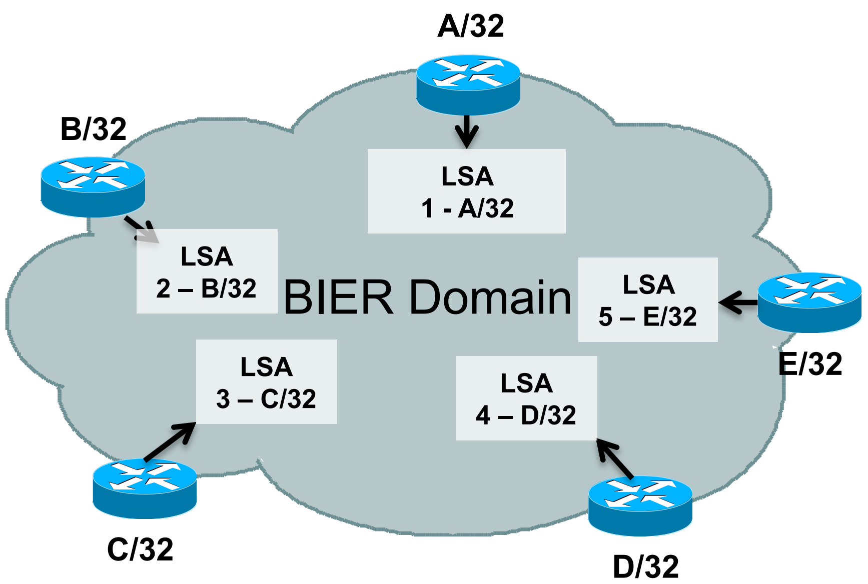
BIER BOF
IETF 91 – Honolulu, Hawaii

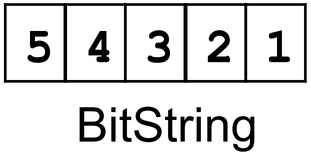
draft-wijnands-bier-architecture-01

IJsbrand Wijnands (ED)	Cisco
Eric Rosen (ED)	Juniper
Andrew Dolganow	Alcatel-Lucent
Tony Przygienda	Ericsson
Sam Aldrin	Huawei

Overview

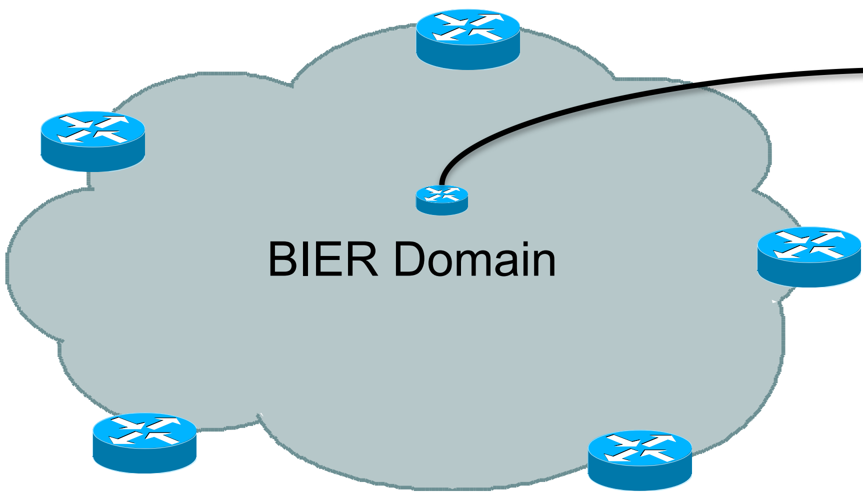
Basic Idea BIER





1. Assign a unique Bit Position from a BitString to each BFER in the BIER domain.
2. Each BFER floods their Bit Position to BFR-prefix mapping using the IGP (OSPF, ISIS)

