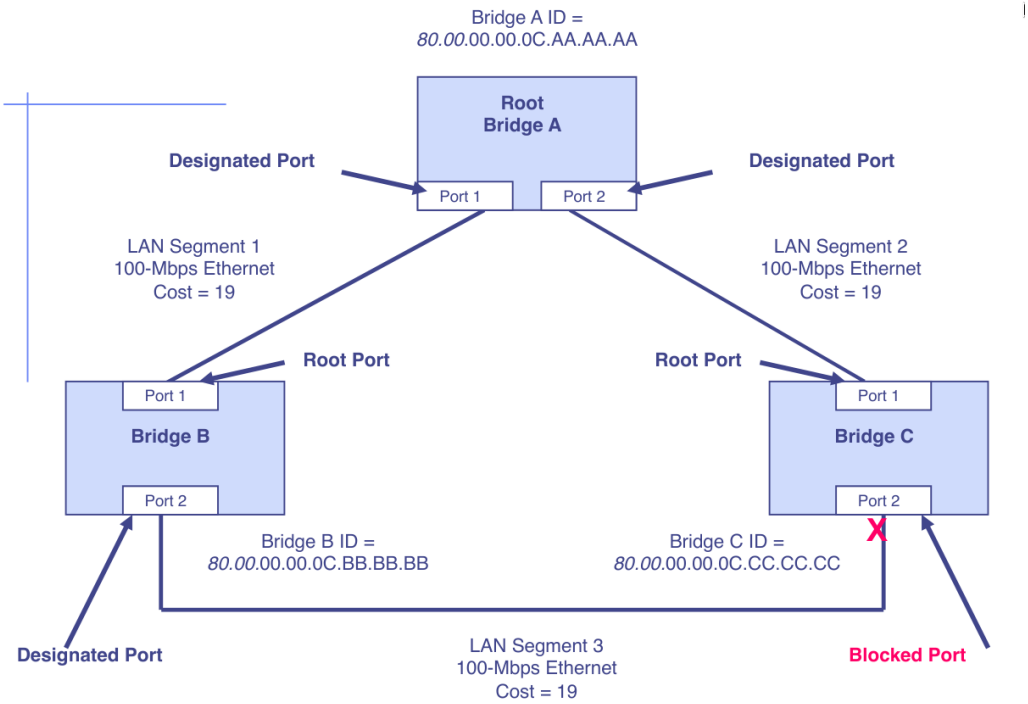


Figure 17: Pruning the Topology into a Tree

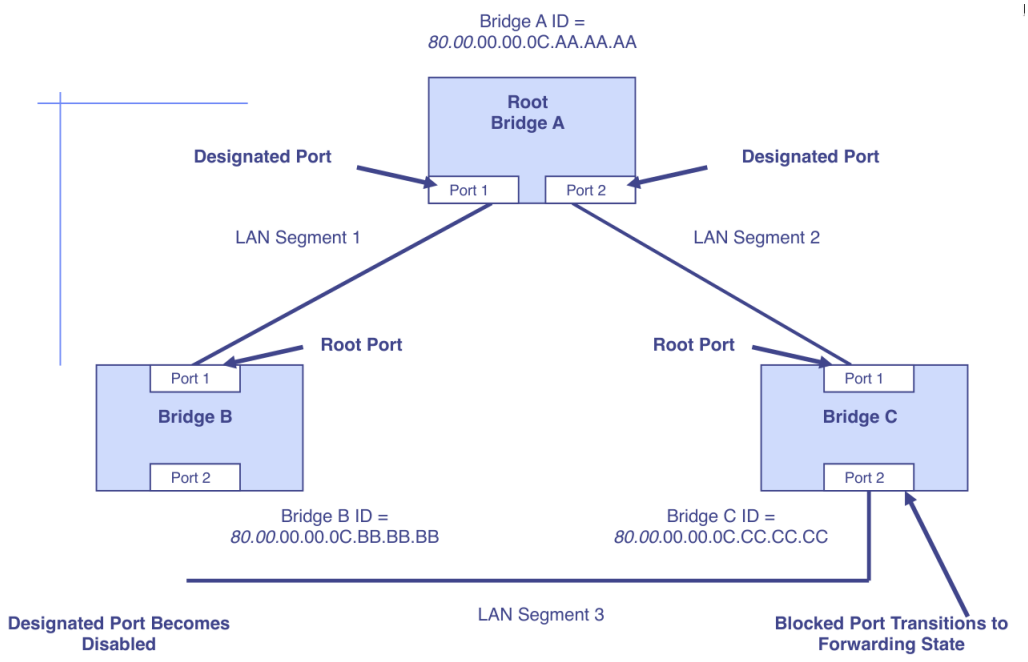


Figure 18: Reacting to Changes

2.7.1 Scaling the STP

- Keep the switched network small; it shouldn't span more than 7 switches.
- Use BDPU skew detection on Cisco switches.
- Use IEEE 802.1w: Provides rapid reconfiguration of the spanning tree. Also known as RSTP.

3 Addressing & Naming

3.1 Guidelines for Addressing & Naming

- Use a structured model for addressing & naming. This makes it easier to:
  - Read network maps.
  - Operate network management software.
  - Recognise devices in protocol analyser traces.
  - Meet goals for usability.
  - Design filters on firewalls & routers.
  - Implement route summarisation.
- Assign addresses & names hierarchically.
- Decide in advance if you will use:
  - Central or distributed authority for addressing & naming.
  - Public or private addressing.
  - Static or dynamic addressing & naming.

