

BIER Egress Protection

draft-chen-bier-egress-protect-00

Huaimo Chen
Aijun Wang
Gyan S. Mishra
Yanhe Fan
Lei Liu
Xufeng Liu

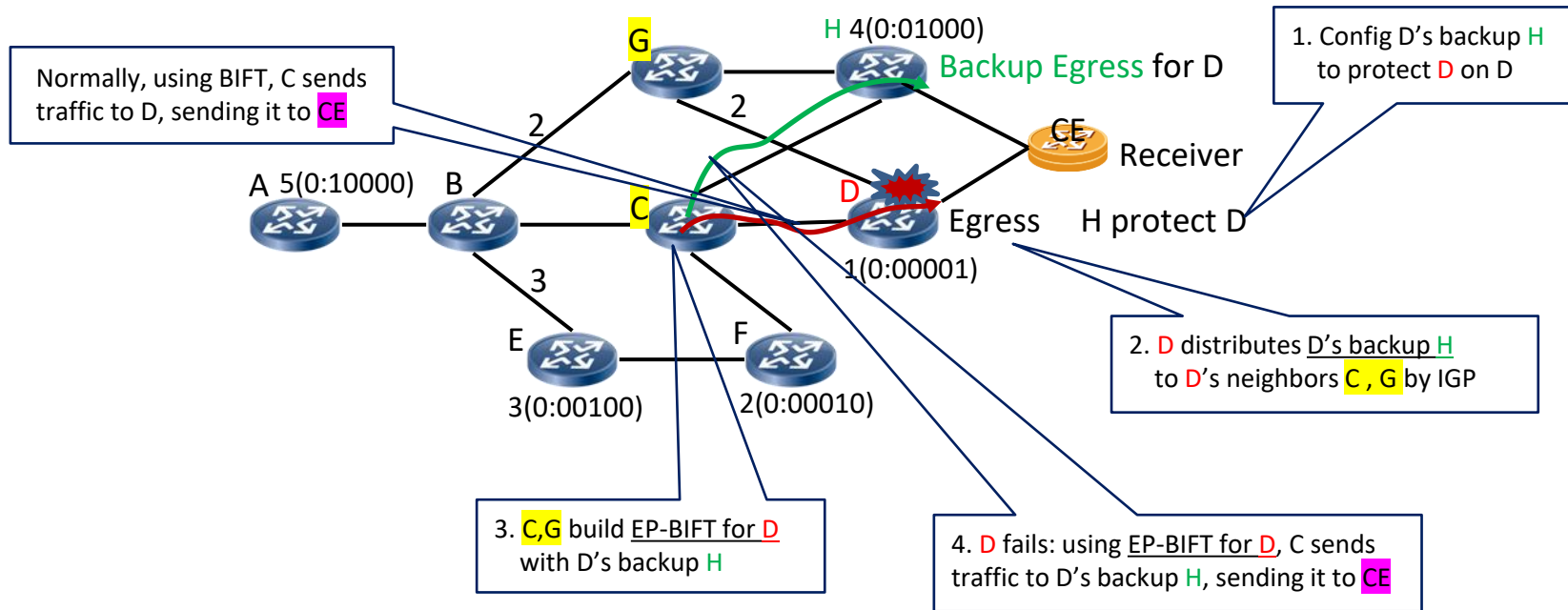
IETF 109

Introduction

BIER Egress Protection (EP)

- Fast protection for egress nodes in BIER domain
- No per-flow state in the core

