

IGX/BPX AutoRoute White Paper

Document ID: 10752

Introduction

Before Routing Occurs

Events that Trigger the Routing Process

Select the Best Route in a Flat Network

Select the Best Route in a Tiered Network

Secure the Route

Select the Order in which Connections Are Routed

Other Routing Features

Routing Software Features

Other Routing Algorithms

- Centralized Routing Intelligence

- OSPF

- Dijkstra's Algorithm

Reroute Performance

Network Tuning

NetPro Discussion Forums – Featured Conversations

Related Information

Introduction

This document explains how routing is performed within a Cisco IGX/BPX network running software releases 8.4 and 8.5. This document describes the routing algorithm itself (along with its strengths and weaknesses), compares other routing schemes, and provides guidelines for optimizing network performance.

From a routing perspective, both the IGX and BPX run the same software and behave identically. Each node in a network, whether IGX or BPX, is fully aware of the current network topology and state, including trunk alarm conditions and the amount of traffic routed over each trunk in the network. This allows routing decisions to be made in a fully distributed manner, provides for rapid re-route times, and allows multiple connections to be routed simultaneously. In addition, because network congestion and loading are incorporated into the route selection algorithm, the network is able to guarantee that delay and Quality of Service (QoS) requirements are satisfied.

Before Routing Occurs

IGX/BPX routing algorithms are predicated on the assumption that each individual node in the network has all the information necessary to make routing decisions at the moment a decision needs to be made. As a result, most of the work involved in selecting a route is performed before the actual re-route event takes place. This work takes the form of inter-node messages that allow the nodes to tell each other about changes in the network. These messages, or updates, are roughly grouped into two categories:

- Static trunk attributes endpoints, trunk creation/deletion, configuration attributes.
- Dynamic trunk state alarm state, current loading.

Static trunk attributes updates refer to information about the line that is configured manually and changes infrequently. These updates are sent when a line is added to or deleted from the network, or when the user re-configures the trunk in some way. Configuration items of interest to the routing algorithm are:

- Line speed (for example: T1, E1, T3, fractional T3)

- Line encoding (connections may be restricted from routing on Zero Code Suppression (ZCS) lines)
- Statistical reserve
- Endpoint nodes
- Satellite vs. terrestrial
- Ability to support Frame Relay and/or ForeSight (some older model NTCs and TXRs may not have the correct firmware installed)

Dynamic trunk state updates refer to information that changes dynamically:

- Alarm state An update is sent whenever a trunk transitions into or out of alarm.
- Current load Each time a connection is routed, an update is sent to every node in the network that describes the allocated bandwidth on each trunk in that node. This allows other nodes to calculate the amount of open (free) space on each trunk in the network.

By sending these updates to all nodes in a timely manner, each node has the information necessary to make intelligent routing decisions for those connections for which it is responsible.

Events that Trigger the Routing Process

Each node maintains a list of connections that need to be routed. A software state machine runs continually in each node, trying to route connections on the list. Although this list is usually empty, two types of events cause connections to be placed on the list:

1. Configuration changes made by a user, such as:
 - ◆ Add a connection
 - ◆ "Ups" a downed connection
 - ◆ Reconfigures a connection to need more bandwidth than is available on the current route
 - ◆ Configures a preferred route for a connection that does not match the connection's current route
 - ◆ Manually initiates a re-route
 - ◆ Deletes a trunk from the network and the trunk is carrying connection traffic
 - ◆ Increases the amount of statistical reserve on a fully-loaded trunk
2. Failures affecting inter-node trunks, such as:
 - ◆ A trunk line alarm is declared (for example, RED alarm)
 - ◆ A trunk line card fails and no backup card exists
 - ◆ An entire node fails (power loss), causing all of its trunks to fail
 - ◆ The communication failure test (comm-fail) fails on a trunk

Select the Best Route in a Flat Network

If the user has configured a preferred route, and if the trunks specified in the route are not in alarm and have sufficient open space for the connection, that route is used. Otherwise, the routing software selects the shortest path with available bandwidth. Here is a brief description of the route selection algorithm:

1. Examine all possible one-hop routes leaving this node.
2. Ignore any trunks that this connection is restricted from using (for example, satellite or ZCS).
3. Ignore any trunks that have insufficient open space for this connection.
4. If the resulting list of one-hop routes allows us to reach the remote endpoint node and more than one feasible route exists, use the tie-breaker (described at the end of this section) then secure the route.
5. If the resulting list of routes allows us to reach the remote endpoint, but only one feasible route exists, secure the route.

