Computer Networks

Mohammed El-Hajj

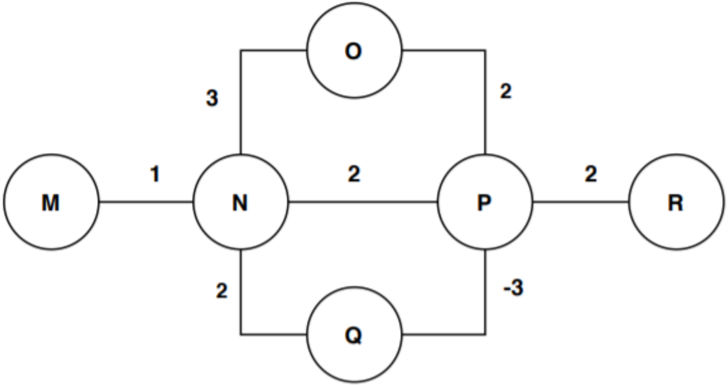
Jacobs University Bremen

October 10, 2021

Part 5: Internet Routing

Section 21: Distance Vector Routing (RIP)

Bellman-Ford

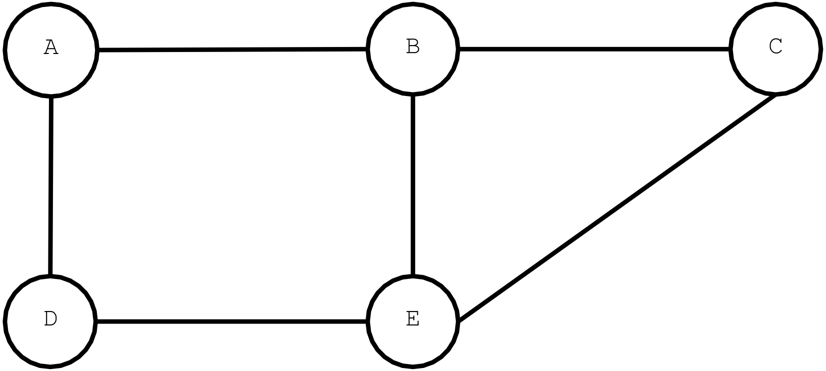


Bellman-Ford

- Let $G = (V, E)$ be a graph with the vertices V and the edges E with $n = |V|$ and $m = |E|$. Let D be an $n \times n$ distance matrix in which $D(i, j)$ denotes the distance from node $i \in V$ to the node $j \in V$.
- Let H be an $n \times n$ matrix in which $H(i, j) \in E$ denotes the edge on which node $i \in V$ forwards a message to node $j \in V$.
- Let M be a vector with the link metrics, S a vector with the start node of the links and D a vector with the end nodes of the links.

1. Set $D(i, j) = \infty$ for $i \neq j$ and $D(i, j) = 0$ for $i = j$.
2. For all edges $l \in E$ and for all nodes $k \in V$: Set $i = S[l]$ and $j = D[l]$ and $d = M[l] + D(j, k)$.
3. If $d < D(i, k)$, set $D(i, k) = d$ and $H(i, k) = l$.
4. Repeat from step 2 if at least one $D(i, k)$ has changed. Otherwise, stop.

Bellman-Ford Example (1/2)



Bellman-Ford Example(2/2)

Round 0:

A	Dest.	Link	Cost
	A	local	0

B	Dest.	Link	Cost
	B	local	0

C	Dest.	Link	Cost
	C	local	0

D	Dest.	Link	Cost
	D	local	0

E	Dest.	Link	Cost
	E	local	0

Round 1:

A	Dest.	Link	Cost
	A	local	0
	B	A-B	1
	D	A-D	1

B	Dest.	Link	Cost
	A	A-B	1
	B	local	0
	C	B-C	1
	E	B-E	1

C	Dest.	Link	Cost
	B	B-C	1
	C	local	0
	E	C-E	1

D	Dest.	Link	Cost
	A	A-D	1
	D	local	0
	E	D-E	1

E	Dest.	Link	Cost
	B	B-E	1
	C	C-E	1
	D	D-E	1
	E	local	0

Round 2:

A	Dest.	Link	Cost
	A	local	0
	B	A-B	1
	C	A-B	2
	D	A-D	1
	E	A-B	2

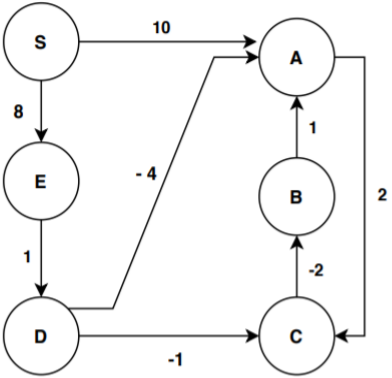
B	Dest.	Link	Cost
	A	A-B	1
	B	local	0
	C	B-C	1
	D	A-B	2
	E	B-E	1

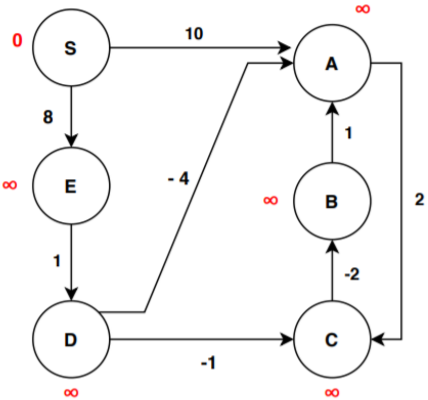
C	Dest.	Link	Cost
	A	B-C	2
	B	B-C	1
	C	local	0
	D	C-E	2
	E	C-E	1

D	Dest.	Link	Cost
	A	A-D	1
	B	A-D	2
	C	D-E	2
	D	local	0
	E	D-E	1

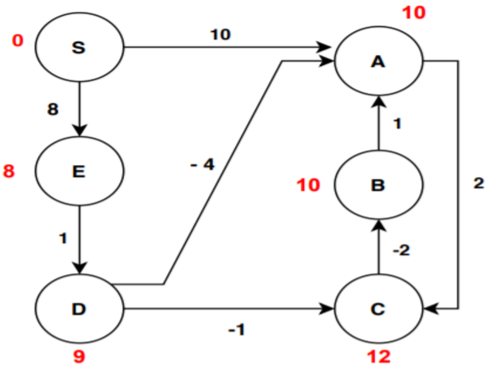
E	Dest.	Link	Cost
	A	B-E	2
	B	B-E	1
	C	C-E	1
	D	D-E	1
	E	local	0

Bellman-Ford Example-2





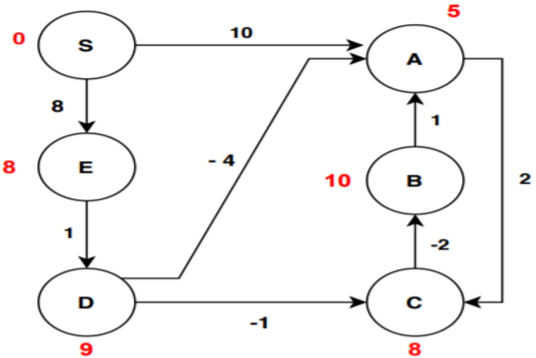
Bellman-Ford Example-2



$$D[A] > D[S] + W[S,A] \implies \infty > 0 + 10 \implies \infty > 10 \implies True$$

