OSPF Features

The list on the opposite page summarizes features of the OSPF protocol.

OSPF has the following features:

- Created specifically for use in large Internet Protocol (IP) internetworks. OSPF is one of a number of link state protocols. NetWare Link Services Protocol (NLSP) and Intermediate System to Intermediate System (IS-IS) are examples of other link state protocols.
- Based on cost. Unlike Routing Information Protocol (RIP), there is no unreachable metric.
- Uses different size subnet masks in the same network, giving more efficient use of available address space.
- Supports numbered and unnumbered point-to-point networks.
- Enables type of service (TOS) routing and equal cost multipath. Most vendors, including Bay Networks, do not support TOS. With release 11.02, equal cost multipath is supported.
- Converges more quickly than RIP. In an OSPF environment, link state advertisements (LSAs), not networks, are exchanged. These advertisements reflect actual network topology information, not "word of mouth" information. Because LSAs are triggered by any change in the network and flooded to all routers, there are no count to infinity issues.
- Can be CPU-intensive, particularly when OSPF is recalculating new routes. Link state database and SPF tree consume additional memory.
- Requires more planning and careful use of network address assignments to use its best features. RIP is plug-and-play.
- Uses Dijkstra's SPF algorithm.
- Request for Comment (RFC) 1583/2178 (OSPF version 2) represents the present standard.

OSPF Features

- Created to use in large IP networks
- Link state protocol based on cost.
- Supports numbered and unnumbered links.
- Enables use of different subnet masks in same network.
- Supports TOS routing and equal cost multipath.
- Converges quickly.
- CPU- and memory-intensive
- Requires careful planning.
- Defined in RFC 1583/2178

Basic Components of OSPF

The figure on the opposite page illustrates the basic components of OSPF.

Routers using link state protocols, such as OSPF, do not exchange routing information. They exchange link state information, which is maintained by each router in a database describing the domain's topology. This database is called the link state database (LSDB).

The database is often displayed in technical literature as a diagram with a graph composed of nodes and edges.

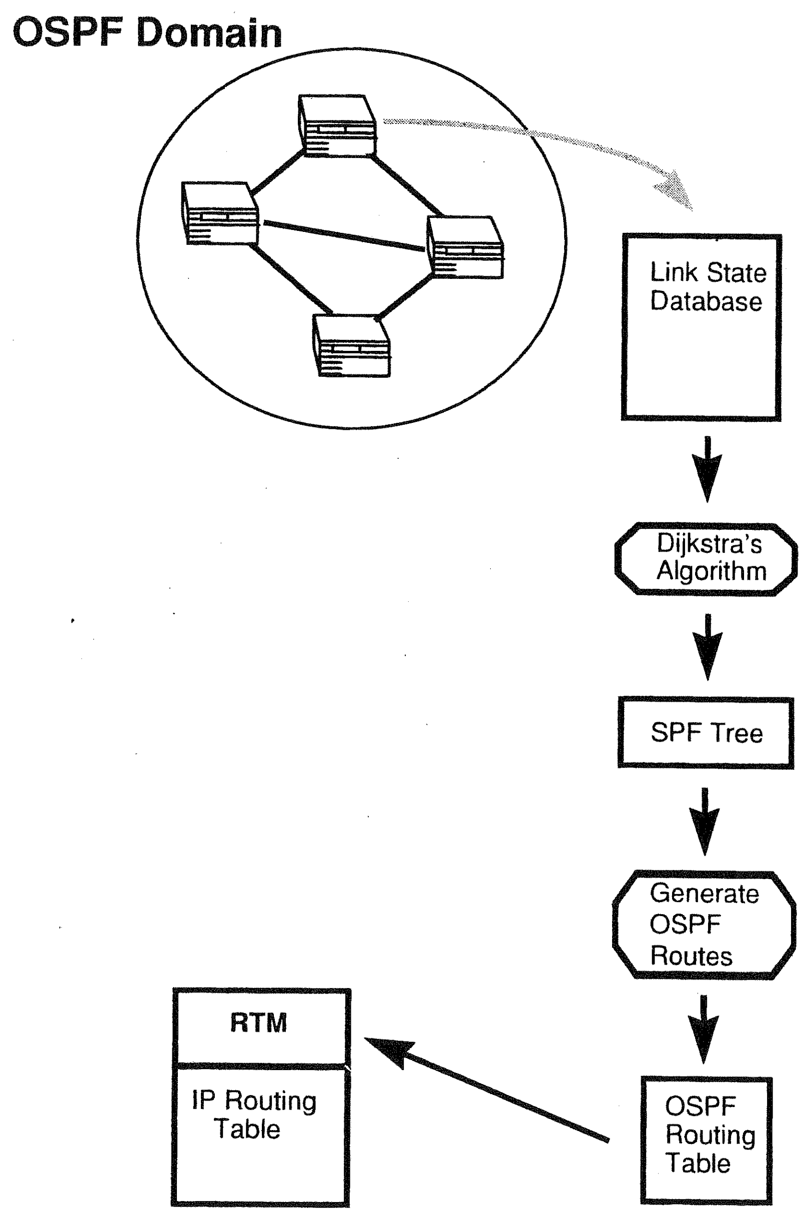
The LSDB is a data structure containing LSAs. Each participating router has an identical database. Each advertisement in the LSDB was built by one of the routers in the OSPF domain and sent to every other OSPF (flooding) router.