3.2 Public & Private IP Addresses

Public IP addresses are managed by the Internet Assigned Number Authority (IANA). Users are assigned IP addresses by Internet Service Providers (ISPs). ISPs obtain allocations of IP addresses from their appropriate Regional Internet Registry (RIR).

- American Registry for Internet Numbers (ARIN) serves North America and parts of the Caribbean.
- RIPE Network Coordination Centre (RIPE NCC) serves Europe, the Middle East, and Central Asia.
- Asia-Pacific Network Information Centre (APNIC) serves Asia and the Pacific region.
- Latin American and Caribbean Internet Addresses Registry (LACNIC) serves Latin America and parts of the Caribbean.
- African Network Information Centre (AfriNIC) serves Africa.

The private IP address ranges are:

- `10.0.0.0` – `10.255.255.255`.
- `172.16.0.0` – `172.31.255.255`.
- `192.168.0.0` – `192.168.255.255`.

3.3 Criteria for Using Static vs Dynamic Addressing

- The number of end systems.
- The likelihood of needing to re-number.
- The need for high availability.
- Security requirements.
- The importance of tracking addresses.
- Whether end systems need additional information (DHCP can provide more than just an address).

3.4 The Anatomy of an IP Address

An IP address consists of a **Prefix** & a **Host**.

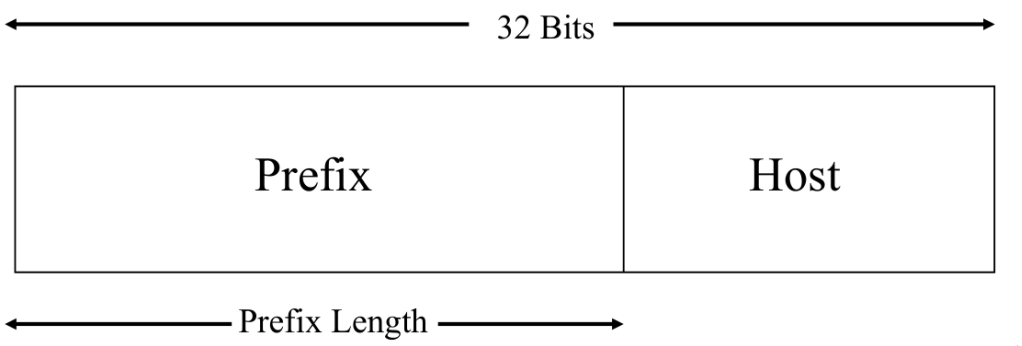


Figure 19: The Two Parts of an IP Address

An IP address is accompanied by an indication of the prefix length, either a **subnet mask** or the `/<length>` notation, e.g. `192.168.10.1 255.255.255.0` or `192.168.10.1/24`. The subnet mask is 32 bits long and specifies which part of an IP address is the network/subnet field and which part is the host field. The network/subnet portion of the mask is all `1`s in binary. The host portion of the mask is all `0`s in binary. The binary expression must be converted back to dotted-decimal notation for entering into configurations. Alternatively, you can use the **slash notation**, e.g. `/24`, which specifies the number of `1`s.

3.5 Designing Networks with Subnets

3.5.1 Addresses to Avoid when Subnetting

- A node address of all `1`s (broadcast).
- A node address of all `0`s (network).
- A subnet address of all `1`s (all subnets).
- A subnet address of all `0`s (confusing). CISCO IOS configuration permits a subnet address of all zeros with the `ip subnet-zero` command.

4 Dynamic Routing Algorithms

4.1 Routing Algorithms

A router can be seen as a device that contains two processes:

- The first process is the **forwarding process** which handles each packet as it arrives, looking up for the outgoing line to use for it.
- The other process is responsible for filling in & updating the routing tables; this is where the **routing algorithm** comes into play.

Certain properties are desirable for a routing algorithm, such as correctness, simplicity, robustness, stability, fairness, & optimality. Fairness & optimality may sound obvious, but they are often contradictory goals: suppose that there is enough traffic between A & A', B & B', C & C' to saturate the horizontal links. To maximise the total flow, the XX' traffic should be shut down completely. Evidently, some sort of compromise between global efficiency & fairness to individual connections is needed.

4.1.1 The Optimality Principle

If a router J is on the optimal path from the router I to the router K, then the optimal path from J to K follows the same route. A direct consequence of the optimality principle is that we can see that all optimal routes from all sources to a given destination form a tree rooted at the destination called a **sink tree**; the tree in the below figure is a sink tree for router B in the subnet, where the metric is the number of hops. The goal of all routing algorithms is to discover & use the sink tree for all routers.

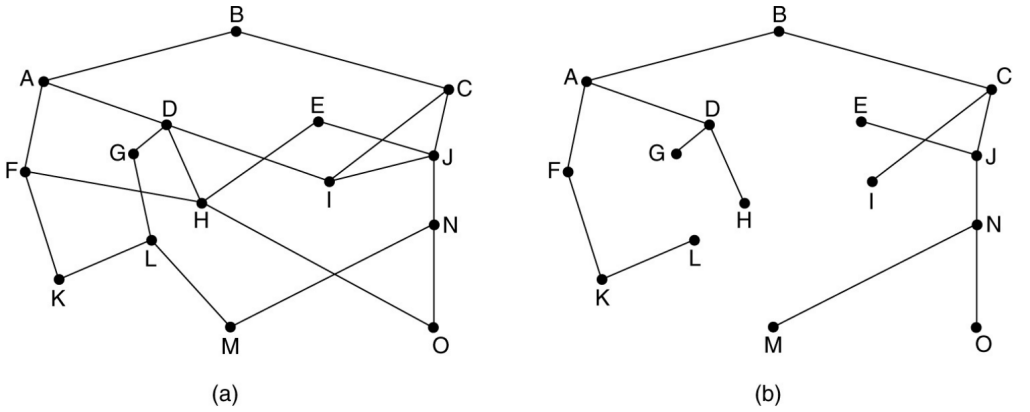


Figure 20: Sink Trees

4.2 Types of Routing Algorithms

Static (non-adaptive) routing algorithms don't base their routing decisions on measurements or estimates of the current traffic and/or topology. The choice of the route to use from I to J is computed in advance (offline) and downloaded into the routers at network boot time.

