Routing Overview

Table of Contents

What is routing?

Routing is a way to get one packet from one destination to the next. Routers or software in a computer determines the next network point to which a packet should be forwarded toward its final destination. The router is connected to at least two networks and makes a decision which way to send each data packet based on its current state of the networks it is connected to. A router is located at any point of networks or gateway, including each Internet POP. A router creates or maintains a table of the available routes and their conditions and uses this information along with distance and cost algorithms to determine the best route for a given packet. Typically, a packet may travel through a number of network points with routers before arriving at its destination.

Routing Components

Routing involves two basic activities: determining the optimal routing paths for destination networks and transporting information groups, also known as packets, through an internet work. Within the context of routing, the latter can be referred to as switching.

Path Determination

A metric is a standard of measurement, such as path length, that is used by routing algorithms to determine the optimal path to a destination. To aid in this process of path determination, routing algorithms initialize and maintain routing tables, which contain route information. This information can vary widely depending on which routing algorithm generated the routes.

Routing algorithms fill routing tables with a list of networks and its corresponding "next hop" on the way its destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop.

Routing Table

The following is an example showing a small part of the routing table on stj0.bos.ma.verio.net:

```
stj0>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route
Gateway of last resort is 207.31.207.17 to network 0.0.0.0
B 212.0.212.0/24 [200/24] via 129.250.16.133, 2d08h
B 206.102.168.0/24 [200/32] via 129.250.16.133, 5d07h
B 206.51.253.0/24 [200/25] via 129.250.16.133, 5d05h
B 205.204.1.0/24 [200/35] via 129.250.16.133, 5d07h
B 204.238.34.0/24 [200/24] via 129.250.16.133, 5d00h
B 204.17.221.0/24 [200/35] via 129.250.16.133, 5d07h
B 199.0.199.0/24 [200/32] via 129.250.16.133, 5d07h
B 198.17.215.0/24 [200/37] via 129.250.16.133, 5d07h
B 194.204.14.0/24 [200/42] via 129.250.16.133, 2d22h
B 192.153.89.0/24 [200/24] via 129.250.16.133, 1d08h
B 192.68.132.0/24 [200/35] via 129.250.16.133, 5d07h
B 170.170.0.0/16 [200/25] via 129.250.16.133, 5d05h
B 208.152.73.0/24 [200/25] via 129.250.16.133, 5d05h
B 205.152.84.0/24 [200/25] via 129.250.16.133, 5d05h
```

Each line of this list tells first, where the route was learned from. Some routes will be set by static routes on the router. Other routes will be broadcast using routing protocols such as OSPF and BGP to other routers announcing the availability of a route through that router. The routes that have a B in front are learned by BGP. The second piece of information there is the route that was learned. In the case of the first line, the route for the class C 212.0.212.0/24 was learned through BGP. The next hop for this router to send the packet to is 129.250.16.133, which is most likely an interface on the router itself, which corresponds to a connection to another router in the path to the destination network. Finally, the last line stats how long the route has been up and stable. If a circuit is bouncing, then routes will readjust around the circuit, but the number in the last column will be dramatically smaller.

The current backbone routing table is at about 68,000 routes at this time, which is a route for each separate network on the Internet at large. Some routes could be a whole Class A and others, like the above example, will be a Class C, or anything in between. For more

information on the backbone routing policy, see the section on this topic later in this lesson.

Switching

Switching algorithms are relatively simple and are basically the same for most routing protocols. When a router receives a packet, the packet's destination IP is compared to its routing table to find which next hop the router should forward the packet to. Once the router determines this, the router sends the packet to the MAC address of next hop, exactly like communicating across a Local Area Network.

As it examines the packet's destination protocol address, the router determines that it either knows or does not know how to forward the packet to the next hop. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, it changes the destination physical address to that of the next hop and transmits the packet.

The next hop may, in fact, be the ultimate destination host. If not, the next hop is usually another router, which executes the same switching decision process. As the packet moves through the internetwork, its physical address changes, but its protocol address remains constant.

Algorithm Types

- Static versus dynamic
- Single-path versus multi-path
- Link state versus distance vector

Dynamic vs. Static

Static routing algorithms are hardly algorithms at all, but are table mappings established by the network administrator prior to the beginning of routing. These mappings do not change unless the network administrator alters them. Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

Because static routing systems cannot react to network changes, they generally are considered unsuitable for today's large, changing networks. Most of the dominant routing algorithms in the 1990s are dynamic routing algorithms, which adjust to changing network

circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, stimulating routers to rerun their algorithms and change their routing tables accordingly.

Dynamic routing algorithms can be supplemented with static routes where appropriate. A *router of last resort* (a router to which all unroutable packets are sent), for example, can be designated to act as a repository for all unroutable packets, ensuring that all messages are at least handled in some way.

Single-Path vs. Multipath

Some sophisticated routing protocols support multiple paths to the same destination. Unlike single-path algorithms, these multipath algorithms permit traffic multiplexing over multiple lines. The advantages of multipath algorithms are obvious: They can provide substantially better throughput and reliability.