Using Dijkstra's shortest path first (SPF) algorithm, and working from the LSDB, each router constructs a tree of shortest paths with itself as the root (called the SPF tree). All routers run this algorithm in parallel. The SPF tree gives the route to each destination in the autonomous system. A routing table can be derived from the SPF tree.

The LSDB does not contain a best route. The SPF tree derived from this database contains the best route, and the route is put in the routing table.

The router forwards a datagram to the next hop router based on the routing table.

Basic Components of OSPF



The Network as a Graph

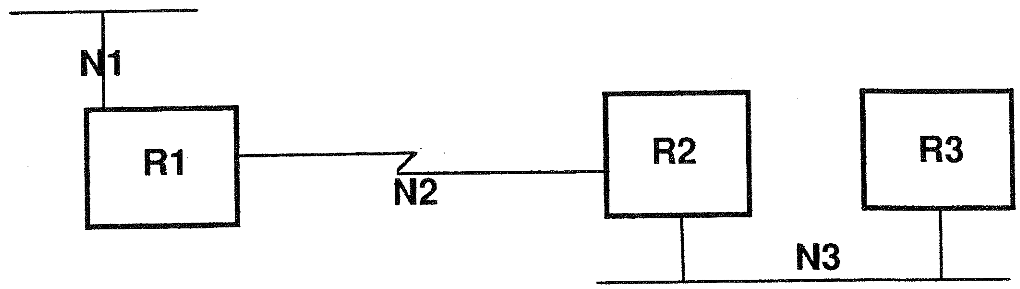
The figure on the opposite page illustrates some examples of how physical networks are represented as a graph by OSPF.

The LSDB is often called the topological database. It is depicted as a directed graph. Many terms used in the display of router information reflect this graphic origin. To understand the displays, it is useful to know some basic terminology:

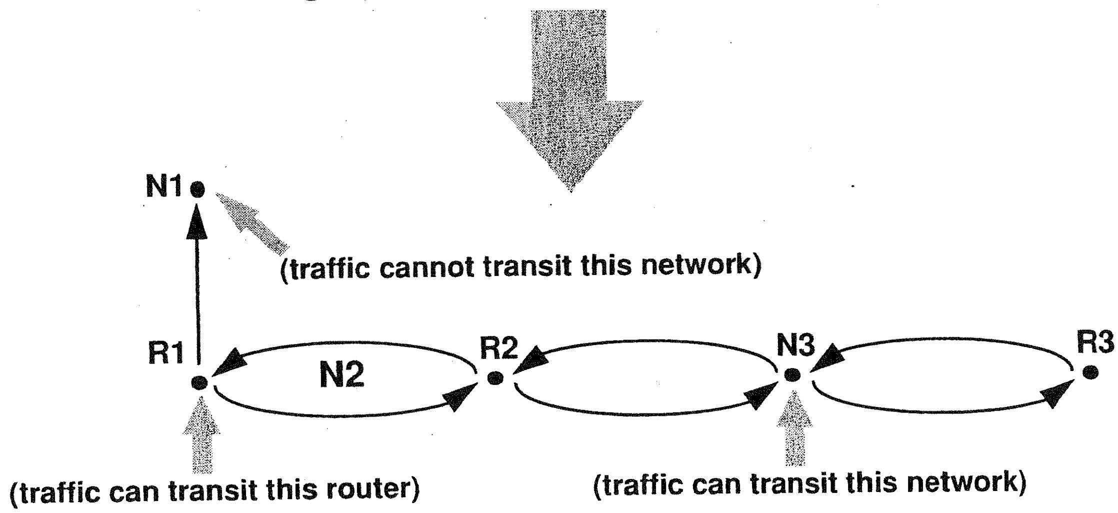
- Vertices (nodes) of the graph are either networks or routers.
- Vertices (networks or routers) are connected by a graph edge.
- Vertices and edges can depict how a line carries traffic. For example, a vertice (network or router) carrying transit traffic is graphed by having both incoming and outgoing edges.

  A router with an interface on a broadcast network with no other routers is represented as two vertices (one representing the router and one representing the network) connected by a unidirectional edge. The unidirectional edge indicates that this network does not carry transit traffic (a stub network).

The Network as a Graph



As a graph, this network looks like this



Link State Operation

Each router builds an advertisement describing its immediate surroundings, for example, the state of its links.

In the advertisement, only directly connected networks are included.

