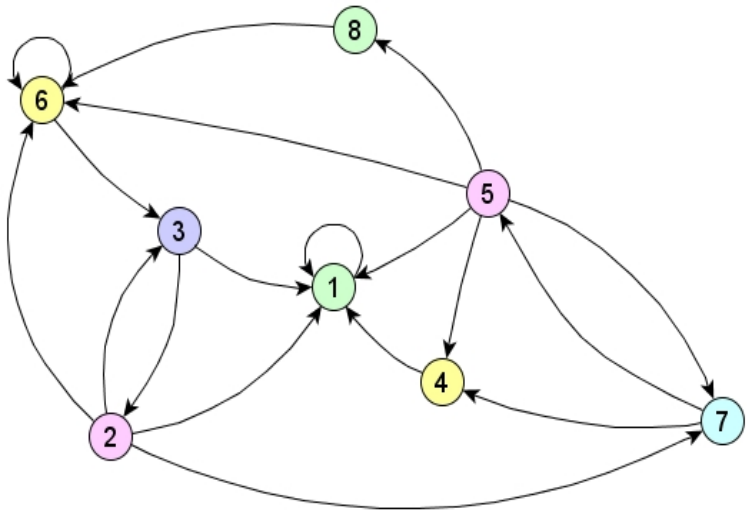
chapter 1 Network graphs

1.1 multigraphs

A **directed** (**oriented**) **graph** G is a pair (V;A) of sets, with A subset of VxV.The set V = V (G) is called the **vertex** (**node**) **set** of the graph G and its elements are called the **vertices** (**nodes**) of the graph. The set A = A(G) is called the **arc set** of the graph G and its elements are called the **arcs** of the graph.



We make the remark that the above definition does not allow multiple arcs between a pair of vertices. When this is allowed, the corresponding concept is called multigraph, concept which will be covered in a subsequent section.

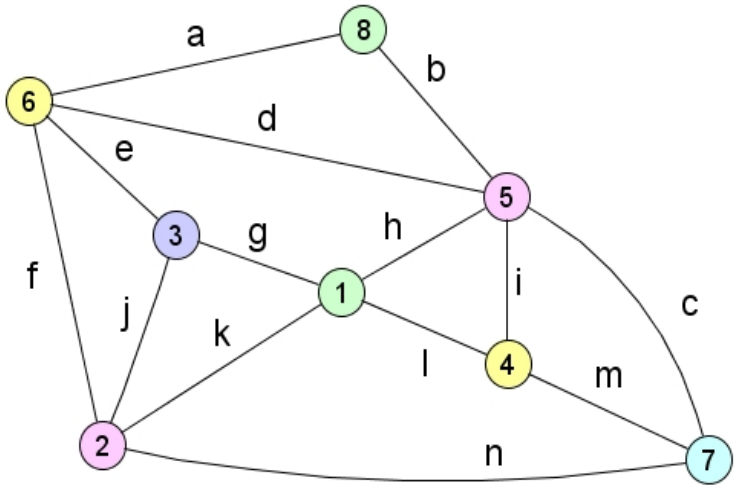
We continue our sequence of definitions and notations without putting them in a formal definition setting.

Given a directed graph G = (V;A) and an arc (u; v) in A, the vertex u is called the **source** (initial vertex) of the arc and v is called the **target** (final vertex) of the arc. Both u and v bear the generic name of **end-vertices** (**end-nodes**) of the arc, or just the vertices of the arc.

Two arcs are called adjacent if they have an end-vertex in common. Two vertices u and v are called **adjacent** if they are the end-vertices of the same arc, i.e., if (u; v) or (v; u) in A. A vertex and an arc are called **incident** if the vertex is one of the end-vertices of the arc.

An **undirected graph** G is a pair (V;E) where V is the vertex set and E is a family of subsets with 2 elements of V . The elements of E are unordered pairs of distinct elements of V and are called **edges**.

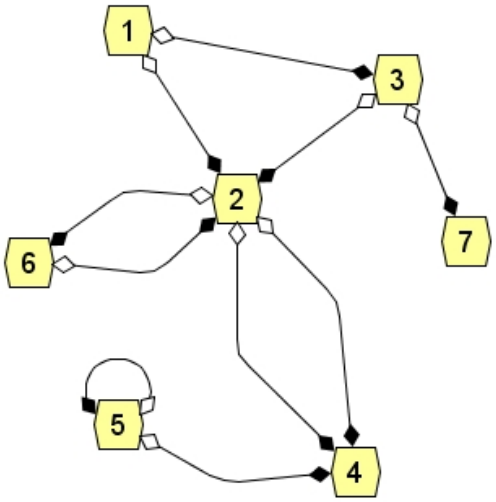
We make the remark that the above definition does not allow multiple edges between 2 given vertices and does not allow loops (edges from one vertex to itself). When this is allowed, we talk about multigraphs, concept which will be covered in a subsequent section.



Given a vertex subset U of V , we denote by E_U the set edges which have one of its end-vertices in U, and the other one, not in U. Also, denote by E(v) the vertex set {u in V : {v; u} in E}. The vertices in E(v) are called the neighbors of the vertex v. The set E(v) is often denoted by N(v).

We introduce now the concept of **ports** associated to a graph vertex (node). In a computer network, the physical end points of a communication link correspond to a port, which is identied by a port number. The entrance point of a road into an ancient city was usually made through a gate or a (in latin) port. Keeping in mind these designations (as end point of an arc or edge, in general), here is a formal definition of a multigraph.

A **directed multigraph** is a structure M = (V; A;$\mathcal{P}_V$) where V is the vertex set, $\mathcal{P}_V$ = {P_v : v in V} is a family of disjoint sets indexed by the elements of V while A, the **arc set** of M, is a subset of PxP, where P, the **global port set** is the union of all vertex ports.



If a = (p; q) in A is an arc of M = (V; A;$\mathcal{P}_V$), then p in P is called the **source port** of a and q in P is called the **target port** of a. The vertex function v_M : P -> V associated to the multigraph M is a function which associates to each port the unique vertex it belongs to. The function v_M is well defined because the sets in the family $\mathcal{P}_V$ are mutually disjoint. Given an arc a in A, we define the source function s : A -> V by s((p; q)) = v_M (p) and the target function t : A -> V by t((p; q)) = v_M(q).

The concept of undirected multigraph is similar, as stated in the next definition.

An **undirected multigraph** is a structure M = (V; E;$\mathcal{P}_V$) where V Is the vertex set, $\mathcal{P}_V$ = {P_v : v in V} is a family of disjoint sets indexed by the elements of V while E, the **edge set** of M, is a subset of PxP, where P, the global port set is the union of all vertex ports.

1.2 the graph of a network

A **network element** is a physical or logical device that performs a specific network related activity. A list of network elements includes but is not limited to, devices like workstations, hubs, repeaters, bridges, switches, routers, modems. Network elements are expected to perform some sort o processing of the data that originates, transits or end at the device. The communication links are used to connect network elements and have the primary role of transmitting data. Communication links are not supposed, in general, to process the data that transits them. In the case such a processing occurs, it is destined to enhance data characteristics, but not to alter the cognitive content of that data.

A **network** consist of network elements and the communication links between the network elements. The **graph of a network** is a directed multigraph graph G = (V; A; $\mathcal{P}_V$) whose node (vertex) set V consists of the network elements and the arc set consists of the links between the network elements. The port set P_v at a particular vertex (node) v of the graph corresponds to the physical or logical ports of the network element, each port representing one end-point of a communication link. The arc set A corresponds to physical communication links like copper wires, optical fiber pipes or frequency channels.

1.3 collision domains

In a half duplex Ethernet network, a **collision** is the result of two devices on the same Ethernet network attempting to transmit data at exactly the same time. The network detects the "collision" of the two transmitted packets and discards them both. Collisions are a natural occurrence on Ethernets. Ethernet uses Carrier Sense Multiple Access / Collision Detect (CSMA/CD) as its method of allowing devices to "take turns" using the signal carrier line. When a device wants to transmit, it checks the signal level of the line to determine whether someone else is already using it. If it is already in use, the device waits and retries, perhaps in a few seconds. If it isn't in use, the device transmits. However, two devices can transmit at the same time in which case a collision occurs and both devices detect it. Each device then waits a random amount of time and retries until successful in getting the transmission sent.

A **collision domain** is a section of a network where data packets can collide with one another when being sent on a shared medium or through repeaters, in particular, when using early versions of Ethernet. A network collision occurs when more than one device attempts to send a packet on a network segment at the same time. Collisions are resolved using CSMA/CD in which

the competing packets are discarded and re-sent one at a time. This becomes a source of inefficiency in the network.

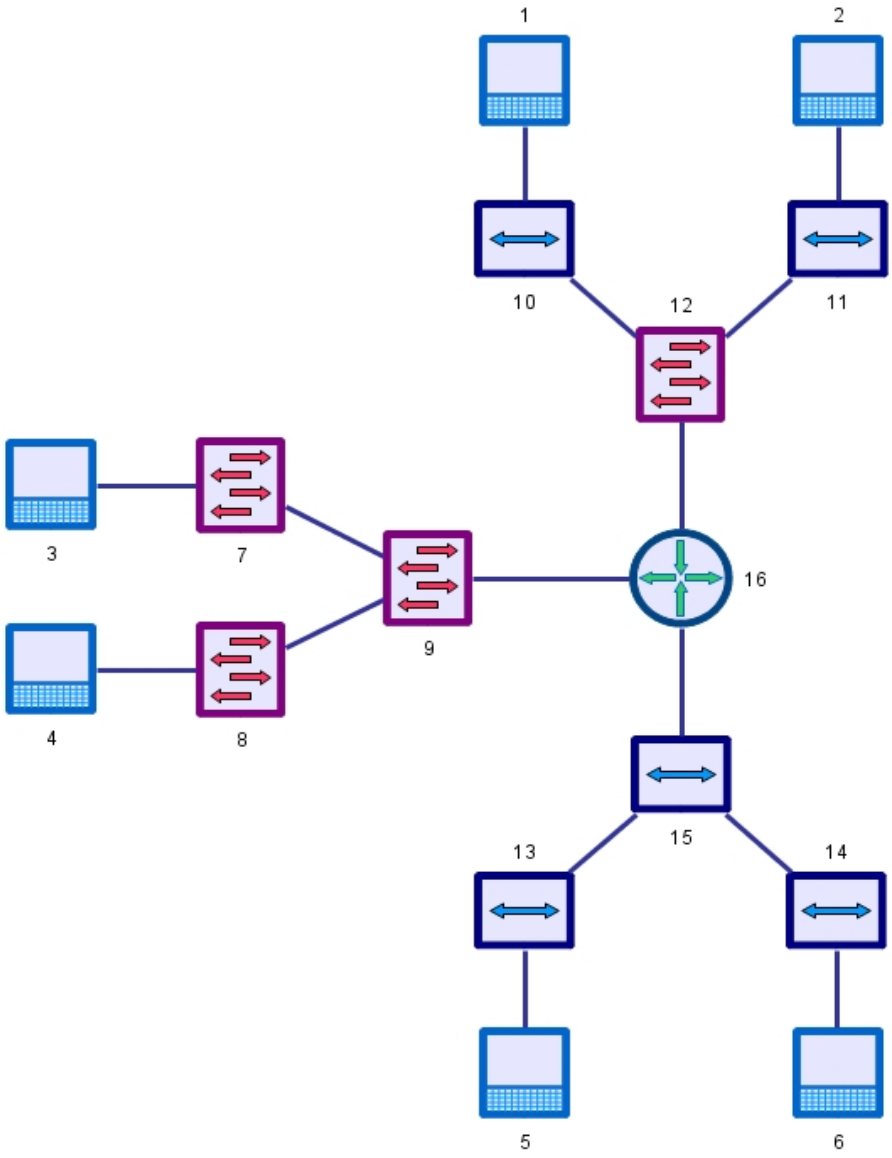


Figure 4 – A small local area network.

Only one device in the collision domain may transmit at any one time, and the other devices in the domain listen to the network in order to avoid data collisions. Because only one device may be transmitting at any one time, total network bandwidth is shared among all devices. Collisions also decrease network efficiency on a collision domain; if two devices transmit simultaneously, a collision occurs, and both devices must retransmit at a later time. Collision domains are found in a hub environment where each host segment connects to a hub that represents only one collision domain and only one broadcast domain. Collision domains are also found in wireless networks such as Wi-Fi.