Dynamic (adaptive) routing algorithms change their routing decisions to reflect changes in topology and usually changes in traffic as well. They differ how they get their information, e.g. locally, from the adjacent router, or from all routers, when they change the routes, or what metrics they use for optimisation, e.g. distance, number of hops, estimated transit time, etc. Computer networks usually use dynamic routing algorithms, since that static ones don't take in calculus the network loads.

4.2.1 Flooding Routing

Flooding is a static algorithm in which every incoming packet is sent out on every outgoing line except the one that it arrived on. It generates a vast number of duplicate packets, and therefore some measures must be taken to dampen the process, such as having a hop counter contained in the header (decremented by each router) with the packet being discarded when the counter reaches 0 or keeping track of which packets have been flooded so that flooding again can be avoided.

In **selective flooding**, the routers don't send every incoming packet out on every line, but only on those lines that are going in the approximate right direction.

Flooding is not practical in most applications, but it does have some use in applications where tremendous robustness is highly desirable (e.g., military applications, where routers are deployed at once, radio networks, etc.). Flooding can also be used as a metric against which other routing algorithms can be compared. Flooding always chooses the shortest path because it chooses every possible path in parallel. Consequently, no other algorithm can produce a shorter delay (if we ignore the overhead...).

4.3 Shortest Path Routing

The idea of **shortest path routing** is to build a graph of the subnet, which each node of the graph representing a router and each arc of the graph representing a communication line (link). To choose a route between a given pair of routers,

the algorithm needs to find the shortest path between them on the graph.

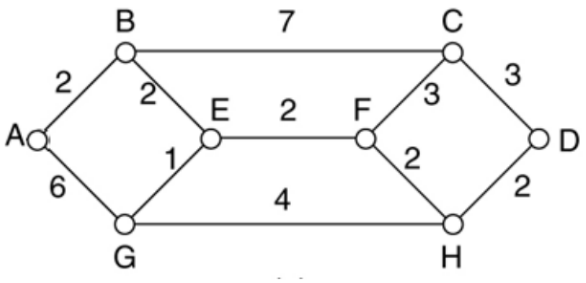


Figure 21: Example Subnet Graph

Metrics for shortest path routing include:

- Number of hops: above, ABC & ABE are equally long.
- Geographic distance: ABC is clearly much longer than ABE, assuming that the above diagram is to scale.
- Other metrics: each arch could be labelled with the mean queuing & transmission delay for some standard test packet as determined by hourly test runs; with this graph labelling, the shortest path is the fastest path rather than the path with the fewest hops or kilometers.

The labels on the arcs could be computed as a function of many factors: distance, average traffic, bandwidth, communication cost, mean queue length, measured delay, & other factors. By changing the criteria, the algorithm will then compute the shortest path according to the measuring criteria or combination of criteria.

4.3.1 Dijkstra's Algorithm for Computing the Shortest Path

Each node is labeled with its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labeled with ∞. As the algorithm proceeds and paths are found, the label may change to reflect better paths. A label may be either tentative or permanent; initially are labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed after that.