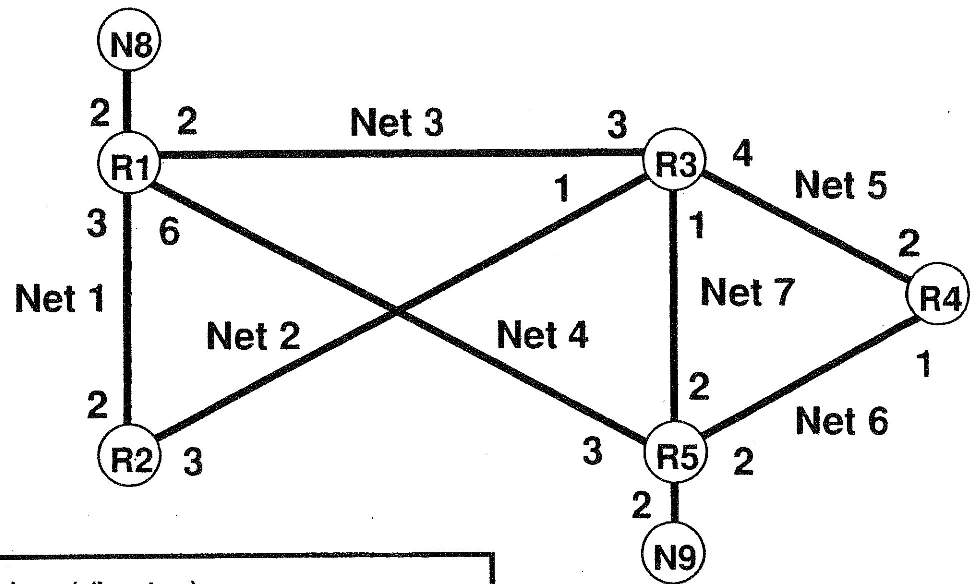
LSAs can propagate throughout the OSPF domain.

In the figure on the opposite page, the LSDB of all routers contains information about the complete network. This information is pieced together from advertisements received from each of the other participating routers within the OSPF domain.

Note that two of the vertices (N8 and N9) are networks representing a broadcast network with one router.

All other vertices represent routers connected by point-to-point links.

Link State Operation



Link State Database (all routers)
R2 Net 1/2, Net 2/3, R1/2, R3/3
R1 Net 1/3, Net 3/2, Net 4/6, Net 8/2, R2/3, R3/2, R5/6
R3 Net 3/3, Net 2/1, Net 7/1, Net 5/4, R1/3, R5/1, R2/1, R4/4
R4 Net 5/2, Net 6/1, R3/2, R5/1
R5 Net 4/3, Net 7/2, Net 6/2, Net 9/2, R1/3, R3/2, R4/2

Dijkstra's Algorithm

The figure on the opposite page illustrates Dijkstra's algorithm.

The Dijkstra algorithm is graph theory applied to routing. It is used in various link state routing protocols such as OSPF, Intermediate System to Intermediate System (IS-IS), and NetWare Link Services Protocol (NLSP).

With this algorithm, the routers can determine the best path to each destination in the internetwork, where:

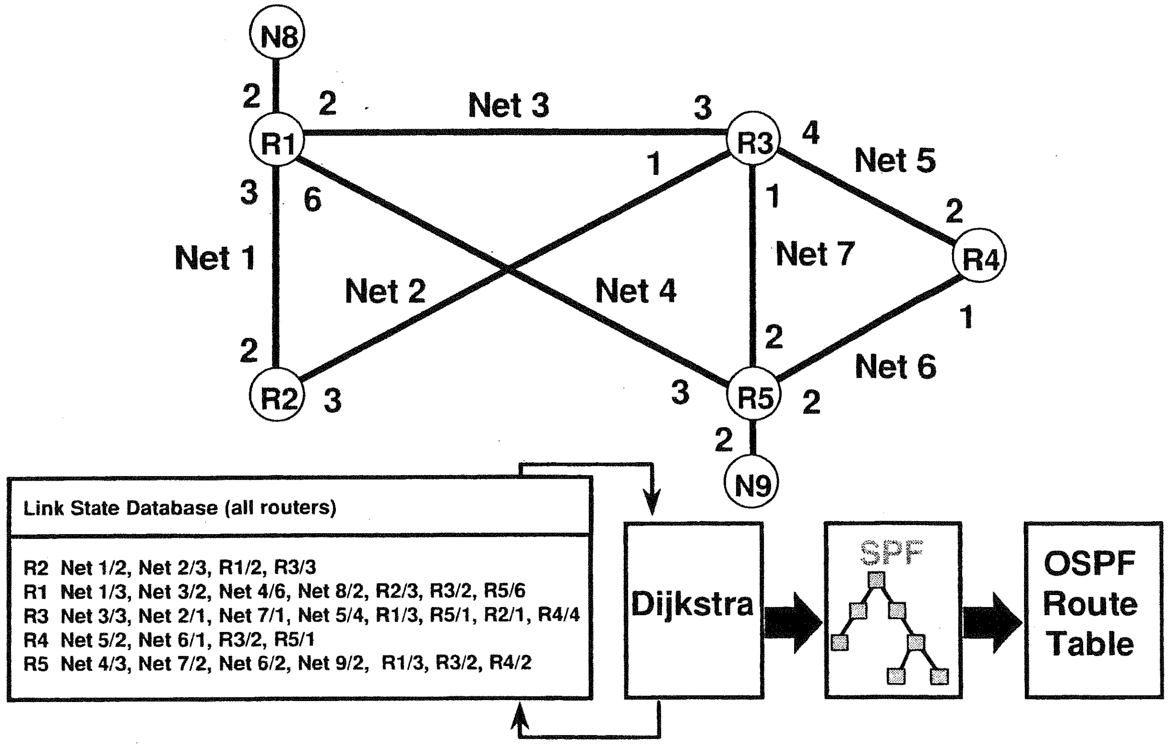
- The best path is the path with the overall lowest cost.
- The destination is either a network or a router.