Modern wired networks use a network switch to eliminate collisions. By connecting each device directly to a port on the switch, either each port on a switch becomes its own collision domain (in the case of half duplex links) or the possibility of collisions is eliminated entirely in the

case of full duplex links.

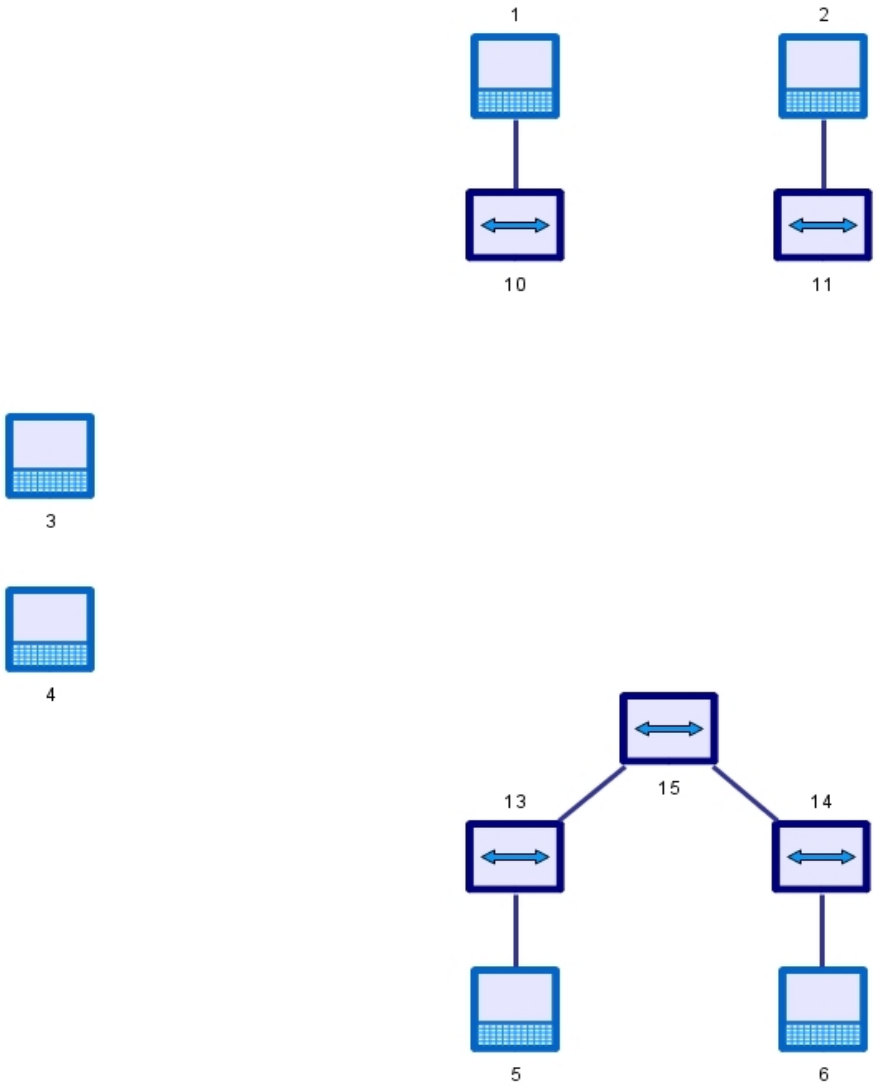


Figure 5 – Collision domains

How do we identify collision domains using a network's graph? If we eliminate the nodes which split the collision domains (bridges, switches and routers), a collision domain will correspond to a connected component of the newly formed network graph.

In the case pictured above, we ens up with 5 connected components, i.e., 5 collision domains.

1.4 broadcast domains

A **broadcast domain** is a logical division of a computer network, in which all nodes can reach each other by broadcast at the data link layer. A broadcast domain can be within the same LAN segment or it can be bridged to other LAN segments. In terms of current popular technologies:

Any computer connected to the same Ethernet repeater or switch is a member of the same broadcast domain. Further, any computer connected to the same set of inter-connected

switches/repeaters is a member of the same broadcast domain.

Routers and other higher-layer devices form boundaries between broadcast domains. This is as compared to a collision domain, which would be all nodes on the same set of inter-connected repeaters, divided by switches and learning bridges. Collision domains are generally smaller than, and contained within, broadcast domains. While some layer two network devices are able to divide the collision domains, broadcast domains are only divided by layer 3 network devices such as routers or layer 3 switches.

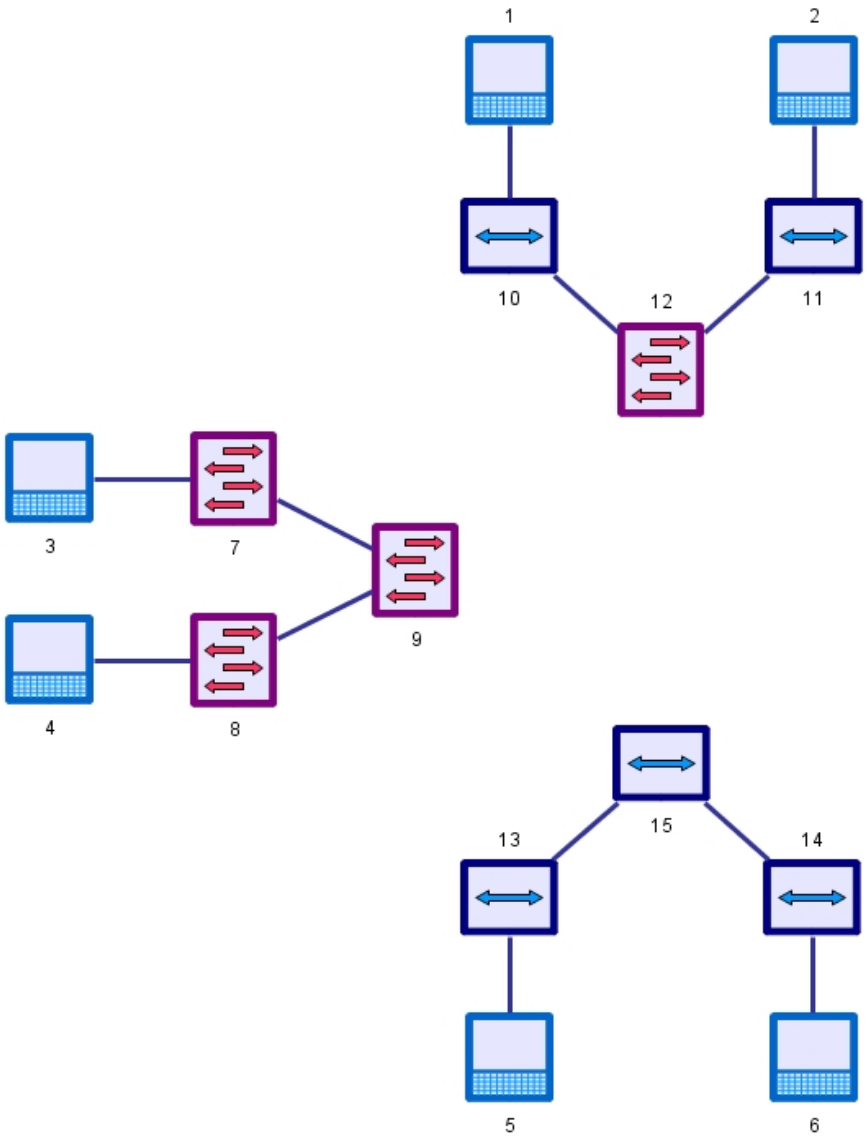
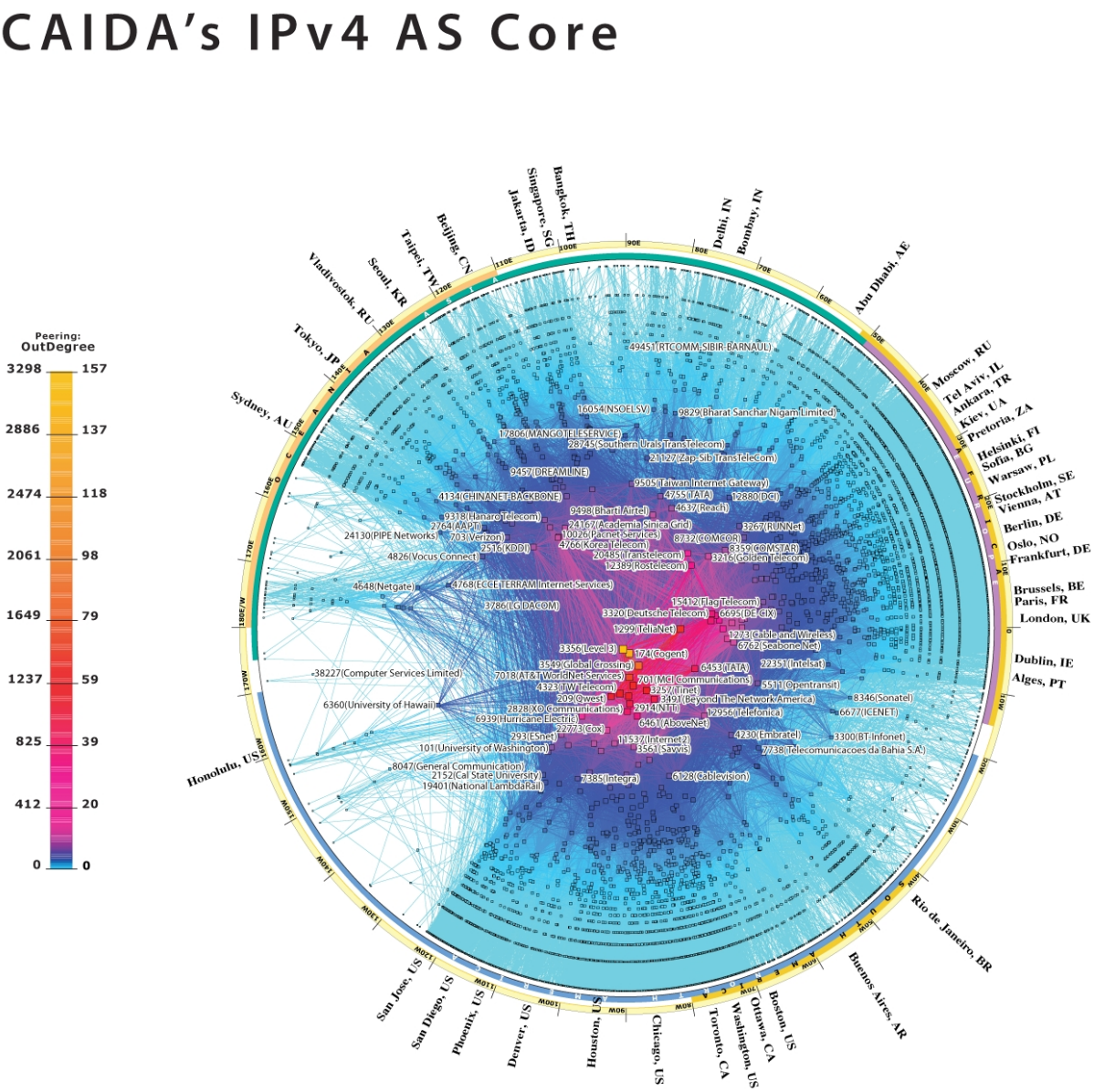


Figure 6 – Broadcast domains

How do we identify broadcast domains using a network's graph? If we eliminate the nodes which split the broadcast domains (routers or layer 3 switches), a broadcast domain will correspond to a connected component of the newly formed network graph. If we eliminate the router (the only broadcast domain splitting device) in the network graph pictured in Figure 4, we end up with 3 connected components, i.e., 3 broadcast domains.

1.5 networks representations

While a node level representation of the global network is not available, Autonomous System (see next chapter) representations are available. CAIDA – the Cooperative Association for Internet Data Analysis (http://www.caida.org/home/) provides periodically snapshots of IPv4 and IPv6 Internet topology at AS level. The latest data and pictures date back to August 2010.