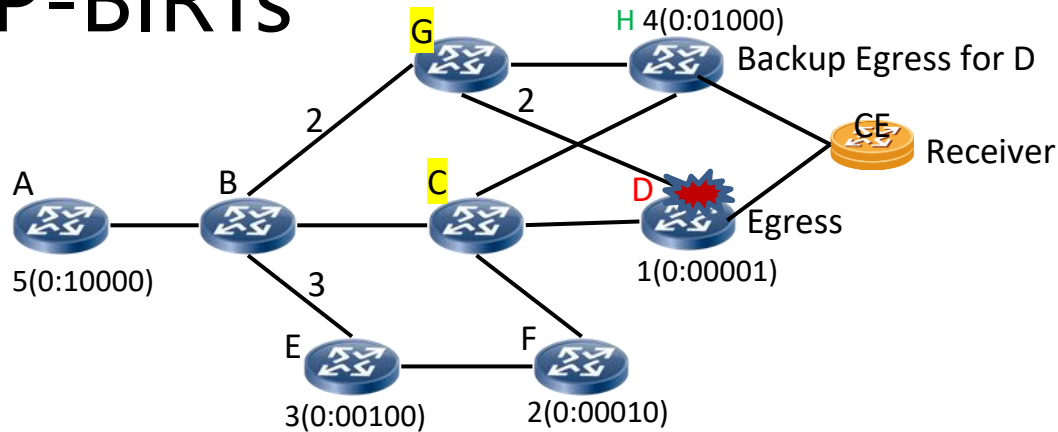
BIER Egress Protection Overview



BIER Egress Protection (EP) Idea

1. Configure backup egress to protect an egress (e.g., H is configured to protect D on D)
2. Egress distributes its backup to its neighbors (e.g., D distributes "H backup D" to C and G)
3. Neighbors build EP-BIFT for egress, and EP-BIFT from EP-BIRT (e.g., C builds EP-BIRT for D, G builds EP-BIRT for D),
4. Forwards packets using EP-BIFT for egress (e.g., D) when egress fails, to backup egress, sending packets to CE receiver.

Build EP-BIRTs



- Each neighbor of egress Y has an EP-BIRT: EP-BIRT for Y (e.g., C and G are neighbors of D, C has EP-BIRT for D, G has EP-BIRT for D)
- EP-BIRT for Y has route to Y's backup egress without going Y (as Y fails), and backup information for Y (e.g., EP-BIRT for D on C has route to H without going D as D fails, backup information for D is {EP=1, BE-BFER = H}, meaning D as egress is protected by backup egress H)

Building EP-BIRT for Y on BFR (e.g., building EP-BIRT for D on C)

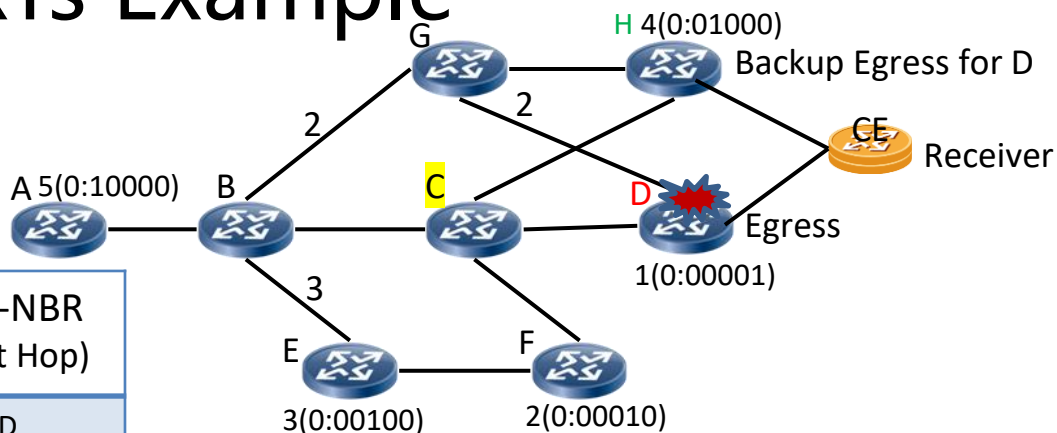
1. Copy BIRT to EP-BIRT for Y,
2. Add backup information to EP-BIRT for Y

Row with BFER = Y and BFR-NBR = Y has backup information {EP=1, BE-BFER=BFR-id of backup egress}. Each of other rows has {EP=0, BE-BFER=NULL}.

EP-BIRTs Example

Building EP-BIRT for **D** on BFR **C**

1. Copy BIRT to EP-BIRT for **D**



EP-BIRT for **D**
copied from BIRT

BFR-id (SI:Bitstring)	BFR-Prefix of Dest BFER	BFR-NBR (Next Hop)
1(0:00001)	D	D
2(0:00010)	F	F
3(0:00100)	E	F
4(0:01000)	H	H
5(0:10000)	A	B

2. Add backup information to EP-BIRT for **D**

EP-BIRT for **D**
w/ backup info

BFR-id (SI:Bitstring)	BFR-Prefix of Dest BFER	BFR-NBR (Next Hop)	{EP, BE=BFER} (Backup info)
1(0:00001)	D	D	EP=1, BE=BFER=H
2(0:00010)	F	F	EP=0, BE=BFER=0
3(0:00100)	E	F	EP=0, BE=BFER=0
4(0:01000)	H	H	EP=0, BE=BFER=0
5(0:10000)	A	B	EP=0, BE=BFER=0