6. If we have examined all possible routes and no feasible route exists, mark this connection as "Failed – no route found" and try again later.
7. If there are no one–hop routes that allow us to reach the remote endpoint node, examine all possible two–hop routes.
8. Go to step 2.

If more than one feasible route exists, the tie–breaker selects the least congested route. To accomplish this, each hop of the candidate routes is sorted by the amount of open space. This allows the most congested trunk in each route to be identified. The route with the most open space on its most heavily loaded trunk is then selected. Another way to describe this is to select the route with the least restrictive "choke point."

Select the Best Route in a Tiered Network

Tiered networks were introduced in release 8.0 as an alternative approach to building large networks. The main concept behind tiered networks is to construct high–capacity node clusters at primary Points–of–Presence (POPs) and place smaller capacity nodes at secondary and tertiary POPs. In a tiered network, a node is identified as either a routing node or a non–routing node.

Non–routing nodes (or feeders) are usually co–located with a routing node and are unaware of the presence of other nodes in the network, network topology, and so on. Routing nodes behave exactly as before, but are also responsible for selecting routes for connections that terminate on any feeder nodes that they are responsible for. An alternative is to think of non–routing nodes as feeder nodes into a hub, or routing node.

For example, a number of IGXs can be designated as non–routing nodes and connected to a co–located BPX acting as a routing node in a large POP, while other IGXs may act as routing nodes in smaller POPs. This allows a large, high–capacity network to be built without necessarily having a large number of routing nodes.

Route selection in a tiered network consists of up to three components:

1. Feeder to local hub
2. Local hub to remote hub
3. Remote hub to remote feeder

Step 2 is performed as in a flat network. The hub nodes are aware of the network state and topology, and select the shortest viable route. Steps 1 and 3 are less of an issue if one assumes that the hub and feeder are co–located. In this case, either the user or the network management system selects which hub–feeder trunk should be used.

Secure the Route

Now that a viable route has been identified, the routing software must allocate resources to secure the route. This entails determining a cell address (VPI/VCI, for example) that works along the entire route, allocating bandwidth on each trunk of the route, and informing other nodes of the new state of the network. While these actions are being performed, we must obtain "locks" that prevent conflicts with other nodes that may be routing their own connections.

The first step in securing a route is to lock each node on the route. A lock request is sent to each node. If the request succeeds, that node becomes a "slave" and will not honor lock requests from any other node until the lock is released. If the lock request is refused, this is called a routing "collision" and indicates that some other node is attempting to route a connection via the same node(s). The routing node frees any locks it may have obtained and then pauses for a time before trying again. The amount of time the node pauses increases with each consecutive collision, and is factored by a random amount to avoid oscillation. The actual formula for the collision "back–off" time is:

1. On the first collision, sleep for a random amount of time between 0 and 1 second.
2. On the second collision, sleep for a random time between 0 and 2 seconds, and so on, up through the fifth consecutive collision, when the wait time is a random interval between 0 and 5 seconds.
3. On the sixth collision, wait for a random time between 0 and 6 seconds, but add an additional time equal to the node number times 0.5 seconds.
4. On the seventh collision, wait for 0 to 7 seconds plus an additional time equal to the node number times 0.5, and so on.

The probability of collisions depends on:

- The number of nodes attempting to route connections at the same time.
- The number of diverse routes that allow multiple nodes to route simultaneously without colliding.

Optimizing to avoid collisions is complex and depends on the topology of the network. One method is to reduce the number of "master" nodes, which reduces the number of nodes that can attempt to route connections simultaneously. For example, in a hub-and-spoke topology, having all connections mastered at the hub node ensures that no collisions will occur. However, in a mesh network topology, an argument can be made to distribute mastership as much as possible to allow simultaneous, parallel routes to be secured.

