draft-lu-fn-transport

IETF 80 - Prague, Czech Republic
March 27 – April 1, 2011

Ericsson

W. Lu
S. Kini
A. Csaszar
G. Enyedi
J. Tantsura
A. Tian

Transport of Fast Notification Messages

- This draft specifies a generic, light-weight event notification protocol - Fast Notification (FN) as a separate transport layer, which focuses on quick, reliable and secure delivery of notifications

- It describes design goals, the message format and options for delivering the notifications.

Design Goals

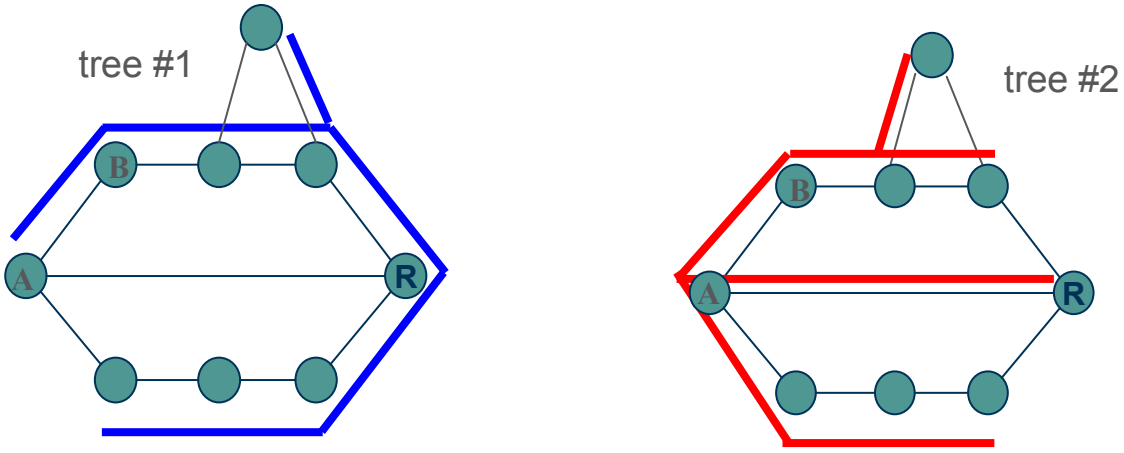
- A light-weight event notification mechanism that facilitates quick dissemination of information:
  - **Fast:** done/processed in FW plane
  - **Reliable:** under network failure conditions
  - **Secure:** provide means to verify the authenticity of a notification
  - **Independent:** not dependent upon routing protocol flooding procedures

Transport Logic - Distribution of the Notifications

- The distribution of a notification to multiple receivers can be implemented in many ways (see draft's appendix).

- The option proposed in the draft - **dual redundant trees**
  - Bi-dir multicast trees
- This option allows each notification to be delivered to any node in the area in case of single node or link failure.

