An Introduction to Computer Networks (week 5)

Stanford Univ CS144 Fall 2012

PDF generated using the open source mwlib toolkit. See http://code.pediapress.com/ for more information.
PDF generated at: Tue, 06 Nov 2012 09:25:26 UTC

Contents

Articles

Routing 1

Flooding (computer networking) 1
Source routing 3
Spanning tree 4
Spanning Tree Protocol 6
Bellman–Ford algorithm 17
Dijkstra's algorithm 21
Routing Information Protocol 27
Open Shortest Path First 30
Autonomous System (Internet) 41
Border Gateway Protocol 43
Request for Comments 56

Physical and Link 61

Ethernet 61
Ethernet over twisted pair 68
Carrier sense multiple access with collision detection 73

Wireless 75

Wi-Fi 75
IEEE 802.11 87

Ebooks Dedicated to Richard Beckett 98

References

Article Sources and Contributors 99
Image Sources, Licenses and Contributors 103

Article Licenses

License 105

Routing

Flooding (computer networking)

Flooding is a simple routing algorithm in which every incoming packet is sent through every outgoing link.

Flooding is used in bridging and in systems such as Usenet and peer-to-peer file sharing and as part of some routing protocols, including OSPF, DVMRP, and those used in ad-hoc wireless networks.

flooding algorithm

Algorithm

There are several variants of flooding algorithm. Most work roughly as follows:

1. Each node acts as both a transmitter and a receiver.
2. Each node tries to forward every message to every one of its neighbors except the source node.

This results in every message eventually being delivered to all reachable parts of the network.

Algorithms may need to be more complex than this, since, in some case, precautions have to be taken to avoid wasted duplicate deliveries and infinite loops, and to allow messages to eventually expire from the system. A variant of flooding called *selective flooding* partially addresses these issues by only sending packets to routers in the same direction. In selective flooding the routers don't send every incoming packet on every line but only on those lines which are going approximately in the right direction.

flooding algorithm with ack messages

Advantages

- If a packet can be delivered, it will (probably multiple times).
- Since flooding naturally utilizes every path through the network, it will also use the shortest path.
- This algorithm is very simple to implement.

Problems

- Flooding can be costly in terms of wasted bandwidth. While a message may only have one destination it has to be sent to every host. In the case of a ping flood or a denial of service attack, it can be harmful to the reliability of a computer network.
- Messages can become duplicated in the network further increasing the load on the networks bandwidth as well as requiring an increase in processing complexity to disregard duplicate messages.
- Duplicate packets may circulate forever, unless certain precautions are taken:
  - Use a hop count or a time to live count and include it with each packet. This value should take into account the number of nodes that a packet may have to pass through on the way to its destination.
  - Have each node keep track of every packet seen and only forward each packet once
  - Enforce a network topology without loops

Examples

Open Shortest Path First - Used for transferring updates to the topology (LSAs)

References

- Computer Networks, 5th Edition. A. Tanenbaum, D. Wetherall. pages 368-370. ISBN 978-0-13-212695-3.

External links

- Ricochet [1] A flooding networking app written in Java

References

[1] http://ricochet.wiki.sourceforge.net/

Source routing

In computer networking, **source routing** allows a sender of a packet to partially or completely specify the route the packet takes through the network. In contrast, in non-source routing protocols, routers in the network determine the path based on the packet's destination.

Source routing allows easier troubleshooting, improved traceroute, and enables a node to discover all the possible routes to a host. It also allows a source to directly manage network performance by forcing packets to travel over one path to prevent congestion on another.

In the Internet Protocol, two header options are available which are rarely used: "strict source and record route" (SSRR) and "loose source and record route" (LSRR). Because of security concerns, packets marked LSRR are frequently blocked on the Internet. If not blocked, LSRR can allow an attacker to spoof its address but still successfully receive response packets.[1]

Policy-based routing can also be used to route packets using their source addresses.

References

[1] Rik Farrow. "Source Address Spoofing" (http://www.microsoft.com/technet/archive/community/columns/security/netdef/sourcead.mspx). *TechNet*. Microsoft Corporation. .

External links

- "Source Routing" (http://www.iss.net/security_center/advice/Underground/Hacking/Methods/Technical/Source_Routing/default.htm). IBM Internet Security Systems.
- RFC 791 (http://www.ietf.org/rfc/rfc0791.txt)
- Source Routing Not Considered Harmful (http://www.gweep.net/~crimson/network/lsrr.html)

Spanning tree

In the mathematical field of graph theory, a **spanning tree** T of a connected, undirected graph G is a tree composed of all the vertices and some (or perhaps all) of the edges of G. Informally, a spanning tree of G is a selection of edges of G that form a tree *spanning* every vertex. That is, every vertex lies in the tree, but no cycles (or loops) are formed. On the other hand, every bridge of G must belong to T.

A spanning tree (blue heavy edges) of a grid graph

A spanning tree of a connected graph G can also be defined as a maximal set of edges of G that contains no cycle, or as a minimal set of edges that connect all vertices.

In certain fields of graph theory it is often useful to find a minimum spanning tree of a weighted graph. Other optimization problems on spanning trees have also been studied, including the maximum spanning tree, the minimum tree that spans at least k vertices, the minimum spanning tree with at most k edges per vertex (Degree-Constrained Spanning Tree), the spanning tree with the largest number of leaves (closely related to the smallest connected dominating set), the spanning tree with the fewest leaves (closely related to the Hamiltonian path problem), the minimum diameter spanning tree, and the minimum dilation spanning tree.

Fundamental cycles

Adding just one edge to a spanning tree will create a cycle; such a cycle is called a **fundamental cycle**. There is a distinct fundamental cycle for each edge; thus, there is a one-to-one correspondence between fundamental cycles and edges not in the spanning tree. For a connected graph with V vertices, any spanning tree will have V-1 edges, and thus, a graph of E edges will have E-V+1 fundamental cycles. For any given spanning tree these cycles form a basis for the cycle space.

Dual to the notion of a fundamental cycle is the notion of a **fundamental cutset**. By deleting just one edge of the spanning tree, the vertices are partitioned into two disjoint sets. The fundamental cutset is defined as the set of edges that must be removed from the graph G to accomplish the same partition. Thus, there are precisely V-1 fundamental cutsets for the graph, one for each edge of the spanning tree.

The duality between fundamental cutsets and fundamental cycles is established by noting that cycle edges not in the spanning tree can only appear in the cutsets of the other edges in the cycle; and *vice versa*: edges in a cutset can only appear in those cycles containing the edge corresponding to the cutset.

Spanning forests

A **spanning forest** is a type of subgraph that generalises the concept of a spanning tree. However, there are two definitions in common use. One is that a spanning forest is a subgraph that consists of a spanning tree in each connected component of a graph. (Equivalently, it is a maximal cycle-free subgraph.) This definition is common in computer science and optimization. It is also the definition used when discussing minimum spanning forests, the generalization to disconnected graphs of minimum spanning trees. Another definition, common in graph theory, is that a spanning forest is any subgraph that is both a forest (contains no cycles) and spanning (includes every vertex).

Counting spanning trees

The number *t(G)* of spanning trees of a connected graph is a well-studied invariant. In some cases, it is easy to calculate *t(G)* directly. For example, if *G* is itself a tree, then *t(G)=1*, while if *G* is the cycle graph C_n with *n* vertices, then *t(G)=n*. For any graph *G*, the number *t(G)* can be calculated using Kirchhoff's matrix-tree theorem (follow the link for an explicit example using the theorem).

Cayley's formula counts the number of spanning trees on a complete graph. There are $2^{2-2} = 1$ trees in K_2, $3^{3-2} = 3$ trees in K_3, and $4^{4-2} = 16$ trees in K_4.

Cayley's formula is a formula for the number of spanning trees in the complete graph K_n with *n* vertices. The formula states that $t(K_n) = n^{n-2}$. Another way of stating Cayley's formula is that there are exactly n^{n-2} labelled trees with *n* vertices. Cayley's formula can be proved using Kirchhoff's matrix-tree theorem or via the Prüfer code.

If *G* is the complete bipartite graph $K_{p,q}$, then $t(G) = p^{q-1}q^{p-1}$, while if *G* is the *n*-dimensional hypercube graph Q_n, then $t(G) = 2^{2^n-n-1} \prod_{k=2}^{n} k^{\binom{n}{k}}$. These formulae are also consequences of the matrix-tree theorem.

If *G* is a multigraph and *e* is an edge of *G*, then the number *t(G)* of spanning trees of *G* satisfies the *deletion-contraction recurrence t(G)=t(G-e)+t(G/e)*, where *G-e* is the multigraph obtained by deleting *e* and *G/e* is the contraction of *G* by *e*, where multiple edges arising from this contraction are not deleted.

Uniform spanning trees

A spanning tree chosen randomly from among all the spanning trees with equal probability is called a uniform spanning tree (UST). This model has been extensively researched in probability and mathematical physics.

Algorithms

The classic spanning tree algorithm, depth-first search (DFS), is due to Robert Tarjan. Another important algorithm is based on breadth-first search (BFS).

Parallel algorithms typically take different approaches than BFS or DFS. Halperin and Zwick designed an optimal randomized parallel algorithm that runs in O(log n) time with high probability on EREW PRAM.[1] The Shiloach-Vishkin algorithm, due to Yossi Shiloach and Uzi Vishkin, is the basis for many parallel implementations.[2] Bader and Cong's algorithm is shown to run fast in practice on a variety of graphs.[3]

The most common distributed algorithm is the Spanning Tree Protocol, used by OSI link layer devices to create a spanning tree using the existing links as the source graph in order to avoid broadcast storms.

References

- Eppstein, David (1999). "Spanning trees and spanners" [4]. *Handbook of Computational Geometry*. Elsevier. pp. 425–461.
- Garey, Michael R.; Johnson, David S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman. ISBN 0-7167-1045-5. A2.1: ND2, pg.206.
- Wu, Bang Ye; Chao, Kun-Mao (2004). *Spanning Trees and Optimization Problems*. CRC Press. ISBN 1-58488-436-3.
- Gabow, Harold N.; Myers, Eugene W. (1978). "Finding All Spanning Trees of Directed and Undirected Graphs" [5]. *SIAM J. Comput.* **7** (280).

References

[1] http://citeseer.ist.psu.edu/652374.html
[2] http://citeseer.ist.psu.edu/context/74288/0
[3] http://portal.acm.org/citation.cfm?id=1196220
[4] http://www.ics.uci.edu/~eppstein/pubs/Epp-TR-96-16.pdf
[5] http://link.aip.org/link/?SMJCAT/7/280/1

Spanning Tree Protocol

The **Spanning Tree Protocol** (**STP**) is a network protocol that ensures a loop-free topology for any bridged Ethernet local area network. The basic function of STP is to prevent bridge loops and the broadcast radiation that results from them. Spanning tree also allows a network design to include spare (redundant) links to provide automatic backup paths if an active link fails, without the danger of bridge loops, or the need for manual enabling/disabling of these backup links.

Spanning Tree Protocol (STP) is standardized as IEEE 802.1D. As the name suggests, it creates a spanning tree within a mesh network of connected layer-2 bridges (typically Ethernet switches), and disables those links that are not part of the spanning tree, leaving a single active path between any two network nodes.

STP is based on an algorithm invented by Radia Perlman while working for Digital Equipment Corporation.[1][2]

Protocol operation

The collection of bridges in a local area network (LAN) can be depicted as a graph whose nodes are bridges and LAN segments (or cables), and whose edges are the interfaces connecting the bridges to the segments. To break loops in the LAN while maintaining access to all LAN segments, the bridges collectively compute a spanning tree. The spanning tree is not necessarily a minimum cost spanning tree. A network administrator can reduce the cost of a spanning tree, if necessary, by altering some of the configuration parameters in such a way as to affect the choice of the root of the spanning tree. The spanning tree that the bridges compute using the Spanning Tree Protocol can be determined using the following rules. The example network at the right, below, will be used to illustrate the rules.

1. An example network. The numbered boxes represent bridges (the number represents the bridge ID). The lettered clouds represent network segments.

2. The smallest bridge ID is 3. Therefore, bridge 3 is the root bridge.

3. Assuming that the cost of traversing any network segment is 1, the least cost path from bridge 4 to the root bridge goes through network segment c. Therefore, the root port for bridge 4 is the one on network segment c.

4. The least cost path to the root from network segment e goes through bridge 92. Therefore the designated port for network segment e is the port that connects bridge 92 to network segment e.

5. This diagram illustrates all port states as computed by the spanning tree algorithm. Any active port that is not a root port or a designated port is a blocked port.

6. After link failure the spanning tree algorithm computes and spans new least-cost tree.

Select a root bridge. The *root bridge* of the spanning tree is the bridge with the smallest (lowest) bridge ID. Each bridge has a configurable priority number and a MAC Address; the bridge ID contains both numbers combined together - Bridge priority + MAC (32768.0200.0000.1111). The Bridge priority default is 32768 and can only be

configured in multiples of 4096. To compare two bridge IDs, the priority is compared first, as if looking at a real number anything less than 32768...will become the target of being the root. If two bridges have equal priority then the MAC addresses are compared; for example, if switches A (MAC=0200.0000.1111) and B (MAC=0200.0000.2222) both have a priority of 32768 then switch A will be selected as the root bridge. If the network administrators would like switch B to become the root bridge, they must set its priority to be less than 32768 or configure the spanning tree a root primary/secondary. When configuring the root primary and root secondary the switch will automatically change the priority accordingly, 24577 and 28673 respectively with the default configuration.

Determine the least cost paths to the root bridge. The computed spanning tree has the property that messages from any connected device to the root bridge traverse a least cost path, i.e., a path from the device to the root that has minimum cost among all paths from the device to the root. The cost of traversing a path is the sum of the costs of the segments on the path. Different technologies have different default costs for network segments. An administrator can configure the cost of traversing a particular network segment. The property that messages always traverse least-cost paths to the root is guaranteed by the following two rules.

Least cost path from each bridge. After the root bridge has been chosen, each bridge determines the cost of each possible path from itself to the root. From these, it picks one with the smallest cost (a least-cost path). The port connecting to that path becomes the *root port* (RP) of the bridge.

Least cost path from each network segment. The bridges on a network segment collectively determine which bridge has the least-cost path from the network segment to the root. The port connecting this bridge to the network segment is then the *designated port* (DP) for the segment.

Disable all other root paths. Any active port that is not a root port or a designated port is a *blocked port* (BP).

Modifications in case of ties. The above rules over-simplify the situation slightly, because it is possible that there are ties, for example, two or more ports on a single bridge are attached to least-cost paths to the root or two or more bridges on the same network segment have equal least-cost paths to the root. To break such ties:

Breaking ties for root ports. When multiple paths from a bridge are least-cost paths, the chosen path uses the neighbor bridge with the lower bridge ID. The root port is thus the one connecting to the bridge with the lowest bridge ID. For example, in figure 3, if switch 4 was connected to network segment d instead of segment c, there would be two paths of length 2 to the root, one path going through bridge 24 and the other through bridge 92. Because there are two least cost paths, the lower bridge ID (24) would be used as the tie-breaker in choosing which path to use.

Breaking ties for designated ports. When more than one bridge on a segment leads to a least-cost path to the root, the bridge with the lower bridge ID is used to forward messages to the root. The port attaching that bridge to the network segment is the *designated port* for the segment. In figure 4, there are two least cost paths from network segment d to the root, one going through bridge 24 and the other through bridge 92. The lower bridge ID is 24, so the tie breaker dictates that the designated port is the port through which network segment d is connected to bridge 24. If bridge IDs were equal, then the bridge with the lowest MAC address would have the designated port. In either case, the loser sets the port as being blocked.

The final tie-breaker. In some cases, there may still be a tie, as when two bridges are connected by multiple cables. In this case, multiple ports on a single bridge are candidates for root port. In this case, the path which passes through the port on the neighbor bridge that has the lowest port identifier [Port priority(default=128) + Port number] is used.

In summary, the sequence of events to determine the best received BPDU (which is your best path to the root) is

1. lowest root bridge id
2. lowest root path cost
3. lowest sender bridge id
4. lowest sender port id

Data rate and STP path cost

Data rate	STP Cost (802.1D-1998)	RSTP Cost (802.1D-2004 / 802.1w)[3]
4 Mbit/s	250	5,000,000
10 Mbit/s	100	2,000,000
16 Mbit/s	62	1,250,000
100 Mbit/s	19	200,000
1 Gbit/s	4	20,000
2 Gbit/s	3	10,000
10 Gbit/s	2	2,000

Bridge Protocol Data Units

The above rules describe one way of determining what spanning tree will be computed by the algorithm, but the rules as written require knowledge of the entire network. The bridges have to determine the root bridge and compute the port roles (root, designated, or blocked) with only the information that they have. To ensure that each bridge has enough information, the bridges use special data frames called **Bridge Protocol Data Units** (BPDUs) to exchange information about bridge IDs and root path costs.

A bridge sends a BPDU frame using the unique MAC address of the port itself as a source address, and a destination address of the STP multicast address 01:80:C2:00:00:00.

There are three types of BPDUs:

- Configuration BPDU (CBPDU), used for Spanning Tree computation
- Topology Change Notification (TCN) BPDU, used to announce changes in the network topology
- Topology Change Notification Acknowledgment (TCA)

BPDUs are exchanged regularly (every 2 seconds by default) and enable switches to keep track of network changes and to start and stop forwarding at ports as required.

When a device is first attached to a switch port, it will not immediately start to forward data. It will instead go through a number of states while it processes BPDUs and determines the topology of the network. When a host is attached such as a computer, printer or server the port will always go into the forwarding state, albeit after a delay of about 30 seconds while it goes through the listening and learning states (see below). The time spent in the listening and learning states is determined by a value known as the forward delay (default 15 seconds and set by the root bridge). However, if instead another *switch* is connected, the port may remain in blocking mode if it is determined that it would cause a loop in the network. Topology Change Notification (TCN) BPDUs are used to inform other switches of port changes. TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

STP switch port states:

- **Blocking** - A port that would cause a switching loop, no user data is sent or received but it may go into forwarding mode if the other links in use were to fail and the spanning tree algorithm determines the port may transition to the forwarding state. BPDU data is still received in blocking state. Prevents the use of looped paths.
- **Listening** - The switch processes BPDUs and awaits possible new information that would cause it to return to the blocking state. It does not populate the MAC address table and it does not forward frames.
- **Learning** - While the port does not yet forward frames it does learn source addresses from frames received and adds them to the filtering database (switching database). It populates the MAC Address table, but does not

forward frames.

- **Forwarding** - A port receiving and sending data, normal operation. STP still monitors incoming BPDUs that would indicate it should return to the blocking state to prevent a loop.
- **Disabled** - Not strictly part of STP, a network administrator can manually disable a port

To prevent the delay when connecting hosts to a switch and during some topology changes, Rapid STP was developed and standardized by IEEE 802.1w, which allows a switch port to rapidly transition into the forwarding state during these situations.

Bridge Protocol Data Unit fields

The bridge ID, or BID, is a field inside a BPDU packet. It is eight bytes in length. The first two bytes are the Bridge Priority, an unsigned integer of 0-65,535. The last six bytes are a MAC address supplied by the switch. In the event that MAC Address Reduction is used, the first two bytes are used differently. The first four bits are a configurable priority, and the last twelve bits carry either the VLAN ID or MSTP instance number.

Evolutions and extensions

The first spanning tree protocol was invented in 1985 at the Digital Equipment Corporation by Radia Perlman.[1] In 1990, the IEEE published the first standard for the protocol as 802.1D,[4] based on the algorithm designed by Perlman. Subsequent versions were published in 1998[5] and 2004,[6] incorporating various extensions.

Although the purpose of a standard is to promote interworking of equipment from different vendors, different implementations of a standard are not guaranteed to work, due for example to differences in default timer settings. The IEEE encourages vendors to provide a "Protocol Implementation Conformance Statement", declaring which capabilities and options have been implemented,[6] to help users determine whether different implementations will interwork correctly.

Also, the original Perlman-inspired Spanning Tree Protocol, called DEC STP, is not a standard and differs from the IEEE version in message format as well as timer settings. Some bridges implement both the IEEE and the DEC versions of the Spanning Tree Protocol, but their interworking can create issues for the network administrator, as illustrated by the problem discussed in an on-line Cisco document.[7]

Rapid Spanning Tree Protocol

In 2001, the IEEE introduced Rapid Spanning Tree Protocol (RSTP) as 802.1w. RSTP provides significantly faster spanning tree convergence after a topology change, introducing new convergence behaviors and bridge port roles to do this. RSTP was designed to be backwards-compatible with standard STP.

While STP can take 30 to 50 seconds to respond to a topology change, RSTP is typically able to respond to changes within $3 \times$ Hello times (default: 3 times 2 seconds) or within a few milliseconds of a physical link failure. The so-called Hello time is an important and configurable time interval that is used by RSTP for several purposes; its default value is 2 seconds.[8][9]

Standard IEEE 802.1D-2004 incorporates RSTP and obsoletes the original STP standard.[10]

Rapid Spanning Tree Operation

RSTP adds new bridge port roles in order to speed convergence following a link failure.

RSTP bridge port roles:

- **Root** - A forwarding port that is the best port from Nonroot-bridge to Rootbridge
- **Designated** - A forwarding port for every LAN segment
- **Alternate** - An alternate path to the root bridge. This path is different than using the root port.
- **Backup** - A backup/redundant path to a segment where another bridge port already connects.
- **Disabled** - Not strictly part of STP, a network administrator can manually disable a port

Additional RSTP Operation Details:

- Detection of root switch failure is done in 3 hello times, which is 6 seconds if default hello times have not been changed.
- Ports may be configured as edge ports if they are attached to a LAN that has no other bridges attached. These edge ports transition directly to the forwarding state. RSTP still continues to monitor the port for BPDUs in case a bridge is connected. RSTP can also be configured to automatically detect edge ports. As soon as the bridge detects a BPDU coming to an edge port, the port becomes a non-edge port.
- Unlike in STP, RSTP will respond to BPDUs sent from the direction of the root bridge. An RSTP bridge will "propose" its spanning tree information to its designated ports. If another RSTP bridge receives this information and determines this is the superior root information, it sets all its other ports to discarding. The bridge may send an "agreement" to the first bridge confirming its superior spanning tree information. The first bridge, upon receiving this agreement, knows it can rapidly transition that port to the forwarding state bypassing the traditional listening/learning state transition. This essentially creates a cascading effect away from the root bridge where each designated bridge proposes to its neighbors to determine if it can make a rapid transition. This is one of the major elements that allows RSTP to achieve faster convergence times than STP.
- As discussed in the port role details above, RSTP maintains backup details regarding the discarding status of ports. This avoids timeouts if the current forwarding ports were to fail or BPDUs were not received on the root port in a certain interval.
- RSTP will revert to legacy STP on an interface if a legacy version if an STP BPDU is detected on that port.

Per-VLAN Spanning Tree and Per-VLAN Spanning Tree Plus

In Ethernet switched environments where multiple Virtual LANs exist, it is often necessary to create multiple spanning trees so that traffic from different VLANs uses different links. Cisco's proprietary versions of Spanning Tree Protocol that allows for the creation of a spanning tree for each VLAN are called *Per-VLAN Spanning Tree* (PVST) and *Per-VLAN Spanning Tree Plus* (PVST+). Both PVST and PVST+ protocols are Cisco proprietary protocols and they cannot be used on most third party switches. Some equipment from Force10 Networks, Extreme Networks, Avaya and Blade Network Technologies support PVST+.[11][12][13] Extreme Networks does so with two limitations (lack of support on ports where the VLAN is untagged/native and also on the VLAN with ID 1). PVST works only with ISL (Cisco's proprietary protocol for VLAN encapsulation) due to its embedded Spanning tree ID. This is the default protocol on Cisco switches that support ISL. Due to high penetration of the IEEE 802.1Q VLAN trunking standard and PVST's dependence on ISL, Cisco defined a different PVST+ standard that is compatible with 802.1Q encapsulation. This became the default protocol for Cisco switches when Cisco discontinued and removed ISL support from its switches. PVST+ can tunnel across an MSTP Region.[14]

Rapid Per-VLAN Spanning Tree

This is Cisco's proprietary version of Rapid Spanning Tree Protocol. It creates a spanning tree for each VLAN, just like PVST. Cisco refers to this as Rapid Per-VLAN Spanning Tree (RPVST).

VLAN Spanning Tree Protocol

In Juniper Networks environment, if compatibility to Cisco's proprietary PVST protocol is required, VLAN Spanning Tree Protocol (VSTP) can be configured. VSTP maintains a separate spanning-tree instance for each VLAN configured in the switch. The VSTP protocol is only supported by the EX and MX Series from Juniper Networks. There are two restrictions to the compatibility of VSTP:

1. VSTP supports only 253 different spanning-tree topologies. If there are more than 253 VLANs, it is recommended to configure RSTP in addition to VSTP, and VLANs beyond 253 will be handled by RSTP.
2. MVRP does not support VSTP. If this protocol is in use, VLAN membership for trunk interfaces must be statically configured[15].

By default, VSTP uses the RSTP protocol as its core spanning-tree protocol, but usage of STP can be forced if the network includes old bridges[16].

For more information about configuring VSTP on Juniper Networks switches, see the official documentation Understanding VSTP [16].

Multiple Spanning Tree Protocol

The *Multiple Spanning Tree Protocol* (MSTP), originally defined in IEEE 802.1s and later merged into IEEE 802.1Q-2005, defines an extension to RSTP to further develop the usefulness of virtual LANs (VLANs). This "Per-VLAN" Multiple Spanning Tree Protocol configures a separate Spanning Tree for each VLAN group and blocks all but one of the possible alternate paths within each Spanning Tree.

If there is only one Virtual LAN (VLAN) in the network, single (traditional) STP works appropriately. If the network contains more than one VLAN, the logical network configured by single STP would work, but it is possible to make better use of the alternate paths available by using an alternate spanning tree for different VLANs or groups of VLANs.

MSTP allows formation of MST regions that can run multiple MST instances (MSTI). Multiple regions and other STP bridges are interconnected using one single common spanning tree (CST).

MSTP is similar to Cisco Systems' Multiple Instances Spanning Tree Protocol (MISTP), and is an evolution of the Spanning Tree Protocol and the Rapid Spanning Tree Protocol. It was introduced in IEEE 802.1s as an amendment to 802.1Q, 1998 edition. Standard IEEE 802.1Q-2005 now includes MSTP.

Unlike some proprietary per-VLAN spanning tree implementations,[17] MSTP includes all of its spanning tree information in a single BPDU format. Not only does this reduce the number of BPDUs required on a LAN to communicate spanning tree information for each VLAN, but it also ensures backward compatibility with RSTP (and in effect, classic STP too). MSTP does this by encoding additional region information after the standard RSTP BPDU as well as a number of MSTI messages (from 0 to 64 instances, although in practice many bridges support fewer). Each of these MSTI configuration messages conveys the spanning tree information for each instance. Each instance can be assigned a number of configured VLANs and frames (packets) assigned to these VLANs operate in this spanning tree instance whenever they are inside the MST region. In order to avoid conveying their entire VLAN to spanning tree mapping in each BPDU, bridges encode an MD5 digest of their VLAN to instance table in the MSTP BPDU. This digest is then used by other MSTP bridges, along with other administratively configured values, to determine if the neighboring bridge is in the same MST region as itself.

MSTP is fully compatible with RSTP bridges, in that an MSTP BPDU can be interpreted by an RSTP bridge as an RSTP BPDU. This not only allows compatibility with RSTP bridges without configuration changes, but also causes

any RSTP bridges outside of an MSTP region to see the region as a single RSTP bridge, regardless of the number of MSTP bridges inside the region itself. In order to further facilitate this view of an MST region as a single RSTP bridge, the MSTP protocol uses a variable known as remaining hops as a time to live counter instead of the message age timer used by RSTP. The message age time is only incremented once when spanning tree information enters an MST region, and therefore RSTP bridges will see a region as only one "hop" in the spanning tree. Ports at the edge of an MST region connected to either an RSTP or STP bridge or an endpoint are known as boundary ports. As in RSTP, these ports can be configured as edge ports to facilitate rapid changes to the forwarding state when connected to endpoints.

Shortest Path Bridging

The IEEE approved the IEEE 802.1aq standard May 2012,[18] also known and documented in most books as Shortest Path Bridging (SPB). SPB allows all links to be active through multiple equal cost paths, and provides much larger layer 2 topologies, faster convergence, and improves the use of the mesh topologies through increase bandwidth between all devices by allowing traffic to load share across all paths on a mesh network.[19][20] SPB consolidates multiple existing functionalities, including Spanning Tree Protocol (STP), Multiple Spanning Tree Protocol (MSTP), Rapid Spanning Tree Protocol (RSTP), and Multiple MAC Registration Protocol (MMRP) into a one link state protocol.[21] SPB is designed to virtually eliminate human error during configuration and preserves the plug-and-play nature that established Ethernet as the de facto protocol at Layer 2.[21]

References

[1] Perlman, Radia (1985). "An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN". *ACM SIGCOMM Computer Communication Review* **15** (4): 44–53. doi:10.1145/318951.319004.

[2] Perlman, Radia (2000). *Interconnections, Second Edition*. USA: Addison-Wesley. ISBN 0-201-63448-1.

[3] "802.1D IEEE Standard for Local and Metropolitan Area Networks. Media Access Control (MAC) Bridges" (http://standards.ieee.org/getieee802/download/802.1D-2004.pdf). IEEE. 2004. p. 154. . Retrieved 19 April 2012.

[4] LAN/MAN Standards Committee of the IEEE Computer Society, ed. (1990). *ANSI/IEEE Std 802.1D*. IEEE

[5] LAN/MAN Standards Committee of the IEEE Computer Society, ed. (1998). *ANSI/IEEE Std 802.1D, 1998 Edition, Part 3: Media Access Control (MAC) Bridges*. IEEE

[6] LAN/MAN Standards Committee of the IEEE Computer Society, ed. (2004). *ANSI/IEEE Std 802.1D - 2004: IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges*. IEEE

[7] (PDF) *Understanding Issues Related to Inter-VLAN Bridging* (http://www.cisco.com/warp/public/473/inter-vlan_11072.pdf). Cisco Systems, Inc.. 11072.

[8] Waldemar Wojdak (March 2003 [CPCI203]). "Rapid Spanning Tree Protocol: A new solution from an old technology" (http://www.compactpci-systems.com/articles/id/?203). . Retrieved 2008-08-04.

[9] "Understanding Rapid Spanning Tree Protocol (802.1w)" (http://www.cisco.com/en/US/tech/tk389/tk621/technologies_white_paper09186a0080094cfa.shtml). . Retrieved 2008-11-27.

[10] *IEEE 802.1D-2004*, IEEE, 2004-06-04, "Since the original Spanning Tree Protocol (STP) has been removed from the 2004 revision of IEEE Std 802.1D, an implementation of RSTP is required for any claim of conformance for an implementation of IEEE Std 802.1Q-2003 that refers to the current revision of IEEE Std 802.1D"

[11] "Technical Documentation" (https://www.force10networks.com/CSPortal20/TechTips/0050B_HowDoIConfigureSpanningTree.aspx). Force10Networks. . Retrieved 2011-01-25.

[12] "ExtremeXOS Operating System, Version 12.5" (http://www.extremenetworks.com/libraries/products/DSExtXOS_1030.pdf) (PDF). Extreme Networks. 2010. . Retrieved 2011-01-25.

[13] "BLADE PVST+ Interoperability with Cisco" (http://www.bladenetwork.net/userfiles/file/PDFs/WP_PVST_SpanningTree_Cisco.pdf) (PDF). 2006. . Retrieved 2011-01-25.

[14] "Bridging Between IEEE 802.1Q VLANs" (http://www.cisco.com/en/US/docs/ios/12_1t/12_1t3/feature/guide/dtbridge.html#wp1020686). Cisco Systems. . Retrieved 2011-01-25.

[15] http://www.juniper.net/techpubs/en_US/junos10.0/topics/concept/bridging-mvrp-ex-series.html

[16] https://www.juniper.net/techpubs/en_US/junos9.4/topics/concept/spanning-trees-ex-series-vstp-understanding.html

[17] "CiscoWorks LAN Management Solution 3.2 Deployment Guide" (https://www.cisco.com/en/US/prod/collateral/netmgtsw/ps6504/ps6528/ps2425/white_paper_c07-552114.html#wp9003215). August 2009. . Retrieved 2010-01-25.

[18] Shuang Yu (8 May 2012). "IEEE APPROVES NEW IEEE 802.1aq™ SHORTEST PATH BRIDGING STANDARD" (http://standards.ieee.org/news/2012/802.1aq.html). IEEE. . Retrieved 2 June 2012.

[19] Peter Ashwood-Smith (24 Feb 2011). "Shortest Path Bridging IEEE 802.1aq Overview" (http://meetings.apnic.net/__data/assets/pdf_file/0012/32007/APRICOT_SPB_Overview.pdf). Huawei. . Retrieved 11 May 2012.

[20] Jim Duffy (11 May 2012). "Largest Illinois healthcare system uproots Cisco to build $40M private cloud" (http://www.pcadvisor.co.uk/news/internet/3357242/largest-illinois-healthcare-system-uproots-cisco-build-40m-private-cloud/). PC Advisor. . Retrieved 11 May 2012. "Shortest Path Bridging will replace Spanning Tree in the Ethernet fabric."

[21] "IEEE Approves New IEEE 802.1aq Shortest Path Bridging Standard" (http://www.techpowerup.com/165594/IEEE-Approves-New-IEEE-802.1aq-Shortest-Path-Bridging-Standard.html). Tech Power Up. 7 May 2012. . Retrieved 11 May 2012.

External links

- Cisco home page for the Spanning-Tree protocol family (http://www.cisco.com/en/US/tech/tk389/tk621/tsd_technology_support_protocol_home.html) (discusses CST, MISTP, PVST, PVST+, RSTP, STP)
- STP article in the Wireshark wiki (http://wiki.wireshark.org/STP) Includes a sample PCAP-file of captured STP traffic.
- Perlman, Radia. "Algorhyme" (http://web.archive.org/web/20110719212324/http://www.csua.berkeley.edu/~ranga/humor/algorhyme.txt). University of California at Berkeley. Archived from the original (http://www.csua.berkeley.edu/~ranga/humor/algorhyme.txt) on 2011-07-19. Retrieved 2011-09-01.
- IEEE Standards
  - ANSI/IEEE 802.1D-2004 standard (http://standards.ieee.org/getieee802/download/802.1D-2004.pdf), section 17 discusses RSTP (Regular STP is no longer a part of this standard. This is pointed out in section 8.)
  - ANSI/IEEE 802.1Q-2005 standard (http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf), section 13 discusses MSTP
- RFCs
  - RFC 2674-1999, proposed standard, Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions
  - RFC 1525-1993, - SBRIDGEMIB, proposed standard, Definitions of Managed Objects for Source Routing Bridges
  - RFC 1493-1993 - BRIDGEMIB, draft standard, Definitions of Managed Objects for Bridges
- Spanning Tree Direct vs Indirect Link Failures - CCIE Study (http://blog.ipexpert.com/2010/03/22/spanning-tree-direct-vs-indirect-link-failures/)

Bellman–Ford algorithm

Bellman–Ford algorithm

Class	Single-source shortest path problem (for weighted directed graphs)
Data structure	Graph
Worst case performance	$O(\|V\|\|E\|)$
Worst case space complexity	$O(\|V\|)$

The **Bellman–Ford algorithm** computes single-source shortest paths in a weighted digraph.[1] For graphs with only non-negative edge weights, the faster Dijkstra's algorithm also solves the problem. Thus, Bellman–Ford is used primarily for graphs with negative edge weights. The algorithm is named after its developers, Richard Bellman and Lester Ford, Jr.

Negative edge weights are found in various applications of graphs, hence the usefulness of this algorithm.[2] However, if a graph contains a "negative cycle", i.e., a cycle whose edges sum to a negative value, then walks of arbitrarily low weight can be constructed by repeatedly following the cycle, so there may not be a *shortest* path. In such a case, the Bellman-Ford algorithm can detect negative cycles and report their existence, but it cannot produce a correct "shortest path" answer if a negative cycle is reachable from the source.[3][1]

Algorithm

Bellman–Ford is based on dynamic programming approach. In its basic structure it is similar to Dijkstra's Algorithm, but instead of greedily selecting the minimum-weight node not yet processed to relax, it simply relaxes *all* the edges, and does this $|V| - 1$ times, where $|V|$ is the number of vertices in the graph. The repetitions allow minimum distances to propagate accurately throughout the graph, since, in the absence of negative cycles, the shortest path can visit each node at most only once. Unlike the greedy approach, which depends on certain structural assumptions derived from positive weights, this straightforward approach extends to the general case.

Bellman–Ford runs in $O(|V|\cdot|E|)$ time, where $|V|$ and $|E|$ are the number of vertices and edges respectively.

```
procedure BellmanFord(list vertices, list edges, vertex source)
   // This implementation takes in a graph, represented as lists of vertices
   // and edges, and modifies the vertices so that their distance and
   // predecessor attributes store the shortest paths.

   // Step 1: initialize graph
   for each vertex v in vertices:
       if v is source tn v.distance := 0
       else v.distance := infinity
       v.predecessor := null

   // Step 2: relax edges repeatedly
   for i from 1 to size(vertices)-1:
       for each edge uv in edges: // uv is the edge from u to v
           u := uv.source
           v := uv.destination
           if u.distance + uv.weight < v.distance:
               v.distance := u.distance + uv.weight
```

```
            v.predecessor := u

// Step 3: check for negative-weight cycles
for each edge uv in edges:
    u := uv.source
    v := uv.destination
    if u.distance + uv.weight < v.distance:
        error "Graph contains a negative-weight cycle"
```

Proof of correctness

The correctness of the algorithm can be shown by induction. The precise statement shown by induction is:

Lemma. After *i* repetitions of *for* cycle:

- If Distance(*u*) is not infinity, it is equal to the length of some path from *s* to *u*;
- If there is a path from *s* to *u* with at most *i* edges, then Distance(u) is at most the length of the shortest path from *s* to *u* with at most *i* edges.

Proof. For the base case of induction, consider `i=0` and the moment before *for* cycle is executed for the first time. Then, for the source vertex, `source.distance = 0`, which is correct. For other vertices *u*, `u.distance =` **`infinity`**, which is also correct because there is no path from *source* to *u* with 0 edges.

For the inductive case, we first prove the first part. Consider a moment when a vertex's distance is updated by `v.distance := u.distance + uv.weight`. By inductive assumption, `u.distance` is the length of some path from *source* to *u*. Then `u.distance + uv.weight` is the length of the path from *source* to *v* that follows the path from *source* to *u* and then goes to *v*.

For the second part, consider the shortest path from *source* to *u* with at most *i* edges. Let *v* be the last vertex before *u* on this path. Then, the part of the path from *source* to *v* is the shortest path from *source* to *v* with at most *i-1* edges. By inductive assumption, `v.distance` after $i-1$ cycles is at most the length of this path. Therefore, `uv.weight + v.distance` is at most the length of the path from *s* to *u*. In the ith cycle, `u.distance` gets compared with `uv.weight + v.distance`, and is set equal to it if `uv.weight + v.distance` was smaller. Therefore, after *i* cycles, `u.distance` is at most the length of the shortest path from *source* to *u* that uses at most *i* edges.

If there are no negative-weight cycles, then every shortest path visits each vertex at most once, so at step 3 no further improvements can be made. Conversely, suppose no improvement can be made. Then for any cycle with vertices $v[0], ..., v[k-1]$,

```
v[i].distance <= v[(i-1) mod k].distance + v[(i-1) mod k]v[i].weight
```

Summing around the cycle, the $v[i]$.distance terms and the $v[i-1 \pmod k]$ distance terms cancel, leaving

```
0 <= sum from 1 to k of v[i-1 (mod k)]v[i].weight
```

I.e., every cycle has nonnegative weight.

Finding negative cycles

When the algorithm is used to find shortest paths, the existence of negative cycles is a problem, preventing the algorithm from finding a correct answer. However, since it terminates upon finding a negative cycle, the Bellman-Ford algorithm can be used for applications in which this is the target to be sought - for example in cycle-cancelling techniques in network flow analysis.[1]

Applications in routing

A distributed variant of the Bellman–Ford algorithm is used in distance-vector routing protocols, for example the Routing Information Protocol (RIP). The algorithm is distributed because it involves a number of nodes (routers) within an Autonomous system, a collection of IP networks typically owned by an ISP. It consists of the following steps:

1. Each node calculates the distances between itself and all other nodes within the AS and stores this information as a table.
2. Each node sends its table to all neighboring nodes.
3. When a node receives distance tables from its neighbors, it calculates the shortest routes to all other nodes and updates its own table to reflect any changes.

The main disadvantages of the Bellman–Ford algorithm in this setting are as follows:

- It does not scale well.
- Changes in network topology are not reflected quickly since updates are spread node-by-node.
- Count to infinity (if link or node failures render a node unreachable from some set of other nodes, those nodes may spend forever gradually increasing their estimates of the distance to it, and in the meantime there may be routing loops).

Yen's improvement

Yen (1970) described an improvement to the Bellman–Ford algorithm for a graph without negative-weight cycles. Yen's improvement first assigns some arbitrary linear order on all vertices and then partitions the set of all edges into one of two subsets. The first subset, E_f, contains all edges (v_i, v_j) such that $i < j$; the second, E_b, contains edges (v_i, v_j) such that $i > j$. Each vertex is visited in the order $v_1, v_2, ..., v_{|V|}$, relaxing each outgoing edge from that vertex in E_f. Each vertex is then visited in the order $v_{|V|}, v_{|V|-1}, ..., v_1$, relaxing each outgoing edge from that vertex in E_b. Yen's improvement effectively halves the number of "passes" required for the single-source-shortest-path solution.

