Leon Bruckman

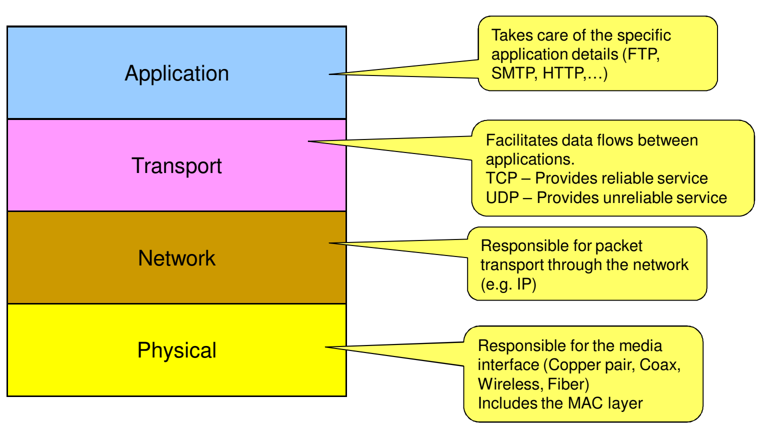
IP Networks

- The goal of the TCP/IP protocol suite is to provide data communication between computing devices
- TCP/IP is the basis for the Internet and provides world wide communications
- TCP/IP protocols are developed in layers
  - Each layer has its own tasks and a simple and well defined interface to the upper and lower layers
- TCP/IP protocols are developed according to standards
  - Mostly IETF and IEEE
  - Not all defined options are actually used
  - Some of the protocols are ambiguous

1

Leon Bruckman

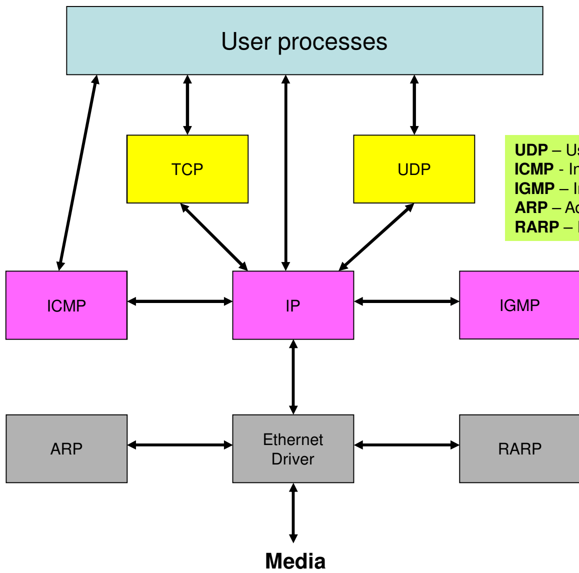
TCP/IP layering model



2

Leon Bruckman

Main TCP/IP suite protocols

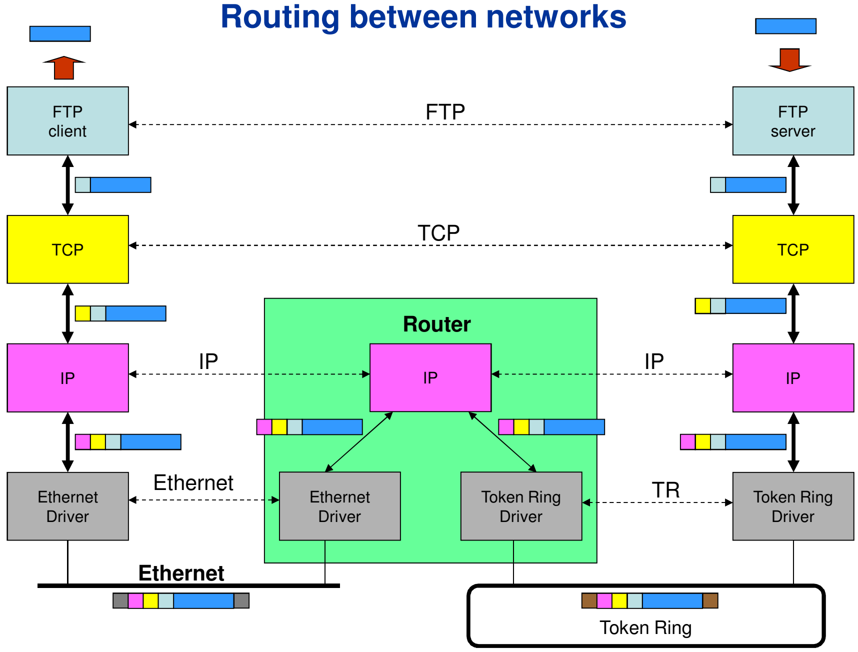


UDP – User Datagram Protocol
ICMP - Internet Control Message Protocol
IGMP – Internet Group Management Protocol
ARP – Address Resolution Protocol
RARP – Reverse ARP

3

Leon Bruckman

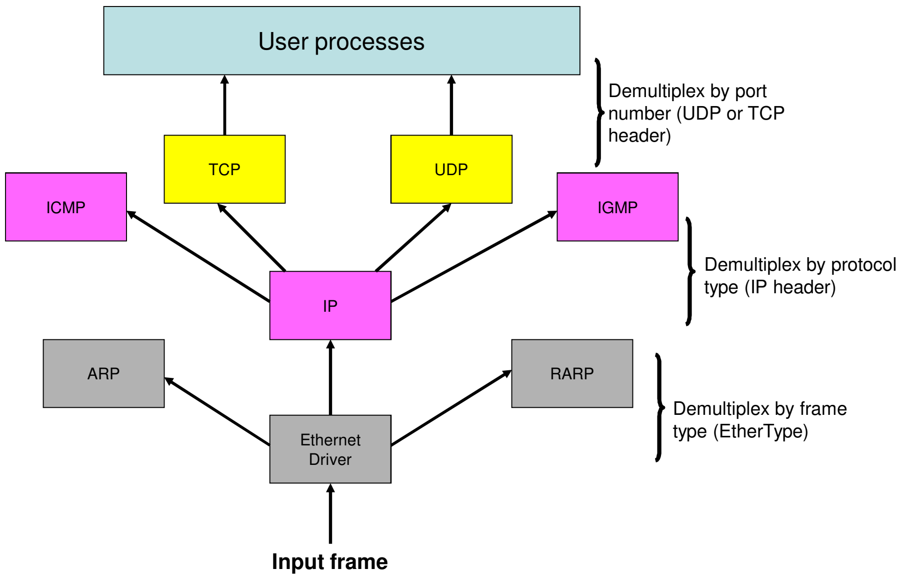
Routing between networks



4

Leon Bruckman

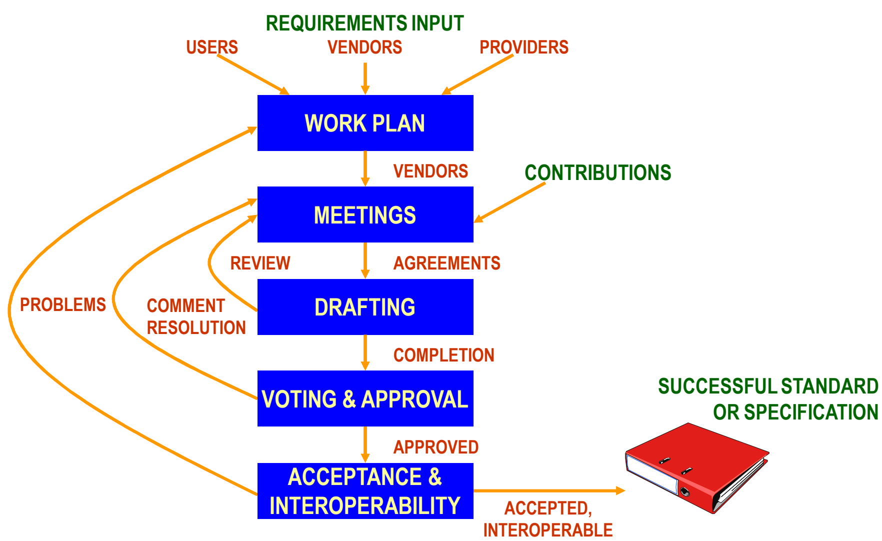
Demultiplexing



5

Leon Bruckman

The standardization process



6

Leon Bruckman

Internet Engineering Task Force (IETF)

- IETF is the group that writes most of the TCP/IP suite standards
- IETF is divided into AREAS that are responsible for different aspect of the networking
  - AREAS are divided into Working Groups
- Standards issued by IETF are known as Request For Comments (RFCs)
  - RFCs are numbered
  - For example RFC 768 defines the UDP
- IETF RFCs and drafts can be freely downloaded from the IETF site:
  - www.ietf.org
- Anyone can propose a draft, but for a draft to become an RFC it has to get the support of more than one company

7

Leon Bruckman

IP principles

- IP is the basis for all TCP/IP protocols
  - All the data of the UDP, TCP, ICMP, IGMP, etc protocols is transported through the network in IP Datagrams
- IP provides Connectionless Unreliable service
- Unreliable:
  - There is no guarantee for a datagram to reach its destination
  - If something goes wrong the IP layer may send back an ICMP message, further processing is the responsibility of higher layers
- Connectionless
  - No need to set up a "connection" before starting datagrams forwarding
  - IP does not keep any datagram related information after the datagram leaves the router
  - Datagrams may arrive out of order
  - Advantage: Stateless

8

Leon Bruckman

The IP datagram

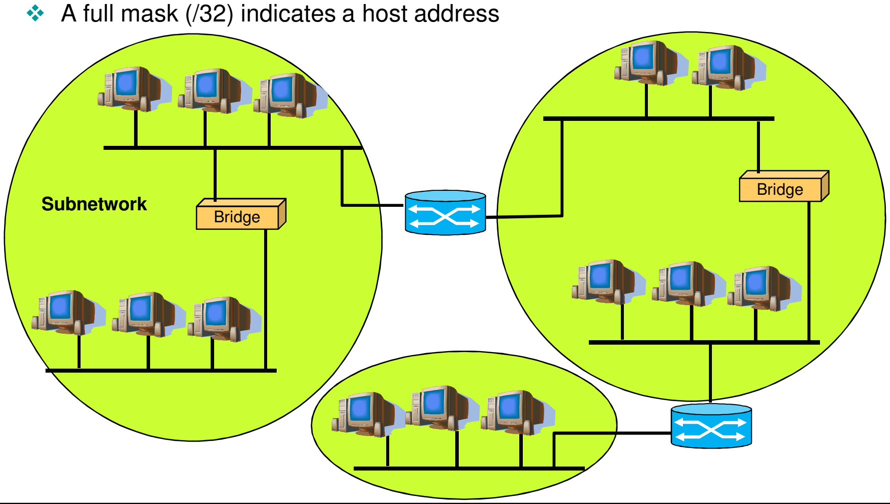
<table>
<tr><td>Version</td><td>IHL</td><td>Type of Service</td><td colspan="2">Total Datagram Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment offset</td></tr>
<tr><td colspan="2">Time To Live</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

9

Leon Bruckman

IP subnets

- Subnets are defined by an IP address and a Mask that indicates the “active” bits
  - Using Variable Length Subnet Masks (VLSM) the notation is: 210.15.16.0/21
- A full mask (/32) indicates a host address



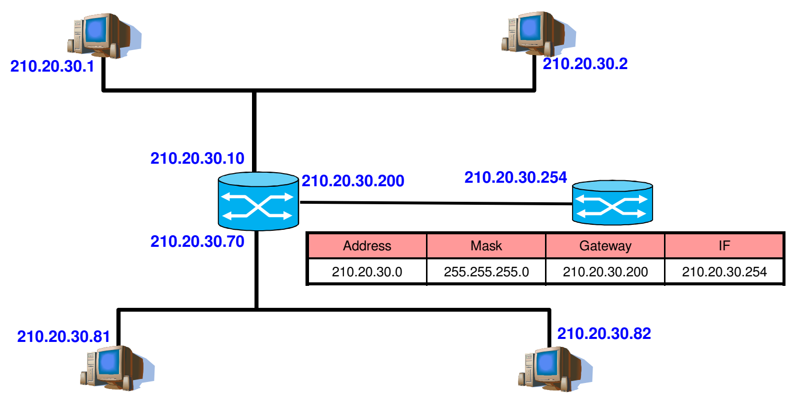
10

Leon Bruckman

Routing tables

Address	Mask	Gateway	IF
0.0.0.0	0.0.0.0	210.20.30.10	210.20.30.1
210.20.30.0	255.255.255.192	Local	210.20.30.1

Address	Mask	Gateway	IF
0.0.0.0	0.0.0.0	210.20.30.10	210.20.30.2
210.20.30.0	255.255.255.192	Local	210.20.30.2



Address	Mask	Gateway	IF
210.20.30.0	255.255.255.0	210.20.30.200	210.20.30.254

Address	Mask	Gateway	IF
0.0.0.0	0.0.0.0	210.20.30.70	210.20.30.81
210.20.30.64	255.255.255.192	Local	210.20.30.81

Address	Mask	Gateway	IF
0.0.0.0	0.0.0.0	210.20.30.70	210.20.30.82
210.20.30.64	255.255.255.192	Local	210.20.30.82

11

Leon Bruckman

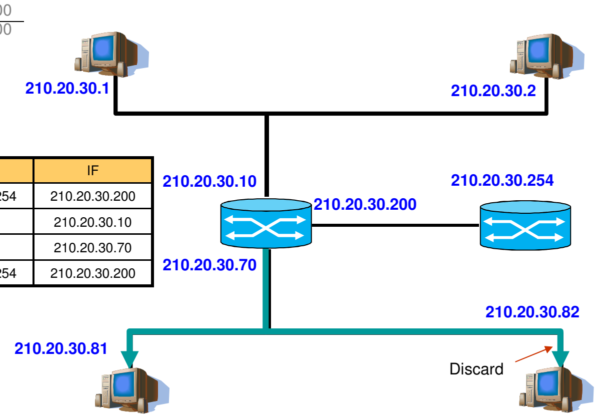
Routing tables - continued

IP DA = 210.20.30.81

DA = 11010010 00010100 00011110 0101001
MASK = 11111111 11111111 11111111 11000000
RES = 11010010 00010100 00011110 01000000

Entry = 210.20.30.64

Address	Mask	Gateway	IF
0.0.0.0	0.0.0.0	210.20.30.254	210.20.30.200
210.20.30.0	255.255.255.192	Local	210.20.30.10
210.20.30.64	255.255.255.192	Local	210.20.30.70
210.20.30.192	255.255.255.192	210.20.30.254	210.20.30.200



12

Leon Bruckman

Routing protocols

- Routing is the process of selecting paths in a network along which to send network traffic.
- In packet switching networks, routing directs packet forwarding, the transit of logically addressed packets from their source toward their ultimate destination through intermediate nodes
  - Intermediate nodes are typically hardware devices called routers, bridges, gateways, firewalls, or switches.
- Small networks may involve manually configured routing tables (static routing) or non-adaptive routing, while larger networks involve complex topologies and may change rapidly, making the manual construction of routing tables unfeasible.
- Dynamic routing dominates the Internet.
  - However, the configuration of the routing protocols often requires a skilled touch
  - One should not suppose that networking technology has developed to the point of the complete automation of routing.

13

Leon Bruckman

Distance vector algorithms

- Distance vector algorithms use the Bellman-Ford algorithm.
  - This approach assigns a number, the cost, to each of the links between each node in the network.
- Nodes will send information from point A to point B via the path that results in the lowest total cost (i.e. the sum of the costs of the links between the nodes used).
- The algorithm operates in a very simple manner.
  - When a node first starts, it only knows of its immediate neighbors, and the direct cost involved in reaching them.
  - Each node, on a regular basis, sends to each neighbor its own current idea of the total cost to get to all the destinations it knows of.
  - The neighboring node(s) examine this information, and compare it to what they already 'know'; anything which represents an improvement on what they already have, they insert in their own routing table(s).
  - Over time, all the nodes in the network will discover the best next hop for all destinations, and the best total cost.

14

Leon Bruckman

Link-state algorithms

- When applying link-state algorithms, each node uses as its fundamental data a map of the network in the form of a graph.
- To produce this, each node floods the entire network with information about what other nodes it can connect to, and each node then independently assembles this information into a map.
- Using this map, each router then independently determines the least-cost path from itself to every other node using a standard shortest paths algorithm such as Dijkstra's algorithm.
- The result is a tree rooted at the current node such that the path through the tree from the root to any other node is the least-cost path to that node.
  - This tree then serves to construct the routing table, which specifies the best next hop to get from the current node to any other node.

15

Leon Bruckman

Routing Information Protocol - RIP

- Distance vector routing protocol
- Serves as Internal Gateway Protocol (IGP)
  - Another protocol is needed for EGP
- Operates over UDP
- Best metric:
  - Assume **D(i,j)** is the least cost path from **i** to **j**
  - Assume **d(i,j)** is the cost of the path from **i** to **j**
    - **d** is infinite if **i** and **j** are not neighboring stations
  - The best path can be computed as:

$$D(i,j) = \min_{k} [d(i,k) + D(k,j)]$$

  - The best path starts at neighboring station k

16

Leon Bruckman

Basic RIP algorithm

- Each station keeps a routing table that includes an entry for each reachable destination
  - The entry includes the cost (**D**) and the gateway station (**G**)
- Each station sends to its neighbors updated regarding its routing table
  - These updates include all the routing table information
- When an update arrives from a neighboring station **G'** the following algorithm is implemented:
  - For every destination **N** add the cost of reaching **G'** to the cost received in the new message, call this new cost **D'**
  - Compare **D** to **D'**
  - If **D'** is less than **D** replace (**D**,**G**) with (**D'**,**G'**)
  - If the update was originated by **G**, always update
    - This allows to increase the cost of the selected gateway

17

Leon Bruckman

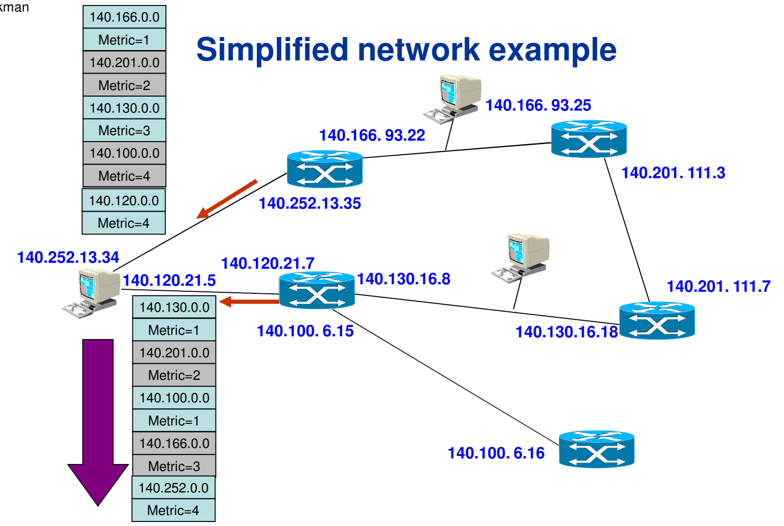
Handling network changes

- Every station transmits its data once every 30 seconds
  - To avoid stale information from failed stations to remain in the routing table, entries that are not refreshed during a 180 seconds period are removed
- A station may indicate that no path is available through it to a destination by indicating a cost of 16
- **Split Horizon**: A station does not send to a neighbor information that it learned from that neighbor.
  - **Reverse Poisoning:** It sends infinite cost
- Triggered updates: When the cost of a path changes the new values has to be forwarded immediately
  - To avoid updating storms the actual transmission is delayed between 1 and 5 seconds.