Routers can also step through the LSDB and build an SPF tree based on the best path with this algorithm. In each iteration of the algorithm, all known paths to a destination network or router are mapped, and the lowest-cost path chosen as the new branch to the tree.

This process is repeated until the best path to each destination is discovered. The result is a logical tree defining the best path to all points in the network.

Each router in the OSPF domain runs this algorithm in parallel.

Dijkstra's Algorithm



Dijkstra's Algorithm—Step 1

This example demonstrates how R2 discovers the best path to R1, R3, R4, R5, N8, and N9.

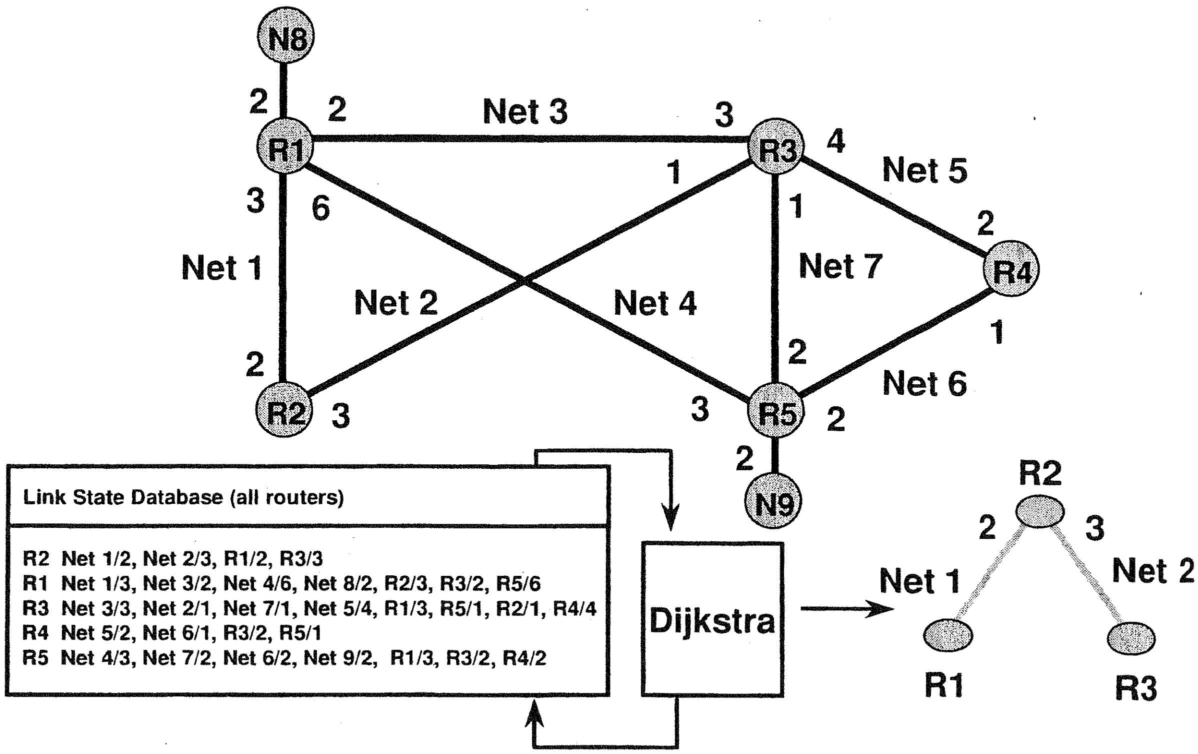
Cost metrics are added to the graph, and the cost is calculated on the outgoing interface only.

Step 1

The figure on the opposite page illustrates step 1.

R2 starts by looking at its directly connected links to R1 and R3, and adds them as potential branches to the OPSF tree.

