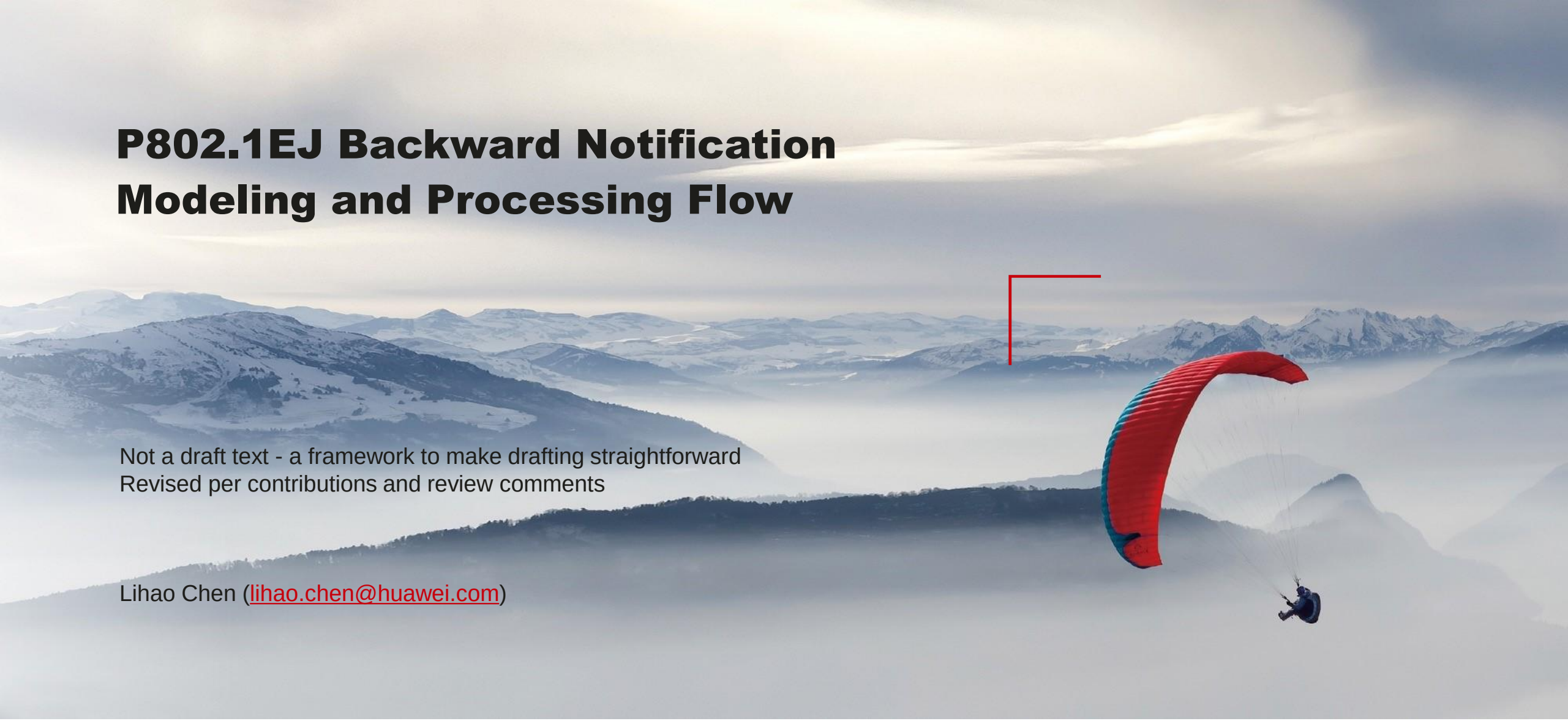
P802.1EJ Backward Notification Modeling and Processing Flow



Not a draft text - a framework to make drafting straightforward
Revised per contributions and review comments

Lihao Chen (lihao.chen@huawei.com)

Links

- Individual text contribution:
  - https://www.ieee802.org/1/files/public/docs2026/ej-chen-individual-text-0826-v00.pdf

- PAR and CSD, and other previous contributions (BN characteristics, taxonomy, use cases, and industry practices) can be found at:
  - https://1.ieee802.org/tsn/802-1ej/

Notes

- The intention is to use these slides as a framework to make later drafting straightforward.
- Welcome to comment on and contribute to the content in these slides, or any parts that may be missing.
- These slides will be continuously updated, and the parts that reach consensus will be gradually written into the draft standard.
- The current structure of the individual draft is shown on the right.

Reference: https://www.ieee802.org/1/files/public/docs2026/ej-chen-individual-text-0826-v01.pdf

Contents

1. Overview
   - 1.1 Scope
   - 1.2 Introduction
   - 1.3 Reference conventions

2. Normative references

3. Definitions

4. Abbreviations and acronyms

5. Conformance
   - 5.1 Requirements terminology
   - 5.2 Protocol Conformance Statement (PCS)
   - 5.3 Conformant systems, components, and functionality
   - 5.4 XXX requirements

6. Architecture and Principles
   - 6.1 Condition identification
   - 6.2 Backward Notification triggering
   - 6.3 Backward Notification signaling
   - 6.4 End station BNM processing
   - 6.5 Bridge BNM processing

7. Service Interface

8. Backward Notification
   - 8.1 Bridge component BN aware process
   - 8.2 End station component BN aware process
   - 8.3 Variables controlling operation
   - 8.4 BN procedures
   - 8.5 Encoding of the BNM PDU