18

Leon Bruckman

Simplified network example

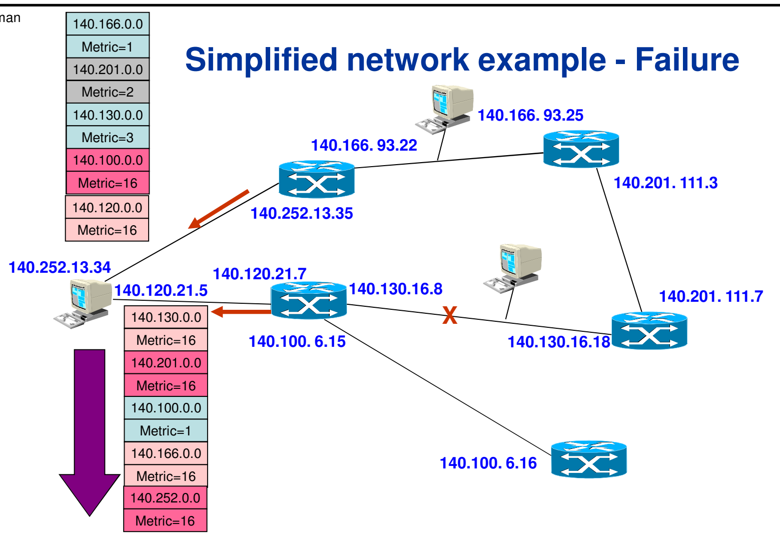


Address	Gateway	Metric
140.252.0.0/16	Local	1
140.120.0.0/16	Local	1
140.166.0.0/16	140.252.13.35	2
140.130.0.0/16	140.120.21.7	2
140.201.0.0/16	140.252.13.35	3
140.100.0.0/16	140.120.21.7	2

19

Leon Bruckman

Simplified network example - Failure



Address	Gateway	Metric
140.252.0.0/16	Local	1
140.120.0.0/16	Local	1
140.166.0.0/16	140.252.13.35	2
140.130.0.0/16	140.252.13.35	4
140.201.0.0/16	140.252.13.35	3
140.100.0.0/16	140.120.21.7	2

20

Leon Bruckman

RIP Limitations

- The maximum cost is limited to 15
  - Relevant for small networks only
- Resolving loops may take a long time
  - Somehow alleviated by Reverse Poisoning
  - See RFC 1058 section 2.2
- Cost is static
  - No relationship with network status
- RIP treats networks as flat networks – No area concept

21

Leon Bruckman

Open Shortest Path First - OSPF

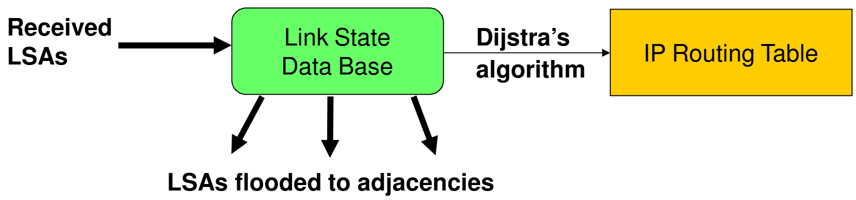
- Link state routing protocol
- Serves as Internal Gateway Protocol (IGP)
  - Another protocol is needed for EGP
- Operates directly over IP
  - Does not use UDP nor TCP
- It gathers link state information from available routers and constructs a topology map of the network.
  - The topology determines the routing table presented to the IP Layer which makes routing decisions based solely on the destination IP address found in IP packets.
- It computes the shortest path tree for each route using a method based on Dijkstra's algorithm, a shortest path first algorithm.
- OSPF detects changes in the topology, such as link failures, very quickly and converges on a new loop-free routing structure within seconds.

22

Leon Bruckman

OSPF principles

- Each router builds and maintains a Link State Data Base
  - It includes the local information and all information received from other routers
- Every output port has a defined cost
- Each router floods its routing data to its adjacencies (a sub-group of its neighbors) using Link State Advertisements (LSAs)
- All router shall have the same Link Sate Data Base
- Based on the Link State DB each router builds a tree with itself as the root and the destinations as leaves

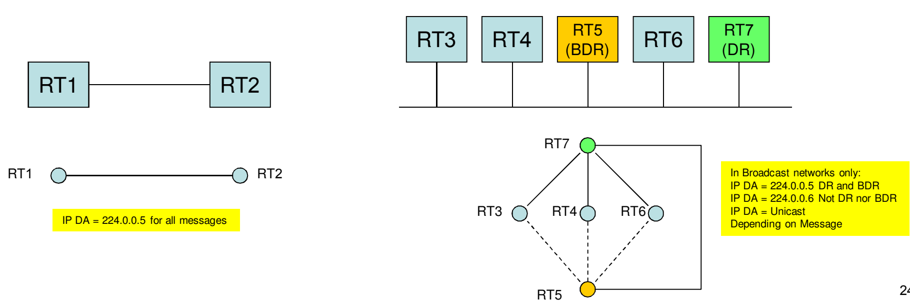


23

Leon Bruckman

Adjacencies

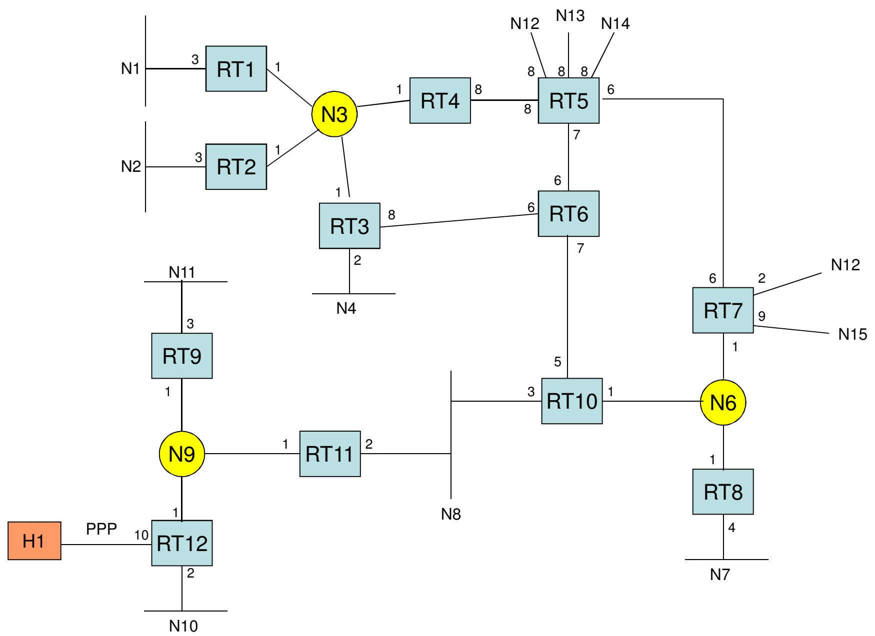
- Hello messages are used to discover the neighbors and to select a Designated Router (DR) and a Backup DR in networks that are not point to point
- Once the DR and Backup DR are selected adjacencies are defined
  - Not all neighbors become adjacent
- Information is flooded only to adjacencies
  - When a router is added to the network it has to set up an adjacency to the DR only



24

Leon Bruckman

Autonomous System (AS)



25

Leon Bruckman

Directed Graph

From

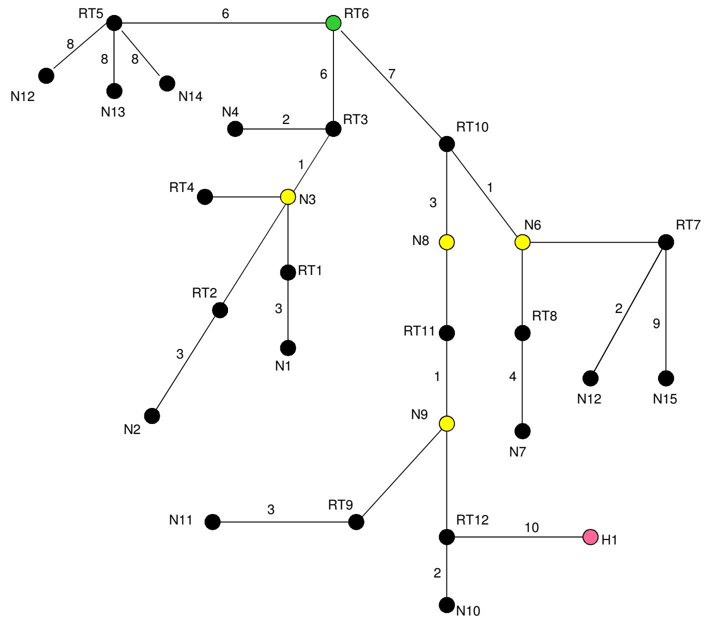
	RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT10	RT11	RT12	N3	N6	N8	N9
RT1													0			
RT2													0			
RT3						6							0			
RT4					8								0			
RT5				8		6	6									
RT6			8		7					5						
RT7					6									0		
RT8														0		
RT9																0
RT10						7								0	0	
RT11															0	0
RT12																0
N1	3															
N2		3														
N3	1	1	1	1												
N4			2													
N5																
N6							1	1		1						
N7								4								
N8										3	2					
N9									1		1	1				
N10												2				
N11									3							
N12					8		2									
N13					8											
N14					8											
N15							9									
H1																

To

26

Leon Bruckman

The Tree for RT6



27

Leon Bruckman

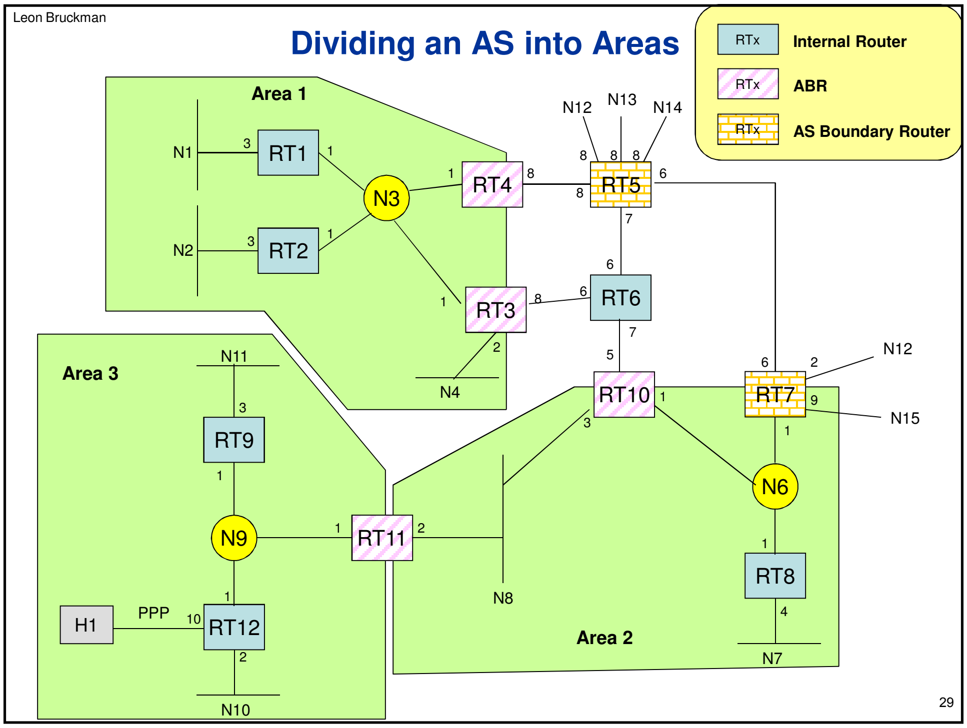
OSPF Areas

- OSPF allows to divide the AS into areas
- Each area has its own Link State Data Base
- The full area topology is known only to routers that belong to the area
  - This enables to improve scalability
- Each area is identified by a number that is similar to an IP address
  - The Backbone area is identified by 0.0.0.0
  - All areas must have a Area Border Router (ABR) to the Backbone area
- Routing between areas is implemented in 3 segments:
  - From the source router to the ABR in its area
  - From the ABR to the ABR in the destination area
  - From the ABR in the destination area to the destination

28

Leon Bruckman

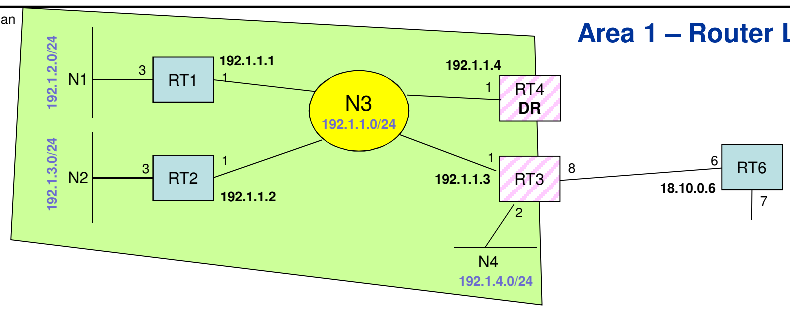
Dividing an AS into Areas



29

Leon Bruckman

Area 1 – Router LSA



RT3's router-LSA for Area 1

LS Age	0	Always true on origination
LS Type	1	Router LSA
Link State ID	192.1.1.3	RT3's Router ID
Advertising Router	192.1.1.3	RT3's Router ID
Bit E	0	Not an AS boundary router
Bit B	1	Area border router
Number of links	2	
Link ID	192.1.1.4	IP address of DR
Link Data	192.1.1.3	RT3's IP interface to net
Type	2	Connects to transit network
TOS metrics	0	
Metric	1	
Link ID	192.1.4.0	IP Network number
Link Data	0xffffff00	Network mask
Type	3	Connects to stub network
TOS metrics	0	
Metric	2	

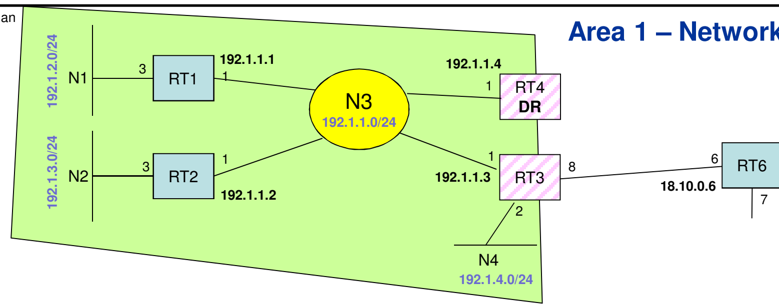
RT3's router-LSA for Backbone area

LS Age	0	Always true on origination
LS Type	1	Router LSA
Link State ID	192.1.1.3	RT3's Router ID
Advertising Router	192.1.1.3	RT3's Router ID
Bit E	0	Not an AS boundary router
Bit B	1	Area border router
Number of links	1	
Link ID	18.10.0.6	Neighbor's Router ID
Link Data	0.0.0.3	MIB-II ifIndex of P-P link
Type	1	Connects to router
TOS metrics	0	
Metric	8	

30

Leon Bruckman

Area 1 – Network LSA



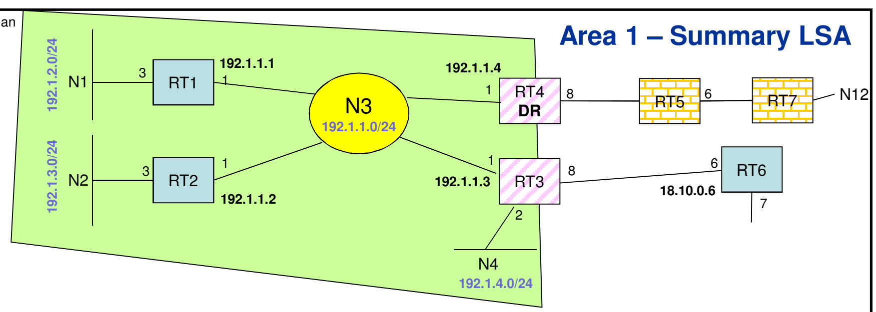
Network-LSA generated by RT4 on behalf of Network N3

LS Age	0	Always true on origination
LS Type	2	Network LSA
Link State ID	192.1.1.4	IP address of DR
Advertising Router	192.1.1.4	RT4's Router ID
Bit E	0	Not an AS boundary router
Bit B	1	Area border router
Network mask	0xffffff00	
Attached Router	192.1.1.4	Router ID
Attached Router	192.1.1.1	Router ID
Attached Router	192.1.1.2	Router ID
Attached Router	192.1.1.3	Router ID

31

Leon Bruckman

Area 1 – Summary LSA



Summary-LSA for Network N1, originated by Router RT4 into the backbone

LS Age	0	Always true on origination
LS Type	3	Summary LSA
Link State ID	192.1.2.0	N1's IP network number
Advertising Router	192.1.1.4	RT4's Router ID
Bit E	0	Not an AS boundary router
Bit B	1	Area border router
Mask	0xffffff00	
Metric	4	

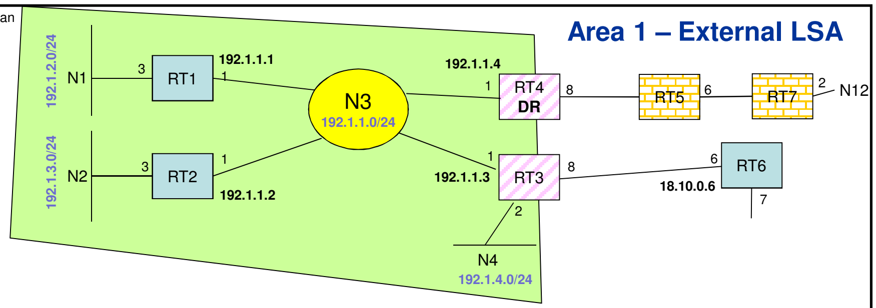
Summary-LSA for AS boundary router RT7, originated by Router RT4 into Area 1

LS Age	0	Always true on origination
LS Type	4	Summary LSA
Link State ID	RT7	Router RT7's ID
Advertising Router	192.1.1.4	RT4's Router ID
Bit E	0	Not an AS boundary router
Bit B	1	Area border router
Metric	14	

32

Leon Bruckman

Area 1 – External LSA



AS-external-LSA for Network N12, originated by Router RT7

LS Age	0	Always true on origination
LS Type	5	AS-external-LSA
Link State ID	N12	N12's IP network number
Advertising Router	RT7	RT7's Router ID
Bit E	1	AS boundary router
Bit B	1	Area border router
Mask	0xffffff00	N12's IP network mask
Metric	2	
Forwarding address	0.0.0.0	Forwarded to the advertising router (RT7)

33

Leon Bruckman

OSPF Limitations

- Cost is static
  - No relationship with network status
- Same route selected for all packets, even if more than one route is available
  - If the path are equivalent (same cost) then a proprietary Cisco process can be used to take advantage of the various paths: Equal Cost Multi Path (ECMP)
  - But, there are many issues with ECMP

34

Leon Bruckman

What is QoS ?