Link State vs. Distance Vector

Link-state algorithms (also known as *shortest path first* algorithms) flood routing information to all nodes in the internetwork. Each router, however, sends only the portion of the routing table that describes the state of its own links. Distance- vector algorithms (also known as *Bellman-Ford* algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance-vector algorithms send larger updates only to neighboring routers.

Because they converge more quickly, link- state algorithms are somewhat less prone to routing loops than distance- vector algorithms. On the other hand, link- state algorithms require more CPU power and memory than distance- vector algorithms. Link-state algorithms, therefore, can be more expensive to implement and support. Despite their differences, both algorithm types perform well in most circumstances.

Routing Metrics

Routing algorithms have used many different metrics to determine the best route. Sophisticated routing algorithms can base route selection

on multiple metrics, combining them in a single (hybrid) metric. All the following metrics have been used:

- Path Length
- Reliability
- Delay
- Bandwidth
- Load
- Communication Cost

Path length is the most common routing metric. Some routing protocols allow network administrators to assign arbitrary costs to each network link. In this case, path length is the sum of the costs associated with each link traversed. Other routing protocols define *hop count*, a metric that specifies the number of passes through internetworking products, such as routers, that a packet must take en route from a source to a destination.

Reliability, in the context of routing algorithms, refers to the dependability (usually described in terms of the bit-error rate) of each network link. Some network links might go down more often than others. After a network fails, certain network links might be repaired more easily or more quickly than other links. Any reliability factors can be taken into account in the assignment of the reliability ratings, which are arbitrary numeric values usually assigned to network links by network administrators.

Routing delay refers to the length of time required to move a packet from source to destination through the internetwork. Delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, network congestion on all intermediate network links, and the physical distance to be traveled. Because delay is a conglomeration of several important variables, it is a common and useful metric.

Bandwidth refers to the available traffic capacity of a link. All other things being equal, a 10-Mbps Ethernet link would be preferable to a 64-kbps leased line. Although bandwidth is a rating of the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. If, for example, a faster link is busier, the actual time required to send a packet to the destination could be greater.

Load refers to the degree to which a network resource, such as a router, is busy. Load can be calculated in a variety of ways, including CPU utilization and packets processed per second. Monitoring these parameters on a continual basis can be resource-intensive itself.

Communication cost is another important metric, especially because some companies may not care about performance as much as they care about operating expenditures. Even though line delay may be longer, they will send packets over their own lines rather than through the public lines that cost money for usage time.

Dynamic Routing protocols
Routing Information Protocol (RIP)

The Routing Information Protocol(RIP) is a distance-vector protocol that uses hop count as its metric. The original incarnation of RIP was the Xerox protocol, GWINFO. A later version, known as *routed*(pronounced "route dee"), shipped with Berkeley Standard Distribution (BSD) Unix in 1982. RIP itself evolved as an Internet routing protocol, and other protocol suites use modified versions of RIP. The AppleTalk Routing Table Maintenance Protocol (RTMP) and the Banyan VINES Routing Table Protocol (RTP), for example, both are based on the Internet Protocol (IP) version of RIP.

RIP Routing Metric

RIP uses a single routing metric (hop count) to measure the distance between the source and a destination network. Each hop in a path from source to destination is assigned a hop-count value, which is typically 1. When a router receives a routing update that contains a new or changed destination-network entry, the router adds one to the metric value indicated in the update and enters the network in the routing table. The IP address of the sender is used as the next hop.

RIP prevents routing loops from continuing indefinitely by implementing a limit on the number of hops allowed in a path from the source to a destination. The maximum number of hops in a path is 15. If a router receives a routing update that contains a new or changed entry, and if increasing the metric value by one causes the metric to be infinity (that is, 16), the network destination is considered unreachable.

Routing Updates

RIP sends routing-update messages at regular intervals and when the network topology changes, by default, 30 seconds. When a router receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. The metric value for the path is increased by one, and the sender is indicated as the next hop. RIP routers maintain only the best route (the route with the lowest metric value) to a destination. After updating its routing table, the router immediately begins transmitting routing updates to inform other network routers of the change. These updates are sent independently of the regularly scheduled updates that RIP routers send.

Open Shortest Path First (OSPF)

OSPF is a link-state protocol. We could think of a link as being an interface on the router. The state of the link is a description of that interface and of its relationship to its neighboring routers. A description of the interface would include, for example, the IP address of the interface, the mask, the type of network it is connected to, the routers connected to that network and so on. The collection of all these link-states would form a link-state database.