$$D[A] = D[S] + W[S,A] \implies D[A] = 0 + 10 = 10$$

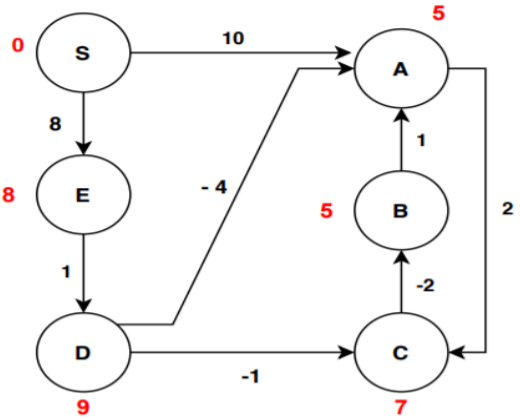
Bellman-Ford Example-2



$$D[C] > D[D] + W[D,C] \implies 12 > 9 + (-1) \implies 12 > 8 \implies True$$

$$D[C] = D[D] + W[D,C] \implies D[C] = 9 + (-1) = 8$$

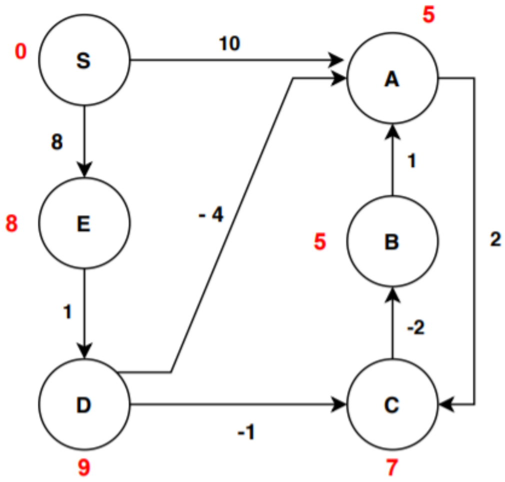
Bellman-Ford Example-2



$$D[C] > D[A] + W[A,C] \implies 8 > 5 + 2 \implies 8 > 7 \implies True$$

$$D[C] = D[A] + W[A,C] = 5 + 2 = 7$$

Bellman-Ford Example-2



Count-to-Infinity

- Consider the following topology:

  A --- B --- C

- After some distance vector exchanges, C has learned that he can reach A by sending packets via B.
- When the link between A and B breaks, B will learn from C that he can still reach A at a higher cost (count of hops) by sending a packet to C.
- This information will now be propagated to C, C will update the hop count and subsequently announce a more expensive not existing route to B.
- This counting continues until the costs reach infinity.

Split Horizon

- Idea: Nodes never announce the reachability of a network to neighbors from which they have learned that a network is reachable.
- Does not solve the count-to-infinity problem in all cases:

```
A --- B
 \   /
   C
   |
   D
```

- If the link between C and D breaks, B will not announce to C that it can reach D via C and A will not announce to C that it can reach D via C (split horizon).
- But after the next round of distance vector exchanges, A will announce to C that it can reach D via B and B will announce to C that it can reach D via A.

Split Horizon with Poisoned

- Idea: Nodes announce the unreachability of a network to neighbors from which they have learned that a network is reachable.
- Does not solve the count-to-infinity problem in all cases:

```
A --- B
 \   /
   C
   |
   D
```

- C now actively announces infinity for the destination D to A and B.
- However, since the exchange of the distance vectors is not synchronized to a global clock, the following exchange can happen:

Split Horizon with Poisoned

1. C first announces infinity for the destination D to A and B.
2. A and B now update their local state with the metric infinity for the destination D directly via C. The other stale information to reach D via the other directly connected node is not updated.
3. A and B now send their distance vectors. A and B now learn that they can not reach D via the directly connected nodes. However, C learns that it can reach D via either A or B.
4. C now sends its distance vector which contains false information to A and B and the count-to-infinity process starts.

Routing Information Protocol

- The Routing Information Protocol version 2 (RIP-2) defined in RFC 2453 is based on the Bellman-Ford algorithm.
- RIP defines infinity to be 16 hops. Hence, RIP can only be used in networks where the longest paths (the network diameter) is smaller than 16 hops.
- RIP-2 runs over the User Datagram Protocol (UDP) and uses the well-known port number 520.
- RIPng, defined in RFC 2080, adds support for IPv6 and uses UDP port 521.
- RIPng assumes that security is provided using IPv6 security mechanisms.

RIPng Message

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  command (1)  |  version (1)  |       must be zero (2)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                Route Table Entry 1 (20)                       ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                         ...                                   ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                Route Table Entry N (20)                       ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- The command field indicates, whether the message is a request or a response.
- The version field contains the protocol version number.

RIPng Route Table Entry

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                        IPv6 prefix (16)                       ~
|                                                               |
+---------------------------------------------------------------+
|         route tag (2)         | prefix len (1)|  metric (1)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

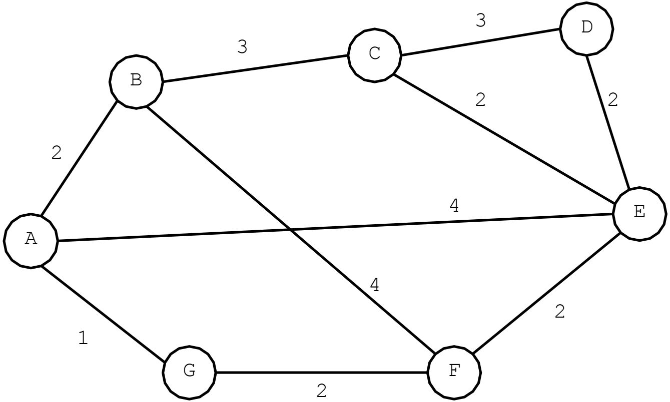
- The Route Tag field marks entries which contain external routes (established by an EGP).

