BIER IPv6 Encapsulation

1. draft-xu-bier-encapsulation-06 ------(Since 2014.9)
2. draft-pfister-bier-over-ipv6-01 -------(Since 2016.9)
3. draft-zhang-bier-bierin6-03 ----------(Since 2017.11)
4. **draft-xie-bier-ipv6-encapsulation-03 ----(Since 2018.4)**

Presenter: Jingrong Xie

IETF105 2019-07-24 Montreal

Applicability to understand

- Fast-path “BIER natively in IPv6”:
  - End.BIER in DA for fast-path.
  - Indication of VRF using SA makes the NH chain even shorter.
- Single encapsulation for one-hop and multi-hop replication:
  - Intra-domain bypassing Non-BIER routers .
  - Inter-domain deployment in IPv6-only networks.
- Complete and unique solution:
  - draft-ietf-bier-ipv6-requirements-01
  - draft-xie-bier-ipv6-encapsulation-03 //This presentation
  - draft-xie-bier-ipv6-isis-extension-00
  - draft-xie-bier-ipv6-mvpn-01
  - draft-geng-bier-ipv6-inter-domain-00

Meet the Requirements

#	Requirements in <draft-ietf-bier-ipv6-requirements>	Comply
1	**L2 Agnostic**	Full
2	**Hop by hop DA modification not required (*)**	No
3	**L4 Inspection not required**	Full
4	**Multicast address in SA field not allowed**	Full
5	**Incorrect bits not causing damage**	Full
6	**SA filtering support**	Full
7	**BIER architecture support (SD/BSL/SI/ECMP/Bypass)**	Full
8	**Keep it simple with single encapsulation**	Full
9	**Hardware fast path**	Full

* Suggested to deprecate.

Request for

- Reading the drafts and the slides below !
- Further Review and Comments !
- Adoption call this year when all things clear !

Thank you !

Main concerns received

Questions since ietf104: Jeffrey

- Jeffrey: How does multicast DA works in LAN env ?
  - Per the feedback since IETF104, we decide to use unicast DA only.
- Jeffrey: If you use unicast DA, does the packet be sent to control plane?
  - Good point.
  - It is improved in the updated draft by using a “BIER specific IPv6 address” as destination address.
  - “BIER specific IPv6 address” = End.BIER (same as BFR-prefix or different)
  - End.BIER = Introduce new function (just like SRv6 End). Advertise by new sub-TLV in IGP.
  - Data plane can be programmed to treat End.BIER function specially (more details in the later slides) .
- Jeffrey: Use of BIER multicast address FF0X::AB37 does not seem right to me. Assuming that we agree that a unicast DA address must be used.
  - Very clear points on this.
  - It is improved in the updated draft (03) by using “unicast DA address” only.

Questions since ietf104: Ice

- Ice: One more point about HW support, We can't assume, if you support BIER we can support other bier solution like ethernet we have discussed here.
  - Ack! BIERv6/BIER-ETH both need a little new hardware capability!
  - Both require the whole BIER header process! That’s exactly the same!
- Ice: One more point about HW support, Where to put the bits? With SRv6, one of things is extention can be put in by application, because it's L3… But for routers, its a pain as the bits are so far inside the packet and it needs read deep inside
  - Enlightenment from SRv6, L3 router is not only responsible for encapsulation, but for reading inside the packet.
  - SRv6 first introduces the processing of extension header in fast-path!
  - Key to this is the use of a “Function Specific IPv6 Address” in IPv6 DA as a preceding indication! After that, simply “**process the desired packet**” and “**drop the undesired packet**” **without walking through the EH chain**.
  - It doesn’t need reading deeply inside ----it’s the shallowest.