OSPF vs. RIP

The rapid growth and expansion of today's networks has pushed RIP to its limits. RIP has certain limitations that could cause problems in large networks:

- RIP has a limit of 15 hops. A RIP network that spans more than 15 hops (15 routers) is considered unreachable.
- RIP cannot handle Variable Length Subnet Masks (VLSM). Given the shortage of IP addresses and the flexibility VLSM gives in the efficient assignment of IP addresses, this is considered a major flaw.
- Periodic broadcasts of the full routing table will consume a large amount of bandwidth. This is a major problem with large networks especially on slow links and WAN clouds.
- RIP converges slower than OSPF. In large networks convergence gets to be in the order of minutes. RIP routers will go through a period of a hold-down and garbage collection and will slowly time-out information that has not been received recently. This is inappropriate in large environments and could cause routing inconsistencies.
- RIP has no concept of network delays and link costs. Routing decisions are based on hop counts. The path with the lowest hop

count to the destination is always preferred even if the longer path has a better aggregate link bandwidth and slower delays.

- RIP networks are flat networks. There is no concept of areas or boundaries. With the introduction of classless routing and the intelligent use of aggregation and summarization, RIP networks seem to have fallen behind.

Link State Algorithm

OSPF uses a link-state algorithm in order to build and calculate the shortest path to all known destinations. The algorithm by itself is quite complicated. The following is a very high level, simplified way of looking at the various steps of the algorithm:

1. Upon initialization or due to any change in routing information, a router will generate a link-state advertisement. This advertisement will represent the collection of all link-states on that router.
2. All routers will exchange link-states by means of flooding. Each router that receives a link-state update should store a copy in its link-state database and then propagate the update to other routers.
3. After the database of each router is completed, the router will calculate a Shortest Path Tree to all destinations. The router uses the Dijkstra algorithm, or SPF Algorithm, to calculate the shortest path tree. The destinations, the associated cost and the next hop to reach those destinations will form the IP routing table.
4. In case no changes in the OSPF network occur, such as cost of a link or a network being added or deleted, OSPF should be very quiet. Any changes that occur are communicated via link-state packets, and the Dijkstra algorithm is recalculated to find the shortest path.

SPF Algorithm

The OSPF routing algorithm is based on Dijkstra shortest path algorithm. the term "Shortest path" is inaccurate because what we really want to find is the "Optimum path". To find the optimum path in a network means to find a path with a minimal cost, considering factors like time , money and quality of the received data. The route is not chosen only by the cost, because in every network three constraints must be considered :

1. Delay
2. Throughput
3. Connectivity

If delay is excessive or if throughput is too little, the network does not meet the needs of the users. The third constraint is quite obvious: The gateways and networks must be able to reach each other; otherwise, all other least-cost criteria are irrelevant.

After a router is assured that its interfaces are functioning, it uses the OSPF Hello protocol to acquire neighbors, which are routers connected to each other. The router sends hello packets to its neighbors and receives their hello packets. In addition to helping acquire neighbors, hello packets also act as keep-alives to let routers know that other routers are still functional.

Each router periodically sends an LSA to provide information on a router's adjacencies or to inform others when a router's state changes. By comparing established adjacencies to link states, failed routers can be detected quickly and the network's topology altered appropriately. From the topological database generated from LSAs, each router calculates a shortest-path tree, with itself as root. The shortest-path tree, in turn, yields a routing table.

For more information on any terminology discussed here, go to www.mot.com/MIMS/ISG/Products/ons/ospf/faqs/ospfintro.html

Routing Hierarchy

Unlike RIP, OSPF can operate within a hierarchy. The largest entity within the hierarchy is the *autonomous system* (AS), which is a collection of networks under a common administration that share a common routing strategy. OSPF is an intra-AS (interior gateway) routing protocol, although it is capable of receiving routes from and sending routes to other ASs.

An AS can be divided into a number of *areas*, which are groups of contiguous networks and attached hosts. Routers with multiple interfaces can participate in multiple areas. These routers, which are called *area border routers*, maintain separate topological databases for each area.

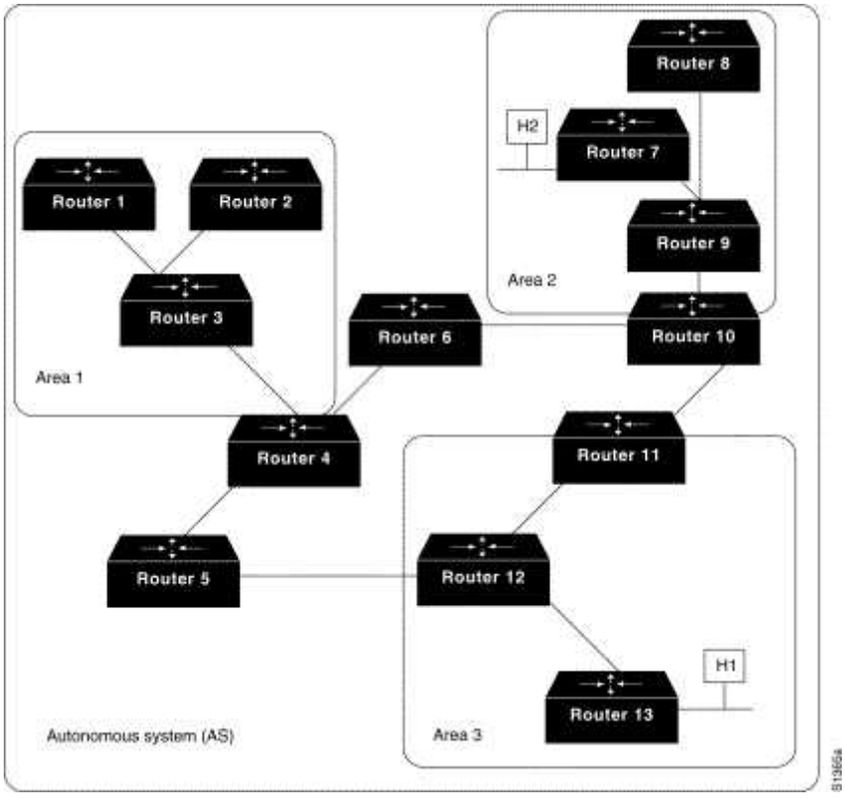
A *topological database* is essentially an overall picture of networks in relationship to routers. The topological database contains the collection of LSAs received from all routers in the same area. Because routers within the same area share the same