9. Backward Notification (BN) Management

10. YANG data model for BN

Annex A (normative) PICS proforma—BN implementations
   - A.1 Introduction
   - A.2 Abbreviations and special symbols
   - A.3 Instructions for completing the PICS proforma
   - A.4 PICS proforma—<BN implementations>

Annex B (informative) BN usage guidelines and comparison with other mechanisms

Annex C (informative) Bibliography

Annex Z (informative) Outstanding issues

Agenda

- Part 1 - Architecture and Model

- Part 2 - Triggering
- Part 3 - Signal Construction
- Part 4 - Forwarding and Receiving

- Part 5 - Management

- Part 6 - Open Questions

Terminology and Notation

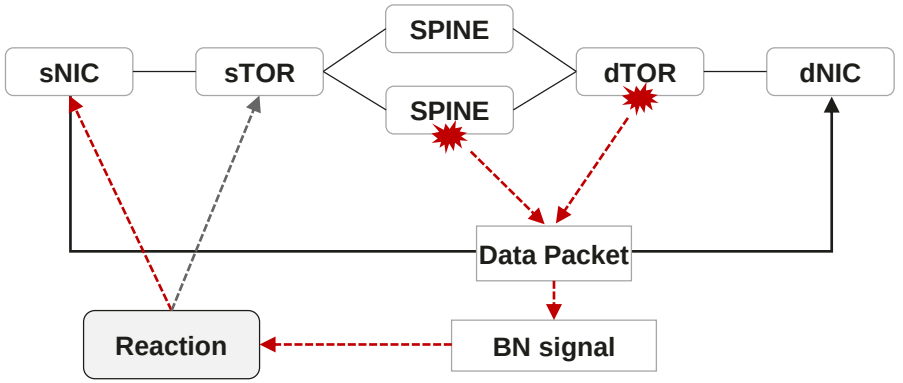
- Reuse terms already defined in IEEE 802 / 802.1Q where possible.
- **BN**: Backward Notification.
- BN System: An End Station or Bridge Component conforming to the BN provisions of this standard.
- BN entity: The protocol entity within a BN system that performs the BN functions. A BN entity may generate BNMs, and/or identify incoming BNMs and expose the extracted information to entities above the data link layer.
- Backward Notification Message (**BNM**): A message transmitted by a BN System to signal network conditions toward the source end station.

BN lifecycle

- Scope: … specifies how BNs are triggered and constructed, and a service interface for exposing the information to entities above the data link layer.

5 Expose

Notifications are exposed to entities above the data link layer (UEC PCM, RoCE CC, load balancing, etc.) through a service interface.



1 Trigger

Any bridge configured to detect a qualified condition — queue/buffer threshold, delay, ECN, drop, trim, link failure.
Throttling mechanism.

4 Receive

Processing upon receipt of a BNM, including notification information extraction.

3 Forward

BNM is forwarded backward following normal L2/L3 forwarding rules, as it carries destination address that is derived from the original data packet's source address. No additional path state required.

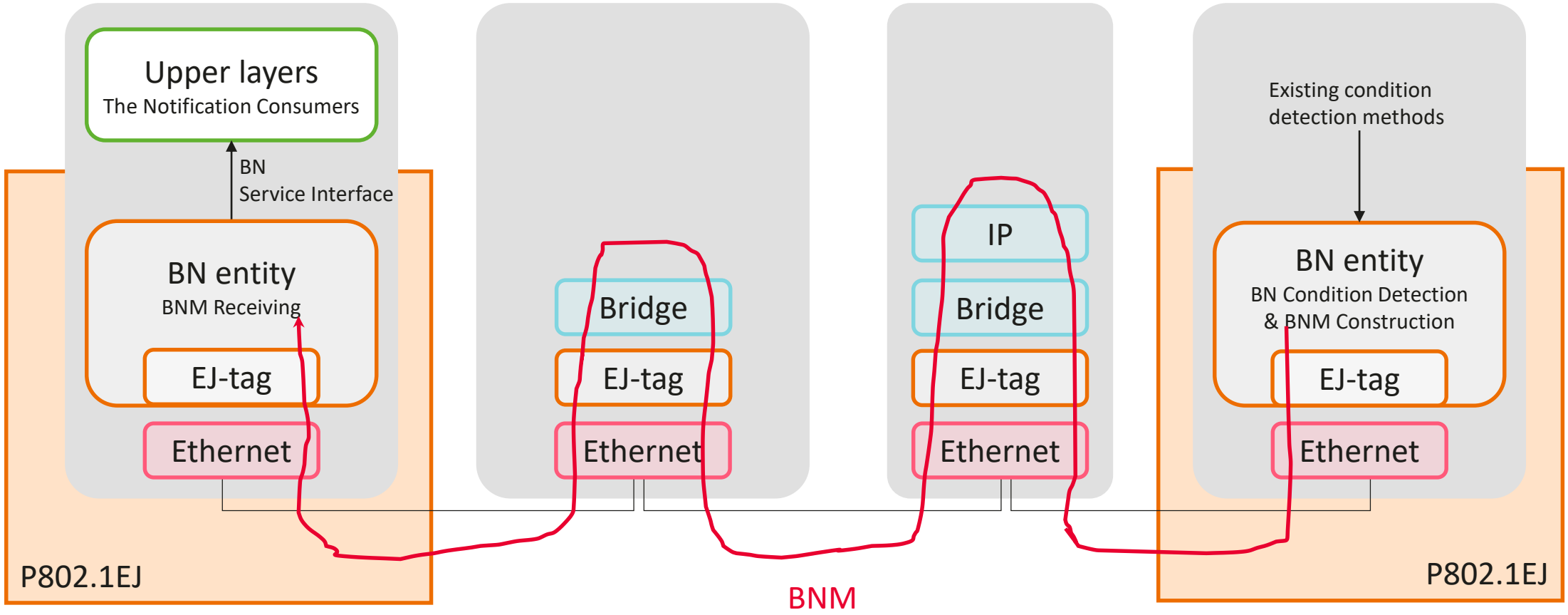
2 Signals

BN message (BNM) itself is derived from the original data packet, carrying signal type and value (network conditions). Additionally, truncated original packet, location, timestamp, etc.

