Edge TLV Transport Protocol (ETTP) Proposed ETTPDU and State Machines

Definitions, Semantics and State Machines

802.1Qbg Presentation

March 15, 2010

Contributing Authors

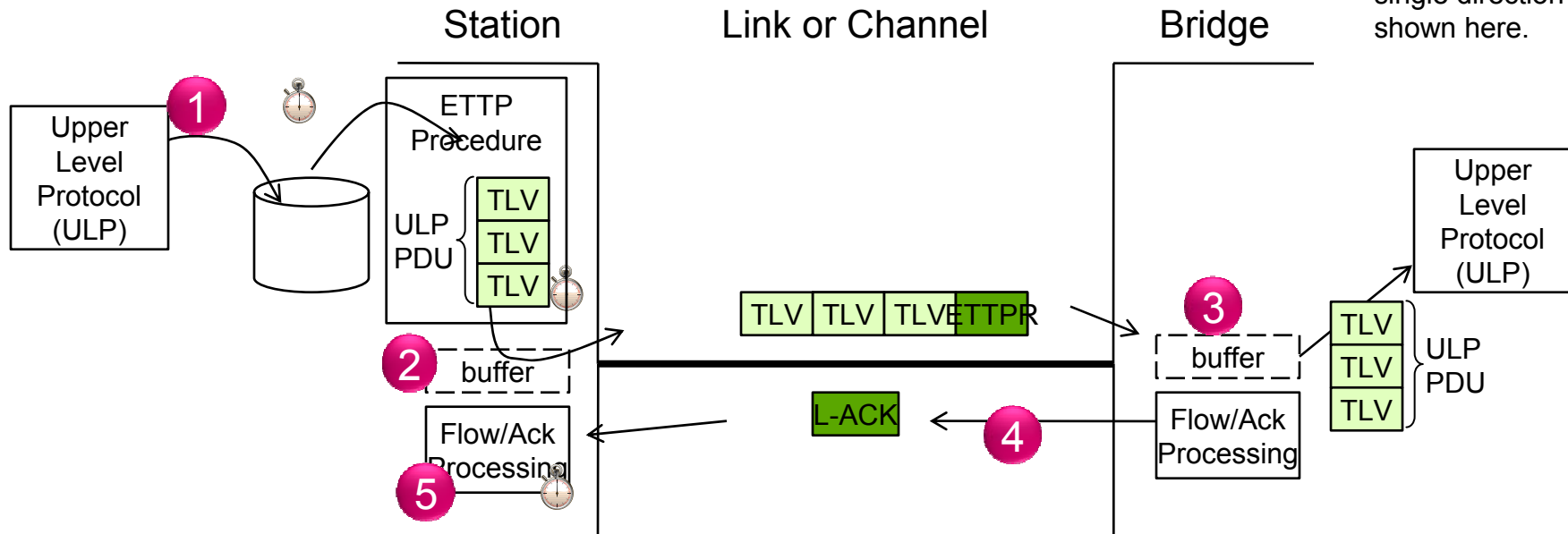| Company	Contacts
BNT	Daya Kamath
BNT	Jay Kidambi
BNT	Vijoy Pandey
Broadcom	Uri Elzur
Brocade	Anoop Ghanwani
Chelsio	Asgeir Eiriksson
Emulex	Chait Tumuluri
HP	Paul Bottroff
HP	Paul Congdon
HP	Chuck Hudson
HP	Michael Krause
IBM	Vivek Kashyap
IBM	Renato Recio
IBM	Rakesh Sharma
Juniper	Srikanth Kilaru
QLogic	Manoj Wadekar

Edge TLV Transport Protocol (ETTP) Goals
(Update to 11/09 "bg-hudson-tlvtransport-1109-v02.pdf" presentation)

- What should stay the same
  - TLV format (but allow multiple of the same TLV type)
  - Addressing
  - Strong push to minimize traffic
- Very simple
- Meets needs of VSI Discovery Protocol
- Minimizes traffic for new protocols
- Easy to learn (re-uses LLDP knowledge)
- Could carry other client protocols
- The intent is to have the server's virtualization infrastructure (e.g. Hypervisor) implement ETTP, versus having the NIC implement ETTP.

Edge TLV Transport Protocol (ETTP)
Lower-level Transport Overview

ETTP has symmetric behavior, but only a single direction shown here.

Station **Link or Channel** **Bridge**

Upper Level Protocol (ULP)

ETTP Procedure

ULP PDU

TLV
TLV
TLV

TLV | TLV | TLV | ETTPR

buffer

Flow/Ack Processing

L-ACK

buffer

Flow/Ack Processing

TLV
TLV
TLV

ULP PDU

Upper Level Protocol (ULP)

1 ULP protocols pass outgoing ULP PDU to lower-level transport through a ETTP procedure.

2 Frame with ULP PDU (i.e. a set of ULP TLVs) is transmitted but the frame is not yet deleted from the transmit buffer. A L-ACK timer is set.

3 Arriving frame is received into a receive 'buffer', where it is held until it is removed by ETTP procedure that passes the ULP PDU to the associated upper level protocol.