Notes

[1] Bang-Jensen, Jørgen; Gregory Gutin. "Section 2.3.4: The Bellman-Ford-Moore algorithm" (http://www.cs.rhul.ac.uk/books/dbook/). *Digraphs: Theory, Algorithms and Applications* (First ed.). ISBN 978-1-84800-997-4. .

[2] Sedgewick, Robert. "Section 21.7: Negative Edge Weights" (http://safari.oreilly.com/0201361213/ch21lev1sec7). *Algorithms in Java* (Third ed.). ISBN 0-201-36121-3.

[3] Kleinberg, Jon; Tardos, Éva (2006), Algorithm Design, New York: Pearson Education, Inc..

References

- Bellman, Richard (1958). "On a routing problem". *Quarterly of Applied Mathematics* **16**: 87–90. MR0102435..
- Ford, L. R., Jr.; Fulkerson, D. R. (1962). *Flows in Networks*. Princeton University Press..
- Cormen, Thomas H.; Leiserson, Charles E., Rivest, Ronald L.. *Introduction to Algorithms*. MIT Press and McGraw-Hill., Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 24.1: The Bellman–Ford algorithm, pp. 588–592. Problem 24-1, pp. 614–615.

- Cormen, Thomas H.; Leiserson, Charles E., Rivest, Ronald L.. *Introduction to Algorithms*. MIT Press and McGraw-Hill., Third Edition. MIT Press, 2009. ISBN 978-0-262-53305-8. Section 24.1: The Bellman–Ford algorithm, pp. 651–655.
- Heineman, George T.; Pollice, Gary; Selkow, Stanley (2008). "Chapter 6: Graph Algorithms". *Algorithms in a Nutshell*. O'Reilly Media. pp. 160–164. ISBN 978-0-596-51624-6.
- Yen, Jin Y. (1970). "An algorithm for finding shortest routes from all source nodes to a given destination in general networks". *Quarterly of Applied Mathematics* **27**: 526–530. MR0253822..

External links

- C code example (http://snippets.dzone.com/posts/show/4828)
- Open Source Java Graph package with Bellman-Ford Algorithms (http://code.google.com/p/annas/)

Dijkstra's algorithm

Dijkstra's algorithm

Dijkstra's algorithm runtime

Class	Search algorithm
Data structure	Graph
Worst case performance	$O(\|E\| + \|V\| \log \|V\|)$

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1956 and published in 1959,[1][2] is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.

For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path first is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First).

Dijkstra's original algorithm does not use a min-priority queue and runs in $O(|V|^2)$. The idea of this algorithm is also given in (Leyzorek et al. 1957). The implementation based on a min-priority queue implemented by a Fibonacci heap and running in $O(|E| + |V| \log |V|)$ is due to (Fredman & Tarjan 1984). This is asymptotically the fastest known single-source shortest-path algorithm for arbitrary directed graphs with unbounded nonnegative weights.

Algorithm

Let the node at which we are starting be called the **initial node**. Let the **distance of node Y** be the distance from the **initial node** to Y. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

Illustration of Dijkstra's algorithm search for finding path from a start node to a goal node in a robot motion planning problem. Open nodes represent the "tentative" set. Filled nodes are visited ones, with color representing the distance: the greener, the farther. Nodes in all the different directions are explored uniformly, appearing as a more-or-less circular wavefront as Dijkstra's algorithm uses a heuristic identically equal to 0.

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Mark all nodes unvisited. Set the initial node as current. Create a set of the unvisited nodes called the *unvisited set* consisting of all the nodes except the initial node.
3. For the current node, consider all of its unvisited neighbors and calculate their *tentative* distances. For example, if the current node A is marked with a tentative distance of 6, and the edge connecting it with a neighbor B has length 2, then the distance to B (through A) will be 6+2=8. If this distance is less than the previously recorded tentative distance of B, then overwrite that distance. Even though a neighbor has been examined, it is not marked as "visited" at this time, and it remains in the *unvisited set*.
4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the *unvisited set*. A visited node will never be checked again; its distance recorded now is final and minimal.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the *unvisited set* is infinity (when planning a complete traversal), then stop. The algorithm has finished.
6. Set the unvisited node marked with the smallest tentative distance as the next "current node" and go back to step 3.

Description

> ***Note:*** *For ease of understanding, this discussion uses the terms* ***intersection****,* ***road*** *and* ***map*** *— however, formally these terms are* ***vertex****,* ***edge*** *and* ***graph****, respectively.*

Suppose you want to find the shortest path between two intersections on a city map, a starting point and a destination. The order is conceptually simple: to start, mark the distance to every intersection on the map with infinity. This is done not to imply there is an infinite distance, but to note that that intersection has not yet been *visited*; some variants of this method simply leave the intersection unlabeled. Now, at each iteration, select a *current* intersection. For the first iteration the current intersection will be the starting point and the distance to it (the intersection's label) will be zero. For subsequent iterations (after the first) the current intersection will be the closest unvisited intersection to the starting point—this will be easy to find.

From the current intersection, update the distance to every unvisited intersection that is directly connected to it. This is done by determining the sum of the distance between an unvisited intersection and the value of the current intersection, and relabeling the unvisited intersection with this value if it is less than its current value. In effect, the intersection is relabeled if the path to it through the current intersection is shorter than the previously known paths. To facilitate shortest path identification, in pencil, mark the road with an arrow pointing to the relabeled intersection if you label/relabel it, and erase all others pointing to it. After you have updated the distances to each neighboring intersection, mark the current intersection as *visited* and select the unvisited intersection with lowest distance (from the starting point) -- or lowest label—as the current intersection. Nodes marked as visited are labeled with the

shortest path from the starting point to it and will not be revisited or returned to.

Continue this process of updating the neighboring intersections with the shortest distances, then marking the current intersection as visited and moving onto the closest unvisited intersection until you have marked the destination as visited. Once you have marked the destination as visited (as is the case with any visited intersection) you have determined the shortest path to it, from the starting point, and can trace your way back, following the arrows in reverse.

Of note is the fact that this algorithm makes no attempt to direct "exploration" towards the destination as one might expect. Rather, the sole consideration in determining the next "current" intersection is its distance from the starting point. In some sense, this algorithm "expands outward" from the starting point, iteratively considering every node that is closer in terms of shortest path distance until it reaches the destination. When understood in this way, it is clear how the algorithm necessarily finds the shortest path, however it may also reveal one of the algorithm's weaknesses: its relative slowness in some topologies.

Pseudocode

In the following algorithm, the code `u := vertex in Q with smallest dist[]`, searches for the vertex `u` in the vertex set `Q` that has the least `dist[u]` value. That vertex is removed from the set `Q` and returned to the user. `dist_between(u, v)` calculates the length between the two neighbor-nodes `u` and `v`. The variable `alt` on lines 20 & 22 is the length of the path from the root node to the neighbor node `v` if it were to go through `u`. If this path is shorter than the current shortest path recorded for `v`, that current path is replaced with this `alt` path. The `previous` array is populated with a pointer to the "next-hop" node on the source graph to get the shortest route to the source.

```
1  function Dijkstra(Graph, source):
2      for each vertex v in Graph:                                // Initializations
3          dist[v] := infinity ;                                  // Unknown distance function from
4                                                                 // source to v
5          previous[v] := undefined ;                             // Previous node in optimal path
6      end for                                                    // from source
7
8      dist[source] := 0 ;                                        // Distance from source to source
9      Q := the set of all nodes in Graph ;                       // All nodes in the graph are
10                                                                // unoptimized - thus are in Q
11     while Q is not empty:                                      // The main loop
12         u := vertex in Q with smallest distance in dist[] ;    // Start node in first case
13         remove u from Q ;
14         if dist[u] = infinity:
15             break ;                                            // all remaining vertices are
16         end if                                                 // inaccessible from source
17
18         for each neighbor v of u:                              // where v has not yet been
19                                                                // removed from Q.
20             alt := dist[u] + dist_between(u, v) ;
21             if alt < dist[v]:                                  // Relax (u,v,a)
22                 dist[v] := alt ;
23                 previous[v] := u ;
24                 decrease-key v in Q;                           // Reorder v in the Queue
25             end if
```

```
26          end for
27      end while
28  return dist;
```

If we are only interested in a shortest path between vertices `source` and `target`, we can terminate the search at line 13 if `u = target`. Now we can read the shortest path from `source` to `target` by reverse iteration:

```
S := empty sequence
u := target
while previous[u] is defined:                          // Construct the shortest path with a stack S
    insert u at the beginning of S                     // Push the vertex into the stack
    u := previous[u]                                   // Traverse from target to source
end while ;
```

Now sequence `S` is the list of vertices constituting one of the shortest paths from `source` to `target`, or the empty sequence if no path exists.

A more general problem would be to find all the shortest paths between `source` and `target` (there might be several different ones of the same length). Then instead of storing only a single node in each entry of `previous[]` we would store all nodes satisfying the relaxation condition. For example, if both `r` and `source` connect to `target` and both of them lie on different shortest paths through `target` (because the edge cost is the same in both cases), then we would add both `r` and `source` to `previous[target]`. When the algorithm completes, `previous[]` data structure will actually describe a graph that is a subset of the original graph with some edges removed. Its key property will be that if the algorithm was run with some starting node, then every path from that node to any other node in the new graph will be the shortest path between those nodes in the original graph, and all paths of that length from the original graph will be present in the new graph. Then to actually find all these shortest paths between two given nodes we would use a path finding algorithm on the new graph, such as depth-first search.

Running time

An upper bound of the running time of Dijkstra's algorithm on a graph with edges E and vertices V can be expressed as a function of $|E|$ and $|V|$ using big-O notation.

For any implementation of vertex set Q the running time is in $O(|E| \cdot dk_Q + |V| \cdot em_Q)$, where dk_Q and em_Q are times needed to perform decrease key and extract minimum operations in set Q, respectively.

The simplest implementation of the Dijkstra's algorithm stores vertices of set Q in an ordinary linked list or array, and extract minimum from Q is simply a linear search through all vertices in Q. In this case, the running time is $O(|E| + |V|^2) = O(|V|^2)$.

For sparse graphs, that is, graphs with far fewer than $O(|V|^2)$ edges, Dijkstra's algorithm can be implemented more efficiently by storing the graph in the form of adjacency lists and using a binary heap, pairing heap, or Fibonacci heap as a priority queue to implement extracting minimum efficiently. With a binary heap, the algorithm requires $\Theta((|E| + |V|) \log |V|)$ time (which is dominated by $\Theta(|E| \log |V|)$, assuming the graph is connected). To avoid O(|V|) look-up in decrease-key step on a vanilla binary heap, it is necessary to maintain a supplementary index mapping each vertex to the heap's index (and keep it up to date as priority queue Q changes), making it take only $O(\log |V|)$ time instead. The Fibonacci heap improves this to $O(|E| + |V| \log |V|)$.

Note that for directed acyclic graphs, it is possible to find shortest paths from a given starting vertex in linear time, by processing the vertices in a topological order, and calculating the path length for each vertex to be the minimum length obtained via any of its incoming edges.[3]

Related problems and algorithms

The functionality of Dijkstra's original algorithm can be extended with a variety of modifications. For example, sometimes it is desirable to present solutions which are less than mathematically optimal. To obtain a ranked list of less-than-optimal solutions, the optimal solution is first calculated. A single edge appearing in the optimal solution is removed from the graph, and the optimum solution to this new graph is calculated. Each edge of the original solution is suppressed in turn and a new shortest-path calculated. The secondary solutions are then ranked and presented after the first optimal solution.

Dijkstra's algorithm is usually the working principle behind link-state routing protocols, OSPF and IS-IS being the most common ones.

Unlike Dijkstra's algorithm, the Bellman-Ford algorithm can be used on graphs with negative edge weights, as long as the graph contains no negative cycle reachable from the source vertex *s*. (The presence of such cycles means there is no shortest path, since the total weight becomes lower each time the cycle is traversed.)

The A* algorithm is a generalization of Dijkstra's algorithm that cuts down on the size of the subgraph that must be explored, if additional information is available that provides a lower bound on the "distance" to the target. This approach can be viewed from the perspective of linear programming: there is a natural linear program for computing shortest paths, and solutions to its dual linear program are feasible if and only if they form a consistent heuristic (speaking roughly, since the sign conventions differ from place to place in the literature). This feasible dual / consistent heuristic defines a nonnegative reduced cost and A* is essentially running Dijkstra's algorithm with these reduced costs. If the dual satisfies the weaker condition of admissibility, then A* is instead more akin to the Bellman-Ford algorithm.

The process that underlies Dijkstra's algorithm is similar to the greedy process used in Prim's algorithm. Prim's purpose is to find a minimum spanning tree that connects all nodes in the graph; Dijkstra is concerned with only two nodes. Prim's does not evaluate the total weight of the path from the starting node, only the individual path.

Dynamic programming perspective

From a dynamic programming point of view, Dijkstra's algorithm is a successive approximation scheme that solves the dynamic programming functional equation for the shortest path problem by the **Reaching** method.[4][5][6]

In fact, Dijkstra's explanation of the logic behind the algorithm,[7] namely

> **Problem 2.** Find the path of minimum total length between two given nodes P and Q .
>
> We use the fact that, if R is a node on the minimal path from P to Q , knowledge of the latter implies the knowledge of the minimal path from P to R .

is a paraphrasing of Bellman's famous Principle of Optimality in the context of the shortest path problem.

Notes

[1] Dijkstra, Edsger; Thomas J. Misa, Editor (2010-08). "An Interview with Edsger W. Dijkstra". *Communications of the ACM* **53** (8): 41–47. doi:10.1145/1787234.1787249. "What is the shortest way to travel from Rotterdam to Groningen? It is the algorithm for the shortest path which I designed in about 20 minutes. One morning I was shopping with my young fiancée, and tired, we sat down on the café terrace to drink a cup of coffee and I was just thinking about whether I could do this, and I then designed the algorithm for the shortest path."

[2] Dijkstra 1959

[3] http://www.boost.org/doc/libs/1_44_0/libs/graph/doc/dag_shortest_paths.html

[4] Sniedovich, M. (2006). "Dijkstra's algorithm revisited: the dynamic programming connexion" (http://matwbn.icm.edu.pl/ksiazki/cc/cc35/cc3536.pdf) (PDF). *Journal of Control and Cybernetics* **35** (3): 599–620. . Online version of the paper with interactive computational modules. (http://www.ifors.ms.unimelb.edu.au/tutorial/dijkstra_new/index.html)

[5] Denardo, E.V. (2003). *Dynamic Programming: Models and Applications*. Mineola, NY: Dover Publications. ISBN 978-0-486-42810-9.

[6] Sniedovich, M. (2010). *Dynamic Programming: Foundations and Principles*. Francis & Taylor. ISBN 978-0-8247-4099-3.

[7] Dijkstra 1959, p. 270

References

- Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs" (http://www-m3.ma.tum.de/twiki/pub/MN0506/WebHome/dijkstra.pdf). *Numerische Mathematik* **1**: 269–271. doi:10.1007/BF01386390.
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "Section 24.3: Dijkstra's algorithm". *Introduction to Algorithms* (Second ed.). MIT Press and McGraw-Hill. pp. 595–601. ISBN 0-262-03293-7.
- Fredman, Michael Lawrence; Tarjan, Robert E. (1984). "Fibonacci heaps and their uses in improved network optimization algorithms" (http://www.computer.org/portal/web/csdl/doi/10.1109/SFCS.1984.715934). *25th Annual Symposium on Foundations of Computer Science* (IEEE): 338–346. doi:10.1109/SFCS.1984.715934.
- Fredman, Michael Lawrence; Tarjan, Robert E. (1987). "Fibonacci heaps and their uses in improved network optimization algorithms" (http://portal.acm.org/citation.cfm?id=28874). *Journal of the Association for Computing Machinery* **34** (3): 596–615. doi:10.1145/28869.28874.
- Zhan, F. Benjamin; Noon, Charles E. (February 1998). "Shortest Path Algorithms: An Evaluation Using Real Road Networks". *Transportation Science* **32** (1): 65–73. doi:10.1287/trsc.32.1.65.
- Leyzorek, M.; Gray, R. S.; Johnson, A. A.; Ladew, W. C.; Meaker, Jr., S. R.; Petry, R. M.; Seitz, R. N. (1957). *Investigation of Model Techniques — First Annual Report — 6 June 1956 — 1 July 1957 — A Study of Model Techniques for Communication Systems*. Cleveland, Ohio: Case Institute of Technology.

External links

- Oral history interview with Edsger W. Dijkstra (http://purl.umn.edu/107247), Charles Babbage Institute University of Minnesota, Minneapolis.
- Dijkstra's Algorithm in C# (http://www.codeproject.com/KB/recipes/ShortestPathCalculation.aspx)
- Fast Priority Queue Implementation of Dijkstra's Algorithm in C# (http://www.codeproject.com/KB/recipes/FastHeapDijkstra.aspx)
- Applet by Carla Laffra of Pace University (http://www.dgp.toronto.edu/people/JamesStewart/270/9798s/Laffra/DijkstraApplet.html)
- Animation of Dijkstra's algorithm (http://www.cs.sunysb.edu/~skiena/combinatorica/animations/dijkstra.html)
- Visualization of Dijkstra's Algorithm (http://students.ceid.upatras.gr/~papagel/english/java_docs/minDijk.htm)
- Shortest Path Problem: Dijkstra's Algorithm (http://www-b2.is.tokushima-u.ac.jp/~ikeda/suuri/dijkstra/Dijkstra.shtml)
- Dijkstra's Algorithm Applet (http://www.unf.edu/~wkloster/foundations/DijkstraApplet/DijkstraApplet.htm)
- Open Source Java Graph package with implementation of Dijkstra's Algorithm (http://code.google.com/p/annas/)
- Haskell implementation of Dijkstra's Algorithm (http://bonsaicode.wordpress.com/2011/01/04/programming-praxis-dijkstra’s-algorithm/) on Bonsai code
- Java Implementation of Dijkstra's Algorithm (http://algowiki.net/wiki/index.php?title=Dijkstra's_algorithm) on AlgoWiki
- QuickGraph, Graph Data Structures and Algorithms for .NET (http://quickgraph.codeplex.com/)
- Implementation in Boost C++ library (http://www.boost.org/doc/libs/1_43_0/libs/graph/doc/dijkstra_shortest_paths.html)
- Implementation in T-SQL (http://hansolav.net/sql/graphs.html)
- A Java library for path finding with Dijkstra's Algorithm and example applet (http://www.stackframe.com/software/PathFinder)

- A MATLAB program for Dijkstra's algorithm (http://www.mathworks.com/matlabcentral/fileexchange/20025-advanced-dijkstras-minimum-path-algorithm)

Routing Information Protocol

The **Routing Information Protocol** (**RIP**) is a distance-vector routing protocol, which employs the hop count as a routing metric. RIP 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. A hop count of 16 is considered an infinite distance and used to deprecate inaccessible, inoperable, or otherwise undesirable routes in the selection process.

RIP implements the split horizon, route poisoning and holddown mechanisms to prevent incorrect routing information from being propagated. These are some of the stability features of RIP. It is also possible to use the so called RMTI[1] (**R**outing Information Protocol with **M**etric-based **T**opology **I**nvestigation) algorithm to cope with the count-to-infinity problem. With its help, it is possible to detect every possible loop with a very small computation effort.

Originally each RIP router transmitted full updates every 30 seconds. In the early deployments, routing tables were small enough that the traffic was not significant. As networks grew in size, however, it became evident there could be a massive traffic burst every 30 seconds, even if the routers had been initialized at random times. It was thought, as a result of random initialization, the routing updates would spread out in time, but this was not true in practice. Sally Floyd and Van Jacobson showed in 1994[2] that, without slight randomization of the update timer, the timers synchronized over time. In most current networking environments, RIP is not the preferred choice for routing as its time to converge and scalability are poor compared to EIGRP, OSPF, or IS-IS (the latter two being link-state routing protocols), and (without RMTI) a hop limit severely limits the size of network it can be used in. However, it is easy to configure, because RIP does not require any parameters on a router unlike other protocols (see here [3] for an animation of basic RIP simulation visualizing RIP configuration and exchanging of Request and Response to discover new routes).

RIP uses the User Datagram Protocol (UDP) as its transport protocol, and is assigned the reserved port number 520.[4]

Versions

There are three versions of the Routing Information Protocol: *RIPv1*, *RIPv2*, and *RIPng*.

RIP version 1

The original specification of RIP, defined in RFC 1058,[5] uses classful routing. The periodic routing updates do not carry subnet information, lacking support for variable length subnet masks (VLSM). This limitation makes it impossible to have different-sized subnets inside of the same network class. In other words, all subnets in a network class must have the same size. There is also no support for router authentication, making RIP vulnerable to various attacks.

RIP version 2

Due to the deficiencies of the original RIP specification, RIP version 2 (RIPv2) was developed in 1993[6] and last standardized in 1998.[7] It included the ability to carry subnet information, thus supporting Classless Inter-Domain Routing (CIDR). To maintain backward compatibility, the hop count limit of 15 remained. RIPv2 has facilities to fully interoperate with the earlier specification if all *Must Be Zero* protocol fields in the RIPv1 messages are properly specified. In addition, a *compatibility switch* feature[7] allows fine-grained interoperability adjustments.

In an effort to avoid unnecessary load on hosts that do not participate in routing, RIPv2 *multicasts* the entire routing table to all adjacent routers at the address 224.0.0.9, as opposed to RIPv1 which uses broadcast. Unicast addressing is still allowed for special applications.

(MD5) authentication for RIP was introduced in 1997.[8][9]

RIPv2 is Internet Standard STD56 (which is RFC 2453).

Route tags were also added in RIP version 2. This functionality allows for routes to be distinguished from internal routes to external redistributed routes from EGP protocols.

RIPng

RIPng (RIP next generation), defined in RFC 2080,[10] is an extension of RIPv2 for support of IPv6, the next generation Internet Protocol. The main differences between RIPv2 and RIPng are:

- Support of IPv6 networking.
- While RIPv2 supports RIPv1 updates authentication, RIPng does not. IPv6 routers were, at the time, supposed to use IPsec for authentication.
- RIPv2 allows attaching arbitrary tags to routes, RIPng does not;
- RIPv2 encodes the next-hop into each route entries, RIPng requires specific encoding of the next hop for a set of route entries.

RIPng sends updates on UDP port 521 using the multicast group FF02::9.

Limitations

- Without using RMTI, Hop count can not exceed 15, in the case that it exceeds this limitation, it will be considered invalid.
- Most RIP networks are flat. There is no concept of areas or boundaries in RIP networks.
- Variable Length Subnet Masks were not supported by RIP version 1.
- Without using RMTI, RIP has slow convergence and count to infinity problems.

Implementations

- Cisco IOS, software used in Cisco routers (supports version 1, version 2 and RIPng)
- Cisco NX-OS software used in Cisco Nexus data center switches (supports RIPv1 and RIPv2)
- Junos software used in Juniper routers, switches, and firewalls (supports RIPv1 and RIPv2)
- routed[11] and route6d, included in most BSD Unix systems
- Routing and Remote Access, a Windows Server feature, contains RIP support
- Quagga, a free open source routing software suite based on GNU Zebra
- BIRD, a free open source routing software suite
- OpenBSD, includes a RIP implementation

Similar protocols

- **IGRP**: Cisco's proprietary Interior Gateway Routing Protocol (IGRP) was a somewhat more capable protocol than RIP. It belongs to the same basic family of distance-vector routing protocols. Cisco has ceased support and distribution of IGRP in their router software. It was replaced by the Enhanced Interior Gateway Routing Protocol (EIGRP) which is a completely new design. While EIGRP still uses a distance-vector model, it relates to IGRP only in using the same routing metrics. IGRP supports multiple metrics for each route, including bandwidth, delay, load, MTU, and reliability.

References

[1] "RMTI project" (http://userp.uni-koblenz.de/~vnuml/rmti/). .
[2] The Synchronization of Periodic Routing Messages (http://www.icir.org/floyd/papers/sync_94.pdf), S. Floyd & V. Jacobson,April 1994
[3] http://www.learningocean.com/view.php?cid=899&protocol=RIP&title=1.%20RIP%20Basic&ctype=1
[4] "*Port Numbers*" (http://www.iana.org/assignments/port-numbers) (plain text). The Internet Assigned Numbers Authority (IANA). 2008-05-22. . Retrieved 2008-05-25.
[5] RFC 1058, *Routing Information Protocol*, C. Hendrik, The Internet Society (June 1988)
[6] RFC 1388, *RIP Version 2 - Carrying Additional Information*, G. Malkin, The Internet Society (January 1993)
[7] RFC 2453, *RIP Version 2*, G. Malkin, The Internet Society (November 1998)
[8] RFC 2082, *RIP-2 MD5 Authentication*, F. Baker, R. Atkinson, The Internet Society (January 1997)
[9] RFC 4822, *RIPv2 Cryptographic Authentication*, R. Atkinson, M. Fanto, The Internet Society (January 2007)
[10] RFC 2080, *RIPng for IPv6*, G. Malkin, R. Minnear, The Internet Society (January 1997)
[11] Man-page for "routed(8)" on FreeBSD, http://www.freebsd.org/cgi/man.cgi?query=routed&sektion=8

Eigrp

Further reading

- Malkin, Gary Scott (2000). *RIP: An Intra-Domain Routing Protocol*. Addison-Wesley Longman. ISBN 0-201-43320-6.
- Edward A. Taft, *Gateway Information Protocol (revised)* (Xerox Parc, Palo Alto, May, 1979)
- *Xerox System Integration Standard - Internet Transport Protocols* (Xerox, Stamford, 1981)

External links

- Interactive RIP simulation (http://www.visualland.net/view.php?cid=899&protocol=RIP&title=1.RIP Basic&ctype=1)

Open Shortest Path First

Open Shortest Path First (**OSPF**) is an link-state routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS). It is defined as OSPF Version 2 in RFC 2328 (1998) for IPv4.[1] The updates for IPv6 are specified as OSPF Version 3 in RFC 5340 (2008).[2]

OSPF is perhaps the most widely used interior gateway protocol (IGP) in large enterprise networks. IS-IS, another link-state dynamic routing protocol, is more common in large service provider networks. The most widely used exterior gateway protocol is the Border Gateway Protocol (BGP), the principal routing protocol between autonomous systems on the Internet.

Overview

OSPF is an interior gateway protocol that routes Internet Protocol (IP) packets solely within a single routing domain (autonomous system). It 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.

The **link-state** information is maintained on each router as a link-state database (LSDB) which is a tree-image of the entire network topology. Identical copies of the LSDB are periodically updated through flooding on all OSPF routers.

The OSPF routing policies to construct a route table are governed by link cost factors (*external metrics*) associated with each routing interface. Cost factors may be the distance of a router (round-trip time), network throughput of a link, or link availability and reliability, expressed as simple unitless numbers. This provides a dynamic process of traffic load balancing between routes of equal cost.

An OSPF network may be structured, or subdivided, into routing *areas* to simplify administration and optimize traffic and resource utilization. Areas are identified by 32-bit numbers, expressed either simply in decimal, or often in octet-based dot-decimal notation, familiar from IPv4 address notation.

By convention, area 0 (zero) or `0.0.0.0` represents the core or *backbone* region of an OSPF network. The identifications of other areas may be chosen at will; often, administrators select the IP address of a main router in an area as the area's identification. Each additional area must have a direct or virtual connection to the backbone OSPF area. Such connections are maintained by an interconnecting router, known as *area border router* (ABR). An ABR maintains separate link state databases for each area it serves and maintains summarized routes for all areas in the network.

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.

OSPF uses multicast addressing for route flooding on a broadcast network link. For non-broadcast networks special provisions for configuration facilitate neighbor discovery.[1] OSPF multicast IP packets never traverse IP routers, they never travel more than one hop. OSPF reserves the multicast addresses `224.0.0.5` for IPv4 or `FF02::5` for IPv6 (all SPF/link state routers, also known as `AllSPFRouters`) and `224.0.0.6` for IPv4 or `FF02::6` for IPv6 (all Designated Routers, `AllDRouters`), as specified in RFC 2328[3] and RFC 5340.[4]

For routing multicast IP traffic, OSPF supports the Multicast Open Shortest Path First protocol (MOSPF) as defined in RFC 1584.[5] Neither Cisco nor Juniper Networks include MOSPF in their OSPF implementations. PIM (Protocol Independent Multicast) in conjunction with OSPF or other IGPs, (Interior Gateway Protocol), is widely deployed.

The OSPF protocol, when running on IPv4, can operate securely between routers, optionally using a variety of authentication methods to allow only trusted routers to participate in routing. OSPFv3, running on IPv6, no longer supports protocol-internal authentication. Instead, it relies on IPv6 protocol security (IPsec).

OSPF version 3 introduces modifications to the IPv4 implementation of the protocol.[2] Except for virtual links, all neighbor exchanges use IPv6 link-local addressing exclusively. The IPv6 protocol runs per link, rather than based on the subnet. All IP prefix information has been removed from the link-state advertisements and from the *Hello* discovery packet making OSPFv3 essentially protocol-independent. Despite the expanded IP addressing to 128-bits in IPv6, area and router Identifications are still based on 32-bit values.

Neighbor relationships

Routers in the same broadcast domain or at each end of a point-to-point telecommunications link form *adjacencies* when they have detected each other. This detection occurs when a router identifies itself in a *hello* OSPF protocol packet. This is called a *two-way state* and is the most basic relationship. The routers in an Ethernet or frame relay network select a *designated router* (DR) and a *backup designated router* (BDR) which act as a hub to reduce traffic between routers. OSPF uses both unicast and multicast to send "hello packets" and link state updates.

As a link state routing protocol, OSPF establishes and maintains neighbor relationships in order to exchange routing updates with other routers. The neighbor relationship table is called an adjacency database in OSPF. Provided that OSPF is configured correctly, OSPF forms neighbor relationships only with the routers directly connected to it. In order to form a neighbor relationship between two routers, the interfaces used to form the relationship must be in the same area. An interface can only belong to a single area. (A neighbor state simulation [6] shows how neighbor state changes from Down to Full Adjacency progressively with exchanging Hello, DD, Request, Update, and Ack packets).

Area types

An OSPF domain is divided into *areas* that are labeled with 32-bit area identifiers. The area identifiers are commonly, but not always, written in the dot-decimal notation of an IPv4 address. However, they are *not* IP addresses and may duplicate, without conflict, any IPv4 address. The area identifiers for IPv6 implementations of OSPF (OSPFv3) also use 32-bit identifiers written in the same notation. While most OSPF implementations will right-justify an area number written in a format other than dotted decimal format (e.g., area 1), it is wise to always use dotted-decimal formats. Most implementations expand area 1 to the area identifier 0.0.0.1, but some have been known to expand it as 1.0.0.0.

Areas are logical groupings of hosts and networks, including their routers having interfaces connected to any of the included networks. Each area maintains a separate link state database whose information may be summarized towards the rest of the network by the connecting router. Thus, the topology of an area is unknown outside of the area. This reduces the amount of routing traffic between parts of an autonomous system. (An ABR simulation [7] shows how an ABR lets areas know each other's network addresses by flooding Summary LSA.)

Several special area types are defined.

Backbone area

The backbone area (also known as *area 0* or *area 0.0.0.0*) forms the core of an OSPF network. All other areas are connected to it, and inter-area routing happens via routers connected to the backbone area and to their own associated areas. It is the logical and physical structure for the 'OSPF domain' and is attached to all nonzero areas in the OSPF domain. Note that in OSPF the term Autonomous System Boundary Router (ASBR) is historic, in the sense that many OSPF domains can coexist in the same Internet-visible autonomous system, RFC1996 (ASGuidelines 1996, p. 25).[8]

The backbone area is responsible for distributing routing information between nonbackbone areas. The backbone must be contiguous, but it does not need to be physically contiguous; backbone connectivity can be established and maintained through the configuration of virtual links.

All OSPF areas must connect to the backbone area. This connection, however, can be through a virtual link. For example, assume area 0.0.0.1 has a physical connection to area 0.0.0.0. Further assume that area 0.0.0.2 has no direct connection to the backbone, but this area does have a connection to area 0.0.0.1. Area 0.0.0.2 can use a virtual link through the *transit area* 0.0.0.1 to reach the backbone. To be a transit area, an area has to have the transit attribute, so it cannot be stubby in any way.

Stub area

A stub area is an area which does not receive route advertisements external to the autonomous system (AS) and routing from within the area is based entirely on a default route. A Stub Area simulation [9] shows how an ABR deletes type 4, 5 LSAs from internal routers, sends them a default route of 0.0.0.0 and turns itself into a default gateway. This reduces LSDB and routing table size for internal routers.

Modifications to the basic concept of stub areas exist in the *not-so-stubby* area (NSSA). In addition, several other proprietary variations have been implemented by systems vendors, such as the *totally stubby area* (TSA) and the *NSSA not so stubby area*, both an extension in Cisco Systems routing equipment.

Not-so-stubby area

A *not-so-stubby area* (NSSA) is a type of stub area that can import autonomous system external routes and send them to other areas, but still cannot receive AS-external routes from other areas. NSSA is an extension of the stub area feature that allows the injection of external routes in a limited fashion into the stub area. A case study simulates [10] an NSSA getting around the Stub Area problem of not being able to import external addresses. It visualizes the following activities: the ASBR imports external addresses with a type 7 LSA, the ABR converts a type 7 LSA to type 5 and floods it to other areas, the ABR acts as an "ASBR" for other areas. The ABR's do not take type 5 LSA's and then convert to type 7 LSA's for the area.

Proprietary extensions

Several vendors (Cisco, Juniper, Alcatel-Lucent, Huawei, Quagga), now implement the below two extensions to stub and NSSA area and although not covered by RFC they are considered by many to be standard features in OSPF implementations.

Totally stubby area

> A *totally stubby area* is similar to a stub area. However, this area does not allow *summary* routes in addition to not having *external* routes, that is, *inter-area* (IA) routes are not summarized into totally stubby areas. The only way for traffic to get routed outside of the area is a default route which is the only Type-3 LSA advertised into the area. When there is only one route out of the area, fewer routing decisions have to be made by the route processor, which lowers system resource utilization.
>
> Occasionally, it is said that a TSA can have only one ABR.[11]

NSSA totally stubby area

An addition to the standard functionality of an NSSA, the *totally stubby NSSA* is an NSSA that takes on the attributes of a TSA, meaning that type 3 and 4 summary routes are not flooded into this type of area. It is also possible to declare an area both totally stubby and not-so-stubby, which means that the area will receive only the default route from area 0.0.0.0, but can also contain an autonomous system boundary router (ASBR) that accepts external routing information and injects it into the local area, and from the local area into area 0.0.0.0.

Redistribution into an NSSA area creates a special type of LSA known as TYPE 7, which can exist only in an NSSA area. An NSSA ASBR generates this LSA, and an NSSA ABR router translates it into type 5 LSA which gets propagated into the OSPF domain.

A newly acquired subsidiary is one example of where it might be suitable for an area to be simultaneously not-so-stubby and totally stubby if the practical place to put an ASBR is on the edge of a totally stubby area. In such a case, the ASBR does send externals into the totally stubby area, and they are available to OSPF speakers within that area. In Cisco's implementation, the external routes can be summarized before injecting them into the totally stubby area. In general, the ASBR should not advertise default into the TSA-NSSA, although this can work with extremely careful design and operation, for the limited special cases in which such an advertisement makes sense.

By declaring the totally stubby area as NSSA, no external routes from the backbone, except the default route, enter the area being discussed. The externals do reach area 0.0.0.0 via the TSA-NSSA, but no routes other than the default route enter the TSA-NSSA. Routers in the TSA-NSSA send all traffic to the ABR, except to routes advertised by the ASBR.

Transit area

A transit area is an area with two or more OSPF border routers and is used to pass network traffic from one adjacent area to another. The transit area does not originate this traffic and is not the destination of such traffic.

Path preference

OSPF uses *path cost* as its basic routing metric, which was defined by the standard not to equate to any standard value such as speed, so the network designer could pick a metric important to the design. In practice, it is determined by the speed (bandwidth) of the interface addressing the given route, although that tends to need network-specific scaling factors now that links faster than 100 Mbit/s are common. Cisco uses a metric like 10^8/bandwidth (the base value, 10^8 by default, can be adjusted). So, a 100Mbit/s link will have a cost of 1, a 10Mbit/s a cost of 10 and so on. But for links faster than 100Mbit/s, the cost would be <1.

Metrics, however, are only directly comparable when of the same type. Four types of metrics are recognized. An intra-area route is always preferred to an External route regardless of metric. In decreasing preference, these types are:

1. Intra-area
2. Inter-area
3. External Type 1, which includes both the external path cost and the sum of internal path costs to the ASBR that advertises the route,
4. External Type 2, the value of which is solely that of the external path cost

Traffic engineering

OSPF-TE is an extension to OSPF extending the expressivity to allow for traffic engineering and use on non-IP networks (RFC 3630).[12] More information about the topology can be exchanged using opaque LSA carrying type-length-value elements. These extensions allow OSPF-TE to run completely out of band of the data plane network. This means that it can also be used on non-IP networks, such as optical networks.

OSPF-TE is used in GMPLS networks as a means to describe the topology over which GMPLS paths can be established. GMPLS uses its own path setup and forwarding protocols, once it has the full network map.

In the Resource Reservation Protocol (RSVP), OSPF-TE is used for recording and flooding RSVP signaled bandwidth reservations for Label switched paths within the link-state database.

Other extensions

RFC 3717 documents work in optical routing for IP, based on "constraint-based" extensions to OSPF and IS-IS.[13]

OSPF router types

OSPF defines the following router types:

- Area border router (ABR)
- Autonomous system boundary router (ASBR)
- Internal router (IR)
- Backbone router (BR)

The router type is an attribute of an OSPF process. A given physical router may have one or more OSPF processes. For example, a router that is connected to more than one area, and which receives routes from a BGP process connected to another AS, is both an area border router and an autonomous system boundary router.

Each router has an identifier, customarily written in the dotted decimal format (e.g., 1.2.3.4) of an IP address. This identifier must be established in every OSPF instance. If not explicitly configured, the highest logical IP address will be duplicated as the router identifier. However, since the router identifier is not an IP address, it does not have to be a part of any routable subnet in the network, and often isn't to avoid confusion.

These router types should not be confused with the terms *designated router (DR)*, or *backup designated router (BDR)*, which are attributes of a router interface, not the router itself.

Area border router

An area border router (ABR) is a router that connects one or more areas to the main backbone network. It is considered a member of all areas it is connected to. An ABR keeps multiple copies of the link-state database in memory, one for each area to which that router is connected.

Autonomous system boundary router

An autonomous system boundary router (ASBR) is a router that is connected to more than one Routing protocol and that exchanges routing information with routers in other protocols. ASBRs typically also run an exterior routing protocol (e.g., BGP), or use static routes, or both. An ASBR is used to distribute routes received from other, external ASs throughout its own autonomous system. (An interactive ASBR simulation [14] shows how an ASBR creates External LSA for external addresses and floods them to all areas via ABR.) Routers in other areas use ABR as next hop to access external addresses. Then ABR forwards packets to the ASBR that announces the external addresses.

Internal router

An internal router is a router that has OSPF neighbor relationships with interfaces in the same area. An internal router has all its interfaces in a single area.

Backbone router

Backbone routers are all routers that are connected to the OSPF backbone, irrespective of whether they are also area border routers or internal routers of the backbone area. An area border router is always a backbone router, since all areas must be either directly connected to the backbone or connected to the backbone via a virtual link (spanning across another area to get to the backbone).

Designated router

A *designated router* (DR) is the router interface elected among all routers on a particular multiaccess network segment, generally assumed to be broadcast multiaccess. A DR Election Simulation [15] visualizes the basic neighbor discovery process (Hello), flooding (224.0.0.6), DR election (priority, RID). Special techniques, often vendor-dependent, may be needed to support the DR function on nonbroadcast multiaccess (NBMA) media. It is usually wise to configure the individual virtual circuits of a NBMA subnet as individual point-to-point lines; the techniques used are implementation-dependent.

Do not confuse the DR with an OSPF router type. A given physical router can have some interfaces that are designated (DR), others that are backup designated (BDR), and others that are non-designated. If no router is DR or BDR on a given subnet, the DR is first elected, and then a second election is held if there is more than one BDR.[16] (A DR Election Detail Simulation [17] shows a step-by-step DR election example: How neighbor list, neighbor state, DR, and BDR are changed when receiving Hello) The DR is elected based on the following default criteria:

- If the priority setting on an OSPF router is set to 0, that means it can NEVER become a DR or BDR (Backup Designated Router).
- When a DR fails and the BDR takes over, there is another election to see who becomes the replacement BDR.
- The router sending the Hello packets with the highest priority wins the election.
- If two or more routers tie with the highest priority setting, the router sending the Hello with the highest RID (Router ID) wins. NOTE: a RID is the highest logical (loopback) IP address configured on a router, if no logical/loopback IP address is set then the Router uses the highest IP address configured on its active interfaces. (e.g. 192.168.0.1 would be higher than 10.1.1.2).
- Usually the router with the second highest priority number becomes the BDR.
- The priority values range between 0 - 255,[18] with a higher value increasing its chances of becoming DR or BDR.
- IF a HIGHER priority OSPF router comes online AFTER the election has taken place, it will not become DR or BDR until (at least) the DR and BDR fail.
- If the current DR 'goes down' the current BDR becomes the new DR and a new election takes place to find another BDR. If the new DR then 'goes down' and the original DR is now available, still previously chosen BDR will become DR.