Design Principles

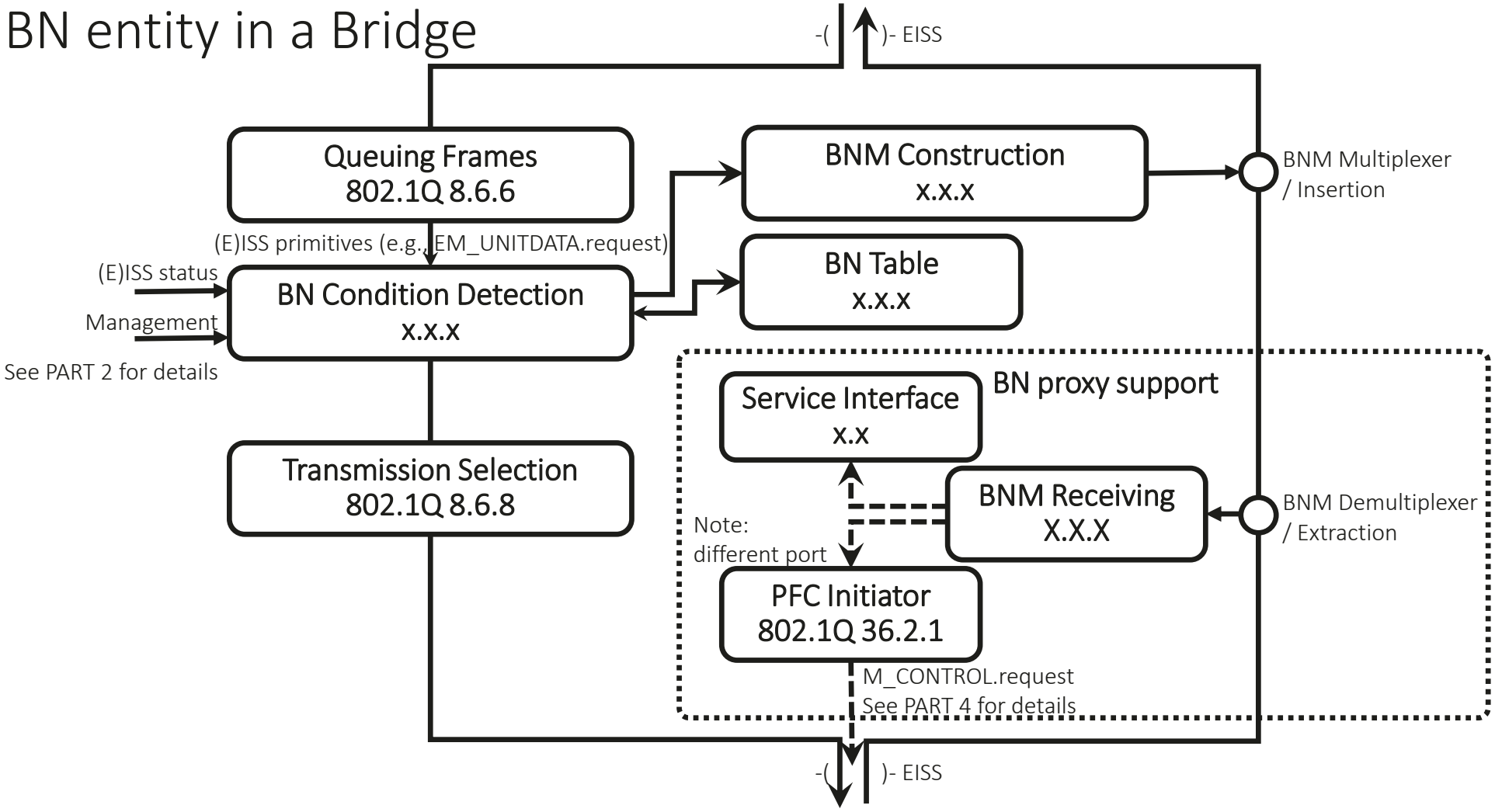
- Low latency.
- Acceptable overhead and complexity.
- No specified reaction on end stations.
- Flexibility to support future needs, e.g., new upper-layer mechanisms and use cases.

BN in end-to-end layered model



- The P802.1EJ covers generation and termination of BNMs, and the service interface.
- The BNM is expected to contain an L2 tag (i.e., the ‘EJ-tag’ in the figure). Details follow in PART 3.