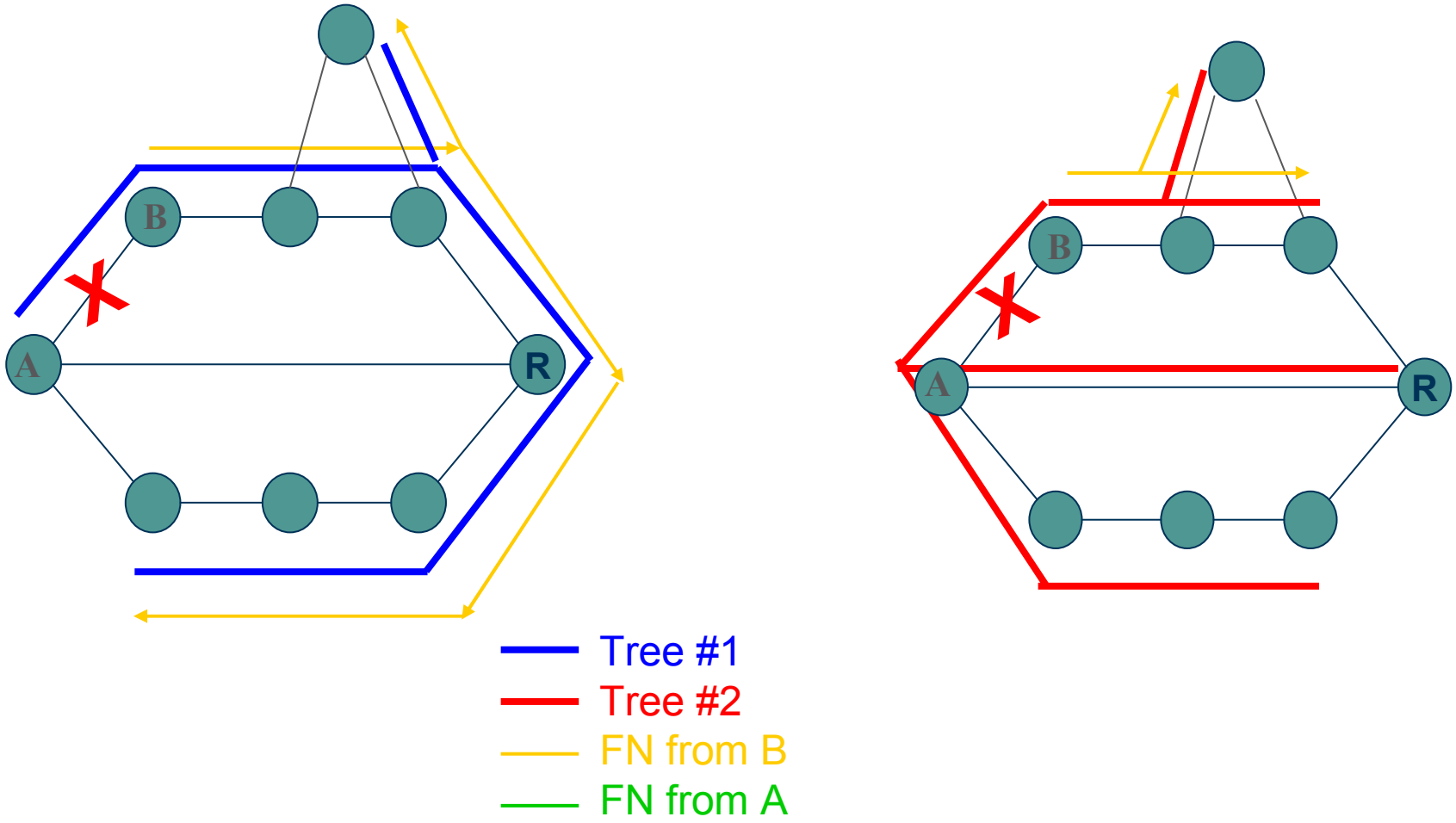
Pair of Redundant Trees



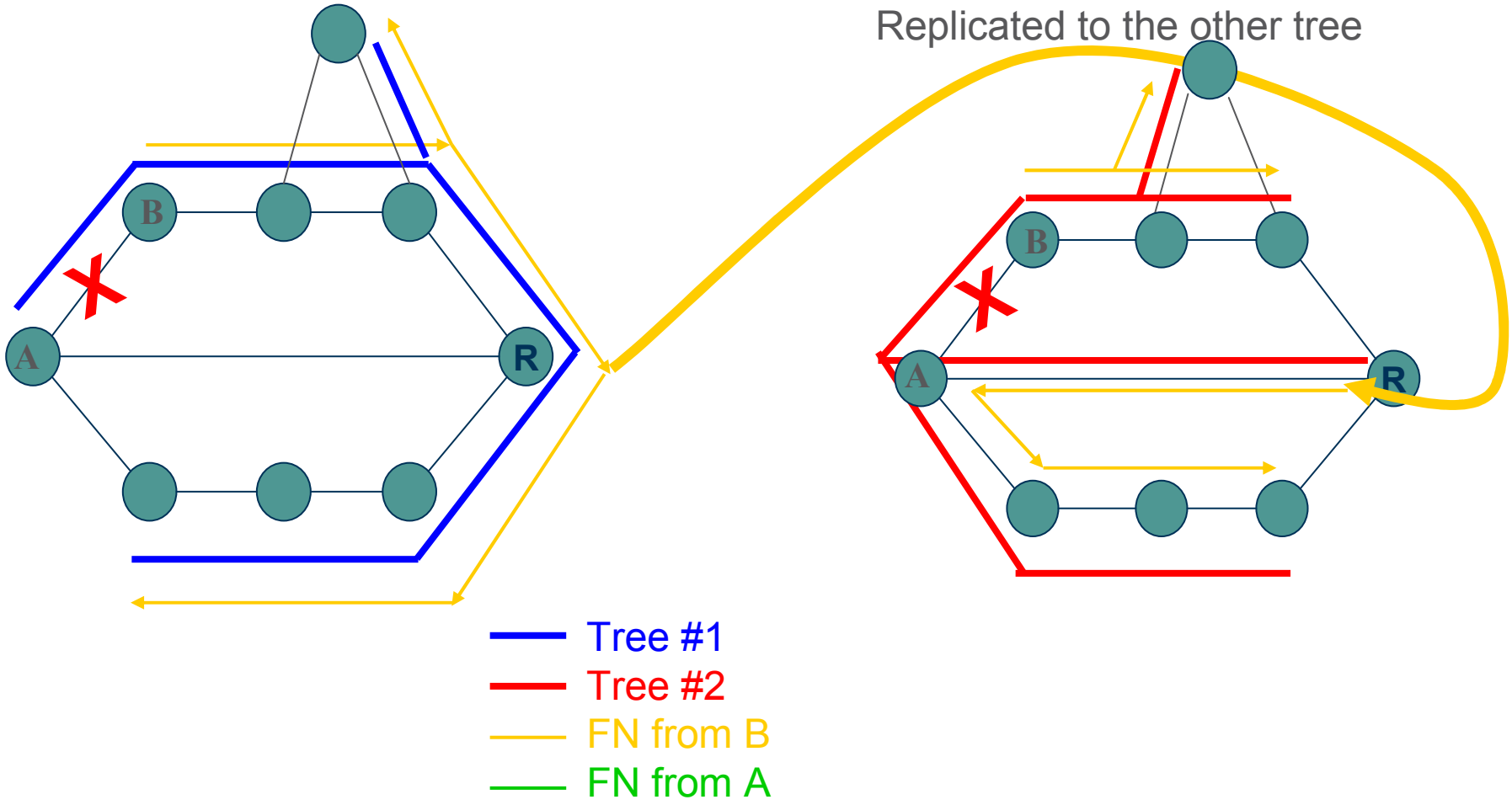
In case of link A-B failure, the notifications originated on node B will reach R on tree #1 and get replicated on the tree#2 while notifications originated on node A will reach R on tree #2 and get replicated on the tree#1

From R, each node is reachable through one of the trees, so each node will be notified about both events