- The term Quality of Service (QoS) refers to **resource reservation** control mechanisms.
- QoS can provide different **priority** to different users or data flows, or **guarantee** a certain level of **performance** to a data flow in accordance with requests from the application program or the internet service provider policy.
  - Quality of Service guarantees are important if the network capacity is limited, for example in cellular data communication, especially for real-time streaming multimedia applications, for example voice over IP (VoIP) and IPTV, since these often require fixed bit rate and are delay sensitive.
- A network or protocol that supports QoS may agree on a **traffic contract** with the application software and reserve capacity in the network nodes.

35

Leon Bruckman

The difference between QoS and Classes of Service (CoS)

- CoS is a queuing discipline while QoS covers a wider range of techniques to manage bandwidth and network resources.
  - CoS classifies packets by examining packet parameters or CoS markings and places packets in queues of different priorities based on predefined criteria.
- QoS has to do with guaranteeing certain levels of network performance to meet service contracts or to support real-time traffic.
  - With QoS, some method is used to reserve bandwidth across a network in advance of sending packets.
- CoS is like classifying packages for delivery via regular mail, second-day delivery, or next-day delivery.
- QoS is what the delivery company does to ensure your packages are delivered on time (such as package tracking, air transport, door-to-door pickup and drop-off).

36

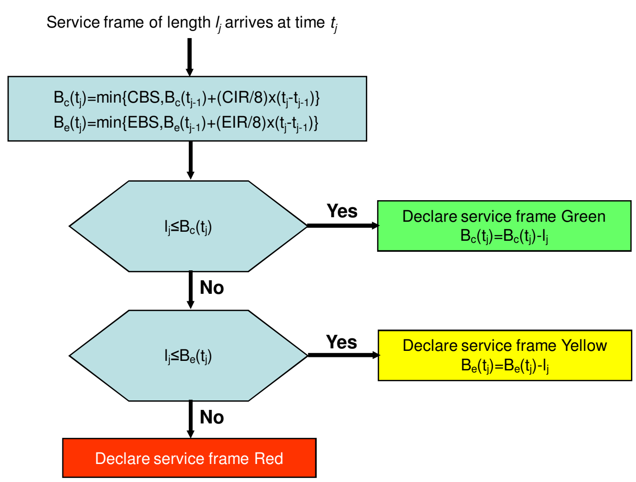
Leon Bruckman

The traffic contract

- Packet services are based on contracts “signed” at the time of service provisioning between the customer and the service provider
- The user commits to transmit at a specific rate with specific parameters:
  - Committed Information rate: CIR
  - Committed Burst Size: CBS
  - Excess Information Rate: EIR
  - Excess Burst Size: EBS
- The Service Provider commits to provide the service with the requested QoS:
  - Throughput
  - Delay
  - Delay variation

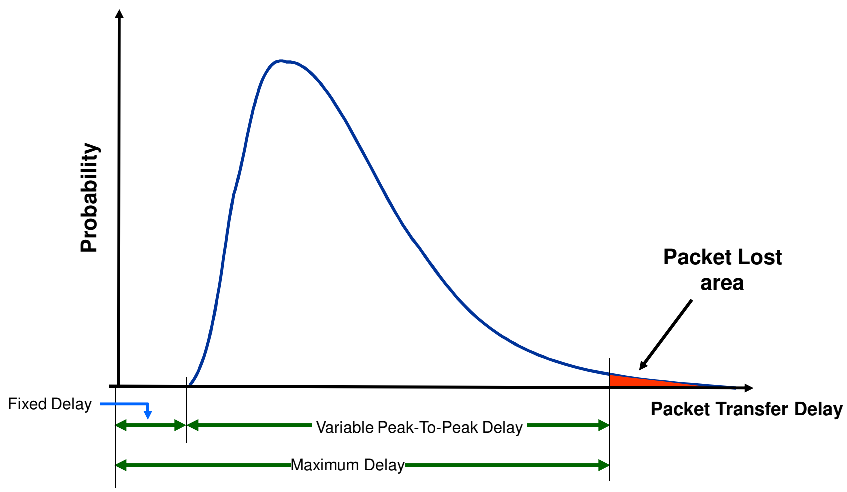
Leon Bruckman

Simplified policing algorithm



Leon Bruckman

Packet Transfer Delay Probability



39

Leon Bruckman

Connection Admission Control - CAC

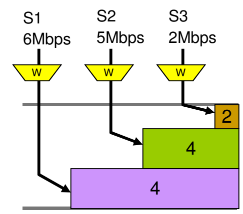
- The role of CAC is to decide whether there are sufficient free resources on the requested link to allow a new connection.
  - A connection can only be accepted if sufficient resources are available to establish the connection end-to-end with its required quality of service.
  - The agreed quality of service of existing connections in the network must not be affected by the new connection.
- If the network has the required resources, the CAC may allow a connection request to proceed; if not, the CAC will indicate this and notify the originator of the request that the request has been refused.
- CAC has a role only during connection provisioning and connection release

40

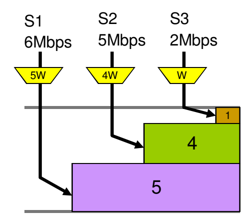
Leon Bruckman

Scheduling schemes

- Strict priority – simplest scheduling scheme
  - Queue n has strict priority over queue n+1
  - In bursty networks flows assigned to queue n may starve flows assigned to queue n+1
- Fair queueing
  - Each queue is served equally
    - Rate of queue ni = Rate / n
  - No prioritization
- Weighted fair queuing
  - Each queue is served according to a predetermined weight
    - Rate of queue ni = Rate * wi/(w1+w2+…+wn)
  - No starvation



Equal Fairness



Weighted Fairness

41

Leon Bruckman

Integrated Services (IntServ)

- IntServ or integrated services is an architecture that specifies the elements to guarantee quality of service (QoS) on networks.
- Flow Specs describe what the reservation is for, while RSVP is the underlying mechanism to signal it across the network.
- There are two parts to a Flow Spec:
  - What does the traffic look like?
    - Done in the Traffic SPECification part, also known as **TSPEC**.
    - TSPECs include token bucket algorithm parameters.
  - What guarantees does it need? Done in the service Request SPECification part, also known as **RSPEC**.
    - RSPECs specify what requirements there are for the flow. It can be:
    - **Best effort**: no reservation is needed. (EIR, EBS)
    - **Controlled Load** : there may be occasional glitches when two people access the same resource by chance, but generally both delay and drop rate are fairly constant at the desired rate. This setting is likely to be used by soft QoS applications.
    - **Guaranteed**: gives an absolutely bounded service, where the delay is promised to never go above a desired amount, and packets never dropped, provided the traffic stays within spec. (CIR, CBS)

42

Leon Bruckman

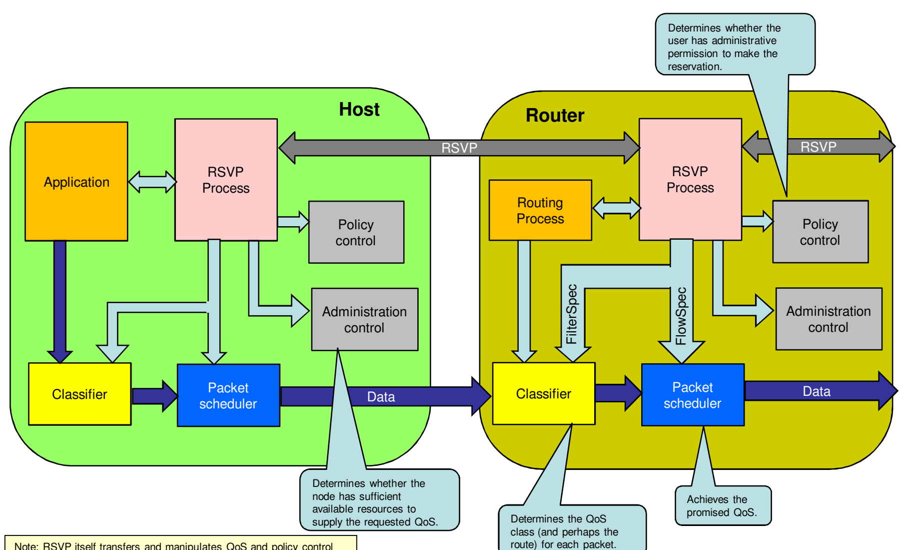
Resource Reservation Protocol (RSVP)

- RSVP is a Transport layer protocol designed to reserve resources across a network for an integrated services Internet.
  - It makes reservations for unidirectional data flows.
  - It is receiver-oriented, i.e., the receiver of a data flow initiates and maintains the resource reservation used for that flow.
  - It provides transparent operation through routers that do not support it.
- RSVP does not transport application data but is rather an Internet control protocol, like ICMP, IGMP, or routing protocols
  - It runs directly on top of IP with Protocol number=46
- RSVP is not itself a routing protocol; RSVP is designed to operate with current and future routing protocols.
- *RSVP by itself is rarely deployed in telecommunications networks today but the traffic engineering extension of RSVP, or RSVP-TE, is becoming more widely accepted nowadays in many QoS-oriented networks.*

43

Leon Bruckman

RSVP in hosts and routers



Note: RSVP itself transfers and manipulates QoS and policy control parameters as opaque data, passing them to the appropriate traffic control and policy control modules for interpretation.

44

Leon Bruckman

RSVP sessions

- RSVP defines a "session" to be a data flow with a particular destination and transport-layer protocol.
- RSVP treats each session independently.
- An RSVP session is defined by the triplet:
- (DestAddress, ProtocolId [, DstPort]).
  - DestAddress, the IP destination address of the data packets, may be a unicast or multicast address.
  - ProtocolId is the IP protocol ID.
  - The optional DstPort parameter is a "generalized destination port", i.e., some further demultiplexing point in the transport or application protocol layer.
    - DstPort could be defined by a UDP/TCP destination port field, by an equivalent field in another transport protocol, or by some application-specific information.

Leon Bruckman

Reservation styles

Creates a distinct reservation for data packets from a particular sender, not sharing them with other senders' packets for the same session.

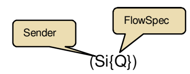
Creates a single reservation shared by selected upstream senders. Unlike the WF style, the SE style allows a receiver to explicitly specify the set of senders to be included.

Selection Sender	Reservations	
	Distinct	Shared
Explicit	Fixed Filter (FF)	Shared Explicit (SE)
Wildcard	Not defined	Wildcard Filter (WF)

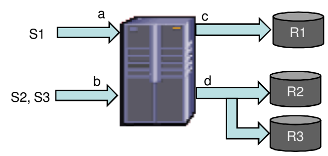
Creates a single reservation shared by flows from all upstream senders. This reservation may be thought of as a shared "pipe", whose "size" is the largest of the resource requests from all receivers, independent of the number of senders using it.

Leon Bruckman

Examples of styles



Data packets from each Si are routed to both outgoing interfaces.



a	c	
Sends	Reserves	Receives
WF(*{4B})	*{4B}	WF (* {4B})
b	d	
Sends	Reserves	Receives
WF(*{4B})	*{3B}	WF(*{3B})
		WF(*{2B})

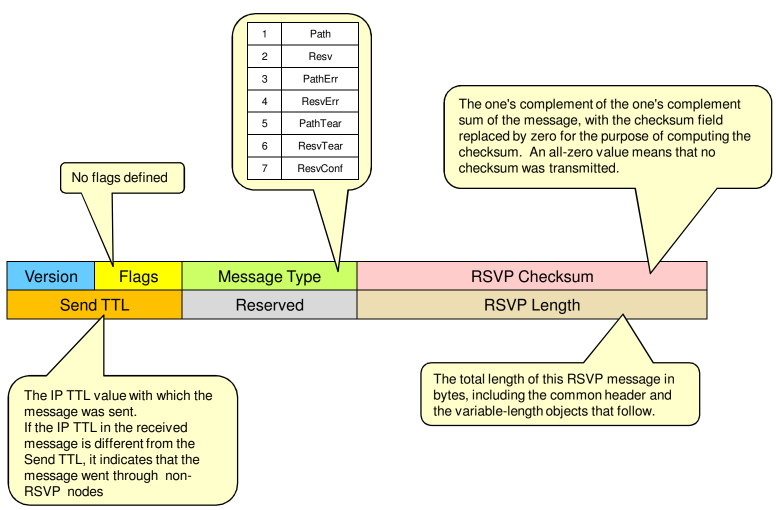
a	c	
Sends	Reserves	Receives
FF(S1{4B})	S1{4B}	FF(S1{4B}, S2{5B})
	S2{5B}	
b	d	
Sends	Reserves	Receives
FF(S2{5B}, S3{B})	S1{3B}	FF(S1{3B}, S3{B})
	S3{B}	FF(S1{B})

a	c	
Sends	Reserves	Receives
SE(S1{3B})	(S1,S2){B}	SE((S1,S2){B})
b	d	
Sends	Reserves	Receives
SE((S2,S3){3B})	(S1,S2,S3{3B})	SE((S1,S3){3B})
		SE(S2{2B})

47

Leon Bruckman

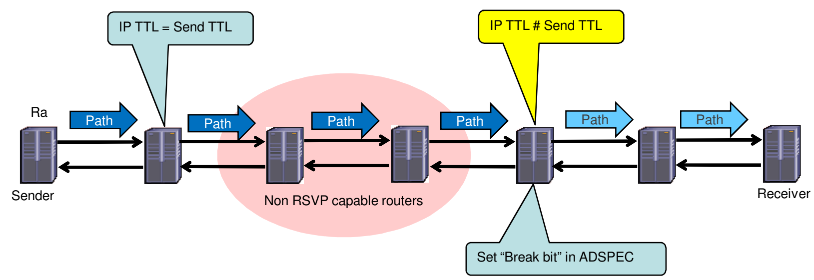
RSVP common header



48

Leon Bruckman

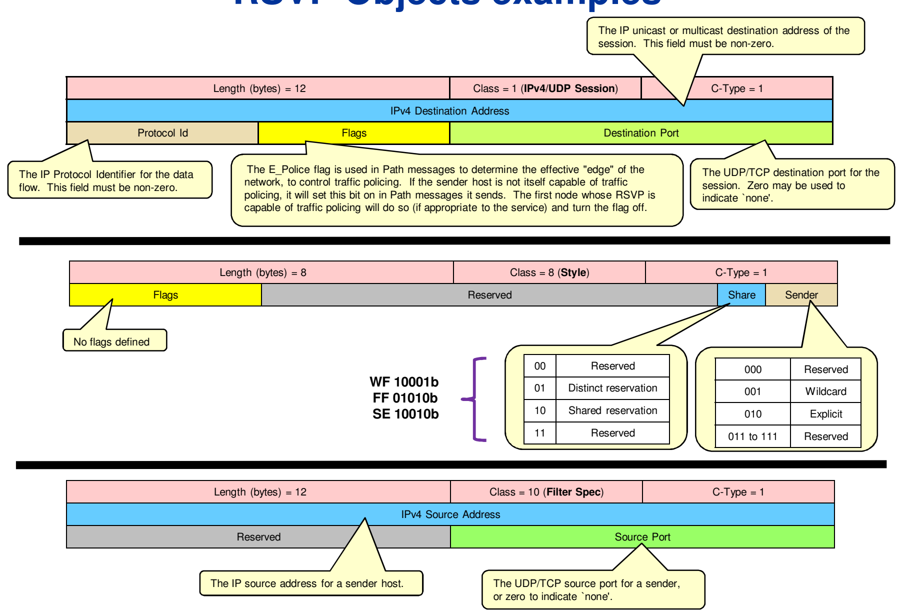
Non RSVP network elements discovery



49

Leon Bruckman

RSVP Objects examples



50

Leon Bruckman

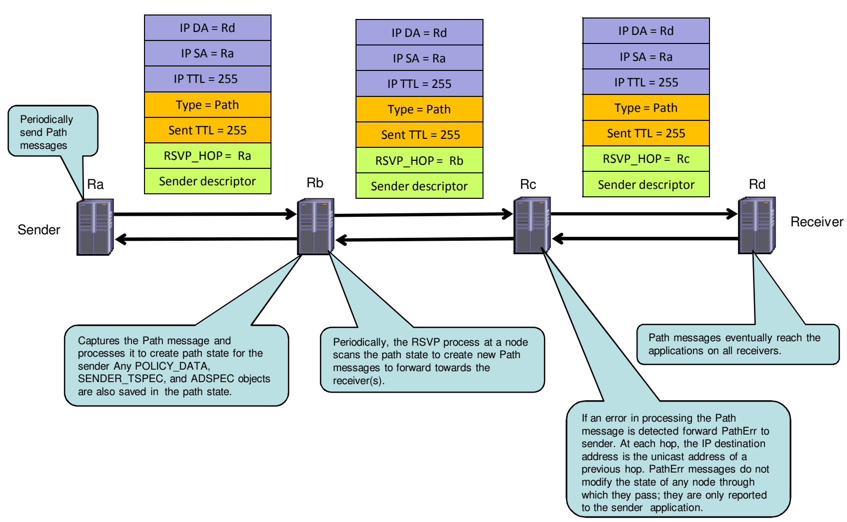
RSVP soft states

- RSVP takes a "soft state" approach to managing the reservation state in routers and hosts.
  - RSVP soft state is created and periodically refreshed by Path and Resv messages.
  - The state is deleted if no matching refresh messages arrive before the expiration of a "cleanup timeout" interval.
  - State may also be deleted by an explicit "teardown" message.
- At the expiration of each "refresh timeout" period and after a state change, RSVP scans its state to build and forward Path and Resv refresh messages to succeeding hops.
- When a route changes, the next Path message will initialize the path state on the new route, and future Resv messages will establish reservation state there.
  - The state on the now-unused segment of the route will time out.
- RSVP sends its messages as IP datagrams with no reliability enhancement.
  - Periodic transmission of refresh messages by hosts and routers is expected to handle the occasional loss of an RSVP message.