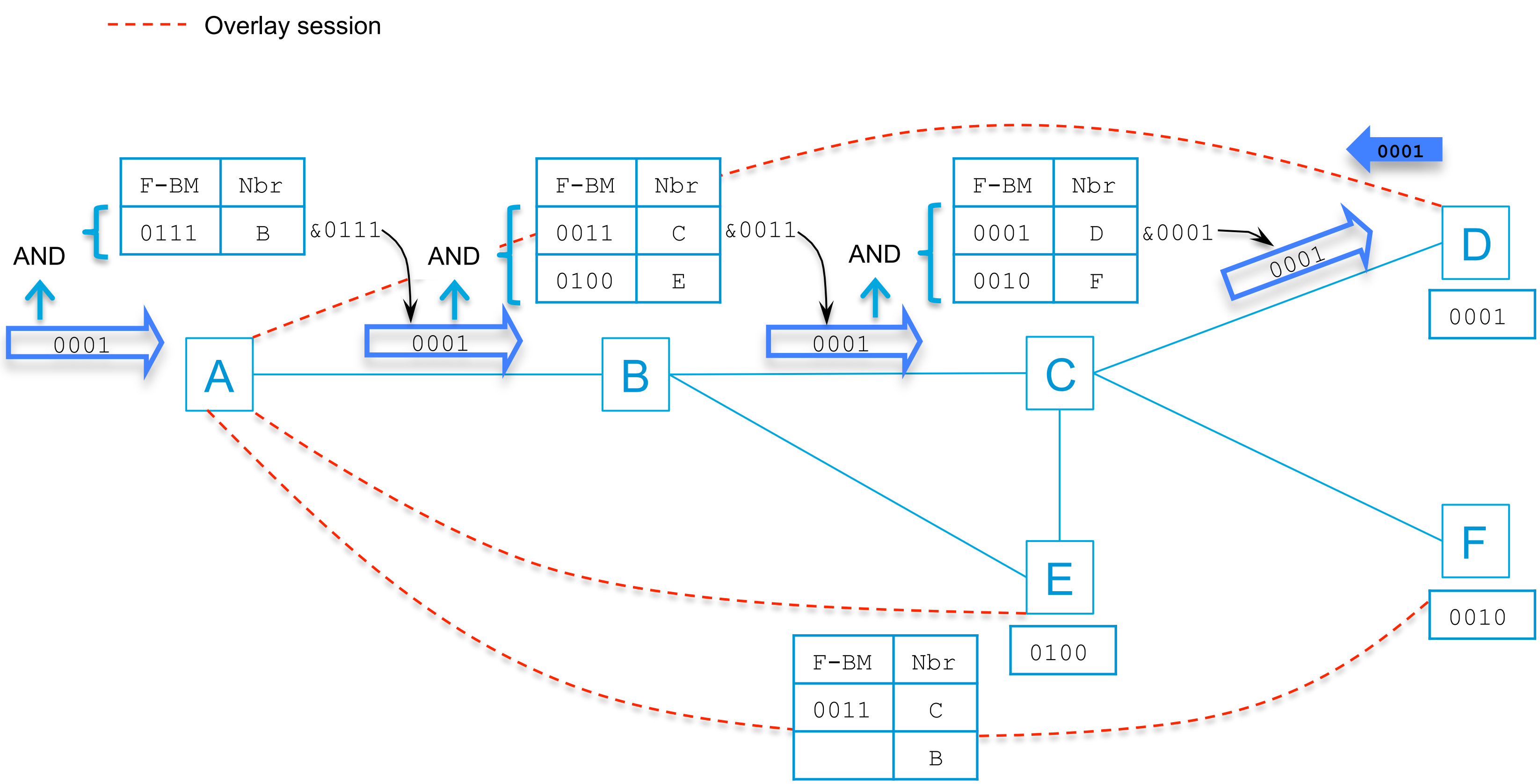
Basic Idea BIER



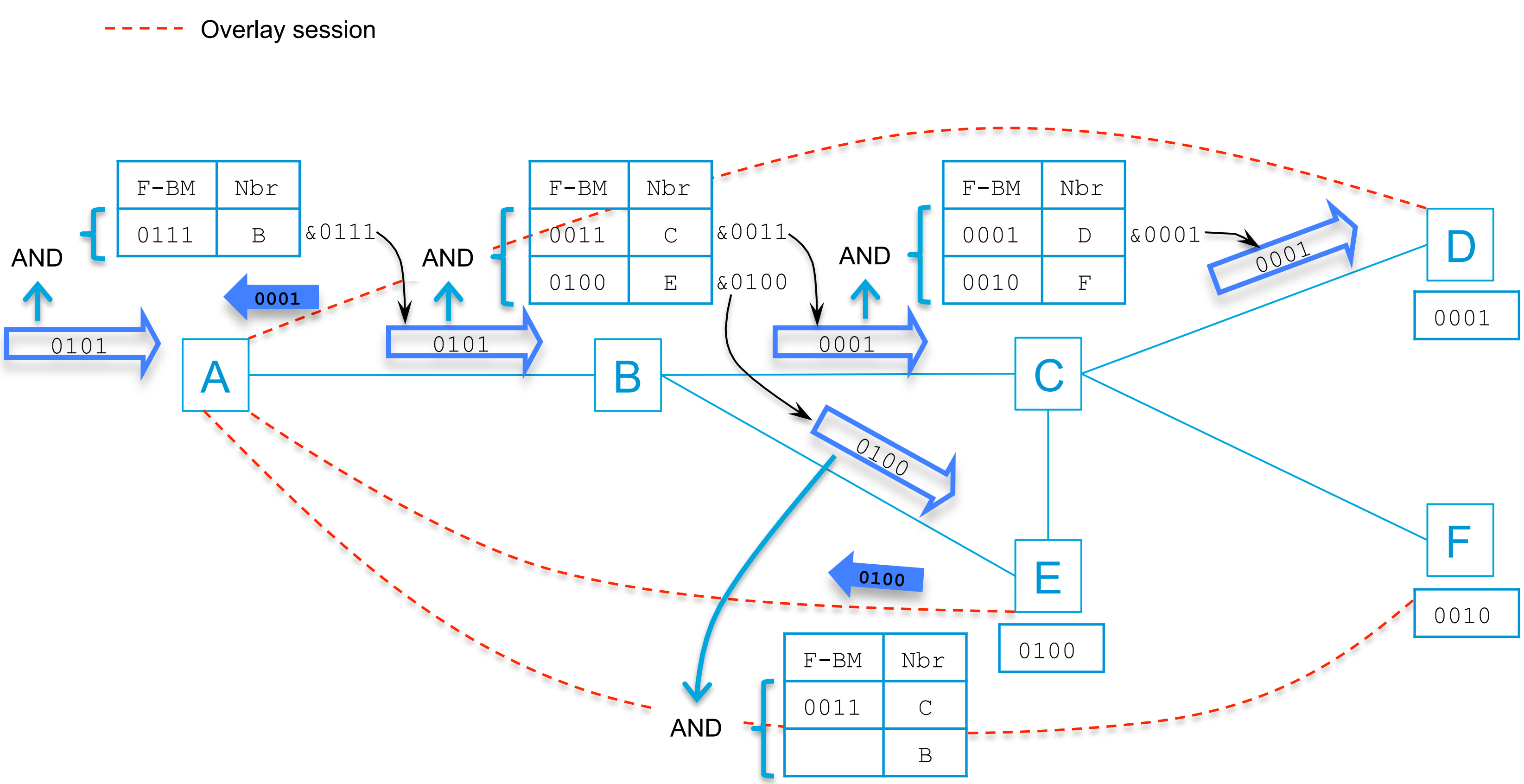
BitMask	Nbr
0011	A
0100	B
1000	C

1. Assign a unique Bit Position from a mask to each edge router in the BIER domain.
2. Each edge router floods their bit-position-to-ID mapping with a new LSA – OSPF or ISIS
3. All BFR's use unicast RIB to calculate a best path for each BFR-prefix
4. Bit Positions are OR'd together to form a Bit Mask per BFR-nbr
5. Packets are forwarded and replicated hop-by-hop using the Bit Forwarding Table..