Dijkstra's Algorithm—Step 1



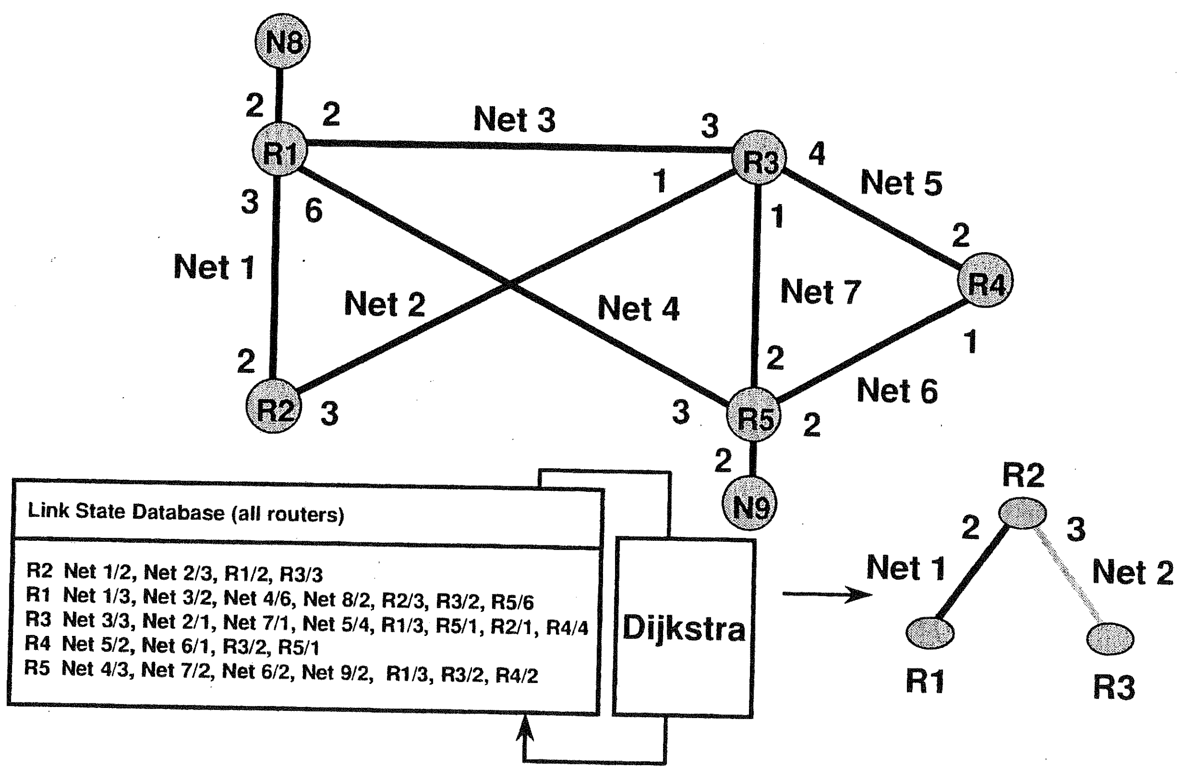
Dijkstra's Algorithm—Step 2

Step 2

The figure on the opposite page illustrates step 2.

The link to R1 is added to the SPF tree because its cost of 2 is lower than the cost of the link to R3 (3).

Dijkstra's Algorithm—Step 2



Dijkstra's Algorithm—Step 3

Step 3

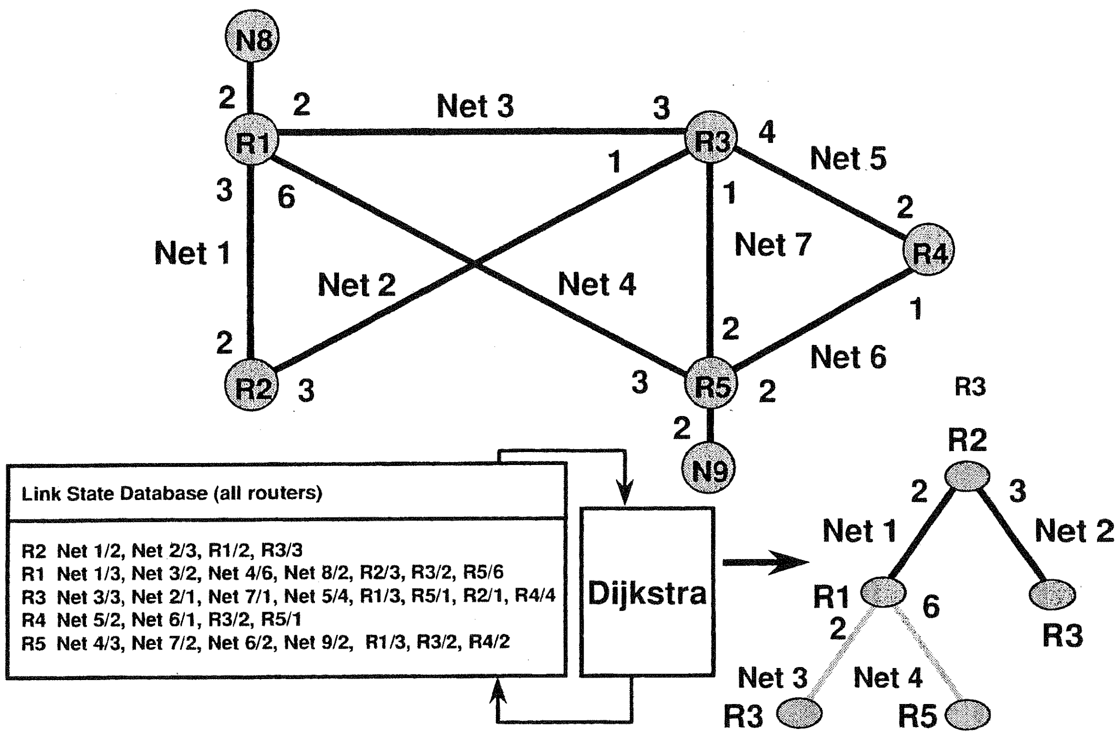
The figure on the opposite page illustrates step 3.