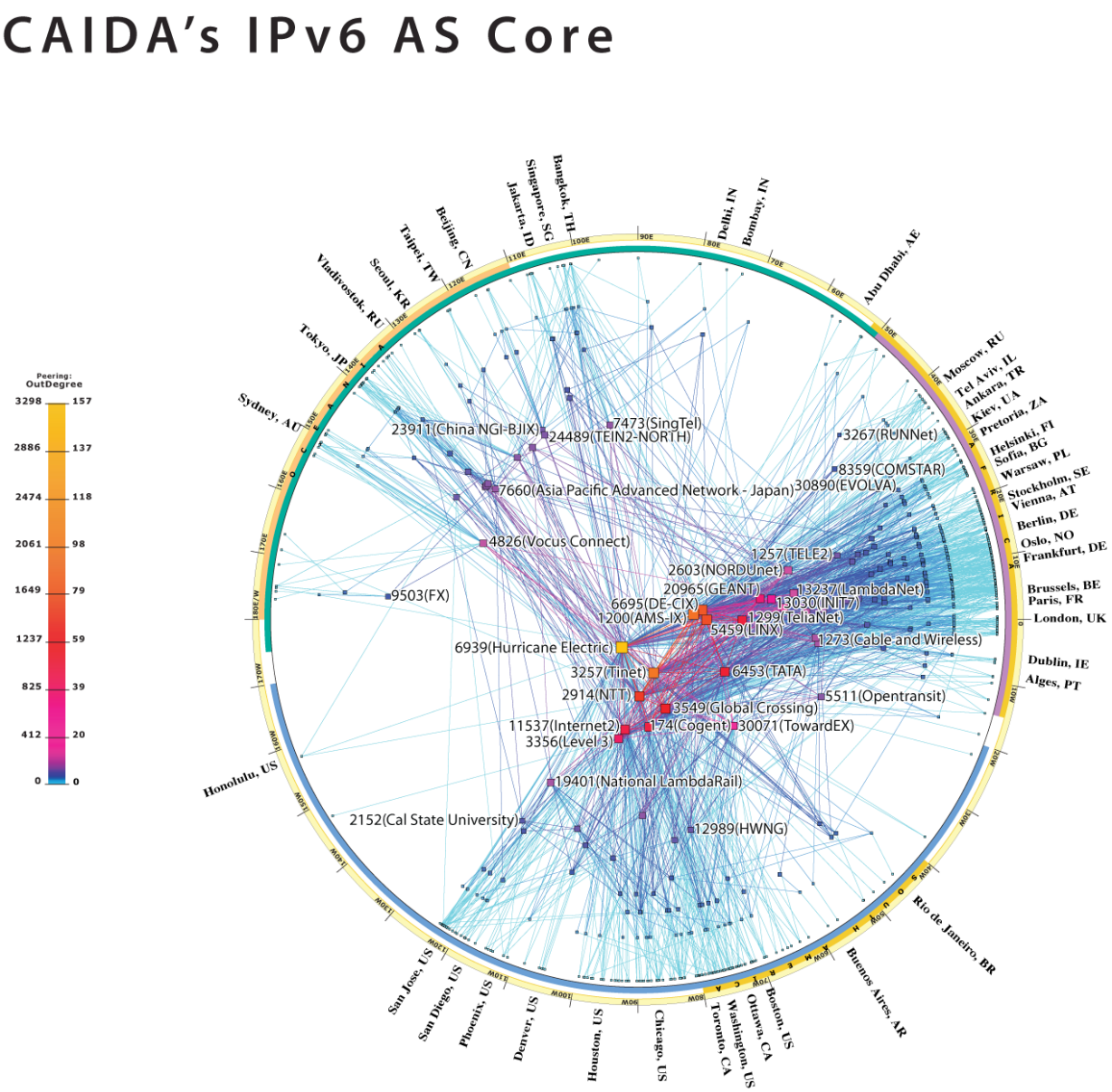
A pretty accurate picture of the Ipv4 map of the internet in Aug. 2010 (26,702 ASs - 96% of the Autonomous Systems), presenting 16,802,061 IP addresses and 18,796,744 IP links.

This image does not include quite a few private or corporate network, but are still very relevant to the expansion level of the global communication network.

We conclude this internet gallery with the IPv6 internet topology map, also as of Aug. 2010, picture provided by CAIDA, as well.

Surprisingly or not, countries where the internet is to a higher degree government regulated, are more likely to have a sophisticated IPv6 presence.

In absolute values, the Aug. 2010 figures for IPv6 autonomous systems, as provided by CAIDA, are as follows - 715 ASs, with 99.6% of globally routeable network prefixes , presenting 8,551 IPv6 addresses and 21,852 IPv6 links.



1.6 slots

The network graph defined in 1.2 models the physical network. Quite often, this is not enough. In the pretty basic case of an IP communication link, the physical medium may be shared by several sockets. A socket is identified by the IP address of the interface and by a port number. While the data packets that originate or end at diverse socket ports do not mix, they still share the same physical communication link. To model this fact, we introduce the concept of **slot** associated to a network graph arc (link). Formally, this is how it is defined.

Let G = (V; A; $\mathcal{P}_V$) be a network graph and let C_A = {C_a : a in A} be a family of disjoint sets indexed by the arc (link) set. The union S of all sets of the family S_A is called the **global sloth set**, while for a in A, S_a is the **slot set** of the arc (link) a.

An **extended network graph** is a structure X = (V; A; $\mathcal{P}_v$; S_A), where G = (V; A; $\mathcal{P}_V$) is a network graph and S_A is a family of slot sets for the network graph G.

Slots can be viewed in general, as the result of a multiplexing scheme, which allows concurrent transmission over the same shared media.

Such an example can be provided by TDMA - Time Division Multiple Access - a channel access method for shared medium networks. It allows several users to share the same frequency channel by dividing the signal into different time slots. The users transmit in rapid succession, one after the other, each using its own time slot. This allows multiple stations to share the same transmission medium (e.g. radio frequency channel) while using only a part of its channel capacity. TDMA is used in the digital 2G cellular systems such as Global System for Mobile Communications (GSM) [TDMA].

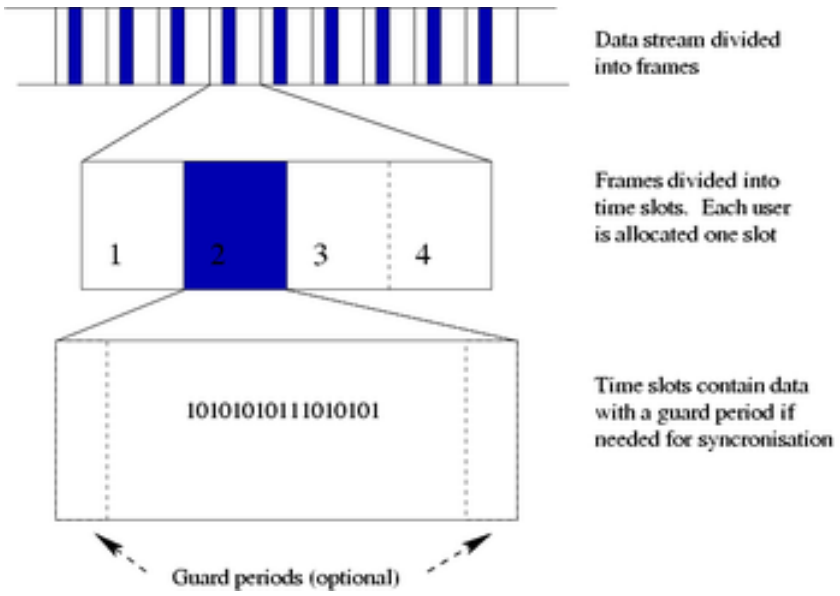


Figure 9 - Time slot allocation on a shared frequency channel

1.7 graphML and yEd

GraphML is an XML-based file format for graphs. The GraphML file format results from the joint effort of the graph drawing community to define a common format for exchanging graph structure data. The main tags defined by GraphML are:

- graph
- node
- edge
- port

The GraphML specification, besides the predefined tag attributes, allows the creation of other tag specific attributes (properties).

On the other side, an extension of GraphML with a **slot tag** is highly desirable.

A very simple example of a graph description using the graphml format:

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
      http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
      <graph id="G" edgedefault="undirected">
            <node id="n0"/>
            <node id="n1"/>
            <node id="n2"/>
            <edge source="n0" target="n2"/>
            <edge source="n1" target="n2"/>
      </graph>
</graphml>
```

For other details about the GraphML format, read the GraphML primer [GMLP] or the actual GraphML specification [GMLS].

The GraphML format is supported by several graph drawing/visualization tools, including yEd, Gephi, Visone and Network Workbench.

chapter 2 Routing principles

When a message is sent across the internet from a source to its destination it is typically split in a number of pieces (called, at the level of interest for us – packets) which make their way independently across a maze of switches, routers, firewalls, bridges. At most of these network nodes, a decision has to be made, namely, on which port (communication link) should the incoming packet be sent? This decision process is called **routing**.

Why routing? Networks are complex, dynamic and heterogeneous. They include :

- the packet switched network - the internet
- the circuit switched networks – telephony
- wireless networks
- etc.

The routing process usually directs forwarding on the basis of routing tables which maintain a record of the routes to various network destinations. Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing. Most routing algorithms use only one network path at a time, but multipath routing techniques enable the use of multiple alternative paths.

The way the routing decision is made is largely dependent on the nature of next destination. A destination which may be inside the same routing unit (autonomous system - AS) or across two different routing units (ASs).

2.1 autonomous systems

An Autonomous System (AS) is a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators that presents a common, clearly defined routing policy to the Internet.

Examples of ASs are:

- An ISP (Internet Server Provider)
- A campus network
- A business division

A unique Autonomous System Number (ASN) is allocated to each AS. AS numbers are important because the ASN uniquely identifies each network on the Internet. Until 2007, AS numbers were defined as 16-bit integers, which allowed for a maximum of 65536 assignments. The Internet Assigned Numbers Authority (IANA) has designated AS numbers 64512 through 65534 to be used for private purposes. The ASNs 0, 59392-64511 and 65535 are reserved by the IANA and should not be used in any routing environment. ASN 0 may be used to label non-routed networks. All other ASNs (1-54271) are subject to assignment by IANA, and, as of September 9, 2008, only 49152-54271 remained unassigned. RFC 4893 [RFC4893] introduced 32-bit AS numbers, which IANA has begun to allocate.

These numbers are written either as simple integers, or in the form x.y, where x and y are 16-

bit numbers. Numbers of the form 0.y are exactly the old 16-bit AS numbers, 1.y numbers and 65535.65535 are reserved, and the remainder of the space is available for allocation.

The accepted textual representation of Autonomous System Numbers is defined in RFC 5396. [RFC5396]. The number of unique autonomous networks in the routing system of the Internet exceeded 5000 in 1999, 30000 in late 2008, and 35000 in mid 2010.

Routing inside ASs is done using Interior Gateway Protocols (IGP), like:

- OSPF
- IS-IS
- RIP

Routing between different ASs is done using Exterior Gateway Protocols (EGP), like:

- BGP - Border Gateway Protocol
- Inter Domain routing

2.2 routing tables

An example of a routing table for a workstation with a Unix/Linux OS can be obtained by running the command:

 # route -n