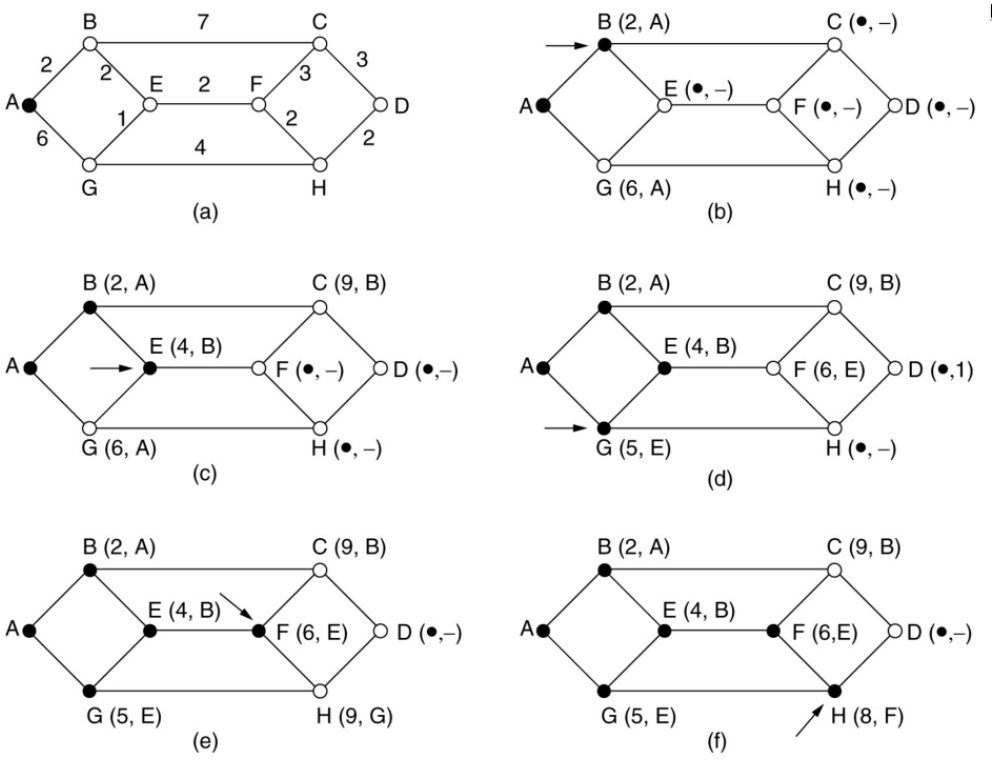


Figure 22: Dijkstra's Algorithm Example

4.4 Distance Vector Routing

Distance Vector Routing was used in ARPANET and is sometimes used in the Internet under the name **RIP**. Each router maintains a table (i.e., a vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbours. The routing table contains an entry for each router in the subnet. This entry contains two parts:

- Preferred outgoing line to use for that destination.
- The estimation of the time or distance to that destination (the used metric can be the number of hops, time delay in milliseconds, the total number of packets queued along that or something similar).

The router is assumed to know the distance to each of its neighbours:

- If the metric is hops, the distance is just hop.
- If the metric is time, the router can measure it directly with special echo packets.
- If it is queue length, the router examines each of its queues.

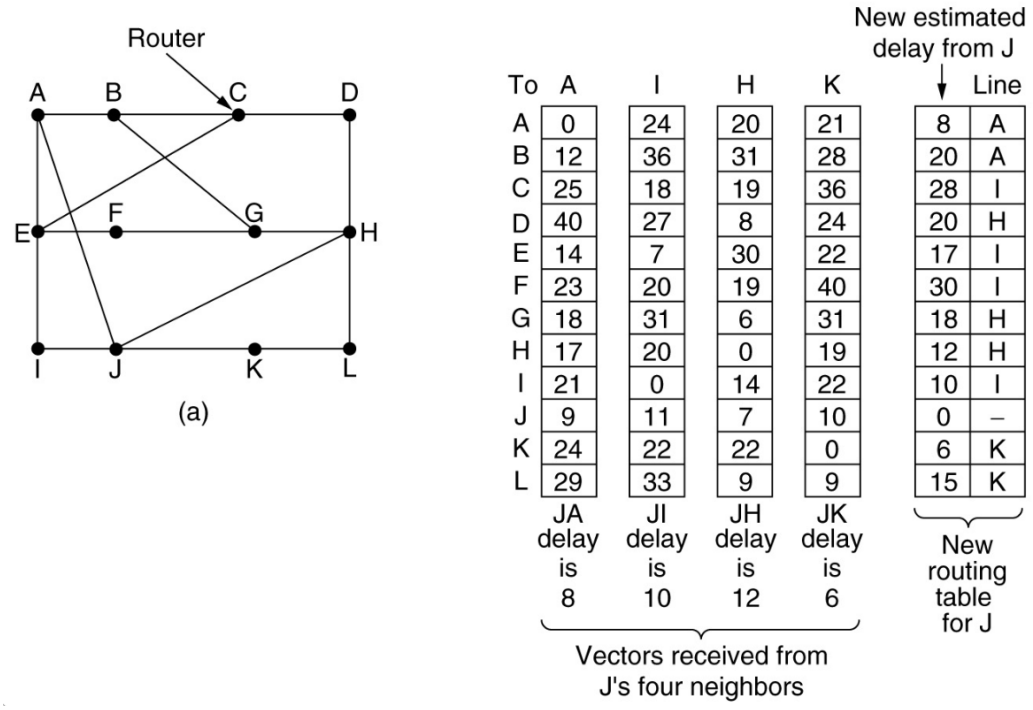


Figure 23: Distance Vector Routing

1. J measures the delays to its neighbours.
2. J computes its new routes to router G:
   - It knows that it can get to A in 8ms and A claims to get to G in 18ms, so J knows that it can count on packets with 26ms to G if it routes the packets through A.
   - Similarly, it computes delays to G via I $(10 + 31)$, via H $(12 + 6)$, & via K $(6 + 31)$. The best of these values is 18 so it makes an entry in its routing table that the delay to G is 18ms and the route to use is via H.
3. The same calculation is performed to all other destinations.

4.5 Link State Routing

Distance vector routing was replaced by link state routing for two reasons:

- Distance vector routing's delay metric is **queue length**; it didn't take in calculus the line bandwidth.
- The distance vector routing algorithm takes too long to converge.

Each router must do the following:

- Discover their neighbours and learn their net addresses.
- Measure the delay or cost to each of their neighbours.
- Construct a packet telling all it has just learned.
- Send this packet to all the other routers.
- Compute the shortest path to every other router.

4.5.1 Distance Vector vs Link State Routing

- With distance vector routing, each node has information only about the next hop.
- Distance vector routing makes poor routing decisions if directions are not completely correct (e.g., because a node is down). If parts of the directions are incorrect, the routing may be incorrect until the routing algorithms have re-converged.
- In link state routing, each node has a complete map of the topology.
- If a node fails in link state routing, each node can calculate a new route.
- The main difficulty with link state routing is that all nodes need to have a consistent view of the network.