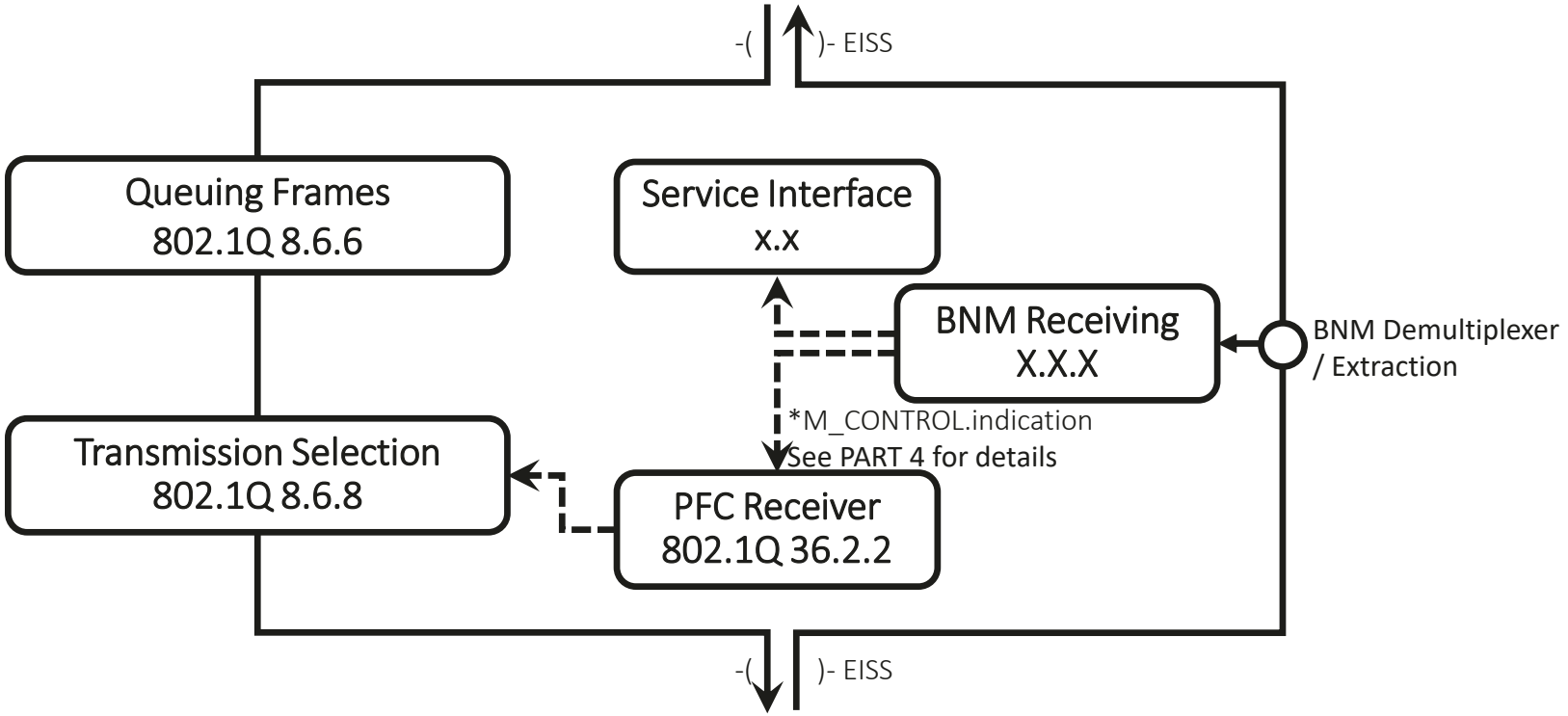
BN entity in a Bridge



- Roles: BNM generation point, relay, and optionally receiver / proxy

Reference: https://www.ieee802.org/1/files/public/docs2022/dw-congdon-individual-text-1122-v01.pdf

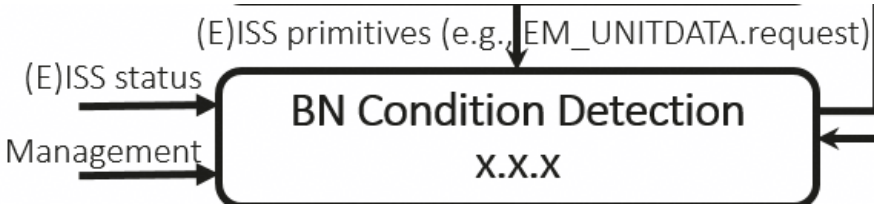
BN entity in an End Station (only ‘receiving’ related functions)



- Roles: BNM receiver; extraction from BNM and exposure to upper layers

Reference: https://www.ieee802.org/1/files/public/docs2022/dw-congdon-individual-text-1122-v01.pdf

Trigger Taxonomy and Principles



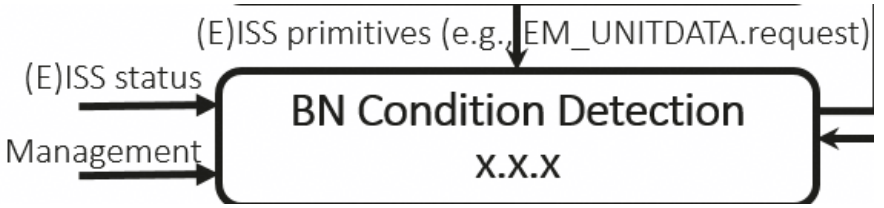
- Specific trigger methods may reuse existing techniques; EJ is not expected to define new trigger methods.
- Three families: state-triggered / event-triggered / request-triggered
  - State triggers - when the threshold of a state (queue depth, queuing delay, available bandwidth, etc.) is exceeded.
  - Event triggers - when a packet event (congestion drop, trim, etc.) happens, or a link state changes (degradation, failure).
  - Request triggers - invoked by management or other request.