51

Leon Bruckman

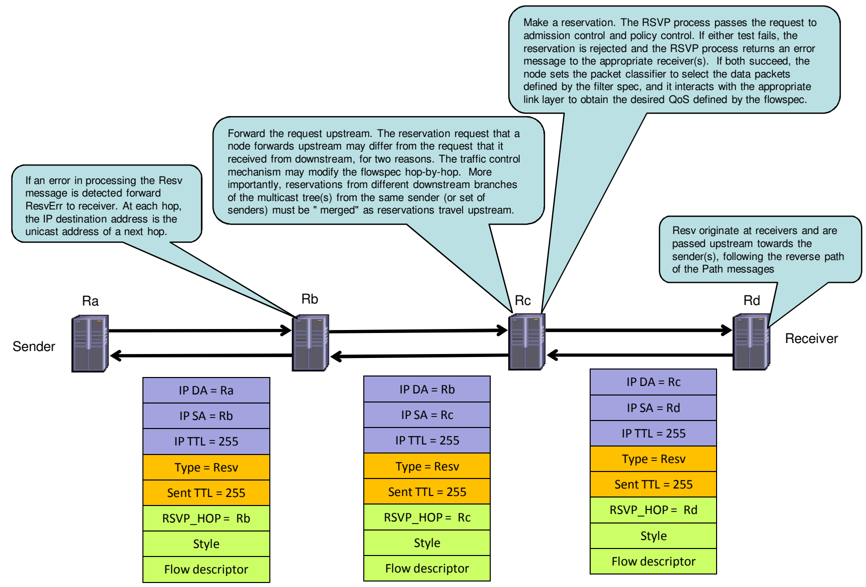
RSVP Path message flow



52

Leon Bruckman

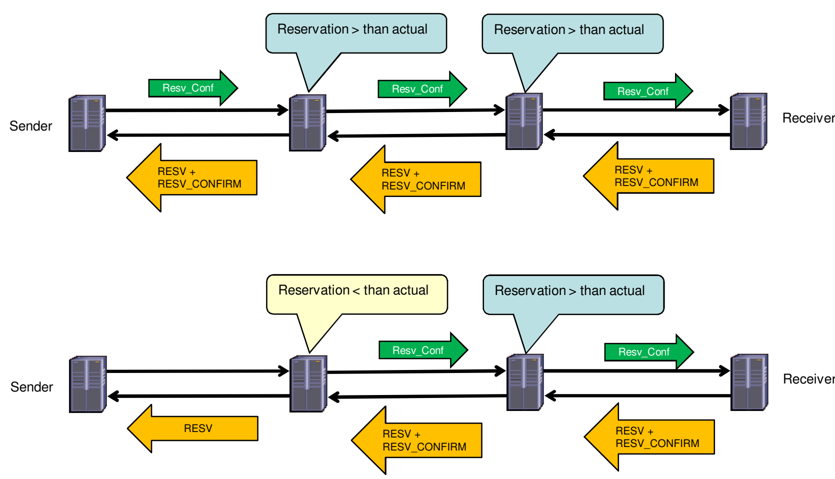
RSVP Resv message flow



53

Leon Bruckman

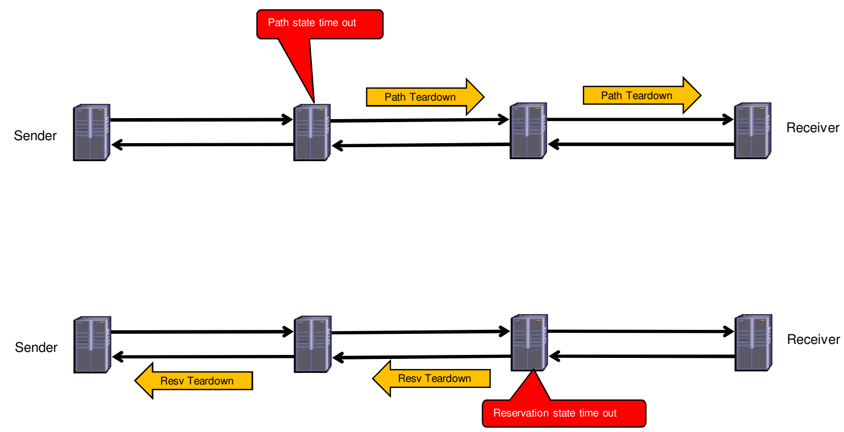
Confirmation



54

Leon Bruckman

Tear down



55

Leon Bruckman

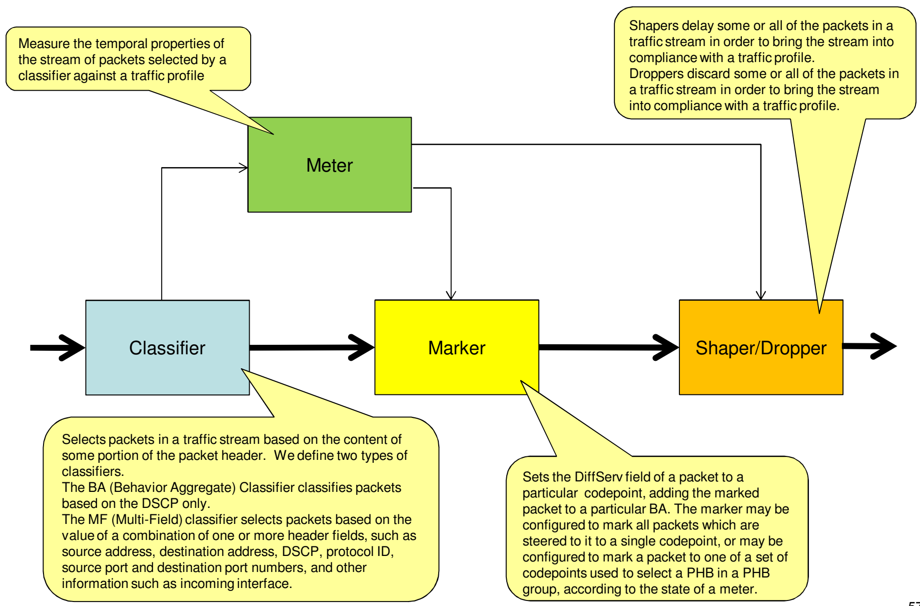
Differentiated Services - DiffServ

- The differentiated services architecture is based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different behavior aggregates.
- A DiffServ domain is a contiguous set of DiffServ nodes which operate with a common service provisioning policy and set of Per Hop Behavior (PHB) groups implemented on each node.
- In a DiffServ domain all the IP packets crossing a link and requiring the same DiffServ behavior are said to constitute a Behavior Aggregate (BA).
- At the ingress node of the DiffServ domain, the packets are classified and marked with a DiffServ Code Point (DSCP) which corresponds to their BA.
- At each transit node, the DSCP is used to select the PHB that determines the scheduling treatment and, in some cases, drop probability for each packet.

56

Leon Bruckman

Traffic Classification and Conditioning



57

Leon Bruckman

Per-Hop Behavior

- The Per-Hop Behavior (PHB) is indicated by encoding a 6-bit value—called the Differentiated Services Code Point (DSCP)—into the 8-bit Differentiated Services (DS) field of the IP packet header and IPv6 packet header.
- Most networks use the following commonly-defined Per-Hop Behaviors:
  - **Default PHB**—which is typically best-effort traffic
  - **Expedited Forwarding (EF) PHB**—dedicated to low-loss, low-latency traffic
    - EF traffic is often given strict priority queuing above all other traffic classes.
    - Because an overload of EF traffic will cause queuing delays and affect the jitter and delay tolerances within the class, EF traffic is often strictly controlled through admission control, policing and other mechanisms.
    - Typical networks will limit EF traffic to no more than 30%—and often much less—of the capacity of a link
  - **Assured Forwarding (AF) PHB**— which gives assurance of delivery under conditions
    - Assured forwarding allows the operator to provide assurance of delivery as long as the traffic does not exceed some subscribed rate.
    - Traffic that exceeds the subscription rate faces a higher probability of being dropped if congestion occurs.

58

Leon Bruckman

DSCP code points

Group	Code point	Description
EF	101110	Expedite forwarding
AF11	001010	Assured Forwarding low drop probability
AF12	001100	Assured Forwarding medium drop probability
AF13	001110	Assured Forwarding high drop probability
AF21	010010	Assured Forwarding low drop probability
AF22	010100	Assured Forwarding medium drop probability
AF23	010110	Assured Forwarding high drop probability
AF31	011010	Assured Forwarding low drop probability
AF32	011100	Assured Forwarding medium drop probability
AF33	011110	Assured Forwarding high drop probability
AF41	100010	Assured Forwarding low drop probability
AF42	100100	Assured Forwarding medium drop probability
AF43	100110	Assured Forwarding high drop probability
BE	000000	Best Effort

RSVP

Guaranteed

Controlled load

Best effort

Leon Bruckman

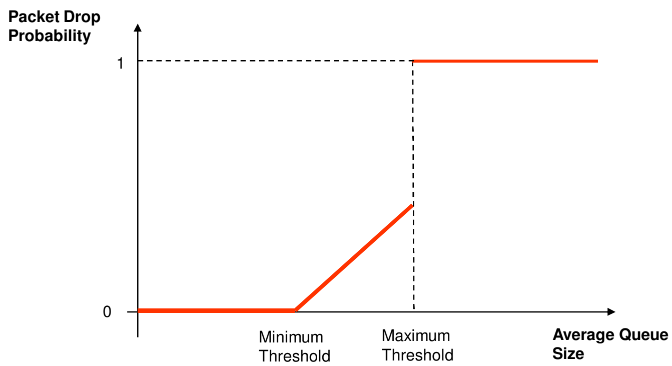
Weighted Random Early Detection - WRED

- Should congestion occur between classes, the traffic in the higher class is given priority. If congestion occurs within a class, the packets with the higher drop precedence are discarded first. To prevent issues associated with tail drop, the random early detection (RED) or weighted random early detection (WRED) algorithms are often used to drop packets.
- Random Early Detection (RED) is a congestion avoidance mechanism that takes advantage of TCP's congestion control mechanism.
  - By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate.
  - Assuming the packet source is using TCP, it will decrease its transmission rate until all the packets reach their destination, indicating that the congestion is cleared.
- Weighted RED (WRED) generally drops packets selectively based on IP precedence.
  - Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence.
  - Thus, higher priority traffic is delivered with a higher probability than lower priority traffic.

Leon Bruckman

WRED curves

- The minimum threshold value should be set high enough to maximize the link utilization. If the minimum threshold is too low, packets may be dropped unnecessarily, and the transmission link will not be fully used.
- The difference between the maximum threshold and the minimum threshold should be large enough to avoid global synchronization. If the difference is too small, many packets may be dropped at once, resulting in global synchronization.



61

Leon Bruckman

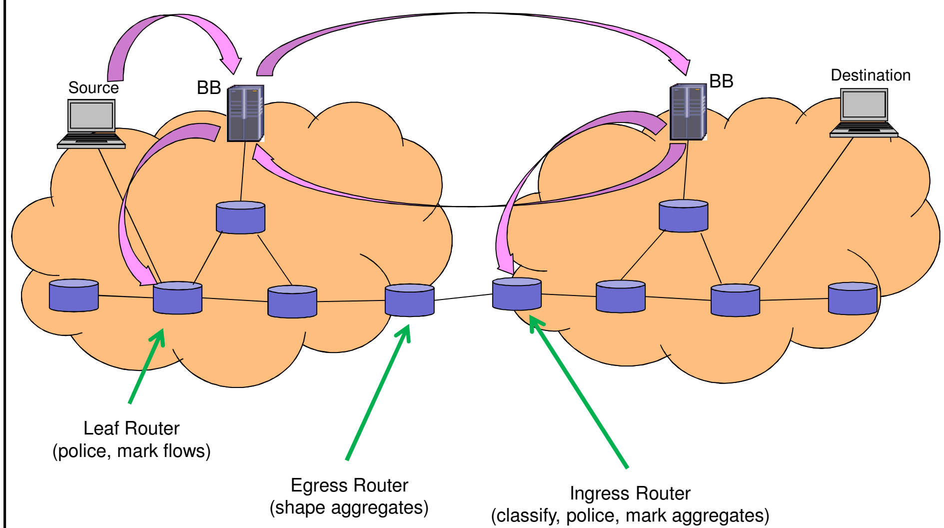
Bandwidth Broker - BB

- Bandwidth Broker (BB) is an agent that has some knowledge of an organization's priorities and policies and allocates QoS resources with respect to those policies.
- In order to achieve an end-to-end allocation of resources across separate domains, the BB managing a domain will have to communicate with its adjacent peers, which allows end-to-end services to be constructed out of purely bilateral agreements.
- Admission control (CAC) is one of the main tasks that a Bandwidth Broker has to perform, in order to decide whether an incoming resource reservation request will be accepted or not.
  - The BB acts as a Policy Decision Point (PDP) in deciding whether to allow or reject a flow, whilst the edge routers acts as Policy Enforcement Points (PEPs) to police traffic (allowing and marking packets, or simply dropping them).
- Bandwidth Brokers can be configured with organizational policies, keep track of the current allocation of marked traffic, and interpret new requests to mark traffic in light of the policies and current allocation.

62

Leon Bruckman

DiffServ Architecture

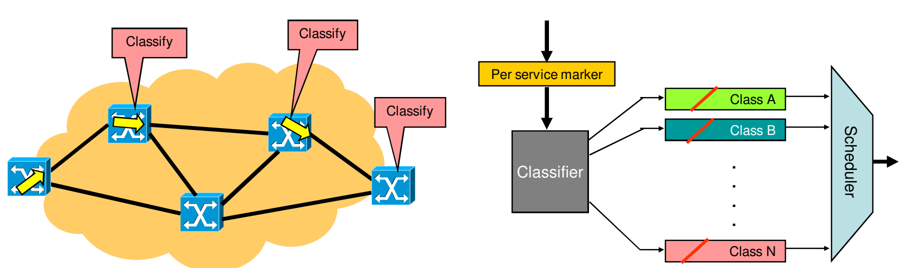


63

Leon Bruckman

Aggregated Classes of Service (CoS)

- No specific resources reservation per flow
  - Stateless protocol
  - Example: DiffServ
- Aggregated buffers for each CoS
  - Limited resource scheme
- Advantage: Scalability and simplicity
- Disadvantage: No per service deterministic behavior, based on statistics

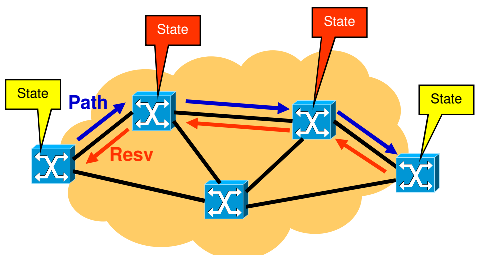


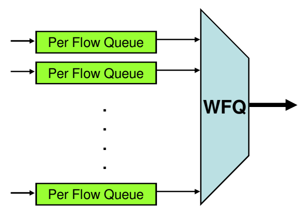
64

Leon Bruckman

Per flow QoS

- Reserve special resources per flow through the network
  - Statefull protocol
  - Example: RSVP
- Separated buffer for each flow
  - Resource “hungry” scheme
- Advantage: Per service deterministic behavior
- Disadvantage: Scalability issues

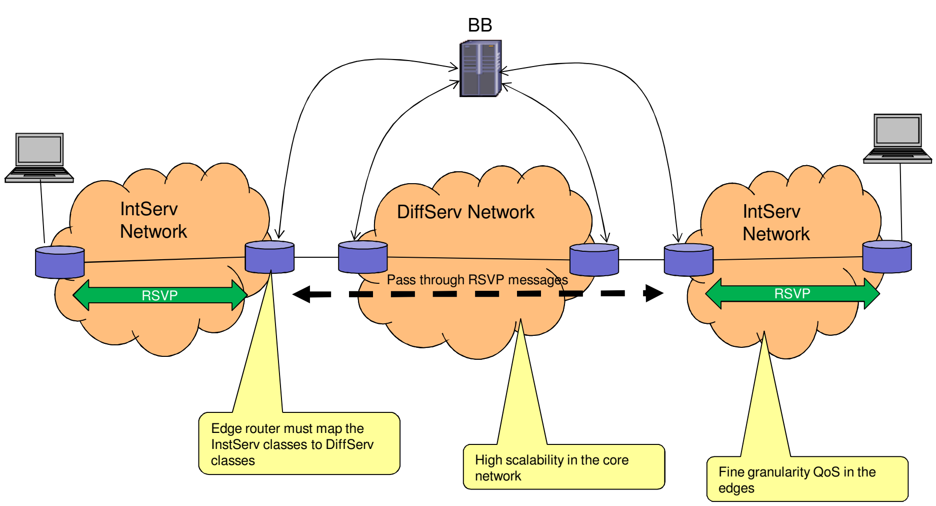




65

Leon Bruckman

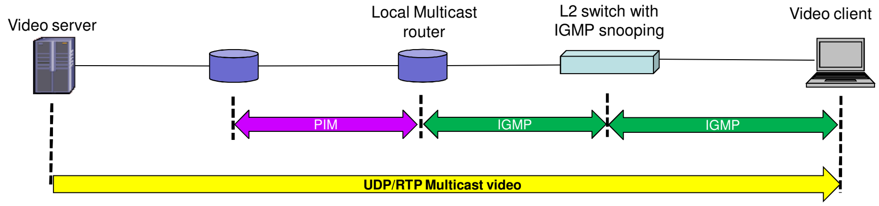
IntServ-DiffServ Interconnection



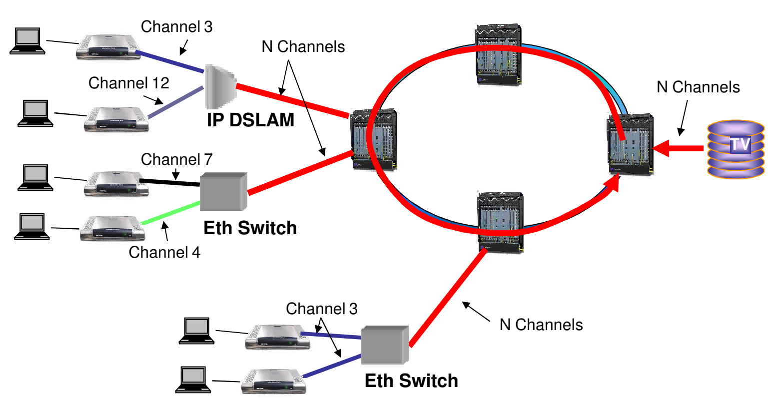
66

Internet Group Management Protocol - IGMP

- The Internet Group Management Protocol (IGMP) is a communications protocol used to manage the membership of Internet Protocol multicast groups.
  - IGMP is used by IP hosts and adjacent multicast routers to establish multicast group memberships.
- IGMP can be used for online streaming video and gaming, and allows more efficient use of resources when supporting these types of applications.
- IGMP is only needed for IPv4 networks, as multicast is handled differently in IPv6 networks.