Questions since ietf104: Tony

- **Tony: About unicast DA to bypass non-bier capable nodes, Tunnel exist and it works. Dont do shortcuts.**
  - The proposal of BIERv6 is a P2MP tunnel beginning from BFIR ending at BFERs.
  - Overlay things are processed *only* on BFERs ! BFRs will not be impacted !
    - Extension Headers after the BIER option header are not walked through on BFRs
    - For example, Frag/AH/ESP.
- **Tony: If 6man allows you to not change SA, thats cool. But i think, putting it inside options is overruled.**
  - It's very common for IPv6 to change options data in EH while not change the SA.
  - There are many examples (see previous page).
  - Before hurrying to 6man, I think it may be our responsibility to make it clear that this is the way our "BIER natively in IPv6" solution want to go. Shell we ?
- **Tony: Next-proto = BIER is far more practical.**
  - Different intuition/impression of using IPv6 EH in Layer-3 fast-path process! ----Eric Rosen @2016.
  - Impractical in most cases if the variable size header have to be processed in hardware! ---- Douglas E. Comer @2004.
  - Let's explore/compare the different proposals with rationale…

Questions since ietf104: Toerless

- **Toerless: One of the interesting question is, whether you are allowed to change the bitstrings at transit without calling the operation be decapsulation and re-encapsulation…… And if we call it as the later one, should the node which re-encapsulate are required to change the source-address?**
  - There is clear text in RFC8200 allowing IPv6 options “change en route”.
  - There are examples of “change en route” options, for example, 0x23/0x26/0x6D/0xEE. See below:
  - https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2
- **Toerless: I was wondering, no extension headers, it will take away the pain of 8200 stuffs…But then we may have to re-write the source and destination address at each hop.**
  - Ack. We think the same: using a Layer 3 extension header don’t have to re-write the source address. Using Layer-4 may have to re-write the source address.
- **Toerless: We need to wait for the result in SRv6 arch and encap work as they consider 8200 requirements during review. It applicable to hear as well.**
  - Ack. Significant progress in SRv6 since then. Drafts lastcall or adopted as WG docs.
- **Toerless: Waterfall model? Requirements first, then find solutions? so wondering how to go about it ?**
  - Ack. Discuss different solutions and the pros and cons, and Summarize the requirements from there ……

The deep insights by Eric Rosen @2016

- **Why not consider IPv6 Extension Header:**

  Is the proposal being compared to another proposal that does use IPv6 extension headers?

  (None of the proposals I've seen so far use IPv6 extension headers.) //Intuition[0]

  https://www.ietf.org/mail-archive/web/bier/current/msg00986.html

- **Hints that should be understand and taken carefully:**

  "it shouldn't require hop-by-hop modification of the IP destination address field, //Intuition[1]

  and it shouldn't require the BFRs to inspect layer 4, //Intuition[2].

  and it shouldn't require changes to layer 4, //Intuition[3].

  and it shouldn't allow a multicast address to be put in the IP source address field,

  and it shouldn't assume that bits never get set incorrectly,

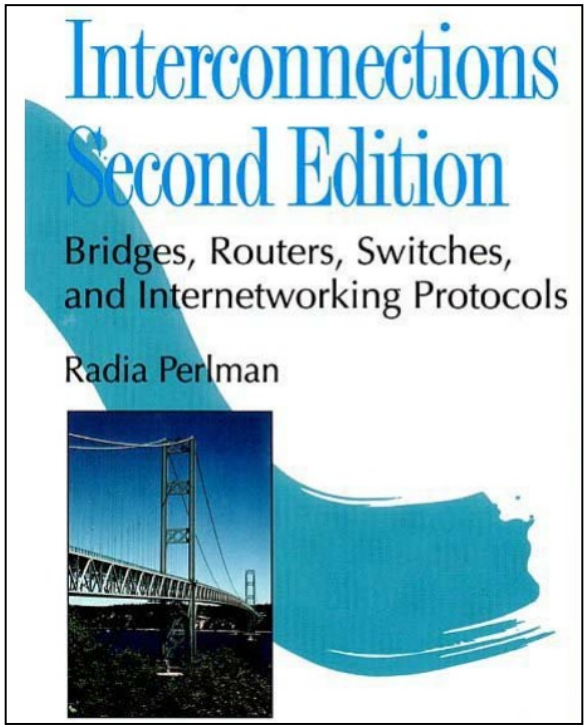
  and it shouldn't require changes in source address filtering procedures.