Trigger sources	**(E)ISS primitives** *EM_UNITDATA.request* *(per packet)*	**(E)ISS status** *MAC_Enabled* *MAC_Operational*	**Layer management interfaces**
State-triggered	queue depth, delay, ABW, …	—	—
Event-triggered	drop, trim, …	link failure	other state changes
Request-triggered	—	—	implementation specific

Trigger Taxonomy and Principles Cont.

- Trigger Enablement: EJ may support trigger enablement management per-type per-level (e.g., port, queue).
- Triggering point:
  - Triggering before enqueue is faster when possible. However, some triggers do depend on post-dequeue information (e.g., queueing delay) and some switch architecture may expose relevant information after dequeue.
  - Two options: to treat triggering point as implementation specific (preferred), or to explicitly extend the EISS so that a post-dequeue condition can be signaled to the pre-enqueue trigger logic.
- Incoming packet marks and information (not specified by EJ):
  - An incoming packet may carry marks (e.g., BN-eligible, BN-generated) indicating whether the packet is eligible for triggering a BN. A non-BN entity shall determine the eligibility and ensure ineligible packets do not trigger BN.
  - An incoming packet may carry marks indicating that this packet should unconditionally trigger a BN (Request-triggered). A non-BN entity shall recognize this mark, and invoke the layer management interface to trigger the BN.
  - An incoming packet may carry information reflecting values from previous hops along the path. A non-BN entity may use that information as the local state value to determine whether the threshold of a state trigger is reached, and trigger a BN.

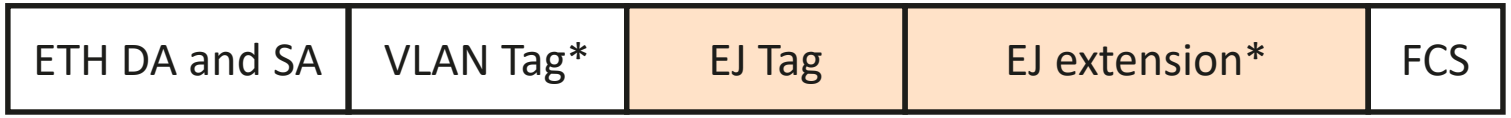
Trigger Suppression: Two Dimensions



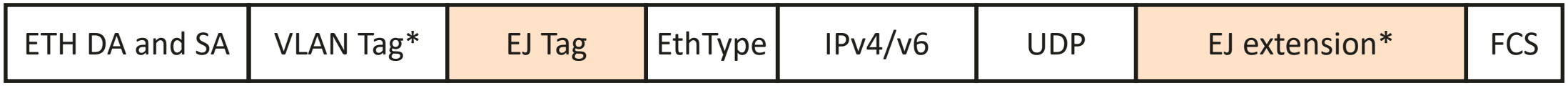
- The BN entity is expected to apply suppression at each trigger input — a packet (EM_UNITDATA.request), a change of the (E)ISS status, or a management request — so that the volume of BNMs stays acceptable and free of redundancy.
- **Multi-Condition Arbitration**: selecting which condition is reported when several apply.
  - A more severe condition may take precedence over less severe ones. For example, once a link failure is causing mass drops, per-packet drop events need not be reported.
- **Throttling**: limiting how often BNMs are emitted.
  - Throttling toolbox: rate limiting, probabilistic sampling, coalescing (tracing the trend of severity and/or keeping the most severe value), etc.
- BN is expected to support management of these trigger-suppression-related configurations, e.g., enable/disable, precedence configuration for arbitration.
- The suppression mechanism introduces overhead, and may require state, so there are trade-offs. Further discussion needed and contributions with more details are anticipated.

BNM Encapsulation Options

- Encapsulation options:
  - Layer 2



  - IPv4; IPv6.



- Notes:
  - The main BN information is carried in the EJ tag. Why? See the next page.
  - The (*optional) EJ extension field may carry further information on demand, e.g., the truncated (trimmed) original packet.
  - In L2 encapsulation, the Ethernet DA is the original packet's SA so that the BNM is forwarded toward the source.
  - In IP encapsulation, the IPv4 or IPv6 destination address is likewise swapped from the original packet.
  - The VLAN tag (*if present) precedes the EJ tag.

The ‘EJ Tag’

- Benefits:
  - **Fast path**: An L2 tag sits near the start of the frame, so it can be recognized and acted on without parsing up the stack.
  - **No dependency on higher layers**: An L2 tag requires no upper-layer header to be present or parsed. An additional IP/UDP carriage may be defined for other purposes; it does not change this.
  - **Consistent with related industry activities**: An L2 tag — for example, the CSIG tag — provides a clean, compact envelope that can carry telemetry information and BN alike. Full or partial reuse of that envelope is the expected direction.

- NOTE: This presentation focuses on what EJ needs. How EJ & EH can be linked together should become clearer as both develop. Neither should be dependent on the other to operate.

- Open for discussion.

BNM PDU Format

- BNM PDU is expected to include:
  - Signal type: Type of the trigger.
  - Signal value: Quantized or enumerable signal value.
  - Location: Metadata locating the point at which the BN was triggered.
  - Flags: An undefined set that can be used for, e.g., a flag for EJ extension indication (trimming), a reason code, or other uses.

- BNM PDU may include:
  - An extension field, which may carry, e.g., a truncated copy of the original packet (Back-to-Sender trimming).