4.5.2 Basic Principles of Link State Routing

1. Each router establishes a relationship called an **adjacency** with its neighbours.
2. Each router generates **Link State Advertisements (LSAs)** which are distributed to all routers. An LSA consists of a link ID, the state of the link, the cost, & the neighbours of the link.
3. Each router maintains a database of all received LSAs called a **topological database** or **link state database**, which describes the network as a graph with weighted edges.
4. Each router uses its link state database to run a shortest path algorithm (Dijkstra's algorithm) to produce the shortest path to each network.

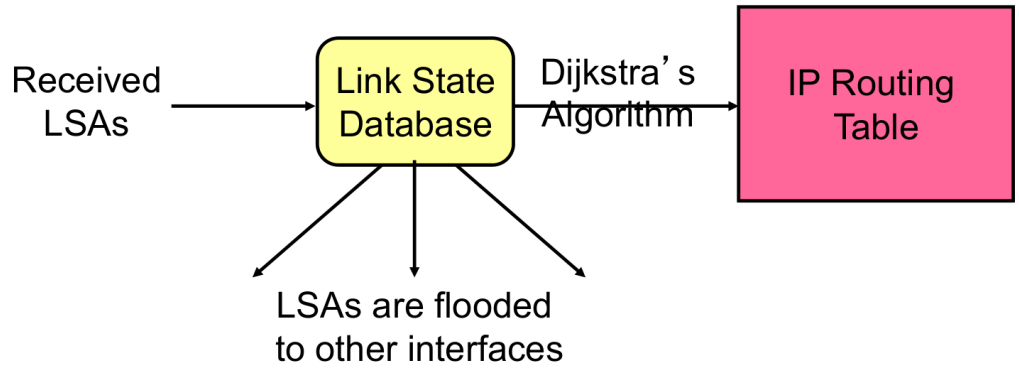


Figure 24: Operation of a Link State Routing Protocol

4.6 OSPF

The **OSPF (Open Shortest Path First)** routing protocol is the most important link state routing protocol on the Internet. The complexity of OSPF is significant. Features of OSPF include:

- Provides authentication of routing messages.
- Enables load balancing by allowing traffic to be split evenly across routes with equal cost.
- Type-of-Service routing allows the setup of different routes dependent on the TOS field.
- Supports subnetting.
- Supports multicasting.
- Allows hierarchical routing.

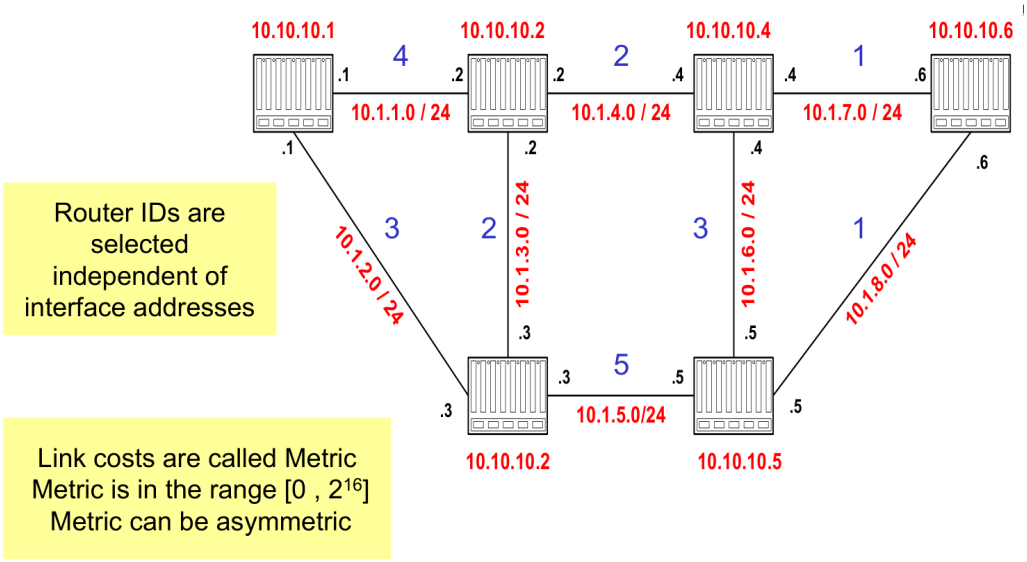


Figure 25: OSPF in an Example Network

4.6.1 Link State Advertisement (LSA)

Each router sends its LSA to all routers in the network using a method called **reliable flooding**. The LSA of router `10.10.10.1` from the previous example is as follows:

- Link State ID: `10.10.10.1` = Router ID.
- Advertising Router: `10.10.10.1` = Router ID.
- Number of links: 3 = 2 links plus the router itself.
- Description of Link 1: Link ID = `10.1.1.1`, Metric = 4.
- Description of Link 2: Link ID = `10.1.2.1`, Metric = 3.
- Description of Link 3: Link ID = `10.10.10.1`, Metric = 0.

Each router has a database which contains the LSAs from all the other routers. The collection of all LSAs is called the **link-state database**. Each router has an identical link-state database; this is very useful for debugging, as every router has a complete description of the network. If neighbouring routers discover each other for the first time, they will exchange their link-state databases. The link-state databases are synchronised using **reliable flooding**.