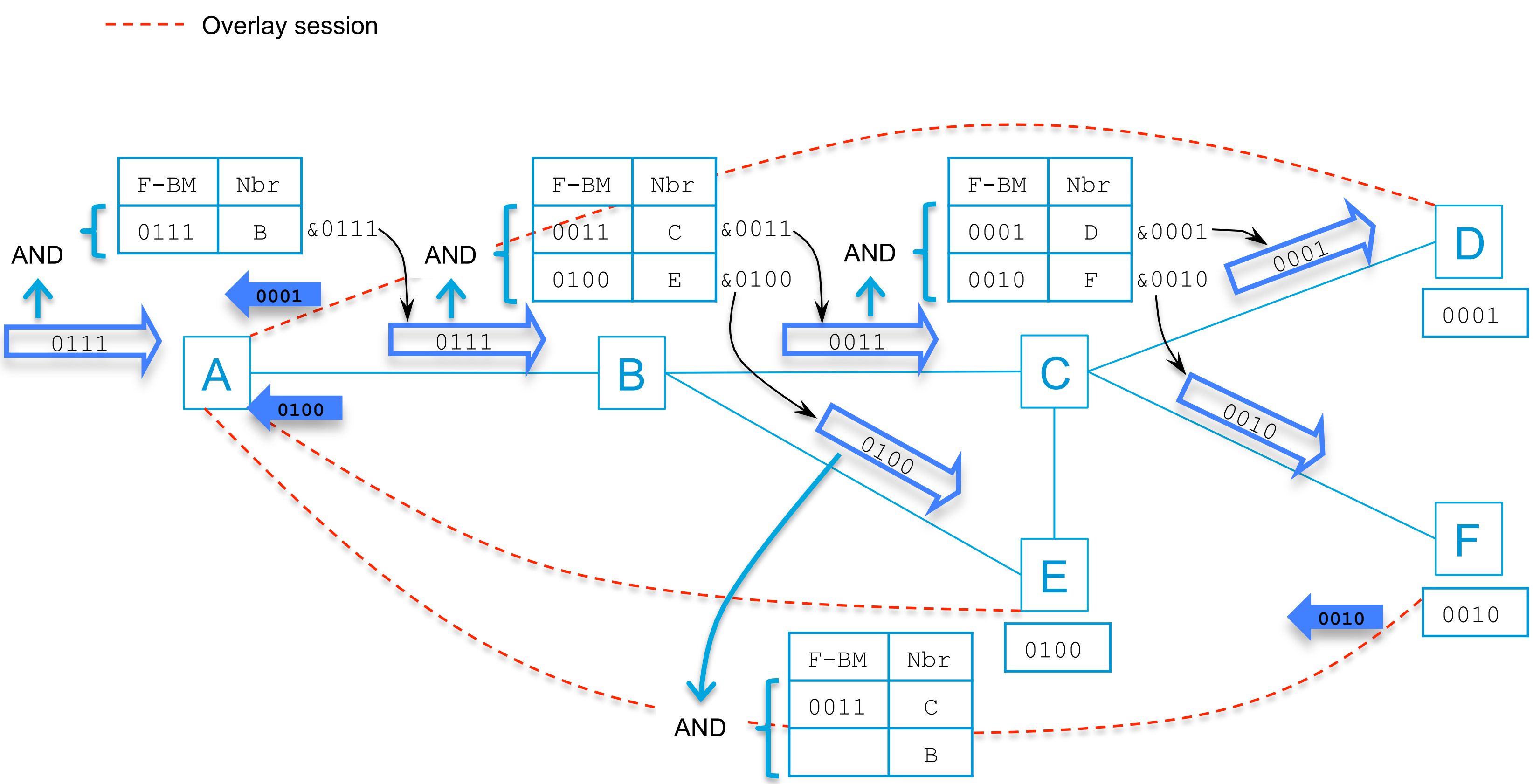
BIER forwarding example



BIER forwarding example



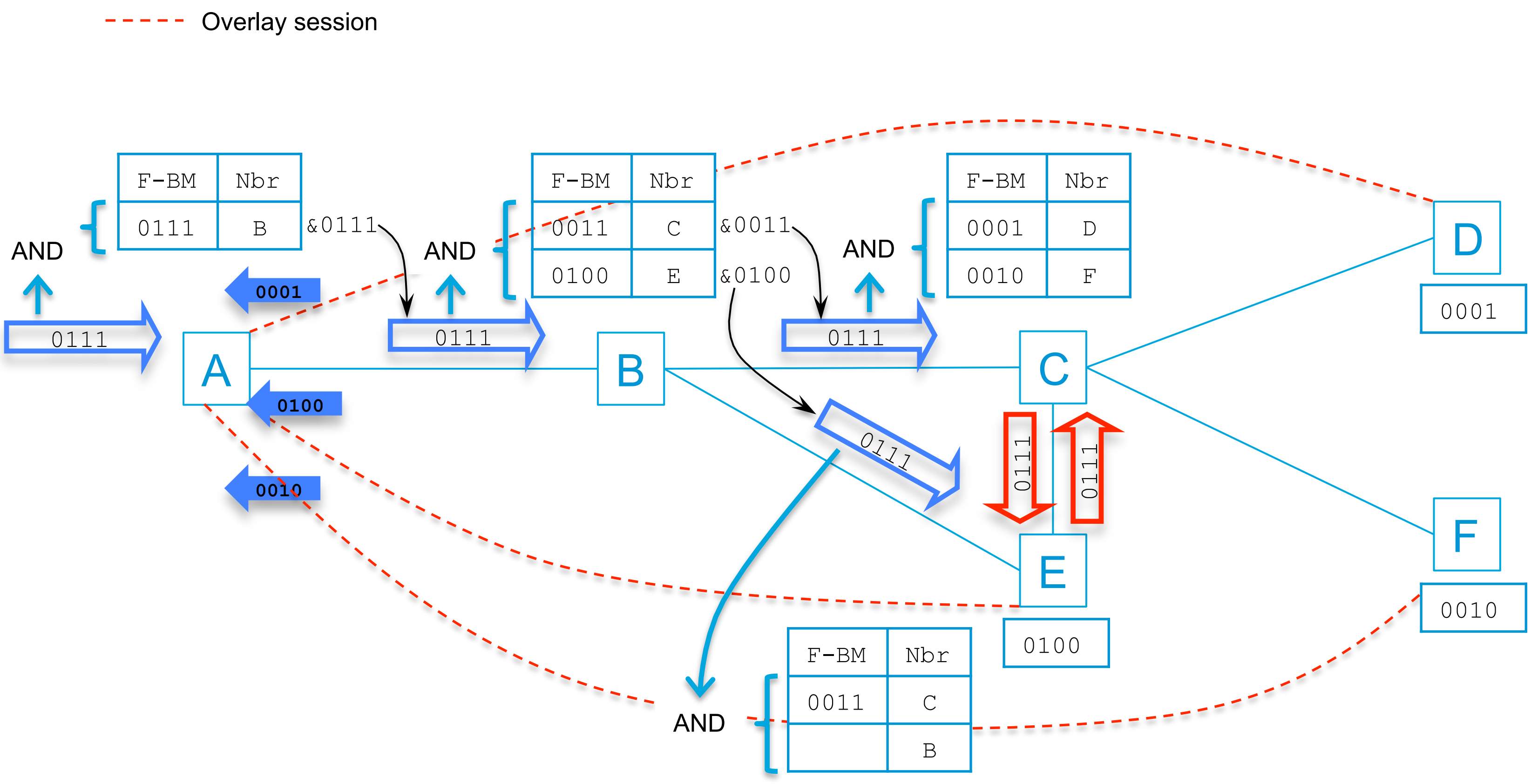
BIER forwarding example



BIER forwarding example

- As you can see from the previous slides, the result from the bitwise AND (&) between the Bit Mask in the packet and the Forwarding table is copied in the packet for each neighbor.
- This is the key mechanism to prevent duplication.