Section 22: Link State Routing (OSPF)

Dijkstra's Shortest Path Algorithm

1. All nodes are initially labeled with infinite costs (the costs are not yet known).
2. The cost of the root node is set to 0, the root node is marked as the current node.
3. The current node's cost label is marked permanent.
4. For each node *A* adjacent to the current node *C* , the costs for reaching *A* are calculated as the costs of *C* plus the costs for the link from *C* to *A*. If the sum is less than *A*'s cost label, the label is updated with the new cost and the name ofthe current node.
5. If there are still nodes with tentative cost labels, a node with the smallest costs is selected as the new current node. Goto step 3 if a new current node was selected.
6. The shortest paths to a destination node is given by following the labels from the destination node towards the root.

Dijkstra Example



Open Shortest Path First (OSPF)

- The Open Shortest Path First (OSPF) protocol defined in RFC 2328 is a link state routing protocol.
- OSPF version 3 is defined in RFC 5340 and supports IPv6.
- Every node independently computes the shortest paths to all the other nodes by using Dijkstra’s shortest path algorithm.
- The link state information is distributed by flooding.
- OSPF introduces the concept of areas in order to control the flooding and computational processes.
- An OSPF area is a group of a set of networks within an autonomous system.
- The internal topology of an OSPF area is invisible for other OSPF areas. The routing within an area (intra-area routing) is constrained to that area.

OSPF Areas

- The OSPF areas are inter-connected via the OSPF backbone area (OSPF area 0). A path from a source node within one area to a destination node in another area has three segments (inter-area routing):
  1. An intra-area path from the source to a so called area border router.
  2. A path in the backbone area from the area border of the source area to the area border router of the destination area.
  3. An intra-area path from the area border router of the destination area to the destination node.

OSPF Router Classification

- OSPF routers are classified according to their location in the OSPF topology:
  1. *Internal Router*: A router where all interfaces belong to the same OSPF area.
  2. *Area Border Router*: A router which connects multiple OSPF areas. An area border router has to be able to run the basic OSPF algorithm for all areas it is connected to.
  3. *Backbone Router*: A router that has an interface to the backbone area. Every area border router is automatically a backbone router.
  4. *AS Boundary Router*: A router that exchanges routing information with routers belonging to other autonomous systems.

OSPF Stub Areas

- *Stub Areas* are OSPF areas with a single area border router.
- The routing in stub areas can be simplified by using default forwarding table entries, which significantly reduces the overhead.

OSPF Message Header

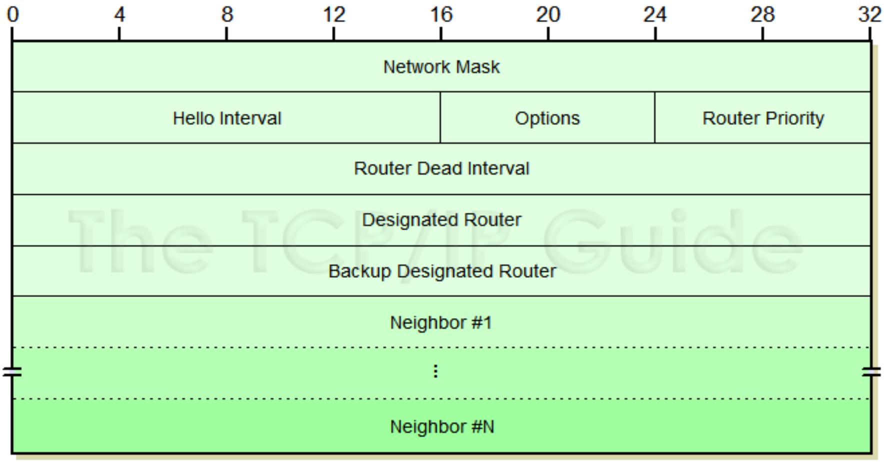
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version #   |     Type      |         Packet Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Router ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Area ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |             AuType            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Authentication                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: Indicates the type of OSPF message:

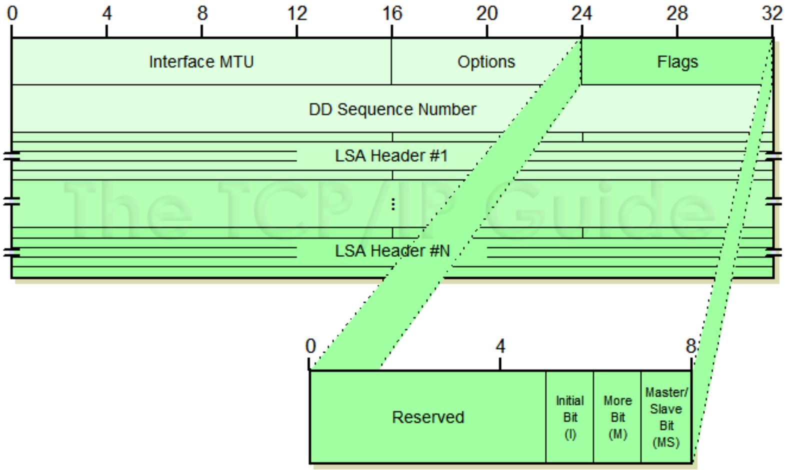
Type Value	OSPF Message Type
1	*Hello*
2	*Database Description*
3	*Link State Request*
4	*Link State Update*
5	*Link State Acknowledgment*

Authentication Type Value	OSPF Authentication Type
0	No Authentication
1	Simple Password Authentication
2	Cryptographic Authentication