4.7 OSPF Packet Format

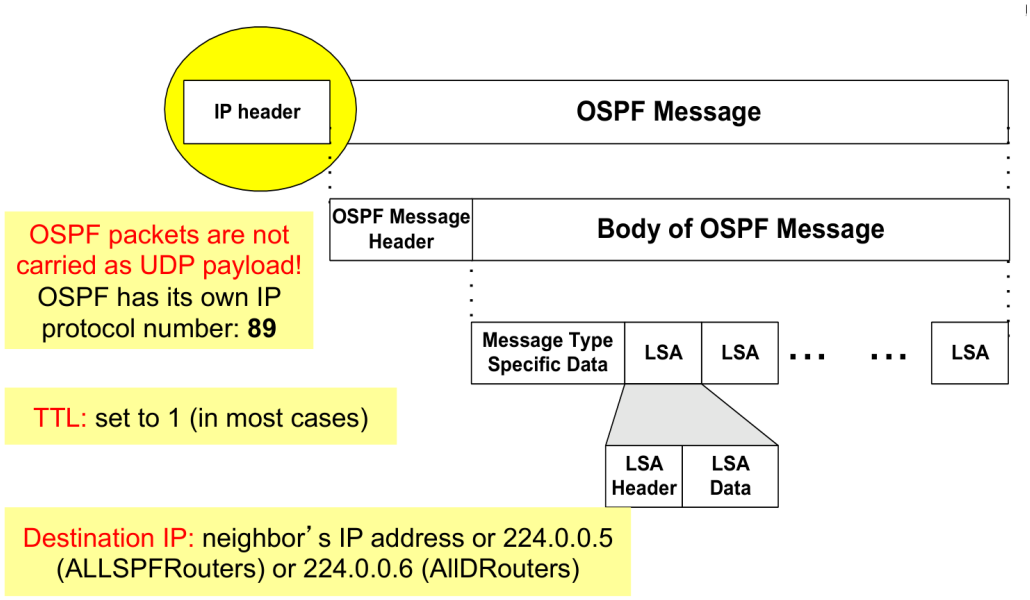


Figure 26: OSPF Packet Format 1

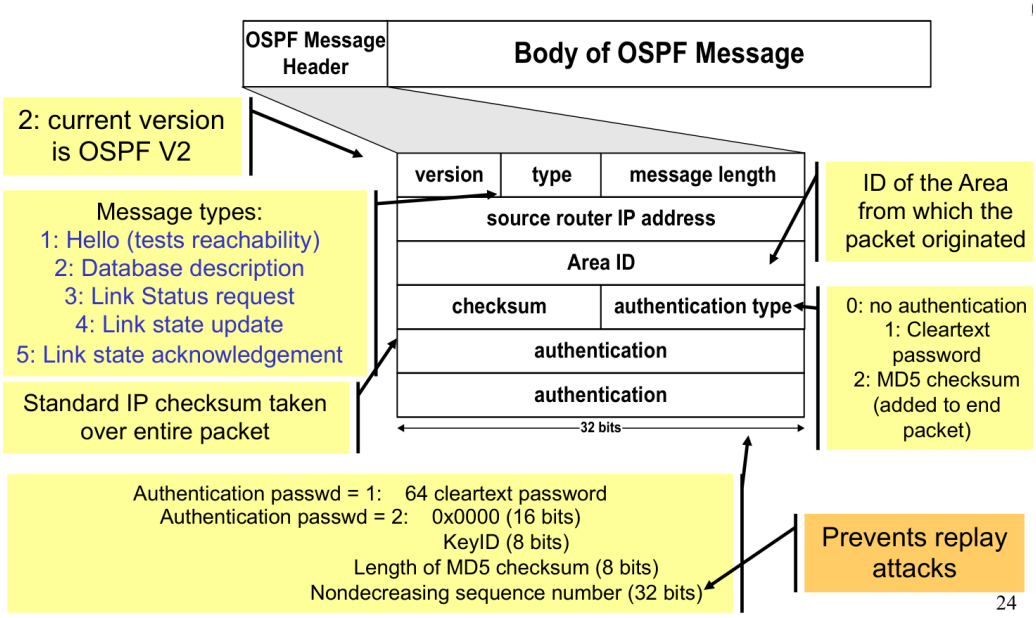


Figure 27: OSPF Packet Format 2

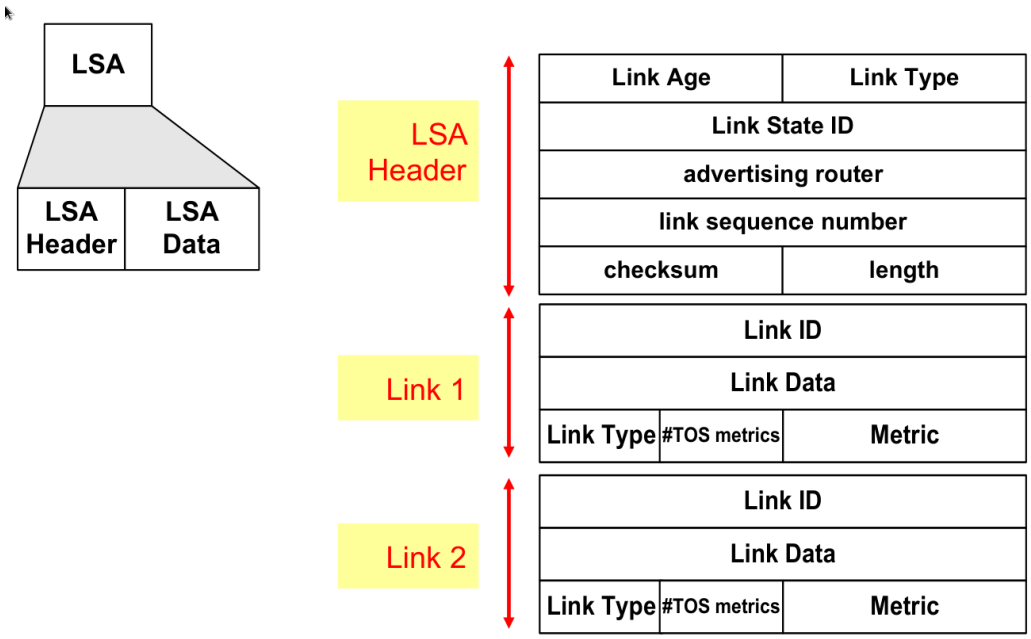


Figure 28: OSPF LSA Format

4.8 Discovery of Neighbours

The routers multicast **OSPF Hello packets** on all OSPF-enable interfaces. If two routers share a link, they can become neighbours and establish adjacency.