```
Kernel IP routing table

Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
172.16.55.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
172.16.50.0     172.16.55.36    255.255.255.0   UG    0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         172.16.55.1     0.0.0.0         UG    0      0        0 eth0
```

Here is the explanation for this routing table headers:

- **Destination** : The destination network or destination host.
- **Gateway** : The gateway address or '*' if none set.
- **Genmask** : The netmask for the destination net; 255.255.255.255 for a host destination and 0.0.0.0 for the default route.
- **Flags** : Possible flags include
  - U (route is up)
  - H (target is a host)
  - G (use gateway)
  - R (reinstate route for dynamic routing)
  - D (dynamically installed by daemon or redirect)
  - M (modified from routing daemon or redirect)
  - A (installed by addrconf)

    - C (cache entry)
    - ! (reject route)
- **Metric** : The distance to the target (usually counted in hops). It is not used by recent kernels, but may be needed by routing daemons.
- **Ref** : Number of references to this route. (Not used in the Linux kernel.)
- **Use** : Count of lookups for the route. Depending on the use of -F and -C this will be either route cache misses (-F) or hits (-C).
- **Iface** : Interface to which packets for this route will be sent.
- **MSS** : Default maximum segment size for TCP connections over this route.
- **Window** : Default window size for TCP connections over this route.
- **irtt** : Initial RTT (Round Trip Time). The kernel uses this to guess about the best TCP protocol parameters without waiting on (possibly slow) answers.
- **HH** (cached only) : The number of ARP entries and cached routes that refer to the hardware header cache for the cached route. This will be -1 if a hardware address is not needed for the interface of the cached route (e.g. lo).
- **Arp** (cached only) : Whether or not the hardware address for the cached route is up to date.

A simplified version of a routing table may have this structure:

Destination Network	Next Router	Port	Route Cost
123.0.0.0	127.5.21.2	2	11
144.12.0.0	127.3.11.6	2	7
182.34.0.0	151.23.76.3	3	14
81.0.0.0	114.12.53.201	4	21
Default	127.3.11.6	2	7

2.3 routing algorithms

When it comes to routing, routers have to make a decision on which link an incoming should be sent. This decision is made, in general, in accordance with a routing table, table which specifies the outgoing link based on the prefix of the IP address of the packet's final destination. Since network conditions may change, these routing tables should be dynamic in nature. Not always. Either way, the content of these routing tables is set according to a routing algorithm, algorithm which takes in account information exchanged by the router with its neighbors.

Although there are many types of routing protocols, three major classes are in widespread use on IP networks:

1. Interior gateway routing via link-state routing protocols, such as OSPF and IS-IS
2. Interior gateway routing via path vector or distance vector protocols, such as RIP, IGRP and EIGRP
3. Exterior gateway routing. BGP v4 is the routing protocol used by the public Internet.

The specific characteristics of routing protocols include:

- the manner in which they either prevent routing loops from forming or break them up if they do
- the manner in which they select preferred routes, using information about hop costs
- the time they take to converge
- how well they scale up
- many other factors

The link-state protocols are performed by every switching node in the network (i.e. nodes that are prepared to forward packets; in the Internet, these are called routers). The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. The collection of best paths will then form the node's routing table.

This contrasts with distance-vector routing protocols, which work by having each node share its routing table with its neighbors. In a link-state protocol the only information passed between nodes is connectivity related. Link state algorithms are sometimes characterized informally as each router 'telling the world about its neighbors'.

2.4 distance vector routing

Distance vector algorithms use the Bellman-Ford algorithm. This approach assigns a number, the cost, to each of the links between each node in the network. Nodes will send information from point A to point B via the path that results in the lowest total cost (i.e. the sum of the costs of the links between the nodes used). The algorithm operates in a very simple manner. When a node first starts, it only knows of its immediate neighbors, and the direct cost involved in reaching them. (This information, the list of destinations, the total cost to each, and the next hop to send data to get there, makes up the routing table, or distance table.) Each node, on a regular basis, sends to each neighbor its own current idea of the total cost to get to all the destinations it knows of. The neighboring node(s) examine this information, and compare it to what they already 'know'; anything which represents an improvement on what they already have, they insert in their own routing table(s).

Over time, all the nodes in the network will discover the best next hop for all destinations, and the best total cost. When one of the nodes involved goes down, those nodes which used it as their next hop for certain destinations discard those entries, and create new routing-table information. They then pass this information to all adjacent nodes, which then repeat the process. Eventually all the nodes in the network receive the updated information, and will then discover new paths to all the destinations which they can still "reach".

Distance vector routing protocols

- RIP - Routing Information Protocol
- BGP - Border Gateway Protocol
- EIGRP – Enhanced Interior Gateway Routing Protocol – a proprietary CISCO routing

protocol, using bandwidth, delay, load, MTU (Maximum Transmission Unit) and reliability as metrics.

2.4.1 routing information protocol – RIP

RIP is a distance vector protocol that uses **hop count** as metric. It prevents routing loops by implementing a limit on the number of hops allowed in a path from the source to a destination. The maximum number of hops allowed for RIP is 15. This hop limit, however, also limits the size of networks that RIP can support.

Version 1 of RIP was specified in RFC 1058, while version 2 has been standardized in RFC 2453 (1994). RIPng (RIP next generation) is an extension of RIPv2 for IPv6 and is defined in RFC 2080 (1997).

RIP works this way:

- Each node keeps routing entry for each destination in the network
- Send routing table to neighbors regularly
- Neighbors compare entries

Routing tables consist of:

- Address – destination IP
- Gateway
- Interface – hardware interface to be used
- Metric – number of hops
- Timer – time of last update

Lately, RIP is no longer the preferred choice for routing, because of high convergence time and poor scalability.

2.4.2 border gateway protocol – BGP

BGP is the main inter-AS routing protocol. The first version of BGP dates back to 1989, as a replacement for **EGP** (Exterior Gateway Protocol). The current version is BGPv4 and is specified on a sequence of RFCs starting with RFC 1771 and ending with RFC 4271.

BGP (Border Gateway Protocol) is a protocol for exchanging routing information between gateway hosts (each with its own router) in a network of autonomous systems. BGP is often the protocol used between gateway hosts on the Internet. The routing table contains a list of known routers, the addresses they can reach, and a cost metric associated with the path to each router so that the best available route is chosen.

Hosts using BGP communicate using the Transmission Control Protocol (TCP) and send updated router table information only when one host has detected a change. Only the affected part of the routing table is sent. BGP-4, the latest version, lets administrators configure cost metrics based on policy statements. (BGP-4 is sometimes called BGP4, without the hyphen.)

BGP communicates with autonomous (local) networks using Internal BGP (IBGP) since it doesn't work well with IGP. The routers inside the autonomous network thus maintain two routing tables: one for the interior gateway protocol and one for IBGP.

BGP-4 makes it easy to use Classless Inter-Domain Routing (CIDR), which is a way to have more addresses within the network than with the current IP address assignment scheme.

2.5 link state routing

Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing. Two primary problems caused its demise. First, since the delay metric was queue length, it did not take line bandwidth into account when choosing routes. Initially, all the lines were 56 kbps, so line bandwidth was not an issue, but after some lines had been upgraded to 230 kbps and others to 1.544 Mbps, not taking bandwidth into account was a major problem. Of course, it would have been possible to change the delay metric to factor in line bandwidth, but a second problem also existed, namely, the algorithm often took too long to converge (the count-to-infinity problem). For these reasons, it was replaced by an entirely new type of algorithm, now called **link state routing**. Variants of link state routing are now widely used. The idea behind link state routing can be stated as five parts. Each router must do the following:

1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors.
3. Construct a packet telling all it has just learned.
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

Main facts

- In Link Stare routing, routers cooperate to maintain an updated map of the network.
- In Distance Vector routing, routers cooperate to maintain routing tables.
- The complexity of the LS algorithms is O(L * log(N)), where L is the number of links and N is the number of nodes.

Link state routing protocols

- OSPF
- IS-IS

2.5.1 open shortest path first – OSPF

OSPF is an interior gateway protocol, using a link state routing algorithm. It operates within the boundaries of an AS.

The current version of OSPF is version 2 (RFC 2328)(1998) for IPv4. The IPv6 update is called version 3 and is defined in RFC 5340 (2008).

OSPF is the most used interior gateway protocol in large enterprise networks. IS-IS, on the other side, is most common in large service providers networks.

OSPF gathers link state information from available routers and constructs a topology map of the network. The topology determines the routing table presented to the Internet Layer which makes routing decisions based solely on the destination IP address found in IP packets. OSPF was designed to support variable-length subnet masking (VLSM) or Classless Inter-Domain Routing (CIDR) addressing models.

OSPF detects changes in the topology, such as link failures, very quickly and converges on a new loop-free routing structure within seconds. It computes the shortest path tree for each route using a method based on Dijkstra's algorithm, a shortest path first algorithm.

OSPF does not use a TCP/IP transport protocol (UDP, TCP), but is encapsulated directly in IP datagrams with protocol number 89. This is in contrast to other routing protocols, such as the Routing Information Protocol (RIP), or the Border Gateway Protocol (BGP). OSPF handles its own error detection and correction functions.

2.5.2 intermediate system to intermediate system – IS-IS

IS-IS is an interior gateway protocol, designed for use within an administrative domain or network. This is in contrast to Exterior Gateway Protocols, primarily Border Gateway Protocol (BGP), which is used for routing between autonomous systems .

IS-IS is a link-state routing protocol, operating by reliably flooding link state information throughout a network of routers. Each IS-IS router independently builds a database of the network's topology, aggregating the flooded network information. Like the OSPF protocol, IS-IS uses Dijkstra's algorithm for computing the best path through the network. Packets (datagrams) are then forwarded, based on the computed ideal path, through the network to the destination.

IS-IS was standardized by ISO in 1992 in ISO 10589 for communication between Intermediate Systems (as opposed to end systems or hosts). IS-IS was developed at the same time with OSPF. IS-IS was later extended (in RFC 1195) to provide support for datagram routing in IP.

2.5.3 comparing IS-IS and OSPF

Both IS-IS and OSPF are link state protocols, and both use the same Dijkstra algorithm for computing the best path through the network. As a result, they are conceptually similar. Both support variable length subnet masks, can use multicast to discover neighboring routers using *hello packets*, and can support authentication of routing updates.

While OSPF is natively built to route IP and is itself a Layer 3 protocol that runs on top of IP, IS-IS is natively an OSI network layer protocol. The widespread adoption of IP worldwide may have contributed to OSPF's popularity. IS-IS does not use IP to carry routing information messages. IS-IS is neutral regarding the type of network addresses for which it can route. OSPF, on the other hand, was designed for IPv4. This allowed IS-IS to be easily used to support IPv6. To operate with IPv6 networks, the OSPF protocol was rewritten in OSPF v3 (in RFC 2740).

IS-IS routers build a topological representation of the network. This map indicates the subnets which each IS-IS router can reach, and the lowest-cost (shortest) path to a subnet is used to forward traffic.

IS-IS differs from OSPF in the way that "areas" are defined and routed between. IS-IS routers are designated as being: Level 1 (intra-area); Level 2 (inter area); or Level 1-2 (both). Level 2 routers are inter area routers that can only form relationships with other Level 2 routers. Routing information is exchanged between Level 1 routers and other Level 1 routers, and Level 2 routers only exchange information with other Level 2 routers. Level 1-2 routers exchange information with both levels and are used to connect the inter area routers with the intra area routers. In OSPF, areas are delineated on the interface such that an area border router (ABR) is actually in two or more areas at once, effectively creating the borders between areas inside the ABR, whereas in IS-IS area borders are in between routers, designated as Level 2 or Level 1-2. The result is that an IS-IS router is only ever a part of a single area. IS-IS also does not require Area 0 (Area Zero)

to be the backbone area through which all inter-area traffic must pass. The logical view is that OSPF creates something of a spider web or star topology of many areas all attached directly to Area Zero and IS-IS by contrast creates a logical topology of a backbone of Level 2 routers with branches of Level 1-2 and Level 1 routers forming the individual areas.

IS-IS also differs from OSPF in the methods by which it reliably floods topology and topology change information through the network. However, the basic concepts are similar.

OSPF has a larger set of extensions and optional features. However IS-IS is less "chatty" and can scale to support larger networks. Given the same set of resources, IS-IS can support more routers in an area than OSPF. This has contributed to IS-IS as an ISP-scale protocol.

The TCP/IP implementation, known as "Integrated IS-IS" or "Dual IS-IS", is described in RFC 1195.

chapter 3 Addresses and routing in IPv6

3.1 Ipv6 addresses

3.1.1 IPv6 address classes

IPv6 addresses are classified by the primary addressing and routing methodologies common in networking: unicast addressing, anycast addressing, and multicast addressing.[RFC4291]

- A **unicast** address identifies a single network interface. The Internet Protocol delivers packets sent to a unicast address to that specific interface.
- An **anycast** address is assigned to a group of interfaces, usually belonging to different nodes. A packet sent to an anycast address is delivered to just one of the member interfaces, typically the *nearest* host, according to the routing protocol's definition of distance. Anycast addresses cannot be identified easily, they have the same format of unicast addresses, and differ only by their presence in the network at multiple points. Almost any unicast address can be employed as an anycast address.
- A **multicast** address is also used by multiple hosts, which acquire the multicast address destination by participating in the multicast distribution protocol among the network routers. A packet that is sent to a multicast address is delivered to all interfaces that have joined the corresponding multicast group.

IPv6 does not implement **broadcast** addressing. Broadcast's traditional role is subsumed by multicast addressing to the *all-nodes* link-local multicast group ff02::1. However, the use of the all-nodes group is not recommended, and most IPv6 protocols use a dedicated link-local multicast group to avoid disturbing every interface in the network.