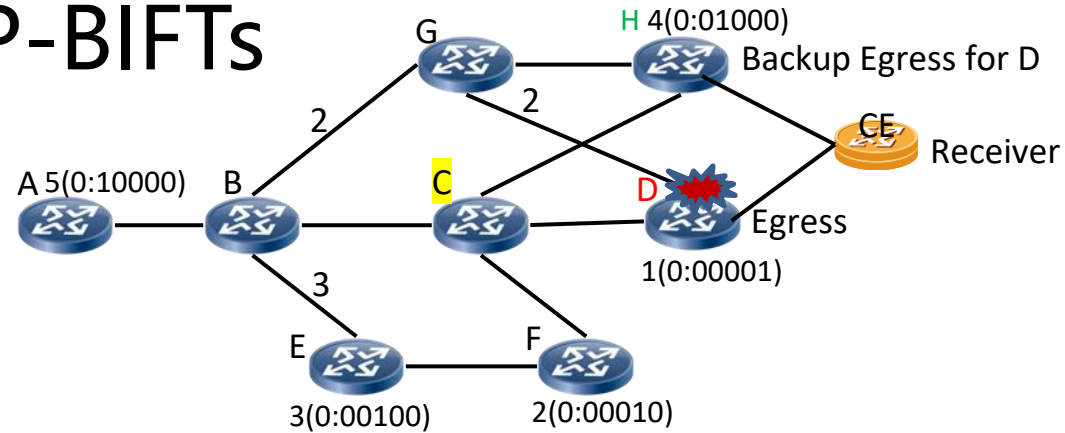
Row with BFER=D
and BFR-NBR=D

Row with **BFER = D** and **BFR-NBR = D** has **{EP=1, BE=BFER=H}**

EP-BIFTs

EP-BIFT is derived from EP-BIRT (similar to BIFT from BIRT)

Rows with same SI, BFR-NBR and backup information, have F-BM = OR of BitStrings in rows



EP-BIRT for D

BFR-id (SI:Bitstring)	BFR-Prefix of Dest BFER	BFR-NBR (Next Hop)	{EP, BE=BFER} (Backup info)
1(0:00001)	D	D	EP=1, BE=BFER=H
2(0:00010)	F	F	EP=0, BE=BFER=0
3(0:00100)	E	F	EP=0, BE=BFER=0
4(0:01000)	H	H	EP=0, BE=BFER=0
5(0:10000)	A	B	EP=0, BE=BFER=0

F-BM for F and E
= 00010 OR 00100
= 00110

EP-BIFT for D
derived from
EP-BIRT for D

BFR-id (SI:Bitstring)	F-BM	BFR-NBR (Next Hop)	{EP, BE=BFER} (Backup info)
1(0:00001)	00001	NULL	EP=1, BE=BFER=H
2(0:00010)	00110	F	EP=0, BE=BFER=0
3(0:00100)	00110	F	EP=0, BE=BFER=0
4(0:01000)	01000	H	EP=0, BE=BFER=0
5(0:10000)	10000	B	EP=0, BE=BFER=0

C forwards packets using EP-BIFT for D when D fails

Packets to D are sent to backup egress H, delivering them to CE

Updated Forwarding Procedure

```
Packet = the packet received by BFR;  
FOR each BFER k (from rightmost in Packet's BitString) {  
  IF BFER k is the BFR itself {  
    copies Packet, sends the copy to the multicast  
    flow overlay and clears bit k in Packet's BitString  
  } else {  
    finds the row in EP-BIFT for the sub-domain using  
    Packet's SI and BitString as the key/index  
    IF EP == 1 {  
      clears bit k in Packet's BitString; //BFER k is PE-BFER  
      adds bit j in Packet's BitString; //BFER j is BE-BFER  
    } ELSE {  
      IF BFR-NBR in the row is not NULL {  
        Copies Packet, updates copy's BitString by ANDing  
        it with F-BM in row, sends updated copy to BFR-NBR  
      } // BFR-NBR == NULL, not sent Packet to BFR-NBR  
      updates Packet's BitString by ANDing it with the  
      INVERSE of the F-BM in the row  
    }  
  }  
}
```

When egress k fails, if it is protected (EP==1), then packet to k is not sent to k (by clearing bit k), but is sent to k's backup egress j (by adding bit j)

Next Steps

- Welcome comments