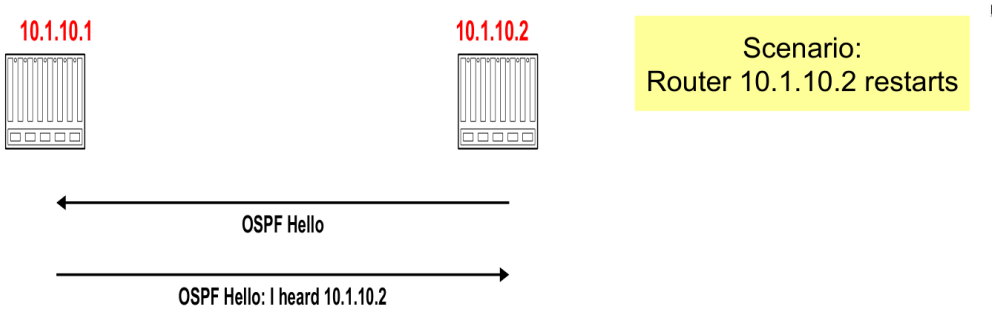


Figure 29: Discovery of Neighbours

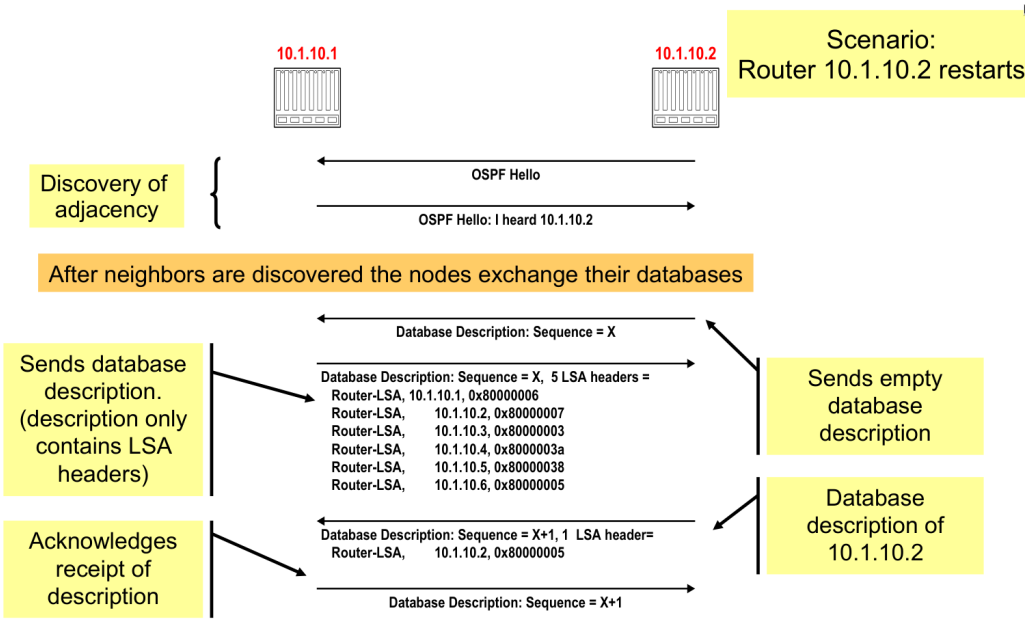


Figure 30: Neighbours Discovery & Database Synchronisation

4.9 Regular LSA Exchanges

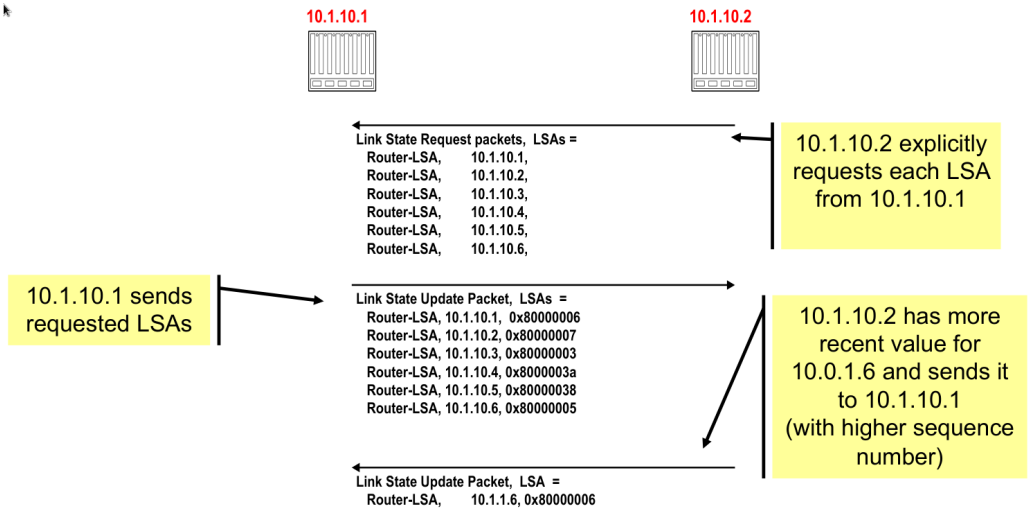


Figure 31: Regular LSA Exchanges

4.10 Routing Data Distribution

LSA-Updates are distributed to all other routers via **reliable flooding**.