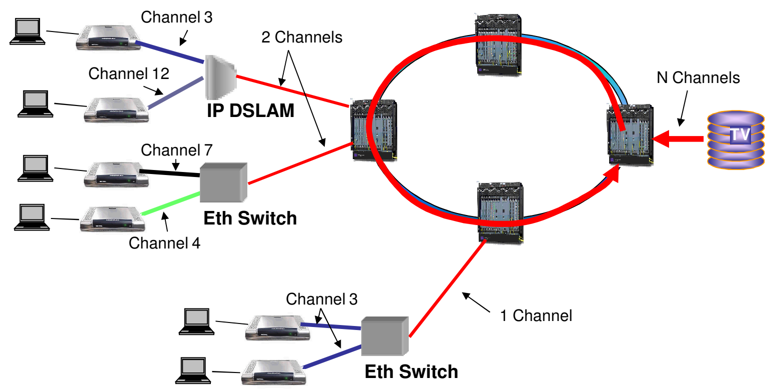


IPTV network without IGMP snooping



Leon Bruckman

IPTV network with IGMP snooping



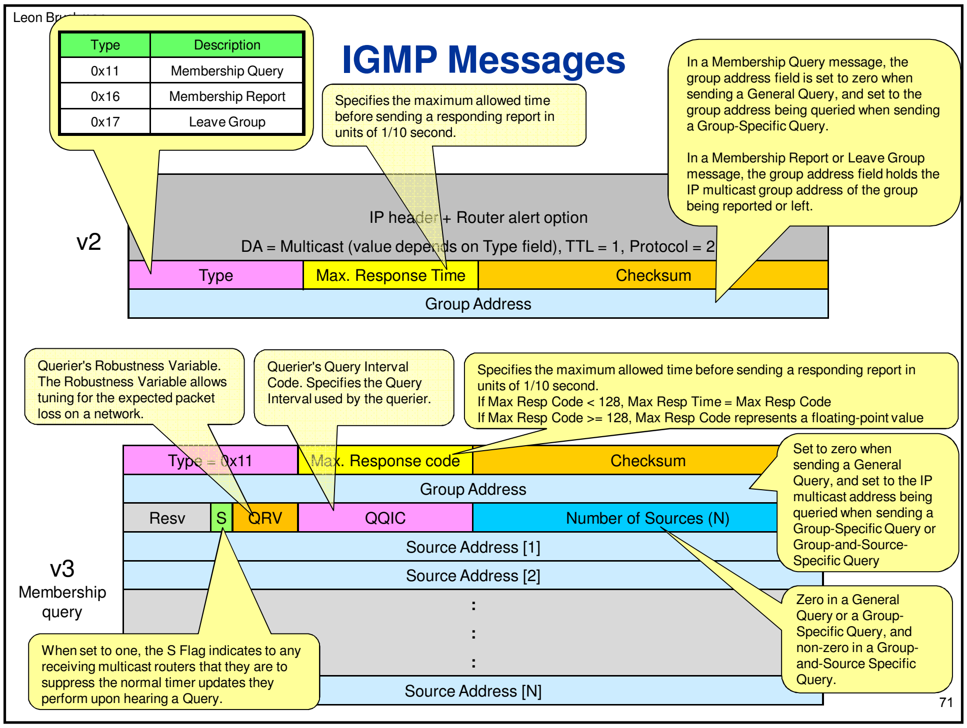
69

Leon Bruckman

IGMP versions

- There are 3 versions of IGMP
- v1
  - Hosts can join multicast groups.
  - There are no leave messages. Routers use a time-out based mechanism to discover the groups that are of no interest to the members.
- v2
  - Leave messages were added to the protocol.
  - Allow group membership termination to be quickly reported to the routing protocol, which is important for high-bandwidth multicast groups and/or subnets with highly volatile group membership.
- v3
  - It allows hosts to specify the list of sources from which they want to receive traffic from. Traffic from other sources is blocked inside the network.
  - It also allows hosts to block inside the network packets that come from sources that sent unwanted traffic.

70



Leon Bruckman

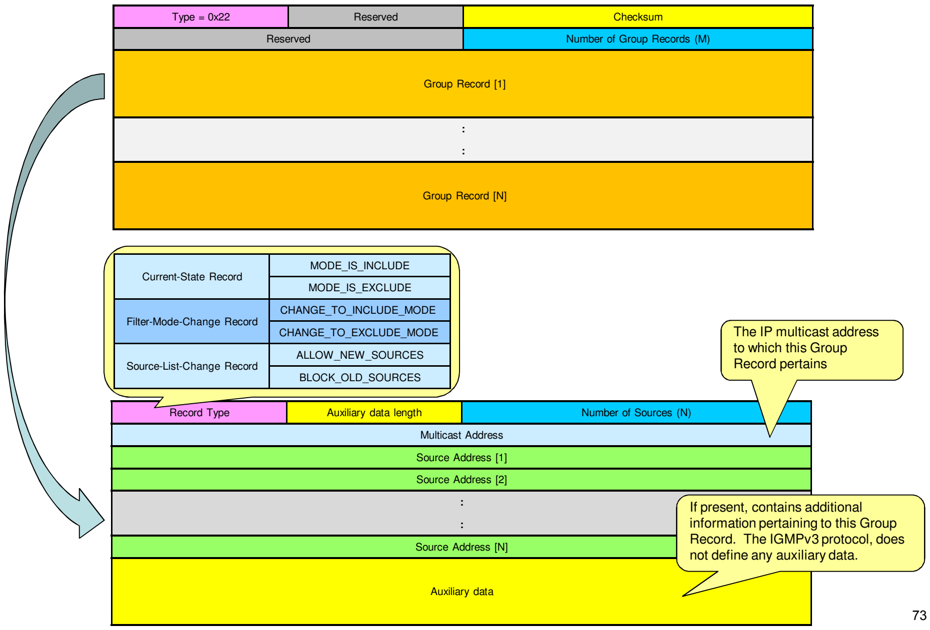
IGMPv3 Query variants

- A **General Query** is sent by a multicast router to learn the complete multicast reception state of the neighboring interfaces (that is, the interfaces attached to the network on which the Query is transmitted).
  - In a General Query, both the Group Address field and the Number of Sources (N) field are zero.
- A **Group-Specific Query** is sent by a multicast router to learn the reception state, with respect to a single multicast address, of the neighboring interfaces.
  - In a Group-Specific Query, the Group Address field contains the multicast address of interest, and the Number of Sources (N) field contains zero.
- A **Group-and-Source-Specific Query** is sent by a multicast router to learn if any neighboring interface desires reception of packets sent to a specified multicast address, from any of a specified list of sources.
  - In a Group-and-Source-Specific Query, the Group Address field contains the multicast address of interest, and theSource Address [i] fields contain the source address(es) of interest.

72

Leon Bruckman

Version 3 Membership Report Message



73

Leon Bruckman

Socket State

- For each socket on which IP Multicast Listen is desired, the system records the desired multicast reception state for that socket. That state conceptually consists of a set of records of the form:
  - (interface, multicast-address, filter-mode, source-list)
- The socket state evolves in response to each invocation of IP Multicast Listen request on the socket, as follows:

Request	New socket state
INCLUDE()	The entry corresponding to the requested interface and multicast address is deleted if present. If no such entry is present, the request is ignored.
EXCLUDE(a,b)	EXCLUDE(a,b)
INCLUDE(d,e)	INCLUDE(d,e)

74

Leon Bruckman

Interface State

- In addition to the per-socket multicast reception state, a system must maintain or compute multicast reception state for each of its interfaces. That state conceptually consists of a set of records of the form:
  - *(multicast-address, filter-mode, source-list)*
- At most one record per multicast-address exists for a given interface.
  - This per-interface state is derived from the per-socket state, but may differ from the per-socket state when different sockets have differing filter modes and/or source lists for the same multicast address and interface.
- For example, suppose one application or process invokes the following operation on socket s1:
  - *IPMulticastListen (s1, i, m, INCLUDE, {a, b, c})*
- Suppose another application or process invokes the following operation on socket s2:
  - *IPMulticastListen (s2, i, m, INCLUDE, {b, c, d})*
- The reception state of interface **i** for multicast address **m** has filter mode **INCLUDE** and **source list {a, b, c, d}**.
  - After a multicast packet has been accepted from an interface by the IP layer, its subsequent delivery to the application or process listening on a particular socket depends on the multicast reception state of that socket.

75

Leon Bruckman

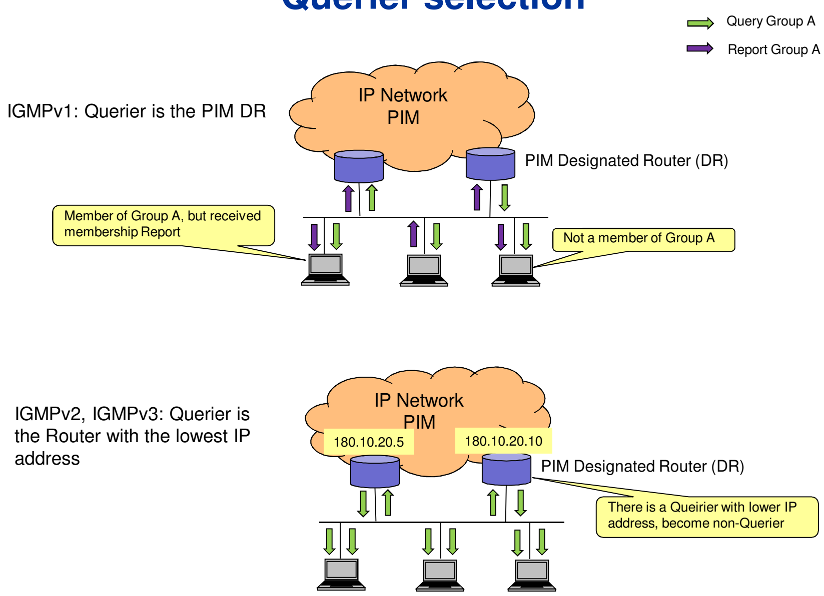
Action on Change of Interface State

- Action on Change of Interface State
  - A change of interface state causes the system to immediately transmit a State-Change Report from that interface.
  - The type and contents of the Group Record(s) in that Report are determined by comparing the filter mode and source list for the affected multicast address before and after the change, according to the table below.

Old state	New state	State-Change Record Sent
INCLUDE(a,b,c,d,e)	INCLUDE(d,e,f,g)	ALLOW(f,g) ; BLOCK(a,b,c)
EXCLUDE(a,b,c,d,e)	EXCLUDE(d,e,f,g)	ALLOW(a,b,c) ; BLOCK(f,g)
INCLUDE(a,b,c,d,e)	EXCLUDE(d,e,f,g)	TO_EXCLUDE(d,e,f,g)
EXCLUDE(a,b,c,d,e)	INCLUDE(d,e,f,g)	TO_INCLUDE(d,e,f,g)

76

Querier selection



Host basic operation

- IGMPv1
  - It sends a report when it joins a multicast group.
  - The Querier periodically sends a query messages to determine the active members of a group. Whenever a host receives a query message, it responds with report messages (one report per group) for all its associated multicast groups.
  - If the group membership is not refreshed by subsequent reports (in response to general queries), the group information is removed (Leave)
- IGMPv2
  - Joining a Group is similar to IGMPv1
  - A last host sends a leave group message when it is no longer a member of the multicast group. When a querier receives a leave group message for multicast group, it generates a group specific query to check whether there are any other member hosts for that particular group.
- IGMPv3
  - Send v3 report messages to indicate their multicast reception states while responding to queries or when they need to indicate any change in their reception states.
  - Reception state information associated with a group is placed as part of a group record (GR).

Leon Bruckman

Host Suppression

- In IGMPv1 and IGMPv2, a host would cancel sending a pending membership reports if a similar report was observed from another member on the network.
  - In IGMPv3, this suppression of host membership reports has been removed.
- The following points explain the reasons behind this decision:
  - Routers may want to track per-host membership status on an interface
    - Fast leave, accounting
  - Membership Report suppression does not work well on bridged LANs.
    - Many bridges and Layer2/Layer3 switches that implement IGMP snooping do not forward IGMP membership report messages across LAN segments in order to prevent membership report suppression.
    - Removing membership report suppression eases the job of these IGMP snooping devices.
  - By eliminating membership report suppression, hosts have fewer messages to process; this leads to a simpler state machine implementation.
  - In IGMPv3, a single membership report now bundles multiple multicast group records to decrease the number of packets sent.
    - In comparison, the previous versions of IGMP required that each multicast group be reported in a separate message.

79

Leon Bruckman

Interoperability between IGMP versions

- IGMP version 3 hosts and routers interoperate with hosts and routers that have not yet been upgraded to IGMPv3.
  - This compatibility is maintained by hosts and routers taking appropriate actions depending on the versions of IGMP operating on hosts and routers within a network.
- Query Version Distinctions
  - The IGMP version of a Membership Query message is determined as follows:
    - IGMPv1 Query: length = 8 octets AND Max Resp Code field is zero
    - IGMPv2 Query: length = 8 octets AND Max Resp Code field is non-zero
    - IGMPv3 Query: length >= 12 octets
- Message translation
  - IGMPv1 and IGMPv2 Report → IGMPv3 Group Record Mode Is EXCLUDE()
  - IGMPv2 Leave → IGMPv3 Group Record Change to INCLUDE()

80

Source Specific Multicast

- The Source Specific Multicast (SSM) feature is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources to which the receivers have explicitly joined.
  - For multicast groups configured for SSM, only source-specific multicast distribution trees (no shared trees) are created.
  - The 232/8 IPv4 address range is currently allocated for SSM
  - In IPv6, the FF3x::/32 range (where 'x' is a valid IPv6 multicast scope value) is reserved for SSM semantics although today SSM allocations are restricted to FF3x::/96.
- The benefits of source-specific multicast include:
  - Elimination of cross-delivery of traffic when two sources simultaneously use the same source-specific destination address. The simultaneous use of an SSM destination address by multiple sources and different applications is explicitly supported.
  - Avoidance of the need for inter-host coordination when choosing source-specific addresses, as a consequence of the above.
- SSM is particularly well-suited to dissemination-style applications with one or more senders whose identities are known before the application begins.
  - For instance, a data dissemination application that desires to provide a secondary data source in case the primary source fails over might implement this by using one channel for each source and advertising both of them to receivers.

SSM aware operation

- An SSM-aware host does not send, and SSM-aware routers ignore, any of the following record types for an SSM address.
  - MODE_IS_EXCLUDE as part of a Current-State Record
  - CHANGE_TO_EXCLUDE_MODE as part of a Filter-Mode-Change Record
- A router never generates an IGMPv1, IGMPv2 query for an address in the SSM range.
- It is important that a router does not accept non-source-specific reception requests for an SSM destination address.
  - The rules of IGMPv3 require a router, upon receiving such a membership report, to revert to earlier version compatibility mode for the group in question.
  - If the router were to revert in this situation, it would prevent an IGMPv3-capable host from receiving SSM service for that destination address, thus creating a potential for an attacker to deny SSM service to other hosts on the same link.

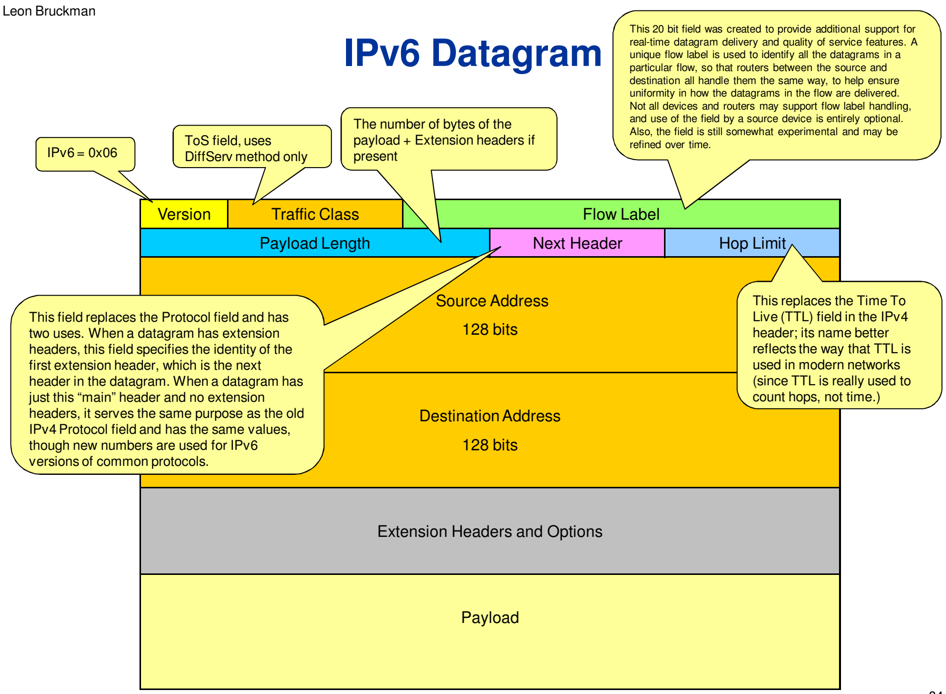
Leon Bruckman

IPv6 Design Goals

- **Larger Address Space**: IPv6 had to provide more addresses for the growing Internet.
- **Better Management of Address Space**: IPv6 not only includes more addresses, but also a more capable way of dividing the address space.
- **Easier TCP/IP Administration**: Resolve some of the current labor-intensive requirements of IPv4, such as the need to configure IP addresses.
- **Modern Design For Routing:** IPv6 was created specifically for efficient routing in our current Internet, and with the flexibility for the future.
- **Better Support For Multicasting:** Multicasting was an option under IPv4 from the start, but support for it has been slow in coming.
- **Better Support For Security:** Today, security on the public Internet is a big issue, and the future success of the Internet requires that security concerns be resolved.
- **Better Support For Mobility:** IPv6 builds on Mobile IP and provides mobility support within IP itself.

Leon Bruckman

IPv6 Datagram



84

Leon Bruckman

Flow Label

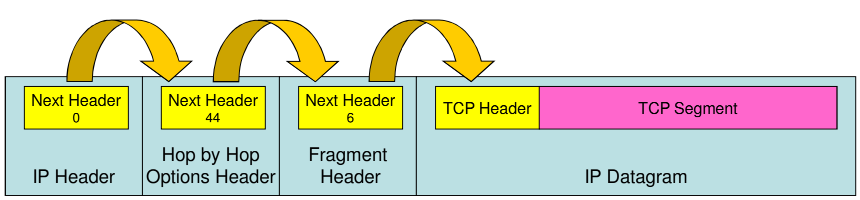
- Traditionally, flow classifiers have been based on the **5-tuple** of the source and destination addresses, ports, and the transport protocol type.
  - However, some of these fields may be unavailable due to either fragmentation or encryption, or locating them past a chain of IPv6 extension headers may be inefficient.
  - Additionally, if classifiers depend only on IP layer headers, later introduction of alternative transport layer protocols will be easier.
- The usage of the **3-tuple** of the **Flow Label** and the **Source** and **Destination** Address fields enables efficient IPv6 flow classification, where only IPv6 main header fields in fixed positions are used.
- The minimum level of IPv6 flow support consists of labeling the flows.
  - A specific goal is to enable and encourage the use of the flow label for various forms of stateless load distribution, especially across Equal Cost Multi-Path (EMCP) and/or Link Aggregation Group (LAG) paths.
- A Flow Label of zero is used to indicate packets that have not been labeled.