3.1.2 address formats

The IPv6 address is 128 bits long and are (in general) represented by a sequence of 8 2-byte groups. A typical example: **0f01:0db8:85a3:0000:0000:8a2e:0370:7334**. A zero group may be represented by a single zero digit, like in **0f01:0db8:85a3:0:0:8a2e:0370:7334**. If there is a single consecutive sequence of zeroes of interest, the can be replaced by ::, as in this representation of the same address: **0f01:0db8:85a3::8a2e:0370:7334**. Sometimes, when it helps, two groups of 2 bytes may be replaced by a dotted decimal representation, like this one: **::ffff:192.0.2.128**, instead of **::ffff:c000:280**.

Unicast and **anycast** addresses have a 64-bit network prefix used for routing and a 64-bit interface identifier.

bits	48 (or more)	16 (or less)	64
fields	Routing prefix	Subnet ID	Interface ID

The bits of the subnet ID are used by the network admin to size the subnets. The 64-bit interface ID is generated (in general) from the MAC (physical address) of the interface, using the

modified EUI-64 format (an **FF:FE** group inserted in the middle of the MAC address). It may be assigned manually or randomly, as well, if desired so.

Link-local addresses have a 64-bit network prefix used for routing and a 64-bit interface identifier.

bits	10	54	64
fields	Prefix	Zeroes	Interface ID

Multicast addresses are formed according to several specific formatting rules, depending on the application.

bits	8	4	4	112
fields	prefix	Flag	Scope	Group ID

3.1.3 IPv6 address scopes

Every IPv6 address, except the unspecified address (::), has a "scope", which specifies in which part of the network it is valid.

In the unicast addressing class, link-local addresses and the loopback address have *link-local* scope, which means they are to be used in the directly attached network (link). All other addresses (except Unique local addresses) have *global* (or *universal*) scope, which means they are globally routable, and can be used to connect to addresses with *global* scope anywhere, or addresses with *link-local* scope on the directly attached network.

Unique local addresses are not globally routable, so their scope is limited to the extent of the network(s) in which they are used. These addresses will only be routed by routers or tunnels whose routing tables have been specifically configured to allow it.

The scope of an anycast address is defined identically to that of a unicast address.

For multicasting, the four least-significant bits of the second address octet of a multicast address (ff0*s*::) identify the address scope, i.e. the span over which the multicast address is propagated. Currently defined scopes are:

Value	Scope name
0x0	Reserved
0x1	Interface-local
0x2	Link-local
0x4	Admin-local
0x5	Site-local
0x8	Organization-local
0xe	Global
0xf	Reserved

For more details on IPv6 addresses, check - http://en.wikipedia.org/wiki/IPv6_address .

3.2 routing in IPv6

3.2.1 basics

While IPv4 had to face two main problems, namely a shortage of address space and the size of routing tables, routing in IPv6 is somewhat different, mainly because of the generous size of the address space.

In terms of algorithms and techniques, routing is similar with IPv4 routing, with the following amendments:

- similar to IPv4 routing with CIDR
- minimal changes to dynamic routing protocols
- improved source routing option (routing header)

3.2.2 unicast routing

The unicast routing model is defined in RFC 2374 [RFC2374]. It is strictly hierarchical with three levels:

- **Public topology** – providers and exchangers offering Internet transit services.
- **Site topology** – local topology not offering services to nodes external to the organization.
- **Interface ID** – assigned to any interface connected to the internet.

The public topology prefix consists of the following fields:

- **FP** – 3 bits - Format prefix
- **TLA ID** – 13 bits - Top Level Aggregation ID
- **RESV** – 8 bits - Reserved - to enlarge TLA to NLA
- **NLA ID** – 24 bits - Next Level Aggregation ID

The site topology part consists of the fields:

- **SLA ID** – 16 bits - Site Level Aggregation ID
- **IFC ID** – 64 bits - Interface ID

There are two types of aggregation:

- Per Provider – addresses depend on the provider you are connected to
- Per Exchange – addresses depend on the Exchange we are connected to

Therefore, when changing the Provider or the Exchanger, the network has to be **renumbered**.

3.2.3 anycast addresses

- Anycast addresses are unicast addresses assigned to several interfaces (belonging to different nodes, in general)
- A packet sent to an anycast address should reach the nearest interface with that particular

address

- Anycast addresses are experimental

3.3 IPv6 dynamic routing protocols

In general, the existing IPv4 routing protocols need minor changes to be adapted to IPv6 routing.

- Changes are related to address format
- in the case of integrated routing, the changes provide support for both IPv4 and IPv6

3.3.1 RIPng

- defined in RFC 2080 [RFC2080]
- minimal changes versus RIP
- IGP (Interior Gateway Protocol) used in small and static LAN
- based on distance vector algorithm – convergence problems
- several implementations
  - GATEd
  - MRTd
  - Kame routed6d
  - Zebra
  - Cisco

3.3.2 OSPFv6

OSPVv6 is just OSPFv2 for IPv6 and is defined in RFC 2740 [RFC2740].

- IGP recommended by IETF:
  - based on link-state algorithms for fast convergence
  - network divided in areas – good for scalability
- minimal changes:
  - address format, prefixes, lds, etc.
  - authentication eliminated
- it does not use Integrated Routing – two copies of OSPF running for v4 and v6
- several implementations
  - Ericsson – Telebit
  - IBM
  - Zebra

- Gated
- MRTd
- Cisco

3.3.3 inter domain routing BGP4+

- Used between ISPs and between ISP and large companies
- Changes
  - RFC 2858 [RFC2858] defines multiprotocol extensions (IPv6, IPX, etc.) to BGP-4
  - RFC 2545 [RFC2545] defines how to use extensions for IPv6 (Scopes, Next Hop, etc.)
- used in 6BONE and in main IPv6 exchanges
- several implementations
  - GateD
  - MRTd
  - Kame BGPd
  - Zebra
  - Cisco

3.4 relationship between addresses and routing in IPv6

3.4.1 the global network structure

The Internet is organized into routing domains that exchange information on the reachability of networks on which they are composed. These routing domains do not have equal importance, and we have already seen that IDRP makes a distinction between **Transit Routing Domain** (TRD) and **End Routing Domain** (ERD).

ERDs are associated with the network's end users—that is, to organizations connected to the Internet that usually have connections with only one TRD. Sometimes an ERD can have connections with many TRDs; in this case, the ERD is called multihomed. It, however, maintains its ERD nature—that is, it doesn't operate as a transit domain—and it therefore remains a leaf.

Another possibility is that two ERDs have a private link because they have to exchange large volumes of traffic, without passing through the Internet.

TRDs are usually associated with Internet Service Providers (ISPs); in the following text, we will simply call them providers.These providers can be subdivided into the following categories:

- **Direct Service Providers**: These providers connect end users and connect themselves to international backbones. Examples of Direct Service Providers are America Online and NSFnet regional.
- **Indirect Service Providers**: These providers administer large international backbones. They connect only DSPs and large users.

3.4.2 IPv4 issues

In IPv4, no relationship exists between addresses and topology. In fact, addresses are directly assigned to end users and, even if an effort is made to assign addresses by nations or continents, this use poses no particular benefits for routing. The Internet, by its nature, doesn't respect nations' political borders. For example, Italian organizations can connect to Italian providers and these to European providers, but they can also connect to American providers. As a result, Italian networks are announced partly in Europe and partly in the United States. This situation is likely to become more and more complicated with the coming of a telecommunications free market.

In this situation, ERD routers don't present any particular drawbacks; in fact, it is sufficient that they maintain in their routing table one entry for each network within the ERD and one default network for all other networks. The default entry points to the TRD of the provider to which the ERD is connected.

The case of TRD routers (also called core routers) is more complex. In fact, they must maintain in their routing tables one entry for each network connected to the Internet (this is undoubtedly true for Indirect Service Provider routers). Therefore, the routing tables tend to explode with the dizzying growth of the Internet.

To limit the growth of routing tables, the Classless Inter-Domain Routing (CIDR) was introduced with BGP-4. The CIDR allows grouping of announcements of many networks whose addresses are contiguous in only one entry. Nevertheless, the CIDR cannot bring important benefits due to the assignment philosophy of IPv4 addresses. In fact, it is not sure that contiguous addresses are assigned to users connected to the same TRD and that the TRD can therefore group them.

3.4.3 The IPv6 Solution

To solve the problems cited in the previous section, IPv6 migrates from a scheme based on the assignment of addresses to end users (like that of IPv4) to a **provider-based scheme**. In this new scheme, each Direct Service Provider is assigned a set of addresses that it divides into smaller sets to be assigned to its users. Because the IPv6 address is much longer than the IPv4 address, it can easily contain this new hierarchy level. Sets of addresses assigned to the users can be grouped by definition by the provider because they are the result of a partition.

For ERDs' routers, the situation remains unchanged. They continue to have one entry for each network within the ERD, one default entry toward the TRD, and they announce their set of addresses to the TRD with only one entry.

For Indirect Service Providers' TRD routers, the situation is completely different. In fact, now each Direct Service Provider announces all its networks with only one entry; therefore, the size of routing tables is proportional to the number of providers, not to the number of networks.

For the Direct Service Provider's TRD routers, the situation can change significantly if many connections are made with other providers (either Direct or Indirect). In fact, all networks associated with a provider are announced with a single entry in routing tables in this case.

Other possible aggregation schemes have been proposed. For example, providers can be aggregated on a continental basis, or Indirect Service Providers can be assigned address sets to

be subdivided by assigning the addresses to Direct Service Providers, and the Direct Service Providers, in their turn, can assign the addresses to end users. The usefulness of these schemes is questionable.

What is not questionable, however, is that the providers' assignment of addresses to end users brings about a significant containment of routing tables (that we can estimate in two orders of magnitude). IPv6 will therefore follow this approach.

3.4.4 drawbacks for users

The main drawback for users happens when they decide to change providers—that is, to buy Internet services from another ISP. In fact, users have to renumber their networks. This operation is simplified by IPv6 Neighbor Discovery, but it still can cause some inefficiency.

Nevertheless, a user can operate with addresses from provider A while still being connected to provider B. In this case, provider B must explicitly announce addresses assigned to the user by provider A. All Internet routers should have one additional entry to indicate that the user, though having addresses from provider A, can still be reached through provider B. This situation can occur for a limited period of time during a transition to allow the user to renumber networks without service interruptions; however, this situation cannot continue indefinitely because it will rapidly recreate the unacceptable growth of routing tables, as in the previously analyzed IPv4 case.

3.4.5 multihomed routing domains

The previously discussed theories apply to ERDs that are connected to only one TRD. However, what happens when we want an ERD to be multihomed—that is, to be connected to many TRDs—without becoming a TRD, but remaining a leaf routing domain?

Examples of multihomed ERDs are routing domains in a big organization covering the whole nation that decides to connect to the Internet in many points through different providers, or even that of an international organization that decides to connect its network to the Internet in the nations where its main subsidiaries are located.

There are several reasons to have an ERD multihomed. The two main reasons are the larger availability of bandwidth, and the possibility of having alternative paths in case of errors and, therefore, a more reliable network.

In IPv6, an entire domain can be multihomed, but also a single subnet or a single host can be. A multihomed host can, in turn, be multihomed because it has many IPv6 addresses assigned to different interfaces (this case is common in reliable hosts) or because it has many addresses associated with the same interface (for example, a LAN with many prefixes associated with different providers). This topic is still the subject of debate in the Internet community, and at the time this chapter was written, only an Internet Draft on this topic is available.

RFC 1887 provides four possible solutions for connecting an ERD to many TRDs. C. Huitema, who highlights the existing implications between multihoming and upper layer protocols, proposes a fifth solution.

3.5 possible solutions

3.5.1 SOLUTION #1

A multihomed organization obtains a prefix independently of the providers to which it is connected. This solution causes an additional entry in all core routers, and it is acceptable only for a few very large organizations. This solution does not scale to all organizations that will connect to the Internet in the future and that want to be multihomed because many hundreds of thousands of organizations could want this capability.

3.5.2 SOLUTION #2

The organization is assigned as many different prefixes as there are providers it will be connected to. In each part of the network, the organization will use a prefix chosen on the basis of the distance of that part of the network to a particular provider. For example, let's suppose that an organization has a network covering Italy, France, and Spain, and that it wants to be connected to the Internet in these three nations. For the Italian part of the network, it will use addresses derived from the set it has been assigned by an Italian provider; for the French part, addresses from a French provider; and for the Spanish part, addresses from a Spanish provider.

For this solution, core routers don't need to maintain any additional information for the organization because it will be reached as three separate organizations that are part of three different providers. Routers within the organization can be efficiently configured by using private links, without upgrading the ERD to a TRD.

The main disadvantage of this solution is the lack of backup mechanisms in case one of the three connections with the providers fails. The part of the network configured with addresses of that provider simply becomes unreachable because those addresses are not announced by the other two providers. Announcing them would be possible, but doing so would be much more expensive than in the preceding case because core routers should maintain three entries for the organization, one for each prefix used on the network. Moreover, if a provider is changed, all addresses associated with that provider should be changed, too.

Also, note that, with the previous approach, packets enter the organization via the point that is closest to the source node (which tends to maximize the load on the internal network); with this second solution, packets enter the organization via the point that is closest to the destination node (which tends to maximize the load on the Internet).

3.5.3 SOLUTION #3

Now suppose that a second organization uses provider A's prefix as the prefix for its networks because provider A is meant to be used as the default to the Internet. Other TRDs to which this organization is connected will advertise A's prefix only in restricted and controlled areas. For example, let's suppose that this organization also belongs to the Italian Public Administration network, administered by provider B. Provider B will advertise, within the public administration network, that this organization can be reached by a set of addresses from provider A. This capability entails that routers of the TRD of B have an explicit entry in routing tables for the organization, but it doesn't introduce any additional entry on core routers.