  In addition, it should be possible to use it to support the entire BIER architecture" //Intuition[4].

  https://www.ietf.org/mail-archive/web/bier/current/msg00991.html

IPv6 Extension Header Fast-Path Processing

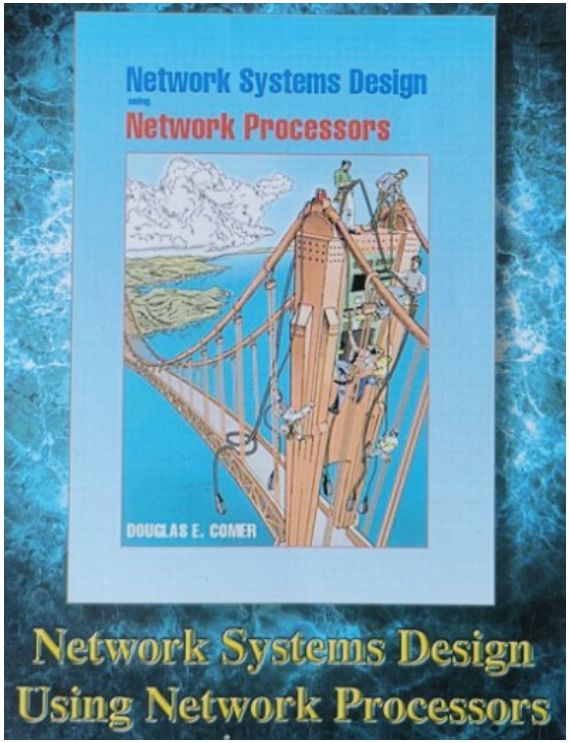
No Idea how to use **E**xt**H**dr --Radia Perlman @2000



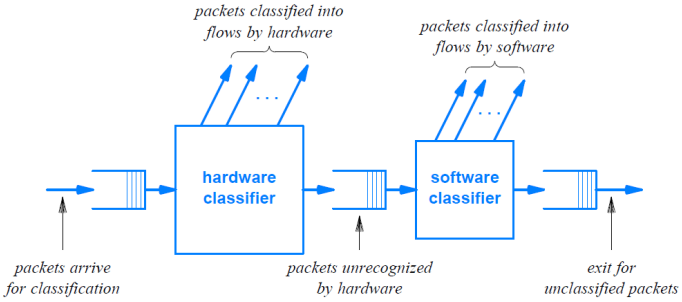
- IPv4, IPv6, and CLNP have a provision for "options." Each has several options defined, and any subset of them (including none) can appear in any packet.
- ISO, being a standards body, was reluctant to specify how to define options—thus, it provided this level of "flexibility."
- However, just to show that it does have the clout to make rules, it decreed that no option should appear twice.
- This restriction places a burden on routers (are they supposed to check that no option appears twice every time they forward a packet?) and reduces flexibility.
- For example, it precludes using padding in multiple places or using both the globally unique form of an option and the source-specified version in the same packet.

- [XJR] Even the ISO didn’t know how to use/leverage the ExtHdr or Options.
- [XJR] They didn’t suppose the ExtHdr handled in fast-path.