DR's exist for the purpose of reducing network traffic by providing a source for routing updates. The DR maintains a complete topology table of the network and sends the updates to the other routers via multicast. All routers in a multi-access network segment will form a slave/master relationship with the DR. They will form adjacencies with the DR and BDR only. Every time a router sends an update, it sends it to the DR and BDR on the multicast address 224.0.0.6. The DR will then send the update out to all other routers in the area, to the multicast address 224.0.0.5. This way all the routers do not have to constantly update each other, and can rather get all their updates from a single source. The use of multicasting further reduces the network load. DRs and BDRs are always setup/elected on OSPF broadcast networks. DR's can also be elected on NBMA (Non-Broadcast Multi-Access) networks such as Frame Relay or ATM. DRs or BDRs are not elected on point-to-point links (such as a point-to-point WAN connection)

because the two routers on either sides of the link must become fully adjacent and the bandwidth between them cannot be further optimized. DR LSDB Synch Simulation [19] shows how DR and non-DR routers evolve from 2-way to full adjacency relationships by exchanging DD, Request, and Update.

Backup designated router

A *backup designated router* (BDR) is a router that becomes the designated router if the current designated router has a problem or fails. The BDR is the OSPF router with second highest priority at the time of the last election.

OSPF v3 Packet Formats

The "Main OSPF Packet Header" is the same for all 5 types of packets (with exception of the Type field) whereas the following sub-headers will vary from type to type and are shown below the Main OSPF Packet Header.

The Main OSPF Packet Header

Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version								Type								Packet Length															
4	32	Router ID																															
8	64	Area ID																															
12	96	Checksum																Instance ID								0							

As per Appendix A.3 of RFC 5340 (OSPFv3 for IPv6) there are 5 OSPF Packet formats as follows:

Type	Description
1	Hello
2	Database Description
3	Link State Request
4	Link State Update
5	Link State Acknowledgement

The five different formats for each "Type" of OSPF v3 packet are listed below:

Type 1: The Hello Packet

Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	3 {Ver}								1 {Type}								Packet Length															
4	32	Router ID																															
8	64	Area ID																															
12	96	Checksum																Instance ID								0							
16	128	Interface ID																															
20	160	Rtr Priority								Options (Explained below)																							
24	192	HelloInterval																RouterDeadInterval															
28	224	Designated Router ID																															
32	256	Backup Designated Router ID																															
36	288	Neighbor ID																															

~	~	...

Type 2: The Database Description Packet

Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	**0**	3 {Ver}								2 {Type}								Packet Length															
4	**32**	Router ID																															
8	**64**	Area ID																															
12	**96**	Checksum																Instance ID								0							
16	**128**	0								Options (Explained below)																							
20	**160**	Interface MTU																0								0	0	0	0	0	I	M	M S
24	**192**	DD sequence number																															
28	**224**	An LSA Header																															
32	**256**																																
36	**288**																																
40	**320**																																
44	**352**																																
~	**~**	...																															

Type 3: The OSPF Link State Request Packet

Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	**0**	3 {Ver}								3 {Type}								Packet Length															
4	**32**	Router ID																															
8	**64**	Area ID																															
12	**96**	Checksum																Instance ID								0							
16	**128**	0																LS Type															
20	**160**	Link State ID																															
24	**192**	Advertising Router																															
~	**~**	...																															

Type 4: The OSPF Link State Update Packet

Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	3 {Ver}								4 {Type}								Packet Length															
4	32	Router ID																															
8	64	Area ID																															
12	96	Checksum																Instance ID								0							
16	128	# LSAs																															
20	160	LSAs																															
24	192																																
28	224																																
32	256																																
36	288																																
~	~	...																															

Type 5: The OSPF Link State Acknowledgement Packet

Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	3 {Ver}								5 {Type}								Packet Length															
4	32	Router ID																															
8	64	Area ID																															
12	96	Checksum																Instance ID								0							
16	128	An LSA Header (Shown below)																															
20	160																																
24	192																																
28	224																																
32	256																																
~	~	...																															

The OSPFv3 (24 Bit) Options Field

This "Options Field" is used in OSPF Hello packets, Database Description packets, and certain LSAs (router-LSAs, network-LSAs, inter-area-router-LSAs, and link-LSAs).

(Note: Previous OSPF versions {v1 & v2} DO NOT support all of the options/fields listed here.)

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
																	*	*	DC	R	N	x	E	V6

Explanation of the bits in the Options field:

There are currently only 7-bits assigned.

V6-bit: "V6" stands for IP[v6] routing calculations are to be used.

E-bit: "E" stands for [E]xternal as in AS-External-LSA flooding as specified in OSPFv2.

x-bit: This is currently deprecated. It was previously used by MOSPF.

N-bit: "N" stands for [N]SSA (Not So Stubby Area) and used for routers which are attached to NSSA networks.

R-bit: "R" stands for [R]outer and specifies whether the router is Active or not.

DC-bit: "DC" stands for [D]emand [C]ircuts and is specified in RFC 1793.

*-bits: These two bits are reserved for migration of OSPFv2 protocol extensions.

The remaining 16-bits have yet to be assigned.

OSPF in broadcast and non-broadcast networks

In broadcast multiple-access networks, neighbor adjacency is formed dynamically using multicast hello packets to 224.0.0.5. A DR and BDR are elected normally, and function normally.

For non-broadcast multiple-access networks (NBMA), RFC 2328 defined the following two official modes for OSPF:

- nonbroadcast
- point-to-multipoint

Cisco has defined the following three additional modes for OSPF in NBMA topologies:

- point-to-multipoint nonbroadcast
- broadcast
- point-to-point

Implementations

- BIRD implements both OSPFv2 and OSPFv3
- GNU Zebra, a GPL routing suite for Unix-like systems supporting OSPF
- Netware implements OSPF in its Multi Protocol Routing module.
- OpenBSD includes an OpenOSPFD implementation within the OpenBGPD protocol.
- Quagga, a fork of GNU Zebra for Unix-like systems
- XORP, a routing suite implementing RFC2328 (OSPFv2) and RFC2740 (OSPFv3) for both IPv4 and IPv6
- Windows NT 4.0 Server, Windows 2000 Server and Windows Server 2003 implement OSPFv2 in the Routing and Remote Access Service, although the functionality was removed in Windows Server 2008.

Applications

OSPF was the first widely deployed routing protocol that could converge a network in the low seconds, and guarantee loop-free paths. It has many features that allow the imposition of policies about the propagation of routes that it may be appropriate to keep local, for load sharing, and for selective route importing more than IS-IS. IS-IS, in contrast, can be tuned for lower overhead in a stable network, the sort more common in ISP than enterprise networks. There are some historical accidents that made IS-IS the preferred IGP for ISPs, but ISP's today may well choose to use the features of the now-efficient implementations of OSPF,[20] after first considering the pros and cons of IS-IS in service provider environments.[21]

As mentioned, OSPF can provide better load-sharing on external links than other IGPs. When the default route to an ISP is injected into OSPF from multiple ASBRs as a Type I external route and the same external cost specified, other routers will go to the ASBR with the least path cost from its location. This can be tuned further by adjusting the external cost.

In contrast, if the default route from different ISPs is injected with different external costs, as a Type II external route, the lower-cost default becomes the primary exit and the higher-cost becomes the backup only.

The only real limiting factor that may compel major ISPs to select IS-IS over OSPF is if they have a network with more than 850 routers. There is mention of an OSPF network with over 1000 routers,[22] but that is quite uncommon and the network must be specifically designed to minimize overhead to achieve stable operation.

References

[1] Moy, J. (April 1998). [RFC 2328 "OSPF Version 2"]. The Internet Society. OSPFv2. RFC 2328. Retrieved 2007-09-28.

[2] Coltun, R.; D. Ferguson, J Moy, A. Lindem (July 2008). [RFC 5340 "OSPF for IPv6"]. The Internet Society. OSPFv3. RFC 5340. Retrieved 2008-07-23.

[3] "RFC 2328 - OSPF Version 2" (http://tools.ietf.org/html/rfc2328#page-185). Tools.ietf.org. 1998-04-02. . Retrieved 2011-11-30.

[4] "RFC 5340 - OSPF for IPv6" (http://tools.ietf.org/html/rfc5340#page-57). Tools.ietf.org. . Retrieved 2011-11-30.

[5] RFC 1584, *Multicast Extensions to OSPF*, J. Moy, The Internet Society (March 1994)

[6] http://www.youtube.com/watch?v=ajCwMMXGHVc

[7] http://www.visualland.net/view.php?cid=931

[8] Hawkinson, J; T. Bates (March 1996). "Guidelines for creation, selection, and registration of an Autonomous System" (http://tools.ietf.org/html/rfc1930). Internet Engineering Task Force. ASguidelines. . Retrieved 2007-09-28.

[9] http://www.visualland.net/view.php?cid=760

[10] http://www.visualland.net/view.php?cid=761

[11] "Stub Area Design Golden Rules" (http://www.groupstudy.com/bookstore/samples/thomas/). Groupstudy.com. . Retrieved 2011-11-30.. Note: This is not necessarily true. If there are multiple ABRs, as might be required for high availability, routers interior to the TSA will send non-intra-area traffic to the ABR with the lowest intra-area metric (the "closest" ABR) but that requires special configuration.

[12] Katz, D; D. Yeung (September 2003). [RFC 3630 "Traffic Engineering (TE) Extensions to OSPF Version 2"]. The Internet Society. OSPF-TEextensions. RFC 3630. Retrieved 2007-09-28.

[13] Rajagopalan, B; J. Luciani & D. Awduche (March 2004). [RFC 3717 "IP over Optical Networks: A Framework"]. Internet Engineering Task Force. OSPFoverOptical. RFC 3717. Retrieved 2007-09-28.

[14] http://www.visualland.net/view.php?cid=927

[15] http://www.visualland.net/view.php?cid=924

[16] RFC 2328, page 75

[17] http://www.visualland.net/view.php?cid=940

[18] http://www.cisco.com/en/US/docs/ios/iproute_ospf/command/reference/iro_cr_book.pdf

[19] http://www.visualland.net/view.php?cid=941

[20] Berkowitz, Howard (1999). "OSPF Goodies for ISPs" (http://www.nanog.org/meetings/nanog17/abstracts.php?pt=MTE0OSZuYW5vZzE3&nm=nanog17). *North American Network Operators Group NANOG 17*. Montreal. OSPFforISPs.

[21] Katz, Dave (2000). "OSPF and IS-IS: A Comparative Anatomy" (http://www.nanog.org/meetings/nanog19/abstracts.php?pt=MTA4NCZuYW5vZzE5&nm=nanog19). *North American Network Operators Group NANOG 19*. Albuquerque. OSPFvsISIS.

[22] "CCIE, CCNA, CCNP and other Cisco Certifications" (http://www.groupstudy.com/archives/cisco/200011/msg01985.html). GroupStudy.com. . Retrieved 2011-11-30.

Further reading

- Colton, Andrew. *OSPF for Cisco Routers*. Rocket Science Press. ISBN 978-0972286213.
- Doyle, Jeff; Carroll, Jennifer. *Routing TCP/IP* (http://www.ciscopress.com/bookstore/product.asp?isbn=1587052024). **1** (2nd ed.). Cisco Press. ISBN 978-1-58705-202-6.
- Moy, John T.. *OSPF: Anatomy of an Internet Routing Protocol*. Addison-Wesley. ISBN 978-0201634723.
- Parkhurst, William R.. *Cisco OSPF Command and Configuration Handbook*. ISBN 978-1-58705-071-8.
- "Configuring OSPF Authentication" (http://www.netcordia.com/resources/tech-tips/configuring-ospf-authentication.asp). *Tech Tips*. Netcordia. Retrieved 2009-09-10.
- Basu, Anindya; Riecke, Jon (2001). "Stability issues in OSPF routing". *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. SIGCOMM '01. pp. 225–236. doi:10.1145/383059.383077. ISBN 1-58113-411-8. CiteSeerX: 10.1.1.99.6393 (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.99.6393).

External links

- IETF OSPF Working Group (http://www.ietf.org/html.charters/ospf-charter.html)
- OSPF Basics (http://www.setup32.com/network-administration/networking/know-ospf.php)
- Cisco OSPF (http://www.cisco.com/en/US/tech/tk365/tk480/tsd_technology_support_sub-protocol_home.html)
- Cisco OSPF Areas and Virtual Links (http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094aaa.shtml)
- Summary of OSPF v2 (http://www.freesoft.org/CIE/Topics/89.htm)
- OSPF and IS-IS: A Comparative Anatomy (http://www.nanog.org/meetings/nanog19/presentations/katz.ppt) by Dave Katz, Juniper
- IS-IS and OSPF difference discussion (http://www.join.uni-muenster.de/Dokumente/drafts/draft-bhatia-manral-diff-isis-ospf-01.txt) (Vishwas Manral, Manav bhatia and Yasuhiro Ohara)
- Data Communication Lectures of Manfred Lindner - Part OSPF Fundamentals (http://www.ict.tuwien.ac.at/lva/384.081/infobase/L41-OSPF_Fundamentals_v4-5.pdf)
- Data Communication Lectures of Manfred Lindner - Part OSPF Areas (http://www.ict.tuwien.ac.at/lva/384.081/infobase/L42-OSPF_Advanced_v4-4.pdf)
- Good comparison of IS-IS vs OSPF (http://www.ecse.rpi.edu/Homepages/koushik/shivkuma-teaching/sp2002/ip2002-isis-ospf.pdf)

Autonomous System (Internet)

Within the Internet, 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.[1]

Originally the definition required control by a single entity, typically an Internet service provider or a very large organization with independent connections to multiple networks, that adhere to a single and clearly defined routing policy, as originally defined in RFC 1771.[2] The newer definition in RFC 1930 came into use because multiple organizations can run BGP using private AS numbers to an ISP that connects all those organizations to the Internet. Even though there may be multiple Autonomous Systems supported by the ISP, the Internet only sees the routing policy of the ISP. That ISP must have an officially registered Autonomous System Number (**ASN**).

A unique ASN is allocated to each AS for use in BGP routing. 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.[3] ASNs 64496-64511 are reserved for use in documentation and sample code.[4] The ASNs 0 and 65535 are reserved; ASN 0 may be used to identify non-routed networks. All other ASNs are subject to assignment by IANA. RFC 4893 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.[3] The accepted textual representation of Autonomous System Numbers is defined in RFC 5396.[5]

The number of unique autonomous networks in the routing system of the Internet exceeded 5000 in 1999, 30000 in late 2008, 35000 in mid 2010, and 42000 in late 2012. [6]

Assignment

AS numbers are assigned in blocks by the Internet Assigned Numbers Authority (IANA) to Regional Internet Registries (RIRs). The appropriate RIR then assigns AS numbers to entities within its designated area from the block assigned by the IANA. Entities wishing to receive an ASN must complete the application process of their local RIR and be approved before being assigned an ASN. Current IANA ASN assignments can be found on the IANA website.[3]

Types

Autonomous Systems can be grouped into three categories, depending on their connectivity and operating policy.

A *multihomed* Autonomous System is an AS that maintains connections to more than one other AS. This allows the AS to remain connected to the Internet in the event of a complete failure of one of their connections. However, this type of AS would not allow traffic from one AS to pass through on its way to another AS.

A *stub* Autonomous System refers to an AS that is connected to only one other AS. This may be an apparent waste of an AS number if the network's routing policy is the same as its upstream AS's. However, the stub AS may in fact have peering with other Autonomous Systems that is not reflected in public route-view servers. Specific examples include private interconnections in the financial and transportation sectors.

A *transit* Autonomous System is an AS that provides connections through itself to other networks. That is, network A can use network B, the transit AS, to connect to network C. ISPs are always transit ASs, because they provide connections from one network to another. The ISP is considered to be 'selling transit service' to the customer network, thus the term *transit AS*.

References

[1] RFC 1930, Section 3
[2] RFC 1771, original definition (now obsolete) of the Border Gateway Protocol (BGP)
[3] Autonomous System Numbers (http://www.iana.org/assignments/as-numbers)
[4] RFC 5398
[5] RFC 5396, *Textual Representation of Autonomous System (AS) Numbers*, G. Huston, G. Michaelson, The Internet Society (December 2008)
[6] Tony Bates, Philip Smith, Geoff Huston. "CIDR report" (http://www.cidr-report.org/as2.0/). . Retrieved 2010-09-17.

External links

- ASN Resource Guide (http://www.apnic.net/services/asn_guide.html)
- AS Number plugin for Mozilla Firefox (http://www.asnumber.networx.ch)
- CIDR and ASN assignment report update continuously (http://www.cidr-report.org/as2.0/)
- Exploring Autonomous System Numbers (http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-1/autonomous_system_numbers.html)
- 32-bit ASN presentation at Doha, Qatar in 11-2007 (http://www.menog.net/meetings/menog2/presentations/philip-smith-32bit-asn.pdf)
- ASN Searching and Tracing by Fixed Orbit (http://www.fixedorbit.com/search.htm)
- ASN Lookup (http://networktools.nl/as-info)
- ashunt, an AS traceroute utility (part of netsniff-ng) (http://www.netsniff-ng.org/)
- Autonomous System Whois Lookup (http://ipduh.com/ip/whois/as/)

Border Gateway Protocol

Border Gateway Protocol (**BGP**) is the protocol which makes core routing decisions on the Internet. It maintains a table of IP networks or 'prefixes' which designate network reach-ability among autonomous systems (AS). It is a path vector protocol, or a variant of a Distance-vector routing protocol. BGP does not use traditional Interior Gateway Protocol (**IGP**) metrics, but makes routing decisions based on path, network policies and/or rule-sets. For this reason, it is more appropriately termed a reach-ability protocol rather than routing protocol.

BGP was created to replace the Exterior Gateway Protocol (**EGP**) to allow fully decentralized routing in order to transition from the core ARPAnet model to a decentralized system that included the NSFNET backbone and its associated regional networks. This allowed the Internet to become a truly decentralized system. Since 1994, version four of the BGP has been in use on the Internet. All previous versions are now obsolete. The major enhancement in version 4 was support of Classless Inter-Domain Routing and use of route aggregation to decrease the size of routing tables. Since January 2006, version 4 is codified in RFC 4271, which went through more than 20 drafts based on the earlier RFC 1771 version 4. RFC 4271 version corrected a number of errors, clarified ambiguities and brought the RFC much closer to industry practices.

Most Internet service providers must use BGP to establish routing between one another (especially if they are multihomed). Therefore, even though most Internet users do not use it directly, BGP is one of the most important protocols of the Internet. Compare this with Signaling System 7 (SS7), which is the inter-provider core call setup protocol on the PSTN. Very large private IP networks use BGP internally. An example would be the joining of a number of large OSPF (Open Shortest Path First) networks where OSPF by itself would not scale to size. Another reason to use BGP is multihoming a network for better redundancy, either to multiple access points of a single ISP (RFC 1998), or to multiple ISPs.

Operation

BGP neighbors, called peers, are established by manual configuration between routers to create a TCP session on port 179. A BGP speaker will periodically (every 30 seconds) send 19-byte keep-alive messages to maintain the connection.[1] Among routing protocols, BGP is unique in using TCP as its transport protocol.

When BGP runs between two peers in the same autonomous system (AS), it is referred to as *Internal BGP* (*IBGP* or *Interior Border Gateway Protocol*). When it runs between autonomous systems, it is called *External BGP* (*EBGP* or *Exterior Border Gateway Protocol*). Routers on the boundary of one AS exchanging information with another AS are called border or edge routers. In the Cisco operating system, IBGP routes have an administrative distance of 200 and that of EBGP is 20; IBGP is thus less preferred than either external BGP or any interior routing protocol. Other router implementations also prefer EBGP to IGPs, and IGPs to IBGP.

Extensions negotiation

During the OPEN, BGP speakers can negotiate[2] optional capabilities of the session, including multiprotocol extensions and various recovery modes. If the multiprotocol extensions to BGP[3] are negotiated at the time of creation, the BGP speaker can prefix the Network Layer Reach-ability Information (NLRI) it advertises with an address family prefix. These families include the IPv4 (default), IPv6, IPv4/IPv6 Virtual Private Networks and multicast BGP. Increasingly, BGP is used as a generalized signaling protocol to carry information about routes that may not be part of the global Internet, such as VPNs.[4]

Finite-state machine

BGP state machine

In order to make decisions in its operations with peers, a BGP peer uses a simple finite state machine (FSM) that consists of six states: Idle; Connect; Active; OpenSent; OpenConfirm; and Established. For each peer-to-peer session, a BGP implementation maintains a state variable that tracks which of these six states the session is in. The BGP protocol defines the messages that each peer should exchange in order to change the session from one state to another. The first state is the “Idle” state. In the “Idle” state, BGP initializes all resources, refuses all inbound BGP connection attempts and initiates a TCP connection to the peer. The second state is “Connect”. In the “Connect” state, the router waits for the TCP connection to complete and transitions to the "OpenSent" state if successful. If unsuccessful, it starts the ConnectRetry timer and transitions to the "Active" state upon expiration. In the "Active" state, the router resets the ConnectRetry timer to zero and returns to the "Connect" state. In the "OpenSent" state, the router sends an Open message and waits for one in return in order to transition to the "OpenConfirm" state. Keepalive messages are exchanged and, upon successful receipt, the router is placed into the “Established” state. In the “Established” state, the router can send/receive: Keepalive; Update; and Notification messages to/from its peer.

- **Idle State**:
  - Refuse all incoming BGP connections
  - Start the initialization of event triggers.
  - Initiates a TCP connection with its configured BGP peer.
  - Listens for a TCP connection from its peer.
  - Changes its state to Connect.
  - If an error occurs at any state of the FSM process, the BGP session is terminated immediately and returned to the Idle state. Some of the reasons why a router does not progress from the Idle state are:
    - TCP port 179 is not open.
    - A random TCP port over 1023 is not open.
    - Peer address configured incorrectly on either router.
    - AS number configured incorrectly on either router .
- **Connect State**:
  - Waits for successful TCP negotiation with peer.
  - BGP does not spend much time in this state if the TCP session has been successfully established.
  - Sends Open message to peer and changes state to OpenSent.
  - If an error occurs, BGP moves to the Active state. Some reasons for the error are:
    - TCP port 179 is not open.
    - A random TCP port over 1023 is not open.
    - Peer address configured incorrectly on either router.
    - AS number configured incorrectly on either router.
- **Active State**:

- If the router was unable to establish a successful TCP session, then it ends up in the Active state.
- BGP FSM will try to restart another TCP session with the peer and, if successful, then it will send an Open message to the peer.
- If it is unsuccessful again, the FSM is reset to the Idle state.
- Repeated failures may result in a router cycling between the Idle and Active states. Some of the reasons for this include:
  - TCP port 179 is not open.
  - A random TCP port over 1023 is not open.
  - BGP configuration error.
  - Network congestion.
  - Flapping network interface.
- **OpenSent State**:
  - BGP FSM listens for an Open message from its peer.
  - Once the message has been received, the router checks the validity of the Open message.
  - If there is an error it is because one of the fields in the Open message doesn't match between the peers, e.g. BGP version mismatch, MD5 password mismatch, the peering router expects a different My AS. The router will then send a Notification message to the peer indicating why the error occurred.
  - If there is no error, a Keepalive message is sent, various timers are set and the state is changed to OpenConfirm.
- **OpenConfirm State**:
  - The peer is listening for a Keepalive message from its peer.
  - If a Keepalive message is received and no timer has expired before reception of the Keepalive, BGP transitions to the Established state.
  - If a timer expires before a Keepalive message is received, or if an error condition occurs, the router transitions back to the Idle state.
- **Established State**:
  - In this state, the peers send Update messages to exchange information about each route being advertised to the BGP peer.
  - If there is any error in the Update message then a Notification message is sent to the peer, and BGP transitions back to the Idle state.
  - If a timer expires before a Keepalive message is received, or if an error condition occurs, the router transitions back to the Idle state.

BGP router connectivity and learning routes

In the simplest arrangement all routers within a single AS and participating in BGP routing must be configured in a full mesh: each router must be configured as peer to every other router. This causes scaling problems, since the number of required connections grows quadratically with the number of routers involved. To alleviate the problem, BGP implements two options: route reflectors (RFC 4456) and confederations (RFC 5065). The following discussion of basic UPDATE processing assumes a full IBGP mesh.

Basic update processing

A given BGP router may accept NLRI UPDATEs from multiple neighbors and advertise NLRI (Network Layer Reach-ability Information) to the same, or a different set, of neighbors. Conceptually, BGP maintains its own "master" routing table, called the *Loc-RIB* (Local Routing Information Base), separate from the main routing table of the router. For each neighbor, the BGP process maintains a conceptual *Adj-RIB-In* (Adjacent Routing Information Base, Incoming) containing the NLRI received from the neighbor, and a conceptual *Adj-RIB-Out* (Outgoing) for NLRI to be sent to the neighbor.

Conceptual, in the preceding paragraph, means that the physical storage and structure of these various tables are decided by the implementer of the BGP code. Their structure is not visible to other BGP routers, although they usually can be interrogated with management commands on the local router. It is quite common, for example, to store the two Adj-RIBs and the Loc-RIB together in the same data structure, with additional information attached to the RIB entries. The additional information tells the BGP process such things as whether individual entries belong in the Adj-RIBs for specific neighbors, whether the per-neighbor route selection process made received policies eligible for the Loc-RIB, and whether Loc-RIB entries are eligible to be submitted to the local router's routing table management process.

By *eligible to be submitted*, BGP will submit the routes that it considers best to the main routing table process. Depending on the implementation of that process, the BGP route is not necessarily selected. For example, a directly connected prefix, learned from the router's own hardware, is usually most preferred. As long as that directly connected route's interface is active, the BGP route to the destination will not be put into the routing table. Once the interface goes down, and there are no more preferred routes, the Loc-RIB route would be installed in the main routing table. Until recently, it was a common mistake to say *BGP carries policies*. BGP actually carried the information with which rules inside BGP-speaking routers could make policy decisions. Some of the information carried that is explicitly intended to be used in policy decisions are communities and multi-exit discriminators (MED).

Route selection

The BGP standard specifies a number of decision factors, more than are used by any other common routing process, for selecting NLRI (Network Layer Reach-ability Information) to go into the Loc-RIB (Routing Information Base). The first decision point for evaluating NLRI is that its next-hop attribute must be reachable (or resolvable). Another way of saying the next-hop must be reachable is that there must be an active route, already in the main routing table of the router, to the prefix in which the next-hop address is located.

Next, for each neighbor, the BGP process applies various standard and implementation-dependent criteria to decide which routes conceptually should go into the Adj-RIB-In. The neighbor could send several possible routes to a destination, but the first level of preference is at the neighbor level. Only one route to each destination will be installed in the conceptual Adj-RIB-In. This process will also delete, from the Adj-RIB-In, any routes that are withdrawn by the neighbor.

Whenever a conceptual Adj-RIB-In changes, the main BGP process decides if any of the neighbor's new routes are preferred to routes already in the Loc-RIB. If so, it replaces them. If a given route is withdrawn by a neighbor, and there is no other route to that destination, the route is removed from the Loc-RIB, and no longer sent, by BGP, to the main routing table manager. If the router does not have a route to that destination from any non-BGP source, the withdrawn route will be removed from the main routing table.

Per-neighbor decisions

After verifying that the next hop is reachable, if the route comes from an internal (i.e. IBGP) peer, the first rule to apply according to the standard is to examine the LOCAL_PREFERENCE attribute. If there are several IBGP routes from the neighbor, the one with the highest LOCAL_PREFERENCE is selected unless there are several routes with the same LOCAL_PREFERENCE. In the latter case the route selection process moves to the next tie breaker. While LOCAL_PREFERENCE is the first rule in the standard, once reach-ability of the NEXT_HOP is verified, Cisco and several other vendors first consider a decision factor called WEIGHT which is local to the router (i.e. not transmitted by BGP). The route with the highest WEIGHT is preferred.

The LOCAL_PREFERENCE, WEIGHT, and other criteria can be manipulated by local configuration and software capabilities. Such manipulation is outside the scope of the standard but is commonly used. For example the COMMUNITY attribute (see below) is not directly used by the BGP selection process. The BGP neighbor process however can have a rule to set LOCAL_PREFERENCE or another factor based on a manually programmed rule to set the attribute if the COMMUNITY value matches some pattern matching criterion. If the route was learned from an external peer the per-neighbor BGP process computes a LOCAL_PREFERENCE value from local policy rules and then compares the LOCAL_PREFERENCE of all routes from the neighbor.

At the per-neighbor level - ignoring implementation-specific policy modifiers - the order of tie breaking rules is:

1. Prefer the route with the shortest AS_PATH. An AS_PATH is the set of AS numbers that must be traversed to reach the advertised destination. AS1-AS2-AS3 is shorter than AS4-AS5-AS6-AS7.
2. Prefer routes with the lowest value of their ORIGIN attribute.
3. Prefer routes with the lowest MULTI_EXIT_DISC (multi-exit discriminator or MED) value.

Before the most recent edition of the BGP standard, if an UPDATE had no MULTI_EXIT_DISC value, several implementations created a MED with the least possible value. The current standard however specifies that missing MEDs are to be treated as the highest possible value. Since the current rule may cause different behavior than the vendor interpretations, BGP implementations that used the nonstandard default value have a configuration feature that allows the old or standard rule to be selected.

Decision factors at the Loc-RIB level

Once candidate routes are received from neighbors, the Loc-RIB software applies additional tie-breakers to routes to the same destination.

1. If at least one route was learned from an external neighbor (i.e., the route was learned from EBGP), drop all routes learned from IBGP.
2. Prefer the route with the lowest interior cost to the NEXT_HOP, according to the main Routing Table. If two neighbors advertised the same route, but one neighbor is reachable via a low-bitrate link and the other by a high-bitrate link, and the interior routing protocol calculates lowest cost based on highest bitrate, the route through the high-bitrate link would be preferred and other routes dropped.

If there is more than one route still tied at this point, several BGP implementations offer a configurable option to load-share among t

Communities

BGP communities are attribute tags that can be applied to incoming or outgoing prefixes to achieve some common goal (RFC 1997). While it is common to say that BGP allows an administrator to set policies on how prefixes are handled by ISPs, this is generally not possible, strictly speaking. For instance, BGP natively has no concept to allow one AS to tell another AS to restrict advertisement of a prefix to only North American peering customers. Instead, an ISP generally publishes a list of well-known or proprietary communities with a description for each one, which essentially becomes an agreement of how prefixes are to be treated. Examples of common communities include local preference adjustments, geographic or peer type restrictions, DoS avoidance (black holing), and AS prepending

options. An ISP might state that any routes received from customers with community XXX:500 will be advertised to all peers (default) while community XXX:501 will restrict advertisement to North America. The customer simply adjusts their configuration to include the correct community(ies) for each route, and the ISP is responsible for controlling who the prefix is advertised to. The end user has no technical ability to enforce correct actions being taken by the ISP, though problems in this area are generally rare and accidental.

It is a common tactic for end customers to use BGP communities (usually ASN:70,80,90,100) to control the local preference the ISP assigns to advertised routes instead of using MED (the effect is similar). It should also be noted that the community attribute is transitive, but communities applied by the customer very rarely become propagated outside the next-hop AS.

Extended communities

The BGP Extended Community Attribute was added in 2006 in order to extend the range of such attributes and to provide a community attribute structuring by means of a type field. The extended format consists of one or two octets for the type field followed by seven or six octets for the respective community attribute content. The definition of this Extended Community Attribute is documented in RFC 4360. The IANA administers the registry for BGP Extended Communities Types.[5] The Extended Communities Attribute itself is a transitive optional BGP attribute. However, a bit in the type field within the attribute decides whether the encoded extended community is of a transitive or non-transitive nature. The IANA registry therefore provides different number ranges for the attribute types. Due to the extended attribute range, its usage can be manifold. RFC 4360 exemplarly defines the "Two-Octet AS Specific Extended Community", the "IPv4 Address Specific Extended Community", the "Opaque Extended Community", the "Route Target Community" and the "Route Origin Community". A number of BGP QoS drafts[6] also use this Extended Community Attribute structure for inter-domain QoS signalling.

Uses of multi-exit discriminators

MEDs, defined in the main BGP standard, were originally intended to show to another neighbor AS the advertising AS's preference as to which of several links are preferred for inbound traffic. Another application of MEDs is to advertise the value, typically based on delay, of multiple AS that have presence at an IXP, that they impose to send traffic to some destination.

Multipath BGP

An extension to BGP is the use of Multipathing - this typically requires identical MED, weight, origin and AS-path although some implementations provide the ability to relax the AS-path checking to only expect an equal path length rather than the actual AS numbers in the path being expected to match too. This can then be extended further with features like Cisco's dmzlink-bw which enables a ratio of traffic sharing based on bandwidth values configured on individual links.

Message header format

The following is the BGP version 4 message header format:

bit offset	0–15	16–23	24–31
0	Marker		
32			
64			
96			
128	Length	Type	

- **Marker**: Included for compatibility, must be set to all ones.
- **Length**: Total length of the message in octets, including the header.
- **Type**: Type of BGP message. The following values are defined:
  - Open (1)
  - Update (2)
  - Notification (3)
  - KeepAlive (4)
  - Route-Refresh (5)

BGP problems and mitigation

Internal BGP scalability

An autonomous system with internal BGP (IBGP) must have all of its IBGP peers connect to each other in a full mesh (where everyone speaks to everyone directly). This full-mesh configuration requires that each router maintain a session to every other router. In large networks, this number of sessions may degrade performance of routers, due either to a lack of memory, or too much CPU process requirements.

Route reflectors and confederations both reduce the number of IBGP peers to each router and thus reduce processing overhead. Route reflectors are a pure performance-enhancing technique, while confederations also can be used to implement more fine-grained policy.

Route reflectors[7] reduce the number of connections required in an AS. A single router (or two for redundancy) can be made a route reflector: other routers in the AS need only be configured as peers to them.

Confederations are sets of autonomous systems. In common practice,[8] only one of the confederation AS numbers is seen by the Internet as a whole. Confederations are used in very large networks where a large AS can be configured to encompass smaller more manageable internal ASs.

Confederations can be used in conjunction with route reflectors. Both confederations and route reflectors can be subject to persistent oscillation unless specific design rules, affecting both BGP and the interior routing protocol, are followed.[9]

However, these alternatives can introduce problems of their own, including the following:

- route oscillation
- sub-optimal routing
- increase of BGP convergence time[10]

Additionally, route reflectors and BGP confederations were not designed to ease BGP router configuration. Nevertheless, these are common tools for experienced BGP network architects. These tools may be combined, for example, as a hierarchy of route reflectors.

Instability

The routing tables managed by a BGP implementation are adjusted continually to reflect actual changes in the network, such as links breaking and being restored or routers going down and coming back up. In the network as a whole it is normal for these changes to happen almost continuously, but for any particular router or link changes are supposed to be relatively infrequent. If a router is misconfigured or mismanaged then it may get into a rapid cycle between down and up states. This pattern of repeated withdrawal and re-announcement known as route flapping can cause excessive activity in all the other routers that know about the broken link, as the same route is continuously injected and withdrawn from the routing tables. The BGP design is such that delivery of traffic may not function while routes are being updated. On the Internet, a BGP routing change may cause outages for several minutes.

A feature known as *route flap damping* (RFC 2439 [11]) is built into many BGP implementations in an attempt to mitigate the effects of route flapping. Without damping the excessive activity can cause a heavy processing load on routers, which may in turn delay updates on other routes, and so affect overall routing stability. With damping, a route's flapping is exponentially decayed. At the first instance when a route becomes unavailable and quickly reappears, damping does not take effect, so as to maintain the normal fail-over times of BGP. At the second occurrence, BGP shuns that prefix for a certain length of time; subsequent occurrences are timed out exponentially. After the abnormalities have ceased and a suitable length of time has passed for the offending route, prefixes can be reinstated and its slate wiped clean. Damping can also mitigate denial of service attacks; damping timings are highly customizable.

It is also suggested in RFC 2439 (under "Design Choices -> Stability Sensitive Suppression of Route Advertisement") that route flap dampening is a feature more desirable if implemented to Exterior Border Gateway Protocol Sessions (EBGP sessions or simply called exterior peers) and not on Interior Border Gateway Protocol Sessions (IBGP sessions or simply called internal peers); With this approach when a route flaps inside an autonomous system, it is not propagated to the external ASs - flapping a route to an EBGP will have a chain of flapping for the particular route throughout the backbone. This method also successfully avoids the overhead of route flap dampening for IBGP sessions.

However, subsequent research has shown that flap damping can actually lengthen convergence times in some cases, and can cause interruptions in connectivity even when links are not flapping.[12][13] Moreover, as backbone links and router processors have become faster, some network architects have suggested that flap damping may not be as important as it used to be, since changes to the routing table can be absorbed much faster by routers. This has led the RIPE Route Working Group to write that "with the current implementations of BGP flap damping, the application of flap damping in ISP networks is NOT recommended. ... If flap damping is implemented, the ISP operating that network will cause side-effects to their customers and the Internet users of their customers' content and services ... . These side-effects would quite likely be worse than the impact caused by simply not running flap damping at all."[14] Improving stability without the problems of flap damping is the subject of current research.[15]

Routing table growth

One of the largest problems faced by BGP, and indeed the Internet infrastructure as a whole, is the growth of the Internet routing table. If the global routing table grows to the point where some older, less capable, routers cannot cope with the memory requirements or the CPU load of maintaining the table, these routers will cease to be effective gateways between the parts of the Internet they connect. In addition, and perhaps even more importantly, larger routing tables take longer to stabilize (see above) after a major connectivity change, leaving network service unreliable, or even unavailable, in the interim.

BGP table growth on the Internet.

Until late 2001, the global routing table was growing exponentially, threatening an eventual widespread breakdown of connectivity. In an attempt to prevent this, ISPs cooperated in keeping the global routing table as small as possible, by using Classless Inter-Domain Routing (CIDR) and route aggregation. While this slowed the growth of the routing table to a linear process for several years, with the expanded demand for multihoming by end user networks the growth was once again superlinear by the middle of 2004. A full IPv4 BGP table as of September 2012 is in excess of 430,000 prefixes.[16]

Number of AS on the Internet vs number of registered AS.

Route summarization is often used to improve aggregation of the BGP global routing table, thereby reducing the necessary table size in routers of an AS. Consider AS1 has been allocated the big address space of 172.16.0.0/16, this would be counted as one route in the table, but due to customer requirement or traffic engineering purposes, AS1 wants to announce smaller, more specific routes of 172.16.0.0/18, 172.16.64.0/18 and 172.16.128.0/18. The prefix 172.16.192.0/18 does not have any hosts so AS1 does not announce a specific route 172.16.192.0/18. This all counts as AS1 announcing four routes.

AS2 will see the 4 routes from AS1 (172.16.0.0/16, 172.16.0.0/18, 172.16.64.0/18 and 172.16.128.0/18) and it is up to the routing policy of AS2 to decide whether or not to take a copy of the four routes or, as 172.16.0.0/16 overlaps all the other specific routes, to just store the summary, 172.16.0.0/16.

If AS2 wants to send data to prefix 172.16.192.0/18, it will be sent to the routers of AS1 on route 172.16.0.0/16. At AS1's router, it will either be dropped or a destination unreachable ICMP message will be sent back, depending on the configuration of AS1's routers.

If AS1 later decides to drop the route 172.16.0.0/16, leaving 172.16.0.0/18, 172.16.64.0/18 and 172.16.128.0/18, AS1 will drop the number of routes it announces to three. AS2 will see the three routes, and depending on the routing policy of AS2, it will store a copy of the three routes, or aggregate the prefix's 172.16.0.0/18 and 172.16.64.0/18 to 172.16.0.0/17, thereby reducing the number of routes AS2 stores to only two: 172.16.0.0/17 and 172.16.128.0/18.

If AS2 wants to send data to prefix 172.16.192.0/18, it will be dropped or a destination unreachable ICMP message will be sent back at the routers of AS2 (not AS1 as before), because 172.16.192.0/18 would not be in the routing table.

Load-balancing problem

Another factor causing this growth of the routing table is the need for load balancing of multi-homed networks. It is not a trivial task to balance the inbound traffic to a multi-homed network across its multiple inbound paths, due to limitation of the BGP route selection process. For a multi-homed network, if it announces the same network blocks across all of its BGP peers, the result may be that one or several of its inbound links become congested while the other links remain under-utilized, because external networks all picked that set of congested paths as optimal. Like most other routing protocols, the BGP protocol does not detect congestion.

To work around this problem, BGP administrators of that multihomed network may divide a large continuous IP address block into smaller blocks, and tweak the route announcement to make different blocks look optimal on different paths, so that external networks will choose a different path to reach different blocks of that multi-homed network. Such cases will increase the number of routes as seen on the global BGP table.

One solution growing in popularity to address the load balancing issue is to deploy BGP/LISP (Locator/Identifier Separation Protocol) gateways within an Internet exchange point to allow ingress traffic engineering across multiple links. This technique does not increase the number of routes seen on the global BGP table.

IP Hijacking

IP hijacking (sometimes referred to as BGP hijacking, prefix hijacking or route hijacking) is the illegitimate take over of groups of IP addresses by corrupting Internet routing tables. This can be caused both by programming errors, attacks on a network or attempts at censorship. This will cause internet sites to go out of service, as routers will no longer find them.