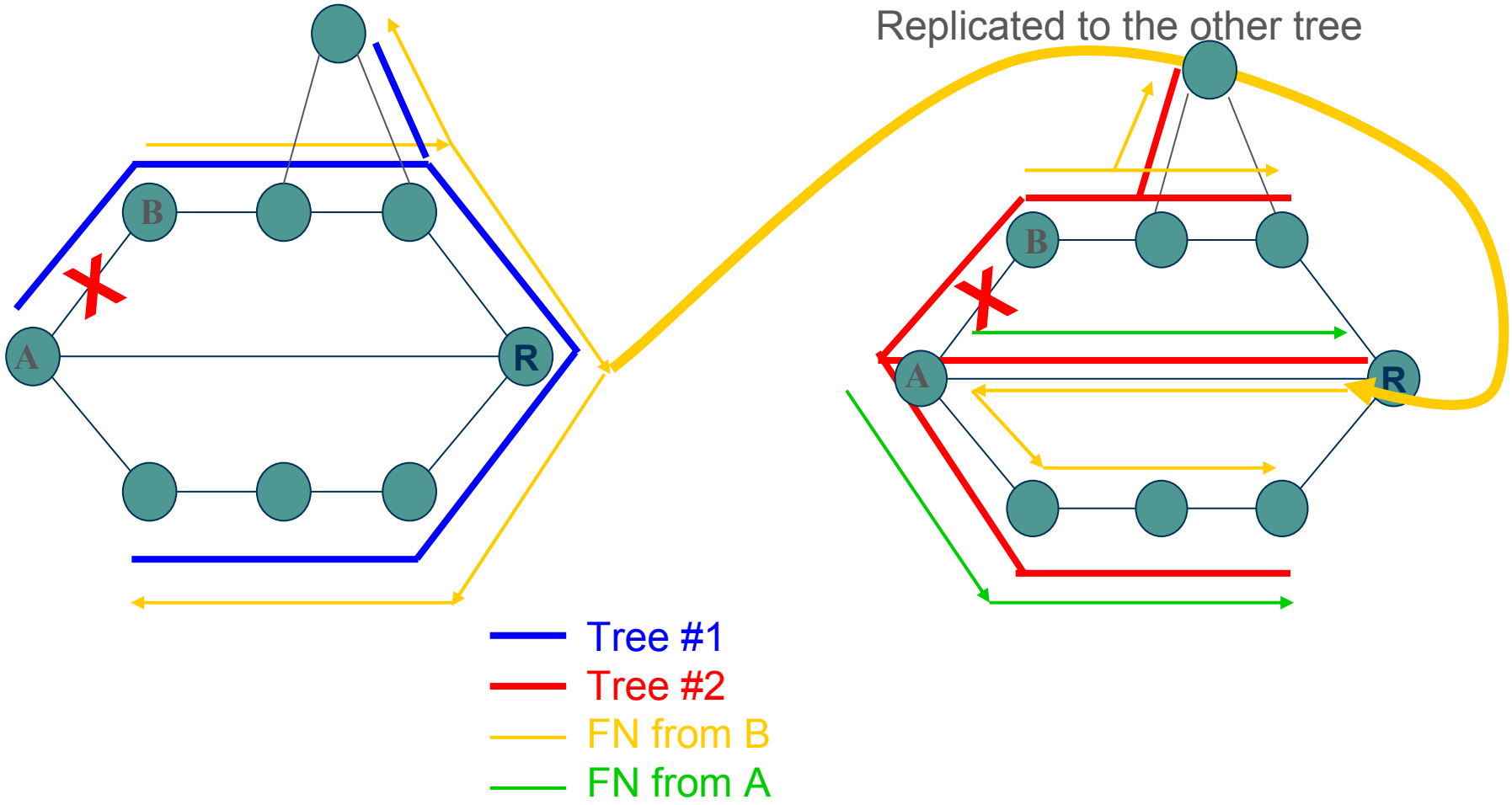
Pair of Redundant Trees in working



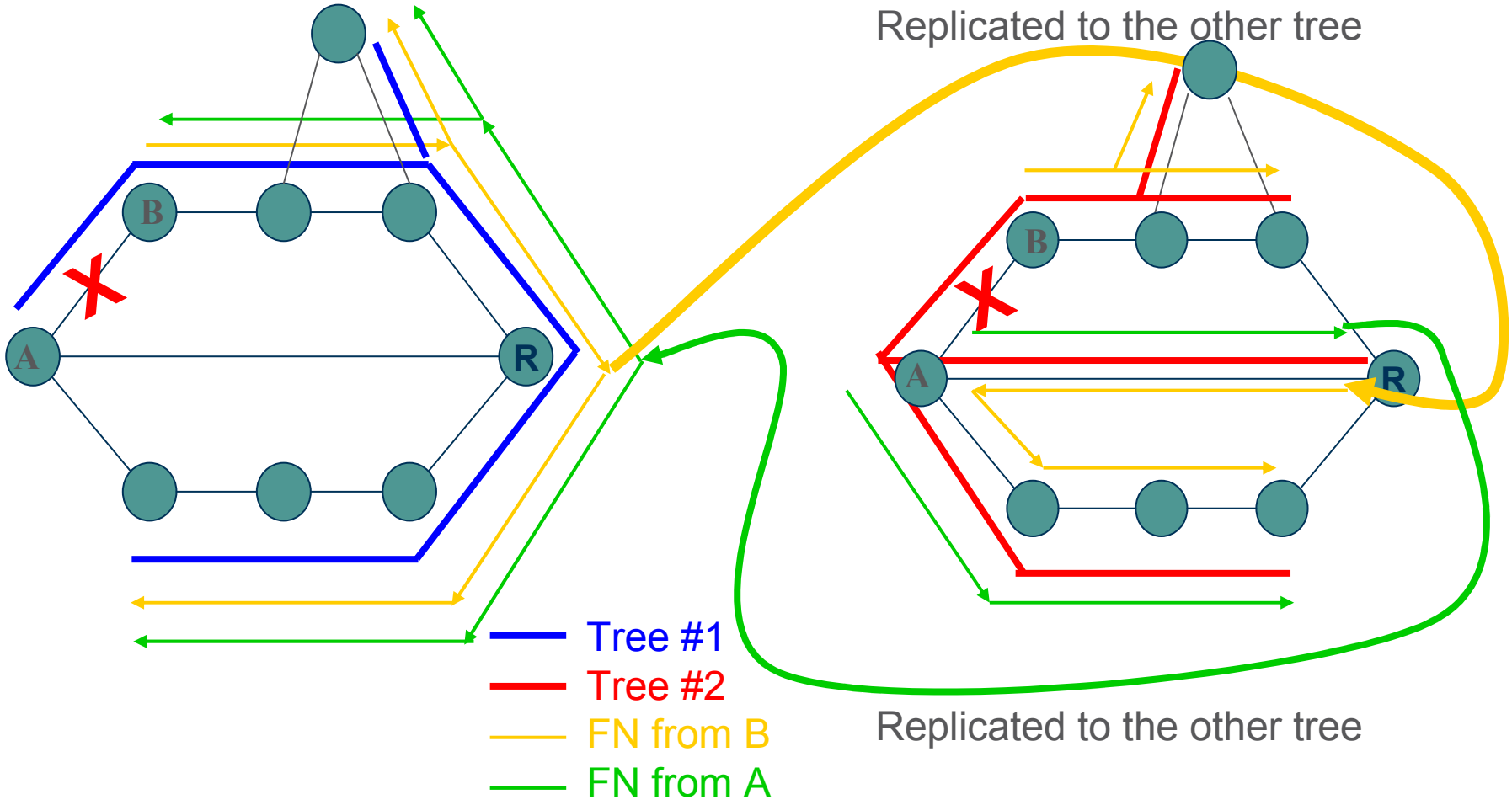
Pair of Redundant Trees in working



Pair of Redundant Trees in working



Pair of Redundant Trees – all done



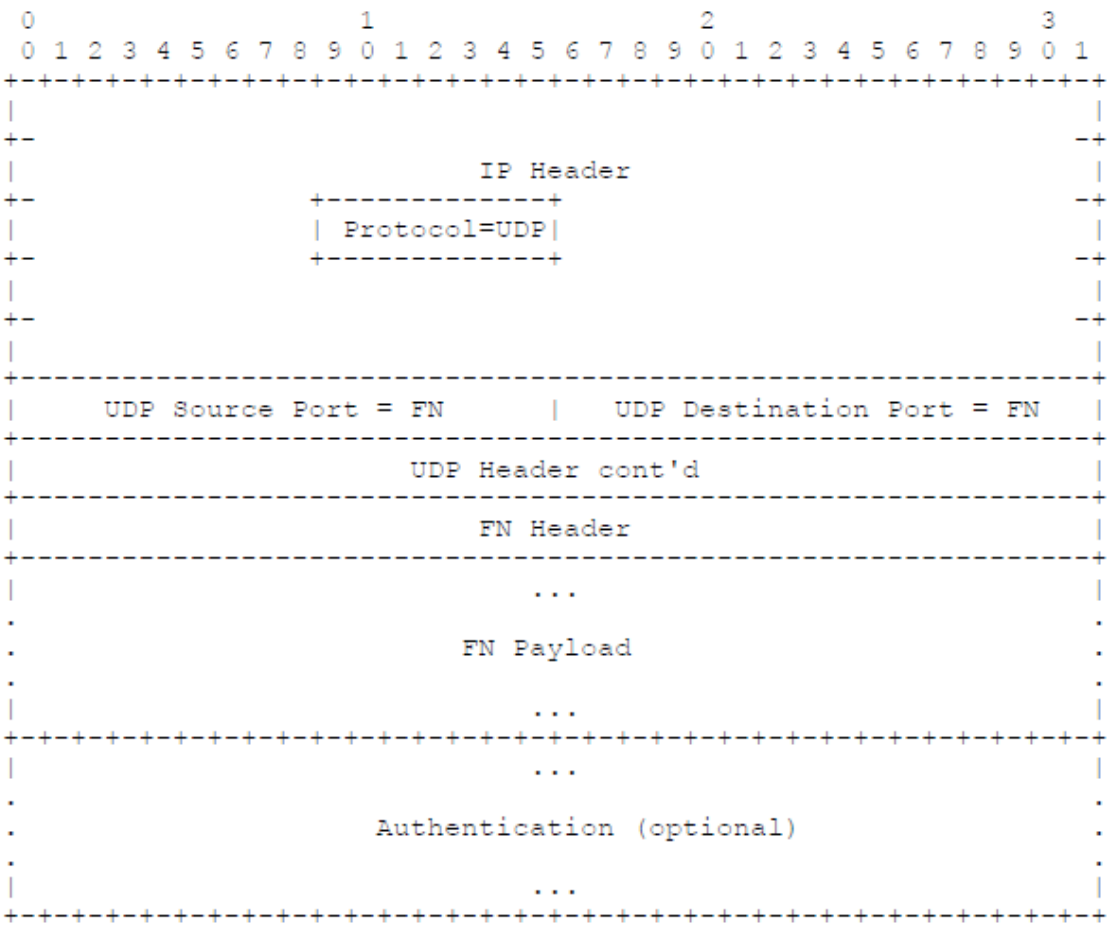
Message Encoding

- **Seamless Encapsulation**
  - An application may define its own message for FN

- **Dedicated FN Message**
  - FN message is to be distributed in UDP with well-known ports (subject to IANA's allocation)

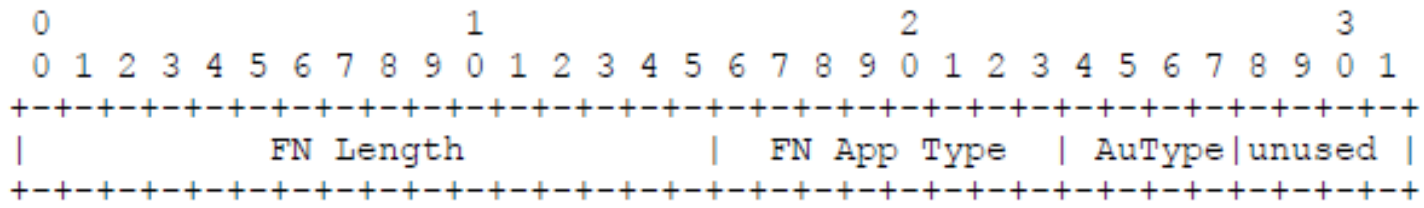