information, they have identical topological databases.

The term *domain* sometimes is used to describe a portion of the network in which all routers have identical topological databases. Domain is frequently used interchangeably with AS. An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if the AS were not partitioned.

Area partitioning creates two different types of OSPF routing, depending on whether the source and destination are in the same or different areas. Intra-area routing occurs when the source and destination are in the same area; inter-area routing occurs when they are in different areas.

An OSPF *backbone* is responsible for distributing routing information between areas. It consists of all area border routers, networks not wholly contained in any area, and their attached routers.



In the figure, Routers 4, 5, 6, 10, 11, and 12 make up the backbone. If Host H1 in Area 3 wants to send a packet to Host H2 in area 2, the packet is sent to Router 13, which forwards the packet to Router 12, which sends the packet to Router 11. Router 11 then forwards the packet along the backbone to area border Router 10, which sends the

packet through two intra-area routers (Router 9 and Router 7) to be forwarded to Host H2.

The backbone itself is an OSPF area, so all backbone routers use the same procedures and algorithms to maintain routing information within the backbone that any area router would. The backbone topology is invisible to all intra-area routers, as are individual area topologies to the backbone.

Areas can be defined in such a way that the backbone is not contiguous. In this case, backbone connectivity must be restored through *virtual links*. Virtual links are configured between any backbone routers that share a link to a non-backbone area and function as if they were direct links.

Border Gateway Protocol (BGP)

See RFC 1771 fore more information:

http://www.freesoft.org/CIE/RFC/1771/index.htm

BGP Routing

As with any routing protocol, BGP maintains routing tables, transmits routing updates, and bases routing decisions on routing metrics. The primary function of a BGP system is to exchange network-reachability information, including information about the list of autonomous system paths, with other BGP systems. This information can be used to construct a graph of autonomous system connectivity from which routing loops can be pruned and with which autonomous system-level policy decisions can be enforced.

BGP devices exchange routing information upon initial data exchange and after incremental updates. When a router first connects to the network, BGP routers exchange their entire BGP routing tables. Similarly, when the routing table changes, routers send the portion of their routing table that has changed. BGP routers do not send regularly scheduled routing updates, and BGP routing updates advertise only the optimal path to a network.

Each BGP router maintains a routing table that lists all feasible paths to a particular network. The router does not refresh the routing table, however. Instead, routing information received from peer routers is retained until an incremental update is received.

BGP uses a single routing metric to determine the best path to a given network. This metric consists of an arbitrary unit number that specifies the degree of preference of a particular link. The BGP metric is typically assigned to each link by the network administrator. The value assigned to a link can be based on any number of criteria, including the number of autonomous systems through which the path passes, stability, speed, delay, or cost.

BGP Message Types

Four BGP message types are specified in RFC 1771, *A Border Gateway Protocol 4 (BGP-4)*: open message, update message, notification message, and keep-alive message.

The *open message* opens a BGP communications session between peers and is the first message sent by each side after a transport-protocol connection is established. Open messages are confirmed using a keep-alive message sent by the peer device and must be confirmed before updates, notifications, and keep-alives can be exchanged.

An *update message* is used to provide routing updates to other BGP systems, allowing routers to construct a consistent view of the network topology. Updates are sent using the Transmission-Control Protocol (TCP) to ensure reliable delivery. Update messages can withdraw one or more unfeasible routes from the routing table and simultaneously can advertise a route while withdrawing others.

The *notification message* is sent when an error condition is detected. Notifications are used to close an active session and to inform any connected routers of why the session is being closed.

The keep-alive message notifies BGP peers that a device is active. Keep-alives are sent often enough to keep the sessions from expiring.