85

Leon Bruckman

Extension Headers

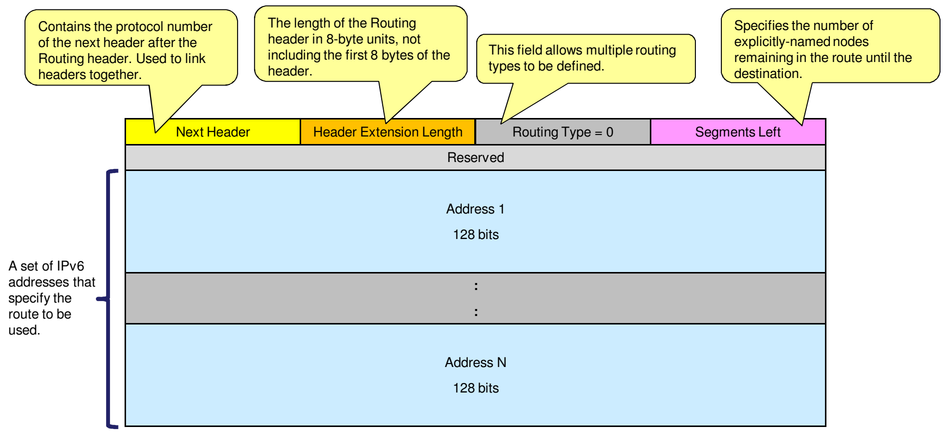
- After the mandatory "main" header in an IPv6 datagram, one or more extension headers may appear before the encapsulated payload.
- These headers were created in an attempt to provide both flexibility and efficiency in the creation of IPv6 datagrams.
- All fields that are needed only for special purposes are put into extension headers and placed in the datagram when needed.
  - This allows the size of the main datagram header to be made small and streamlined, containing only those fields that really must be present all the time.
- There is one complication hidden in the header chaining mechanism: the processing of complete headers may require a walk through quite a long chain of extension headers which hinders the processing performance.
  - To minimize this, IPv6 specifies a particular order of extension headers. Generally speaking, headers important for all forwarding nodes must be placed first, headers important just for the addressee are located on the end of the chain.



86

Leon Bruckman

IPv6 Routing Extension Header Format – Type 0

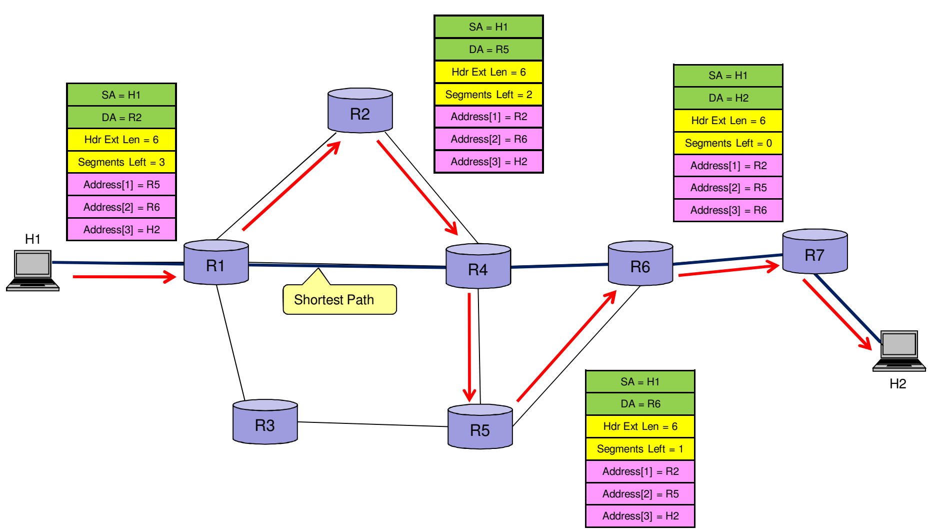


- A single Routing Extension Header type 0 may contain multiple intermediate node addresses, and the same address may be included more than once.
  - This allows a packet to be constructed such that it will oscillate between two Routing Extension Header type 0 processing hosts or routers many times.
  - This allows a stream of packets from an attacker to be amplified along the path between two remote routers, which could be used to cause congestion along arbitrary remote paths and hence act as a denial-of-service mechanism.
- *The severity of this threat is considered to be sufficient to warrant deprecation of Routing Extension Header type 0 entirely.*

87

Leon Bruckman

Source routing example

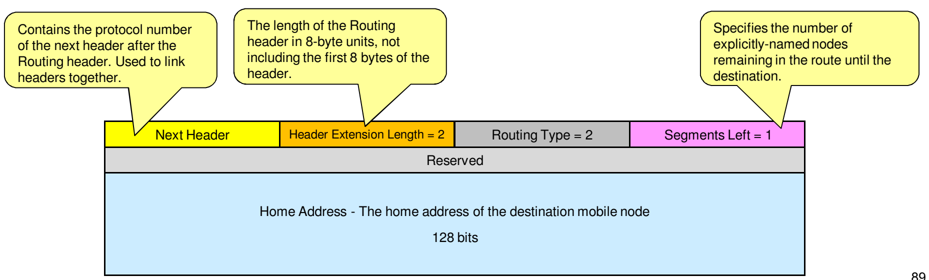


88

Leon Bruckman

IPv6 Routing Extension Header Format – Type 2

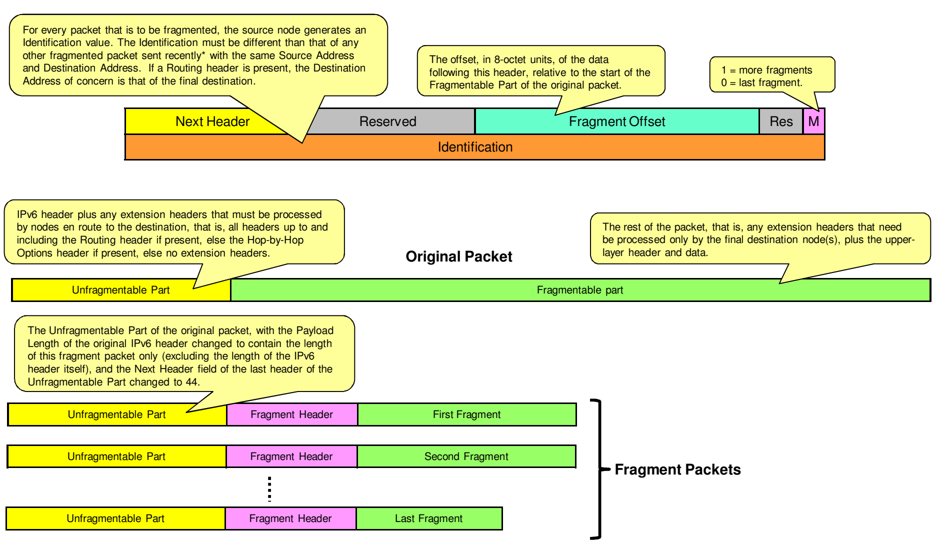
- Mobile IPv6 defines a new routing header variant, the type 2 routing header, to allow the packet to be routed directly from a correspondent to the mobile node's **care-of address**.
- The mobile node's care-of address is inserted into the IPv6 Destination Address field.
  - Once the packet arrives at the care-of address, the mobile node retrieves its home address from the routing header, and this is used as the final destination address for the packet.
- **Home address:** A unicast routable address assigned to a mobile node, used as the permanent address of the mobile node. This address is within the mobile node's home link.
- **Care-of address**: A unicast routable address associated with a mobile node while visiting a foreign link; the subnet prefix of this IP address is a foreign subnet prefix.
- The inclusion of **home addresses** makes the use of the **care-of address** transparent above the network layer (e.g., TCP "sees" the home address).



89

Leon Bruckman

Fragmentation



90

Fragmentation rules

- Used by an IPv6 source to send a packet larger than would fit in the path MTU to its destination.
  - In IPv4 fragmentation may be performed also by routers along a packet's delivery path
- An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address, and Fragment Identification.
- The Unfragmentable Part of the reassembled packet consists of all headers up to, but not including, the Fragment header of the first fragment packet (that is, the packet whose Fragment Offset is zero), with the following two changes:
  - The Next Header field of the last header of the Unfragmentable Part is obtained from the Next Header field of the first fragment's Fragment header.
  - The Payload Length of the reassembled packet is computed from the length of the Unfragmentable Part and the length and offset of the last fragment.
    - See RFC 2460 for the formula for computing the Payload Length of the reassembled original packet
- The Fragment header is not present in the final, reassembled packet.

More Extension headers

- **Hop-by-Hop Options Header**
  - Used to carry optional information that must be examined by every node along a packet's delivery path.
  - This is the only extension header examined and processed by every node along a packet's delivery path, including the source and destination nodes.
    - It must immediately follow the IPv6 header.
  - Examples: Header padding, Router Alert
- **Destination Options Header**
  - Used to carry optional information that need be examined only by a packet's destination node(s).
  - Example: Header padding,

Leon Bruckman

Extension Header Order

- When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order:
  - IPv6 header
  - Hop-by-Hop Options header
  - Destination Options header
    - For options to be processed by the first destination that appears in the IPv6 Destination Address field plus subsequent destinations listed in the Routing header.
  - Routing header
  - Fragment header
  - Authentication header
  - Encapsulating Security Payload header
  - Destination Options header
    - For options to be processed only by the final destination of the packet.
  - Upper-layer header
- Nevertheless, IPv6 nodes must accept and attempt to process extension headers in any order and occurring any number of times in the same packet, except for the Hop-by-Hop Options header which is restricted to appear immediately after an IPv6 header only.

93

Leon Bruckman

IPv6 Address Notations

IPv6 Address: 128.91.45.157.220.40.0.0.0.0.252.87.212.200.31.255

Hex	805B	2D9D	D728	0000	0000	FC57	DAC8		1FFF	
Leading zeros suppressed	805B	2D9D	D728	0	0	FC57	DAC8		1FFF	
Zero compressed	805B	2D9D	D728	:	:	FC57	DAC8		1FFF	
Mixed notation	805B	2D9D	D728	:	:	FC57	212	200	31	255

Zero Compressed example
CIDR notation

FF00:4501:0:0:0:0:0:32/56

FF00:4501::32/56

IPv4 address 212.200.31.255

IPv4 mapped IPv6 Address 0:0:0:0:0:F:212.200.31.255

IPv6 Address Mixed Notation ::F:212.200.31.255

94

Leon Bruckman

Too many IPv6 address representations

- A single IPv6 address can be text represented in many ways. Examples are shown below.
  - 2001:db8:0:0:1:0:0:1
  - 2001:**0db8**:0:0:1:0:0:1
  - 2001:db8:**:**1:0:0:1
  - 2001:db8:**:**0:1:0:0:1
  - 2001:**0db8::**1:0:0:1
  - 2001:db8:0:0:1:**:**1
  - 2001:db8:**0000**:0:1:**:**1
  - 2001:**DB8**:0:0:1:**:**1
- This flexibility has caused many problems for operators, systems engineers, and customers.

95

Leon Bruckman

RFC 5952 versus RFC 4291

Requirement	RFC 4291	RFC 5952
Leading Zeros in a 16-Bit Field	It is not necessary to write the leading zeros in an individual field.	Leading zeros must be suppressed.
Zero Compression	A special syntax is available to compress the zeros. The use of "::" indicates one or more groups 16 bits of zeros.	The use of the symbol "::" must be used to its maximum capability.
		The symbol "::" must not be used to shorten just one 16-bit 0 field.
		When there is an alternative choice in the placement of a "::", the longest run of consecutive 16-bit 0 fields must be shortened (i.e., the sequence with three consecutive zero fields is shortened in 2001: 0:0:1:0:0:0:1). When the length of the consecutive 16-bit 0 fields are equal (i.e., 2001:db8:0:0:1:0:0:1), the first sequence of zero bits must be shortened.
Uppercase or Lowercase	No mention	The characters "a", "b", "c", "d", "e", and "f" in an IPv6 address must be represented in lowercase.

- 2001:db8:0:0:1:0:0:1
- 2001:**0db8**:0:0:1:0:0:1
- 2001:db8:**:**1:0:0:1
- 2001:db8:**:**0:1:0:0:1
- 2001:**0db8::**1:0:0:1
- 2001:db8:0:0:1:**:**1
- 2001:db8:**0000**:0:1:**:**1
- 2001:**DB8**:0:0:1:**:**1

2001:db8::1:0:0:1

96

Leon Bruckman

IPv6 Special Unicast Addresses

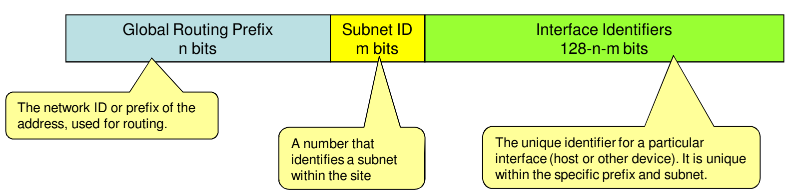
- **Unspecified Address**: 0:0:0:0:0:0:0:0 (::/128). It indicates the absence of an address.
  - One example of its use is in the Source Address field of any IPv6 packets sent by an initializing host before it has learned its own address.
  - The unspecified address must not be used as the destination address of IPv6 packets or in IPv6 Routing headers.
  - An IPv6 packet with a source address of unspecified must never be forwarded by an IPv6 router.
- **Loopback Address**: 0:0:0:0:0:0:0:1 (::1/128). It may be used by a node to send an IPv6 packet to itself and it must not be assigned to any physical interface.
  - The loopback address must not be used as the source address in IPv6 packets that are sent outside of a single node.
  - An IPv6 packet with a destination address of loopback must never be sent outside of a single node and must never be forwarded by an IPv6 router.
  - A packet received on an interface with a destination address of loopback must be dropped.

97

Leon Bruckman

IPv6 Global Unicast Address format

- **Global unicast addresses**: All Global Unicast addresses other than those that start with binary 000 have a 64-bit interface ID field (i.e., n + m = 64)
  - Global Unicast addresses that start with binary 000 have no such constraint on the size or structure of the interface ID field.
  - Examples of Global Unicast addresses that start with binary 000 are the IPv4 mapped IPv6 addresses



- Global Unicast Addresses for IPv6 are assigned by the Internet Assigned Numbers Authority (IANA) and fall within the IPv6 prefix 2000::/3.

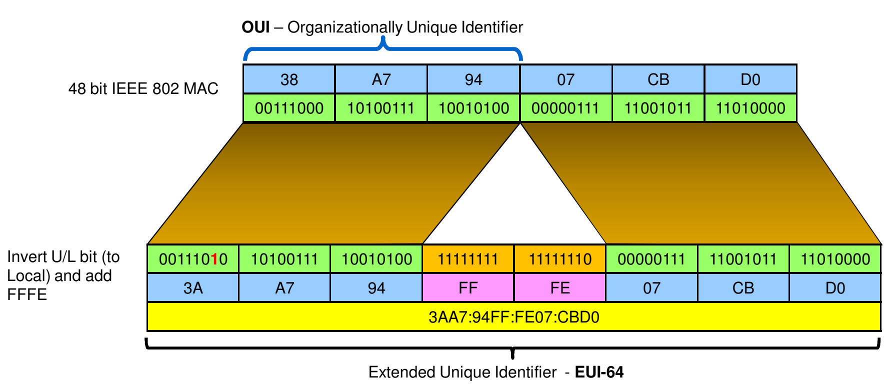
98

Local addresses

- Site-Local Addresses
  - These addresses have the scope of an entire site, or organization. They allow addressing within an organization without need for using a public prefix.
  - Routers will forward datagrams using site-local addresses within the site, but not outside it to the public Internet.
  - First nine bits: 1111 1110 11xx – FEC, FEE, FED, FEF
  - *Deprecated, new implementations must treat this prefix as Global Unicast*
- Link-Local Addresses
  - These addresses refer only to a particular physical link (physical network).
  - Routers will not forward datagrams using link-local addresses at all, not even within the organization; they are only for local communication on a particular physical network segment.
  - First nine bits: 1111 1110 10xx – FE8, FE9, FEA, FEB

Physical Address Mapping

- Instead of using arbitrary “made-up” identifiers for hosts, we can base the interface ID on the underlying data link layer hardware address, as long as that address is no greater than 64 bits in length.
  - Since virtually all devices use layer two addresses of 64 bits or fewer, there is no problem in using those addresses for the interface identifier in IP addresses.
- The IP address can be derived from the MAC address and the network identifier.
  - It also means we can in the future tell the IP address from the MAC address and vice-versa



Leon Bruckman

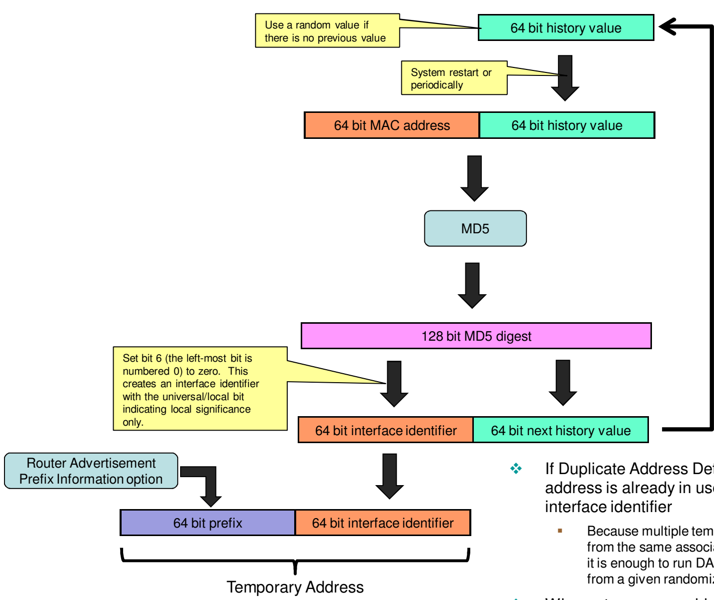
The Concern With IPv6 Addresses

- The division of IPv6 addresses into distinct topology and interface identifier portions raises an issue new to IPv6 in that a fixed portion of an IPv6 address (i.e., the interface identifier) can contain an identifier that remains constant even when the topology portion of an address changes (e.g., as the result of connecting to a different part of the Internet).
  - This is of particular concern with the expected proliferation of next-generation network-connected devices (e.g., PDAs, cell phones, etc.) in which large numbers of devices are in practice associated with individual users (i.e., not shared).
  - Thus, the interface identifier embedded within an address could be used to track activities of an individual, even as they move topologically within the internet.
  - In IPv4, when an address changes, the entire address (including the local part of the address) usually changes. It is this new issue that this document addresses.
- Many machines function as both **clients** and **servers**.
  - In such cases, the machine would need a DNS name for its use as a **server**.
    - Whether the address stays fixed or changes has little privacy implication since the DNS name remains constant and serves as a constant identifier.
  - When acting as a **client** (e.g., initiating communication), however, such a machine may want to vary the addresses it uses.
  - In such environments, one may need multiple addresses: a "public" (i.e., non-secret) **server** address, registered in the DNS, that is used to accept incoming connection requests from other machines, and a "temporary" address used to shield the identity of the **client** when it initiates communication.
    - These two cases are roughly analogous to telephone numbers and caller ID, where a user may list their telephone number in the public phone book, but disable the display of its number via caller ID when initiating calls.