Once locks are obtained, the next step is to program all the cards involved in passing data along the route. The master node sends messages to each node along the route, directing them to program the trunk card with the appropriate cell address. Additional messages are sent to the interface cards in each endpoint node to activate the connection, program the cell header, and so on.

Once each node along the route has been properly configured, the master node frees all locks and the process begins again with the next connection.

Select the Order in which Connections Are Routed

The previous sections describe how the list of connections to be routed is formed and the steps taken to route an individual connection. Since multiple connections may need to be routed, it is important to understand the order in which the IGX/BPX routes connections. First, connections are sorted by class-of-service (COS). A connection's COS is configured when the connection is first added, and can range from 0 to 15, with 0 being the highest priority. Sorting the list of connections in COS-order ensures that the most important connections are routed first. Secondly, connections with identical COSs are sorted from highest to lowest bandwidth, with higher bandwidth connections being routed first. This allows bandwidth-intensive connections to route while the network still has maximum available bandwidth, and smaller connections can then fit into the remaining bandwidth.

Other Routing Features

- Preferred routes

The preferred route feature allows the user more control over which trunks are used to route a connection. The user may enter the route hop-by-hop, or use a shorthand method to transform a connection's current route into the preferred route.

Once a connection has a configured preferred route, the IGX/BPX will always try to route the connection along the specified path.

If the preferred route is not available (either one of the trunks is failed, or no bandwidth is available), then an alternate route will be used. However, should the preferred route become viable again, the connection will automatically be re-routed back onto its preferred route.

- Directed routes

Once a preferred route is configured, it may also be designated as a directed route. In this case, no alternate route will be used if the preferred route is unavailable. The connection will either route along its preferred route or not at all.

- Route avoid

At the time a connection is added, the user may direct the system to not route the connection on various types of trunks. The user can choose to avoid satellite or terrestrial trunks, and to avoid ZCS encoded trunks. These trunk types are strictly avoided; the connection will never be routed on a restricted trunk type.

- Connection bundling

Once a route has been selected for an individual connection, the routing software attempts to bundle other connections with it in order to save time. As routes are evaluated and a candidate route is eventually selected, the routing software keeps track of how much bandwidth is available along the route. The software then searches for other connections with the same COS and endpoint nodes. As many connections as will fit are "packed" along with the original connection and the route is secured for all of them at the same time.

Routing Software Features

When compared to other systems, the main advantages of Cisco routing software are:

- Speed

In any system, routing consists of two basic steps:

1. Select a route.
2. Allocate resources to secure the route.

In the IGX/BPX, both of these steps can be done rapidly. Route selection is optimized because topology and loading information have been distributed incrementally as the network changed. When faced with a routing decision, each node already has all the information necessary to make an intelligent choice. Allocating resources to secure a route can also be done quickly. Because IGX/BPX networks are composed of a cell-based switching fabric, securing the route is merely a matter of programming the correct cell addresses in each node along the route. This is simpler and faster than architectures that require a more complex allocation of resources at intermediate nodes. Secondly, control information is passed between nodes on high priority, bandwidth-on-demand queues. This affords better performance than architectures that allocate a small, fixed-size communication channel for control traffic.

- Accuracy

Real-time, event-driven database updates allow each node to maintain a detailed and accurate view of the state of the network. This view includes network topology and the amount of bandwidth "booked" on each inter-node trunk. There is no time-consuming partial broadcast or flooding of topology information, hence no measurable convergence time. As a result, routing decisions are based on the true and current state of the network and are therefore guaranteed to yield the "best" route available at that time.

- Distributed intelligence

In an IGX/BPX network, each node is responsible for autonomously routing the connections for which it is responsible. If a trunk failure affects connections at several nodes, each node will re-route its connections in parallel. Depending on the topology of the network, this may yield substantial

improvements in re-route performance.

- Route selection considers congestion in addition to topology

In addition to distributing topology information, each node also distributes information about the amount of "booked" bandwidth on each trunk. Trunks that do not have sufficient bandwidth available to satisfy the guaranteed requirements for a connection (such as CIR) are excluded from consideration when selecting a route. This ensures that a network user receives the level of service (QoS) that was contracted. Less sophisticated algorithms do not consider congestion when making routing decisions and are therefore unable to provide deterministic quality or meet QoS guarantees.