Finite State Engine

In general, a state machine is any device that stores the status of something at a given time and can operate on input to change the status and/or cause an action or output to take place for any given change. With BGP-4, the FSM maintains a listing of the state of each BGP peering session. There are 6 possible states that a session can be.

1. Idle – BGP is refusing all connections and no resources are allocated to this. In response to an initialization request, initiated by either the system or an operator, the local system initializes all BGP resources, starts the ConnectRetry timer, initiates a transport connection to other BGP peer, and changes its state to Connect.
2. Connect – BGP is now waiting for the transport protocol connection to be completed.
3. Active – BGP is trying to acquire a peer by initiating a transport protocol connection.
4. OpenSent – BGP has sent a request to its peer to request that the peering session be brought up. It is now waiting for an Open Message back from its peer to acknowledge that the remote peer is ready.
5. OpenConfirm – BGP has received the OpenMessage and is waiting for a keepalive or a notification message. If BGP receives a Keepalive, then the connection will be established. All other messages or lack of messages will result in the session being brought back to an Idle state.
6. Established – BGP can now exchange Update, Notification, and Keepalive messages with its peer. This does not mean that any routing updates have been exchanged, only that the BGP connection is capable of doing so.

Confederations

A router can only be within one BGP Autonomous System. That means that all Verio routers are only within the 2914 AS. However, because each BGP router must be a peer with all of its neighbors and ideally be setup in a full-mesh topology, we only run 2914 in a full-mesh configuration on our backbone. Each of our legacy ISP's uses a 65000 number Autonomous System number to for the regional BGP as part of a confederation of the main 2914 AS. Any AS above 65000 has been delegate by ARIN to be private AS numbers for use such as this, similar to the 192.168 network within IP space, for example. This allows regions to run BGP internally, referred to as IBGP, and then at their regional border router encapsulate all the routes in the 2914 AS. Where the regional router connects to the backbone, External BGP, or EBGP, is used to broadcast routes throughout the backbone and the Internet at large.

Administrative Distance

Routers must have a way of assembling the various routing updates that they receive from all of the routing protocols, static routes, and connected interfaces that it is receiving. Cisco and other vendors came up with similar plans to attack this issue. The solution is called administrative distance. Administrative distance as it applies to assembling a routing table simply gives more weight to different types of ways that a router learns a route. For example, a static route will get more weight than a route learned through OSPF. The chart below lists the default administrative distances that Cisco routers use to put the various routing protocols into a hierarchy. Other vendors use a very similar system.

Route Source	Default Distance
Connected interface	0
Static route	1
Enhanced IGRP summary route	5
External BGP	20
Internal Enhanced IGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
External Enhanced IGRP	170
Internal BGP	200
Unknown	255

These default values can be manipulated to different values than the default value. A common implementation of this is using a route to Null0 with an administrative distance of 250 to simply hold a route until another routing protocol with a higher administrative distance overrides the route. See the *Place Holder Routes* for more information.

Convergence

Convergence is the process of agreement, by all routers, on optimal routes. When a network event causes routes to either go down or become available, routers distribute routing update messages. Routing update messages permeate networks, stimulating recalculation of

optimal routes and eventually causing all routers to agree on these routes. High levels of network instability can lead to packet loss, increased network latency and time to convergence. At the extreme, high levels of routing instability have lead to the loss of internal connectivity in wide-area, national networks. Routing algorithms that converge slowly can cause routing loops or network outages.

Place Holder Routes

Why does my traceroute not go to the router the customer is connected to when they are down? At the main regional border router for each of our affiliates, place holder routes are put into place to hold the routes for all of the IP's for that network. Place holder routes are always routed to Null0 with a very low administrative distance, usually 250.

These routes serve two purposes. One, if a customer went down and there were no place holder route, the traceroute would simply timeout and not go anywhere. That is, these Null0 routes allow your traceroute to at least go to the region that the customer is connected to. Second, part of BGP-4 is a system to ignore routing flaps. Routing flaps are routes that change from active to inactive very quickly, usually because of a bouncing circuit. Any time a circuit goes up or down within a system running a link-state routing protocol, an announcement is made concerning the change of routes due to that circuit. These routing changes can cause a lot of havoc on networks if they are not controlled properly. Thus, these place holder routes essentially dampen any route flaps so that they only affect the regional network and are not propagated to our backbone and to the Internet at large. If these were routing announcements were propagated, some routes to Verio would be completely ignored even after the circuit was repaired.

Routing Arbiter Database

The Routing Arbiter Database, also known as the RADB, contains a list of registered routes for ISP's across the network. The Routing Arbiter Database is hosted at www.radb.net and is a component of the distributed Internet Routing Registry. While ARIN delegates AS Numbers and IP addresses, it is actually the RADB that registers the routes for all Internet IP's. Verio, as well as almost all ISP's, filter routes based upon routes that are registered with the RADB. Route announcements that do not coincide with a registered route or are not registered with the RADB will not be visible to most of the Internet.