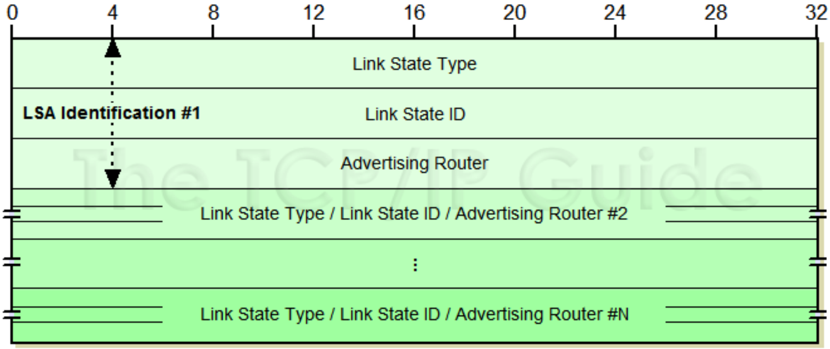
OSPF Hello Message



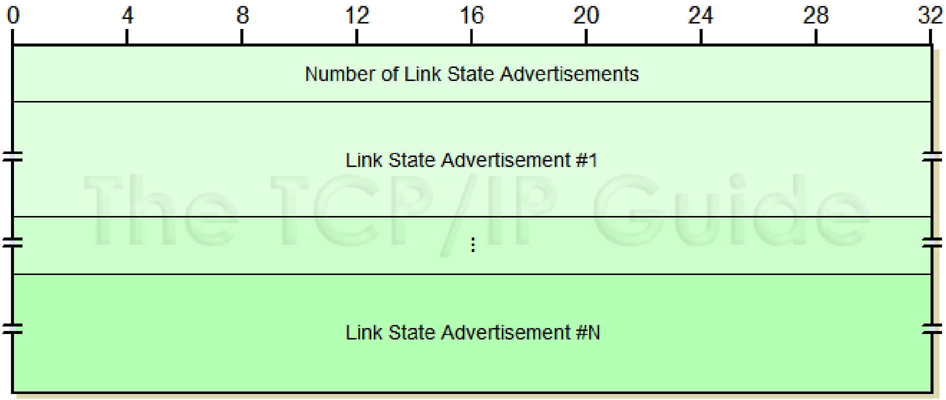
OSPF Database Description Message



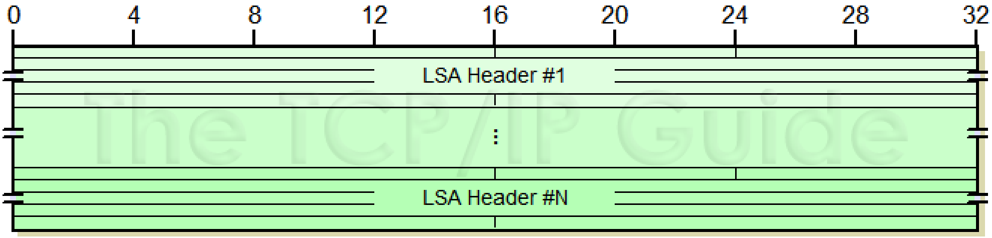
OSPF Link State Request Message



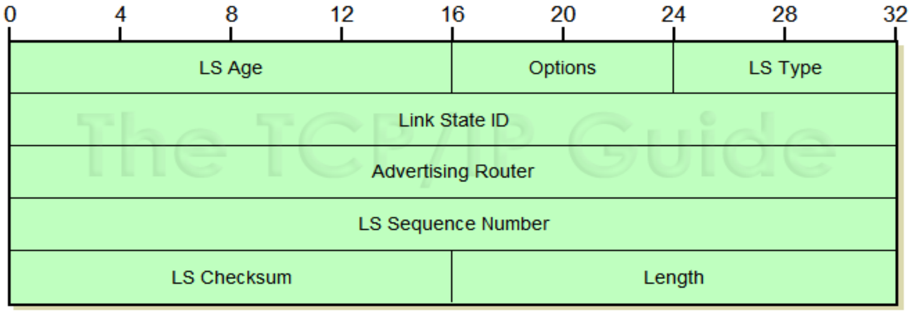
OSPF Link State Update Message



OSPF Link State Ack. Message



OSPF Link State Advertisement Header



Section 23: Path Vector Policy Routing (BGP)

Border Gateway Protocol (RFC 4271)

- The Border Gateway Protocol version 4 (BGP-4) exchanges reachability information between autonomous systems.
- BGP-4 peers construct AS connectivity graphs to
  - detect and prune routing loops and
  - enforce policy decisions.
- BGP peers generally advertise only routes that should be seen from the outside (advertising policy).
- The final decision which set of announced paths is actually used remains a local policy decision.
- BGP-4 runs over a reliable transport (TCP) and uses the well-known port 179.

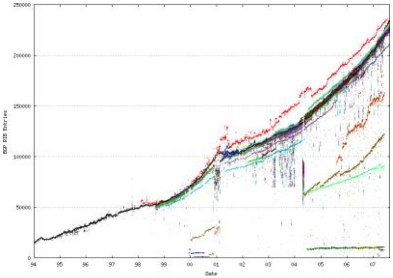
AS Categories (RFC 1772)

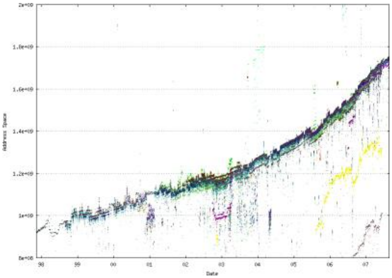
- *Stub AS*:
  - A Stub AS has only a single peering relationship to one other AS.
  - A Stub AS only carries local traffic.
- *Multihomed AS*:
  - A Multihomed AS has peering relationships with more than one other AS, but refuses to carry transit traffic.
- *Transit AS*:
  - A Transit AS has peering relationships with more than one other AS, and is designed (under certain policy restrictions) to carry both transit and local traffic.

Routing Policies

- Policies are provided to BGP in the form of configuration information and determined by the AS administration.
- Examples:
  1. A multihomed AS can refuse to act as a transit AS for other AS's. (It does so by only advertising routes to destinations internal to the AS.)
  2. A multihomed AS can become a transit AS for a subset of adjacent AS's, i.e., some, but not all, AS's can use the multihomed AS as a transit AS. (It does so by advertising its routing information to this set of AS's.)
  3. An AS can favor or disfavor the use of certain AS's for carrying transit traffic from itself.
- Routing Policy Specification Language (RFC 2622)