Requirements of a router for use of BGP for Internet and backbone-of-backbones purposes

Routers, especially small ones intended for Small Office/Home Office (SOHO) use, may not include BGP software. Some SOHO routers simply are not capable of running BGP using BGP routing tables of any size. Other commercial routers may need a specific software executable image that contains BGP, or a license that enables it. Open source packages that run BGP include GNU Zebra, Quagga, OpenBGPD, BIRD, XORP and Vyatta. Devices marketed as Layer 3 switches are less likely to support BGP than devices marketed as routers, but high-end Layer 3 Switches usually can run BGP.

Products marketed as switches may or may not have a size limitation on BGP tables, such as 20,000 routes, far smaller than a full Internet table plus internal routes. These devices, however, may be perfectly reasonable and useful when used for BGP routing of some smaller part of the network, such as a confederation-AS representing one of several smaller enterprises that are linked, by a BGP backbone of backbones, or a small enterprise that announces routes to an ISP but only accepts a default route and perhaps a small number of aggregated routes.

A BGP router used only for a network with a single point of entry to the Internet may have a much smaller routing table size (and hence RAM and CPU requirement) than a multihomed network. Even simple multihoming can have modest routing table size. See RFC 4098 for vendor-independent performance parameters for single BGP router convergence in the control plane. The actual amount of memory required in a BGP router depends on the amount of BGP information exchanged with other BGP speakers, and the way in which the particular router stores BGP information. The router may have to keep more than one copy of a route, so it can manage different policies for route advertising and acceptance to a specific neighboring AS. The term view is often used for these different policy relationships on a running router.

If one router implementation takes more memory per route than another implementation, this may be a legitimate design choice, trading processing speed against memory. A full IPv4 BGP table as of September 2012 is in excess of 430,000 prefixes.[17] Large ISPs may add another 50% for internal and customer routes. Again depending on

implementation, separate tables may be kept for each view of a different peer AS.

Free and open source implementations

- Bird Internet routing daemon, a GPL routing package for Unix-like systems.
- GNU Zebra, a GPL routing suite supporting BGP4.
- OpenBGPD, a BSD licensed implementation by the OpenBSD team.
- Quagga, a fork of GNU Zebra for Unix-like systems.
- XORP, the eXtensible Open Router Platform, a BSD licensed suite of routing protocols.
- VNE [18], a C# software library implementing BGP

Simulators

- BGPviz [19], a Flash application that presents a graphical visualization of BGP routes and updates for any real AS on the Internet
- SSFnet [20], SSFnet network simulator includes a BGP implementation developed by BJ Premore
- C-BGP [21], a BGP simulator able to perform large-scale simulation trying to model the ASes of the Internet or modelling ASes as large as Tier-1.[22]
- BGP++ [23], a patch integrating GNU Zebra software on ns-2 and GTNetS network simulators
- ns-BGP [24], a BGP extension for ns-2 simulator based on the SSFnet implementation
- NetViews [25], a Java application that monitors and visualizes BGP activity in real time.

Test equipment

Systems for testing BGP conformance, load or stress performance come from vendors such as:

- Ixia
- Spirent Communications
- Agilent Technologies

References

[1] RFC4274

[2] Capabilities Advertisement with BGP-4 (http://www.ietf.org/rfc/rfc2842.txt), RFC 2842, R. Chandra & J. Scudder,May 2000

[3] Multiprotocol Extensions for BGP-4 (http://www.ietf.org/rfc/rfc2858.txt), RFC 2858, T. Bates *et al.*,June 2000

[4] BGP/MPLS VPNs. (http://www.ietf.org/rfc/rfc2547.txt), RFC 2547, E. Rosen and Y. Rekhter,April 2004

[5] IANA registry for BGP Extended Communities Types (http://www.iana.org/assignments/bgp-extended-communities), IANA,2008

[6] IETF drafts on BGP signalled QoS (http://www.bgp-qos.org/forum/viewforum.php?f=6), Thomas Knoll,2008

[7] BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP) (http://www.ietf.org/rfc/rfc4456.txt), RFC 4456, T. Bates *et al.*, April 2006

[8] Autonomous System Confederations for BGP (http://www.ietf.org/rfc/rfc5065.txt), RFC 5065, P. Traina *et al.*, February 2001

[9] Border Gateway Protocol (BGP) Persistent Route Oscillation Condition (http://www.ietf.org/rfc/rfc3345.txt), RFC 3345, D. McPherson *et al.*, August 2002

[10] Terminology for Benchmarking BGP Device Convergence in the Control Plane (http://www.ietf.org/rfc/rfc4098.txt), RFC 4098, H. Berkowitz *et al.*, June 2005

[11] http://www.ietf.org/rfc/rfc2439.txt

[12] Route Flap Damping Exacerbates Internet Routing Convergence (http://www.sigcomm.org/sigcomm2002/papers/routedampening.pdf)

[13] Zhang, Beichuan; Pei Dan, Daniel Massey, Lixia Zhang (June 2005). "Timer Interaction in Route Flap Damping" (http://www.cs.arizona.edu/~bzhang/paper/05-icdcs-dtimer.pdf). *IEEE 25th International Conference on Distributed Computing Systems*. . Retrieved 2006-09-26. "We show that the current damping design leads to the intended behavior only under persistent route flapping. When the number of flaps is small, the global routing dynamics deviates significantly from the expected behavior with a longer convergence delay."

[14] (http://www.ripe.net/docs/routeflap-damping.html)

[15] http://tools.ietf.org/internet-drafts/draft-li-bgp-stability-01.txt

[16] Potaroo - BGP Table data (http://bgp.potaroo.net/index-bgp.html)

[17] Potaroo - BGP Table data (http://bgp.potaroo.net/index-bgp.html)

[18] http://wiki.ucis.nl/VNE
[19] http://www.ris.ripe.net/bgpviz/
[20] http://www.ssfnet.org/homePage.html
[21] http://cbgp.info.ucl.ac.be/
[22] Quoitin, Bruno; Steve Uhlig (November 2005). "Modeling the Routing of an Autonomous System with C-BGP" (http://inl.info.ucl.ac.be/publications/modeling-routing-autonomous-system-c-). *IEEE Network Magazine* **19** (6). .
[23] http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/
[24] http://www.ensc.sfu.ca/~ljilja/cnl/projects/BGP/
[25] http://netlab.cs.memphis.edu/projects_netviews.html

Further reading

- Chapter "Border Gateway Protocol (BGP)" (http://docwiki.cisco.com/wiki/Border_Gateway_Protocol) in the Cisco "IOS Technology Handbook"

External links

- Cyclops (http://cyclops.cs.ucla.edu/) A BGP network audit tool (prefix hijack, route leakage) by UCLA
- Codenomicon Defensics for BGP (http://www.codenomicon.com/products/bgp.shtml), a test automation framework for BGP Fuzzing and robustness testing
- LinkRank (http://linkrank.cs.ucla.edu/) A tool for BGP routing visualization by University of California, Los Angeles
- A look at fuzzing BGP for security testing (http://www.breakingpointsystems.com/community/blog/protocolfuzzingBGP)
- BGP Routing Resources (http://www.bgp4.as/) (includes a dedicated section on BGP & ISP Core Security (http://www.bgp4.as/security))
- BGP table statistics (http://bgp.potaroo.net/)
- bgpTables (http://bgptables.merit.edu/) A global BGP visibility analysis tool by Merit Networks
- ASNumber Firefox Extension (http://www.asnumber.networx.ch/) showing the AS number and additional information of the website currently open
- RIPE Routing Information Service (http://www.ripe.net/ris/index.html) collecting over 550 IPv4 and IPv6 BGP feeds at 14 sites around the world
- RIS Looking Glass (http://www.ris.ripe.net/cgi-bin/lg/index.cgi) into the Default Free Routing zone of the Internet
- RISwhois (http://www.ripe.net/projects/ris/tools/riswhois.html) providing IPv4/IPv6 Address to BGP AS Origin Mapping
- RIS BGPlay (http://www.ris.ripe.net/bgplay/) BGP routing visualization tool by Università degli Studi Roma Tre
- Linux Magazine: Demystifying BGP (http://www.linux-mag.com/id/1353/) (Good, Detailed BGP explanation; requires registration)
- BGP config generator (http://www.oriontechnologysolutions.com/administration/basic-bgp-config-generator/) Free web-based tool to generate a Cisco/Quagga BGP configuration
- Some important BGP RFCs
  - RFC 4271, A Border Gateway Protocol 4 (BGP-4)
  - RFC 4456, BGP Route Reflection - An Alternative to Full Mesh Internal BGP (IBGP)
  - RFC 4278, Standards Maturity Variance Regarding the TCP MD5 Signature Option (RFC 2385) and the BGP-4 Specification
  - RFC 4277, Experience with the BGP-4 Protocol
  - RFC 4276, BGP-4 Implementation Report
  - RFC 4275, BGP-4 MIB Implementation Survey

  - RFC 4274, BGP-4 Protocol Analysis
  - RFC 4273, Definitions of Managed Objects for BGP-4
  - RFC 4272, BGP Security Vulnerabilities Analysis
  - RFC 5492, Capabilities Advertisement with BGP-4
  - RFC 5065, Autonomous System Confederations for BGP
  - RFC 2918, Route Refresh Capability for BGP-4
  - RFC 1772, Application of the Border Gateway Protocol in the Internet Protocol (BGP-4) using SMIv2
  - RFC 4893, BGP Support for Four-octet AS Number Space
  - RFC 2439, BGP Route Flap Damping
  - RFC 4760, Multiprotocol Extensions for BGP-4
- Obsolete RFCs
  - RFC 3392, Obsolete - Capabilities Advertisement with BGP-4
  - RFC 2796, Obsolete - BGP Route Reflection - An Alternative to Full Mesh IBGP
  - RFC 3065, Obsolete - Autonomous System Confederations for BGP
  - RFC 1965, Obsolete - Autonomous System Confederations for BGP
  - RFC 1771, Obsolete - A Border Gateway Protocol 4 (BGP-4)
  - RFC 1657, Obsolete - Definitions of Managed Objects for the Fourth Version of the Border Gateway
  - RFC 1655, Obsolete - Application of the Border Gateway Protocol in the Internet
  - RFC 1654, Obsolete - A Border Gateway Protocol 4 (BGP-4)
  - RFC 1105, Obsolete - Border Gateway Protocol (BGP)
  - RFC 2858, Obsolete - Multiprotocol Extensions for BGP-4
- BGP Interactions at Router Startup Described as a Sequence Diagram (http://www.eventhelix.com/RealtimeMantra/Networking/bgp_startup.pdf) (PDF)
- BGP Protocol Service Level Traffic Variations [[Mu Dynamics]] (http://www.mudynamics.com/products/protocol.html)]
- BGP Route Reflection Troubleshooting (http://blog.ipexpert.com/2009/12/21/bgp-route-reflection-â-troubleshooting/)

Request for Comments

In computer network engineering, a **Request for Comments** (**RFC**) is a memorandum published by the Internet Engineering Task Force (IETF) describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems.

Through the Internet Society, engineers and computer scientists may publish discourse in the form of an RFC, either for peer review or simply to convey new concepts, information, or (occasionally) engineering humor. The IETF adopts some of the proposals published as RFCs as Internet standards.

Request For Comments documents were invented by Steve Crocker in 1969 to help record unofficial notes on the development of the ARPANET. They have since become the official record for Internet specifications, protocols, procedures, and events.[1]

History

The inception of the RFC format occurred in 1969 as part of the seminal ARPANET project.[2] Today, it is the official publication channel for the Internet Engineering Task Force (IETF), the Internet Architecture Board (IAB), and—to some extent—the global community of computer network researchers in general.

The authors of the first RFCs typewrote their work and circulated hard copies among the ARPA researchers. Unlike the modern RFCs, many of the early RFCs were actual requests for comments. The RFC leaves questions open and is written in a less formal style. This less formal style is now typical of Internet Draft documents, the precursor step before being approved as an RFC.

In December 1969, researchers began distributing new RFCs via the newly operational ARPANET. RFC 1, entitled "Host Software", was written by Steve Crocker of the University of California, Los Angeles (UCLA), and published on April 7, 1969. Although written by Steve Crocker, the RFC emerged from an early working group discussion between Steve Crocker, Steve Carr and Jeff Rulifson.

In RFC 3, which first defined the RFC series, Crocker started attributing the RFC series to the "Network Working Group". This wasn't so much a formal committee as a loose association of researchers interested in the ARPANET project. In effect, it was anyone who wanted to join in on meetings and discussions about the project.

Many of the subsequent RFCs of the 1970s also came from UCLA, because UCLA was one of the first Interface Message Processors (IMPs) on ARPANET. The Augmentation Research Center (ARC) at Stanford Research Institute, directed by Douglas Engelbart, was another of the four first ARPANET nodes and the source of early RFCs. The ARC became the first Network Information Center, which was managed by Elizabeth J. Feinler to distribute them along with other network information. From 1969 until 1998, Jon Postel served as the RFC editor. (On his death in 1998, his obituary was published as RFC 2468.)

Following the expiration of the original ARPANET contract with the U.S. federal government, the Internet Society (acting on behalf of the IETF) contracted with the Networking Division of the University of Southern California (USC) Information Sciences Institute (ISI) to assume the editorship and publishing responsibilities (under the direction of the IAB). Sandy Ginoza joined USC/ISI in 1999 to work on RFC editing, and Alice Hagens in 2005.[3] Bob Braden took over the role of RFC project lead, while Joyce K. Reynolds continued to be part of the team until October 13, 2006.

In July 2007, "streams" of RFCs were defined, so the editing duties could be divided. IETF documents came from IETF working groups or submissions sponsored by an IETF area director from the Internet Engineering Steering Group. The IAB can publish its own documents. A research stream of documents comes from the Internet Research Task Force (IRTF), and independent stream includes documents from other outside sources.[4] A new model was proposed in 2008, refined, and published in August 2009, splitting the task into several roles.[5] The streams were also refined in December 2009, with standards defined for their style.[6] In January 2010 the RFC editor function was

moved to a contractor, Association Management Solutions with Glenn Kowack serving as interim series editor.[7]

Production and evolution

The *RFC Editor* assigns each RFC a unique serial number. Once assigned a number and published, an RFC is never rescinded or modified; if the document requires amendments, the authors publish a revised document. Therefore, some RFCs supersede others; the superseded RFCs are said to be *deprecated*, *obsolete*, or *obsoleted by* the superseding RFC. Together, the serialized RFCs compose a continuous historical record of the evolution of Internet standards and practices. For more details about RFCs and the RFC process, see RFC 2026, "The Internet Standards Process, Revision 3".[8]

The RFC production process differs from the standardization process of formal standards organizations such as ISO. Internet technology experts may submit an Internet Draft without support from an external institution. Standards-track RFCs are published with approval from the IETF, and are usually produced by experts participating in working groups, which first publish an Internet Draft. This approach facilitates initial rounds of peer review before documents mature into RFCs.

The RFC tradition of pragmatic, experience-driven, after-the-fact standards authorship accomplished by individuals or small working groups can have important advantages over the more formal, committee-driven process typical of ISO and national standards bodies.

Most RFCs use a common set of terms such as "MUST" and "NOT RECOMMENDED" (as defined by RFC 2119), Augmented Backus–Naur Form (ABNF) (as defined by RFC 5234) as a metalanguage, and simple text-based formatting, in order to keep the RFCs consistent and easy to understand.[8]

Sub-series

The RFC series contains three sub-series for IETF RFCs:

BCP

Best Current Practice; mandatory IETF RFCs not on standards track, see below.

FYI

For Your Information; informational RFCs promoted by the IETF as specified in RFC 1150 (FYI 1). In 2011 RFC 6360 obsoleted FYI 1 and concluded this sub-series.

STD

Standard; this used to be the third and highest maturity level of the IETF standards track specified in RFC 2026 (BCP 9). In 2011 RFC 6410 (a new part of BCP 9) reduced the standards track to two maturity levels.

Streams

There are four streams of RFCs: (1) IETF, (2) IRTF, (3) IAB, and (4) *independent submission*. Only the IETF creates BCPs and RFCs on standards track. An *independent submission* is checked by the IESG for conflicts with IETF work; the quality is assessed by an *independent submission editorial board*. In other words, IRTF and *independent* RFCs are supposed to contain relevant info or experiments for the Internet at large not in conflict with IETF work, compare RFC 4846, RFC 5742, and RFC 5744.

Obtaining RFCs

The official source for RFCs on the World Wide Web is the RFC Editor [9]. Almost any individual published RFC, for example RFC 5000, can be retrieved via the URL: http://www.rfc-editor.org/rfc/rfc5000.txt

Every RFC is submitted as plain ASCII text and is published in that form, but may also be available in other formats. However, as of 2008 the definitive version of any standards-track specification is the ASCII version.

For easy access to the metadata of an RFC, including abstract, keywords, author(s), publication date, errata, status, and especially later updates, the *RFC Editor* site offers a search form with many features. A redirection sets some efficient parameters, example: http://purl.net/net/rfc/5000 [10]

The official International Standard Serial Number (ISSN) of the RFC series is 2070-1721.[6]

Status

Not all RFCs are standards.[11] Each RFC is assigned a designation with regard to status within the Internet standardization process. This status is one of the following: *Informational*, *Experimental*, *Best Current Practice* (*BCP*), *Standards Track*, or *Historic* (sic). Standards-track documents are further divided into *Proposed Standard*, *Draft Standard*, and *Internet Standard* documents. The term *Historic* is applied to deprecated standards-track documents or obsolete RFCs that were published before the standards track was established. Only the IETF, represented by the Internet Engineering Steering Group (IESG), can approve standards-track RFCs.

Each RFC is static; if the document is changed, it is submitted again and assigned a new RFC number. If an RFC becomes an Internet Standard (STD), it is assigned an STD number but retains its RFC number; however, when an Internet Standard is updated, its number stays the same and it simply refers to a different RFC or set of RFCs. A given Internet Standard, STD *n*, may be RFCs *x* and *y* at a given time, but later the same standard may be updated to be RFC *z* instead. For example, in 2007 RFC 3700 was an Internet Standard—STD 1—and in May 2008 it was replaced with RFC 5000, so RFC 3700 changed to *Historic*, RFC 5000 became an Internet Standard, and as of May 2008 STD 1 is RFC 5000. When STD 1 is updated again, it will simply refer to a newer RFC that will have completed the standards track, but it will still be STD 1. Best Current Practices work in a similar fashion; BCP *n* refers to a certain RFC or set of RFCs, but which RFC or RFCs may change over time.

The definitive list of Internet Standards is itself an Internet Standard, STD 1: *Internet Official Protocol Standards*.[12]

Status "informational"

An *informational* RFC can be nearly anything from April 1 jokes over proprietary protocols up to widely recognized essential RFCs like Domain Name System Structure and Delegation (RFC 1591). Some informational RFCs formed the FYI sub-series. Some old FYIs are still interesting, e.g., the *Internet User's Glossary* RFC 1983 (FYI 18), and *The Tao of IETF* RFC 4677 (FYI 17).

Status "experimental"

An *experimental* RFC can be an IETF document or an individual submission to the 'RFC Editor'. A draft is designated experimental if it is unclear the proposal will work as intended or unclear if the proposal will be widely adopted. Experimental RFCs may be promoted to standards track if it becomes popular and works well.[13]

Status "best current practice"

The *best current practice* (BCP) subseries collects administrative documents and other texts which are considered as official rules and not only *informational*, but which do not affect *over the wire data*. The border between standards track and BCP is often unclear. If a document only affects the Internet Standards Process, like BCP 9,[14] or IETF administration, it is clearly a BCP. If it only defines rules and regulations for Internet Assigned Numbers Authority (IANA) registries it is less clear; most of these documents are BCPs, but some are on the standards track.

The BCP series also covers technical recommendations for how to practice Internet standards; for instance the recommendation to use source filtering to make DoS attacks more difficult (RFC 2827: "*Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*") is BCP 38 [15].

Status "historic"

A *historic* RFC is one that has been made obsolete by a newer version, documents a protocol that is not considered interesting in the current Internet, or has been removed from the standards track for other reasons. Some obsolete RFCs are not classified as historic, because the Internet standards process generally does not allow normative references from a standards track RFC to another RFC with lower status. Also, few are interested in working through the required procedural details to get RFCs classified as historic and update all RFCs normatively depending on it.

Status "unknown"

Status *unknown* is used for some very old RFCs, where it is unclear which status the document would get if it were published today. Some of these RFCs would not be published at all today; an early RFC was often just that: a simple request for comments, not intended to specify a protocol, administrative procedure, or anything else for which the RFC series is used today.

References

[1] "RFC's, Internet Request For Comments" (http://www.livinginternet.com/i/ia_rfc.htm). Livinginternet.com. . Retrieved 2012-04-03.

[2] "Stephen D. Crocker, "How the Internet Got Its Rules", The New York Times, 6 April 2009" (http://www.nytimes.com/2009/04/07/opinion/07crocker.html?_r=1&em). Nytimes.com. April 7, 2009. . Retrieved 2012-04-03.

[3] Leslie Daigle (March 2010). "RFC Editor in Transition: Past, Present, and Future" (http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_13-1/131_rfc.html). *The Internet Protocol Journal* (Cisco Systems) **13** (1). . Retrieved August 17, 2011.

[4] Leslie Diagle (July 2007). "The RFC Series and RFC Editor" (http://tools.ietf.org/html/rfc4844). *RFC 4844*. . Retrieved August 17, 2011.

[5] O. Kolkman, Ed (August 2009). "RFC Editor Model (Version 1)". *RFC 5620*.

[6] Leslie Diagle, Olaf Kolkman (December 2009). "RFC Streams, Headers, and Boilerplates" (http://tools.ietf.org/html/rfc5741). *RFC 5741*. . Retrieved August 17, 2011.

[7] Glenn Kowack (January7, 2010). "RFC Editor Transition Announcement" (http://www.ietf.org/ibin/c5i?mid=6&rid=49&gid=0&k1=934&k2=7324&tid=1263251951). . Retrieved August 7, 2011.

[8] "*RFC Index*" (http://www.rfc-editor.org/rfc-index2.html). RFC Editor. 2008-05-25. . Retrieved 2008-05-26.

[9] http://www.rfc-editor.org/rfc.html

[10] http://purl.net/net/rfc/5000

[11] Huitema, C.; Postel, J.; Crocker, S. (April 1995). "Not All RFCs are Standards (RFC 1796)" (http://tools.ietf.org/html/rfc1796). The Internet Engineering Task Force. . Retrieved 2008-05-19. "[E]ach RFC has a status…: Informational, Experimental, or Standards Track (Proposed Standard, Draft Standard, Internet Standard), or Historic."

[12] "Internet Official Protocol Standards (STD 1)" (ftp://ftp.rfc-editor.org/in-notes/std/std1.txt) (plain text). RFC Editor. May 2008. . Retrieved 2011-04-26.

[13] "7.5. Informational and Experimental RFCs" (http://www.ietf.org/tao.html#anchor45), *The Tao of IETF* (http://www.ietf.org/tao.html), , retrieved 2011-12-21

[14] Scott O. Bradner (October 1996). "RFC 2026: The Internet Standards Process — Revision 3" (http://tools.ietf.org/html/rfc2026). *BCP 9* (http://tools.ietf.org/html/bcp9). IETF. . Retrieved 2011-10-31.

[15] http://tools.ietf.org/html/bcp38

This article is based on material taken from the Free On-line Dictionary of Computing prior to 1 November 2008 and incorporated under the "relicensing" terms of the GFDL, version 1.3 or later.

External links

- RFC Editor (http://www.rfc-editor.org/)
  - RFC Database (http://www.rfc-editor.org/rfc.html)
  - RFC Errata (http://www.rfc-editor.org/errata.php)
  - RFC Frequently Asked Questions (http://www.rfc-editor.org/rfcfaq.html)
  - RFC Index (http://www.rfc-editor.org/rfc-index2.html) (text)
  - Official RFC standardization status (http://www.rfc-editor.org/rfcxx00.html)
- Mexican RFC (http://www.rfcs.com.mx) Mexican document.
- IETF's RFC page (http://www.ietf.org/rfc.html)
- RFC Index (http://tools.ietf.org/rfc/) (HTML) With the text of each RFC, also mentions what other RFCs this one "updates" or is "updated by".

Physical and Link

Ethernet

Ethernet /ˈiːθərnɛt/ is a family of computer networking technologies for local area networks (LANs). Ethernet was commercially introduced in 1980 and standardized in 1985 as IEEE 802.3. Ethernet has largely replaced competing wired LAN technologies.

An 8P8C modular connector (often called RJ45) commonly used on cat 5 cables in Ethernet networks

The Ethernet standards comprise several wiring and signaling variants of the OSI physical layer in use with Ethernet. The original 10BASE5 Ethernet used coaxial cable as a shared medium. Later the coaxial cables were replaced by twisted pair and fiber optic links in conjunction with hubs or switches. Data rates were periodically increased from the original 10 megabits per second, to 100 gigabits per second.

Systems communicating over Ethernet divide a stream of data into shorter pieces called frames. Each frame contains source and destination addresses and error-checking data so that damaged data can be detected and re-transmitted. As per the OSI model Ethernet provides services up to and including the data link layer.

Since its commercial release, Ethernet has retained a good degree of compatibility. Features such as the 48-bit MAC address and Ethernet frame format have influenced other networking protocols.

History

Ethernet was developed at Xerox PARC between 1973 and 1974.[1][2] It was inspired by ALOHAnet, which Robert Metcalfe had studied as part of his PhD dissertation.[3] The idea was first documented in a memo that Metcalfe wrote on May 22, 1973.[1][4] In 1975, Xerox filed a patent application listing Metcalfe, David Boggs, Chuck Thacker and Butler Lampson as inventors.[5] In 1976, after the system was deployed at PARC, Metcalfe and Boggs published a seminal paper.[6][7]

Metcalfe left Xerox in June 1979 to form 3Com.[1][8] He convinced Digital Equipment Corporation (DEC), Intel, and Xerox to work together to promote Ethernet as a standard. The so-called "DIX" standard, for "Digital/Intel/Xerox" specified 10 Mbit/s Ethernet, with 48-bit destination and source addresses and a global 16-bit Ethertype-type field. It was published on September 30, 1980 as "The Ethernet, A Local Area Network. Data Link Layer and Physical Layer Specifications".[9] Version 2 was published in November, 1982[10] and defines what has become known as Ethernet II. Formal standardization efforts proceeded at the same time.

Ethernet initially competed with two largely proprietary systems, Token Ring and Token Bus. Because Ethernet was able to adapt to market realities and shift to inexpensive and ubiquitous twisted pair wiring, these proprietary protocols soon found themselves competing in a market inundated by Ethernet products and by the end of the 1980s, Ethernet was clearly the dominant network technology.[1] In the process, 3Com became a major company. 3Com shipped its first 10 Mbit/s Ethernet 3C100 transceiver in March 1981, and that year started selling adapters for PDP-11s and VAXes, as well as Multibus-based Intel and Sun Microsystems computers.[11]:9 This was followed quickly by DEC's Unibus to Ethernet adapter, which DEC sold and used internally to build its own corporate

network, which reached over 10,000 nodes by 1986, making it one of the largest computer networks in the world at that time.[12] An Ethernet adapter card for the IBM PC was released in 1982 and by 1985, 3Com had sold 100,000.[8]

Since then Ethernet technology has evolved to meet new bandwidth and market requirements.[13] In addition to computers, Ethernet is now used to interconnect appliances and other personal devices.[1] It is used in industrial applications and is quickly replacing legacy data transmission systems in the world's telecommunications networks.[14] By 2010, the market for Ethernet equipment amounted to over $16 billion per year.[15]

Standardization

In February 1980, the Institute of Electrical and Electronics Engineers (IEEE) started project 802 to standardize local area networks (LAN).[16][8] The "DIX-group" with Gary Robinson (DEC), Phil Arst (Intel), and Bob Printis (Xerox) submitted the so-called "Blue Book" CSMA/CD specification as a candidate for the LAN specification.[9] In addition to CSMA/CD, Token Ring (supported by IBM) and Token Bus (selected and henceforward supported by General Motors) were also considered as candidates for a LAN standard. Competing proposals and broad interest in the initiative led to strong disagreement over which technology to standardize. In December 1980, the group was split into three subgroups, and standardization proceeded separately for each proposal.[8]

Delays in the standards process put at risk the market introduction of the Xerox Star workstation and 3Com's Ethernet LAN products. With such business implications in mind, David Liddle (General Manager, Xerox Office Systems) and Metcalfe (3Com) strongly supported a proposal of Fritz Röscheisen (Siemens Private Networks) for an alliance in the emerging office communication market, including Siemens' support for the international standardization of Ethernet (April 10, 1981). Ingrid Fromm, Siemens' representative to IEEE 802, quickly achieved broader support for Ethernet beyond IEEE by the establishment of a competing Task Group "Local Networks" within the European standards body ECMA TC24. As early as March 1982 ECMA TC24 with its corporate members reached agreement on a standard for CSMA/CD based on the IEEE 802 draft.[11]:8 Because the DIX proposal was most technically complete and because of the speedy action taken by ECMA which decisively contributed to the conciliation of opinions within IEEE, the IEEE 802.3 CSMA/CD standard was approved in December 1982.[8] IEEE published the 802.3 standard as a draft in 1983 and as a standard in 1985.

Approval of Ethernet on the international level was achieved by a similar, cross-partisan action with Fromm as liaison officer working to integrate International Electrotechnical Commission, TC83 and International Organization for Standardization (ISO) TC97SC6, and the ISO/IEEE 802/3 standard was approved in 1984.

Evolution

Ethernet evolved to include higher bandwidth, improved media access control methods, and different physical media. The coaxial cable was replaced with point-to-point links connected by Ethernet repeaters or switches to reduce installation costs, increase reliability, and improve management and troubleshooting. Many variants of Ethernet remain in common use.

Ethernet stations communicate by sending each other data packets: blocks of data individually sent and delivered. As with other IEEE 802 LANs, each Ethernet station is given a 48-bit MAC address. The MAC addresses are used to specify both the destination and the source of each data packet. Ethernet establishes link level connections, which can be defined using both the destination and source addresses. On reception of a transmission, the receiver uses the destination address to determine whether the transmission is relevant to the station or should be ignored. Network interfaces normally do not accept packets addressed to other Ethernet stations. Adapters come programmed with a globally unique address.[17] An Ethertype field in each frame is used by the operating system on the receiving station to select the appropriate protocol module (i.e. the Internet protocol module). Ethernet frames are said to be *self-identifying*, because of the frame type. Self-identifying frames make it possible to intermix multiple protocols on the same physical network and allow a single computer to use multiple protocols together.[18] Despite the evolution

of Ethernet technology, all generations of Ethernet (excluding early experimental versions) use the same frame formats[19] (and hence the same interface for higher layers), and can be readily interconnected through bridging.

Due to the ubiquity of Ethernet, the ever-decreasing cost of the hardware needed to support it, and the reduced panel space needed by twisted pair Ethernet, most manufacturers now build Ethernet interfaces directly into PC motherboards, eliminating the need for installation of a separate network card.[20]

Shared media

10BASE5 Ethernet equipment

Ethernet was originally based on the idea of computers communicating over a shared coaxial cable acting as a broadcast transmission medium. The methods used were similar to those used in radio systems,[21] with the common cable providing the communication channel likened to the *Luminiferous aether* in 19th century physics, and it was from this reference that the name "Ethernet" was derived.[22]

Original Ethernet's shared coaxial cable (the shared medium) traversed a building or campus to every attached machine. A scheme known as carrier sense multiple access with collision detection (CSMA/CD) governed the way the computers shared the channel. This scheme was simpler than the competing token ring or token bus technologies.[23] Computers were connected to an Attachment Unit Interface (AUI) transceiver, which was in turn connected to the cable (later with thin Ethernet the transceiver was integrated into the network adapter). While a simple passive wire was highly reliable for small networks, it was not reliable for large extended networks, where damage to the wire in a single place, or a single bad connector, could make the whole Ethernet segment unusable.[24]

Through the first half of the 1980s, Ethernet's 10BASE5 implementation used a coaxial cable 0.375 inches (9.5 mm) in diameter, later called "thick Ethernet" or "thicknet". Its successor, 10BASE2, called "thin Ethernet" or "thinnet", used a cable similar to cable television cable of the era. The emphasis was on making installation of the cable easier and less costly.

Since all communications happen on the same wire, any information sent by one computer is received by all, even if that information is intended for just one destination.[25] The network interface card interrupts the CPU only when applicable packets are received: The card ignores information not addressed to it.[26] Use of a single cable also means that the bandwidth is shared, such that, for example, available bandwidth to each device is halved when two stations are simultaneously active.

Collisions corrupt transmitted data and require stations to retransmit. The lost data and retransmissions reduce throughput. In the worst case where multiple active hosts connected with maximum allowed cable length attempt to transmit many short frames, excessive collisions can reduce throughput dramatically. However, a Xerox report in 1980 studied performance of an existing Ethernet installation under both normal and artificially generated heavy load. The report claims that 98% throughput on the LAN was observed.[27] This is in contrast with token passing LANs (token ring, token bus), all of which suffer throughput degradation as each new node comes into the LAN, due to token waits. This report was controversial, as modeling showed that collision-based networks became unstable under loads as low as 40% of nominal capacity. Many early researchers failed to understand the subtleties of the CSMA/CD protocol and how important it was to get the details right, and were really modeling somewhat different networks (usually not as good as real Ethernet).[28]

Repeaters and hubs

For signal degradation and timing reasons, coaxial Ethernet segments had a restricted size. Somewhat larger networks could be built by using an Ethernet repeater. Early repeaters had only two ports, allowing, at most, a doubling of network size. Once repeaters with more than two ports became available, it was possible to wire the network in a star topology. Early experiments with star topologies (called "Fibernet") using optical fiber were published by 1978.[29]

A 1990s network interface card supporting both coaxial cable-based 10BASE2 (BNC connector, left) and twisted pair-based 10BASE-T (8P8C connector, right)

Shared cable Ethernet was always hard to install in offices because its bus topology was in conflict with the star topology cable plans designed into buildings for telephony. Modifying Ethernet to conform to twisted pair telephone wiring already installed in commercial buildings provided another opportunity to lower costs, expand the installed base, and leverage building design, and, thus, twisted-pair Ethernet was the next logical development in the mid-1980s.

Ethernet on unshielded twisted-pair cables (UTP) began with StarLAN at 1 Mbit/s in the mid-1980s. In 1987 SynOptics introduced the first twisted-pair Ethernet at 10 Mbit/s in a star-wired cabling topology with a central hub, later called LattisNet.[30][31][8] These evolved into 10BASE-T, which was designed for point-to-point links only, and all termination was built into the device. This changed repeaters from a specialist device used at the center of large networks to a device that every twisted pair-based network with more than two machines had to use. The tree structure that resulted from this made Ethernet networks easier to maintain by preventing most faults with one peer or its associated cable from affecting other devices on the network.

Despite the physical star topology and the presence of separate transmit and receive channels in the twisted pair and fiber media, repeater based Ethernet networks still use half-duplex and CSMA/CD, with only minimal activity by the repeater, primarily the Collision Enforcement signal, in dealing with packet collisions. Every packet is sent to every port on the repeater, so bandwidth and security problems are not addressed. The total throughput of the repeater is limited to that of a single link, and all links must operate at the same speed.

Bridging and switching

While repeaters could isolate some aspects of Ethernet segments, such as cable breakages, they still forwarded all traffic to all Ethernet devices. This created practical limits on how many machines could communicate on an Ethernet network. The entire network was one collision domain, and all hosts had to be able to detect collisions anywhere on the network. This limited the number of repeaters between the farthest nodes. Segments joined by repeaters had to all operate at the same speed, making phased-in upgrades impossible.

Patch cables with patch fields of two Ethernet switches

To alleviate these problems, bridging was created to communicate at the data link layer while isolating the physical layer. With bridging, only well-formed Ethernet packets are forwarded from one Ethernet segment to another; collisions and packet errors are isolated. Prior to learning of network devices on the different segments, Ethernet bridges (and switches) work somewhat like Ethernet repeaters, passing all traffic between segments. After the bridge learns the addresses associated with each port, it forwards network traffic only to the necessary segments, improving overall

performance. Broadcast traffic is still forwarded to all network segments. Bridges also overcame the limits on total segments between two hosts and allowed the mixing of speeds, both of which are critical to deployment of Fast Ethernet.

In 1989, the networking company Kalpana introduced their EtherSwitch, the first Ethernet switch.[32] This worked somewhat differently from an Ethernet bridge, where only the header of the incoming packet would be examined before it was either dropped or forwarded to another segment. This greatly reduced the forwarding latency and the processing load on the network device. One drawback of this cut-through switching method was that packets that had been corrupted would still be propagated through the network, so a jabbering station could continue to disrupt the entire network. The eventual remedy for this was a return to the original store and forward approach of bridging, where the packet would be read into a buffer on the switch in its entirety, verified against its checksum and then forwarded, but using more powerful application-specific integrated circuits. Hence, the bridging is then done in hardware, allowing packets to be forwarded at full wire speed.

When a twisted pair or fiber link segment is used and neither end is connected to a repeater, full-duplex Ethernet becomes possible over that segment. In full-duplex mode, both devices can transmit and receive to and from each other at the same time, and there is no collision domain. This doubles the aggregate bandwidth of the link and is sometimes advertised as double the link speed (e.g., 200 Mbit/s).[33] The elimination of the collision domain for these connections also means that all the link's bandwidth can be used by the two devices on that segment and that segment length is not limited by the need for correct collision detection.

Since packets are typically delivered only to the port they are intended for, traffic on a switched Ethernet is less public than on shared-medium Ethernet. Despite this, switched Ethernet should still be regarded as an insecure network technology, because it is easy to subvert switched Ethernet systems by means such as ARP spoofing and MAC flooding.

The bandwidth advantages, the slightly better isolation of devices from each other, the ability to easily mix different speeds of devices and the elimination of the chaining limits inherent in non-switched Ethernet have made switched Ethernet the dominant network technology.[34]

Advanced networking

Simple switched Ethernet networks, while a great improvement over repeater-based Ethernet, suffer from single points of failure, attacks that trick switches or hosts into sending data to a machine even if it is not intended for it, scalability and security issues with regard to broadcast radiation and multicast traffic, and bandwidth choke points where a lot of traffic is forced down a single link.

Advanced networking features in switches and routers combat these issues through means including spanning-tree protocol to maintain the active links of the network as a tree while allowing physical loops for redundancy, port security and protection features such as MAC lock down and broadcast radiation filtering, virtual LANs to keep different classes of users separate while using the same physical infrastructure, multilayer switching to route between different classes and link aggregation to add bandwidth to overloaded links and to provide some measure of redundancy.

A core Ethernet switch

Networking advances IEEE 802.1aq (SPB) include the use of the link-state routing protocol IS-IS to allow larger networks with shortest path routes between devices. In 2012 it was stated by David Allan and Nigel Bragg, in *802.1aq Shortest Path Bridging Design and Evolution: The Architect's Perspective* that Shortest path bridging is one of the most significant enhancements in ethernets history.[35]

Varieties of Ethernet

The Ethernet physical layer evolved over a considerable time span and encompasses coaxial, twisted pair and fiber optic physical media interfaces and speeds from 10 Mbit to 100 Gbit. The most common forms used are 10BASE-T, 100BASE-TX, and 1000BASE-T. All three utilize twisted pair cables and 8P8C modular connectors. They run at 10 Mbit/s, 100 Mbit/s, and 1 Gbit/s, respectively. Fiber optic variants of Ethernet offer high performance, electrical isolation and distance (tens of kilometers with some versions). In general, network protocol stack software will work similarly on all varieties.

Ethernet frames

A data packet on the wire is called a frame. A frame begins with preamble and start frame delimiter, followed by an Ethernet header featuring source and destination MAC addresses. The middle section of the frame consists of payload data including any headers for other protocols (e.g., Internet Protocol) carried in the frame. The frame ends with a 32-bit cyclic redundancy check, which is used to detect corruption of data in transit.

Autonegotiation

Autonegotiation is the procedure by which two connected devices choose common transmission parameters, e.g. speed and duplex mode. Autonegotiation was an optional feature on first introduction of 100BASE-TX, while it is also backward compatible with 10BASE-T. Autonegotiation is mandatory for 1000BASE-T.

Notes

[1] *The History of Ethernet* (http://www.youtube.com/watch?v=g5MezxMcRmk). NetEvents.tv. 2006. . Retrieved September 10, 2011.

[2] "Ethernet Prototype Circuit Board" (http://americanhistory.si.edu/collections/object.cfm?key=35&objkey=96). Smithsonian National Museum of American History. 1973. . Retrieved September 2, 2007.

[3] Gerald W. Brock (September 25, 2003). *The Second Information Revolution*. Harvard University Press. p. 151. ISBN 0-674-01178-3.

[4] Mary Bellis. "Inventors of the Modern Computer" (http://inventors.about.com/library/weekly/aa111598.htm). About.com. . Retrieved September 10, 2011.

[5] U.S. Patent 4,063,220 (http://www.google.com/patents?vid=4063220) "Multipoint data communication system (with collision detection)"

[6] Robert Metcalfe; David Boggs (July 1976). "Ethernet: Distributed Packet Switching for Local Computer Networks" (http://www.acm.org/classics/apr96/). *Communications of the ACM* **19** (7): 395–405. doi:10.1145/360248.360253. .