- Look at the next slide to see what happens if the bits are not reset
- If the previous bits would not have been reset, E would forward the packet to C and vice versa.

Forwarding Packets

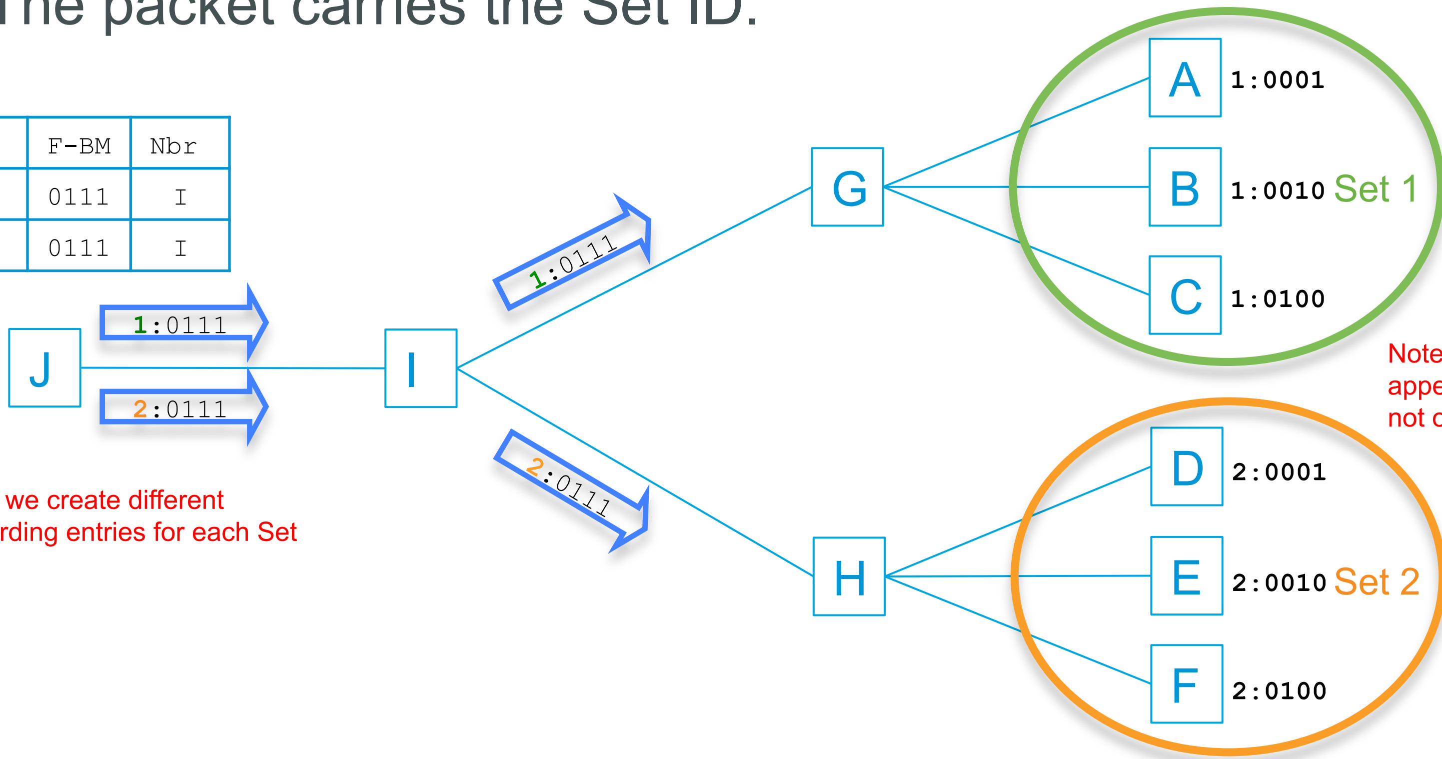


Sets & Area's

BIER Sets

- To increase the scale we group the egress routers in Sets.
- Each Bit Position is unique in the context of a give Set
- The packet carries the Set ID.