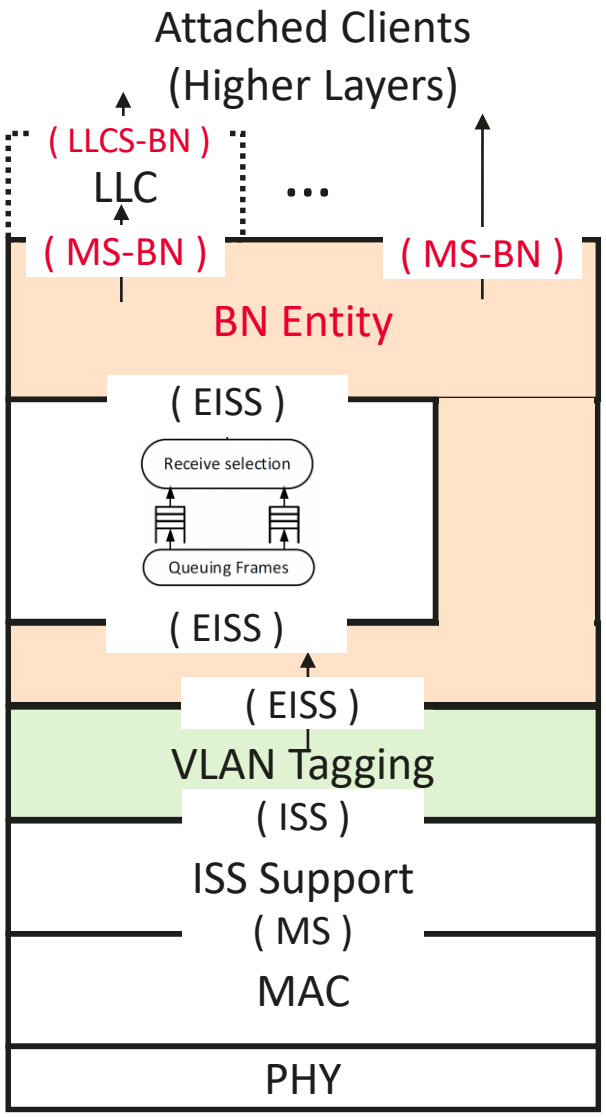
- The signal value calculation is expected to be implementation specific. EJ may provide an informative Annex.

Destination Derivation and Forwarding

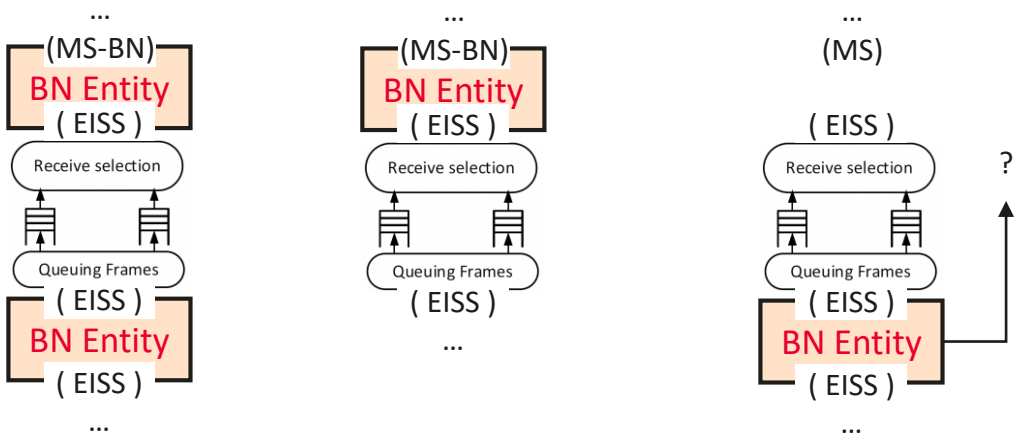
- For a Layer 2 encapsulated BNM, the DMAC is the original packet's SMAC so that the BNM is relayed toward the source.
- For an IPv4/IPv6 encapsulated BNM, the IPv4/IPv6 destination address is the original packet's source address so that the BNM is forwarded toward the source. The MAC addresses are those of the next hop, as determined by normal L3 forwarding.
- The BNM's source address may be configured by management.
- No additional path state required.

- Open question:
  - Is it necessary to support 'multicast' BNMs to multiple sources? - Probably NOT.