Variable size header in slow-path --Douglas E Comer @2004

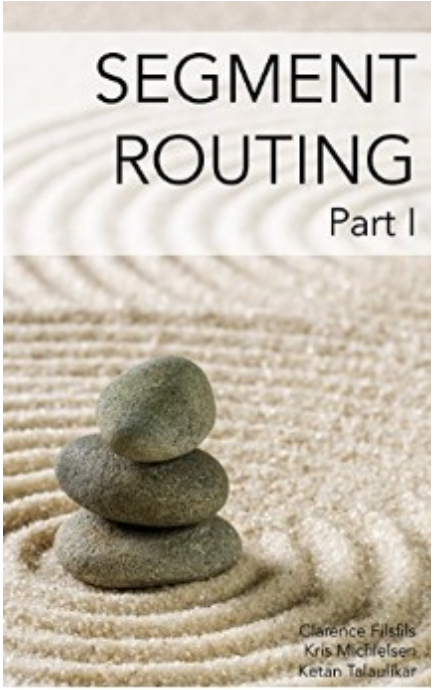


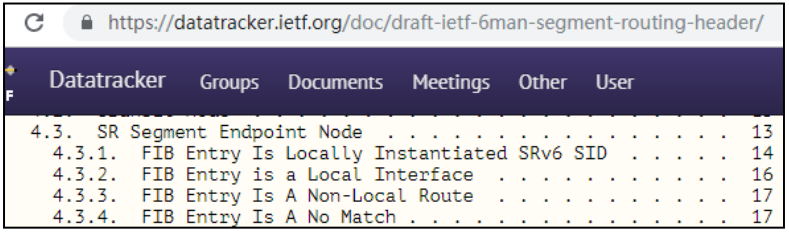
- How can a hardware classifier handle variable-size headers without sacrificing speed?
- If it recognizes a packet, the hardware classifier diverts the packet into an appropriate flow;
- Unrecognized packets a passed to the software classifier.
- It's possible to combine high-speed hardware classification with the low cost and increased flexibility of software classification.



- [XJR] Walk of the EH chain or TLVs is proper for slow-path!
- [XJR] True for decades and the near future I suppose.

SRv6: 1st fast-path process of EH! --Clarence @2017





- Short-cut the process when **FIB lookup** result an “V6 SID” !
- It makes the process of EH in Fast-Path **simple and direct** !
  - Process the desired packet with correct EH/Option.
  - Drop the undesired packet otherwise.
  - It **doesn’t need** to walk through the EH chain!

- [XJR] SRv6 uses a new branch in the FIB lookup for a “Function specific” IPv6 address.
- [XJR] SRv6 introduce new paradigm of IPv6 Extension Header in Fast-Path.

BIER in IPv6
Fast-Path Processing

BIER in L4 payload vs. BIER in L3 EH

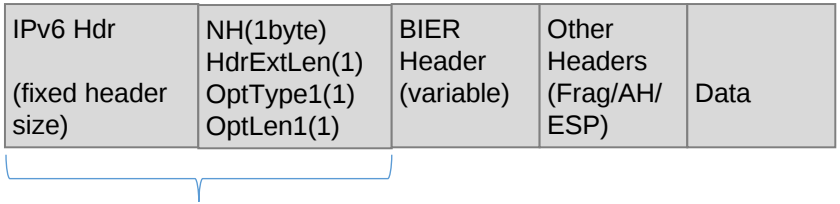
```
Result = FIB Lookup(DA)                                  //Step 0
Switch(Result)
  Case Local Interface IPv6 Address:                     //Step 1
    If packet is (NH=BIER)                               //Step 2 (*A*)
       Process it
    Else if packet is (NH=a/b/c/d & Last_NH=BIER)        //Step 3
       Process it
    Else If packet is (NH=XXXX)                          //Step 4 (*B*)
       Process it
    Else if packet is (NH=a/b/c/d & Last_NH=XXXX)        //Step 5
       Process it
    Else If packet is (NH=YYYY)                          //Step 6 (*C*)
       Process it
    Else if packet is (NH=a/b/c/d & Last_NH=YYYY)        //Step 7
       Process it
    Else
       Do normal things as usual                         //Step 8 (*D*)
  Case Non-Local Routable IPv6 Address
     Do normal routing and forwarding as usual.
```