BGP Routing Table Statistics



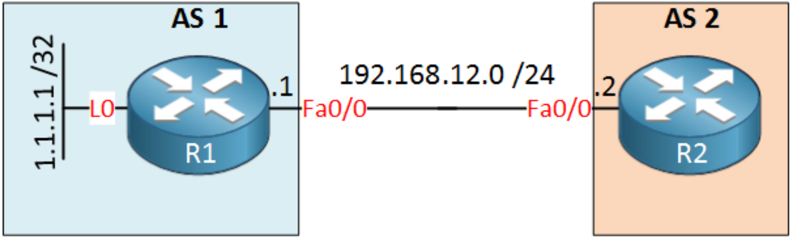


- http://bgp.potaroo.net/
- See also: G. Huston, ”The BGP Routing Table”, Internet Protocol Journal, March 2001

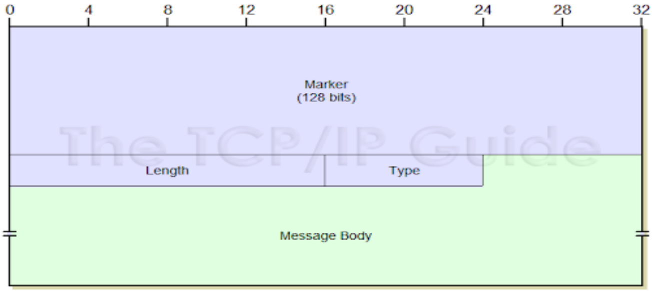
BGP-4 Phases and Messages

- Once the transport connection has been established, BGP-4 basically goes through three phases:
  1. The BGP4 peers exchange OPEN messages to open and confirm connection parameters
  2. The BGP4 peers exchange initially the entire BGP routing table. Incremental updates are sent as the routing tables change. Uses BGP UPDATE messages.
  3. The BGP4 peers exchange so called KEEPALIVE messages periodically to ensure that the connection and the BGP-4 peers are alive.
- Errors lead to a NOTIFICATION message and subsequent close of the transport connection.

BGP-4 Phases and Messages

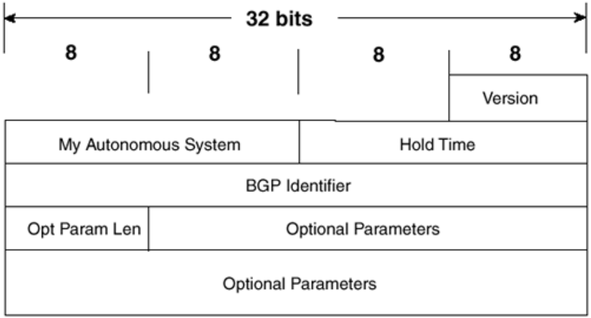


BGP-4 Message Header



- The Marker is used for authentication and synchronization.
- The Type field indicates the message type and the Lengthfield its length.

BGP-4 Open Message



BGP-4 Open Message

- The Version field contains the protocol version number.
- The Autonomous System Number field contains the 16-bit AS number of the sender.
- The Hold Time field specifies the maximum time that the receiver should wait for a response from the sender.
- The BGP Identifier field contains a 32-bit value which uniquely identifies the sender.
- The Opt Parm Len field contains the total length of the Optional Parameters field or zero if no optional parameters are present.
- The Optional Parameters field contains a list of parameters. Each parameter is encoded using a tag-length-value (TLV) triple.

BGP-4 Update Message

```
+-----------------------------------------------------+
|   Unfeasible Routes Length (2 octets)               |
+-----------------------------------------------------+
|  Withdrawn Routes (variable)                        |
+-----------------------------------------------------+
|   Total Path Attribute Length (2 octets)            |
+-----------------------------------------------------+
|    Path Attributes (variable)                       |
+-----------------------------------------------------+
|   Network Layer Reachability Information (variable) |
+-----------------------------------------------------+
```

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attr. Flags  |Attr. Type Code|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

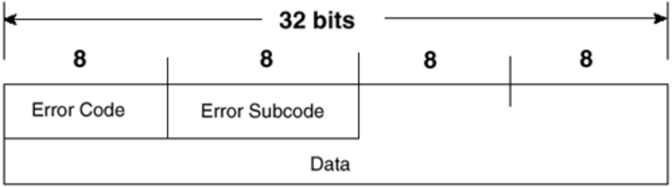
```
+---------------------------+
|   Length (1 octet)        |
+---------------------------+
|   Prefix (variable)       |
+---------------------------+
```

- The UPDATE message consists of two parts:
  1. The list of unfeasible routes that are being withdrawn.
  2. The feasible route to advertise.
- The Unfeasible Routes Length field indicates the total length of the Withdrawn Routes field in bytes.