End station (BN receiver) stack and service interface

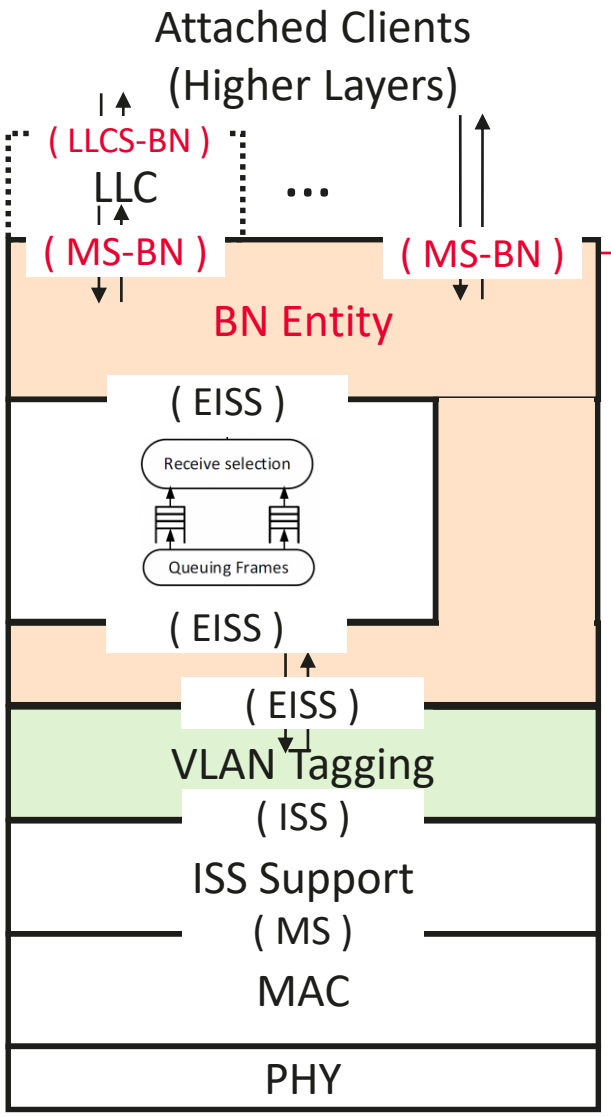


- BN information is extracted by the BN entity.
- The BN entity strips inbound BN tags and passes data in the extended MAC Service (MS-BN) indications to upper layers. See the next page for details.
- Open question: A BNM needs no measurement and can be processed before dequeue, which is faster. Is it appropriate to model it this way (on the left, and 3 more options below)?



Reference: https://www.ieee802.org/1/files/public/docs2026/eh-bottorff-modeling-0926-v05.pdf

MAC Service Backward Notification Extension (MS-BN)



MABN_UNITDATA.indication (
destination_address,
source_address,
mac_service_data_unit, → where EJ extension is delivered
priority,
signal_type,
signal_value,
locator_metadata,
flags
) — From EJ tag

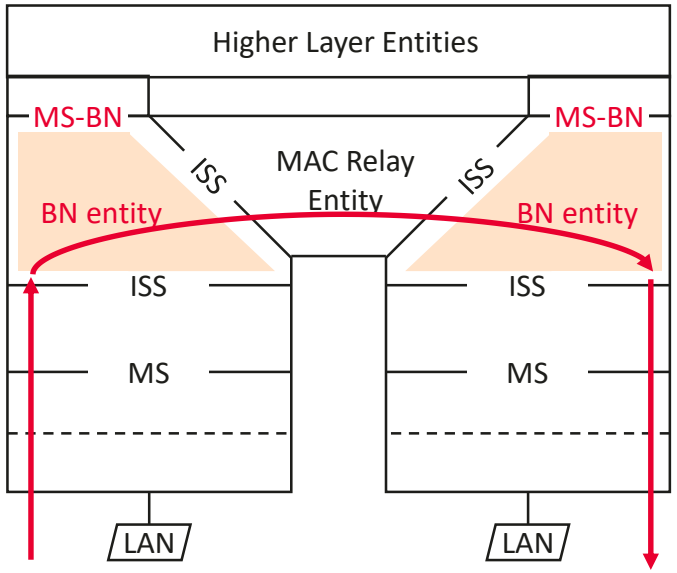
MABN_UNITDATA.request (
…
)

- Similar BN extensions are needed at LLCS.

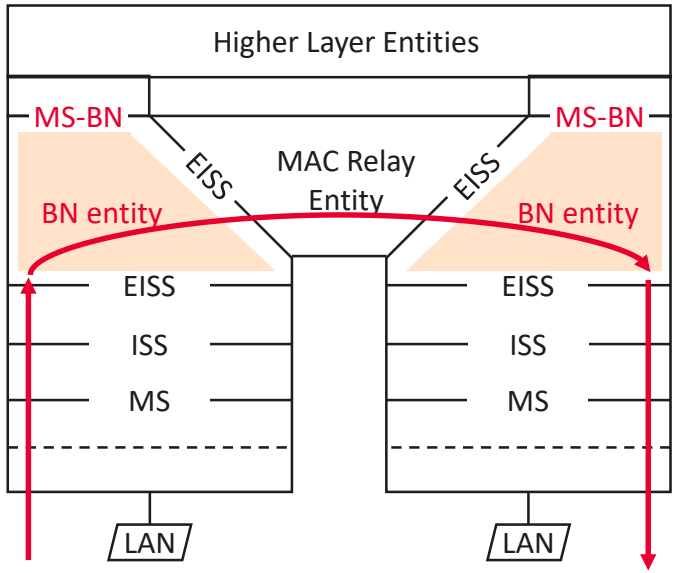
Reference: https://www.ieee802.org/1/files/public/docs2026/eh-bottorff-modeling-0926-v05.pdf