Set	F-BM	Nbr
1	0111	I
2	0111	I



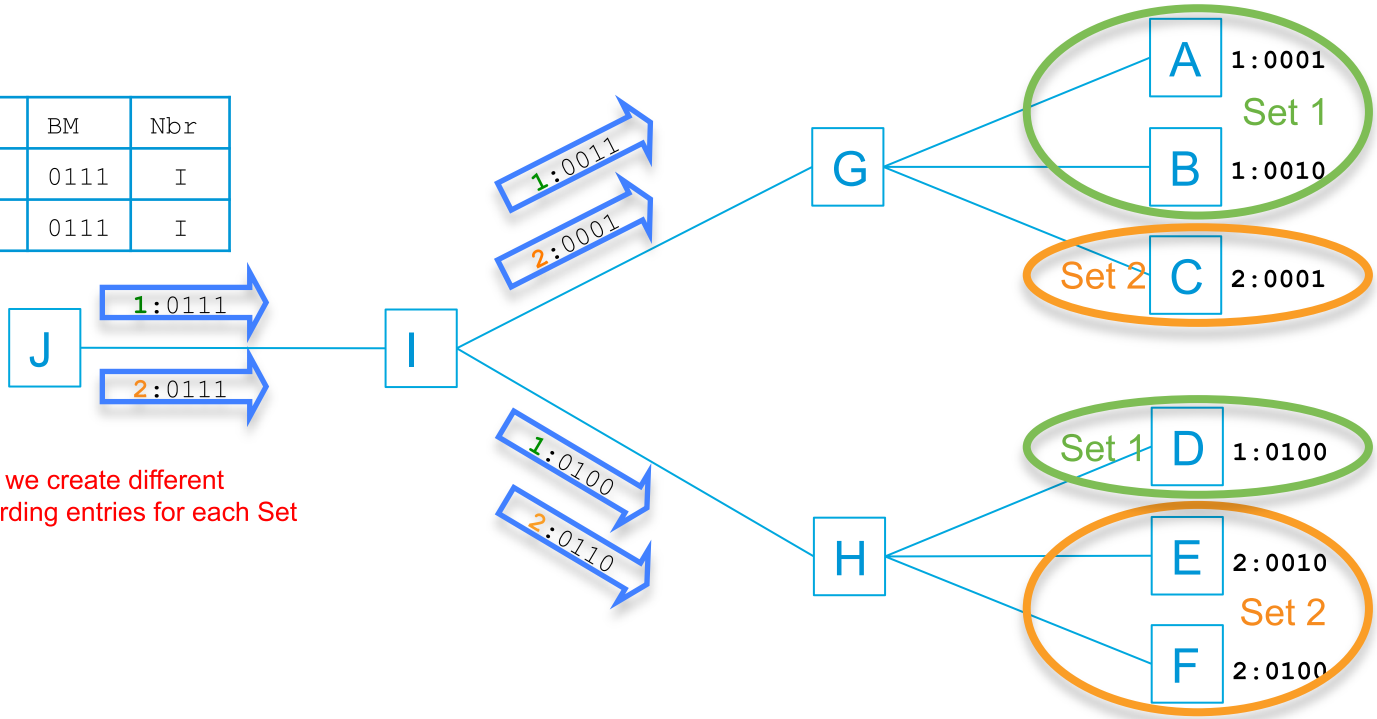
Note, Bit Positions 1,2,3 appear in both Sets, jet do not overlap due to Sets.

Note, we create different forwarding entries for each Set

BIER Sets

- There is no topological restriction which set an egress belongs to
- But it may be more efficient if it follows the topology

Set	BM	Nbr
1	0111	I
2	0111	I



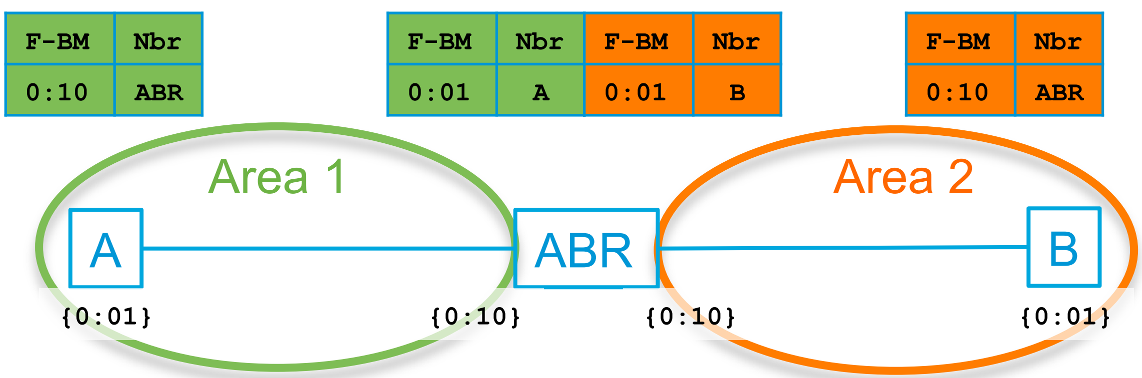
Note, we create different forwarding entries for each Set