3.5.4 SOLUTION #4

The fourth solution can be used when two or more providers have many customers in common. This solution is hypothetical and will become fairly common when the use of IPv6 on public networks is more widespread. In this case, the two providers request a third set of addresses (in addition to the two they already have) to be assigned to customers they have in common and interconnect their TRDs. There is no penalty at the core router level because all users in common between the two providers are advertised with only one entry in the routing tables.

3.5.5 SOLUTION #5

For the fifth solution, each station is assigned as many addresses as there are providers. This situation is illustrated in Figure 7-6, where station X has two addresses: A::X derived from provider A and B::X derived from provider B.

This solution is not perfect. Suppose that X establishes a Telnet session with Y by using its address A::X. If, during the session, provider A becomes overloaded or it cannot reach X through A, the session cannot be rerouted using provider B. This operation will entail the use of address B::X in the IPv6 packet instead of the A::X address, but this use is not possible. In fact, the Telnet application lays on the Transmission Control Protocol (TCP), which also uses the IPv6 address as the connection identifier; according to RFC 793, this address cannot be modified during the connection itself.

A less pragmatic solution is to close the Telnet session and to open another one, this time using the address B::X. A second solution, currently under discussion, is to modify the TCP protocol allowing IPv6 addresses to change during the connection.

A third possibility is that Y inserts a Routing Header to force the routing to pass through B::X. In this way, the destination address in the IPv6 packet remains A::X, but the packet is delivered to B::X, which routes it within itself to A::X—that is, to itself. The only drawback to this solution is represented by the routing header overhead (24 octets in the case of a single intermediate address).

chapter 4 Multicast

Multicast is the delivery of a message or information to a group of destination addresses simultaneously in a single transmission from the source. Data replication can be performed by other network elements, such as routers, when the network topology requires it.

4.1 when does it make sense?

Multicasting is convenient in several situations, namely:

- streaming media
- software distribution services
- application or media stores

The obvious area of convenience is streaming media, when a real time content is delivered to a set of subscribers. This can include a live event or a prerecorded event which is aired at a predefined time. The client list can be an open one, as in the case of free services, or can be a subset of an existing customer base.

Free software distribution services can create, using a delay schema, an ad-hoc group of clients to whom the requested content (like free anti-virus software or free operating systems) is sent simultaneously. This group of clients can be itself dynamic while the content of the delivery is constant. What is specific for case is the unpredictable nature of the clients, since anyone can request anything and at any time.

Application and media stores are different in that the customer base is known in advance, based on registration or payment requirements. The size of the prospective client pool can be further reduced if some sort of login is requested or can be expected prior to submitting an order.

4.2 how does the standard approach work?

Internet Protocol (IP) multicasting is the sending of a single datagram to multiple hosts on a network or internetwork. Of the three delivery methods supported by IP (the other beeing unicast and broadcast), multicasting is the method that is most practical for one-to-many delivery.

Before sending or receiving IP multicast data, a network must be enabled for multicasting, as follows:

- Hosts must be configured to send and receive multicast data.
- Routers must support the Internet Group Membership Protocol (IGMP), multicast forwarding, and multicast routing protocols.

In an IP multicast-enabled intranet, any host can send IP multicast datagrams, and any host can receive IP multicast datagrams, including sending and receiving across the Internet.

The source host sends multicast datagrams to a single Class D IP address, known as the group address. Any host that is interested in receiving the datagrams contacts a local router to

join the multicast group and then receives all subsequent datagrams sent to that address.

Routers use a multicast routing protocol to determine which subnets include at least one interested multicast group member and to forward multicast datagrams only to those subnets that have group members or a router that has downstream group members. The IP header of a multicast datagram includes a Time-to-Live (TTL) value that determines how far routers can forward a multicast datagram.

4.2.1 levels of conformance.

Hosts can be in three different levels of conformance with the Multicast specification, according to the requirements they meet.

Level 0 is the "no support for IP Multicasting" level. Lots of hosts and routers in the Internet are in this state, as multicast support is not mandatory in IPv4 (it is, however, in IPv6). Not too much explanation is needed here: hosts in this level can neither send nor receive multicast packets. They must ignore the ones sent by other multicast capable hosts.

Level 1 is the "support for sending but not receiving multicast IP datagrams" level. Thus, note that it is not necessary to join a multicast group to be able to send datagrams to it. Very few additions are needed in the IP module to make a "Level 0" host "Level 1-compliant".

Level 2 is the "full support for IP multicasting" level. Level 2 hosts must be able to both send and receive multicast traffic. They must know the way to join and leave multicast groups and to propagate this information to multicast routers. Thus, they must include an Internet Group Management Protocol (IGMP) implementation in their TCP/IP stack.

4.2.2 sending multicast datagrams

Mmulticast traffic is handled at the transport layer with UDP, as TCP provides point-to-point connections, not feasible for multicast traffic. (Heavy research is taking place to define and implement new multicast-oriented transport protocols).

In principle, an application just needs to open a UDP socket and fill with a class D multicast address - the destination address where it wants to send data to. However, there are some operations that a sending process must be able to control.

TTL

The TTL (Time To Live) field in the IP header has a double significance in multicast. As always, it controls the live time of the datagram to avoid it being looped forever due to routing errors. Routers decrement the TTL of every datagram as it traverses from one network to another and when its value reaches 0 the packet is dropped.

The TTL in IPv4 multicasting has also the meaning of "threshold". Its use becomes evident with an example: suppose you set a long, bandwidth consuming, video conference between all the hosts belonging to your department. You want that huge amount of traffic to remain in your LAN. Perhaps your department is big enough to have various LANs. In that case you want those hosts belonging to each of your LANs to attend the conference, but in any case you do not want to collapse the entire Internet with your multicast traffic. There is a need to limit how "long" multicast traffic will expand across routers. That's what the TTL is used for. Routers have a TTL

threshold assigned to each of its interfaces, and only datagrams with a TTL greater than the interface's threshold are forwarded.

A list of TTL thresholds and their associated scope follows:

TTL	Scope
0	Restricted to the same host. Won't be output by any interface.
1	Restricted to the same subnet. Won't be forwarded by a router.
<32	Restricted to the same site, organization or department.
<64	Restricted to the same region.
<128	Restricted to the same continent.
<255	Unrestricted in scope. Global.

Nobody knows what "site" or "region" mean exactly. It is up to the administrators to decide what this limits apply to.

The TTL-trick is not always flexible enough for all needs, specially when dealing with overlapping regions or trying to establish geographic, topologic and bandwidth limits simultaneously. To solve this problems, administratively scoped IPv4 multicast regions were established in 1994. It does scoping based on multicast addresses rather than on TTLs. The range 239.0.0.0 to 239.255.255.255 is reserved for this administrative scoping.

Loopback.

When the sending host is Level 2 conformant and is also a member of the group datagrams are being sent to, a copy is looped back by default. This does not mean that the interface card reads its own transmission, recognizes it as belonging to a group the interface belongs to, and reads it from the network. On the contrary, is the IP layer which, by default, recognizes the to-be-sent datagram and copies and queues it on the IP input queue before sending it.

This feature is desirable in some cases, but not in others. So the sending process can turn it on and off at wish.

Interface selection.

Hosts attached to more than one network should provide a way for applications to decide which network interface will be used to output the transmissions. If not specified, the kernel chooses a default one based on system administrator's configuration.

4.2.3 receiving multicast datagrams.

Joining a Multicast Group.

Broadcast is (in comparison) easier to implement than multicast. It doesn't require processes to give the kernel some rules regarding what to do with broadcast packets. The kernel just knows what to do: read and deliver all of them to the proper applications.

With multicast, however, it is necessary to advise the kernel which multicast groups we are interested in. That is, we have to ask the kernel to "join" those multicast groups. Depending on the underlying hardware, multicast datagrams are filtered by the hardware or by the IP layer (and, in some cases, by both). Only those with a destination group previously registered via a join are accepted.

Essentially, when we join a group we are telling the kernel: "OK. I know that, by default, you ignore multicast datagrams, but remember that I am interested in this multicast group. So, do read and deliver (to any process interested in them, not only to me) any datagram that you see in this network interface with this multicast group in its destination field".

Some considerations: first, note that you don't just join a group. You join a group on a particular network interface. Of course, it is possible to join the same group on more than one interface. If you don't specify a concrete interface, then the kernel will choose it based on its routing tables when datagrams are to be sent. It is also possible that more than one process joins the same multicast group on the same interface. They will all receive the datagrams sent to that group via that interface.

As said before, any multicast-capable hosts join the all-hosts group at start-up, so "pinging" 224.0.0.1 returns all hosts in the network that have multicast enabled.

Finally, consider that for a process to receive multicast datagrams it has to ask the kernel to join the group and bind the port those datagrams were being sent to. The UDP layer uses both the destination address and port to demultiplex the packets and decide which socket(s) deliver them to.

Leaving a Multicast Group.

When a process is no longer interested in a multicast group, it informs the kernel that it wants to leave that group. It is important to understand that this doesn't mean that the kernel will no longer accept multicast datagrams destined to that multicast group. It will still do so if there are more processes who issued a "multicast join" petition for that group and are still interested. In that case the host remains member of the group, until all the processes decide to leave the group.

Even more: if you leave the group, but remain bound to the port you were receiving the multicast traffic on, and there are more processes that joined the group, you will still receive the multicast transmissions.

The idea is that joining a multicast group only tells the IP and data link layer (which in some cases explicitly tells the hardware) to accept multicast datagrams destined to that group. It is not a per-process membership, but a per-host membership.

Mapping of IP Multicast Addresses to Ethernet/FDDI addresses.

Both Ethernet and FDDI frames have a 48 bit destination address field. In order to avoid a kind of multicast ARP to map multicast IP addresses to ethernet/FDDI ones, the IANA reserved a range of addresses for multicast: every ethernet/FDDI frame with its destination in the range 01-00-5e-00-00-00 to 01-00-5e-ff-ff-ff (hex) contains data for a multicast group. The prefix 01-00-5e identifies the frame as multicast, the next bit is always 0 and so only 23 bits are left to the multicast address. As IP multicast groups are 28 bits long, the mapping can not be one-to-one. Only the 23 least significant bits of the IP multicast group are placed in the frame. The remaining 5 high-order bits are ignored, resulting in 32 different multicast groups being mapped to the same ethernet/FDDI address. This means that the ethernet layer acts as an imperfect filter, and the IP

layer will have to decide whether to accept the datagrams the data-link layer passed to it. The IP layer acts as a definitive perfect filter.

Full details on IP Multicasting over FDDI are given in RFC 1390: "Transmission of IP and ARP over FDDI Networks". For more information on mapping IP Multicast addresses to ethernet ones, you may consult draft-ietf-mboned-intro-multicast-03.txt: "Introduction to IP Multicast Routing".

If you are interested in IP Multicasting over Token-Ring Local Area Networks, see RFC 1469 for details.

4.3 IP multicast groups and addresses

Every IP multicast group has a **group address**. IP multicast provides only open groups: That is, it is not necessary to be a member of a group in order to send datagrams to the group.

Multicast address are like IP addresses used for single hosts, and is written in the same way: A.B.C.D. Multicast addresses will never clash with host addresses because a portion of the IP address space is specifically reserved for multicast. This reserved range consists of addresses from 224.0.0.0 to 239.255.255.255 (the four high-order bits always set to **1110**).

The addresses in this range are also known as class D IP addresses. In network prefix or classless interdomain routing (CIDR) notation, IP multicast addresses are summarized as 224.0.0.0/4.

Within the Class D range of addresses, certain ranges of multicast addresses have special meaning, as follows:

1. Addresses in the range of 224.0.0.0 to 224.0.0.255 (224.0.0.0/24) are reserved for local subnet multicast traffic. Datagrams sent to addresses in this range are not forwarded by IP routers.
2. Addresses in the range of 224.0.1.0 to 238.255.255.255 are known as globally scoped addresses, which means they can be used for multicasting across an intranet or the Internet. Some addresses within this range are reserved by the Internet Assigned Numbers Authority (IANA) for special purposes.
3. Addresses in the range of 239.0.0.0 to 239.255.255.255 (239.0.0.0/8) are reserved for administratively scoped addresses. Administratively scoped addresses as defined by RFC 2365 are used to prevent the forwarding of multicast traffic across boundaries configured for the address. Multicast boundaries are described in more detail later in this document.

The following are examples of reserved IP multicast addresses:

- 224.0.0.1 - All hosts on this subnet
- 224.0.0.2 - All routers on this subnet
- 224.0.0.5 - Open Shortest Path First (OSPF) version 2, designed to reach all OSPF routers
- 224.0.0.6 - OSPF version 2, designed to reach all OSPF-designated routers and designated backup routers
- 224.0.0.9 - Routing Information Protocol (RIP) version 2

- 224.0.1.1 - Network Time Protocol

4.4 IGMP – Internet Group Management Protocol

4.4.1 basics

The **Internet Group Management Protocol** (**IGMP**) is a communications protocol used by hosts and adjacent routers on IP networks to establish multicast group memberships.

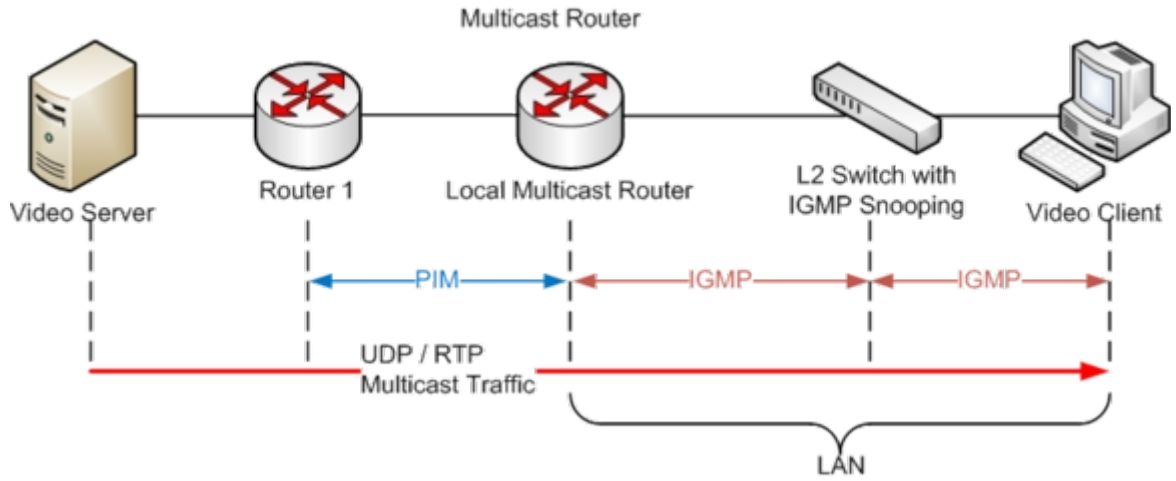
IGMP is an integral part of the IP multicast specification. It is analogous to ICMP for unicast connections. IGMP can be used for online streaming video and gaming, and allows more efficient use of resources when supporting these types of applications.

IGMP is used on IPv4 networks. Multicast management on IPv6 networks is handled by Multicast Listener Discovery (MLD) which uses ICMPv6 messaging contrary to IGMP's bare IP encapsulation.