Dedicated FN Message

FN in UDP packet format



FN Header

Encoding of FN Header



FN Length (16 bits)

The length of the FN message in bytes including the FN header and the FN Payload. The authentication data optionally appended to the FN packet is not considered part of the FN message, although it is included in the length field of the packet's IP header.

FN App Type (8 bits)

Identifies the application using FN.
A value for each application needs to be assigned by IANA.

AuType (4 bits)

Identifies the authentication procedure to be used for the packet.

Authentication

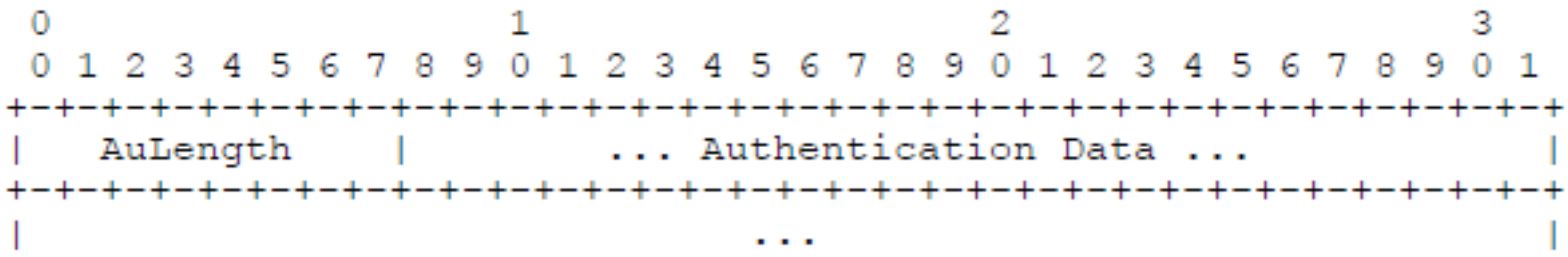
- FN receivers must be able to verify that the packet is sent by an authentic source.
- Options proposed are: simple/MD5/Digital Signatures
- Areas-scoped vs Link-scoped Authentication
  - Since FN is area scooped we propose per-area authentication: a common password, common pre-shared key or digital signatures
  - Operator, however, may prefer per-link authentication, to be explored how to combine with low latency FW plane processing