[7] The experimental Ethernet described in the 1976 paper ran at 2.94 Mbit/s and had eight-bit destination and source address fields, so the original Ethernet addresses were not the MAC addresses they are today.John F. Shoch; Yogen K. Dalal; David D. Redell; Ronald C. Crane (August 1982). "Evolution of the Ethernet Local Computer Network" (http://ethernethistory.typepad.com/papers/EthernetEvolution.pdf). *IEEE Computer* **15** (8): 14–26. doi:10.1109/MC.1982.1654107. . By software convention, the 16 bits after the destination and source address fields specified a "packet type", but, as the paper says, "different protocols use disjoint sets of packet types". Thus the original packet types could vary within each different protocol. This is in contrast to the EtherType in the IEEE Ethernet standard, which specifies the protocol being used.

[8] Urd Von Burg; Martin Kenny (December 2003). "Sponsors, Communities, and Standards: Ethernet vs. Token Ring in the Local Area Networking Business" (http://hcd.ucdavis.edu/faculty/webpages/kenney/articles_files/Sponsors, Communities, and Standards: Ethernet vs.Token Ring in the Local Area Networking Business.pdf). Archived (http://www.webcitation.org/66LCgXKhx) from the original on 2012-03-21. .

[9] Digital Equipment Corporation, Intel Corporation and Xerox Corporation (30 September 1980), *The Ethernet, A Local Area Network. Data Link Layer and Physical Layer Specifications, Version 1.0* (http://ethernethistory.typepad.com/papers/EthernetSpec.pdf), Xerox Corporation, , retrieved 2011-12-10

[10] Digital Equipment Corporation, Intel Corporation and Xerox Corporation (November 1982), *The Ethernet, A Local Area Network. Data Link Layer and Physical Layer Specifications, Version 2.0* (http://decnet.ipv7.net/docs/dundas/aa-k759b-tk.pdf), Xerox Corporation, ,

retrieved 2011-12-10

[11] Robert Breyer & Sean Riley (1999). *Switched, Fast, and Gigabit Ethernet*. Macmillan. ISBN 1-57870-073-6.

[12] Jamie Parker Pearson (1992). *Digital at Work*. Digital Press. p. 163. ISBN 1-55558-092-0.

[13] Rick Merritt (December 20, 2010). *Shifts, growth ahead for 10G Ethernet* (http://www.eetimes.com/electronics-news/4211609/Shifts-growth-ahead-for-10G-Ethernet). E Times. . Retrieved September 10, 2011.

[14] "My oh My – Ethernet Growth Continues to Soar; Surpasses Legacy" (http://www.jaymiescotto.com/jsablog/2011/07/29/my-oh-my-ethernet-growth-continues-to-soar-surpasses-legacy/). Telecom News Now. July 29, 2011. . Retrieved September 10, 2011.

[15] Jim Duffy (February 22, 2010). *Cisco, Juniper, HP drive Ethernet switch market in Q4* (http://www.networkworld.com/news/2010/022210-ethernet-switch-market.html). Network World. . Retrieved September 10, 2011.

[16] Vic Hayes (August 27, 2001). "Letter to FCC" (http://www.ieeeusa.org/policy/policy/2001/01aug27IEEE802.pdf). . Retrieved October 22, 2010. "IEEE 802 has the basic charter to develop and maintain networking standards... IEEE 802 was formed in February 1980..."

[17] In some cases, the factory-assigned address can be overridden, either to avoid an address change when an adapter is replaced or to use locally administered addresses.

[18] Douglas E. Comer (2000). *Internetworking with TCP/IP – Principles, Protocols and Architecture* (4th ed.). Prentice Hall. ISBN 0-13-018380-6. 2.4.9 – Ethernet Hardware Addresses, p. 29, explains the filtering.

[19] Iljitsch van Beijnum. "Speed matters: how Ethernet went from 3Mbps to 100Gbps... and beyond" (http://arstechnica.com/gadgets/news/2011/07/ethernet-how-does-it-work.ars). Ars Technica. . Retrieved July 15, 2011. "All aspects of Ethernet were changed: its MAC procedure, the bit encoding, the wiring... only the packet format has remained the same."

[20] Geetaj Channana (November 1, 2004). "Motherboard Chipsets Roundup" (http://pcquest.ciol.com/content/search/showarticle.asp?artid=63428). PCQuest. . Retrieved October 22, 2010. "While comparing motherboards in the last issue we found that all motherboards support Ethernet connection on board."

[21] There are fundamental differences between wireless and wired shared-medium communications, such as the fact that it is much easier to detect collisions in a wired system than a wireless system.

[22] Charles E. Spurgeon (2000). *Ethernet: The Definitive Guide*. O'Reilly. ISBN 978-1-56592-660-8.

[23] In a CSMA/CD system packets must be large enough to guarantee that the leading edge of the propagating wave of the message got to all parts of the medium before the transmitter could stop transmitting, thus guaranteeing that collisions (two or more packets initiated within a window of time that forced them to overlap) would be discovered. Minimum packet size and the physical medium's total length were, thus, closely linked.

[24] Multipoint systems are also prone to strange failure modes when an electrical discontinuity reflects the signal in such a manner that some nodes would work properly, while others work slowly because of excessive retries or not at all. See standing wave for an explanation. These could be much more difficult to diagnose than a complete failure of the segment.

[25] This "one speaks, all listen" property is a security weakness of shared-medium Ethernet, since a node on an Ethernet network can eavesdrop on all traffic on the wire if it so chooses.

[26] Unless it is put into promiscuous mode.

[27] Shoch, John F. and Hupp, Jon A. (December 1980). "Measured performance of an Ethernet local network" (http://portal.acm.org/citation.cfm?doid=359038.359044#abstract). *Communications of the ACM* (ACM Press) **23** (12): 711–721. doi:10.1145/359038.359044. ISSN 0001-0782. .

[28] Boggs, D.R., Mogul, J.C., and Kent, C.A. (August 1988). "Measured capacity of an Ethernet: myths and reality" (http://portal.acm.org/citation.cfm?doid=52325.52347#abstract). *ACM SIGCOMM Computer Communication Review* (ACM Press) **18** (4): 222–234. doi:10.1145/52325.52347. ISBN 0-89791-279-9. .

[29] Eric G. Rawson; Robert M. Metcalfe (July 1978). "Fibemet: Multimode Optical Fibers for Local Computer Networks" (http://ethernethistory.typepad.com/papers/Fibernet.pdf). *IEEE transactions on communications* **26** (7): 983–990. doi:10.1109/TCOM.1978.1094189. . Retrieved June 11, 2011.

[30] Spurgeon, Charles E. (2000). *Ethernet; The Definitive Guide* (http://books.google.com/books?id=MRChaUQr0Q0C&pg=PA20&lpg=PA20&dq=synoptics+unshielded+twisted+pair&source=bl&ots=oF5HLbKhsN&sig=aw-dUL9TPoDSaZT0I5ztZvchmjE&hl=en&ei=jGSGTfDQNIqDgAex8InECA&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCQQ6AEwADg8#v=onepage&q=synoptics&f=false). Nutshell Handbook. O'Reilly. p. 29. ISBN 1-56592-660-9. .

[31] Urs von Burg (2001). *The Triumph of Ethernet: technological communities and the battle for the LAN standard* (http://books.google.com/books?id=ooBqdIXIqbwC&pg=PA175). Stanford University Press. p. 175. ISBN 0-8047-4094-1. .

[32] The term *switch* was invented by device manufacturers and does not appear in the 802.3 standard.

[33] This is misleading, as performance will double only if traffic patterns are symmetrical.

[34] "Token Ring-to-Ethernet Migration" (http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns74/ns149/net_business_benefit09186a00800c92b9_ps6600_Products_White_Paper.html). Cisco. . Retrieved October 22, 2010. "Respondents were first asked about their current and planned desktop LAN attachment standards. The results were clear—switched Fast Ethernet is the dominant choice for desktop connectivity to the network"

[35] 802.1aq Shortest Path Bridging Design and Evolution: The Architect's Perspective (http://www.eabooks.com.au/epages/eab.sf/en_au/?ObjectPath=/Shops/eabooks/Products/9781118148662)

References

Further reading

- Digital Equipment Corporation, Intel Corporation, Xerox Corporation (September, 1980). *The Ethernet: A Local Area Network* (http://portal.acm.org/citation.cfm?id=1015591.1015594). — Version 1.0 of the DIX specification.
- "Internetworking Technology Handbook" (http://docwiki.cisco.com/wiki/Ethernet_Technologies). Cisco Systems. Retrieved April 11, 2011.

External links

- IEEE 802.3 Ethernet working group (http://www.ieee802.org/3/)
- IEEE 802.3-2008 standard (http://standards.ieee.org/getieee802/802.3.html)

Ethernet over twisted pair

Ethernet over twisted-pair cable (upper) and 8P8C plug (lower)

Ethernet over twisted pair technologies use twisted-pair cables for the physical layer of an Ethernet computer network. Other Ethernet cable standards employ coaxial cable or optical fiber. Early versions developed in the 1980s included StarLAN followed by 10BASE-T. By the 1990s, fast, inexpensive technologies began to emerge. Currently the most popular are 100BASE-TX (fast Ethernet) and 1000BASE-T (gigabit Ethernet), running at 100 Mbit/s and 1000 Mbit/s (1 Gbit/s), respectively. These standards all use 8P8C connectors.[1] Meanwhile higher-speed implementations generally support lower-speed standards inclusively; thus it is possible to mix different generations of equipment. Inclusive capability is designated **10/100** or **10/100/1000**- for connections that support such combinations.[2]:123 The cables usually have four pairs of wires (though 10BASE-T and 100BASE-TX only require two of the pairs). The three standards support both full-duplex and half-duplex communication.

History

The Institute of Electrical and Electronics Engineers (IEEE) standards association ratified several versions of the technology. The first two early designs were StarLAN, standardized in 1986, at one megabit per second,[3] and LattisNet, developed in January 1987, at 10 megabit per second.[4][5] Both were developed before the 10BASE-T standard (published in 1990 as IEEE 802.3i), and both were not compatible with it.[6]

Naming

The common names for the standards derive from aspects of the physical media. The leading number (*10* in 10BASE-T) refers to the transmission speed in Mbit/s. *BASE* denotes that baseband transmission is used. The **T** designates twisted pair cable, where the pair of wires for each signal is twisted together to reduce radio frequency interference and crosstalk between pairs. Where there are several standards for the same transmission speed, they are distinguished by a letter or digit following the T, such as *TX*.

Cabling

8P8C modular plug pin positioning

Pin	Pair	Wire	Color
1	3	tip	white/green
2	3	ring	green
3	2	tip	white/orange
4	1	ring	blue
5	1	tip	white/blue
6	2	ring	orange
7	4	tip	white/brown
8	4	ring	brown

Pin	Pair	Wire	Color
1	2	tip	white/orange
2	2	ring	orange
3	3	tip	white/green
4	1	ring	blue
5	1	tip	white/blue
6	3	ring	green
7	4	tip	white/brown
8	4	ring	brown

Twisted-pair Ethernet standards are such that the majority of cables can be wired "straight-through" (pin 1 to pin 1, pin 2 to pin 2 and so on), but others may need to be wired in the "crossover" form (receive to transmit and transmit to receive).

10BASE-T and 100BASE-TX only require two pairs to operate, located on pins 1 plus 2 and pins 3 plus 6. Since 10BASE-T and 100BASE-TX need only two pairs and Category 5 cable has four pairs, it is possible, but not standards compliant, to run two network connections or use spare pairs for PoE (Power over Ethernet) (or a network connection and two phone lines) over a Category 5 cable by using the normally unused pairs (pins 4–5, 7–8) in 10- and 100-Mbit/s configurations. In practice, great care must be taken to separate these pairs as most 10/100-Mbit/s hubs, switches and PCs electrically terminate the unused pins. Moreover, 1000BASE-T requires all four pairs to operate, pins 1 and 2, 3 and 6 — as well as 4 and 5, 7 and 8.

It is conventional to wire cables for 10- or 100-Mbit/s Ethernet to either the T568A or T568B standards. Since these standards differ only in that they swap the positions of the two pairs used for transmitting and receiving (TX/RX), a cable with T568A wiring at one end and T568B wiring at the other is referred to as a crossover cable. The terms used in the explanations of the 568 standards, tip and ring, refer to older communication technologies, and equate to the positive and negative parts of the connections.

A 10BASE-T or 100BASE-TX node such as a PC, with a connector called medium dependent interfaces (MDI), transmits on pin 1 and 2 and receives on pin 3 and 6 to a network device using a "straight-through" cable. In order for two network devices or two nodes to communicate with each other (such as a switch to another switch or computer to computer) a crossover cable is often required at speeds of 10 or 100 Mbit/s. If available, connections can be made with a straight-through cable by means of an MDI-X port, also known as an "internal crossover" or "embedded crossover" connection. Hub and switch ports with such internal crossovers are usually labelled as such, with "uplink" or "X". For example, 3Com usually labels their ports 1X, 2X, and so on. In some cases a button is provided to allow a port to act as either a normal or an uplink port.

Many modern Ethernet host adapters can automatically detect another computer connected with a straight-through cable and then automatically introduce the required crossover, if needed; if neither of the adapters has this capability, then a crossover cable is required. Most newer switches have automatic crossover ("auto MDI-X" or "auto-uplink") on all ports, eliminating the uplink port and the MDI/MDI-X switch, and allowing all connections to be made with straight-through cables. If both devices being connected support 1000BASE-T according to the standards, they will connect regardless of the cable being used or how it is wired.

A 10BASE-T transmitter sends two differential voltages, +2.5 V or −2.5 V.

100BASE-TX follows the same wiring patterns as 10BASE-T, but is more sensitive to wire quality and length, due to the higher bit rates.

A 100BASE-TX transmitter sends three differential voltages, +1 V, 0 V, or −1 V.[7]

1000BASE-T uses all four pairs bi-directionally and the standard includes auto MDI-X; however, implementation is optional. With the way that 1000BASE-T implements signaling, how the cable is wired is immaterial in actual usage. The standard on copper twisted pair is IEEE 802.3ab for Cat 5e UTP, or 4D-PAM5; four dimensions using PAM (pulse amplitude modulation) with five voltages, −2 V, −1 V, 0 V, +1 V, and +2 V [8] While +2 V to −2 V voltage may appear at the pins of the line driver, the voltage on the cable is nominally +1 V, +0.5 V, 0 V, −0.5 V and −1 V.[9]

100BASE-TX and 1000BASE-T were both designed to require a minimum of Category 5 cable and also specify a maximum cable length of 100 meters. Category 5 cable has since been deprecated and new installations use Category 5e.

Unlike earlier Ethernet standards using broadband and coaxial cable, such as 10BASE5 (thicknet) and 10BASE2 (thinnet), 10BASE-T does not specify the exact type of wiring to be used, but instead specifies certain characteristics that a cable must meet. This was done in anticipation of using 10BASE-T in existing twisted-pair wiring systems that may not conform to any specified wiring standard. Some of the specified characteristics are attenuation, characteristic impedance, timing jitter, propagation delay, and several types of noise. Cable testers are widely available to check these parameters to determine if a cable can be used with 10BASE-T. These characteristics are expected to be met by 100 meters of 24-gauge unshielded twisted-pair cable. However, with high quality cabling, cable runs of 150 meters or longer are often obtained and are considered viable by most technicians familiar with the 10BASE-T specification.

Autonegotiation and duplex mismatch

Many different modes of operations (10BASE-Tx half duplex, 10BASE-T full duplex, 100BASE-TX half duplex, ...) exist for Ethernet over twisted pair, and most network adapters are capable of different modes of operation. 1000BASE-T requires autonegotiation to be on in order to operate.

When two linked interfaces are set to different duplex modes, the effect of this duplex mismatch is a network that functions much more slowly than its nominal speed. Duplex mismatch may be inadvertently caused when an administrator configures an interface to a fixed mode (e.g. 100 Mbit/s full duplex) and fails to configure the remote interface, leaving it set to autonegotiate. Then, when the autonegotiation process fails, half duplex is assumed by the autonegotiating side of the link.

Variants

Speed [Mbit/s]	Distance [m]	Name	Standard / Year	Description
1	100 (nominally)	StarLAN	802.3e 1986[10]	Runs over four wires (two twisted pairs) on telephone twisted pair or Category 3 cable. An active hub sits in the middle and has a port for each node. Manchester coded signaling.
10	100 (nominally)	LattisNet	(pre) 802.3i 1987	Runs over AT&T Premises Distribution System (PDS) wiring or four wires (two twisted pairs) on telephone twisted pair or Category 3 cable.[4][11]
10	100 (nominally)[12]	10BASE-T	802.3i 1990	Runs over four wires (two twisted pairs) on a Category 3 or Category 5 cable. Star topology with an active hub or switch sits in the middle and has a port for each node. This is also the configuration used for 100BASE-T and gigabit Ethernet. Manchester coded signaling.
100	100	100BASE-TX	802.3u 1995	4B5B MLT-3 coded signaling, CAT5 copper cabling with two twisted pairs.
1000	100	1000BASE-T	802.3ab 1999	PAM-5 coded signaling, At least Category 5 cable, with Category 5e strongly recommended copper cabling with four twisted pairs. Each pair is used in both directions simultaneously.

10 000	100	10GBASE-T	802.3an 2006	THP PAM-16 coding. Uses category 6a cable.

Notes

[1] The 8P8C modular connector is often called *RJ45* after a telephone industry standard.

[2] Charles E. Spurgeon (2000). *Ethernet: the definitive guide* (http://books.google.com/books?id=MRChaUQr0Q0C&lpg=PR12&ots=oF6HHaJokI&pg=PA123). OReilly Media. ISBN 978-1-56592-660-8. .

[3] Urs von Burg (2001). *The triumph of Ethernet: technological communities and the battle for the LAN standard* (http://books.google.com/books?id=ooBqdIXIqbwC&pg=PA175). Stanford University Press. pp. 175–176, 255–256. ISBN 978-0-8047-4095-1. .

[4] Paula Musich (August 3, 1987). "User lauds SynOptic system: LattisNet a success on PDS" (http://books.google.com/books?id=hBwEAAAAMBAJ&pg=PA2). *Network World* **4** (31): pp. 2, 39. . Retrieved June 10, 2011.

[5] W.C. Wise, Ph.D. (March 1989). "Yesterday, somebody asked me what I think about LattisNet. Here's what I told him in a nutshell" (http://books.google.com/books?id=4QQAAAAAMBAJ&pg=PA13). *CIO Magazine* **2** (6): p. 13. . Retrieved June 11, 2011. (Advertisement)

[6] *Network Maintenance and Troubleshooting Guide* (http://books.google.com/books?id=AtBeNTDGfhEC&pg=SL2-PA4&lpg=SL2-PA4&dq=starLAN+vs+LattisNet&source=bl&ots=offmXRfqcY&sig=_TcX1hYp1HI8EgTCrxAZoN_zTm8&hl=en&ei=kcGHTZeeK5GtgQfh36zKCA&sa=X&oi=book_result&ct=result&resnum=7&ved=0CDUQ6AEwBg#v=onepage&q&f=false). Fluke Networks. 2002. p. B-4. ISBN 1-58713-800-X. .

[7] David A. Weston (2001). *Electromagnetic Compatibility: principles and applications* (http://books.google.com/books?id=392CdZHdUDEC&pg=PA240). CRC Press. pp. 240–242. ISBN 0-8247-8889-3. . Retrieved June 11, 2011.

[8] Steve Prior. "1000BASE-T Duffer's Guide to Basics and Startup" (http://grouper.ieee.org/groups/802/3/minutes/july98/E2_0798.pdf). . Retrieved 2011-02-18.

[9] Nick van Bavel, Phil Callahan and John Chiang (2004-10-25). "Voltage-mode line drivers save on power" (http://www.eetimes.com/showArticle.jhtml?articleID=51200238). . Retrieved 2011-02-18.

[10] *802.3a,b,c, and e-1988 IEEE Standards for Local Area Networks: Supplements to Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications* (http://ieeexplore.ieee.org/servlet/opac?punumber=2565). IEEE Standards Association. 1987. doi:10.1109/IEEESTD.1987.78883. .

[11] Eric Killorin (November 2, 1987). "LattisNet makes the grade in Novell benchmark tests" (http://books.google.com/books?id=wxIEAAAAMBAJ&pg=PA19). **4**. Network World. p. 19. . Retrieved March 18, 2011.

[12] IEEE Computer Society (2008-12-26), *IEEE Std 802.3-2008*, IEEE

References

Further reading

- IEEE 802.3 standards documents (http://standards.ieee.org/getieee802/802.3.html)
- UTP cable termination standards 568A vs 568B (http://www.ablecables.com.au/568avb.htm)

External links

- How to Make a Network Cable, a how-to article from wikiHow
- How to create your own Ethernet Cables (http://www.ertyu.org/steven_nikkel/ethernetcables.html)
- How to wire a 10Base-T or 100Base-T connector with category 5 cable and 8P8C modular connectors (http://www.alatec.com/info/rj45.html)
- Step by step instructions on how to punch down category 5e cable to a RJ45 (http://cablesupply.com/Video-Tutorials/how-to-punch-down-an-rj45-keystone-jack-with-a-impact-tool.html)
- How to make a crossover patch cable using Cat5e or Cat6 and RJ45 (http://rj45pinout.net/?p=53)

Carrier sense multiple access with collision detection

Carrier sense multiple access with collision detection (**CSMA/CD**) is a Media Access Control method in which:[1]

- a carrier sensing scheme is used.
- a transmitting data station that detects another signal while transmitting a frame, stops transmitting that frame, transmits a jam signal, and then waits for a random time interval before trying to resend the frame.

Simplified Algorithm of CSMA/CD

CSMA/CD is a modification of pure carrier sense multiple access (CSMA). CSMA/CD is used to improve CSMA performance by terminating transmission as soon as a collision is detected, thus shortening the time required before a retry can be attempted.

Algorithm

When a station or device or node wants to send some information, it uses the following algorithm:[2]

Main procedure

1. Is my frame ready for transmission? If yes, it goes on to the next point.
2. Is medium idle? If not, wait until it becomes ready[3]
3. Start transmitting.
4. Did a collision occur? If so, go to collision detected procedure.
5. Reset retransmission counters and end frame transmission.

Collision detected procedure

1. Continue transmission until minimum packet time is reached to ensure that all receivers detect the collision.
2. Increment retransmission counter.
3. Was the maximum number of transmission attempts reached? If so, abort transmission.
4. Calculate and wait random backoff period based on number of collisions.
5. Re-enter main procedure at stage 1.

This can be likened to what happens at a dinner party, where all the guests talk to each other through a common medium (the air). Before speaking, each guest politely waits for the current speaker to finish. If two guests start speaking at the same time, both stop and wait for short, random periods of time (in Ethernet, this time is measured in microseconds). The hope is that by each choosing a random period of time, both guests will not choose the same time to try to speak again, thus avoiding another collision.

Methods for collision detection are media dependent, but on an electrical bus such as 10BASE-5 or 10BASE-2, collisions can be detected by comparing transmitted data with received data or by recognizing a higher than normal signal amplitude on the bus.

Jam signal

The **jam signal** is a signal that carries a 32-bit binary pattern sent by a data station to inform the other stations that they must not transmit.

The maximum jam-time is calculated as follows: The maximum allowed diameter of an Ethernet installation is limited to 232 bits. This makes a round-trip-time of 464 bits. As the slot time in Ethernet is 512 bits, the difference between slot time and round-trip-time is 48 bits (6 bytes), which is the maximum "jam-time".

This in turn means: A station noting a collision has occurred is sending a 4 to 6 byte long pattern composed of 16 1-0 bit combinations. Note: The size of this jam signal is clearly beyond the minimum allowed frame-size of 64 bytes.

The purpose of this is to ensure that any other node which may currently be receiving a frame will receive the jam signal in place of the correct 32-bit MAC CRC, this causes the other receivers to discard the frame due to a CRC error.

Applications

CSMA/CD was used in now obsolete shared media Ethernet variants (10BASE5, 10BASE2) and in the early versions of twisted-pair Ethernet which used repeater hubs. Modern Ethernet networks built with switches and full-duplex connections no longer utilize CSMA/CD though it is still supported for backwards compatibility. IEEE Std 802.3, which defines all Ethernet variants, for historical reasons still bears the title "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications".

Variations of the concept are used in radio frequency systems that rely on frequency sharing, including Automatic Packet Reporting System.

Notes

[1] "Carrier Sense Multiple Access Collision Detect (CSMA/CD) Explained" (http://learn-networking.com/network-design/carrier-sense-multiple-access-collision-detect-csmacd-explained). . Retrieved 2011-07-29.

[2] Tanenbaum, Andrew S. (2003). *Computernetwerken (Computer Networks)* (Fourth edition ed.). Pearson Prentice Hill. ISBN 90-430-0698-X.

[3] On Ethernet, stations must additionally wait the 96 bit interframe gap period.

References

- This article incorporates public domain material from the General Services Administration document "Federal Standard 1037C" (http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm).
- IEEE 802.3 (http://standards.ieee.org/getieee802/802.3.html)

Wireless

Wi-Fi

Wi-Fi (/ˈwaɪfaɪ/, also spelled *Wifi* or *WiFi*) is a popular technology that allows an electronic device to exchange data wirelessly (using radio waves) over a computer network, including high-speed Internet connections. The Wi-Fi Alliance defines Wi-Fi as any "wireless local area network (WLAN) products that are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards".[1] However, since most modern WLANs are based on these standards, the term "Wi-Fi" is used in general English as a synonym for "WLAN".

The Wi-Fi logo used by the Wi-Fi Alliance.

A device that can use Wi-Fi (such as a personal computer, video-game console, smartphone, tablet, or digital audio player) can connect to a network resource such as the Internet via a wireless network access point. Such an access point (or hotspot) has a range of about 20 meters (65 feet) indoors and a greater range outdoors. Hotspot coverage can comprise an area as small as a single room with walls that block radio waves or as large as many square miles — this is achieved by using multiple overlapping access points.

"Wi-Fi" is a trademark of the Wi-Fi Alliance and the brand name for products using the IEEE 802.11 family of standards. Only Wi-Fi products that complete Wi-Fi Alliance interoperability certification testing successfully may use the "Wi-Fi CERTIFIED" designation and trademark.

Wi-Fi has had a checkered security history. Its earliest encryption system, WEP, proved easy to break. Much higher quality protocols, WPA and WPA2, were added later. However, an optional feature added in 2007, called Wi-Fi Protected Setup (WPS), has a flaw that allows a remote attacker to recover the router's WPA or WPA2 password in a few hours on most implementations.[2] Some manufacturers have recommended turning off the WPS feature. The Wi-Fi Alliance has since updated its test plan and certification program to ensure all newly certified devices resist brute-force AP PIN attacks.

History

802.11 technology has its origins in a 1985 ruling by the US Federal Communications Commission that released the ISM band for unlicensed use.[3] In 1991, NCR Corporation with AT&T Corporation invented the precursor to 802.11 intended for use in cashier systems. The first wireless products were under the name WaveLAN.

Vic Hayes has been called the "father of Wi-Fi" by some, due to his involvement in negotiating the initial standards within the IEEE while chairing the workgroup.[4][5]

A large number of patents by many companies are used in 802.11 standard.[6] In 1992 and 1996, Australian organisation CSIRO obtained patents for a method later used in Wi-Fi to "unsmear" the signal.[7] In April 2009, 14 tech companies agreed to pay CSIRO $250 million for infringements on CSIRO patents.[8] This led to WiFi being attributed as an Australian invention,[9] though this has been the subject of some controversy.[10][11] CSIRO won a further $220 million settlement for Wi-Fi patent infringements in 2012 with global firms in the United States required to pay the CSIRO licensing rights estimated to be worth an additional $1 billion in royalties.[8][12][13]

In 1999, the Wi-Fi Alliance was formed as a trade association to hold the Wi-Fi trademark under which most products are sold.[14]

The name

The term *Wi-Fi*, first used commercially in August 1999,[15] was coined by a brand-consulting firm called Interbrand Corporation. The Wi-Fi Alliance had hired Interbrand to determine a name that was "a little catchier than 'IEEE 802.11b Direct Sequence'".[16][17][18] Belanger also stated that Interbrand invented *Wi-Fi* as a play on words with *Hi-Fi* (high fidelity), and also created the Wi-Fi logo.

The Wi-Fi Alliance initially used an advertising slogan for Wi-Fi, "The Standard for Wireless Fidelity",[16] but later removed the phrase from their marketing. Despite this, some documents from the Alliance dated 2003 and 2004 still contain the term *Wireless Fidelity*.[19][20] There was no official statement related to the dropping of the term.

The yin-yang Wi-Fi logo indicates the certification of a product for interoperability.[19]

Non-Wi-Fi technologies intended for fixed points such as Motorola Canopy are usually described as fixed wireless. Alternative wireless technologies include mobile phone standards such as 2G, 3G or 4G.

Wi-Fi certification

The IEEE does not test equipment for compliance with their standards. The non-profit Wi-Fi Alliance was formed in 1999 to fill this void — to establish and enforce standards for interoperability and backward compatibility, and to promote wireless local-area-network technology. As of 2010, the Wi-Fi Alliance consisted of more than 375 companies from around the world.[21][22] The Wi-Fi Alliance enforces the use of the Wi-Fi brand to technologies based on the IEEE 802.11 standards from the Institute of Electrical and Electronics Engineers. This includes wireless local area network (WLAN) connections, device to device connectivity (such as Wi-Fi Peer to Peer aka Wi-Fi Direct), Personal area network (PAN), local area network (LAN) and even some limited wide area network (WAN) connections. Manufacturers with membership in the Wi-Fi Alliance, whose products pass the certification process, gain the right to mark those products with the Wi-Fi logo.

Specifically, the certification process requires conformance to the IEEE 802.11 radio standards, the WPA and WPA2 security standards, and the EAP authentication standard. Certification may optionally include tests of IEEE 802.11 draft standards, interaction with cellular-phone technology in converged devices, and features relating to security set-up, multimedia, and power-saving.[23]

Not every Wi-Fi device is submitted for certification. The lack of Wi-Fi certification does not necessarily imply a device is incompatible with other Wi-Fi devices. If it is compliant or partly compatible, the Wi-Fi Alliance may not object to its description as a Wi-Fi device though technically only certified devices are approved. Derivative terms, such as Super Wi-Fi, coined by the US Federal Communications Commission (FCC) to describe proposed networking in the UHF TV band in the US, may or may not be sanctioned.

Uses

To connect to a Wi-Fi LAN, a computer has to be equipped with a wireless network interface controller. The combination of computer and interface controller is called a *station*. All stations share a single radio frequency communication channel. Transmissions on this channel are received by all stations within range. The hardware does not signal the user that the transmission was delivered and is therefore called a best-effort delivery mechanism. A carrier wave is used to transmit the data in packets, referred to as "Ethernet frames". Each station is constantly tuned in on the radio frequency communication channel to pick up available transmissions.

A sticker indicating to the public that a location is within range of a Wi-Fi network. A dot with curved lines radiating from it is a common symbol for Wi-Fi, representing a point transmitting a signal.[24]

Internet access

A Wi-Fi-enabled device can connect to the Internet when within range of a wireless network. The coverage of one or more (interconnected) access points — called hotspots — can extend from an area as small as a few rooms to as large as many square miles. Coverage in the larger area may require a group of access points with overlapping coverage. Outdoor public Wi-Fi technology has been used successfully in wireless mesh networks in London, UK.

Wi-Fi provides service in private homes, high street chains and independent businesses, as well as in public spaces at Wi-Fi hotspots set up either free-of-charge or commercially. Organizations and businesses, such as airports, hotels, and restaurants, often provide free-use hotspots to attract customers. Enthusiasts or authorities who wish to provide services or even to promote business in selected areas sometimes provide free Wi-Fi access.

Routers that incorporate a digital subscriber line modem or a cable modem and a Wi-Fi access point, often set up in homes and other buildings, provide Internet access and internetworking to all devices connected to them, wirelessly or via cable.

Similarly, there are battery-powered routers that include a cellular mobile Internet radiomodem and Wi-Fi access point. When subscribed to a cellular phone carrier, they allow nearby Wi-FI stations to access the Internet over 2G, 3G, or 4G networks. Many smartphones have a built-in capability of this sort, including those based on Android, Bada, iOS (iPhone), and Symbian, though carriers often disable the feature, or charge a separate fee to enable it, especially for customers with unlimited data plans.[25] "Internet pucks" provide standalone facilities of this type as well, without use of a smartphone; examples include the MiFi- and WiBro-branded devices. Some laptops that have a cellular modem card can also act as mobile Internet Wi-Fi access points.

Wi-Fi also connects places that normally don't have network access, such as kitchens and garden sheds.

City-wide Wi-Fi

Further information: Municipal wireless network

An outdoor Wi-Fi access point

In the early 2000s, many cities around the world announced plans to construct city-wide Wi-Fi networks. There are many successful examples; in 2004, Mysore became India's first Wi-fi-enabled city and second in the world after Jerusalem. A company called WiFiyNet has set up hotspots in Mysore, covering the complete city and a few nearby villages.[26]

In 2005, Sunnyvale, California, became the first city in the United States to offer city-wide free Wi-Fi,[27] and Minneapolis has generated $1.2 million in profit annually for its provider.[28]

In May 2010, London, UK, Mayor Boris Johnson pledged to have London-wide Wi-Fi by 2012.[29] Several boroughs including Westminster and Islington[30][31] already have extensive outdoor Wi-Fi coverage.

Officials in South Korea's capital are moving to provide free Internet access at more than 10,000 locations around the city, including outdoor public spaces, major streets and densely populated residential areas. Seoul will grant leases to KT, LG Telecom and SK Telecom. The companies will invest $44 million in the project, which will be completed in 2015.[32]

Campus-wide Wi-Fi

Many traditional college campuses in the United States provide at least partial wireless Wi-Fi Internet coverage. Carnegie Mellon University built the first campus-wide wireless Internet network, called Wireless Andrew at its Pittsburgh campus in 1993 before Wi-Fi branding originated.[33][34][35]

In 2000, Drexel University in Philadelphia became the United States's first major university to offer completely wireless Internet access across its entire campus.[36]

The Far Eastern University in Manila is the first university in the Philippines to implement a campus-wide WiFi coverage for its students, faculty, and staff.

Direct computer-to-computer communications

Wi-Fi also allows communications directly from one computer to another without an access point intermediary. This is called *ad hoc* Wi-Fi transmission. This wireless ad hoc network mode has proven popular with multiplayer handheld game consoles, such as the Nintendo DS, PlayStation Portable, digital cameras, and other consumer electronics devices. Some devices can also share their Internet connection using ad-hoc, becoming hotspots or "virtual routers".[37]

Similarly, the Wi-Fi Alliance promotes a specification called *Wi-Fi Direct* for file transfers and media sharing through a new discovery- and security-methodology.[38] Wi-Fi Direct launched in October 2010.[39]

Advantages and limitations

Advantages

A keychain-size Wi-Fi detector

Wi-Fi allows cheaper deployment of local area networks (LANs). Also spaces where cables cannot be run, such as outdoor areas and historical buildings, can host wireless LANs.

Manufacturers are building wireless network adapters into most laptops. The price of chipsets for Wi-Fi continues to drop, making it an economical networking option included in even more devices.

Different competitive brands of access points and client network-interfaces can inter-operate at a basic level of service. Products designated as "Wi-Fi Certified" by the Wi-Fi Alliance are backwards compatible. Unlike mobile phones, any standard Wi-Fi device will work anywhere in the world.

Wi-Fi Protected Access encryption (WPA2) is considered secure, provided a strong passphrase is used. New protocols for quality-of-service (WMM) make Wi-Fi more suitable for latency-sensitive applications (such as voice and video). Power saving mechanisms (WMM Power Save) extend battery life.

Limitations

Spectrum assignments and operational limitations are not consistent worldwide: most of Europe allows for an additional two channels beyond those permitted in the US for the 2.4 GHz band (1–13 vs. 1–11), while Japan has one more on top of that (1–14). As of 2007, Europe is essentially homogeneous in this respect.

A Wi-Fi signal occupies five channels in the 2.4 GHz band. Any two channels numbers that differ by five or more, such as 2 and 7, do not overlap. The oft-repeated adage that channels 1, 6, and 11 are the *only* non-overlapping channels is, therefore, not accurate. Channels 1, 6, and 11 are the only *group of three* non-overlapping channels in the U.S. In Europe and Japan using Channels 1, 5, 9, and 13 for 802.11g and n is recommended.

Equivalent isotropically radiated power (EIRP) in the EU is limited to 20 dBm (100 mW).

The current 'fastest' norm, 802.11n, uses double the radio spectrum/bandwidth (40 MHz) compared to 802.11a or 802.11g (20 MHz). This means there can be only one 802.11n network on the 2.4 GHz band at a given location, without interference to/from other WLAN traffic. 802.11n can also be set to use 20 MHz bandwidth only to prevent interference in dense community.

Range

Wi-Fi networks have limited range. A typical wireless access point using 802.11b or 802.11g with a stock antenna might have a range of 32 m (120 ft) indoors and 95 m (300 ft) outdoors. IEEE 802.11n, however, can more than double the range.[40] Range also varies with frequency band. Wi-Fi in the 2.4 GHz frequency block has slightly better range than Wi-Fi in the 5 GHz frequency block which is used by 802.11a and optionally by 802.11n. On wireless routers with detachable antennas, it is possible to improve range by fitting upgraded antennas which have higher gain in particular directions. Outdoor ranges can be improved to many kilometers through the use of high gain directional antennas at the router and remote device(s). In general, the maximum amount of power that a Wi-Fi device can transmit is limited by local regulations, such as FCC Part 15 in the US.

Due to reach requirements for wireless LAN applications, Wi-Fi has fairly high power consumption compared to some other standards. Technologies such as Bluetooth (designed to support wireless PAN applications) provide a much shorter propagation range of <10m[41] and so in general have a lower power consumption. Other low-power technologies such as ZigBee have fairly long range, but much lower data rate. The high power consumption of Wi-Fi

makes battery life in mobile devices a concern.

Researchers have developed a number of "no new wires" technologies to provide alternatives to Wi-Fi for applications in which Wi-Fi's indoor range is not adequate and where installing new wires (such as CAT-5) is not possible or cost-effective. For example, the ITU-T G.hn standard for high speed Local area networks uses existing home wiring (coaxial cables, phone lines and power lines). Although G.hn does not provide some of the advantages of Wi-Fi (such as mobility or outdoor use), it's designed for applications (such as IPTV distribution) where indoor range is more important than mobility.

Due to the complex nature of radio propagation at typical Wi-Fi frequencies, particularly the effects of signal reflection off trees and buildings, algorithms can only approximately predict Wi-Fi signal strength for any given area in relation to a transmitter.[42] This effect does not apply equally to long-range Wi-Fi, since longer links typically operate from towers that transmit above the surrounding foliage.

The practical range of Wi-Fi essentially confines mobile use to such applications as inventory-taking machines in warehouses or in retail spaces, barcode-reading devices at check-out stands, or receiving/shipping stations. Mobile use of Wi-Fi over wider ranges is limited, for instance, to uses such as in an automobile moving from one hotspot to another. Other wireless technologies are more suitable for communicating with moving vehicles.

Data security risks

The most common wireless encryption-standard, Wired Equivalent Privacy (WEP), has been shown to be easily breakable even when correctly configured. Wi-Fi Protected Access (WPA and WPA2) encryption, which became available in devices in 2003, aimed to solve this problem. Wi-Fi access points typically default to an encryption-free (*open*) mode. Novice users benefit from a zero-configuration device that works out-of-the-box, but this default does not enable any wireless security, providing open wireless access to a LAN. To turn security on requires the user to configure the device, usually via a software graphical user interface (GUI). On unencrypted Wi-Fi networks connecting devices can monitor and record data (including personal information). Such networks can only be secured by using other means of protection, such as a VPN or secure Hypertext Transfer Protocol (HTTPS) over Transport Layer Security.

Interference

Wi-Fi connections can be disrupted or the internet speed lowered by having other devices in the same area. Many 2.4 GHz 802.11b and 802.11g access-points default to the same channel on initial startup, contributing to congestion on certain channels. Wi-Fi pollution, or an excessive number of access points in the area, especially on the neighboring channel, can prevent access and interfere with other devices' use of other access points, caused by overlapping channels in the 802.11g/b spectrum, as well as with decreased signal-to-noise ratio (SNR) between access points. This can become a problem in high-density areas, such as large apartment complexes or office buildings with many Wi-Fi access points.

Additionally, other devices use the 2.4 GHz band: microwave ovens, ISM band devices, security cameras, ZigBee devices, Bluetooth devices, video senders, cordless phones, baby monitors, and (in some countries) Amateur radio all of which can cause significant additional interference. It is also an issue when municipalities[43] or other large entities (such as universities) seek to provide large area coverage.

Hardware

Standard devices

A wireless access point (WAP) connects a group of wireless devices to an adjacent wired LAN. An access point resembles a network hub, relaying data between connected wireless devices in addition to a (usually) single connected wired device, most often an ethernet hub or switch, allowing wireless devices to communicate with other wired devices.

An embedded RouterBoard 112 with U.FL-RSMA pigtail and R52 mini PCI Wi-Fi card widely used by wireless Internet service providers (WISPs) in the Czech Republic

Wireless adapters allow devices to connect to a wireless network. These adapters connect to devices using various external or internal interconnects such as PCI, miniPCI, USB, ExpressCard, Cardbus and PC Card. As of 2010, most newer laptop computers come equipped with built in internal adapters.