Per the <draft-zhang-bier-bierin6-03>
(A) need 2~3 steps! --Inefficient
(B) need 4~5 steps!
(C) need 6~7 steps!
(D) need 8 steps! --significant impact

```
Result = FIB Lookup(DA)                                  //Step 0
Switch(Result)
  Case End.BIER:                                         //Step 1
    IF NH=60 and OptType1=BIER and OptLen1=HdrExtLen*8+4 //Step 2(A)
       Process it
    ELSE IF (NH=ICMPv6) or (NH=60 and Dest_NH=ICMPv6)
       Send to CPU.
    ELSE
       Drop the packet.
  Case End.XXXX:                                         //Step 1
    IF NH=60 and OptType1=XXX and OptLen1=HdrExtLen*8+4 //Step 2(B)
       Process it
    ELSE
       Drop the packet
  Case End.YYYY:                                         //Step 1
    IF NH=60 and OptType1=YYY and OptLen1=HdrExtLen*8+4 //Step 2(C)
       Process it
    ELSE
       Drop the packet
  Case Local Interface IPv6 Address:                     //Step 1
       Do normal things as usual                         //Step 2(D)
  Case Non-Local Routable IPv6 Address
       Do normal routing and forwarding as usual.
```

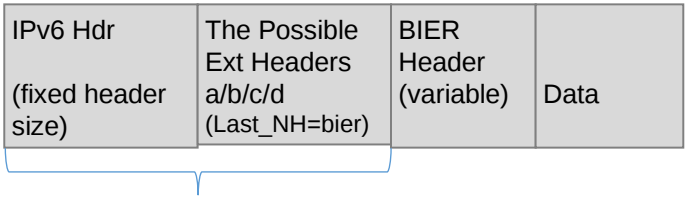
Per the <draft-xie-bier-ipv6-encapsulation>
(A) need 2 steps! --more efficient
(B) need 2 steps!
(C) need 2 steps!
(D) need 2 steps! --minimun impact

Why is that?



Using End.BIER in DA

- Hardware only check the fixed header size !
- Drop packets that confirmed using End.BIER in DA but not the desired packet!
- Support of additional L3 EHs without impact to existing functions.