There are three versions of IGMP, as defined by RFC documents of the Internet Engineering Task Force (IETF). IGMPv1 is defined by RFC 1112, IGMPv2 is defined by RFC 2236 and IGMPv3 was initially defined by RFC 3376 and has been updated by RFC 4604 which defines both IGMPv3 and MLDv2. IGMPv2 improves over IGMPv1 by adding the ability for a host to signal desire to leave a multicast group. IGMPv3 improves over IGMPv2 mainly by adding the ability to listen to multicast originating from a set of source IP addresses.

4.4.2 architecture

A network designed to deliver a multicast service using IGMP might use this basic architecture:



A local multicast router maintains, for a given multicast group (or class D IP address) the list of those interfaces (IP addresses) from the local network that have joined that particular group.

IGMP operates between the client computer and a local multicast router. Switches featuring IGMP snooping derive useful information by observing these IGMP transactions. Protocol

Independent Multicast (PIM) is then used between the local and remote multicast routers, to direct multicast traffic from the multicast server to many multicast clients.

4.4.3 how does IGMP work?

The membership of a host group is **dynamic**; that is, hosts may join and leave groups at any time. There is no restriction on the location or number of members in a host group, but membership in a group may be restricted to only those hosts possessing a private access key. A host may be a member of more than one group at a time. A host need not be a member of a group to send datagrams to it.

A host group may be **permanent** or **transient**. A permanent group has a well-known, administratively assigned IP address. It is the address, not the membership of the group, that is permanent; at any time a permanent group may have any number of members, even zero. A transient group, on the other hand, is assigned an address dynamically when the group is created, at the request of a host. A transient group ceases to exist, and its address becomes eligible for reassignment, when its membership drops to zero.

The creation of transient groups and the maintenance of group membership information is the responsibility of "**multicast agents**", entities that reside in internet gateways or other special-purpose hosts. There is at least one multicast agent directly attached to every IP network or subnetwork that supports IP multicasting. A host requests the creation of new groups, and joins or leaves existing groups, by exchanging messages with a neighboring agent.

Multicast agents retrieve data from devices participating in multicast groups and routes. The agents that retrieve multicast data need SNMP and Ping access to retrieve the data.

Multicast agents are also responsible for internetwork delivery of multicast IP datagrams. When sending a multicast IP datagram, a host transmits it to a local network multicast address which identifies all neighboring members of the destination host group. If the group has members on other networks, a multicast agent becomes an additional recipient of the local multicast and relays the datagram to agents on each of those other networks, via the internet gateway system. Finally, the agents on the other networks each transmit the datagram as a local multicast to their own neighboring members of the destination group.

4.5 PIM – Protocol Independent Multicast

PIM is a family of multicast routing protocols for Internet Protocol (IP) networks that provide one-to-many and many-to-many distribution of data over a LAN, WAN or the Internet. It is termed *protocol-independent* because PIM does not include its own topology discovery mechanism, but instead uses routing information supplied by other traditional routing protocols such as the RIP, OSPF, BGP and Multicast Source Discovery Protocol (MSDP).

There are four variants of PIM:

- **PIM Sparse Mode** (PIM-SM) explicitly builds unidirectional shared trees rooted at a *rendez-vous point* (RP) per group, and optionally creates shortest-path trees per source. PIM-SM generally scales fairly well for wide-area usage. See the PIM Internet Standard RFC 4601.

- **PIM Dense Mode** (PIM-DM) uses dense multicast routing. It implicitly builds shortest-path trees by flooding multicast traffic domain wide, and then pruning back branches of the tree where no receivers are present. PIM-DM is straightforward to implement but generally has poor scaling properties. The first multicast routing protocol, DVMRP used dense-mode multicast routing. See the PIM Internet Standard RFC 3973.
- **Bidirectional PIM** explicitly builds shared bi-directional trees. It never builds a shortest path tree, so may have longer end-to-end delays than PIM-SM, but scales well because it needs no source-specific state. See Bidirectional PIM Internet Standard RFC 5015.
- **PIM source-specific multicast** (PIM-SSM) builds trees that are rooted in just one source, offering a more secure and scalable model for a limited amount of applications (mostly broadcasting of content). In SSM, an IP datagram is transmitted by a source S to an SSM destination address G, and receivers can receive this datagram by subscribing to channel (S,G). See informational RFC 3569

4.5.1 sparse mode

This is the most widespread PIM deployment. This variant is suitable for groups where a very low percentage of the nodes (and their routers) will subscribe to the multicast session. Unlike earlier dense-mode multicast routing protocols such as DVMRP and dense multicast routing which flooded packets across the network and then pruned off branches where there were no receivers, PIM-SM explicitly constructs a tree from each sender to the receivers in the multicast group.

Multicast clients

A router receives explicit Join/Prune messages from those neighboring routers that have downstream group members.

- In order to join a multicast group, G, a host conveys its membership information through the Internet Group Management Protocol (IGMP).
- The router then forwards data packets addressed to a multicast group G to only those interfaces on which explicit joins have been received.
- A Designated Router (DR) sends periodic Join/Prune messages toward a group-specific Rendezvous Point (RP) for each group for which it has active members.
  - Note that one router will be automatically or statically designated as the rendezvous point (RP), and all routers must explicitly join through the RP.
- Each router along the path toward the RP builds a wild card (any-source) state for the group and sends Join/Prune messages on toward the RP.
  - The term route entry is used to refer to the state maintained in a router to represent the distribution tree.
  - A route entry may include such fields as:
    - source address
    - the group address
    - the incoming interface from which packets are accepted

        - the list of outgoing interfaces to which packets are sent
        - timers, flag bits, etc.
    - The wild card route entry's incoming interface points toward the RP
    - The outgoing interfaces point to the neighboring downstream routers that have sent Join/Prune messages toward the RP as well as the directly connected hosts which have requested membership to group G.
- This state creates a shared, RP-centered, distribution tree that reaches all group members.

Multicast sources

- When a data source first sends to a group, its Designated Router (DR) unicasts Register messages to the Rendezvous Point (RP) with the source's data packets encapsulated within.
- If the data rate is high, the RP can send source-specific Join/Prune messages back towards the source and the source's data packets will follow the resulting forwarding state and travel un-encapsulated to the RP.
- Whether they arrive encapsulated or natively, the RP forwards the source's de-capsulated data packets down the RP-centered distribution tree toward group members.
- If the data rate warrants it, routers with local receivers can join a source-specific, shortest path, distribution tree, and prune this source's packets off the shared RP-centered tree.
- For low data rate sources, neither the RP, nor last-hop routers need join a source-specific shortest path tree and data packets can be delivered via the shared RP-tree.

Once the other routers which need to receive those group packets have subscribed, the RP will unsubscribe to that multicast group, unless it also needs to forward packets to another router or node. Additionally, the routers will use reverse-path forwarding to ensure that there are no loops for packet forwarding among routers that wish to receive multicast packets.

4.6 XCAST

4.6.1 basics

Explicit Multi-Unicast (XCAST) is an alternate multicast strategy to IP multicast that provides reception addresses of all destinations with each packet. As such, since the IP packet size is limited in general, XCAST cannot be used for multicast groups of large number of destinations.

XCAST is defined in RFC 5058 – [RFC5058].

4.6.2 how does it work?

The XCAST model generally assumes that the stations participating in the communication are known ahead of time, so that distribution trees can be generated and resources allocated by network elements in advance of actual data traffic.

- Perform a route table lookup in the Xcast routing table to determine the Xcast next hop for

each of the destinations listed in the packet.

- If no Xcast next hop is found, replicate the packet and send a standard unicast to the destination.
- For those destinations for which an Xcast next hop is found, partition the destinations based on their next hops.
- Replicate the packet so that there's one copy of the packet for each of the Xcast next hops found in the previous steps.
- Modify the list of destinations in each of the copies so that the list in the copy for a given next hop includes just the destinations that ought to be routed through that next hop.
- Send the modified copies of the packet on to the next hops.

4.6.3 advantages

- The routers do not need to keep information for every session or channel. This makes Xcast very scalable about the number of sessions it can support.
- There is no need to make a direction assignment.
- They don't need protocols for multicast routing. They are routed correctly thanks to the common unicast protocols.
- There is no critic node. Xcast minimizes the network latencies and maximizes efficiency.
- Symmetric paths are not required.
- With traditional IP multicast routing protocols it is necessary to establish a communication between unicast and multicast routing protocols. That means a slow error recovery. Xcast reacts immediately to unicast routing changes.
- Easier security and register. With Xcast all sources know the channel members and all routers are able to know the number of times each packet has been duplicated in its domain.
- The receptors can be heterogeneous since Xcast allows that every receptor is able to have its own requirements of service in a single multicast channel.
- Simplicity when implementing reliable protocols over Xcast.
- Flexibility: unicast, multicast and Xcast represent costs of bandwidth, signalization and processing respectively. Depending on how the network is built or how it is in a certain moment, it may be better to use a system or another. Xcast is just another alternative.
- Easy transition between different mechanisms.

4.6.4 disadvantages

- They have got big headers. Each packet contains all the remaining destinations.
- It requires a more complex header processing. Every direction needs a look up to the routing table, so it is needed the same number of consults as it was unicast, furthermore, a new header must be generated after every jump. But on the other hand:

    - Xcast is designed for sessions with few users, so in many routers the headers will only have just one address.
    - The header building can become a very easy operation, overwrite a bit map.
    - When the packet reaches a region where the bandwidth is not limited, the packet can become a premature X2U.
- Limits the session to just a few users.

4.7 a new approach - hybrid

This approach follows the XCAST model but the destination list is managed in a different way. The idea is that the actual destination list is encoded as a bit mask relative to an existing (or just created) multicast group.

- each packet contains a message (file) identifier
- the router will identify the packet as belonging to a particular mc group
- once a packet progresses, the list of destinations associated to that particular identifier gets smaller and smaller

The information about all the destinations get to the first filtering router and stays there. Once routing decisions are made, the next routers get their chunk of addresses associate to that big message (or multicast address, if you want). If a routing table is updated at some router, the downstream router(s) are informed about the changes and update themselves. Basically, we need something to identify the full message (file), identify the first and the last segments (packets). Once everything is acknowledged.

Destinations (for established services - subscriber or registration based) can be static, some router database, and only a mask of the actual destinations can be used.

4.7.1 forcing multicast

Multicast can be adopted as a solution even when

chapter 5 Peer to peer traffic

5.1 the overall image

Peer-to-peer (abbreviated to **P2P**) refers to a computer network in which each computer in the network can act as a client or server for the other computers in the network, allowing shared access to files and peripherals without the need for a central server. P2P networks can be set up in the home, a business or over the Internet. Each network type requires all computers in the network to use the same or a compatible program to connect to each other and access files and other resources found on the other computer. P2P networks can be used for sharing content such as audio, video, data or anything in digital format.

P2P is a distributed application architecture that partitions tasks or workloads among peers. Peers are equally privileged participants in the application. Each computer in the network is referred to a node. The owner of each computer on a P2P network would set aside a portion of its resources - such as processing power, disk storage or network bandwidth - to be made directly available to other network participant, without the need for central coordination by servers or stable hosts. With this model, peers are both suppliers and consumers of resources, in contrast to the traditional client–server model where only servers supply (send), and clients consume (receive).

The first peer-to-peer application was the file sharing system **Napster**, originally released in 1999. The concept has inspired new structures and philosophies in many areas of human interaction. Peer-to-peer networking is not restricted to technology; it also covers social processes with a peer-to-peer dynamic. In such context, social peer-to-peer processes are currently emerging throughout society.

5.2 torrents

BitTorrent is a peer-to-peer file sharing protocol used for distributing large amounts of data over the Internet. BitTorrent is one of the most common protocols for transferring large files and it has been estimated that peer-to-peer networks collectively have accounted for roughly 43% to 70% of all Internet traffic (depending on geographical location) as of February 2009. Programmer Bram Cohen designed the protocol in April 2001 and released the first available version on July 2, 2001. It is now maintained by Cohen's company, BitTorrent, Inc. Currently, numerous BitTorrent clients are available for a variety of computing platforms.

Entities:

- torrent file
- tracker/indexer
- seed
- leech
- peer
- swarm

5.2.1 the torrent file

A torrent file is metadata files structured as a bencoded dictionary with the following keys:

- **announce** - the URL of the tracker
- **info** - this maps to a dictionary whose keys are dependent on whether one or more files are being shared:
- **name** - suggested file/directory name where the file(s) is/are to be saved
- **piece length** - number of bytes per piece. This is commonly 256 KiB = 262,144 B.
- **length** - size of the file in bytes (only when one file is being shared)
- **pieces** - a concatenation of all piece's SHA-1 hash. The SHA-1 hash is 160 bits long (or 40 hexadecimal digits), therefore, pieces will be a string whose length is a multiple of 40 hexadecimal digits.
- **files** - a list of dictionaries each corresponding to a file (only when multiple files are being shared). Each dictionary has the following keys:
  - **file path** - a list of strings corresponding to subdirectory names, the last of which is the actual file name
  - **file length** - size of the file in bytes.

All strings must be UTF-8 encoded.

Here is an example of a single file torrent.