BGP-4 Update

- The Withdrawn Routesfield contains a list of IPv4 address prefixes that are being withdrawn from service.
- The Total Path Attribute Length field indicates the total length of the Path Attributes field in bytes.
- The Path Attributes field contains a list of path attributes conveying information such as
  - the origin of the path information,
  - the sequence of AS path segments,
  - the IPv4 address of the next hop border router, or
  - the local preference assigned by a BGP4 speaker.
- The Network Layer Reachability Information field contains a list of IPv4 prefixes that are reachable via the path described in the path attributes fields.

BGP-4 Notification Message



- NOTIFICATION messages are used to reporterrors.
- The transport connection is closed immediately after sending a NOTIFICATION.
- Six error codes plus 20 sub-codes.

BGP-4 Keep Alive Message

- BGP-4 peers periodically exchange KEEPALIVE messages.
- A KEEPALIVE message consists of the standard BGP-4 header with no additional data.
- KEEPALIVE messages are needed to verify that shared state information is still present.
- If a BGP-4 peer does not receive a message within the Hold Time, then the peer will assume that there is a communication problem and tear down the connection.

BGP Communities

- BGP communities are 32 bit values used to convey user-defined information
- A community is a group of destinations which share some common property
- Some well-known communities, e.g.:
  - NO_EXPORT
  - NO_ADVERTISE
- Most take the form AS:nn (written as 701:120) where the meaning of nn (encoded in the last 16 bits) depends on the source AS (encoded in the first 16 bits)
- Mostly used for special treatment of routes

Internal BGP (iBGP)

- Use of BGP to distribute routing information within an AS.
- Requires to setup BGP sessions between all routers within an AS.
- Route Reflectors can be used to reduce the number of internal BGP sessions:
  - The Route Reflector collets all routing information and distributes it to all internal BGP routers.
  - Scales with $O(n)$ instead of $O(n^2)$ internal BGP sessions.
- BGP Confederations are in essence internal sub-ASes that do full mesh iBGP with a few BGP sessions interconnecting the sub-ASes.

BGP Route Selection (cbgp)

1. Ignore if next-hop is unreachable
2. Prefer locally originated networks
3. Prefer highest Local-Pref
4. Prefer shortest AS-Path
5. Prefer lowest Origin
6. Prefer lowest Multi Exit Discriminator (metric)
7. Prefer eBGP over iBGP
8. Prefer nearest next-hop
9. Prefer lowest Router-ID or Originator-ID
10. Prefer shortest Cluster-ID-List
11. Prefer lowest neighbor address

BGP’s and Count-to-Infinity

- BGP does not suffer from the count-to-infinity problem of distance vector protocols:
  - The AS path information allows to detect loops.
- However, BGP iteratively explores longer and longer (loop free) paths.

Multiprotocol BGP

- Extension to BGP-4 that makes it possible to distribute routing information for additional address families
- Announced as a capability in the open message
- Information for new protocol put into new path attributes
- Used to support IPv6, multicast, VPNs, . . .

References

G. Huston.
The BGP Routing Table.
The Internet Protocol Journal, 4(1), March 2001.

R. Chandra and P. Traina.
BGP Communities Attribute.
RFC 1997, Cisco Systems, August 1996.

Y. Rekhter, T. Li, and S. Hares.
A Border Gateway Protocol 4 (BGP-4).
RFC 4271, Juniper Networks, NextHop Technologies, January 2006.

I. van Beijnum.
BGP.
O'Reilly, September 2002.

B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig.
Interdomain Traffic Engineering with BGP.
IEEE Communications Magazine, 41(5):122–128, May 2003.

R. Mahajan, D. Wetherall, and T. Anderson.
Understanding BGP Misconfiguration.
In *Proc. SIGCOMM 2002*. ACM, August 2002.

J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz.
BGP Routing Dynamics Revisited.
SIGCOMM Computer Communication Review, 37(2), April 2007.