BIER Sets

- If a multicast flow has multiple receivers in different Sets, the packet needs to be replicated multiple times by the ingress router, for each set once.
- The Set identifier is part of the packet.
- Can be implemented as MPLS label.

BIER Area

- A bit Mask only needs to be unique in its own area.
- ABR's translate Bit Masks between area's.
- Requires a IP lookup and state on the ABRs.



Forwarding Optimization

Forwarding optimization

Bit Index Routing Table					
5	4	3	2	1	Nbr
			1	1	A
		1			B
	1				C
1					D

- When a packet is received with a BitString, the F-BM needs to be matched against each Neighbor in the topology table to determine which neighbor to replicate the packet to.
- If the router has many neighbors, this may be a long walk for each packet.
- In order to prevent this walk, we have the following optimization.

Forwarding optimization

Bit Index Routing Table

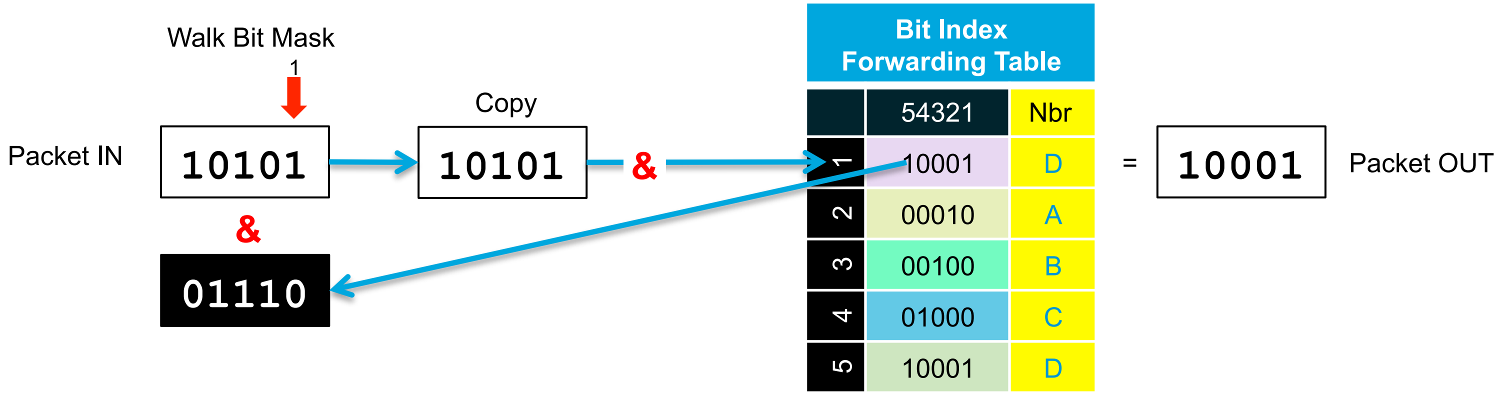
5	4	3	2	1	Nbr
			1		A
		1			B
	1				C
1				1	D