The IGX/BPX routing software does not have these capabilities:

- Cost-based routing

Note: This feature is included in release 9.1.

Currently, you can not configure a cost metric for network trunks (although this feature is supported in release 9.1). Assuming bandwidth is available, the IGX/BPX will always select the route with the least number of hops, which is not necessarily what the user considers to be the "best" route. The preferred route and route avoidance features are often sufficient to remedy these situations, but not always.

- Consideration of propagation delay

Incorporating knowledge of propagation delay into the route selection algorithm is a variation of cost-based routing. In some cases, users may wish to base routing decisions on end-to-end connection delay, either by having the switch regularly measure delay or by configuring a delay estimate for each trunk in the network.

- Network-wide optimization

Since each node routes its connections independently, there is no guarantee that the result will be globally optimized for all connections in the network. A centralized routing intelligence, although typically much slower than a distributed algorithm, has the ability to view the entire network as a whole and direct each slave node in how to route connections.

This has the potential to arrive at a more optimal distribution of connection traffic.

- Load balancing

If multiple routes exist between two nodes, connections will be routed on the shortest available path until all bandwidth on that path is allocated. If multiple paths exist with the same number of hops, a form of load balancing will take place because of the tie-breaker algorithm described earlier. However, there is no software that attempts to distribute the amount of open space on a network-wide basis. This is actually a minor point, since the IGX/BPX route selection algorithm ensures that bandwidth guarantees are always met. However, if traffic could be distributed so that no trunks were fully booked, there is a possibility that connections would be able to burst above their CIR for a longer period of time before congestion takes place, although some connections would be routed through more hops than necessary. This could be a more important feature in switches with unsophisticated routing algorithms because it provides a primitive method of increasing the performance of bursty traffic.

Other Routing Algorithms

This section discusses the advantages and disadvantages of several other routing algorithms used in the industry.

Centralized Routing Intelligence

Some networks provide one central point for routing intelligence. This may be an individual node designated by the user or elected by software to perform this function. Alternatively, routing intelligence may reside in a vendor-specific network management system.

Centralized Routing Intelligence	
Advantage	Disadvantage
The possibility exists to perform some type of network-wide route optimization.	Typically much slower than distributed routing intelligence because all the work is done by one processor. Additional time and overhead are required for communications between master and slaves.
Routing intelligence is concentrated in one entity, so slaves can be "dumb" and have simpler processors, less memory, lower cost, and so on.	A great deal of communication between slaves and master is an
Because processing is "single-threaded" rather than parallel, no mechanism needs to be developed to deal with routing collisions.	inefficient use of network bandwidth. Slower, and potentially less resilient, in the face of failures that cause the master to be separated from one or more slaves. Isolated slaves must either elect a new master or are unable to do any routing. Manual intervention by the user may be necessary.

OSPF

Open Shortest Path First (OSPF) is a routing protocol defined by the IETF in RFC 1583 (previously RFC 1247). It is often considered a standard that allows different routers to interoperate, but the name is something of a misnomer: the algorithm does not guarantee that the shortest path is selected nor that the selected path is "open" (not congested).

OSPF	
Advantage	Disadvantage
Open interface, allows different vendors to interoperate (although there is no guarantee that different vendor implementations support this).	Not connection oriented. All traffic between two nodes always travels on the same route, regardless of congestion or the capacity of individual trunks on that route.
	Slow. Topology and link state messages take a long time to propagate throughout the network (convergence time).

Algorithm considers link cost and selects the least-cost route.	Inefficient. Because OSPF was designed for relatively unintelligent routers, the algorithms for inter-node communications are fairly primitive. Messages are often replicated and received several times by individual nodes.
	Impractical. Experience shows that the combination of slow and inefficient inter-node communications make OSPF impractical for large router networks. As a result, most vendors have implemented proprietary alternatives to OSPF.

Dijkstra's Algorithm

Given a network topology, this algorithm is guaranteed to converge and select the least-cost route. It forms the basis of routing within OSPF. Cisco uses Dijkstra's algorithm in the Network Modeling Tool, but its run-time performance is often unacceptable for a live network.