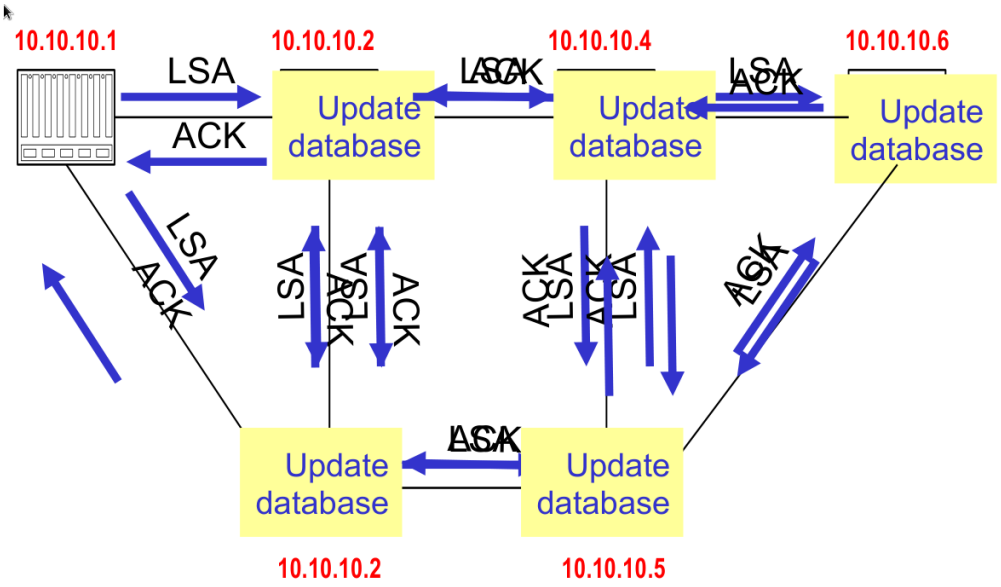


Figure 32: Example: flooding of LSA from `10.10.10.1`

4.11 Dissemination of LSA-Update

A router sends and refloods LSA-Updates whenever the topology changes or link costs change. If a received LSA does not contain new information, the router will not flood the packet. There is one exception: infrequently (every 30 minutes), a router will flood LSAs even if there are not new changes. Acknowledgements of LSA-Updates can be explicit with an `ACK` or implicit via reception of an LSA-Update.

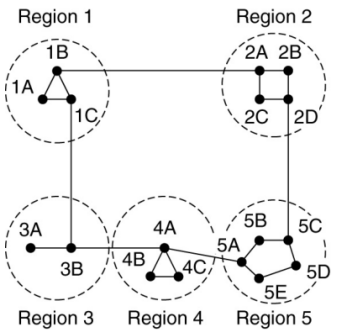
Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented. Dijkstra's algorithm will be installed in the routing tables and the normal operation resumed.

Practical consideration: for a subnet having n routers, with k neighbours, the memory required to store the input data is proportional to kn, which may be a problem for large subnets. Computation time can also be a problem. In many practical situations, the link state algorithm works well.

4.12 Hierarchical Routing

As networks grow in size, the routers' routing tables grow proportionally. More CPU time is also required to scan the routing tables, and more bandwidth is required to send new status reports. At a point, the network may grow so large that it is no longer feasible for every router to have an entry for every other router, so the routing has to be done hierarchically.

The routers are divided into "**regions**", with each router knowing details only about routers in its own region, and knowing no details about the internal structures of other regions. For huge networks, it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, an so on, until we run out of names for aggregations.



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Figure 33: Hierarchical Routing Example

The above diagram shows routing in a two-level hierarchy with five regions. The full routing table for 1A has 17 entries. For hierarchical routing, the routing table has 7 entries. There is a penalty to be paid in the form of increased path length: the best route from 1A to 5c is through region 2. With hierarchical routing, the traffic for region 5 goes through region 3, because that is for most destinations in region 5.

4.12.1 Autonomous Systems

An **anonymous system** is a region of the Internet tat is administered by a single entity. Examples of autonomous regions include Heanet's national network, Eircom's backbone network, & Region Internet Service Provider. Routing is done differently within autonomous systems (**intradomain routing**) and between autonomous systems (**interdomain routing**).

4.12.2 Border Gateway Protocol (BGP)

The **Border Gateway Protocol (BGP)** is an interdomain routing protocol for routing between autonomous systems. BGP is currently in version 4. It uses TCP to send routing messages. BGP is neither a link state nor a distance vector protocol. Routing messages in BGP contain complete routes. Network administrators can specify routing policies. Note that in the context of BGP, a gateway is nothing other than an IP router that connects autonomous systems.

BGP's goal is to find *any* path, not necessarily an optimal one. Since the internals of the autonomous system are never revealed, finding an optimal path is not feasible. For each autonomous system (AS), BGP distinguishes:

- **Local traffic:** traffic with a source or destination in the AS.
- **Transit traffic:** traffic that passes through the AS.
- **Stub AS:** has connection to only one AS, carries local traffic.
- **Multihomed AS:** has connections to more than one AS, but does not carry transit traffic.
- **Transit AS:** has connections to more than one AS, and does carry transit traffic.