101

Leon Bruckman

Generating Temporary Addresses

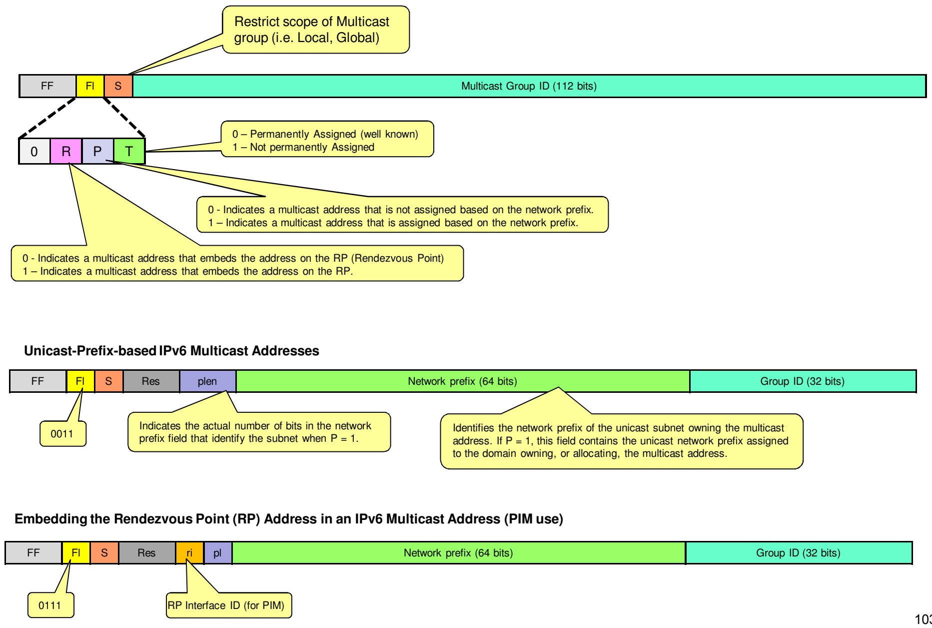


- If Duplicate Address Detection (DAD) indicates the address is already in use, generate a new randomized interface identifier
  - Because multiple temporary addresses are generated from the same associated randomized interface identifier, it is enough to run DAD on the first address generated from a given randomized identifier.
- When a temporary address becomes deprecated (lifetime expiration), a new one should be generated.

102

Leon Bruckman

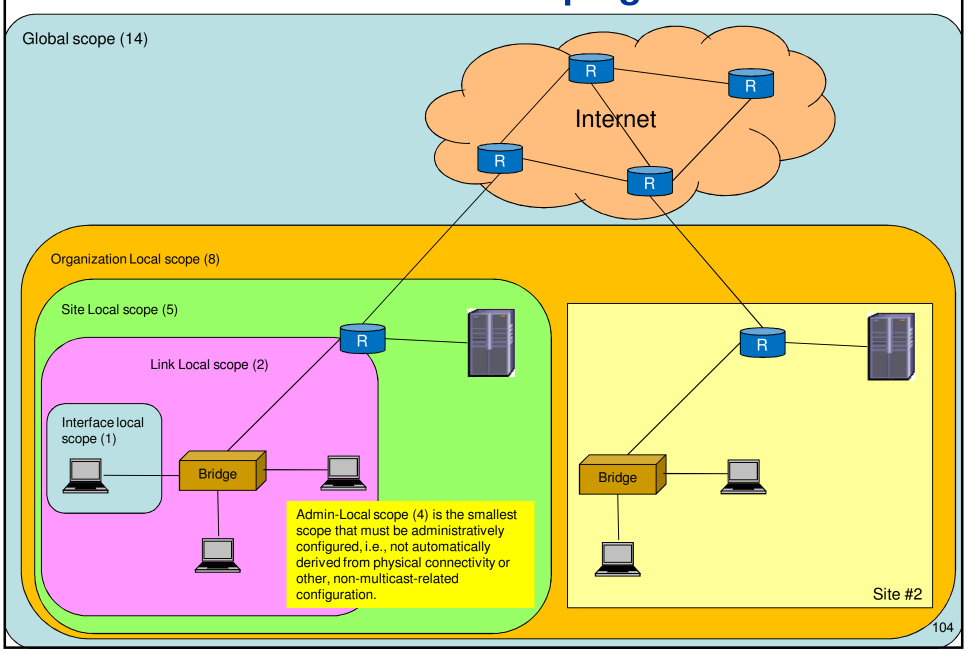
IPv6 Multicast Address Formats



103

Leon Bruckman

Multicast scoping



104

Leon Bruckman

Multicast addresses details

- The "meaning" of a permanently-assigned multicast address is independent of the scope value.
  - For example, if the "NTP servers group" is assigned a permanent multicast address with a group ID of 101 (hex), then:
    - FF01:0:0:0:0:0:0:101 means all NTP servers on the same interface (i.e., the same node) as the sender (Loopback multicast).
    - FF02:0:0:0:0:0:0:101 means all NTP servers on the same link as the sender.
    - FF05:0:0:0:0:0:0:101 means all NTP servers in the same site as the sender.
    - FF0E:0:0:0:0:0:0:101 means all NTP servers in the Internet.
- Non-permanently-assigned multicast addresses are meaningful only within a given scope.
  - For example, a group identified by the non- permanent, site-local multicast address FF15:0:0:0:0:0:0:101 at one site bears no relationship to a group using the same address at a different site, nor to a non-permanent group using the same group ID with a different scope, nor to a permanent group with the same group ID.
- Multicast addresses must not be used as source addresses in IPv6 packets.
- Routers must not forward any multicast packets beyond of the scope indicated by the scop field in the destination multicast address.
- Nodes should not originate a packet to a multicast address whose scop field contains the reserved value F; if such a packet is sent or received, it must be treated the same as packets destined to a global (scop E) multicast address.

Leon Bruckman

Pre-Defined Multicast Addresses

- **All Nodes Addresses**: FF01:0:0:0:0:0:0:1, FF02:0:0:0:0:0:0:1
  - Identify the group of all IPv6 nodes, within scope 1 (interface-local) or 2 (link-local).
- **All Routers Addresses**: FF01:0:0:0:0:0:0:2, FF02:0:0:0:0:0:0:2, FF05:0:0:0:0:0:0:2
  - The above multicast addresses identify the group of all IPv6 routers, within scope 1 (interface-local), 2 (link-local), or 5 (site-local).
- **Solicited-Node Address**: FF02:0:0:0:0:1:FFXX:XXXX
  - Solicited-Node multicast address are computed as a function of a node's unicast and anycast addresses.
  - A Solicited-Node multicast address is formed by taking the low-order 24 bits of an address (unicast or anycast) and appending those bits to the prefix FF02:0:0:0:0:1:FF00::/104 resulting in a multicast address in the range FF02:0:0:0:0:1:FF00:0000 to FF02:0:0:0:0:1:FFFF:FFFF
  - For example, the Solicited-Node multicast address corresponding to the IPv6 address 4037::01:800:200E:8C6C is FF02::1:FF0E:8C6C.
  - A node is required to compute and join (on the appropriate interface) the associated Solicited-Node multicast addresses for all unicast and anycast addresses that have been configured for the node's interfaces (manually or automatically).

Leon Bruckman

Anycast addresses

- Anycast addresses can be considered a conceptual cross between unicast and multicast addressing.
- Where unicast says "*send to this one address*" and multicast says "*send to every member of this group*", anycast says "***send to any one member of this group***".
- Naturally, in choosing which member to send to, we would for efficiency reasons normally send to the closest one—closest in routing terms. So we can normally also consider anycast to mean "**send to the closest member of this group**".
- Anycast was specifically intended to provide flexibility in situations where we need a service that is provided by a number of different servers or routers but don't really care which one provides it.
  - Datagrams sent to the anycast address will automatically be delivered to the device that is easiest to reach
- There is no special anycast addressing scheme: anycast addresses are the same as unicast addresses.
  - When a unicast address is assigned to more than one interface, thus turning it into an anycast address, the nodes to which the address is assigned must be explicitly configured to know that it is an anycast address.

107

Leon Bruckman

Routing Anycast addresses

- For any assigned anycast address, there is a longest prefix P of that address that identifies the topological region in which all interfaces belonging to that anycast address reside.
  - Within the region identified by P, the anycast address must be maintained as a separate entry in the routing system (commonly referred to as a "host route"); outside the region identified by P, the anycast address may be aggregated into the routing entry for prefix P.
  - Note that in the worst case, the prefix P of an anycast set may be the null prefix, i.e., the members of the set may have no topological locality.
    - In that case, the anycast address must be maintained as a separate routing entry throughout the entire Internet, which presents a severe scaling limit on how many such "global" anycast sets may be supported.
    - Therefore, it is expected that support for global anycast sets may be unavailable or very restricted.
- The Subnet-Router anycast address is predefined. Its format is as follows:

Subnet Prefix (n bits)	0000000000000000 (128-n bits)

- Packets sent to the Subnet-Router anycast address will be delivered to one router on the subnet.
  - All routers are required to support the Subnet-Router anycast addresses for the subnets to which they have interfaces.
  - The Subnet-Router anycast address is intended to be used for applications where a node needs to communicate with any one of the set of routers.

108

Leon Bruckman

Anycast considerations

- Anycast address can not be put into IPv6 source address.
  - This is basically because an IPv6 anycast address does not identify a single source node.
    - Incorrect reassembly of fragmented packets due to multiple anycast members sending packets with the same fragment ID to the same destination at about the same time.
    - Errors and other response packets might be delivered to a different anycast member than sent the packet. This might be very likely since asymmetric routing is rather prevalent on the Internet.
- Nondeterministic packet delivery
  - If multiple packets carry an anycast address in IPv6 destination address header, these packets may not reach the same destination node, depending on stability of the routing table.
  - An anycast client needs to make sure that its request fits in a single packet.
    - For any statefull communication with an anycast server, the client uses the responding server's unicast address.
    - Future stateless anycast service requests, however, can be sent to the anycast address.

109

Leon Bruckman

IPv6 Datagram Delivery and Routing

- Most of the concepts related to how datagram delivery is accomplished in IPv6 are the same as in IPv4.
- Changes in Datagram Delivery and Routing in IPv6:
  - **Hierarchical Routing and Aggregation:** One of the goals of the structure used for organizing unicast addresses was to improve routing. The unicast addressing format is designed to provide a better match between addresses and Internet topology, and to facilitate route aggregation.
  - **Scoped Local Addresses:** Local-use addresses including site-local and link-local are defined in IPv6.
  - **Multicast and Anycast Routing:** Multicast is standard in IPv6, not optional as in IPv4. Anycast addressing is a new type of addressing in IPv6.
  - **More Support Functions:** Capabilities must be added to routers to support new features in IPv6. For example, routers play a key role in implementing serverless autoconfiguration and path MTU discovery in the new IPv6 fragmentation scheme.
  - **New Routing Protocols:** Routing protocols such as RIP must be updated to support IPv6.
  - **Multiple addresses per interface:** Easier networks merging (use both addresses during companies consolidation) and addressing scheme change.

110

Leon Bruckman

Transition from IPv4 to IPv6

- **"Dual Stack" Devices**:
  - Routers and some other devices may be programmed with both IPv4 and IPv6 implementations to allow them to communicate with both types of hosts.
- **IPv4/IPv6 Translation:**
  - "Dual stack" devices may be designed to accept requests from IPv6 hosts, convert them to IPv4 datagrams, send the datagrams to the IPv4 destination and then process the return datagrams similarly.
- **IPv4 Tunneling of IPv6:**
  - IPv6 devices that don't have a path between them consisting entirely of IPv6-capable routers may be able to communicate by encapsulating IPv6 datagrams within IPv4. In essence, they would be using IPv6 on top of IPv4; two network layers. The encapsulated IPv4 datagrams would travel across conventional IPv4 routers.

Leon Bruckman

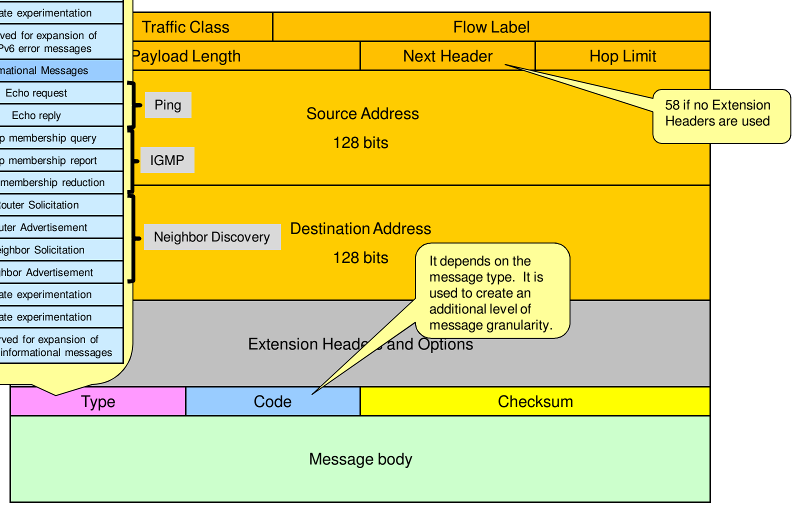
Internet Control Message Protocol - ICMPv6

- ICMPv6 is used by IPv6 nodes to report errors encountered in processing packets, and to perform other internet-layer functions, such as diagnostics (ICMPv6 "ping").
  - The Internet Protocol version 6 (IPv6) uses the Internet Control Message Protocol (ICMP) as defined for IPv4, with a number of changes.
- ICMPv6 is an integral part of IPv6, and the base protocol must be fully implemented by every IPv6 node.
  - On top of the basic functions it performs neighbor discovery, and a framework for extensions to implement future Internet Protocol control aspects.
- ICMPv6 offers a comprehensive solution by offering the different functions earlier subdivided among the different protocols such as ICMP, ARP, and IGMP
  - It further simplifies the communication process by eliminating obsolete messages.

Leon Bruckman

ICMPv6 Datagram

ICMPv6 Error Messages	
1	Destination unreachable
2	Packet too big
3	Time exceeded
4	Parameter problem
101	Private experimentation
102	Private experimentation
127	Reserved for expansion of ICMPv6 error messages
ICMPv6 Informational Messages	
128	Echo request
129	Echo reply
130	Group membership query
131	Group membership report
132	Group membership reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
200	Private experimentation
201	Private experimentation
255	Reserved for expansion of ICMPv6 informational messages



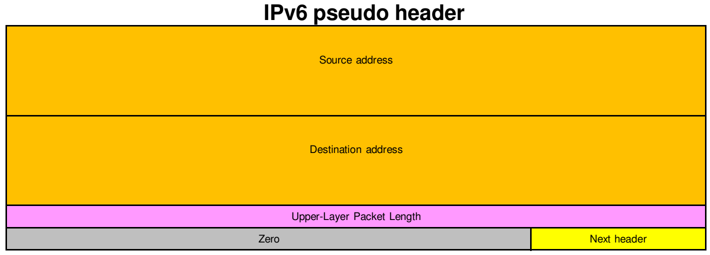
113

Leon Bruckman

Checksum calculation

- The checksum is the 16-bit one's complement of the one's complement sum of the entire ICMPv6 message, starting with the ICMPv6 message type field, and prepended with a "pseudo-header" of IPv6 header fields.
  - The reason for the change is to protect ICMP from misdelivery or corruption of those fields of the IPv6 header on which it depends, which, unlike IPv4, are not covered by an internet-layer checksum.
  - The Next Header field in the pseudo-header for ICMP contains the value 58, which identifies the IPv6 version of ICMP.

IPv6 pseudo header



114

Leon Bruckman

ICMPv6 message processing rules

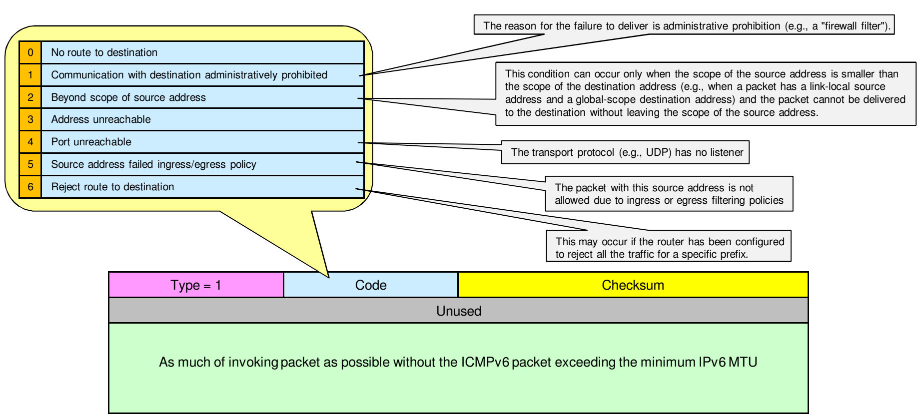
- If an ICMPv6 error message of unknown type is received at its destination, it must be passed to the upper-layer process that originated the packet that caused the error, where this can be identified.
- If an ICMPv6 informational message of unknown type is received, it must be silently discarded.
- Every ICMPv6 error message must include as much of the IPv6 offending packet (the packet that caused the error) as possible without making the error message packet exceed the minimum IPv6 MTU (1280 bytes).
- An error message must not be originated as a result of receiving:
  - An ICMPv6 error message
  - An ICMPv6 redirect message
  - A packet destined to an IPv6 multicast address (some exceptions apply)
  - A packet whose source address does not uniquely identify a single node
- In order to limit the bandwidth and forwarding costs incurred by originating ICMPv6 error messages, an IPv6 node must limit the rate of ICMPv6 error messages it originates.

115

Leon Bruckman

Example: Destination Unreachable Message

- A Destination Unreachable message is generated by a router, or by the IPv6 layer in the originating node, in response to a packet that cannot be delivered to its destination address for reasons **other than congestion**.