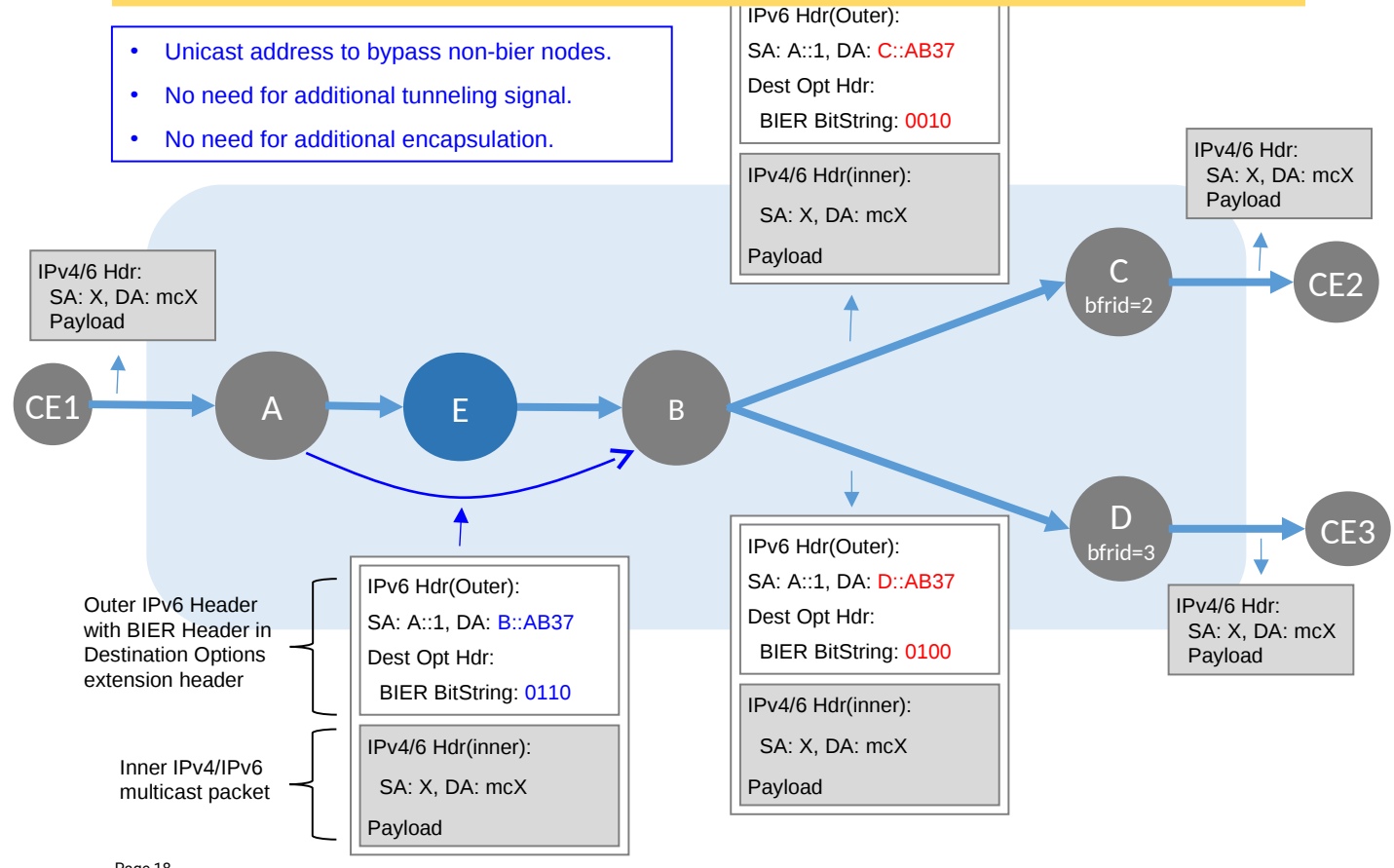


Using Regular IPv6 in DA

- Hardware has to check the ext header chain before getting to Last_NH=BIER!
- Existing functions based on regular IPv6 will be impacted significantly.
- Support of additional L3 EHs will be impractical considering the impact.

The Whole Picture

- Unicast address to bypass non-bier nodes.
- No need for additional tunneling signal.
- No need for additional encapsulation.



Proposal Evolution and Conclusion

The way we get here

Intuition	Rationale
Transport-Independent Layer-4 BIER <draft-xu-bier-encapsulation> @2014.9	Comp up with L2.5 BIER-MPLS and BIER-Ethernet. Haven't explored the advantages of IP (vs L2.5)
BitString in lower-bit of IPv6 DA <draft-pfister-bier-over-ipv6> @2016.9	It requires "Special IPv6 prefix" being handled on each BFR. Lack of the key arch support (e.g., BSL 128 above).
BIER in IPv6 Ext Header //Intiution[0]	Never explored until <bier-6man-encapsulation>.
NextHeader=BIER is direct <draft-zhang-bier-bierin6> @2017.11	Re-work of the 2014 <xu-bier> Layer-4 draft. Summarize from Eric's deep insight : L4 kills the L3 features like SRH/Frag/AH/ESP.
Dest Address should' t change //Intiution[1] <draft-xie-bier-6man-encapsulation>@2018.4	Come up with the idea of using multicast DA, but turns out not suitable for some key scenarios.
SA should change due to BitSting change	There have been "Change en route" type of options and examples.
IPv6 Ext Header is complex and slow <draft-xie-bier-ipv6-encapsulation>@2019.7	It's true for decades until SRv6. FIB had extended simply since then. Use of a BIER specific IPv6 address called End.BIER make this draft complete and unique!!

Conclusion

- **Fast-path handling** (compatible with **FIB extension** introduced by SRv6):
  - Least Impact ----Switch-case doesn’t impact other cases.
  - Most Efficient ----Doesn’t need walking through EHs in fast-path.
  - L3 Extensible ----For future integration with other L3 functions like SRH/AH/etc.
- **Simple one encapsulation for wide scenarios:**
  - One encapsulation for normal forwarding and Non-BIER routers bypassing.
  - One encapsulation for Intra-domain and Inter-domain deployment
- **BIER architecture fully compatible:**
  - Multiple SD, **BSL**, SI, ECMP; Bypassing Non-BIER routers.
- **IPv6 architecture fully compatible:**
  - DestOptHdr with only a **change-en-route** type of TLV, and RFCs as examples.