Backward Notification in Bridge models

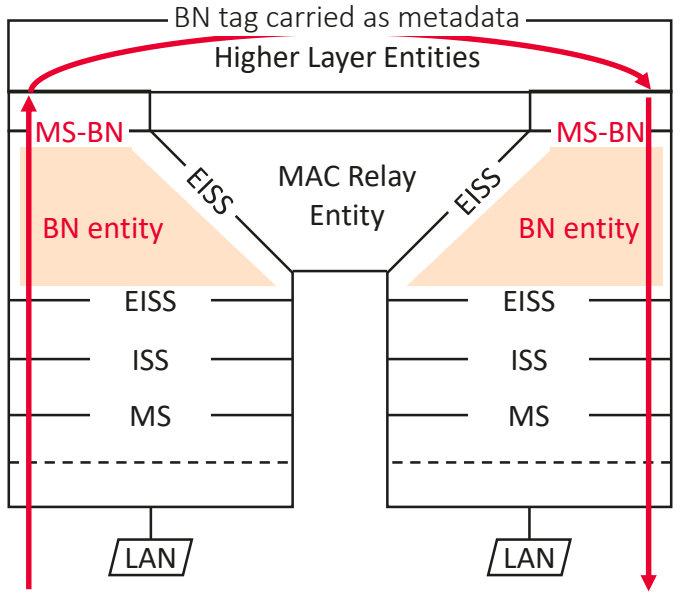
MAC Bridge
ISS BN Shim Model



VLAN Bridge
EISS BN Shim Model



Upper layer forwarding

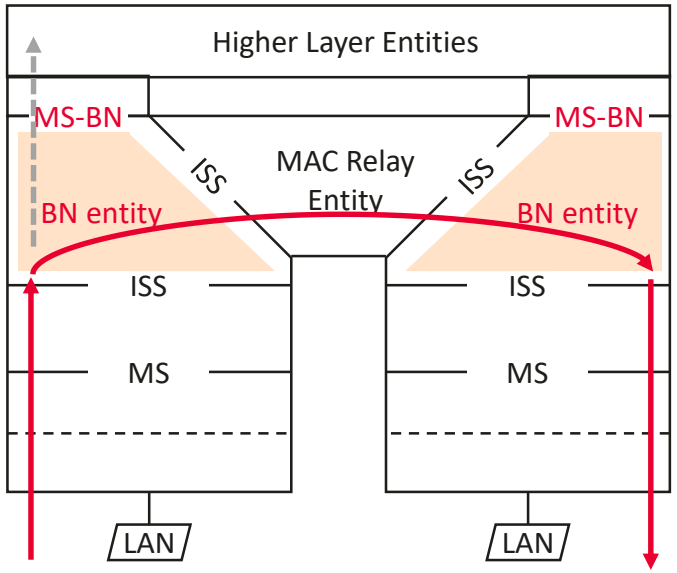


- BN information is carried as data through MAC relay (shim models).
- BN information in BN tags is carried as metadata through MS-BN and upper layer (e.g., IP) forwarding.
- Open question: Since forwarding a BNM requires no BN-specific action, is a simpler model available, or is this model needed at all?

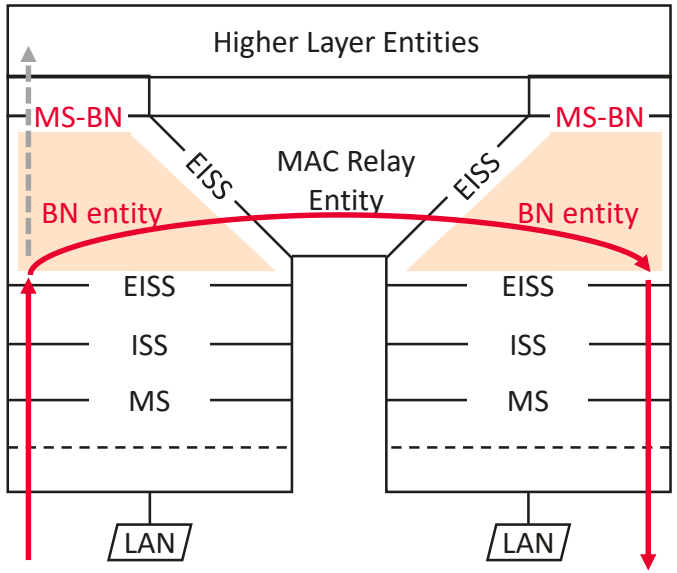
Reference: https://www.ieee802.org/1/files/public/docs2026/eh-bottorff-modeling-0926-v05.pdf

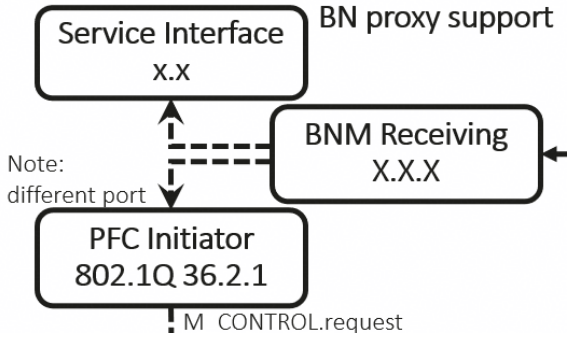
Behavior at Intermediate Bridges

MAC Bridge
ISS BN Shim Model



VLAN Bridge
EISS BN Shim Model





- BNMs are relayed as ordinary frames at intermediate bridges.
- The MS-BN naturally supports a bridge to act as a receiver of BN information and expose it to upper layers. When this happens, the bridge may decide whether to still forward the BNM or not.
- The subsequent actions after BNM receiving are not within the EJ scope. However, if that action is related to Layer 2 control (e.g., PFC), a control path within Layer 2 can be used.

Configurable Items and Hierarchy

- Configurable items, from coarse to fine: BN system level -> port level -> queue level -> flow level
  - The configuration level for each item will be defined based on rationality.

- Variables that control BN operation (read and write):
  - BN function enablement;
  - Trigger enablement;
  - Trigger suppression;
  - Bridge roles

  Note: EJ is not expected to define threshold (of the triggers) configuration.

  Example: bnmMinInterval
- State (read-only) variables:
  - TBD

- These points will be addressed in more detail in follow-up contributions.

YANG Data Model

- Content to be defined.

Open Questions

- Consolidate all open items, ordered by importance (whether they block drafting):
  - (Will be added if the open questions raised in these slides don’t reach consensus after the first round discussion, and if new questions are raised.)

Discuss