In the second iteration, the links to R1 (now part of the tree) are considered.

R2 discovers that from R1 there are paths to R3, R5, and the broadcast network N8. The network N8 is ignored for now because it represents a stub network (no other routers are on it).

Using the SPF tree as a reference, R2 has two possible paths to R3 (Net 3 and Net 2). R2 chooses the path through Net 2 because it is the least cost path to R3. This path becomes a new branch of the tree.

Dijkstra's Algorithm—Step 3



Dijkstra's Algorithm—Step 4

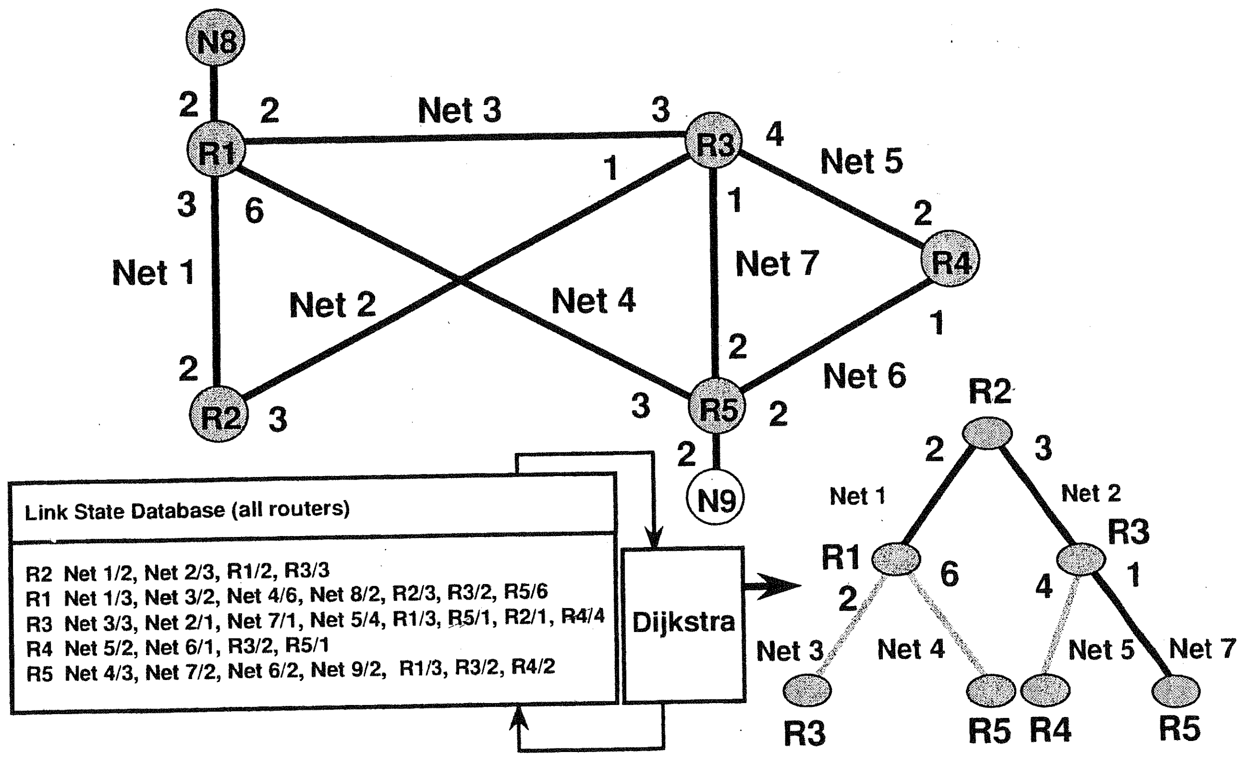
Step 4

The figure on the opposite page illustrates step 4.