Below is an example of how to query the RADB for routing database information:

```
[tethys]:[9:03am]:[/home/rnejdl] > whois -h whois.ra.net
206.50.17.51
route:          206.50.0.0/16
descr:          Verio TX (65016)
origin:         AS2914
member-of:      RS-COMM_GFABG
remarks:        This object is automatically converted from
the RIPE181 registry
notify:         radb-ntfy@verio.net
notify:         ne@tx.verio.net
mnt-by:         MAINT-VC-TX
changed:        db-admin@rr.verio.net 19990811
changed:        auto-dbm@ISI.EDU 19991101
source:         VERIO
[tethys]:[9:03am]:[/home/rnejdl] >
```

For further reading on configuring and troubleshooting these routing protocols and others, go to http://www.cisco.com/univercd/cc/td/doc/product/software/ios11/cbook/ciproute.htm#xtocid24843106

Troubleshooting

Ping, Traceroute, and MTR

Basic PING

```
[tethys]:[11:55pm]:[/home/rnejdl] > ping -s 199.1.11.2
PING 199.1.11.2 (199.1.11.2): 56 data bytes
64 bytes from 199.1.11.2: icmp_seq=0 ttl=251 time=1061.085
ms
64 bytes from 199.1.11.2: icmp_seq=1 ttl=251 time=1207.764
ms
64 bytes from 199.1.11.2: icmp_seq=2 ttl=251 time=971.188
ms
64 bytes from 199.1.11.2: icmp_seq=3 ttl=251 time=1117.044
ms
64 bytes from 199.1.11.2: icmp_seq=4 ttl=251 time=1260.583
ms
^C
--- 199.1.11.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev =
971.188/1123.533/1260.583/102.984 ms
[tethys]:[11:55pm]:[/home/rnejdl] >
```

Basic Traceroute

```
[tethys]:[9:36pm]:[/home/rnejdl] > traceroute ns0.verio.net
traceroute to ns0.verio.net (129.250.15.61), 30 hops max,
40 byte packets
1 hyperion (206.50.17.49) 2.719 ms 2.646 ms 2.609 ms
2 d1xx-l0.tx.verio.net (199.1.11.47) 29.211 ms 29.136 ms
29.045 ms
3 border5-fa1-0-0.dlls.tx.verio.net (199.1.129.65) 36.103
ms 35.040 ms 35.2 75 ms
4 core1-g1-0-0.dlls.tx.verio.net (199.1.141.11) 35.256 ms
35.336 ms 35.388 ms
5 g6-0.dfw2.verio.net (129.250.31.49) 35.738 ms 35.507 ms
35.386 ms
6 p1-2-3.r05.plalca01.us.bb.verio.net (129.250.2.162)
79.915 ms 79.448 ms 7 9.045 ms
7 pao5.pao6.verio.net (129.250.2.130) 79.454 ms 79.413 ms
79.909 ms
8 pao6.sea3.verio.net (129.250.3.90) 102.264 ms 97.870 ms
98.008 ms
9 sea3.sea2.verio.net (129.250.2.229) 97.514 ms 97.664 ms
105.955 ms
10 p1-1-1-0.r01.ptldor01.us.bb.verio.net (129.250.2.33)
102.466 ms 102.228 ms 102.386 ms
11 ns0.verio.net (129.250.15.61) 103.924 ms * 104.237 ms
[tethys]:[10:34pm]:[/home/rnejdl] >
```

Tracerouting through another router

```
[yoda]:[6:00am]:[/export/home/rnejdl] > traceroute -g
stj0.bos.ma.verio.net 199.1.11.2
traceroute: Warning: ckecksums disabled
traceroute to 19 (199.1.11.2), 30 hops max, 48 byte packets
1 hsrp0.noc.verio.net (129.250.32.253) 0.663 ms 0.550 ms
2.266 ms
2 g6-0.dfw2.verio.net (129.250.31.49) 0.706 ms 0.687 ms
2.187 ms
3 dfw2.iad3.verio.net (129.250.2.210) 46.265 ms 46.037 ms
44.957 ms
4 iad3.phl00.verio.net (129.250.3.106) 53.623 ms 49.632 ms
49.872 ms
5 phl00.phl02.verio.net (129.250.3.154) 48.876 ms 50.246 ms
48.677 ms
6 phl02.nyc1.verio.net (129.250.3.126) 52.410 ms 52.688 ms
52.325 ms
7 d3-1-0-1.a00.hrfrct01.us.ra.verio.net (129.250.16.134)
54.680 ms 55.256 ms 55.045 ms
8 h3-0.ovl0.bos.ma.verio.net (207.31.208.138) 58.252 ms
58.104 ms 59.180 ms
9 h1-1-0.stj0.bos.ma.verio.net (207.31.207.18) 58.915 ms
```

```
58.939 ms 59.777 m s
10 h5-0.ovl0.bos.ma.verio.net (207.31.207.17) 59.794 ms
58.018 ms 58.142 ms
11 h1-0-0.htf0.hfd.ct.verio.net (207.31.208.137) 58.858 ms
60.455 ms 59.170 ms
12 vne.nyc1.verio.net (129.250.16.133) 68.516 ms 68.332 ms
68.000 ms
13 nyc1.phl02.verio.net (129.250.3.125) 60.451 ms 60.006 ms
61.994 ms
14 phl02.phl00.verio.net (129.250.3.153) 70.043 ms 70.599
ms 69.056 ms
15 phl00.iad3.verio.net (129.250.3.105) 74.051 ms 73.546 ms
73.575 ms
16 iad3.dfw2.verio.net (129.250.2.209) 59.147 ms 59.636 ms
59.715 ms
17 vtxs2.dfw.verio.net (129.250.31.58) 59.120 ms 59.503 ms
59.509 ms
18 border2-f5-0-0.dlls.tx.verio.net (199.1.141.1) 60.805 ms
61.148 ms 59.648 ms
19 ns.onramp.net (199.1.11.2) 60.330 ms 64.219 ms 60.651 ms
[yoda]:[6:01am]:[/export/home/rnejdl] >
```