Bit Index Forwarding Table

	54321	Nbr
1	10001	D
2	00010	A
3	00100	B
4	01000	C
5	10001	D

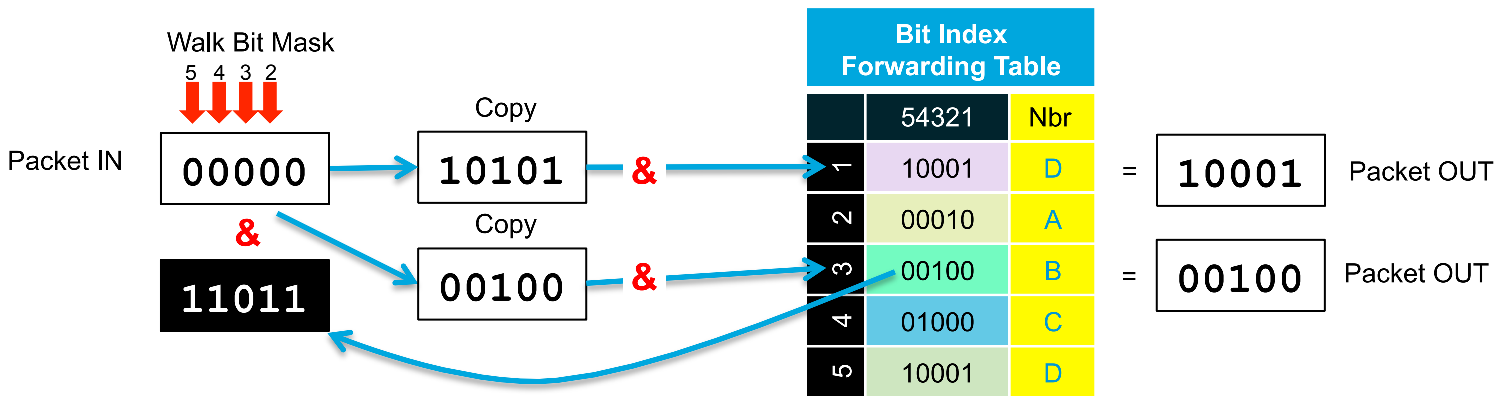
- We translate the Bit Index Routing Table in to a Bit Index Forwarding Table by sorting the table on Bit Position (and not by neighbor).
- We walk the BitString in the packet and index into the BIFT.

Forwarding optimization



- We walk the Bits in the packet, as soon as we hit a ‘1’, we copy the packet, index into the BIFT with the position of the Bit.
- The BitMask entry is reverse ‘&’ with the BitString in the packet.
- This resets the Bits that where processed.

Forwarding optimization



- We walk the Bits in the packet, as soon as we hit a '1', we copy the packet, index into the BIFT with the position of the Bit.
- The Bit Mask entry is reverse '&' with the BitString in the packet.
- This resets the Bits that where processed.

Forwarding optimization

- Walking the bits in a BitString takes less clock cycles compared to walking a list of neighbors.
- For that reason its faster to walk the BitString and index into BIFR
- The table is a NxN bit matrix, where N is the maximum BitString length.
- Bits that where already processed are reset so we don’t processes them if they appears later in a Bit Mask. This way we avoid multiple copies being forwarded.

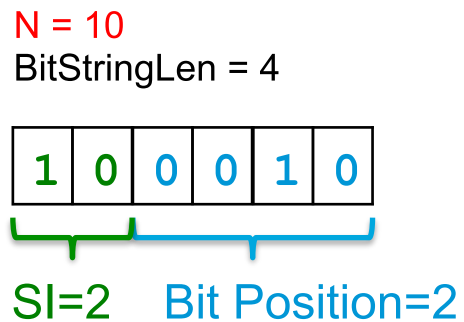
The BFR-id

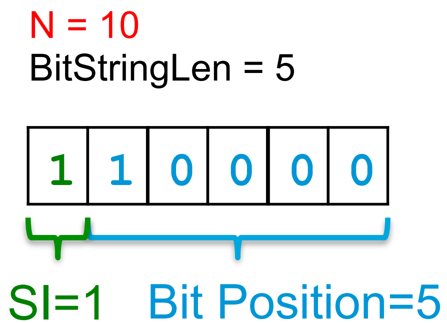
The BFR-id

- A BFER is uniquely identified by a two tuple {Set ID, Bit Position}
- The Bit Position range is depending on the length of the support BitString in the network.
- To make the BFER identifier independent of the BitString length we defined the BFR-id as a number [1,65535]
- We map the BFR-id Number into a {SI, BP}, based on the BitStringLength.

SI = (N-1)/BitStringLength

BP = ((N-1) modulo BitStringLength)+1





Encapsulation

- The architecture document is independent of the encapsulation that is used to store the BitString in the packet.
- Different encapsulations are possible.
- Before we start to engage in an other round of discussions regarding the encapsulation preference ☺, lets analyze the use-cases and the requirements more.
- It is too early to debate a ‘generic’ encoding.

Conclusion

Advantages

- Packets forwarded via BIER follow the unicast path towards the receiver, inheriting unicast features like FRR and LFA.
- There is no per multicast flow state in the network.
- Multicast convergence is as fast as unicast, there is no multicast state to re-converge, signal, etc.
- Nice plugin for SDN, its only the ingress and egress that need to exchange Sender and Receiver information.
- The core network provides many-2-many connectivity between all BIER routers by default, following the IGP.
- No Multicast control protocol in the network.

Disadvantages

- The Bit String Length has an upper bound and may not cover all deployment scenarios.
- Using sets to increase the number of egress routers may cause the ingress to replicate the packet multiple times.
- Existing HW is not optimized for BIER forwarding


NL
1st BoF
BIER
BIT INDEXED EXPLICIT REPLICATION
bier@ietf.org - http://trac.tools.ietf.org/bof/trac/
IETF Nov 2015
Hawaii
Honolulu