OSBRiDGE 3GN - 802.11n Access Point and UMTS/GSM Gateway in one device

Wireless routers integrate a Wireless Access Point, ethernet switch, and internal router firmware application that provides IP routing, NAT, and DNS forwarding through an integrated WAN-interface. A wireless router allows wired and wireless ethernet LAN devices to connect to a (usually) single WAN device such as a cable modem or a DSL modem. A wireless router allows all three devices, mainly the access point and router, to be configured through one central utility. This utility is usually an integrated web server that is accessible to wired and wireless LAN clients and often optionally to WAN clients. This utility may also be an application that is run on a desktop computer, as is the case with as Apple's AirPort, which is managed with the AirPort Utility on Mac OS X and Microsoft Windows.[44]

An Atheros Wi-Fi N draft adaptor with built in Bluetooth on a Sony Vaio E series laptop

Wireless network bridges connect a wired network to a wireless network. A bridge differs from an access point: an access point connects wireless devices to a wired network at the data-link layer. Two wireless bridges may be used to connect two wired networks over a wireless link, useful in situations where a wired connection may be unavailable, such as between two separate homes.

Wireless range-extenders or wireless repeaters can extend the range of an existing wireless network. Strategically placed range-extenders can elongate a

USB wireless adapter

signal area or allow for the signal area to reach around barriers such as those pertaining in L-shaped corridors. Wireless devices connected through repeaters will suffer from an increased latency for each hop, as well as from a reduction in the maximum data throughput that is available. In addition, the effect of additional users using a network employing wireless range-extenders is to consume the available bandwidth faster than would be the case where but a single user migrates around a network employing extenders. For this reason, wireless range-extenders work best in networks supporting very low traffic throughput requirements, such as for cases where but a single user with a Wi-Fi equipped tablet migrates around the combined extended and non-extended portions of the total connected network. Additionally, a wireless device connected to any of the repeaters in the chain will have a data throughput that is also limited by the "weakest link" existing in the chain between where the connection originates and where the connection ends. Networks employing wireless extenders are also more prone to degradation from interference from neighboring access points that border portions of the extended network and that happen to occupy the same channel as the extended network.

The security standard, Wi-Fi Protected Setup, allows embedded devices with limited graphical user interface to connect to the Internet with ease. Wi-Fi Protected Setup has 2 configurations: The Push Button configuration and the PIN configuration. These embedded devices are also called The Internet of Things and are low-power, battery-operated embedded systems. A number of Wi-Fi manufacturers design chips and modules for embedded Wi-Fi, such as GainSpan.[45]

Distance records

Distance records (using non-standard devices) include 382 km (237 mi) in June 2007, held by Ermanno Pietrosemoli and EsLaRed of Venezuela, transferring about 3 MB of data between the mountain-tops of El Águila and Platillon.[46][47] The Swedish Space Agency transferred data 420 km (260 mi), using 6 watt amplifiers to reach an overhead stratospheric balloon.[48]

Embedded systems

Increasingly in the last few years (particularly as of 2007), embedded Wi-Fi modules have become available that incorporate a real-time operating system and provide a simple means of wirelessly enabling any device which has and communicates via a serial port.[49] This allows the design of simple monitoring devices. An example is a portable ECG device monitoring a patient at home. This Wi-Fi-enabled device can communicate via the Internet.[50]

Embedded serial-to-Wi-Fi module

These Wi-Fi modules are designed by OEMs so that implementers need only minimal Wi-Fi knowledge to provide Wi-Fi connectivity for their products.

Multiple access points

Increasing the number of Wi-Fi access points provides network redundancy, support for fast roaming and increased overall network-capacity by using more channels or by defining smaller cells. Except for the smallest implementations (such as home or small office networks), Wi-Fi implementations have moved toward "thin" access points, with more of the network intelligence housed in a centralized network appliance, relegating individual access points to the role of "dumb" transceivers. Outdoor applications may use mesh topologies.

Network security

The main issue with wireless network security is its simplified access to the network compared to traditional wired networks such as ethernet. With wired networking one must either gain access to a building (physically connecting into the internal network) or break through an external firewall. Most business networks protect sensitive data and systems by attempting to disallow external access. Enabling wireless connectivity reduces security if the network uses inadequate or no encryption.[51]

An attacker who has gained access to a Wi-Fi network router can initiate a DNS spoofing attack against any other user of the network by forging a response before the queried DNS server has a chance to reply.[52]

Securing methods

A common measure to deter unauthorized users involves hiding the access point's name by disabling the SSID broadcast. While effective against the casual user, it is ineffective as a security method because the SSID is broadcast in the clear in response to a client SSID query. Another method is to only allow computers with known MAC addresses to join the network,[53] but determined eavesdroppers may be able join the network by spoofing an authorized address.

Wired Equivalent Privacy (WEP) encryption was designed to protect against casual snooping but it is no longer considered secure. Tools such as AirSnort or Aircrack-ng can quickly recover WEP encryption keys.[54] Because of WEP's weakness the Wi-Fi Alliance approved Wi-Fi Protected Access (WPA) which uses TKIP. WPA was specifically designed to work with older equipment usually through a firmware upgrade. Though more secure than WEP, WPA has known vulnerabilities.

The more secure WPA2 using Advanced Encryption Standard was introduced in 2004 and is supported by most new Wi-Fi devices. WPA2 is fully compatible with WPA.[55]

A flaw in a feature added to Wi-Fi in 2007, called Wi-Fi Protected Setup, allows WPA and WPA2 security to be bypassed and effectively broken in many situations. The only remedy as of late 2011 is to turn off Wi-Fi Protected Setup,[56] which is not always possible.

Piggybacking

Piggybacking refers to access to a wireless Internet connection by bringing one's own computer within the range of another's wireless connection, and using that service without the subscriber's explicit permission or knowledge.

During the early popular adoption of 802.11, providing open access points for anyone within range to use was encouraged to cultivate wireless community networks,[57] particularly since people on average use only a fraction of their downstream bandwidth at any given time.

Recreational logging and mapping of other people's access points has become known as wardriving. Indeed, many access points are intentionally installed without security turned on so that they can be used as a free service. Providing access to one's Internet connection in this fashion may breach the Terms of Service or contract with the ISP. These activities do not result in sanctions in most jurisdictions; however, legislation and case law differ considerably across the world. A proposal to leave graffiti describing available services was called warchalking.[58] A Florida court case determined that owner laziness was not to be a valid excuse.

Piggybacking often occurs unintentionally, since most access points are configured without encryption by default and operating systems can be configured to connect automatically to any available wireless network. A user who happens to start up a laptop in the vicinity of an access point may find the computer has joined the network without any visible indication. Moreover, a user intending to join one network may instead end up on another one if the latter has a stronger signal. In combination with automatic discovery of other network resources (see DHCP and Zeroconf) this could possibly lead wireless users to send sensitive data to the wrong middle-man when seeking a destination (*see Man-in-the-middle attack*). For example, a user could inadvertently use an unsecure network to log in to a

website, thereby making the login credentials available to anyone listening, if the website uses an unsecure protocol such as HTTP.

Safety

Further information: Wireless electronic devices and health

The World Health Organization (WHO) says "there is no risk from low level, long-term exposure to wi-fi networks" and the United Kingdom's Health Protection Agency reports that exposure to Wi-Fi for a year results in the "same amount of radiation from a 20-minute mobile phone call." [59][60]

A small percentage of Wi-Fi users have reported adverse health issues after repeat exposure and use of Wi-Fi,[61] though there has been no publication of any effects being observable in double-blind studies. A review of studies involving 725 people that claimed electromagnetic hypersensitivity found no evidence for their claims.[62]

One study speculated that "laptops (Wi-Fi mode) on the lap near the testicles may result in decreased male fertility".[63] Another study found decreased working memory among males during Wi-Fi exposure.[64]

Notes

[1] What is Wi-Fi? - A Word Definition From the Webopedia Computer Dictionary (http://www.webopedia.com/TERM/W/Wi_Fi.html)

[2] http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf

[3] "Wi-Fi (wireless networking technology)" (http://www.britannica.com/EBchecked/topic/1473553/Wi-Fi). *Encyclopædia Britannica*. . Retrieved 2010-02-03.

[4] Ben Charny (December 6, 2002). "CNET Vision series" (http://news.cnet.com/1200-1070-975460.html). CNET. . Retrieved 2011-10-14.

[5] Olga Kharif (April 1, 2003). "Paving the Airwaves for Wi-Fi" (http://www.businessweek.com/technology/content/apr2003/tc2003041_5423_tc107.htm). Bloomberg Businessweek. . Retrieved 2011-10-14.

[6] IEEE-SA - IEEE 802.11 and Amendments Patent Letters of Assurance (http://standards.ieee.org/about/sasb/patcom/pat802_11.html)

[7] David Sygall. " How Australia's top scientist earned millions from Wi-Fi (http://www.smh.com.au/technology/sci-tech/how-australias-top-scientist-earned-millions-from-wifi-20091207-kep4.html)". *The Sydney Morning Herald*, December 7, 2009.

[8] Moses, Asher (June 1, 2010). "CSIRO to reap 'lazy billion' from world's biggest tech companies" (http://www.theage.com.au/technology/enterprise/csiro-to-reap-lazy-billion-from-worlds-biggest-tech-companies-20100601-wsu2.html). *The Age* (Melbourne). . Retrieved 8 June 2010.

[9] World changing Aussie inventions - Australian Geographic (http://www.australiangeographic.com.au/journal/world-changing-aussie-inventions.htm)

[10] How the Aussie government “invented WiFi” and sued its way to $430 million | Ars Technica (http://arstechnica.com/tech-policy/news/2012/04/how-the-aussie-government-invented-wifi-and-sued-its-way-to-430-million.ars)

[11] "Australia's Biggest Patent Troll Goes After AT&T, Verizon and T-Mobile" (http://www.cbsnews.com/8301-505124_162-43340647/australias-biggest-patent-troll-goes-after-at038t-verizon-and-t-mobile/). *CBS News*. .

[12] Australian scientists cash in on Wi-Fi invention: SMH 1 April 2012 (http://www.smh.com.au/it-pro/government-it/australian-scientists-cash-in-on-wifi-invention-20120331-1w5gx.html)

[13] CSIRO wins legal battle over Wi-Fi patent: ABC 1 April 2012 (http://www.abc.net.au/news/2012-04-01/csiro-receives-multi-million-dollar-payment-for-wifi-technology/3925814)

[14] "Wi-Fi Alliance: Organization" (http://www.wi-fi.org/organization.php). *Official industry association web site*. . Retrieved August 23, 2011.

[15] US Patent and Trademark Office.

[16] "WiFi isn't short for "Wireless Fidelity"" (http://www.boingboing.net/2005/11/08/wifi_isnt_short_for_.html). boingboing.net. 2005-11-08. . Retrieved 2007-08-31.

[17] "Wireless Fidelity' Debunked" (http://www.wi-fiplanet.com/columns/article.php/3674591). Wi-Fi Planet. 2007-04-27. . Retrieved 2007-08-31.

[18] "What is the True Meaning of Wi-Fi?" (http://www.teleclick.ca/2005/12/what-is-the-true-meaning-of-wi-fi/). Teleclick. . Retrieved 2007-08-31.

[19] "Securing Wi-Fi Wireless Networks with Today's Technologies" (http://www.wi-fi.org/files/wp_4_Securing Wireless Networks_2-6-03.pdf). Wi-Fi Alliance. 2003-02-06. . Retrieved 2009-11-30.

[20] "WPA Deployment Guidelines for Public Access Wi-Fi Networks" (http://www.wi-fi.org/files/wp_6_WPA Deployment for Public Access_10-28-04.pdf). Wi-Fi Alliance. 2004-10-28. . Retrieved 2009-11-30.

[21] The Wi-Fi Alliance also developed technology that expanded the applicability of Wi-Fi, including a simple set up protocol (Wi-Fi Protected Set Up) and a peer to peer connectivity technology (Wi-Fi Peer to Peer) "Wi-Fi Alliance: Organization" (http://www.wi-fi.org/

organization.php). www.wi-fi.org. . Retrieved 2009-10-22.
[22] "Wi-Fi Alliance: White Papers" (http://www.wi-fi.org/wp/wifi-alliance-certification/). www.wi-fi.org. . Retrieved 2009-10-22.
[23] "Wi-Fi Alliance: Programs" (http://www.wi-fi.org/certification_programs.php). www.wi-fi.org. . Retrieved 2009-10-22.
[24] Karch, Marziah (2010). *Android for Work: Productivity for Professionals*. Apress. ISBN 9781430230007.
[25] "Mifi vs Joikuspot" (http://www.mificlub.com/2010/07/mifi-vs-joikuspot/). mificlub.com. . Retrieved 2010-10-09.
[26] The Telegraph - Say hello to India's first wirefree city (http://www.telegraphindia.com/1060820/asp/opinion/story_6632793.asp)
[27] "Sunnyvale Uses MetroFi" (http://www.unstrung.com/document.asp?doc_id=85119&WT.svl=wire1_1). unstrung.com. . Retrieved 2008-07-16.
[28] "Minneapolis moves ahead with wireless" (http://www.startribune.com/business/111286134.html). The Star Tribune. December 5, 2010. . Retrieved December 5, 2010.
[29] "London-wide wi-fi by 2012 pledge" (http://news.bbc.co.uk/2/hi/uk_news/england/london/8692103.stm). *BBC News*. 2010-05-19. . Retrieved 2010-05-19.
[30] "City of London Fires Up Europe's Most Advanced Wi-Fi Network" (http://www.govtech.com/dc/118717). www.govtech.com. . Retrieved 2007-05-14.
[31] "London gets a mile of free Wi-Fi" (http://www.zdnet.co.uk/news/networking/2005/04/18/london-gets-a-mile-of-free-wi-fi-39195421/). .zdnet.co.uk. . Retrieved 200-04-18.
[32] "Seoul Moves to Provide Free City-Wide WiFi Service" (http://blogs.voanews.com/breaking-news/2011/06/15/seoul-moves-to-provide-free-city-wide-wifi-service/). VOANEWS.COM. . Retrieved 1 April 2012.
[33] Deb Smit (October 5, 2011). "How Wi-Fi got its start on the campus of CMU, a true story" (http://popcitymedia.com/innovationnews/wifi100511.aspx). Pop City Media. . Retrieved October 6, 2011.
[34] "Wireless Andrew: Creating the World's First Wireless Campus" (http://www.cmu.edu/corporate/news/2007/features/wireless_andrew.shtml). Carnegie Mellon University. 2007. . Retrieved October 6, 2011.
[35] Wolter Lemstra; Vic Hayes; John Groenewegen (2010). *The innovation journey of Wi-Fi: the road to global success* (http://books.google.com/books?id=-OMoL5Irm08C&pg=PA121). Cambridge University Press. p. 121. ISBN 978-0-521-19971-1. . Retrieved October 6, 2011.
[36] "About the University" (http://www.drexel.edu/catalog/general/aboutuniversity.htm). Drexel.edu. . Retrieved 2011-10-14.
[37] "Wireless Home Networking with Virtual WiFi Hotspot" (http://techsansar.com/internetworking/wireless-home-networking-virtual-wifi-hotspot-2946/). Techsansar.com. 2011-01-24. . Retrieved 2011-10-14.
[38] "Wi-Fi Direct allows device-to-device links" (http://www.networkworld.com/news/2009/101409-wi-fi-direct.html?hpg1=bn). .
[39] "Wi-Fi gets personal: Groundbreaking Wi-Fi Direct launches today" (http://www.wi-fi.org/news_articles.php?f=media_news&news_id=1011). WiFi Alliance. 2010-10-25. . Retrieved 2011-01-15.
[40] "802.11n Delivers Better Range" (http://www.wi-fiplanet.com/tutorials/article.php/3680781). *Wi-Fi Planet*. 2007-05-31. .
[41] See for example IEEE Standard 802.15.4 section 1.2 scope
[42] "WiFi Mapping Software: Footprint" (http://www.alyrica.net/node/20). Alyrica Networks, Inc.. . Retrieved 2008-04-27.
[43] Wilson, Tracy V.. "How Municipal WiFi Works" (http://computer.howstuffworks.com/municipal-wifi.htm). computer.howstuffworks.com. . Retrieved 2008-03-12.
[44] "Apple.com Airport Utility Product Page" (http://www.apple.com/airportextreme/features/utility.html). Apple, Inc.. . Retrieved 2011-06-14.
[45] GainSpan specifically designs for Wi-Fi technology between Wi-Fi devices. Extremely useful. "GainSpan low-power, embedded Wi-Fi" (http://www.gainspan.com/technology/technology_overview.php). www.gainspan.com. . Retrieved 2010.
[46] "Ermanno Pietrosemoli has set a new record for the longest communication Wi-Fi link" (http://interred.wordpress.com/2007/06/18/ermanno-pietrosemoli-has-set-a-new-record-for-the-longest-communication-wi-fi-link/). . Retrieved 2008-03-10.
[47] "Wireless technology is irreplaceable for providing access in remote and scarcely populated regions" (http://www.apc.org/en/news/strategic/world/wireless-technology-irreplaceable-providing-access). . Retrieved 2008-03-10.
[48] "Long Distance WiFi Trial" (http://www.eslared.org.ve/articulos/Long Distance WiFi Trial.pdf) (PDF). . Retrieved 2008-03-10.
[49] "Quatech Rolls Out Airborne Embedded 802.11 Radio for M2M Market" (http://edageek.com/2008/04/18/embedded-wifi-radio/). . Retrieved 2008-04-29.
[50] "CIE article on embedded Wi-Fi for M2M applications" (http://www.cieonline.co.uk/cie2/articlen.asp?pid=1810&id=19742). . Retrieved 2008-08-27.
[51] "802.11 X Wireless Network in a Business Environment -- Pros and Cons." (http://networkbits.net/wireless-printing/80211-g-pros-cons-of-a-wireless-network-in-a-business-environment/). NetworkBits.net. . Retrieved 2008-04-08.
[52] Bernstein, Daniel J. (2002). "DNS forgery" (http://cr.yp.to/djbdns/forgery.html). . Retrieved 2010-03-24. "An attacker with access to your network can easily forge responses to your computer's DNS requests."
[53] Mateti, Prabhaker (2005). "Hacking Techniques in Wireless Networks" (http://www.cs.wright.edu/~pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm#_Toc77524658). Dayton, Ohio: Department of Computer Science and Engineering Wright State University. . Retrieved 2010-02-28.
[54] "Wireless Vulnerabilities & Exploits" (http://www.wirelessve.org/entries/show/WVE-2005-0020). wirelessve.org. . Retrieved 2008-04-15.
[55] "WPA2 Security Now Mandatory for Wi-Fi CERTIFIED Products" "WPA2 Security Now Mandatory for Wi-Fi CERTIFIED Products" (http://www.wi-fi.org/pressroom_overview.php?newsid=16). *Wi-Fi Alliance*. .

[56] http://www.kb.cert.org/vuls/id/723755 US CERT Vulnerability Note VU#723755
[57] "NoCat's goal is to bring you Infinite Bandwidth Everywhere for Free" (http://nocat.net/). Nocat.net. . Retrieved 2011-10-14.
[58] "Let's Warchalk" (http://www.blackbeltjones.com/warchalking/warchalking0_9.pdf) (PDF). Matt Jones. . Retrieved 2008-10-09.
[59] "Q&A: Wi-fi health concerns" (http://news.bbc.co.uk/2/hi/technology/6677051.stm). BBC News. 2007-05-21. . Retrieved 2011-10-14.
[60] "Electromagnetic Hypersensitivity (EMS)" (http://www.who.int/mediacentre/factsheets/fs296/en/), 2011
[61] "Official website" (http://www.globalnews.ca/programs/16x9/video.html?releasePID=ZDqUIbEn5ki6ptoRe__s2Oz5ODXbDiZ6). Globalnews.ca. . Retrieved 2011-10-14.
[62] ""Electromagnetic Hypersensitivity: A Systematic Review of Provocation Studies ", 2005" (http://www.psychosomaticmedicine.org/content/67/2/224.abstract). Psychosomaticmedicine.org. 2005-03-01. . Retrieved 2011-10-14.
[63] Avendaño C, Mata A, Juarez Villanueva AM, Martinez VS, Sanchez Sarmiento CA (2010) “Laptop expositions affect motility and induce DNA fragmentation in human spermatozoa in vitro by a non-thermal effect: a preliminary report” *American Society for Reproductive Medicine*, 66th Annual Meeting: O-249.
[64] Papageorgiou CC, Hountala CD, Maganioti AE, Kyprianou MA, Rabavilas AD, Papadimitriou GN, Capsalis CN (2011) “Effects of wi-fi signals on the p300 component of event-related potentials during an auditory hayling task” *J Integr Neurosci.* **10(2)**: 189-202; PMID 21714138.

References

Further reading

- *Wireless Networking in the Developing World* (PDF book)

External links

- The Wi-Fi Alliance (http://wi-fi.org/)
- List of Wi-Fi certified products (http://www.wi-fi.org/search_products.php?search=1&advanced=1&lang=en&filter_company_id=&filter_category_id=&filter_subcategory=&filter_cid=&date_from=&date_to=&selected_certifications[]=50&x=44&y=15/)

IEEE 802.11

IEEE 802.11 is a set of standards for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands. They are created and maintained by the IEEE LAN/MAN Standards Committee (IEEE 802). The base version of the standard **IEEE 802.11-2012** has had subsequent amendments. These standards provide the basis for wireless network products using the Wi-Fi brand.

General description

The 802.11 family consists of a series of half-duplex over-the-air modulation techniques that use the same basic protocol. The most popular are those defined by the 802.11b and 802.11g protocols, which are amendments to the original standard. 802.11-1997 was the first wireless networking standard, but 802.11b was the first widely accepted one, followed by 802.11g and 802.11n. 802.11n is a new multi-streaming modulation technique. Other standards in the family (c–f, h, j) are service amendments and extensions or corrections to the previous specifications.

The Linksys WRT54G contains an 802.11b/g radio with two antennas

802.11b and 802.11g use the 2.4 GHz ISM band, operating in the United States under Part 15 of the US Federal Communications Commission Rules and Regulations. Because of this choice of frequency band, 802.11b and g equipment may occasionally suffer interference from microwave ovens, cordless telephones and Bluetooth devices. 802.11b and 802.11g control their interference and susceptibility to interference by using direct-sequence spread spectrum (DSSS) and orthogonal frequency-division multiplexing (OFDM) signaling methods, respectively. 802.11a uses the 5 GHz U-NII band, which, for much of the world, offers at least 23 non-overlapping channels rather than the 2.4 GHz ISM frequency band, where adjacent channels overlap - see list of WLAN channels. Better or worse performance with higher or lower frequencies (channels) may be realized, depending on the environment.

The segment of the radio frequency spectrum used by 802.11 varies between countries. In the US, 802.11a and 802.11g devices may be operated without a license, as allowed in Part 15 of the FCC Rules and Regulations. Frequencies used by channels one through six of 802.11b and 802.11g fall within the 2.4 GHz amateur radio band. Licensed amateur radio operators may operate 802.11b/g devices under Part 97 of the FCC Rules and Regulations, allowing increased power output but not commercial content or encryption.[1]

History

802.11 technology has its origins in a 1985 ruling by the U.S. Federal Communications Commission that released the ISM band for unlicensed use.[2][3]

In 1991 NCR Corporation/AT&T (now Alcatel-Lucent and LSI Corporation) invented the precursor to 802.11 in Nieuwegein, The Netherlands. The inventors initially intended to use the technology for cashier systems. The first wireless products were brought to the market under the name WaveLAN with raw data rates of 1 Mbit/s and 2 Mbit/s.

Vic Hayes, who held the chair of IEEE 802.11 for 10 years and has been called the "father of Wi-Fi" was involved in designing the initial 802.11b and 802.11a standards within the IEEE.[4]

In 1999, the Wi-Fi Alliance was formed as a trade association to hold the Wi-Fi trademark under which most products are sold.[5]

Protocols

802.11 network standards										
802.11 protocol	Release[6]	Freq. (GHz)	Bandwidth (MHz)	Data rate per stream (Mbit/s)[7]	Allowable MIMO streams	Modulation	Approximate indoor range		Approximate outdoor range	
							(m)	(ft)	(m)	(ft)
—	Jun 1997	2.4	20	1, 2	1	DSSS, FHSS	20	66	100	330
a	Sep 1999	5	20	6, 9, 12, 18, 24, 36, 48, 54	1	OFDM	35	115	120	390
		3.7[A]					—	—	5,000	16,000[A]
b	Sep 1999	2.4	20	1, 2, 5.5, 11	1	DSSS	35	115	140	460
g	Jun 2003	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	1	OFDM, DSSS	38	125	140	460
n	Oct 2009	2.4/5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2[B]	4	OFDM	70	230	250	820[8]
			40	15, 30, 45, 60, 90, 120, 135, 150[B]			70	230	250	820[8]
ac (DRAFT)	Nov. 2011	5	20	up to 87.6[9]	8					
			40	up to 200[9]						
			80	up to 433.3[9]						
			160	up to 866.7[9]						

- A1 A2 IEEE 802.11y-2008 extended operation of 802.11a to the licensed 3.7 GHz band. Increased power limits allow a range up to 5,000 m. As of 2009, it is only being licensed in the United States by the FCC.
- B1 B2 Assumes short guard interval (SGI) enabled, otherwise reduce each data rate by 10%.

802.11-1997 (802.11 legacy)

The original version of the standard IEEE 802.11 was released in 1997 and clarified in 1999, but is today obsolete. It specified two net bit rates of 1 or 2 megabits per second (Mbit/s), plus forward error correction code. It specified three alternative physical layer technologies: diffuse infrared operating at 1 Mbit/s; frequency-hopping spread spectrum operating at 1 Mbit/s or 2 Mbit/s; and direct-sequence spread spectrum operating at 1 Mbit/s or 2 Mbit/s. The latter two radio technologies used microwave transmission over the Industrial Scientific Medical frequency band at 2.4 GHz. Some earlier WLAN technologies used lower frequencies, such as the U.S. 900 MHz ISM band.

Legacy 802.11 with direct-sequence spread spectrum was rapidly supplanted and popularized by 802.11b.

802.11a

The 802.11a standard uses the same data link layer protocol and frame format as the original standard, but an OFDM based air interface (physical layer). It operates in the 5 GHz band with a maximum net data rate of 54 Mbit/s, plus error correction code, which yields realistic net achievable throughput in the mid-20 Mbit/s.[10]

Since the 2.4 GHz band is heavily used to the point of being crowded, using the relatively unused 5 GHz band gives 802.11a a significant advantage. However, this high carrier frequency also brings a disadvantage: the effective overall range of 802.11a is less than that of 802.11b/g. In theory, 802.11a signals are absorbed more readily by walls and other solid objects in their path due to their smaller wavelength and, as a result, cannot penetrate as far as those of 802.11b. In practice, 802.11b typically has a higher range at low speeds (802.11b will reduce speed to 5 Mbit/s or even 1 Mbit/s at low signal strengths). 802.11a also suffers from interference,[11] but locally there may be fewer signals to interfere with, resulting in less interference and better throughput.

802.11b

802.11b has a maximum raw data rate of 11 Mbit/s and uses the same media access method defined in the original standard. 802.11b products appeared on the market in early 2000, since 802.11b is a direct extension of the modulation technique defined in the original standard. The dramatic increase in throughput of 802.11b (compared to the original standard) along with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as the definitive wireless LAN technology.

802.11b devices suffer interference from other products operating in the 2.4 GHz band. Devices operating in the 2.4 GHz range include microwave ovens, Bluetooth devices, baby monitors, cordless telephones and some amateur radio equipment.

802.11g

In June 2003, a third modulation standard was ratified: 802.11g. This works in the 2.4 GHz band (like 802.11b), but uses the same OFDM based transmission scheme as 802.11a. It operates at a maximum physical layer bit rate of 54 Mbit/s exclusive of forward error correction codes, or about 22 Mbit/s average throughput.[12] 802.11g hardware is fully backward compatible with 802.11b hardware and therefore is encumbered with legacy issues that reduce throughput when compared to 802.11a by ~21%.

The then-proposed 802.11g standard was rapidly adopted by consumers starting in January 2003, well before ratification, due to the desire for higher data rates as well as to reductions in manufacturing costs. By summer 2003, most dual-band 802.11a/b products became dual-band/tri-mode, supporting a and b/g in a single mobile adapter card or access point. Details of making b and g work well together occupied much of the lingering technical process; in an 802.11g network, however, activity of an 802.11b participant will reduce the data rate of the overall 802.11g network.

Like 802.11b, 802.11g devices suffer interference from other products operating in the 2.4 GHz band, for example wireless keyboards.

802.11-2007

In 2003, task group TGma was authorized to "roll up" many of the amendments to the 1999 version of the 802.11 standard. REVma or 802.11ma, as it was called, created a single document that merged 8 amendments (802.11a, b, d, e, g, h, i, j) with the base standard. Upon approval on March 8, 2007, 802.11REVma was renamed to the then-current base standard **IEEE 802.11-2007**.[13]

802.11n

802.11n is an amendment which improves upon the previous 802.11 standards by adding multiple-input multiple-output antennas (MIMO). 802.11n operates on both the 2.4 GHz and the lesser used 5 GHz bands. It operates at a maximum net data rate from 54 Mbit/s to 600 Mbit/s. The IEEE has approved the amendment and it was published in October 2009.[14][13] Prior to the final ratification, enterprises were already migrating to 802.11n networks based on the Wi-Fi Alliance's certification of products conforming to a 2007 draft of the 802.11n proposal.

802.11-2012

In 2007, task group TGmb was authorized to "roll up" many of the amendments to the 2007 version of the 802.11 standard. REVmb or 802.11mb, as it was called, created a single document that merged ten amendments (802.11k, r, y, n, w, p, z, v, u, s) with the 2007 base standard. In addition much cleanup was done, including a reordering of many of the clauses. Upon publication on March 29, 2012, the new standard was referred to as **IEEE 802.11-2012**.

802.11ac

IEEE 802.11ac is a standard under development which will provide high throughput in the 5 GHz band. This specification will enable multi-station WLAN throughput of at least 1 gigabits per second and a maximum single link throughput of at least 500 megabits per second, by using wider RF bandwidth (80 or 160 MHz), more streams (up to 8), and high-density modulation (up to 256 QAM).

802.11ad

IEEE 802.11ad "WiGig" is a new proposed standard that is already seeing a major push from hardware manufacturers. On 24 July 2012 Marvell and Wilocity announced a new partnership[15] to bring a new tri-band Wi-Fi solution to market. Using 2.4 GHz, 5 GHz and 60 GHz, the new standard can achieve a theoretical maximum throughput of up to 7Gbit/s.[13] This standard is expected to reach the market sometime in early 2014.

Channels and international compatibility

Graphical representation of Wi-Fi channels in the 2.4 GHz band

802.11 divides each of the above-described bands into *channels*, analogous to the way radio and TV broadcast bands are sub-divided. For example the 2.4000–2.4835 GHz band is divided into 13 channels spaced 5 MHz apart, with channel 1 centered on 2.412 GHz and 13 on 2.472 GHz (to which Japan added a 14th channel 12 MHz above

channel 13 which was only allowed for 802.11b). 802.11b was based on DSSS with a total channel width of 22 MHz and did not have steep skirts. Consequently only three channels do not overlap. Even now, many devices are shipped with channels 1, 6 and 11 as preset options even though with the newer 802.11g standard there are *four* non-overlapping channels - 1, 5, 9 and 13. There are now four because the OFDM modulated 802.11g channels are 20 MHz wide.

Availability of channels is regulated by country, constrained in part by how each country allocates radio spectrum to various services. At one extreme, Japan permits the use of all 14 channels for 802.11b, while other countries such as Spain initially allowed only channels 10 and 11, and France only allowed 10, 11, 12 and 13. They now allow channels 1 through 13.[16][17] North America and some Central and South American countries allow only 1 through 11.

In addition to specifying the channel centre frequency, 802.11 also specifies (in Clause 17) a spectral mask defining the permitted power distribution across each channel. The mask requires the signal be attenuated a minimum of 30 dB from its peak amplitude at ±11 MHz from the centre frequency, the point at which a channel is effectively 22 MHz wide. One consequence is that stations can only use every fourth or fifth channel without overlap, typically 1, 6 and 11 in the Americas, and in theory, 1, 5, 9 and 13 in Europe although 1, 6, and 11 is typical there too. Another is that channels 1–13 effectively require the band 2.401–2.483 GHz, the actual allocations being, for example, 2.400–2.4835 GHz in the UK, 2.402–2.4735 GHz in the US, etc.

Spectral masks for 802.11g channels 1–14 in the 2.4 GHz band

Since the spectral mask only defines power output restrictions up to ±11 MHz from the center frequency to be attenuated by −50 dBr, it is often assumed that the energy of the channel extends no further than these limits. It is more correct to say that, given the separation between channels 1, 6, and 11, the signal on any channel should be sufficiently attenuated to minimally interfere with a transmitter on any other channel. Due to the near-far problem a transmitter can impact (desense) a receiver on a "non-overlapping" channel, but only if it is close to the victim receiver (within a meter) or operating above allowed power levels.

Although the statement that channels 1, 6, and 11 are "non-overlapping" is limited to spacing or product density, the 1–6–11 guideline has merit. If transmitters are closer together than channels 1, 6, and 11 (for example, 1, 4, 7, and 10), overlap between the channels may cause unacceptable degradation of signal quality and throughput.[18] However, overlapping channels may be used under certain circumstances. This way, more channels are available.[19]

A *regdomain* in IEEE 802.11 is a regulatory region. Different countries define different levels of allowable transmitter power, time that a channel can be occupied, and different available channels.[20] Domain codes are specified for the United States, Canada, ETSI (Europe), Spain, France, Japan, and China.

Most wifi devices default to regdomain 0, which means least common denominator settings, i.e. the device will not transmit at a power above the allowable power in any nation, nor will it use frequencies that are not permitted in any nation.

The regdomain setting is often made difficult or impossible to change so that the end users do not conflict with local regulatory agencies such as the Federal Communications Commission.

Frames

Current 802.11 standards define "frame" types for use in transmission of data as well as management and control of wireless links.

Frames are divided into very specific and standardized sections. Each frame consists of a MAC header, payload and frame check sequence (FCS). Some frames may not have the payload. The first two bytes of the MAC header form a frame control field specifying the form and function of the frame. The frame control field is further subdivided into the following sub-fields:

- **Protocol Version:** two bits representing the protocol version. Currently used protocol version is zero. Other values are reserved for future use.
- **Type:** two bits identifying the type of WLAN frame. Control, Data and Management are various frame types defined in IEEE 802.11.
- **Sub Type:** Four bits providing addition discrimination between frames. Type and Sub type together to identify the exact frame.
- **ToDS and FromDS:** Each is one bit in size. They indicate whether a data frame is headed for a distribution system. Control and management frames set these values to zero. All the data frames will have one of these bits set. However communication within an IBSS network always set these bits to zero.
- **More Fragments:** The More Fragments bit is set when a packet is divided into multiple frames for transmission. Every frame except the last frame of a packet will have this bit set.
- **Retry:** Sometimes frames require retransmission, and for this there is a Retry bit which is set to one when a frame is resent. This aids in the elimination of duplicate frames.
- **Power Management:** This bit indicates the power management state of the sender after the completion of a frame exchange. Access points are required to manage the connection and will never set the power saver bit.
- **More Data:** The More Data bit is used to buffer frames received in a distributed system. The access point uses this bit to facilitate stations in power saver mode. It indicates that at least one frame is available and addresses all stations connected.
- **WEP:** The WEP bit is modified after processing a frame. It is toggled to one after a frame has been decrypted or if no encryption is set it will have already been one.
- **Order:** This bit is only set when the "strict ordering" delivery method is employed. Frames and fragments are not always sent in order as it causes a transmission performance penalty.

The next two bytes are reserved for the Duration ID field. This field can take one of three forms: Duration, Contention-Free Period (CFP), and Association ID (AID).

An 802.11 frame can have up to four address fields. Each field can carry a MAC address. Address 1 is the receiver, Address 2 is the transmitter, Address 3 is used for filtering purposes by the receiver.

- The Sequence Control field is a two-byte section used for identifying message order as well as eliminating duplicate frames. The first 4 bits are used for the fragmentation number and the last 12 bits are the sequence number.
- An optional two-byte Quality of Service control field which was added with 802.11e.
- The Frame Body field is variable in size, from 0 to 2304 bytes plus any overhead from security encapsulation and contains information from higher layers.
- The Frame Check Sequence (FCS) is the last four bytes in the standard 802.11 frame. Often referred to as the Cyclic Redundancy Check (CRC), it allows for integrity check of retrieved frames. As frames are about to be sent

the FCS is calculated and appended. When a station receives a frame it can calculate the FCS of the frame and compare it to the one received. If they match, it is assumed that the frame was not distorted during transmission.[21]

Management Frames allow for the maintenance of communication. Some common 802.11 subtypes include:

- Authentication frame: 802.11 authentication begins with the WNIC sending an authentication frame to the access point containing its identity. With an open system authentication the WNIC only sends a single authentication frame and the access point responds with an authentication frame of its own indicating acceptance or rejection. With shared key authentication, after the WNIC sends its initial authentication request it will receive an authentication frame from the access point containing challenge text. The WNIC sends an authentication frame containing the encrypted version of the challenge text to the access point. The access point ensures the text was encrypted with the correct key by decrypting it with its own key. The result of this process determines the WNIC's authentication status.
- Association request frame: sent from a station it enables the access point to allocate resources and synchronize. The frame carries information about the WNIC including supported data rates and the SSID of the network the station wishes to associate with. If the request is accepted, the access point reserves memory and establishes an association ID for the WNIC.
- Association response frame: sent from an access point to a station containing the acceptance or rejection to an association request. If it is an acceptance, the frame will contain information such an association ID and supported data rates.
- Beacon frame: Sent periodically from an access point to announce its presence and provide the SSID, and other parameters for WNICs within range.
- Deauthentication frame: Sent from a station wishing to terminate connection from another station.
- Disassociation frame: Sent from a station wishing to terminate connection. It's an elegant way to allow the access point to relinquish memory allocation and remove the WNIC from the association table.
- Probe request frame: Sent from a station when it requires information from another station.
- Probe response frame: Sent from an access point containing capability information, supported data rates, etc., after receiving a probe request frame.
- Reassociation request frame: A WNIC sends a reassociation request when it drops from range of the currently associated access point and finds another access point with a stronger signal. The new access point coordinates the forwarding of any information that may still be contained in the buffer of the previous access point.
- Reassociation response frame: Sent from an access point containing the acceptance or rejection to a WNIC reassociation request frame. The frame includes information required for association such as the association ID and supported data rates.

Control frames facilitate in the exchange of data frames between stations. Some common 802.11 control frames include:

- Acknowledgement (ACK) frame: After receiving a data frame, the receiving station will send an ACK frame to the sending station if no errors are found. If the sending station doesn't receive an ACK frame within a predetermined period of time, the sending station will resend the frame.
- Request to Send (RTS) frame: The RTS and CTS frames provide an optional collision reduction scheme for access points with hidden stations. A station sends a RTS frame to as the first step in a two-way handshake required before sending data frames.
- Clear to Send (CTS) frame: A station responds to an RTS frame with a CTS frame. It provides clearance for the requesting station to send a data frame. The CTS provides collision control management by including a time value for which all other stations are to hold off transmission while the requesting stations transmits.

Data frames carry packets from web pages, files, etc. within the body,[22] using RFC 1042 encapsulation and EtherType numbers for protocol identification.[23]

Standard and amendments

Within the IEEE 802.11 Working Group,[6] the following IEEE Standards Association Standard and Amendments exist:

- IEEE 802.11-1997: The WLAN standard was originally 1 Mbit/s and 2 Mbit/s, 2.4 GHz RF and infrared (IR) standard (1997), all the others listed below are Amendments to this standard, except for Recommended Practices 802.11F and 802.11T.
- IEEE 802.11a: 54 Mbit/s, 5 GHz standard (1999, shipping products in 2001)
- IEEE 802.11b: Enhancements to 802.11 to support 5.5 and 11 Mbit/s (1999)
- IEEE 802.11c: Bridge operation procedures; included in the IEEE 802.1D standard (2001)
- IEEE 802.11d: International (country-to-country) roaming extensions (2001)
- IEEE 802.11e: Enhancements: QoS, including packet bursting (2005)
- IEEE 802.11F: Inter-Access Point Protocol (2003) Withdrawn February 2006
- IEEE 802.11g: 54 Mbit/s, 2.4 GHz standard (backwards compatible with b) (2003)
- IEEE 802.11h: Spectrum Managed 802.11a (5 GHz) for European compatibility (2004)
- IEEE 802.11i: Enhanced security (2004)
- IEEE 802.11j: Extensions for Japan (2004)
- IEEE 802.11-2007: A new release of the standard that includes amendments a, b, d, e, g, h, i and j. (July 2007)
- IEEE 802.11k: Radio resource measurement enhancements (2008)
- IEEE 802.11n: Higher throughput improvements using MIMO (multiple input, multiple output antennas) (September 2009)
- IEEE 802.11p: WAVE—Wireless Access for the Vehicular Environment (such as ambulances and passenger cars) (July 2010)
- IEEE 802.11r: Fast BSS transition (FT) (2008)
- IEEE 802.11s: Mesh Networking, Extended Service Set (ESS) (July 2011)
- IEEE 802.11T: Wireless Performance Prediction (WPP)—test methods and metrics Recommendation cancelled
- IEEE 802.11u: Improvements related to HotSpots and 3rd party authorization of clients, e.g. cellular network offload (February 2011)
- IEEE 802.11v: Wireless network management (February 2011)
- IEEE 802.11w: Protected Management Frames (September 2009)
- IEEE 802.11y: 3650–3700 MHz Operation in the U.S. (2008)
- IEEE 802.11z: Extensions to Direct Link Setup (DLS) (September 2010)
- IEEE 802.11-2012: A new release of the standard that includes amendments k, n, p, r, s, u, v, w, y and z (March 2012)
- IEEE 802.11aa: Robust streaming of Audio Video Transport Streams (June 2012)
- IEEE 802.11ae: Prioritization of Management Frames (March 2012)

In process

- IEEE 802.11ac: Very High Throughput <6 GHz;[24] potential improvements over 802.11n: better modulation scheme (expected ~10% throughput increase), wider channels (estimate in future time 80 to 160 MHz), multi user MIMO;[25] *(~ December 2013)*
- IEEE 802.11ad: Very High Throughput 60 GHz *(~ December 2012)* - see WiGig
- IEEE 802.11af: TV Whitespace *(~ June 2014)*
- IEEE 802.11ah: Sub 1 GHz sensor network, smart metering.*(~ May 2015)*
- IEEE 802.11ai: Fast Initial Link Setup *(~ September 2014)*

To reduce confusion, no standard or task group was named 802.11l, 802.11o, 802.11q, 802.11x, 802.11ab, or 802.11ag.