4 When the receive buffer is emptied, a low-level acknowledge (L-ACK) is sent to the sender.

5 If the L-ACK is received before the timer expires, then the transmit buffer is cleared and ETTP can process another ULP PDU through the ETTP procedure.

If the L-ACK timer expires before the L-ACK is received, then the frame in the transmit buffer is resent (some preset number of times).

ETTP PDU Overview

Ethertype = TBD	Sub-type	Mode	Sequence Number	ULPDU
← 2 Octets →	← 2 Octets →	1 Octet	← 2 Octets →	Optional

- The destination address of the Ethernet frame that contains a ETTPDU has the following semantics:
  - Nearest bridge (01-80-C2-00-00-0E)
  - Nearest Customer Bridge (01-80-C2-00-00-00)
- The source address shall be the sending station or port individual MAC address.
- A new Ethertype will be needed for ETTP.
- If ETTP is performed over Multi-channel,
  then the S-TAG for the channel shall precede the ETTPDU.
- The ETTPDU contains:
  - Sub-type - defines the ULP type included in the PDU. Note for Ack's the sub-type is ignored at the station.
  - Mode – Identifies whether the operation is a:
    - ETTP request (0x00)
    - ETTP acknowledgement (0x01). Sequence number – identifies the sequential order of the PDU, with respect to other ETTPDUs. The starting sequence number may start anywhere for the first ETTPDU, but the sequence number for each subsequent new ETTPDU is incremented by 1.

*Note: MCH LLDP exchanges will run at the physical link;
EVB LLDP exchange and ETTP+VSI exchange will run at the link if the link is not configured for multichannel, channel if it is.
This needs to be documented in MCH semantics.

ETTP Semantics: <ULP, ETTP> interface

- A single link operating in "multi-channel" mode has one ETTP per channel.

- For VSI, there is one VSI agent per channel ETTP and that agent may have multiple VSI instances sharing a single channel.

- The <ULP, ETTP> interface is based on a complete ULPDU (i.e. the group of TLVs that are handed to ETTP for transmission).
  - The number of octets in the ULPDU may be less than the maximum number of octets that can fit into a ETTP frame.
  - The number of ULP TLVs may be less than the maximum number that can fit into a ETTP frame.
  - Procedures are used to describe how the ULP hands off ULP PDUs to ETTP and how ETTP hands off ULP PDUs to the ULP.
  - This, version 0 proposal, is silent on how multiple ULPs arbitrate when sharing the same ETTP.
  - Given the <ULP, ETTP> interface is based on full ULPDUs, no immediate processing is needed at the ETTP level.
  - Link down semantics outside the scope of this specification.

ETTP Semantics: ETTP reliability

- ETTP acknowledgement.
  - A ETTP acknowledge means the ETTPDU was received and there is a free buffer available to enable another send.
    - It doesn't mean the ETTPDU were delivered to the ULP.
  - At the transmit side, if a ETTP Acknowledge is not received within an Ack timer period, ETTP will retry the ETTPDU up to an EVB negotiated Retry Count is reached.
    - The value of the EVB negotiated Retry Count is on a per link basis not channel.
- Once the receive side ETTP delivers the ULP PDU to the ULP through the receive side hand-off procedure, the ETTP buffer becomes available for another send.
  - The Acknowledgement must be sent in a separate ETTPDU (vs piggy backing onto a Transmit message in the opposite direction).
  - The receive side will issue a ETTP Acknowledgement after the completing the receive side hand-off procedure.
  - If the receive side hand-off procedure takes too long, the receive side ETTP may toss the ULP PDU and send back an ACK to indicate the ETTP buffer is free on the receive side.
    - Note: The length of time the receive side waits, before tossing the ULPDU should not be less than the retransmission period times the maximum number of retries.
    - This version 0 proposal is silent regarding raising a flag due to the slowness of the receive side hand-off procedure.
- Sequence numbering must be used to detect duplicate vs new ETTPDUs.
- ETTP will not provide a keep-alive mechanism. Instead each ULP must do so.
- Specification is silent on having a digest provided at the ETTP level.
  - Any TLV database synchronization is left up to the ULP.

Transmit and Receive - <ULP, ETTP> Procedures

ETTP_UNITDATA.request (ulptype,
ulpdu)