In the third iteration, the links to R3 are considered. R2 discovers that from R3 there are paths to R4 and R5 (by way of Net 5 and Net 7). A path to R1 is also available (Net 3), but the tree already contains the least-cost path to R1. Likewise, a path to R3 has already been added to the tree.

Using the SPF tree as a reference, there are also two paths to R5. R2 chooses Net 7 as the best path to R5 at a total cost of 4, and adds this as a branch.

Dijkstra's Algorithm—Step 4



Dijkstra's Algorithm—Step 5

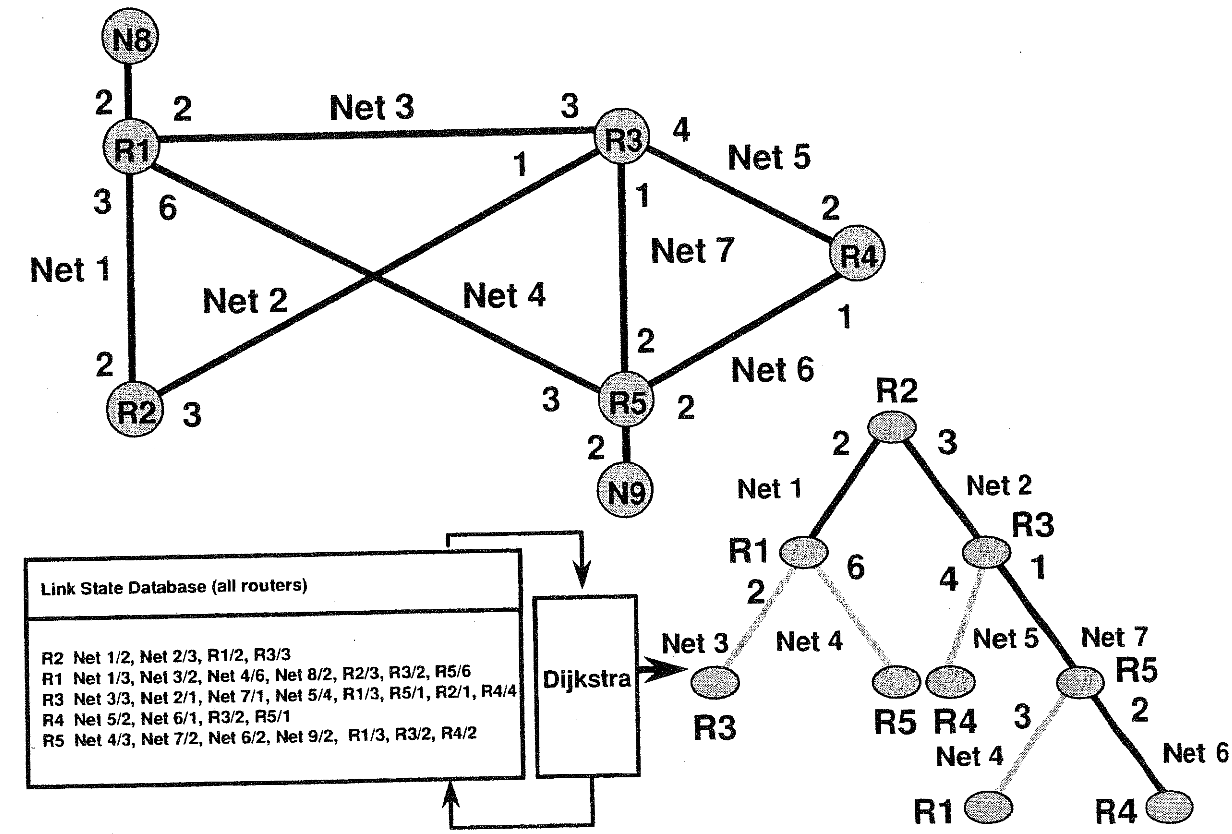
Step 5

The figure on the opposite page illustrates step 5.

Dijkstra is run from the perspective of R5. R2 discovers that from R5 there are paths to R1, R3, and R4 (Net 4, Net 7, Net 6). A path to R1 and R3 has been incorporated into the SPF tree. Using the SPF tree as a reference, there are two paths to R4 (Net 5 and Net 6). R2 chooses the lowest-cost path to R4 (Net 6) at a cost of 6, and adds that path as a branch.

The network N9 is ignored for now because it represents a stub network (no other routers are on it).