802.11F and 802.11T are recommended practices rather than standards, and are capitalized as such.

802.11m is used for standard maintenance. 802.11ma was completed for 802.11-2007 and 802.11mb was completed for 802.11-2012.

Standard or amendment?

Both the terms "standard" and "amendment" are used when referring to the different variants of IEEE standards.

As far as the IEEE Standards Association is concerned, there is only one current standard; it is denoted by IEEE 802.11 followed by the date that it was published. IEEE 802.11-2012 is the only version currently in publication. The standard is updated by means of amendments. Amendments are created by task groups (TG). Both the task group and their finished document are denoted by 802.11 followed by a non-capitalized letter. For example IEEE 802.11a and IEEE 802.11b. Updating 802.11 is the responsibility of task group m. In order to create a new version, TGm combines the previous version of the standard and all published amendments. TGm also provides clarification and interpretation to industry on published documents. New versions of the **IEEE 802.11** were published in 1999, 2007 and 2012.

The working title of 802.11-2007 was 802.11-REVma. This denotes a third type of document, a "revision". The complexity of combining 802.11-1999 with 8 amendments made it necessary to revise already agreed upon text. As a result, additional guidelines associated with a revision had to be followed.

Nomenclature

Various terms in 802.11 are used to specify aspects of wireless local-area networking operation, and may be unfamiliar to some readers.

For example, Time Unit (usually abbreviated TU) is used to indicate a unit of time equal to 1024 microseconds. Numerous time constants are defined in terms of TU (rather than the nearly equal millisecond).

Also the term "Portal" is used to describe an entity that is similar to an 802.1H bridge. A Portal provides access to the WLAN by non-802.11 LAN STAs.

Community networks

With the proliferation of cable modems and DSL, there is an ever-increasing market of people who wish to establish small networks in their homes to share their broadband Internet connection.

Many hotspot or free networks frequently allow anyone within range, including passersby outside, to connect to the Internet. There are also efforts by volunteer groups to establish wireless community networks to provide free wireless connectivity to the public.

Security

In 2001, a group from the University of California, Berkeley presented a paper describing weaknesses in the 802.11 Wired Equivalent Privacy (WEP) security mechanism defined in the original standard; they were followed by Fluhrer, Mantin, and Shamir's paper titled "Weaknesses in the Key Scheduling Algorithm of RC4". Not long after, Adam Stubblefield and AT&T publicly announced the first verification of the attack. In the attack, they were able to intercept transmissions and gain unauthorized access to wireless networks.

The IEEE set up a dedicated task group to create a replacement security solution, 802.11i (previously this work was handled as part of a broader 802.11e effort to enhance the MAC layer). The Wi-Fi Alliance announced an interim specification called Wi-Fi Protected Access (WPA) based on a subset of the then current IEEE 802.11i draft. These started to appear in products in mid-2003. IEEE 802.11i (also known as WPA2) itself was ratified in June 2004, and uses government strength encryption in the Advanced Encryption Standard AES, instead of RC4, which was used in

WEP. The modern recommended encryption for the home/consumer space is WPA2 (AES Pre-Shared Key) and for the Enterprise space is WPA2 along with a RADIUS authentication server (or another type of authentication server) and a strong authentication method such as EAP-TLS.

In January 2005, the IEEE set up yet another task group "w" to protect management and broadcast frames, which previously were sent unsecured. Its standard was published in 2009.[26]

In December 2011, a security flaw was revealed that affects wireless routers with the optional Wi-Fi Protected Setup (WPS) feature. While WPS is not a part of 802.11, the flaw allows a remote attacker to recover the WPS PIN and, with it, the router's 802.11i password in a few hours.[27][28]

Non-standard 802.11 extensions and equipment

Many companies implement wireless networking equipment with non-IEEE standard 802.11 extensions either by implementing proprietary or draft features. These changes may lead to incompatibilities between these extensions.

References

- *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* [29]. (2007 revision). IEEE-SA. 12 June 2007. doi:10.1109/IEEESTD.2007.373646.
- *IEEE 802.11k-2008—Amendment 1: Radio Resource Measurement of Wireless LANs* [30]. IEEE-SA. 12 June 2008. doi:10.1109/IEEESTD.2008.4544755.
- *IEEE 802.11r-2008—Amendment 2: Fast Basic Service Set (BSS) Transition* [31]. IEEE-SA. 15 July 2008. doi:10.1109/IEEESTD.2008.4573292.
- *IEEE 802.11y-2008—Amendment 3: 3650–3700 MHz Operation in USA* [32]. IEEE-SA. 6 November 2008. doi:10.1109/IEEESTD.2008.4669928.

[1] "ARRLWeb: Part 97 - Amateur Radio Service" (http://www.arrl.org/FandES/field/regulations/news/part97/). American Radio Relay League. . Retrieved 2010-09-27.

[2] "Wi-Fi (wireless networking technology)" (http://www.britannica.com/EBchecked/topic/1473553/Wi-Fi). *Encyclopædia Britannica*. . Retrieved 2010-02-03.

[3] Wolter Lemstra , Vic Hayes , John Groenewegen , *The Innovation Journey of Wi-Fi: The Road To Global Success*, Cambridge University Press, 2010, ISBN 0-521-19971-9

[4] http://news.cnet.com/1200-1070-975460.html

[5] "Wi-Fi Alliance: Organization" (http://www.wi-fi.org/organization.php). *Official industry association web site*. . Retrieved August 23, 2011.

[6] "Official IEEE 802.11 working group project timelines" (http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm). Sept. 19, 2009. . Retrieved 2009-10-09.

[7] "Wi-Fi CERTIFIED n: Longer-Range, Faster-Throughput, Multimedia-Grade Wi-Fi® Networks" (http://www.wi-fi.org/register.php?file=wp_Wi-Fi_CERTIFIED_n_Industry.pdf) (registration required). *Wi-Fi Alliance*. September 2009. .

[8] "802.11n Delivers Better Range" (http://www.wi-fiplanet.com/tutorials/article.php/3680781). *Wi-Fi Planet*. 2007-05-31. .

[9] "IEEE802.11ac: The Next Evolution of Wi-Fi Standards" (http://www.qualcomm.com/media/documents/files/ieee802-11ac-the-next-evolution-of-wi-fi.pdf). 2012-05-11. . Retrieved 2012-05-16.

[10] http://www.oreillynet.com/wireless/2003/08/08/wireless_throughput.html

[11] Angelakis, V.; Papadakis, S.; Siris, V.A.; Traganitis, A. (March 2011), "Adjacent channel interference in 802.11a is harmful: Testbed validation of a simple quantification model" (http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05723815), *Communications Magazine* (IEEE) **49** (3): 160–166, doi:10.1109/MCOM.2011.5723815, ISSN 0163-6804,

[12] *Wireless Networking in the Developing World: A practical guide to planning and building low-cost telecommunications infrastructure* (http://wndw.net/pdf/wndw2-en/wndw2-ebook.pdf) (2nd ed.). Hacker Friendly LLC. 2007. p. 425. . page 14

[13] IEEE 802.11-2007

[14] "IEEE-SA - News & Events" (http://standards.ieee.org/announcements/ieee802.11n_2009amendment_ratified.html). Standards.ieee.org. . Retrieved 2012-05-24.

[15] "Tri-Band Chip Partnership" (http://www.brightsideofnews.com/news/2012/7/24/marvell-and-wilocity-partner-up-to-deliver-60ghz-80211ad-wi-fi.aspx). BRIGHT SIDE OF NEWS. . Retrieved 2012-07-24.

[16] "Cuadro nacional de Atribución de Frecuencias CNAF" (http://web.archive.org/web/20080213092618/http://www.mityc.es/Telecomunicaciones/Secciones/Espectro/cnaf). Secretaría de Estado de Telecomunicaciones. Archived from the original (http://www.mityc.es/Telecomunicaciones/Secciones/Espectro/cnaf) on 2008-02-13. . Retrieved 2008-03-05.

[17] "Evolution du régime d'autorisation pour les RLAN" (http://www.arcep.fr/uploads/tx_gspublication/evol-rlan-250703.pdf). French Telecommunications Regulation Authority (ART). . Retrieved 2008-10-26.
[18] "Channel Deployment Issues for 2.4 GHz 802.11 WLANs" (http://www.cisco.com/en/US/docs/wireless/technology/channel/deployment/guide/Channel.html). Cisco Systems, Inc. . Retrieved 2007-02-07.
[19] Garcia Villegas, E.; et al. (2007). "Effect of adjacent-channel interference in IEEE 802.11 WLANs" (https://upcommons.upc.edu/e-prints/bitstream/2117/1234/1/CrownCom07_CReady.pdf). *CrownCom 2007*.. ICST & IEEE. .
[20] IEEE Standard 802.11-2007 page 531
[21] "802.11 Technical Section" (http://wifi.cs.st-andrews.ac.uk/wififrame.html). . Retrieved 2008-12-15.
[22] "Understanding 802.11 Frame Types" (http://www.wi-fiplanet.com/tutorials/article.php/1447501). . Retrieved 2008-12-14.
[23] Olivier Bonaventure. "Computer Networking : Principles, Protocols and Practice" (https://scm.info.ucl.ac.be/release/cnp3/Book/0.2/html/lan/lan.html#wireless-networks). . Retrieved 2012-07-09.
[24] "IEEE P802.11 - TASK GROUP AC" (http://www.ieee802.org/11/Reports/tgac_update.htm). IEEE. November 2009. . Retrieved 2009-12-13.
[25] Fleishman, Glenn (December 7, 2009). "The future of WiFi: gigabit speeds and beyond" (http://arstechnica.com/business/guides/2009/12/wifi-looks-to-1-gigabit-horizon.ars/1). Ars Technica. . Retrieved 2009-12-13.
[26] Jesse Walker, Chair (May 2009). "Status of Project IEEE 802.11 Task Group w: Protected Management Frames" (http://grouper.ieee.org/groups/802/11/Reports/tgw_update.htm). . Retrieved August 23, 2011.
[27] http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf
[28] http://www.kb.cert.org/vuls/id/723755 US CERT Vulnerability Note VU#723755
[29] http://standards.ieee.org/getieee802/download/802.11-2007.pdf
[30] http://standards.ieee.org/getieee802/download/802.11k-2008.pdf
[31] http://standards.ieee.org/getieee802/download/802.11r-2008.pdf
[32] http://standards.ieee.org/getieee802/download/802.11y-2008.pdf

External links

- IEEE 802.11 working group (http://www.ieee802.org/11/)
- Purchase the IEEE 802.11-2012 standard from IEEE, published 29 March, 2012 (http://www.techstreet.com/cgi-bin/detail?vendor_id=4523) (Will return to the free GET IEEE Program (http://standards.ieee.org/about/get/) after 6 months of publication)
- Official 802.11 timeline standards from IEEE (http://www.ieee802.org/11/Reports/802.11_Timelines.htm)

Ebooks Dedicated to Richard Beckett

Article Sources and Contributors

Flooding (computer networking) *Source*: http://en.wikipedia.org/w/index.php?oldid=494991061 *Contributors*: BradBeattie, Brow1911, Hydrox, Jpeob, Kitplane01, Ktr101, Kvng, L3prador, Last Avenue, Maura Dailey, Maxime.Debosschere, Michael Slone, Mmernex, Sengkang, Storkk, 13 anonymous edits

Source routing *Source*: http://en.wikipedia.org/w/index.php?oldid=441839303 *Contributors*: 49oxen, Davidhowland, EngineerFromVega, Evand, Facuq, Interiot, Intgr, JNW2, Kernel Saunters, Nethgirb, Phatom87, Sanmele, Ultimus, Woohookitty, 10 anonymous edits

Spanning tree *Source*: http://en.wikipedia.org/w/index.php?oldid=513268542 *Contributors*: 16@r, AS, Alexsh, Altenmann, Ankitsocute, Anthony Appleyard, Antony.stubbs, Austinmohr, Bkell, Blathnaid, Chinju, Cybercobra, DHN, David Eppstein, David-Sarah Hopwood, Dcoetzee, Dysprosia, Gadykozma, Giftlite, Goooopy, Haham hanuka, HsuanHan, Ichensky, JLeander, Joshxyz, Jrobinjapan, KelvinMo, Knutux, Kvng, LOL, Linas, Luoq, MathMartin, McKay, Mgreenbe, Mhym, Michael Hardy, Mohitsinghr, Momotaro, Msh210, Neelix, Oleg Alexandrov, PM800, Perelaar, Pjvpjv, Pwhitey86, RuM, Ruud Koot, Ruy Pugliesi, Sinofreeman, Svick, Thecheesykid, Tizio, Twri, Uffish, Vlospf, XJamRastafire, Zarvok, 55 anonymous edits

Spanning Tree Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=521003716 *Contributors*: 7, ABF, Aaron.hebert, AaronSw, Abrech, Abune, AdrianSuter, Aldie, Altenmann, Alvestrand, Alvin-cs, Ant1357, Anthony Appleyard, Apjha, Apps123, Arnaud, Atigala2003, B4hand, Baccala@freesoft.org, Banej, BartonM, Bellenion, Bellhead, Bergonz, Blatkinson, Bongwarrior, Booyabazooka, Boscobiscotti, Bryan Derksen, Bstpierre, CALR, CableCat, Capricorn42, Charfles, CrashJunkie, Davepape, David Eppstein, Dicklyon, Dirk gently, Dominus, Dysprosia, DéRahier, EAderhold, ESkog, Easwarno1, Echoray, Edurant, Ellery, Emurphy42, Epbr123, Eug, Everyking, Evyn, Fabriciodosanjossilva, FlyingToaster, ForthOK, Frank, FrankTobia, Friday, Galoubet, Gardar Rurak, Geek2003, Ghettoblaster, Ghewgill, GhosT, Giraffedata, Glass Sword, Grafen, Guy Harris, Habbie, Harshilvyas, Hendersonnv, HorsePunchKid, Ilario, Irmatov, Ixfd64, J.delanoy, Jeffq, Jeroenr, Jesse Viviano, Jkl, JonHarder, Joseanes, Kai-Hendrik, Kaushal.shakya, Kb, KelleyCook, Kgrr, Krellis, Kvng, Larry Yuma, Lee Carre, Lhmathies, Lightmouse, LimoWreck, Lingathoor, Lotje, MER-C, MLD7865 Auto, MadSurgeon, Mchaddock, Mcicogni, Mephistophelian, Michael Hardy, Michael Slone, Miken32, Miketubby, Mincebert, Minna Sora no Shita, Mmeerman, N3rV3, Negrulio, Ngriffeth, Nickfox, Nuno Tavares, Os87, P00r, Palthainon, Pdpatil, Pgallert, Pgr94, Phisches, Plugwash, Pmaccabe, Pmj, Pnm, Populus, PsyberS, RJHall, Radagast83, Rafterman, Redfearnb, Rednectar.chris, Reinderien, Rjwilmsi, RockMFR, Salems19, Sanddune00, Sceptre, Scobra77, SeanDague, Seanx820, Shahid789, Skandha101, Skier Dude, Sricciar, Sriharsh1234, Stagira, Stephan Leeds, Stryqx, Suruena, Tfdb, The Thing That Should Not Be, The emm, Tigey, Timwi, Tinucherian, Tkteun, Tofergregg, Tonkie, Triddle, UncleBubba, Vlhsrp, Weirdy, WikiReviewer.de, Wk muriithi, Yellowking, Yidisheryid, Zack, Zigger, 360 anonymous edits

Bellman–Ford algorithm *Source*: http://en.wikipedia.org/w/index.php?oldid=517025142 *Contributors*: Aaron Rotenberg, Abednigo, Aednichols, Aene, Agthorr, AlexCovarrubias, Altenmann, Anabus, Andris, Arlekean, B3virq3b, Backslash Forwardslash, BenFrantzDale, Bjozen, Bkell, BlankVerse, Brona, CBM, Charles Matthews, CiaPan, Ciphergoth, David Eppstein, Dcoetzee, Docu, Drdevil44, Ecb29, Enochlau, Epbr123, Ferengi, FrankTobia, Fredrik, Fvw, Gadfium, Giftlite, GregorB, Guahnala, Happyuk, Headbomb, Heineman, Helix84, Iceblock, Istanton, Itai, J.delanoy, Jamelan, Jaredwf, Jellyworld, Jftuga, Josteinaj, Justin W Smith, Konstable, LOL, Lavv17, Lone boatman, Mario777Zelda, Mathmike, Mazin07, Mcld, Michael Hardy, Miym, N Shar, Naroza, Nihiltres, Nils Grimsmo, Orbst, P b1999, PanLevan, Pion, Pjrm, Poor Yorick, Posix4e, Pskjs, Quuxplusone, Rjwilmsi, RobinK, Rspeer, Ruud Koot, SQL, Salix alba, Sam Hocevar, Shuroo, Sigkill, Solon.KR, SpyMagician, Stdazi, Stderr.dk, Stern, Str82no1, Tomo, ToneDaBass, Tsunanet, Tvidas, Waldir, Wmahan, Writer130, Zholdas, 165 anonymous edits

Dijkstra's algorithm *Source*: http://en.wikipedia.org/w/index.php?oldid=521253572 *Contributors*: 2001:628:408:104:222:4DFF:FE50:969, 4v4l0n42, 90 Auto, AJim, Abu adam, Adamarnesen, Adamianash, AgadaUrbanit, Agthorr, Ahy1, AlanUS, Alanb, AlcoholVat, Alex.mccarthy, Allan speck, Alquantor, Altenmann, Amenel, Andreasneumann, Angus Lepper, Anog, Apalamarchuk, Aragorn2, Arrenlex, Arsstyleh, AxelBoldt, Aydee, B3virq3b, B6s, BACbKA, B^4, Bcnof, Beatle Fab Four, Behco, BenFrantzDale, Bgwhite, Bkell, Blueshifting, Boemanneke, Borgx, Brona, CGamesPlay, CambridgeBayWeather, Charles Matthews, Christopher Parham, Cicconetti, Cincoutprabu, Clementi, Coralmizu, Crazy george, Crefrog, Css, Csurguine, Cyde, Danmaz74, Daveagp, David Eppstein, Davub, Dcoetzee, Decrypt3, Deflective, Diego UFCG, Digwuren, Dionyziz, Dmforcier, Dmitri666, DoriSmith, Dosman, Dougher, Dreske, Drostie, Dudzcom, Dysprosia, Edemaine, ElonNarai, Erel Segal, Eric Burnett, Esrogs, Ewedistrict, Ezrakilty, Ezubaric, Faure.thomas, Foobaz, Foxj, FrankTobia, Frankrod44, Fresheneesz, GRuban, Gaiacarra, Galoubet, Gerel, Geron82, Gerrit, Giftlite, Gordonnovak, GraemeL, Graham87, Grantstevens, GregorB, Guanaco, Gutza, Hadal, Haham hanuka, Hao2lian, Happyuk, Hell112342, HereToHelp, Herry12, Huazheng, Ibmua, IgushevEdward, IkamusumeFan, Illnab1024, Iridescent, Itai, JBocco, JForget, Jacobolus, Jarble, Jaredwf, Jason.Rafe.Miller, Jellyworld, Jeltz, Jewillco, Jheiv, Jim1138, Joelimlimit, JohnBlackburne, Jongman.koo, Joniscool98, Jorvis, Julesd, Justin W Smith, K3rb, Kbkorb, Keilana, Kesla, Kh naba, King of Hearts, Kku, Kndiaye, Kooo, Kostmo, LC, LOL, Laurinkus, Lavaka, Leonard G., Lone boatman, LordArtemis, LunaticFringe, Mahanga, Mameisam, MarkSweep, Martynas Patasius, MathMartin, Mathiastck, MattGiuca, Matusz, Mccraig, Mcculley, Megharajv, MementoVivere, Merlion444, Mgreenbe, Michael Slone, Mikeo, Mikrosam Akademija 2, Milcke, MindAfterMath, Mkw813, MrOllie, MusicScience, Mwarren us, Nanobear, NetRollller 3D, Nethgirb, Nixdorf, Noogz, Obradovic Goran, Obscurans, Oleg Alexandrov, Oliphaunt, Optikos, Owen, PesoSwe, Piojo, Pol098, PoliticalJunkie, Possum, ProBoj!, Pseudomonas, Pshanka, Pskjs, Pxtreme75, Quidquam, RISHARTHA, Radim Baca, Rami R, RamiWissa, Recognizance, Reliableforever, Rhanekom, Rjwilmsi, Robert Southworth, RodrigoCamargo, RoyBoy, Ruud Koot, Ryan Roos, Ryanli, SQGibbon, Sambayless, Sarkar112, Shd, Sheepeatgrass, Shizny, Shriram, Shuroo, Sidonath, SiobhanHansa, Sk2613, Slakr, Sniedo, Sokari, Someone else, Soytuny, Sprhodes, Stdazi, SteveJothen, Subh83, Sundar, Svick, T0ljan, Tehwikipwnerer, Tesse, Thayts, The Arbiter, TheRingess, Thijswijs, Thom2729, ThomasGHenry, Timwi, Tobei, Tomisti, Torla42, Turketwh, VTBassMatt, Vecrumba, Vevek, Watcher, Wierdy1024, WikiSlasher, Wikipelli, Wildcat dunny, Wphamilton, X7q, Xerox 5B, Ycl6, Yutsi, Yworo, ZeroOne, Zhaladshar, Zr2d2, 576 anonymous edits

Routing Information Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=512414620 *Contributors*: AS, Ahoerstemeier, Ahull, Akc9000, Alan.murray, Alansohn, Aldie, AlephGamma, Alexh19740110, Allolex, Andareed, Andre Engels, Anon lynx, Barberio, Blehfu, Borgx, CSW1986, Callidior, Cybjit, Dajp, DanCasas, Dansen926, David-Sarah Hopwood, Dgtsyb, Dinomite, Druiloor, Drunkenmonkey, Ducknish, Endobson, Enjoi4586, Epbr123, Fab, FlavioMartins, Gamera2, Georgette2, Gmalkin42, Goat-see, Gustavb, H2g2bob, Hadal, Hairy Dude, Hcberkowitz, HeathPetersen, Helon, Honeyman, Iamregin, Iridescent, JHunterJ, JTN, Jakew, Jec, Jnc, Johnuniq, Jon Awbrey, Jwoodger, KD5TVI, Kbrose, Kotiwalo, L33th4x0rguy, Lachlancooper, Lanny68, Lightmouse, Lls71, Maffmommie, Magioladitis, Mercury543210, Methossant, Mgeorge27, Michael j harris, Micmon, Minusf, Mmeerman, Mmernex, MrOllie, MureninC, NCurse, Nealmcb, Neilc, Nimiew, Nixdorf, Nkansahrexford, Northernhenge, Ohnoitsjamie, Oxymoron83, Paul, PaulVIF, Peripitus, Pgallert, Piet Delport, Pmsyyz, Poor Yorick, Ps-cz, R.srinivaas, RJaguar3, Raju5134, RedWolf, Redlazer, ReinforcedReinforcements, Reisio, RexNL, Robert Brockway, Ronz, S.boginskis, Sdrtirs, Seth Nimbosa, Sfisher, Shadowjams, Shahid789, Simon.clayton, Sjc, Skandha101, Suruena, Takanoha, Termininja, ThFabba, ThaddeusB, The Anome, TheGreatFoo, TimonyCrickets, Tomek0001, UncleBubba, Vary, Vickeybhopal, Vjardin, Wisamzaqoot, Yaris678, Yashpundir, Zamfi, 271 anonymous edits

Open Shortest Path First *Source*: http://en.wikipedia.org/w/index.php?oldid=520096249 *Contributors*: A Meteorite, ALargeElk, Agreenhalgh, Aldie, Alexf, Alisterb, Allens, Andrei Stroe, Andyluciano, Anon lynx, Antonvanderleun, Anuragkothari, Ari.takanen, Astralblue, Attilabedo, Avto.prangulashvili, AxelBoldt, Babinos30, Biot, Blaxthos, Bomazi, Bradleyem, Breakingaway, Brianga, C46, Can't sleep, clown will eat me, Caruch6392, Catamorphism, Celinama, Chealer, Cwolfsheep, Dandorid, Darthsco, Deineka, Dewritech, Dgtsyb, Dicklyon, Dinomite, Dougluce, Dysprosia, EJSawyer, Enjoi4586, Evanh2008, Foobarnix, Franboon, Fudoreaper, Giftlite, H8a1r18i9, Hads1, Hangin10, Hashar, Hcberkowitz, Hede2000, Hilmarz, Hns, IMSoP, Iamregin, Ibc111, Insanity Incarnate, JHunterJ, JTN, Jac16888, Jakub Horky, Janizary, Javifs, Jawsper, Jeffreyahaines, Jerome, JialiangGao, Jkurynny, Jnc, John, John of Reading, Johnuniq, Jon Awbrey, JonHarder, Joy, JtiToo, Kbrose, Kcoyle, Kenyon, Kinema, Kingpin13, Knowledge512109, Kozuch, Krypticmind, Lightmouse, Lineslarge, Lotje, Luc4, MARQUIS111, MarsRover, Materialscientist, Mblumber, McHildinger, MessiFCB, Michael Hardy, Mickdermack, Mlewis000, MrOllie, Mwalsh34, Nate, Nbarth, Nealmcb, Niekko1, Niemivh, Ninjamaster8, Niteowlneils, Nixdorf, Nkansahrexford, Nothingofwater, Nubiatech, ObfuscatePenguin, Omarecd, Omicronpersei8, Onceler, Pairadox, Palthainon, Paluchpeter, Pankkake, Pchov, Pgallert, Phil Boswell, Phil Holmes, Ptewiki, Radagast83, Rfc1394, Rick Sidwell, Rjwilmsi, Roshjack47, Rubiojr, Samriva, Seronline, Shahid789, Sigkill, Sigma 7, Silverwindx, Skor, Snowolf, Souliouz, Sperxios, Sreeakshay, Sreya1102, Srleffler, Ssrao08, Stephan Leeds, Storkk, Suruena, Svick, Takanoha, Taral, Teles, Testbells, TheGrimReaper NS, Thejapanesegeek, Tigaente, Tintin1107, Tmaufer, Toe Rag, Tomaxer, Torreslfchero, Uday, UncleBubba, Vdham, Vjardin, Wavelength, Wbenton, Wbenton-test, Wisden17, Wisq, Wk muriithi, Woohookitty, Wrcarm, Wtmitchell, Wtt, Xpclient, Zamfi, Zr2d2, 469 anonymous edits

Autonomous System (Internet) *Source*: http://en.wikipedia.org/w/index.php?oldid=515961974 *Contributors*: 28421u2232nfenfcenc, 2A00:F480:4:2A1:FD90:362:593E:C7DF, Abune, Adzinok, Babajobu, Betacommand, Borgx, Bwoodcock, Canderson7, Charles Matthews, Cmdrjameson, Coppertwig, Cwolfsheep, Cybercobra, Dan100, Duckbill, EmreDuran, Erkan Yilmaz, Etu, Fl, Fudoreaper, Furtim, Gerbrant, Giftlite, Gobbleswoggler, HCBerkowitz, Hashar, HorsePunchKid, Hughesey, IW.HG, Infofarmer, Isoma, JMS Old Al, JTN, Jag123, Johnuniq, Josh Parris, Kbrose, Ke4djt, Kenyon, Kgfleischmann, Kingdon, Ksn, Mandarax, Michael Hardy, Mjuarez, Niteowlneils, Nneonneo, Omarecd, Oneiros, OverlordQ, Pap3rinik, Petri Krohn, Pgilman, Phantomsteve, Pharaoh of the Wizards, Pmsyyz, Qr.brown, RG2, Radagast83, Schneelocke, Scolobb, Serych, SiD3WiNDR, Sigkill, Silvarbullet1, Snaily, The Anome, Timpala, Tjwagner, Ttaylor249, Wrs1864, Петър Петров, 99 anonymous edits

Border Gateway Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=520862516 *Contributors*: 2D, AS, Abhishek191288, Adamantios, Aditya21890, Ahoerstemeier, Aldie, Alexius08, Andareed, Aneah, Anna Lincoln, Aphstein, Ari.takanen, Aryaniae, Ash71p, Astronouth7303, Balabiot, Barrylb, Benjamin Pineau, Bergonz, Betacommand, Bfigura's puppy, Bill.albing, Bobo192, Breno, CBM, Can't sleep, clown will eat me, Charm, Chealer, Chochopk, Chris the speller, ChrisCPearson, Cluth, Colfer2, Crazycomputers, DESiegel, Danhash, DanielSiva, Dannofarrell, DarkCow, David Eppstein, DavidDouthitt, Daydr3am3r, Dgtsyb, Dguye, Digitalme, Dimawik, Dinomite, Donpayette, EagleEarth, Edward, Ego White Tray, ElBenevolente, Elder, Emccrckn, Equendil, Erikrespo, Fontas, Foobaz, Frazzydee, Fsamuels, Fsiler, Func, Furby100, Fylbecatulous, Gail, GastonSenac, Geeoharee, Georgesawyer, Gfrank, Giftlite, Gnebulon, Gonchibolso12, Gosgood, Hads1, Hairy Dude, Hashar, Hcberkowitz, Htaccess, Hughcharlesparker, Humble226, IMSoP, Iamregin, Ianeiloart, Ichthyos, Igxx, IncognitoErgoSum, Inkling, JHunterJ, JTN, Jaeger5432, JamesBWatson, Jandalhandler, Janizary, Japanese Searobin, Jarsyl, Jbond00747, Jgs42, Jnc, Joeinwap, Joel7687, JohnOwens, Johnh, Johnthomastaylor, Jon Awbrey, Jwvo, Kbrose, Kfor, Kinema, Kll, Klykken, Korstellan, Krellis, Kvng, Kyleflaherty, L Kensington, LaMenta3, Laug, Leebert, Lentoyip, Lessaid, Lightmouse, Llywrch, Lsec, Luder, Lukeritchie, MDW19873, Marcod'Itri, Marokwitz, MassimoAr, MathsPoetry, MatthewWilder, MaverickSolutions, Mblumber, Michael Hardy, Michael miceli, Mike Rosoft, Miym, Mozzerati, Mro, Muhandes, Nealmcb,

Negrulio, Nethgirb, Nightstallion, Nihiltres, Nikai, Nneonneo, Nubiatech, Nv8200p, Olipro, Oxymoron83, Petri Krohn, Phatom87, Philip Trueman, Phillipgrasso, Prolixium, Pvinci, Qr.brown, Rdummarf, Redquark, Reedy, RekishiEJ, Requestion, Rhondle, Ric8ard, Rich Farmbrough, Robert Brockway, Ronk01, Root2, Ruud Koot, Schneelocke, Sepiraph, Sherocksfriend, Sigkill, Signalhead, Sim, Snarius, So14k, Southpaw018, Sspecter, Strait, Super Jamie, Suruena, Tekhammer, Tekhnofiend, Telecomtom, Termininja, The Anome, The Thing That Should Not Be, Thedangerouskitchen, Thiseye, Thruston, Tjmao, Tjpayne, Tm knoll, Tmaufer, TreyGeek, Ttwaring, UncleBubba, Vary, Vcappuccio, Versus22, Via strass, Vjardin, Wavelength, Woohookitty, Wrs1864, Xipher, Xoxon1kk1, Yllt, Zakawer, Zalnas, ZippyCycle, 433 anonymous edits

Request for Comments *Source*: http://en.wikipedia.org/w/index.php?oldid=514437715 *Contributors*: 213.132.131.xxx, Abune, Ahoerstemeier, Ais523, Alex rosenberg35, Alvestrand, Amalthea, Ancheta Wis, Andre Engels, AndrewWTaylor, Angela, Antandrus, Anthony Appleyard, Ardonik, Bevo, Bk0, Blahma, BonsaiViking, Bsdlogical, Byrial, CanisRufus, Carlosguitar, Cate, Cburnett, ChrisRuvolo, Christian List, Circeus, CodeMonk, Conversion script, Coolcaesar, Cspan64, Deineka, Delirium, DerHexer, Dilbert, Direktorxxx, Dismas, Donvigy, Dori, Dtobias, EdC, Edcolins, ElKevbo, Elchonon Edelson, Emperorbma, Emurphy42, Esampathj, Espoo, Evercat, Evertype, Folajimi, Furrykef, Galoubet, Gerrit, Grafman, Graham87, GregorB, Grendelkhan, Gtrmp, Hairy Dude, Hayabusa future, Hooloovoo, Howcheng, IMSoP, Javawizard, Jeekc, Jerith, Jiy, Jkrey, Jnc, John254, Jor, Jrp, Kaihsu, Kbrose, Kdz, Kieff, Kingpin13, Klemen Kocjancic, Knuckles, Kvng, LA2, LC, Liftarn, Looxix, Lotje, Lucio, Lupin, MER-C, Mac, Malcolmj1, Man It's So Loud In Here, Marco Krohn, Marnanel, MartinSpamer, Maurice Carbonaro, Mav, McGinnis, Mel Etitis, Mendalus, Mike Rosoft, Mindmatrix, Mormegil, Mrwojo, Mykolas OK, Nanshu, NapoliRoma, Nate Silva, Nazi 2007, Nealmcb, Noumenorian, Nuno Tavares, Omniplex, Outlyer, Paul1337, PaulHanson, Pelex, Phantomsteve, Philip Trueman, Philosopher, Pkarjala, Polimerek, Qwerty0, RandallJones, Randolf Richardson, RedWolf, Rfc1394, Riana, Rich Farmbrough, Rick Block, Ringbang, RobLa, Rossj81, RoySmith, Rune.welsh, SchuminWeb, Scurra, Shadowlynk, Skarebo, Spellmaster, Srbanator, Stephan Leeds, Stephen Gilbert, Suruena, Tero, The Anome, The Belgain, Thickslab, Timwi, Tobias Bergemann, Tothwolf, Ttlkr, Vary, VeryVerily, W Nowicki, Wanion, Weppos, Wilinckx, Wrs1864, XIN3N, Yaronf, Zhen Lin, Zippedmartin, Zundark, Zzyzx11, Ævar Arnfjörð Bjarmason, Μάριος Ζηντίλης, 177 anonymous edits

Ethernet *Source*: http://en.wikipedia.org/w/index.php?oldid=521450382 *Contributors*: 0612, 121a0012, 1lovegal, 1stJahman, 4a6f656c, 7, ARC Gritt, Aapo Laitinen, Abdull, Adashiel, AdjustShift, Adrian.benko, Aeluwas, Ahoerstemeier, Ajraddatz, Akriasas, Alan Liefting, Alansohn, Aldie, Ale jrb, Alecv, Algocu, AlistairMcMillan, Altermike, Alvestrand, Alyssapvr, Amatulic, Ameliorate!, Amillar, Amore proprio, Anaraug, Andareed, Andreas Toth, Andy Dingley, Andybryant, Andyhodapp, Aneah, Angela, Angusmca, Anss123, Applemacintosh10, Arch dude, Arkrishna, Armando, ArnoldReinhold, Artaxiad, AtomEdge, Avono, Azaghal of Belegost, B4hand, BD2412, Bbachrac, Bctwriter, Bdmcmahon, Becritical, Betacommand, Betbest1, BiT, Bidabadi, Bigboymcgee, Bigjimr, Bilbo1507, Biot, BlueAg09, Blutrot, Bobblewik, Boffy b, Bomac, Bookbrad, Bookofjude, Bootleggingly mcbootleg, Boscobiscotti, Bovineone, Branclem, Branko, Brian Patrie, Bruce1ee, Btilm, Bumm13, Buy P%E%P%S%I, C0nanPayne, CRGreathouse, CS46, Calltech, Caltas, CamTarn, Can't sleep, clown will eat me, Capricorn42, Carlo.Ierna, Casey Abell, Catgut, Causantin, Cburnett, CesarB, Chaosgate, Charles Gaudette, Chasingsol, Chealer, Chmod007, Chowbok, Cjdaniel, ClementSeveillac, Closedmouth, CloudNine, Cluth, Cmdrjameson, Colin Marquardt, Conversion script, Corpx, Corti, Corwin8, Cpl Syx, Creidieki, Crispmuncher, Crissov, CryptoDerk, Cstanners, Cybercobra, DVdm, Daa89563, Dachshund, Damian Yerrick, Davehard, David Biddulph, DavidBailey, DavidH, Dcorzine, Deflective, Dennis Brown, Dicklyon, Digitalsushi, Discospinster, Diskdoc, DocWatson42, Dogcow, Donnieleefarrow, Download, Drak2, Drphilharmonic, Dtcdthingy, EEgirl18, EagleOne, Ebear422, Econrad, Eeekster, Efa, Egil, Ehn, Eivind F Øyangen, El Cubano, ElBenevolente, Electron9, Elektrik Shoos, Eleuther, Elinruby, Eloil, Engineerism, Enjoi4586, EoGuy, Epbr123, Evil genius, Ewlyahoocom, Excirial, Facts707, Femto, Fenrisulfr, Fetofs, Fieldday-sunday, Finest1, Firsfron, Flewis, FocalPoint, Frankchn, Frap, Fratrep, Fred Bradstadt, Fredrik, Fresheneesz, Fudoreaper, Fulldecent, Furrykef, G4wsz, GTAKIllerEric, Gah4, Gascreed, Geek2003, Gekkoblaster, Generica, Ghaly, Giftlite, Glacialfox, Glenn, Gnalk, Gnuish, Gogo Dodo, Golgofrinchian, Gopher23, Goplat, Gosub, Graham87, Grandsonofmaaden, Greylion, Grgan, Guanxi, Guy Harris, Gyanlakhwani, H0serdude, HCelik, Haakon, Hadal, HappyCamper, Harryzilber, Harvester, Helix84, Herberthuber, Heron, Hobophobe, Hughey, Hungrymouse, Hussein 95, Hvn0413, Iambk, Iamzemasterraf, Ibc111, Idkyididthis, Igoldste, Indefatigable, Intgr, Iridescent, Irq, Isaac Rabinovitch, Itai, Itusg15q4user, J.delanoy, J04n, Jackfork, Jakohn, JamesBWatson, JamesEG, JeLuF, Jec, Jemichel, Jhartmann, Jim.henderson, John a s, JohnOwens, JohnWittle, Johnblade, Johnnyboyshoots, Johnuniq, JonHarder, Joseph507357, Joyous!, Jtkiefer, Juansempere, Judzillah, Juliancolton, K45671, KGasso, Kakomu, KaragouniS, Karn, Kasabax, Kate.woodcroft, Kbh3rd, Kbolino, Kbrose, Keith D, KelleyCook, Kerotan, Kevmcs, Kgfleischmann, Khalad, Kinema, King of Hearts, Kjkolb, Knucmo2, Ksn, Kuru, Kvng, Kwamikagami, Kwiki, Lallolu, Lamro, Laudaka, Lavenderbunny, Lee Carre, Leebo, Leotohill, Lightmouse, Ligulem, LilHelpa, Limbo socrates, LittleBenW, LittleOldMe, Lockg, Lolpack, LordJumper, Lotu, Lovecz, Lped999, Lukith, Lumos3, Luna Santin, MER-C, MacGyverMagic, Mahjongg, Makeemlighter, Manish soni, Markjx, Martarius, Martijn Hoekstra, Matt.farina, Maury Markowitz, Mav, Mbutts, Melvinvon, Mendel, MessiFCB, Mhare, Michael Hardy, Michael Slone, Michael Thomas Sullivan, MightyWarrior, Mightyms, Mike Rosoft, Mikm, Milan Keršláger, Mindmatrix, Minesweeper, Mmiszka, Modulatum, Mortense, MoussePad, Moxfyre, Mozzerati, MrFish, MrOllie, Mrand, Mrh30, Mwanner, Myanw, N TRoPY, NJM, NawlinWiki, NeaNita, Nelson50, NewEnglandYankee, Nicolaasuni, Nightraider0, Nikevich, Nimiew, Nishkid64, Nixdorf, Nnemo, Norm, Northamerica1000, Now3d, Nroets, Nthep, Nubiatech, Nvj, Nyttend, Oakad, Ohconfucius, Ohnoitsjamie, Oneocean, OverlordQ, OwenX, Oxymoron83, Packetslinger, Palmer1973, Paul, Paul Koning, Peck123, Pekaje, Percy Snoodle, Peruvianllama, Petrb, Peyre, Pgallert, Pgan002, Phil Boswell, Philip Trueman, Piano non troppo, PierreAbbat, Pilatus, PinkMonkeys, Pjbrockmann, Pkirlin, Plasticup, Plugwash, Pluma, Pmsyyz, Praetor alpha, Prolog, Publicly Visible, QmunkE, Quarl, Quintote, RHaworth, RadioFan, Rait, Rapaporta, Rarmy, Recurring dreams, RedWolf, Rednectar.chris, Reedy, Requestion, Rettetast, Rhobite, Rholton, Rich Farmbrough, RichardBennett, Rick Sidwell, Ricsi, Rjwilmsi, Rlcantwell, RoySmith, Roybadami, Royote, Rror, Rsrikanth05, Rufus210, Ruud Koot, Rw4nd4, Rwwww, SF007, SHCarter, Salsa Shark, Sam Hocevar, Sander123, Sceptre, Schneelocke, Scott McNay, ScottJ, SeaChanger, Selmo, Setu, Sfisher, Shadowjams, Shanes, Sharkford, Shieldforyoureyes, Sietse Snel, Silenceisfoo, Sincoskie, Sligocki, SoSaysChappy, Sobelbob, SpamBilly, Spiritia, Steeev, Stephan Leeds, Stephenb, Steven Luo, Stratocracy, Stuart P. Bentley, Suruena, Svick, Swapdisk, Swarnabhra, TAnthony, Tas50, Tedernst, Template namespace initialisation script, Tempodivalse, Thaas00, The Anome, The Random Editor, The Thing That Should Not Be, The quark, The undertow, TheMoog, Thingg, Thue, TimBovee, Tiny plastic Grey Knight, Tizio, Tmontalv, TomPhil, Tonsofpcs, Tooki, Tothwolf, Tpbradbury, Transmission 1000, Trev M, Tunnie, Turgan, Tverbeek, Typo47, Tyz, UU, Ukexpat, Ulric1313, UncleBubba, Unschool, Urvabara, Useight, Utuado, Vashti, Vdm, Veinor, Versus22, Vidmes, Vijaykumar, Viridae, Vmenkov, W Nowicki, Wa2ise, WadeSimMiser, Wasisnt, Wattlesmorse, Wavelength, Wayfarer, WaysToEscape, Wbenton, Wbm1058, Wefa, WhiteDragon, Wik, WikiJaZon, Wikiborg, Wikifranz, WillAndrews, WillLord, Willy on Wheels over Ethernet, Wimt, Wingnutamj, Wk muriithi, Wmahan, Wmasterj, Woohookitty, WriterHound, Wrs1864, Wtshymanski, Xenium, Xnatedawgx, Yintan, Yudiweb, Yuvalnod, Yyy, ZenerV, Zero10one, Zhangyue, Zodon, Zoicon5, 1010 anonymous edits