Matt's Traceroute

```
[yoda]:[5:59am]:[/export/home/rnejdl] > mtr 199.1.11.2
Matt's traceroute  [v0.37]
yoda                                                    Wed
Jan 12 05:59:51 2000

Keys:  D - Display mode    R - Restart s    Packets
Pings
Hostname                                  %Loss  Rcv  Snt
Last Best  Avg  Worst
 1. hsrp0.noc.verio.net                      0%   15   15
0    0    6     69
 2. g6-0.dfw2.verio.net                      0%   14   14
1    0    1      3
 3. vtxs2.dfw.verio.net                      0%   14   14
0    0    0      2
 4. border2-f5-0-0.dlls.tx.verio.net         0%   14   14
1    1    2      4
 5. ns.onramp.net                            0%   14   14
1    1    1      3
```

Show Commands

Show IP Route

```
stj0>show ip route 199.103.247.129
Routing entry for 199.103.247.128/25
Known via "static", distance 1, metric 0
Redistributing via ospf 97
```

```
Advertised by ospf 97 subnets
Routing Descriptor Blocks:
* 204.57.34.2
Route metric is 0, traffic share count is 1

stj0>show ip route 204.57.34.2
Routing entry for 204.57.34.0/30
Known via "connected", distance 0, metric 0 (connected, via
interface)
Redistributing via ospf 97
Advertised by ospf 97 subnets
Routing Descriptor Blocks:
* directly connected, via Serial1/0/0/1:0
Route metric is 0, traffic share count is 1
stj0>
```

What if you want to know what IP's are routed through an interface? Then you can use the following two commands:

```
stj0>show ip route | inc Serial1/0/0/2:0
C 204.57.34.4 is directly connected, Serial1/0/0/2:0
stj0>
stj0>show ip route | inc 204.57.34.2
O E2 204.57.34.244 [110/20] via 204.96.36.10, 12:38:32,
FastEthernet0/0/0
O E2 204.57.34.240 [110/20] via 207.31.207.17, 03:59:59,
Serial1/1/1
O E2 204.57.34.248 [110/20] via 207.31.207.17, 12:38:32,
Serial1/1/1
S 199.103.247.128 [1/0] via 204.57.34.2
stj0>
```

Show IP BGP

Whenever a customer with a BGP sessions calls up, you want to check three things when troubleshooting their connection. You want to check to make sure their BGP peering sessions is configured correctly and up. You want to ensure that the customer is sending their routes to us correctly. Finally, you want to verify that the routes that the customer is sending to us are registered in a routing database. The following three commands will show you how to check this:

```
stj0>show ip bgp summary
BGP router identifier 207.31.207.29, local AS number 65018
BGP table version is 13230196, main routing table version
13230196
66888 network entries and 66942 paths using 8630961 bytes
```

```
of memory
9965 BGP path attribute entries using 815080 bytes of
memory
BGP activity 818809/751916 prefixes, 2670070/2603128 paths
237 prefixes revised.

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down
State/PfxRcd
130.94.57.201 4 65018 78191 78419 13230196 0 0 7w5d 13
130.94.60.201 4 65018 9148706 78419 13230196 0 0 7w5d 66873
206.166.191.200 4 65018 78194 78419 13230196 0 0 7w5d 0
207.31.207.25 4 65018 42086 42263 13230196 0 0 4w1d 0
207.31.207.26 4 65018 78181 78427 13230196 0 0 7w3d 0
207.31.207.28 4 65018 1582948 78419 13230196 0 0 7w5d 40
207.31.207.62 4 11304 926592 3590765 13230194 0 0 00:05:43
1
207.31.207.106 4 6530 77596 3222194 13230194 0 0 3d06h 9
207.31.207.142 4 4133 78306 4051941 13230194 0 0 1w0d 4
207.31.207.150 4 11298 78182 2924324 13230191 0 0 2w6d 2
207.31.208.6 4 10271 0 0 0 0 0 never Active
stj0>

stj0>show ip bgp neighbors 207.31.207.150 received-routes
BGP table version is 5908807, local router ID is
207.31.207.29
Status codes: s suppressed, d damped, h history, * valid, >
best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

*> 208.192.4.0/23 207.31.207.150 281601 0 11298 i
*> 208.215.194.0/23 207.31.207.150 281601 0 11298 i

Total number of prefixes 2
stj0>

[yoda]:[5:36am]:[/export/home/rnejdl] > whois -h
whois.ra.net 208.192.4.0
route: 208.192.4.0/23
descr: GreenNet -- UUNet block 0
origin: AS11298
notify: routing@pn.com
notify: radb-ntfy@verio.net
mnt-by: MAINT-AS3727
changed: routing@pn.com 19980402
```

```
source: RADB
[yoda]:[5:37am]:[/export/home/rnejdl] >
```