Dijkstra's Algorithm—Step 5



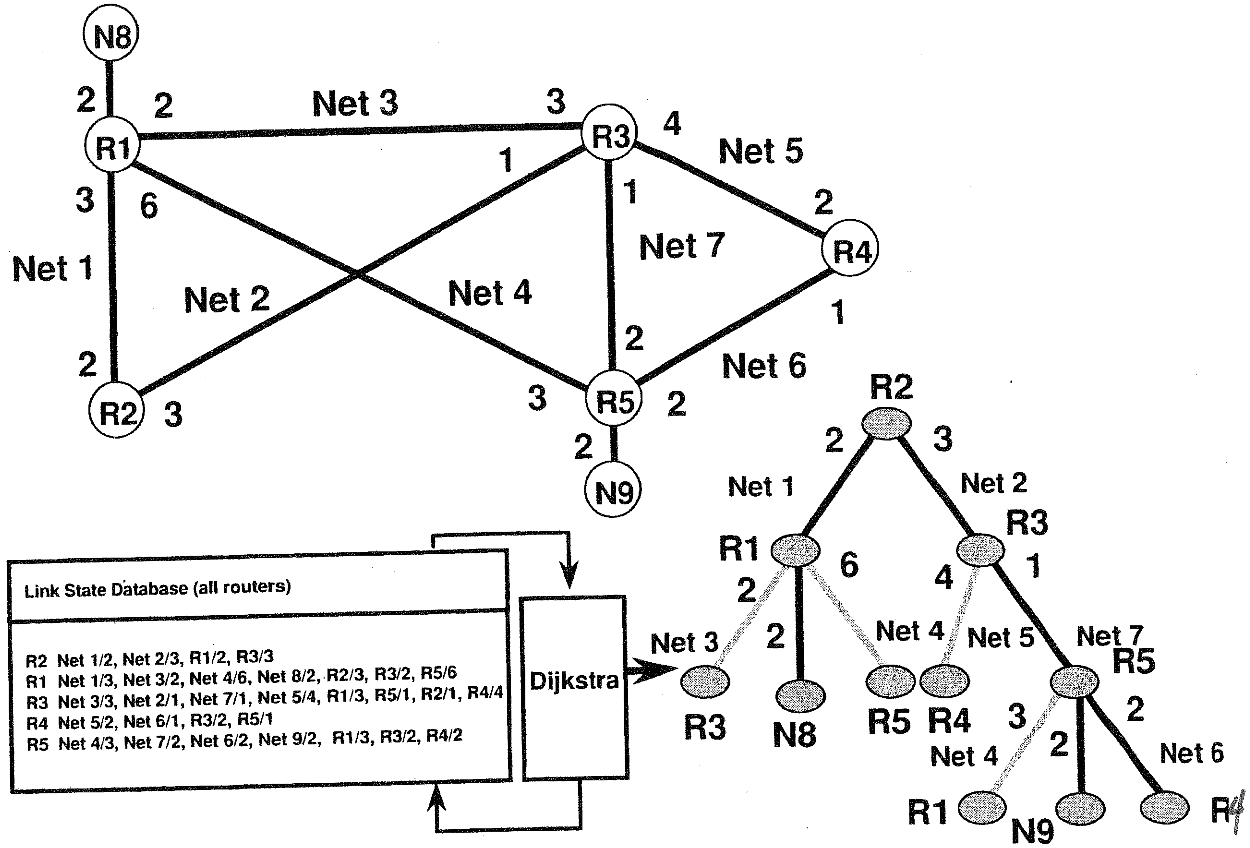
Dijkstra's Algorithm—Step 6

Step 6

The figure on the opposite page illustrates step 6.

Dijkstra is run one more time through the database. Stub networks (N8 and N9) are found and added as leaves to the SPF tree. Now R2 has learned the best path to all known destinations (networks and routers).

Dijkstra's Algorithm—Step 6



- RIP 1 updates ≠ indicate the type of Address in the address space of an update message (1)

- The Router interface must know the mask to apply it to the addr so as not to be ambiguous

By manipulating a RIP Request msg recall (cmd/response)
1 2

Family = 0

metric = 16

a router can obtain an entire Routing table at once (up to 25 entries)

Bellman Ford

RIP 2

added features to msg:

- Routing Domain allows Running of multiple concurrent RIPs

- Route Tag - if used w/ EGP's. indicate autonomous system #

- Subnet mask

- Next hop address

  Ø = indicates next hop is the addr that is sending the RIP msg

  Value = actual next hop addr.

(RIP 1) assumes next hop addr is the sending addr.

OSPF RFC 1247 Request for Comment

- Developed Due to Lack of STD & RIP of Inherent Problems

- ∄ send whole table as an update msg every 30 sec
- sends only its Link state info. to all routers

→ Least Cost Based

- Speed of Link
- Delay
- BER → Retrans → ↑
- Load Leveling (Balancing)
- Traffic type → Diffserv/TOS
- Reachability

- OSPF Link to IP in that it uses TOS Field & DA in IP DATAGRAM
- these Link state info are not encapsulated Like RIPs in Layer 2 LAN encap
- These are PACKETS.

- OSPF

  requires routers to be classified:

  - Internal - all nets Directly conn to this router belong to same area

    routers with Links only to Backbone RTS are internal

  - Border & internal
  - Backbone - intf to Backbone
  - Boundary - uses BGP/EGP with another Auto system

- No unreachable metric