Ethernet over twisted pair *Source*: http://en.wikipedia.org/w/index.php?oldid=518970266 *Contributors*: 28bytes, Aashish.59, Adrian 1001, Afed, Algotr, Amillar, Apple2, Armando, Atlaslin, AttishOculus, Avono, B4hand, Baccala@freesoft.org, Bachrach44, Benhutchings, Billatq, Binksternet, Bjdehut, Boscobiscotti, CannibalSmith, Carre, Chealer, Chetvorno, Conversion script, CraigMatthews, Crissov, Dcsutherland, Dflock, Dicklyon, Dpotter, Drtimmcguinness, E946, Ebeili, EdStauff, El C, Electron9, Epbr123, Ethernetextender, Fages, Frap, FreplySpang, Friviere, Geek2003, Giftlite, Gigglesworth, Giraffedata, Guanxi, Guy Harris, HalfShadow, Herbertxu, Hilarleo, Hwsknudsen, Indefatigable, Itai, Jakohn, Jheiv, Jim.henderson, Jmediate88, Julesd, Ka Faraq Gatri, Karn, Ken g6, Kop, Kvng, Lee Carre, Liftarn, Lightmouse, Lunaverse, Lzur, Mac, Marcel van b, MarkJL, MartinHarper, Masgatotkaca, Merovingian, Mild Bill Hiccup, Mintleaf, MitchellShnier, Ninly, Novajeeper, OrgasGirl, Pcslc, PenguiN42, Piezocuttlefish, Plugwash, Pumbaa80, Rednectar.chris, Rfl, Rhodesh, RichardBennett, Rjwilmsi, RoySmith, Royalguard11, Roybadami, Saayiit, Sarsaparilla, SeanWillard, Sephers, Sietse Snel, Sligocki, Sln3412, SteinAlive, SteinbDJ, Styrofoamcup, Talinus, Tfdb, Thatguyflint, The Anome, Thundt, Timc, Toddintr, Tothwolf, Transmission 1000, UNHchabo, UU, Uldoon, Usemyname3times, Voidxor, W Nowicki, Wikiborg, Wtshymanski, Yonidebest, Zbxgscqf, ZeroJanvier, Zxmaster, Öncel Acar, Тивероолник, 146 anonymous edits

Carrier sense multiple access with collision detection *Source*: http://en.wikipedia.org/w/index.php?oldid=504234097 *Contributors*: ABF, Abdull, Aldie, Alexius08, Althena, Autumnking, Basawala, Bernhard Bauer, Bibop528, Borgx, Boscobiscotti, Carbuncle, CatLikeThief, Costello, Datashark, Denisarona, Eastlaw, Eeekster, EmreDuran, Etu, Garybaus, Getonyourfeet, Glenn, Golgofrinchian, Grubber, I.aditya, Ibc111, Ilario, Indon, IsUsername, Jesse Viviano, Johnuniq, Kbrose, Kdau, KelleyCook, Kvng, Largoplazo, Lee Carre, LiDaobing, MER-C, Mange01, Markyleong, Martintoft, McGeddon, MessiFCB, Michaelas10, Mojodaddy, Nakon, Neutra, NickW557, Nimiew, Nkansahrexford, Nuno Tavares, Od Mishehu, Paul August, Paul Koning, Pbrandao, Piano non troppo, Plugwash, Praveentech, Prunesqualer, Quibik, R54186, Rgoodermote, Rich257, Rick Sidwell, Runtux, Rursus, Scootey, Shadowjams, Simon South, Squadri, Stdazi, SwisterTwister, The Thing That Should Not Be, Theneoindian, Tide rolls, ToastieIL, VasilievVV, Vdsabev, Velella, Vitz-RS, Vks.vivek, Waffleatron9, Whiskars1, Zac439, Zack, Тивероолник, 관인생략, 144 anonymous edits

Wi-Fi *Source*: http://en.wikipedia.org/w/index.php?oldid=521549319 *Contributors*: -Majestic-, 12dstring, 16@r, 19.158, 19.7, 220 of Borg, 2over0, 32deej, 802geek, A Softer Answer, A8UDI, A930913, ABF, AEMoreira042281, AGruntsJaggon, AL SAM, AWoodland, Aamrod, Aapo Laitinen, Abstract Idiot, Abune, Acebulf, ActivExpression, Adaclnews, Adashiel, Addihockey10, Addshore, Aeonsafe, Ahoerstemeier, Ajarmitage, Ajaysreedharan, Ajklein5211, Akerans, Akjar13, Alain r, Alan Liefting, Alansohn, Albedo, Albert109, Aldie, AlexF, AlexTiefling, Alexander UA, Alexius08, AlexiusHoratius, Algocu, AlistairMcMillan, AllanHainey, Allen Moore, Altaïr, Altermike, Alyssa hoffel, Amars, Amol.jawarkar10, Amorrow, AndersTR, Andrei Stroe, Andrew Maiman, Andy, Andy Dingley, AndyTheGrump, Andyiou52345, Andyjsmith, Andywmm9, Anna Lincoln, Anniex10, Anonymous anonymous, Anthony22, Antonrojo, Anttilk, Ardroliat, Aremith, Arichnad, Armando, Armyable, ArnoldReinhold, Arpabone, Arvindn, Asterion, Astral, Auric, Ausinha, Aveekbh, Awostrack, Awpsys, Baka toroi, Barek, BarretB, BartBenjamin, Bartman007, Baseball Bugs, Bawolff, Beavel, BelAir Networks Wireless Mesh, Beland, Belmond, Ben1220, Benhutchings, Beno1000, Berkut, Bevo, Bfigura's puppy, Bhny, Big Brother 1984, Bigbluefish, Bilbobee, BillC, Billywhack, Binkowski, Biscuittin, Blackdragon1157, Blackjewishfag, Blackroo1967, Blair Bonnett, Blooperfoob, Bluefoxicy, Bluezy, Bnordlund, Bobamnertiopsis, Bobblewik, Bobo192, Bonadea, Bongwarrior, Bornhj, Bozonz, Bratch, Brazil4Linux, Bremerenator, Briandjohnson, Brianga, BrightStarSky, Brillow, Broadband118, Brons, Brooknet, Btilm, BuddhaDharma, Bullzeye, Burnte, Bxmpls, C.Crane, C0nanPayne, C3k, CAJ, CGorman, CRH3 EMU, CUSENZA Mario, Cabalamat, Cailil, Calabe1992, Calabraxthis, Caliga10, Callidior, Calmer Waters, Caltas, Caltrop, Cambrant, Can't sleep, clown will eat me, CanisRufus, Capricorn42, CardinalDan, Carter, Casito, Catch2424, Cavanagh12345, Cbruno, Celtus, Charivari, Charles Moss, Charles Nguyen, Charm, Chaser, Chcknwnm, Chealer, Cheeesemonger, Chendy, Cherylcopeland, Chezi-Schlaff, Chitrapa, Chomperhead, Chowbok, Chris 73, Chris Q, Chrisjj, Christin varghese, Chunihan, Circeus, Classicrockfan42, ClementSeveillac, Click23, Clipdude, Closeapple, Closedmouth, Cmdrjameson, CnkALTDS, Coasterlover1994, CobraBK, Cocoaguy, Coffee, Coffin, Colenso, ColinHogben, Colinstu, Cometstyles, CommonsDelinker, Compellingelegance, Corevette, Corvus cornix, Courcelles, Cpl Syx, Crag, Crazycomputers, Crazyviolinist, Crimsondestroyer, Crissov, Cruisermjc, Cryout, Cshay, Cswilly, Cupcake1230, Cy21, CyberSkull, Cyrius, CzarB, D, DAJF, DARTH SIDIOUS 2, DMAch, DVdm, Dan Fuhry, DanPhilpott, Dancraggs, Daniel.Cardenas, DanielTahar, Dank, Danl999, Dannylim, Dark Shikari, Dark jedi requiem, Darkmaster2004, Darkwind, DarthRaider, DaveBurstein, Daven200520, Davhorn, DavidCary, Davidjk, Davidmack, Dawnseeker2000, DeVoteDz, DeadEyeArrow, Debresser, Deflective, Delirium, Dendodge, Denisarona, DerHexer, Devilalbert, Devin, Dieseldrinker, Dilane, Dilettante99, Dirkbb, Discospinster, DistopiaSPM, Djcapelis, Djg2006, Djkurtz, Dlohcierekim, Dmarquard, Dmccarty, Dmchyla, Dobie80, DocWatson42, Docboat, Dogman15, Domeilal, Doniago, DopefishJustin, Dopegirltwurk, Dovid, Doyley, Dposse, Drakcap, Dreadstar, Dreamteamone, Drewzhrodague, Drplaster1, Ducknish, Dust Filter, Dylan Lake,

DylanW, Dystopianray, Dzubint, E Wing, ESkog, EagleEye96, Earthlyreason, Eatsaq, Ecamdog, Eclectek, Ed Cormany, Ed g2s, Edcolins, Editor at Large, Eeekster, Eggnock, Egil, Ehn, Ejrs, Eken7, Elassint, Elbperle, Electro94, Electron9, Elvey, Em978, Enauspeaker, Engmark, Enviroboy, Epachamo, Epbr123, Eptin, Erencexor, Eric Kvaalen, Eric Wester, Eric-Wester, Ericyu, Eskalin, Espoo, Everyking, Evice, EvilPenguin, Excirial, Exit13, FF2010, FGrose, Falcon9x5, Faloopalump, FastLizard4, Favonian, Fbarton, Feedmecereal, Feureau, Fieldday-sunday, Fifieldt, Figueroaedgar, FilippoSidoti, Fireaxe888, Fiskbullar, Fitch, Fleisher, Fleminra, Flowerpotman, Fonzende, Fosnez, Foxj, Fractal3, Freakmighty, Frecklefoot, Freedomlinux, Freewol, Frencheigh, FreplySpang, Friginator, Ftbllfn33, Fubar Obfusco, Funandtrvl, Fuzheado, GGink, GNUtoo, GRAHAMUK, Gabbe, Gadfium, Gaius Cornelius, Galifardeu, Garion96, Gavinatkinson, Gazpacho, Gdo01, General Wesc, Genestr, Giftlite, Gilbertfein, Gilgamesh, Gilliam, Gimmetrow, Glacialfox, Glenn, GliderMaven, GoShow, Gobbleswoggler, Godfather xie, Gojomo, Gokusandwich, Golbez, Golfandme, Gomm, Goodvac, Gorank4, GozzoMan, Grafen, Graham87, Grandfeller, Grandor, Greghe, Gregherlihy, Grim4593, Ground Zero, GroveGuy, Groxx, Guaka, Gulliveig, Gurch, Gutzmer, Guy Harris, Guycarmeli, Guðsþegn, Gwalarn, Gxti, Gyll, Gyrferret, H2g2bob, Hackfreewifi, Hadal, HaeB, Hagrinas, Hairy Dude, Halfbreath, Hankwang, Hanseichbaum, Haoie, Happenstancial, Happinessiseasy, Happysailor, HarisM, Harryzilber, Hawaiiboy99, Hede2000, Hedgehoglet, Helix84, Heman, HenkeB, Hera1aphrodite, Heron, Hertz1888, HexaChord, HiddenInPlainSight, Hillcrest, Holierthanthou, Holo16, Homerjay, Hsan22, Ht1848, Hu12, HybridBoy, Hydrargyrum, Hydroxides, I already forgot, INkubusse, IRP, IReceivedDeathThreats, Iainelder, IceKarma, Iflipti, Immunize, Impulse9, Inbamkumar86, Incompetence, Inkybutton, Inter16, Into The Fray, Intractable, Iridescent, Itai, Itusg15q4user, Ixfd64, Izzysanime, J.delanoy, JD554, JFreeman, JLaTondre, JOSamsung, JPMcGrath, Jafeluv, Jakohn, James Galloway, JamesAM, JamesBWatson, Jamesontai, Jamyskis, Jandalhandler, Janizary, Jarble, Jaryth000, Jason Recliner, Esq., Jason Stormchild, Jauricchio, Javawizard, Jaxsonjo, JayW, Jayharish, Jbowdenhm, Jbrandon2012, Jc4p, Jdlankin, Jeff G., Jeffrey Mall, Jeh, Jehochman, JeremyA, Jerome Charles Potts, Jesse Viviano, Jgamleus, Jh51681, Jhenderson777, Jim.henderson, Jingraham, Jjensen347, Jll544, Jlmerrill, Jmlk17, Jnavas, Jncraton, JoGusto, JoanneB, John, John Mash, John254, JohnCD, JohnCub, Jon787, JonHarder, Jondel, JordoCo, Joseph Solis in Australia, Jossysayir, Joy, Joycloete, Jp78450, Jpatokal, Jprg1966, Jschnur, Jtalledo, Juanscott, Jukeboxlord, Juliano, Justinko, Jvanthou, Jw21, Jwissick, KD5TVI, KGasso, Kantmorie, Katieh5584, Kautoorikrishna9, Kbrose, KeesCook, Keilana, KelleyCook, Kenyon, Kev19, Kevin chen2003, Kevinlie10, Kfluck, Kgopinath, Khalid Mahmood, Kikadeek, Kimiko, Kimvr, Kingboyk, Kingpin13, Kixy, Klingoncowboy4, KnowledgeOfSelf, Koman90, Kostisl, Kotra, Kozuch, Krellis, Kuru, Kvng, Kwamikagami, Kynn, Kzzl, Labboy, Laghlagh, Landroo, Larry laptop, LarryLACa, Laudaka, Lawnchair, Lawrence Cohen, Lcabanel, Le Fou, LeaveSleaves, Leendert, Leif, LeoDV, Leviwack, Lexor, Liam.winder@bozii.net, Liao, Lightmouse, Lights, Lilac Soul, Linesh1987, Linkspamremover, Lino Mastrodomenico, Lisatwo, Little Professor Stonecold, Little-man, Liyang, LizardJr8, Llort, Lmatt, Logan, Logical Cowboy, Logictheo, LordHector, LordOfer, Lucanos, Luci Sandor, Luis007cruz, Luk, M5, M7, MARQUIS111, MCB, MER-C, MLauba, MSTCrow, Mac, Maclaine, Madd the sane, Magioladitis, Magister Mathematicae, Mahendra, Mailer diablo, Mammad2002, Markaci, Markdr, Marketdiamond, Markpeak, Martinhayes, Master737373, Materialscientist, Matt Whyndham, Maxxdxx, Mazarin07, Mboverload, Mbw227, McGeddon, McSly, Mccombs1982, Mckenzie1995, Mcmarmatt, Me Three, Meco, MedicineMen, Meekywiki, Mehrunes Dagon, Melah Hashamaim, MennoMan, Merlion444, Metaclassing, Mgnbar, Mia.bo, Michael Hardy, Michael93555, MichaelMaggs, Michaelsbll, Microfrost, Mifter, Miguel Escopeta, Mike Moreton, Mike.lifeguard, MikeFenney, MikeM8892, Mikemoral, Mindmatrix, Miranda, Miremare, Mirko.fidisk.fi, Misternuvistor, Mjmarcus, Mkzz, Mmdoogie, Mmmcatsoup, Moarmudkipzlulz, MobileMistress, Modulatum, Moehrchen, Mollykate82, Momo san, Monaarora84, Monkeyman, Monochnotos, Monuko, Moreati, Mormegil, Mortense, Mothmolevna, Mowgli, Moyda, Mrmiscellanious, Mshahbaig, Msm20032003, Mtonkin, Mumia-w-18, Mushroom, Muzzle, Mwanner, Mygerardromance, Mytimekept, Namenad, Nankeyman, Nasa-verve, Nbarth, Ncmvocalist, Ndyguy, Nealcardwell, NellieBly, Nelson50, NeoChaosX, Nerminbarman, Networkingguy, Neurolanis, Niemeyerstein en, Night Gyr, Nil0lab, Niri.M, Nixdorf, Njh@bandsman.co.uk, Noah Salzman, Noctibus, Nopetro, Notedgrant, Novalis, Nowcrash, Nposs, Nsaa, Nux, OPless, Obastani, Octahedron80, Ohnoitsjamie, Oldr4ver, Omegatron, Omhafeieio, Omicronpersei8, Onorem, Otivaeey, Oxymoron83, Oystein, Ozseden, Pacific1982, Paddu, PamD, Parrotheadmjb, Parsmutaf, ParticleMan, Patrick, Patrick Fisse, Patstuart, Paul Weaver, Paulshanks, Peak, Pekaje, Pelago, PeregrineAY, Peter, Peter McGinley, Peterl, Petrb, Petri Krohn, Peyre, Phatom87, Phi beta, Phil-welch, PhilKnight, Philip Trueman, Phlegat, Phooto, Photocopier, Phreakuency, Piano non troppo, Pie4all88, Pieleric, Piet Delport, Pimfig, PinkPig, Pip2andahalf, PleaseInsertGirder, PoTi, Pokipsy76, Pol098, Polaralex, Popescucalin, Porschedriver403, Preslethe, PrestonH, Primary0, ProhibitOnions, Project2501a, Pt, Public Menace, Puggs, Pujanmalla, Puneet.kasera, Quantpole, Qui1che, Quintote, QuiteUnusual, Qutezuce, R6144, RA0808, RDOlivaw, RHaworth, Radagast, Radiant chains, RadioActive, Raj bhinde, Rannpháirtí anaithnid, Ravendrop, Ravenperch, Raxian, Ray Radlein, Raysonho, Raywil, Rcassidy, Rce-revo, Reach Out to the Truth, Rearden9, Recognizance, Reconsider the static, Recurring9, Rednectar.chris, Rees11, Regara MkII, Reisio, Rememberway, Renewal36, RenniePet, Requestion, Res2216firestar, Retired username, Retodon8, Rewt, Rfc1394, Rhobite, Riadlem, Ribread2, Rich Farmbrough, Rich257, Richwales, Ricksterrr11, Riddz17, Ripetom, Risk one, Riskariandita, Rjanag, Rjhatl, Rjwilmsi, Rkopplin, Rmallins, Robartin, Robert Cassidy, Robertvan1, Robo56, Roccastudios, RockOfVictory, Rogerbrent, Rojypala, Romanski, Rootxploit, Roscelese, RoseTech, Roybadami, Rpt0, Rror, Rsrikanth05, Rtrohantyagi, RufusThorne, Rumping, RupertMillard, Russoc4, Ryan Postlethwaite, RyanCross, Rynsaha, Ryuch, S kollmor, S.K., S.ferguson, SEWilco, SF007, SHCarter, SMC, SPANGER BANGER, STGM, Saikiri, Sailor iain, Salty!, Sam Hocevar, Sam Korn, Sandeepreddyus, Sander123, Sandstein, Sango123, Sapnamis471, Sastrycpps, Satishchitti02, Saxtonrob, Sbenton, Sbmeirow, Sburpeeduncan, Sceptre, Scholia, SciCorrector, Sckirklan, Scurless, Seadood90, Seajay, Sean Reynolds, Sean2074, Seancoady, Seans Potato Business, Seaphoto, Searchme, Secfan, Seikku Kaita, Seqsea, Sesu Prime, Setveen, Sevenneed, Shanafme, Shandris, Shinpah1, Shirik, Shorty114, Shuipzv3, Shunt010, Sikon, Silverxxx, Silvery, Simm, SimonP, Simonn, SineWave, Sixteen Left, Sjjupadhyay, Skatebiker, Skategem, Skierpage, Skor, Slippyd, Slon02, Slowking Man, Smailis, Smalljim, Smoke, Snori, Snowolf, SoCalSuperEagle, Soap, Socrates2008, Sodapop89, Softlavender, Sonam.bhutia, SonicBlue, Soumyasch, Souseiseki42, SpacedOut, Spathak112, Spearhead, Speck-Made, Spitfire19, Spookee, SpuriousQ, Sraraja, Srborlongan, Srikeit, Ssd, Ssri1983, Stangaa, Star Trek Man, StaticGull, Stephan Leeds, Stephen, Stephenb, Stevenj, Stickee, Styrofoam1994, Suid-Afrikaanse, Sunray, SuperDude115, Superborsuk, Supermanden, Superway25, SusanLesch, Susvolans, Suwa, Svetovid, Swizman, T callahan, T h e M a v e r i c k, Tabledhote, Tainter, TakuyaMurata, Taralsoni, TastyPoutine, Tbhotch, Tbkexile, Tcncv, Teapeat, Teddybearspicnic, Teixant, Tekinera, Teles, Texaswebscout, Texture, Tfl, Thatguy553, The Belgain, The Bryce, The Mark of the Beast, The Rambling Man, The Thing That Should Not Be, The Utterly Annoying Pedant, The undertow, The wub, TheGWO, TheLiberalTruth, TheRealFennShysa, TheTito, TheWishy, Thentukiran, Thepidding, Thetasig, Theymos, Thiago R Ramos, Thingg, Think outside the box, Think smith, ThrashedParanoid, Thunderbrand, Tide rolls, TigerShark, Tim1988, Timbatron, Timeshift9, TippyGoomba, Tiptoety, Tjrana0, Tntnnbltn, Tobiaslw, Toccatafugue, Tohd8BohaithuGh1, Tokachu, Tom.k, Tommy2010, Tony Sidaway, Tony1, Topazg, Tpbradbury, Tree Biting Conspiracy, Tri400, Trlkly, Trusilver, Tsange, Tsenapathy, Ttiotsw, Tucker001, TucsonDavid, Twang, Twasono, Twocs, Twp, Tyler, Tyransc7, UU, Ultraman2008, Umerqureshi, Unprovoked, Unused0030, Useight, Utopianfiat, Valfontis, Valley2city, Vedantm, Vegaswikian, Veledan, Verbal, Versageek, Versus22, Vgautham 91, Viajero, Vibhijain, Vicarious, Vickymesingh, Viktor Laszlo, Vinaywin7, Vishteen, VodkaJazz, Vossman, Vrenator, W Nowicki, WLRoss, Waggers, Wahrmund, Walloon, Walter Görlitz, Wandab12, Warwickcaddie, WatermelonPotion, Watkinsclass67, Wavelength, Wdfarmer, Webwizard, Wehe, Wellithy, Wellivea1, Wendell, Wep, Wereon, West.andrew.g, Whitepaw, Wifieverywhere2007, Wikcerize, Wikipedian314, Wikipelli, Wikiuser100, Wikizeta2, William Avery, WilliamH, Wilton gorske, Wimt, Wingspantt, Wizardist, Wjejskenewr, Wknight94, Wo.luren, Wog7777, Wolfkeeper, Woohookitty, Wormeyman, Wrrr, Wskish, Wtshymanski, Ww, Wysprgr2005, XSSX, Xmnemonic, Xtefan, Xtifr, XxxFilipks, Yosri, Your wasted elf, Yuhong, ZKPilot, Zach Vega, Zack Holly Venturi, Zarherif, Zeroshell, Zgreycoat, Zzuuzz, ᎤᏍᏗ, ᏣᏁᎭ, පසිඳු කාවින්ද, とある白い猫, 滑天下之大稽, 2912 anonymous edits

IEEE 802.11 *Source*: http://en.wikipedia.org/w/index.php?oldid=521061989 *Contributors*: 130.94.122.xxx, 16@r, 28bytes, 2mcm, 802dot11n, 802geek, AboutWeezer, Academic Challenger, Acprisip, Adrians, Afed, Ahoerstemeier, Ahull, Aij, Ajhuang-wiki, Alarob, Alcksmth, Algocu, Alinor, Allstrak, Almamun, Alphax, Amaelzer, Anastrophe, Andres, Andros 1337, AngryBear, Ankur, Anárion, Apoyon, Arendedwinter, ArielGold, Armando, Armin76, ArnoldReinhold, Arrenlex, Arthena, Asbestos, Asocall, Astronautics, Asunasun, Ausinha, Austin Hair, Avargasm, AxelBoldt, Axl, Ayudante, Barbjfox2000, Bbabul01, Bcorr, Beland, Bento00, Bibrydo, Biot, BishopNight, Bluemoose, Bluerasberry, BoKu, Bobblewik, Bobo192, Boobytrapped, Bpdlr, Briandjohnson, Brianski, Bsilverthorn, BurntBiscuits, CAPITALidea, CWii, Callidior, Caltas, Castien, Catgut, Cburnett, Cdc, CecilWard, Cecilyen, Certz, Cffrost, Cgriffiths, Chiefcoolbreeze, Chomperhead, Chris81w, Cimarron, CityOfSilver, Ckujau, Cleared as filed, Coasterlover1994, Conversion script, Cowgod14, Cpl Syx, Crh0872, Crimson30, Crissov, Ctbolt, CyberSkull, Cybercobra, Cyrius, Dafocus, Dake, Dancter, Danie.brink, Danski14, DarwinE, David R. Ingham, DavidAndersen, Dawnseeker2000, Dbenbenn, Ddxc, Defrector, Dejvid, Denisarona, DerHexer, Derek Ross, DirkvdM, Discospinster, Dj245, Djg2006, Dobz116, DocWatson42, Docu, Dogcow, Dqeswn, Drewzhrodague, Drjt87, Dsheffie, Dtwitkowski, Dwheeler, Dzubint, Eadric, EagerToddler39, EagleEye96, Eatrains, Edderso, Editor B, Edokter, Ehn, Elkman, Engineerism, Enjoi4586, Epbr123, Erencexor, Ericyu, Eth01, Europrobe, Everyking, Evgeni Sergeev, Evilspoons, Excirial, ExportRadical, FF2010, Faramir 27, Fennec, Fernandopabon, Fijal, Fleminra, Fosnez, Frecklefoot, Freddyzdead, Fubar Obfusco, Fudoreaper, GCW50, GFellows, Gaelen S., Garth 187, Gauthierm, Gavinito, Gbleem, Gene Nygaard, Geppy, GianlucaCiccarelli, Giftlite, Gilesmorant, Glenn, Glennf, GoingBatty, Goodput, Gorkish, GraemeL, Graham87, Grayshi, Gutza, Guy Harris, Hairy Dude, Heegard, Hellisp, Hetar, Hoho, Hotdog41695, Hqb, Hrhsoleil, Huru179, Hvn0413, Iapetus, Ibair2306, Infinoid, Intgr, Iriseyes, Isaac Dupree, Itai, Itusg15q4user, Ixfd64, J.delanoy, JLaTondre, JPLeRouzic, Ja 62, Jamesday, Jamesscho, Jamessungjin.kim, Janzert, Jdthood, Jesse Viviano, Jessel, Jiang, Jnavas, JocelynDelalande, Joerg Reiher, John Smith 104668, John of Reading, Johnteslade, Jonv112, Jonverve, Joshua Scott, Jsalcedo, Juanjohn, Julesd, Justjohnny, KYSoh, Ka-Ping Yee, Kaihsu, Karada, Karenjc, Karlshea, Karn, Kbrose, Kc7rad, KelleyCook, Kelly Martin, KevinDorekens, Kgfleischmann, Kgrr, Kharker, Khatru2, Khukri, Kingsley16, Koman90, Kord, Kozuch, Kristen Eriksen, Kunz506, Kurt Jansson, Kvng, Kyle Barbour, LA2, LOL, La goutte de pluie, LachlanA, Lag10, Lakers, Lambiam, Lambyte, Larry V, Lawrence Cohen, Le Sage, Lee Daniel Crocker, Lee J Haywood, Leotohill, Lfwlfw, Liebeskind, Lightmouse, LittleDan, LittleOldMe old, Llort, Lmatt, Loraan, LordJumper, LuisVilla, Lzur, M1 essam, MK8, Mailer diablo, Mange01, Martin Blank, Martopg, Materialscientist, Matt Crypto, MattSH, Mattsday, Mauro Bieg, Mav, Mebden, Mentifisto, Mgolden, Mhannigan, Michal.feix, Mifter, Mike Moreton, Mike moreton, Mike1024, Mikeblas, Mikemurphy, Mikeygnyc, Mindbuilder, Mindmatrix, Minkus, Minna Sora no Shita, Mitsuhirato, Mjmarcus, Mjrichardson1, Mkeating24, Mkrist, Mmccalpin, Mmx1, Modster, Mortein, Mr. Zarniwoop, Msaada, Msauve, Mulad, MureninC, Muukalainen, Mwarren us, NJA, NailPuppy, Nakon, Nasa-verve, Navinveenu, Nbarbettini, NetRolller 3D, Nick, Nickshanks, Nicoli nicolivich, Nido, Nikita Borisov, Nil0lab, Ninly, Njh@bandsman.co.uk, Nnemo, Nneonneo, Nothlit, Nuance13x, Nuno Tavares, Octahedron80, OlEnglish, Olivier, Omegatron, Onsly, Opelio, OpenCommunications, Oriondriver, Orrc, Oscabat, OverlordQ, PacoBell, Panarchy, Pandikarthikanmani, Patrick, Paul.j.richardson, Paulc206, Pbyeoh, Pegship, Perey, Peruvianllama, Peter S., Peyre, Pgan002, Phandel, Phantomdj, PhilHibbs, Philip Trueman, Philthecow, Phoebe, Plasticup, Plugwash, Poppafuze, Project2501a, Proski, PseudoSudo, PsyberS, Pt, Puffin, QuantumEleven, Quasipalm, Qwertyshan, R6144, RJHall, Rabbeinu, Radioraiders, Radiosband, Rapomon, Rebel, Recognizance, RedWolf, Rednectar.chris, Renegadeviking, RexNL, Rfl, Rhobite, Rich Farmbrough, Richwales, Rick Sidwell, Rjairam, Rjwilmsi, Rlevse, Roadrunner, RobNich, Robert K S, Robert Merkel, Robertoalencar, Ronz, Rossami, Roybadami, Rprpr, Rrburke, Rsm99833, Rufous, S, S.smith.1-398, S1N3d dW17, SSTwinrova, Sa.vakilian, SamJohnston, Samuella, Sandeshgoel, Sandos, Sandox, Sargant, Sarpkaya92, Sathakselva, Sceptre, Schapel, Sclm, Scohoust, Seaneseor, Sfoskett, Shadowjams, Shanafme, Shanel, Shaper252, Shibboleth, Shoveldude, Sietse Snel, Signalhead, Sintaku, Sir Lothar, Skor, Skybon, Skylercall, Slearl, Slitetech, Slowking Man, Slrobertson, Soap, Solarisphere, Some jerk on the Internet, Soumyasch, Soupmix, SpK, Speciou5, Speculatrix, Spelemann, Spitfire, Splash, Spymanut, Srleffler, Srvmdl, Stefcoetzee, Stefpap, Stephan Leeds, Stephen Bain, Stephen Gilbert, Steve03Mills, Stewarbd, StrengthOfNations, Subharanjan, Suruena, Swiveler, Syndicate, Synthetik, Sysin, TRauMa, TakuyaMurata, Tarquin, Tekeyman, Tgritchie, The Anome, The Epopt, The Thing That Should Not Be, TheGerm, TheWeakWilled, Thenickdude, Thine Antique Pen, Think outside the box, Thisisborin9, Thue, Tide rolls, Timl2k4, Todd Vierling, Toe Rag, Tomh009, Tonsofpcs, TonyW, Travelingseth, Treekids, Trenton11tgs, Tsunaminoai, Twthmoses, Twyaii, Uncle Milty, Unused0025, Vanished user psdfiwnef3niurunfiuh234ruhfwdb7, Vaughan Pratt, VdSV9, Vegaswikian, Velella, VerticalAsymptote, Vipintm, Vlad, W Nowicki, WCat, Warren, Waveguy, Wavelength, Waveletrules, Wavyriver, Widefox, Will Beback, Willy on Wheels over Ethernet, Winterspan, Wphilipw, Wrs1864, Wtshymanski, WulfTheSaxon, Ww, Wysprgr2005, X-Destruction, XSG, Xaosflux, Xichael, Xmnemonic, Y2kboy23, Yamaguchi先生,

Yandman, ZeroOne, Zippy, Zr2d2, Ŷŭmřĩ5ė, 1622 anonymous edits

Image Sources, Licenses and Contributors

File:Flood.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:Flood.gif *License*: unknown *Contributors*: SuperFLoh

File:FloodAck.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:FloodAck.gif *License*: unknown *Contributors*: Bukk, Jahobr, SuperFLoh

File:4x4 grid spanning tree.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:4x4_grid_spanning_tree.svg *License*: Public Domain *Contributors*: David Eppstein

Image:Cayley's formula 2-4.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Cayley's_formula_2-4.svg *License*: unknown *Contributors*: User:Rocchini

Image:Spanning tree protocol at work 1.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Spanning_tree_protocol_at_work_1.svg *License*: Creative Commons Attribution 3.0 *Contributors*: GhosT

Image:Spanning tree protocol at work 2.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Spanning_tree_protocol_at_work_2.svg *License*: Creative Commons Attribution 3.0 *Contributors*: GhosT

Image:Spanning tree protocol at work 3.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Spanning_tree_protocol_at_work_3.svg *License*: Creative Commons Attribution 3.0 *Contributors*: GhosT

Image:Spanning tree protocol at work 4.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Spanning_tree_protocol_at_work_4.svg *License*: Creative Commons Attribution 3.0 *Contributors*: GhosT

Image:Spanning tree protocol at work 5.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Spanning_tree_protocol_at_work_5.svg *License*: Creative Commons Attribution 3.0 *Contributors*: GhosT

Image:Spanning tree protocol at work 6.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Spanning_tree_protocol_at_work_6.svg *License*: Creative Commons Attribution 3.0 *Contributors*: GhosT

Image:Dijkstra Animation.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:Dijkstra_Animation.gif *License*: Public Domain *Contributors*: Ibmua

Image:Dijkstras progress animation.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:Dijkstras_progress_animation.gif *License*: Creative Commons Attribution 3.0 *Contributors*: Subh83

Image:BGP FSM.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:BGP_FSM.svg *License*: GNU Free Documentation License *Contributors*: Johannes Rössel (talk)

File:BGP Table growth.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:BGP_Table_growth.svg *License*: Creative Commons Attribution-Sharealike 3.0,2.5,2.0,1.0 *Contributors*: Mro

File:Internet AS.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Internet_AS.svg *License*: Creative Commons Attribution-Sharealike 3.0,2.5,2.0,1.0 *Contributors*: Mro

Image:Ethernet RJ45 connector p1160054.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Ethernet_RJ45_connector_p1160054.jpg *License*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors*: User:David.Monniaux

File:Loudspeaker.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Loudspeaker.svg *License*: Public Domain *Contributors*: Bayo, Gmaxwell, Gnosygnu, Husky, Iamunknown, Mirithing, Myself488, Nethac DIU, Omegatron, Rocket000, Shanmugamp7, The Evil IP address, Wouterhagens, 23 anonymous edits

File:10Base5transcievers.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:10Base5transcievers.jpg *License*: Creative Commons Attribution-Sharealike 2.5 *Contributors*: Original uploader was Robert.Harker at en.wikipedia

Image:Network card.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Network_card.jpg *License*: GNU Free Documentation License *Contributors*: User:Nixdorf

File:Network switches.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Network_switches.jpg *License*: unknown *Contributors*: ShakataGaNai

File:Coreswitch (2634205113).jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Coreswitch_(2634205113).jpg *License*: Creative Commons Attribution-Sharealike 2.0 *Contributors*: Dave Fischer

File:10baseT cable.jpeg *Source*: http://en.wikipedia.org/w/index.php?title=File:10baseT_cable.jpeg *License*: Public domain *Contributors*: Denniss, GreyCat, Pamri, Super1, Thryduulf, W Nowicki, WikipediaMaster, Yarl

File:10baseT jack.png *Source*: http://en.wikipedia.org/w/index.php?title=File:10baseT_jack.png *License*: Public Domain *Contributors*: Original uploader was Dflock at en.wikipedia Later versions were uploaded by Zcrayfish at en.wikipedia.

Image:Rj45plug-8p8c.png *Source*: http://en.wikipedia.org/w/index.php?title=File:Rj45plug-8p8c.png *License*: Creative Commons Attribution-ShareAlike 3.0 Unported *Contributors*: Aaron Kaase

Image:Wire white green stripe.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_white_green_stripe.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:Wire green.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_green.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:Wire white orange stripe.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_white_orange_stripe.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:Wire blue.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_blue.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:Wire white blue stripe.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_white_blue_stripe.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:Wire orange.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_orange.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:Wire white brown stripe.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_white_brown_stripe.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:Wire brown.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wire_brown.svg *License*: GNU General Public License *Contributors*: Pumbaa80, Rocket000, WikipediaMaster

Image:CSMACD-Algorithm.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:CSMACD-Algorithm.svg *License*: Public Domain *Contributors*: Runtux

Image:PD-icon.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:PD-icon.svg *License*: Public Domain *Contributors*: Alex.muller, Anomie, Anonymous Dissident, CBM, MBisanz, PBS, Quadell, Rocket000, Strangerer, Timotheus Canens, 1 anonymous edits

File:Wi-Fi Logo.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wi-Fi_Logo.svg *License*: Public Domain *Contributors*: Wi-Fi Alliance

File:Au_wifi.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Au_wifi.jpg *License*: Creative Commons Zero *Contributors*: User:Jr223

File:Metro Wireless Node.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Metro_Wireless_Node.jpg *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Robo56

File:WiFi-detector.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:WiFi-detector.jpg *License*: Public Domain *Contributors*: Raysonho@Open Grid Scheduler

File:RouterBoard 112 with U.FL-RSMA pigtail and R52 miniPCI Wi-Fi card.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:RouterBoard_112_with_U.FL-RSMA_pigtail_and_R52_miniPCI_Wi-Fi_card.jpg *License*: GNU Free Documentation License *Contributors*: Kozuch

File:3GN .jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:3GN_.jpg *License*: Public Domain *Contributors*: Original uploader was Rewt at en.wikipedia

File:EBWifi.JPG *Source*: http://en.wikipedia.org/w/index.php?title=File:EBWifi.JPG *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: highwycombe (talk)

File:Wireless adaptor USB.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wireless_adaptor_USB.jpg *License*: Creative Commons Attribution 3.0 *Contributors*: Ben Ben, Silverxxx, 1 anonymous edits

File:Ezurio wism2 small.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:Ezurio_wism2_small.jpg *License*: GNU Free Documentation License *Contributors*: Phooto, 4 anonymous edits

Image:WRT54G v2 Linksys Router Digon3.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:WRT54G_v2_Linksys_Router_Digon3.jpg *License*: unknown *Contributors*: "Jonathan Zander (Digon3)"

Image:2.4 GHz Wi-Fi channels (802.11b,g WLAN).svg *Source*: http://en.wikipedia.org/w/index.php?title=File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).svg *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: Michael Gauthier, Wireless Networking in the Developing World

Image:NonOverlappingChannels2.4GHzWLAN-en.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:NonOverlappingChannels2.4GHzWLAN-en.svg *License*: Creative Commons Attribution 3.0 *Contributors*: Liebeskind

File:2.4 GHz Wi-Fi channels (802.11g WLAN).svg *Source*: http://en.wikipedia.org/w/index.php?title=File:2.4_GHz_Wi-Fi_channels_(802.11g_WLAN).svg *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: 2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).svg: Michael Gauthier, Wireless Networking in the Developing World derivative work: MarkWarren (talk)

License

Creative Commons Attribution-Share Alike 3.0 Unported
//creativecommons.org/licenses/by-sa/3.0/