116

Leon Bruckman

Example: Packet Too Big Message

- Packet Too Big is sent by a router in response to a packet that it cannot forward because the packet is larger than the MTU of the outgoing link.
  - The information in this message is used as part of the Path MTU Discovery process.
- Originating a Packet Too Big Message makes an exception to one of the rules as to when to originate an ICMPv6 error message.
  - Unlike other messages, it is sent in response to a packet received with an IPv6 multicast destination address, or with a link-layer multicast or link-layer broadcast address.
  - This allows Path MTU discovery to work for IPv6 multicast

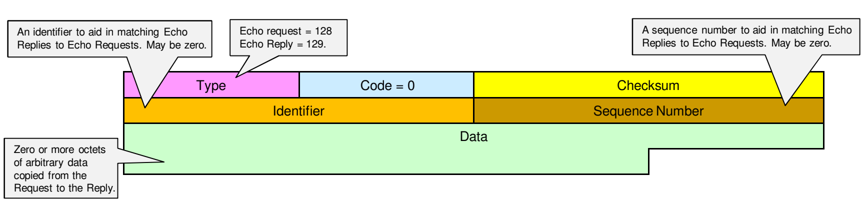
Type = 2	Code = 0	Checksum
MTU of the next-hop link		
As much of invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU		

117

Leon Bruckman

Example: Echo Request/Reply

- Every node must implement an ICMPv6 Echo responder function that receives Echo Requests and originates corresponding Echo Replies.
  - A node should also implement an application-layer interface for originating Echo Requests and receiving Echo Replies, for diagnostic purposes.
- The source address of an Echo Reply sent in response to a unicast Echo Request message is the same as the destination address of that Echo Request message.
- An Echo Reply should be sent in response to an Echo Request message sent to an IPv6 multicast or anycast address.
  - In this case, the source address of the reply is a unicast address belonging to the interface on which the Echo Request message was received.



118

Leon Bruckman

Neighbor Discovery (ND) protocol

- This protocol solves a set of problems related to the interaction between nodes attached to the same link. It defines mechanisms for solving each of the following problems:
- Router Discovery:
  - How hosts locate routers that reside on an attached link.
- Prefix Discovery:
  - How hosts discover the set of address prefixes that define which destinations are on-link for an attached link. (Nodes use prefixes to distinguish destinations that reside on-link from those only reachable through a router.)
- Parameter Discovery:
  - How a node learns link parameters (such as the link MTU) or Internet parameters (such as the hop limit value) to place in outgoing packets.
- Address Autoconfiguration:
  - Introduces the mechanisms needed in order to allow nodes to configure an address for an interface in a stateless manner. (Stateless address autoconfiguration).
- Address resolution:
  - How nodes determine the link-layer address of an on-link destination (e.g., a neighbor) given only the destination's IP address.
- Next-hop determination:
  - The algorithm for mapping an IP destination address into the IP address of the neighbor to which traffic for the destination should be sent. The next- hop can be a router or the destination itself.
- Neighbor Unreachability Detection:
  - How nodes determine that a neighbor is no longer reachable. For neighbors used as routers, alternate default routers can be tried. For both routers and hosts, address resolution can be performed again.
- Duplicate Address Detection:
  - How a node determines whether or not an address it wishes to use is already in use by another node.
- Redirect:
  - How a router informs a host of a better first-hop node to reach a particular destination.

119

Leon Bruckman

Comparison with IPv4

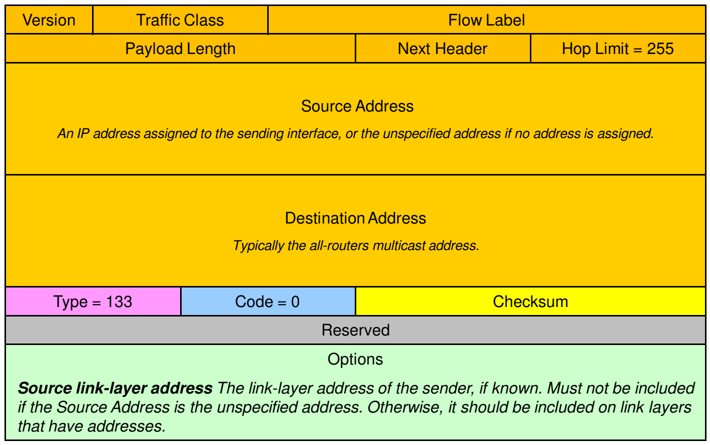
- The IPv6 Neighbor Discovery protocol corresponds to a combination of the IPv4 protocols Address Resolution Protocol (ARP), ICMP Router Discovery, and ICMP Redirect, and more.
- Some improvements of ND (see full list in RFC 4861):
  - Router Discovery is part of the base protocol set.
  - Router Advertisements carry prefixes for a link; there is no need to have a separate mechanism to configure the "netmask".
  - Routers can advertise an MTU for hosts to use on the link, ensuring that all nodes use the same MTU value on links lacking a well-defined MTU.
  - Redirects contain the link-layer address of the new first hop; separate address resolution is not needed upon receiving a redirect.
  - By setting the Hop Limit to 255, Neighbor Discovery is immune to off-link senders that accidentally or intentionally send ND messages. In IPv4, off-link senders can send both ICMP Redirects and Router Advertisement messages.
  - Placing address resolution at the ICMP layer makes the protocol more media-independent than ARP and makes it possible to use generic IP-layer authentication and security mechanisms as appropriate.

120

Leon Bruckman

Router Solicitation Message Format

- Hosts send Router Solicitations in order to prompt routers to generate Router Advertisements quickly.

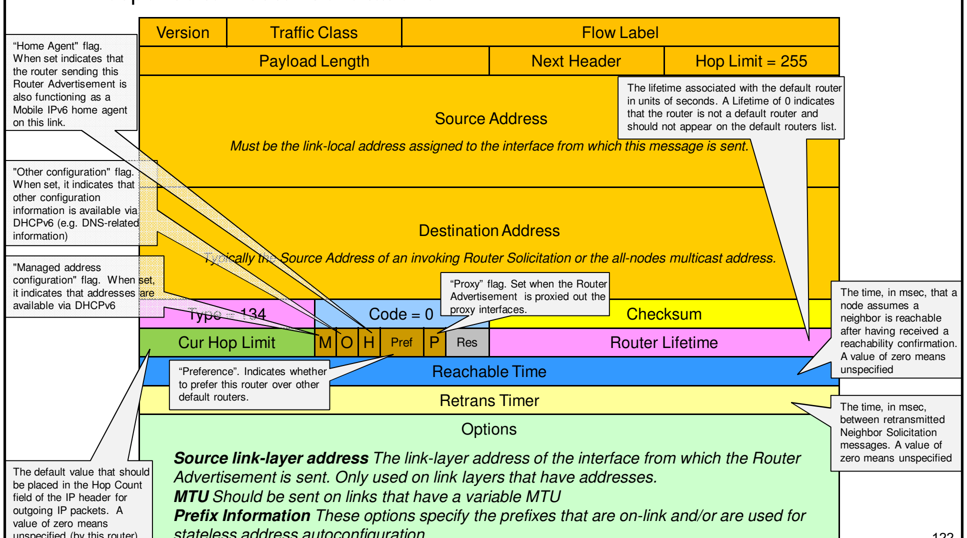


121

Leon Bruckman

Router Advertisement Message Format

- Routers send out Router Advertisement messages periodically, or in response to Router Solicitations.



122

Leon Bruckman

Neighbor Solicitation Message Format

- Nodes send Neighbor Solicitations to request the link-layer address of a target node while also providing their own link-layer address to the target. Neighbor Solicitations are multicast when the node needs to resolve an address and unicast when the node seeks to verify the reachability of a neighbor.

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit = 255
Source Address *Either an address assigned to the interface from which this message is sent or the unspecified address.*			
Destination Address *Either the solicited-node multicast address corresponding to the target address, or the target address.*			
Type = 135	Code = 0	Checksum	
Reserved			
Target Address *The IP address of the target of the solicitation. It must not be a multicast address.*			
Options ***Source link-layer address*** *The link-layer address of the sender, if known. Must not be included if the Source Address is the unspecified address. Otherwise, on link layers that have addresses this option must be included in multicast solicitations and should be included in unicast solicitations.*			

123

Leon Bruckman

Neighbor Advertisement Message Format

- A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly.

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit = 255
Source Address *An address assigned to the interface from which the advertisement is sent.*			
Destination Address *For solicited advertisements, the Source Address of an invoking Neighbor Solicitation or, if the solicitation's Source Address is the unspecified address, the all-nodes multicast address.* *For unsolicited advertisements typically the all nodes multicast address.*			
Type = 136	Code = 0	Checksum	
R | S | O | Reserved			
Target Address *For solicited advertisements, the Target Address field in the Neighbor Solicitation message that prompted this advertisement.* *For an unsolicited advertisement, the address whose link-layer address has changed. Must not be a multicast address.*			
Options ***Target link-layer*** *The link-layer address for the target, i.e., the sender of the advertisement. This option must be included on link layers that have addresses when responding to multicast solicitations. When responding to a unicast Neighbor Solicitation this option should be included.*			

Router flag. When set, the R-bit indicates that the sender is a router.

Solicited flag. When set, indicates that the advertisement was sent in response to a Neighbor Solicitation from the Destination address. The S-bit is used as a reachability confirmation for Neighbor Unreachability Detection.

Override flag. When set, the O-bit indicates that the advertisement should override an existing cache entry and update the cached link-layer address. When it is not set the advertisement will not update a cached link-layer address though it will update an existing Neighbor Cache entry for which no link-layer address is known.

124

Leon Bruckman

Redirect Message Format

<table>
<tr><td colspan="1">Version</td><td colspan="2">Traffic Class</td><td colspan="5">Flow Label</td></tr>
<tr><td colspan="4">Payload Length</td><td colspan="2">Next Header</td><td colspan="2">Hop Limit = 255</td></tr>
<tr><td colspan="8">Source Address
Must be the link-local address assigned to the interface from which this message is sent.</td></tr>
<tr><td colspan="8">Destination Address
The Source Address of the packet that triggered the redirect.</td></tr>
<tr><td colspan="2">Type = 137</td><td colspan="2">Code = 0</td><td colspan="4">Checksum</td></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><td colspan="8">Target Address
An IP address that is a better first hop to use for the ICMP Destination Address. When the target is the actual endpoint of communication, i.e., the destination is a neighbor, the Target Address field must contain the same value as the ICMP Destination Address field. Otherwise, the target is a better first-hop router and the Target Address must be the router's link-local address so that hosts can uniquely identify routers.</td></tr>
<tr><td colspan="8">Destination Address
The IP address of the destination that is redirected to the target.</td></tr>
<tr><td colspan="8">Options
Target link-layer The link-layer address for the target. It should be included (if known).
Redirected Header As much as possible of the IP packet that triggered the sending of the Redirect without making the redirect packet exceed the minimum MTU</td></tr>
</table>

Leon Bruckman

Conceptual Model of a Host

- Hosts maintain the following pieces of information for each interface:
- **Neighbor Cache**
  - A set of entries about individual neighbors to which traffic has been sent recently.
    - Entries are keyed on the neighbor's on-link unicast IP address and contain such information as its link-layer address, a flag indicating whether the neighbor is a router or a host, a pointer to any queued packets waiting for address resolution to complete, etc.
  - A Neighbor Cache entry also contains information used by the Neighbor Unreachability Detection algorithm, including the reachability state, the number of unanswered probes, and the time the next Neighbor Unreachability Detection event is scheduled to take place.
- **Destination Cache**
  - A set of entries about destinations to which traffic has been sent recently. The Destination Cache includes both on-link and off-link destinations and provides a level of indirection into the Neighbor Cache; the Destination Cache maps a destination IP address to the IP address of the next-hop neighbor.
    - This cache is updated with information learned from Redirect messages.

Leon Bruckman

Conceptual Model of a Host - continued

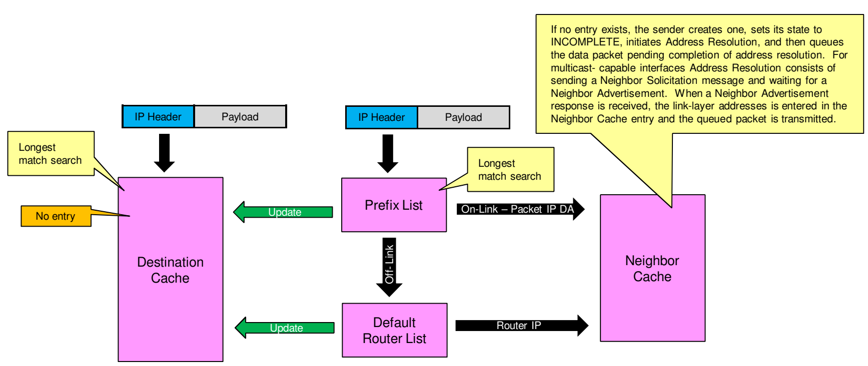
- **Prefix List**
  - A list of the prefixes that define a set of addresses that are on-link.
  - Prefix List entries are created from information received in Router Advertisements.
    - Each entry has an associated invalidation timer value (extracted from the advertisement) used to expire prefixes when they become invalid. A special "infinity" timer value specifies that a prefix remains valid forever, unless a new (finite) value is received in a subsequent advertisement.
- **Default Router List**
  - A list of routers to which packets may be sent.
    - Router list entries point to entries in the **Neighbor Cache**; the algorithm for selecting a default router favors routers known to be reachable over those whose reachability is suspect.
  - Each entry also has an associated invalidation timer value (extracted from Router Advertisements) used to delete entries that are no longer advertised.

127

Leon Bruckman

Conceptual Sending Algorithm

- When sending a packet to a destination, a node uses a combination of the Destination Cache, the Prefix List, and the Default Router List to determine the IP address of the appropriate next hop, an operation known as "next-hop determination".
  - Once the IP address of the next hop is known, the Neighbor Cache is consulted for link-layer information about that neighbor.



- For multicast packets, the next-hop is always the (multicast) destination address and is considered to be on-link.

128

Leon Bruckman

IPv6 Stateless Autoconfiguration

- **Link-Local Address Generation**: The device generates a link-local address. The generated address uses Link-local addresses (1111 1110 10) followed by 54 zeroes and then the 64 bit interface identifier. Typically this will be derived from the data link layer (MAC) address
- **Link-Local Address Uniqueness Test**: The node tests (using the Node Discovery ND protocol) to ensure that the address it generated isn't for some reason already in use on the local network (very unlikely).
- **Link-Local Address Assignment**: Assuming the uniqueness test passes, the device assigns the link-local address to its IP interface. This address can be used for communication on the local network, but not on the wider Internet (since link-local addresses are not routed).
- **Router Contact**: The node next attempts to contact a local router for more information on continuing the configuration. This is done using ND.
- **Router Direction**: The router provides direction to the node on how to proceed with the autoconfiguration. It may tell the node that on this network “stateful” autoconfiguration is in use, and tell it the address of a DHCP server to use. Alternately, it will tell the host how to determine its global Internet address.
- **Global Address Configuration**: Assuming that stateless autoconfiguration is in use on the network, the host will configure itself with its globally-unique Internet address. This address is generally formed from a network prefix provided to the host by the router, combined with the device's identifier as generated in the first step.

129

Leon Bruckman

Site renumbering

- Address configuration should facilitate the graceful renumbering of a site's machines.
  - For example, a site may wish to renumber all of its nodes when it switches to a new network service provider.
  - Renumbering is achieved through the leasing of addresses to interfaces and the assignment of multiple addresses to the same interface.
- At present, upper-layer protocols such as TCP provide no support for changing end-point addresses while a connection is open.
  - If an end-point address becomes invalid, existing connections break and all communication to the invalid address fails.
  - Even when applications use UDP as a transport protocol, addresses must generally remain the same during a packet exchange.
- Dividing valid addresses into **preferred** and **deprecated** categories provides a way of indicating to upper layers that a valid address may become invalid shortly and that future communication using the address will fail, should the address's valid lifetime expire before communication ends.
  - To avoid this scenario, higher layers should use a preferred address (assuming one of sufficient scope exists) to increase the likelihood that an address will remain valid for the duration of the communication.
  - It is up to system administrators to set appropriate prefix lifetimes in order to minimize the impact of failed communication when renumbering takes place.
  - The deprecation period should be long enough that most, if not all, communications are using the new address at the time an address becomes invalid.

130

Leon Bruckman

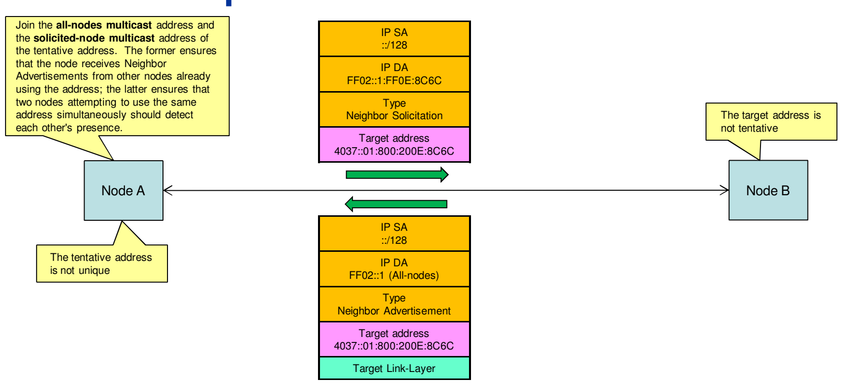
Duplicate Address Detection

- Duplicate Address Detection must be performed on all unicast addresses prior to assigning them to an interface, regardless of whether they are obtained through stateless autoconfiguration, DHCPv6, or manual configuration, with the following exceptions:
  - An interface whose DupAddrDetectTransmits variable is set to zero does not perform Duplicate Address Detection.
  - Duplicate Address Detection must not be performed on anycast addresses (note that anycast addresses cannot syntactically be distinguished from unicast addresses).
- An address on which the Duplicate Address Detection procedure is applied is said to be **tentative** until the procedure has completed successfully.
  - A tentative address is not considered "assigned to an interface" in the traditional sense.
  - The interface must accept Neighbor Solicitation and Advertisement messages containing the tentative address in the Target Address field, but processes such packets differently from those whose Target Address matches an address assigned to the interface.
  - Other packets addressed to the tentative address should be silently discarded.

131

Leon Bruckman

Duplicate address detection



- If the target address in the Neighbor Advertisement message matches a unicast address assigned to the receiving interface, it would possibly indicate that the address is a duplicate but it has not been detected by the Duplicate Address Detection procedure (Duplicate Address Detection is not completely reliable).
  - Otherwise, the advertisement is processed according to the Network Discovery protocol
- If the solicitation is from another node, the tentative address is a duplicate and should not be used (by either node).
- If the solicitation is from the node itself (because the node loops back multicast packets), the solicitation does not indicate the presence of a duplicate address. Detecting the solicitation source in this case:
  - If a Neighbor Solicitation for a tentative address is received before one is sent.
  - If the actual number of Neighbor Solicitations received exceeds the number expected based on the loopback semantics (e.g., the interface does not loop back the packet, yet one or more solicitations was received)

132