The ETTP_UNITDATA.request is invoked by the ULP at the sender to notify
ETTP that a ULPDU is ready to be transmitted.
The ulpdu parameter is unit of work from the ULP. For example, for VSI it
consists of a set of VSI TLVs passed from the VSI ULP to ETTP for
transmission, where the set of TLVs must be less than or equal to the
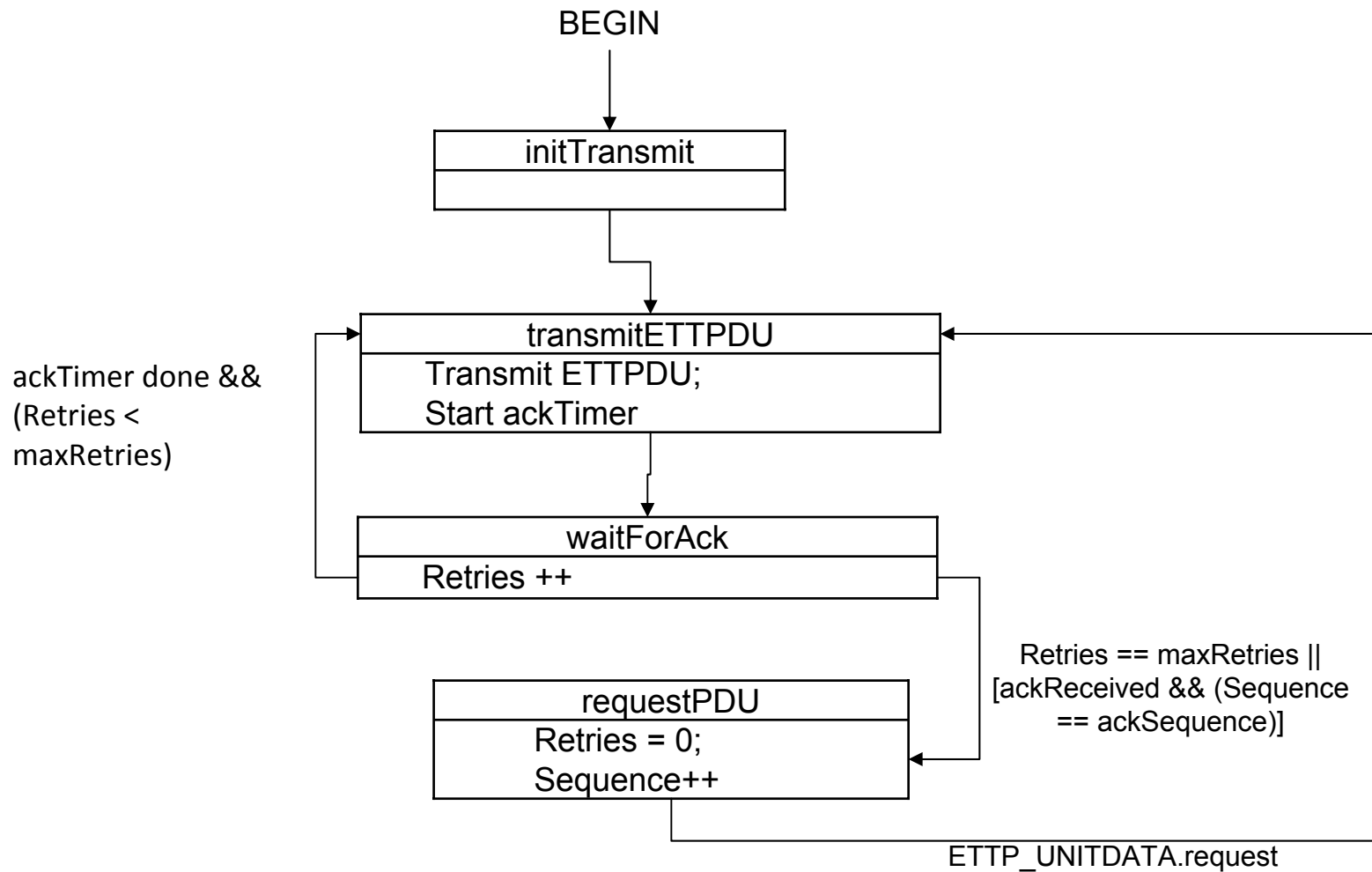maximum allowed ETTPDU.

ETTP_UNITDATA.indication (ulptype,
ulpdu)

The ETTP_UNITDATA.indication is invoked by ETTP at the receiver to
indicate a ULPDU has been successfully received and is available ULP
processing.
The ulpdu parameter is unit of work from the ULP. For example, for VSI it
consists of a set of VSI TLVs passed from the ULP to ETTP for
transmission, where the set of TLVs must be less than or equal to the
maximum allowed ETTPDU.

Note: the <ULP, ETTP> interface procedure above may be implemented in many ways, including a queue.
Also, the system must have a way of associating the ulptype with a specific ULP.

Transmit Side
Transmit State Machine

BEGIN

```
initTransmit
```

transmitETTPDU
Transmit ETTPDU;
Start ackTimer

ackTimer done &&
(Retries <
maxRetries)

waitForAck
Retries ++

Retries == maxRetries ||
[ackReceived && (Sequence
== ackSequence)]

requestPDU
Retries = 0;
Sequence++

ETTP_UNITDATA.request

Note: first entrance into transmit ETTPDU is used to initiate the sequence counting on the receive side.
The starting sequence number may start anywhere for the first ETTPDU. A linkdown event may
restart the sequence number at the same point every time or not.

Receive Side
Enqueue TLV State Machine

BEGIN

initReceive
lastSequence = NULL

receiveWait

ETTPDU received

receiveETTPDU
If (validate(ETTPDU))
    seqETTPDU = sequenceOf(ETTPDU);

seqETTPDU != lastSequence

seqETTPDU == lastSequence

resendACK
sendAcknowledgement(seqETTPDU)

sendACK
Invoke ETTP_UNITDATA.indication procedure

lastSequence = seqETTPDU;