Dijkstra's Algorithm	
Advantage	Disadvantage
Considers link cost in making routing decisions.	Very slow compared to IGX/BPX route selection.
	Does not consider congestion or network loading when making routing decisions.

Reroute Performance

The reroute performance of an IGX/BPX network varies widely and depends on a number of factors:

- Topology How many routes must be examined before a decision can be made?
- Topology Long routes take more time to secure than short routes.
- Collisions Many simultaneous re-routes can cause nodes to interfere with one another. Depending on the number of consecutive collisions, the collision "back-off" interval can become significant.
- Bundling Routing several connections at once provides some performance enhancement.

Network Tuning

Note: Refer to Optimizing SNA Traffic in a Frame Relay Network for information.

You can take a number of steps to tune your network for optimal re-routing times. This process requires the network designer to make trade-offs based on the network design, topology, trunk quality, frequency of outages, and so on. This section outlines the various trade-offs users can consider in tuning their network.

- Centralized vs. Distributed connection mastership

When a PVC is created, one endpoint is designated as the "master" of the connection and is responsible for re-routing the connection when necessary. If many nodes are re-routing simultaneously and are attempting to secure locks on the same intermediate node, a re-route collision may occur. The more collisions that occur, the longer the re-routing process takes. In many networks, it may make sense to minimize the number of connection masters, thus reducing the probability of collisions. For example, if a network has connections from many branch offices to a few central sites, it makes sense to configure the central sites as master nodes. This also affords the opportunity to create re-route bundles (routing several PVCs at once), which also increases performance. Conversely, if PVC connectivity and trunk topology is mesh-like, it may be advantageous to distribute mastership to allow re-routing to occur in parallel.

- Alarm timing

The idea here is to cause re-routing to occur if the outage is going to be short by not declaring a trunk alarm. In release 8.4, a good guideline for re-routing times in a large network is approximately 1.8 seconds per connection bundle.

For example, following the failure of a fully loaded OC-3 trunk, it can take up to three minutes to re-route all 1771 PVCs. It may be worthwhile to identify trunks in the network that are susceptible to short outages and re-configure their alarm integration times. The default alarm integration time is 3 seconds, but this may be increased to as much as 655 seconds (using the **cnftrkparm** command). The trade-off of increasing this timer is that you delay responding to "real" or longer duration outages while minimizing the effect of short outages.

- Re-route hold-off timer

A configurable timer directs the system to wait "x" seconds following a trunk failure before marking the affected PVCs as "failed" and beginning the re-route process. The concept here is similar to delaying alarm integration, but can be applied on a system-wide basis, as opposed to a per-trunk basis. The default hold-off time is zero seconds, but may be increased to as much as 900 seconds (using the **cnfcmparm** command). Once again, this delays reacting to longer duration outages while minimizing the effect of shorter outages.

- Preferred routes

It is almost always a good idea to make use of preferred routes. Many users allow the system to select a route when a PVC is added. Once they verify that this is a "good" route, a simple command exists to mark the current route as the preferred route. The existence of preferred routes speeds up the routing process and allows the user to optimize resources on a network-wide basis. Large networks with complex topologies tend to become less efficient in their use of trunk resources over time. Unless preferred routes are used, each trunk failure and repair tends to further "scramble" the routes and causes PVCs to end up with sub-optimal routing.

- Comm-break timing

If a node becomes completely unreachable because of a power failure or the simultaneous failure of all its trunks, a comm-break is declared between that node and all other nodes in the network. Once the node re-establishes connectivity, it must clear the comm-break state with each of its peers. This process involves exchanging database updates and is intentionally throttled to avoid overloading the recovered node. The default is 30 seconds per node, but with careful analysis and guidance from Cisco support personnel, this timer may be lowered, thus reducing the impact of a catastrophic node outage.

NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the

most recent conversations available in this technology.

NetPro Discussion Forums – Featured Conversations for WAN Switching

Network Infrastructure: WAN Routing and Switching

Related Information

- [Cisco WAN Switching Solutions – Cisco Documentation](#)
 - [Guide to New Names and Colors for WAN Switching Products](#)
 - [Downloads – WAN Switching Software](#)
 - [Technical Support – Cisco Systems](#)
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Apr 17, 2009

Document ID: 10752