Types of Routing Problems

- No/Incorrect Static Route - This is usually found when you are able to traceroute to the customer's serial IP, but not to their ethernet IP address. Logging into the router the customer is connected to and doing a show IP route for the ethernet IP's ether gives a network not in table, routes to the router's default gateway, or routes to the the wrong interface. These are easy to solve, but contact your teamlead if you find this to be the problem as you will not have the enable password to correct this configuration problem.
- Routing loops to customer's serial interface and Verio's router - If you can traceroute to the customer's serial IP, then they are obviously connected. However, if you traceroute to the IP's that the customer should have for their network and it goes to their serial interface and then back to Verio's router and back to the customer's router and so on, then the customer's router is misconfigured and has dropped the IP address for the ethernet side of the interface. Have the customer log into their router and get into enable mode. Then have the customer type the following commands:

```
router> Conf t
router(config)> Int e0
router(config-if) ip address ETHERNETIP SUBNETMASK
router(config-if) ctrl+z
router>
```

Once these commands are entered, the customer should be able to route. If the problem is more complex this or you need further instructions, contact your teamlead.

- Routing loops between two of Verio's routers - Most often, Verio has done some work on the network and changed where the customer connects to, but this is not always the case. The cause, however, is almost always a static route in another router for the customer's Ethernet IP's that should not be there.
- Route goes to incorrect router - This is commonly caused by a customer's IP's being given to 2 customers at the same time. Contact your teamlead or escalate to the region for reprovisioning.

- BGP - Use the tools in the previous section to troubleshoot the customer's BGP sessions.

Terms and Definitions

Routing Table – A listing of networks and their corresponding next hop address. Only one routing table exists per router.

Autonomous System – A group of routers and networks acting under a single routing policy that is controlled by a common network administrator. An autonomous system is also sometimes referred to as a routing domain. An autonomous system is assigned a globally unique number called an Autonomous System Number (ASN).

LSA – Link State Announcement. Used by OSPF, an LSA is used to announce changes in network topology to adjacent routers.

RIP – Routing Information Protocol. An older routing protocol still in use today, RIP is a distance vector routing protocol that uses hop count as its single metric.

BGP – Border Gateway Protocol. Used mostly to exchange routing information between ISP's, BGP calculates its routes based on the number of autonomous systems that a route must go through to reach its destination.

OSPF – Open Shortest Path First. A more advanced routing protocol than RIP, OSPF takes into account more than simply hop count, such as bandwidth, when calculating a shortest path from one point to another. Furthermore, OSPF is not constrained to classful routing like RIP.

Dijkstra Algorithm – Named after the mathematician Dijkstra, the Dijkstra algorithm is used to calculate the shortest path from one vertices to another within any size 2 dimensional graph. For more information on this concept in action, see:

http://magma.mines.edu/Academic/courses/math_cs/macs563/users/centihar/applet/

Finite State Engine - A finite state machine is one that has a limited or finite number of possible states. In the context of BGP, a finite state engine applies to the concept that BGP only has 6 states that a peer

can be in and triggers must occur to move the finite state engine between states.

Metric – A metric is a measure used in calculating the next host to route a packet to. The six main metrics used in routing are path length, reliability, delay, bandwidth, load, and communications cost.

Administrative Distance – An arbitrary system used to combine multiple routing announcements into one routing table.

Link State Routing Protocol – A routing protocol that only updates its routing tables when a link, or line, changes state within the network. These routing protocols are known for their quick convergence time.

Distance Vector Routing Protocol – A routing protocol that uses distance, or hop count, as its means for determining best path. Distance Vector routing protocols sometimes use link state changes to trigger a routing announcement.

Convergence – Convergence describes the concept of all the routers within a network coming into agreement on one overall network topology.

Route Flap – A route flap occurs whenever a route changes state from *up* to *down* to *up* or vice versa. Routing flaps can cause a lot of unnecessary traffic to flow across a network as well as cause significant routing problems.

Peer – A router that is adjacent to another router and exchanges routing information. A peer can also be a point of connectivity between two networks.