Authentication

If AuType is set to 0x0, then the FN packet is not carrying an Authentication field at the end of the packet. Note that even in this case FN application may still use its own authentication mechanism in the payload .

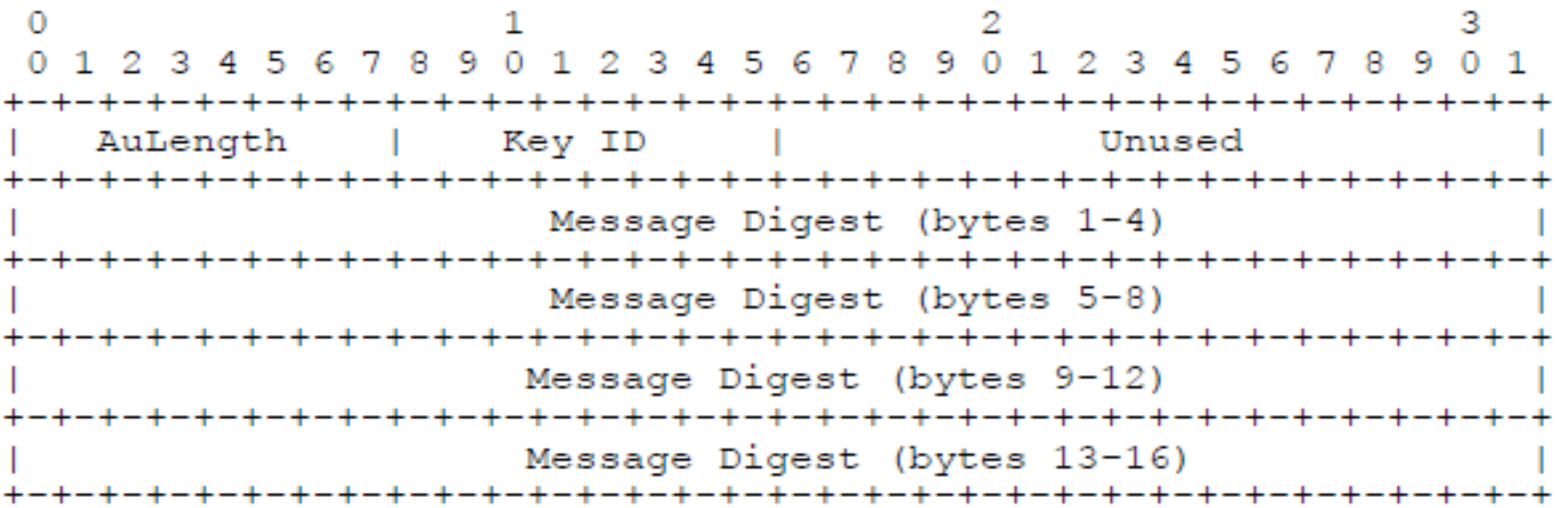
If AuType is non null, an Authentication field must be appended after the FN message.

Authentication field in FN packet



MD5

Authentication field in FN packets with MD5



AuLength

AuLength is set to 20 bytes.

Key ID

This field identifies the algorithm and secret key used to create the message digest appended to the FN packet. This field allows to co-exist multiple pre-shared keys in parallel.

Message Digest

The 16 byte long MD5 hash performed on an object which is the concatenation of the FN message, including the FN header, and the pre-shared secret key identified by Key ID.

Digital Signatures

- Another option is to use public key cryptography to digitally sign the notification to provide certification of authenticity.

- This authentication mechanism is as per OSPF authentication mechanism defined in [RFC2154].

Thanks!

We will take questions after András's presentation