```
{
    'announce': 'http://bttracker.debian.org:6969/announce','info':
    {
          'name': 'debian-503-amd64-CD-1.iso',
          'piece length': 262144,
          'length': 678301696,
          'pieces': '841ae846bc5b6d7bd6e9aa3dd9e551559c82abc1
                     ...
                     d14f1631d776008f83772ee170c42411618190a4'
    }
}
And an example for multiple files
{
     'announce': 'http://tracker.site1.com/announce',
     'info:
     {
          'name': 'directoryName',
```

```
        'piece length': 262144,
        'files':
        [
            {'path': '111.txt', 'length': 111},
            {'path': '222.txt', 'length': 222}
        ],
        'pieces': '6a8af7eda90ba9f851831073c48ea6b7b7e9feeb
                   ...
                   8a43d9d965a47f75488d3fb47d2c586337a20b9f'
    }
}
```

5.2.2 trackers and indexes

A BitTorrent **tracker** is a server that assists in the communication between peers using the BitTorrent protocol. It is also, in the absence of extensions to the original protocol, the only major critical point, as clients are required to communicate with the tracker to initiate downloads. Clients that have already begun downloading also communicate with the tracker periodically to negotiate with newer peers and provide statistics; however, after the initial reception of peer data, peer communication can continue without a tracker.

A BitTorrent **index** is a list of .torrent files, usually including descriptions and other information. Trackers merely coordinate communication between peers attempting to download the payload of the torrents.

Many BitTorrent websites act as both tracker and index. Sites such as these publicize the tracker's URL and allow users to upload torrents to the index with the tracker's URL embedded in them, providing all the features necessary to initiate a download.

There two main varieties of trackers, public and private.

- Public or open trackers can be used by anyone by adding the tracker address to an existing torrent, or they can be used by any newly created torrent.
- A private tracker is a BitTorrent tracker that restricts use, by requiring users to register with the site. The method for controlling registration used amongst many private trackers is an invitation system, in which active and contributing members are given the ability to grant a new user permission to register at the site. Private trackers are faster and safer, but membership conditions can make then unattractive, sometimes.

5.2.3 peers, seeds and leeches

A **peer** is one instance of a BitTorrent client running on a computer on the Internet to which other clients connect and transfer data. Usually a peer does not have the complete le, but only parts of it. However, in the colloquial definition, "peer" can be used to refer to any participant in the swarm (in this case, it's synonymous with "client").

A **seed** is used to refer to a peer who has 100% of the data. When a leech obtains 100% of the data, that peer by definition becomes a seed.

A **leech** refers to a peer who has a negative effect on the swarm by having a very poor share ratio (downloading much more than they upload, creating a ratio less than 1.0). Most leeches are users on asymmetric internet connections and do not leave their BitTorrent client open to seed the file after their download has completed. However, some leeches intentionally avoid uploading by using modified clients or excessively limiting their upload speed.

5.2.4 swarms

Together, all peers (including seeders) sharing a torrent are called a **swarm**. For example, six ordinary peers and two seeders make a swarm of eight. This is a holdover from the predecessor to BitTorrent, a program called Swarmcast, originally from OpenCola Software.

5.3 how do torrents work

Everything starts with a file or a set of files which are intended to be made available by somebody. This somebody is known as the **initial seed**.

A user who wants to upload a file first creates a small torrent descriptor file that they distribute by conventional means (web, email, etc.). They then make the file itself available through a BitTorrent node acting as a seed. Those with the torrent descriptor file can give it to their own BitTorrent nodes which, acting as peers or leechers, download it by connecting to the seed and/or other peers. The file being distributed is divided into segments called pieces. As each peer receives a new piece of the file it becomes a source (of that piece) for other peers, relieving the original seed from having to send that piece to every computer or user wishing a copy. With BitTorrent, the task of distributing the file is shared by those who want it; it is entirely possible for the seed to send only a single copy of the file itself and eventually distribute to an unlimited number of peers. Each piece is protected by a cryptographic hash contained in the torrent descriptor. This ensures that any modification of the piece can be reliably detected, and thus prevents both accidental and malicious modifications of any of the pieces received at other nodes. If a node starts with an authentic copy of the torrent descriptor, it can verify the authenticity of the entire file it receives.

Pieces are typically downloaded non-sequentially and are rearranged into the correct order by the BitTorrent Client, which monitors which pieces it has, can upload to other peers and which it needs. Pieces are of the same size throughout a single download (for example a 10MB file may be transmitted as ten 1MB Pieces or as forty 256kB Pieces). Due to the nature of this approach, the download of any file can be halted at any time and be resumed at a later date, without the loss of previously downloaded information, which in turn makes BitTorrent particularly useful in the transfer of larger files. This also enables the client to seek out readily available pieces and download them immediately, rather than halting the download and waiting for the next (and possibly unavailable) piece in line, which typically reduces the overall length of the download. When a peer completely downloads a file, it becomes an additional seed. This eventual shift from peers to seeders determines the overall "health" of the file (as determined by the number of times a file is available in its complete form).

5.4 what next?

The BitTorrent protocol still requires dedicated servers which act as trackers or indexes.

- In a pure peer-to-peer distributed network, every client should have some service capability, which should eliminate the necessity of dedicated servers.
- A new transport protocol should contain in its header, besides sequence numbers, an identifier for a file or structured message (a 128 bit MD5 digest of the file, let's say)
- Receiving clients express their interest in a particular file (message) identified by its file ID

5.4.1 the advent of zombies

The approach outlined in the previous section might lead to the appearance of source-less, destination-less packets within the network. The life span of such packages should be controlled in an efficient way.

bibliography

[CLSN] Collision - http://searchnetworking.techtarget.com/definition/collision

[GMLP] GraphML primer - http://graphml.graphdrawing.org/primer/graphml-primer.html

[GMLS] GraphML Specification - http://graphml.graphdrawing.org/specification.html

[IPv6] – IPv6 – The new Protocol for Internet and Intranets - http://www.ip6.com/us/book

[MCEX] Multicast Explained - http://tldp.org/HOWTO/Multicast-HOWTO-2.html

[RFC1887] An Architecture for IPv6 Unicast Address Allocation - http://tools.ietf.org/html/rfc1887

[RFC2080] RIPng for IPv6 - http://tools.ietf.org/html/rfc2080

[RFC2545] Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing - http://tools.ietf.org/html/rfc2545

[RFC2740] OSPF for IPv6 - http://tools.ietf.org/html/rfc2740

[RFC2858] Multiprotocol Extensions for BGP-4 - http://tools.ietf.org/html/rfc1887

[RFC2894] Router Renumbering for IPv6 - http://tools.ietf.org/html/rfc2894

[RFC3578] IPv6 Global Unicast Address Format - http://tools.ietf.org/html/rfc3578

[RFC4291] IP Version 6 Addressing Architecture - http://tools.ietf.org/html/rfc4291

[RFC4893] IBGP Support for Four-octet AS Number Space - http://tools.ietf.org/html/rfc4893

[RFC5058] Explicit Multicast (Xcast) Concepts and Options - http://tools.ietf.org/html/rfc5058

[RFC5396] An Architecture for IPv6 Unicast Address Allocation - http://tools.ietf.org/html/rfc5396

[TDMA] TDMA - http://en.wikipedia.org/wiki/Time_division_multiple_access

[V6P] Addresses and Routing in IPV6 - http://www.ipv6-es.com/02/docs/david_fernandez_1.pdf