VSI Discovery and Configuration – Working Draft

Definitions, Semantics and State Machines

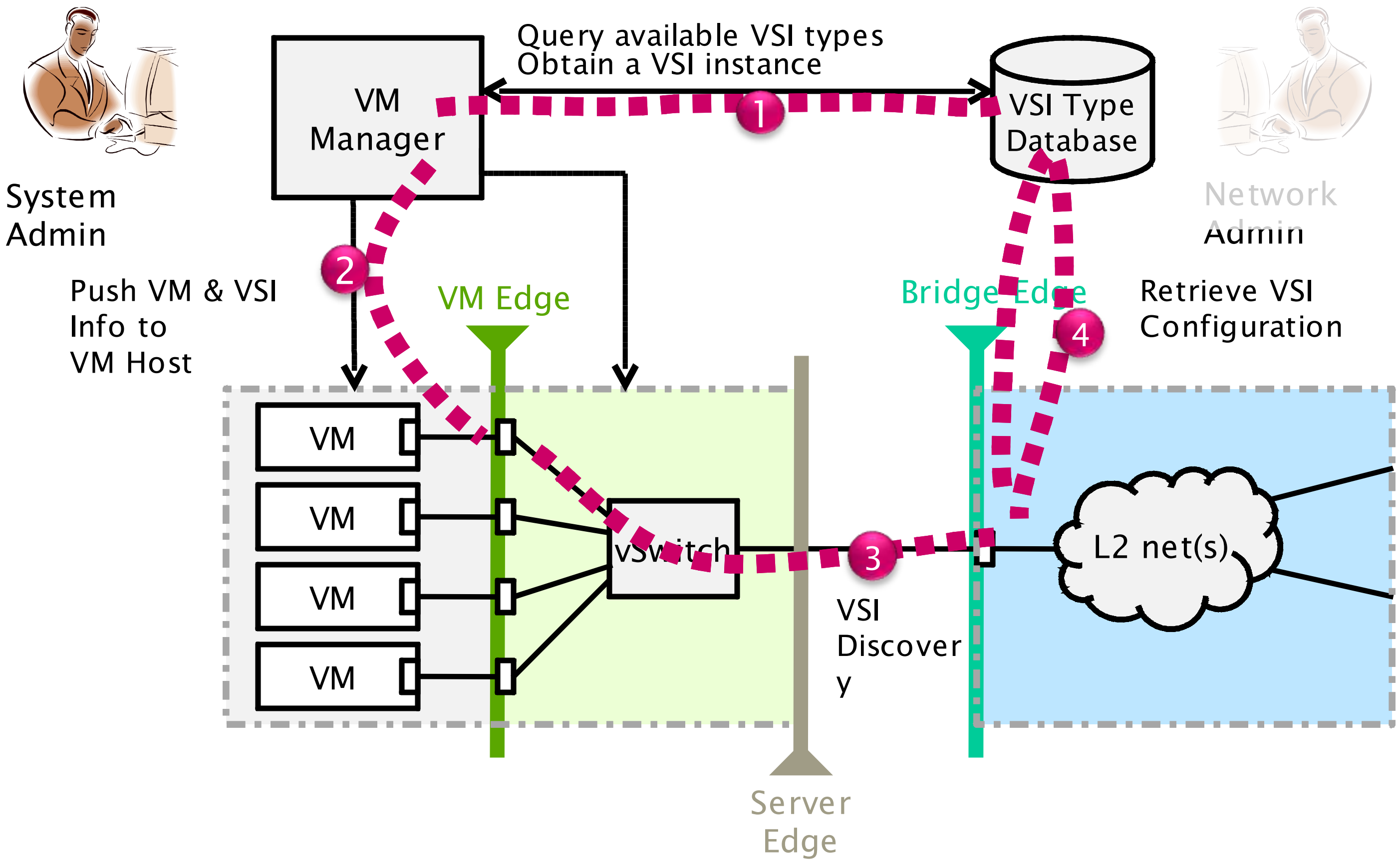
802.1Qbg Presentation

Version 00
03/16/2010

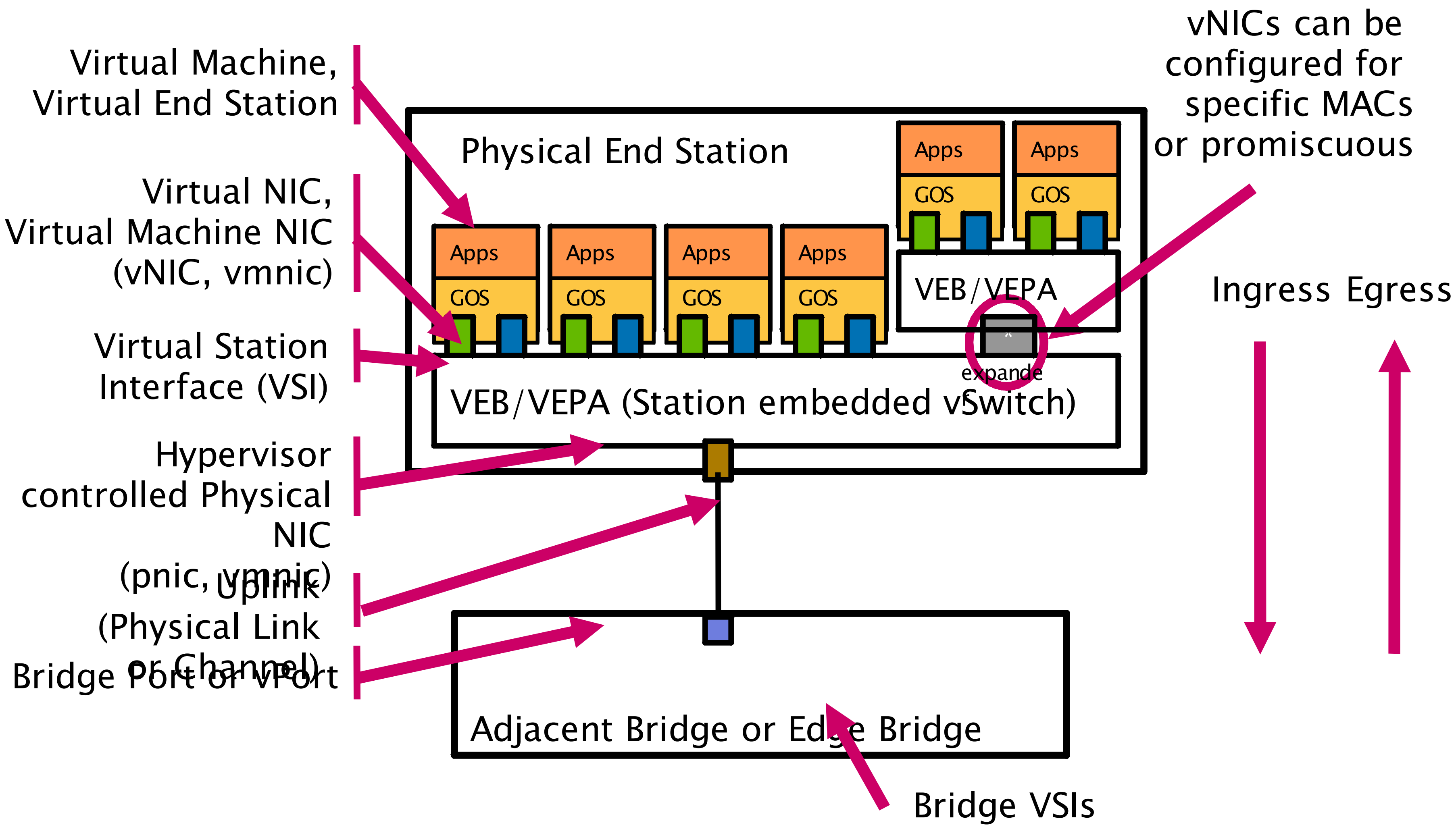
Contributing Authors

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

One Scenario for Configuring Edge Connections (VSIs)



Basic VEB/VEPA Anatomy and Terms



Virtual Machine,
Virtual End Station

Virtual NIC,
Virtual Machine NIC
(vNIC, vmnic)

Virtual Station
Interface (VSI)

Hypervisor
controlled Physical
NIC
(pnic, vmnic)

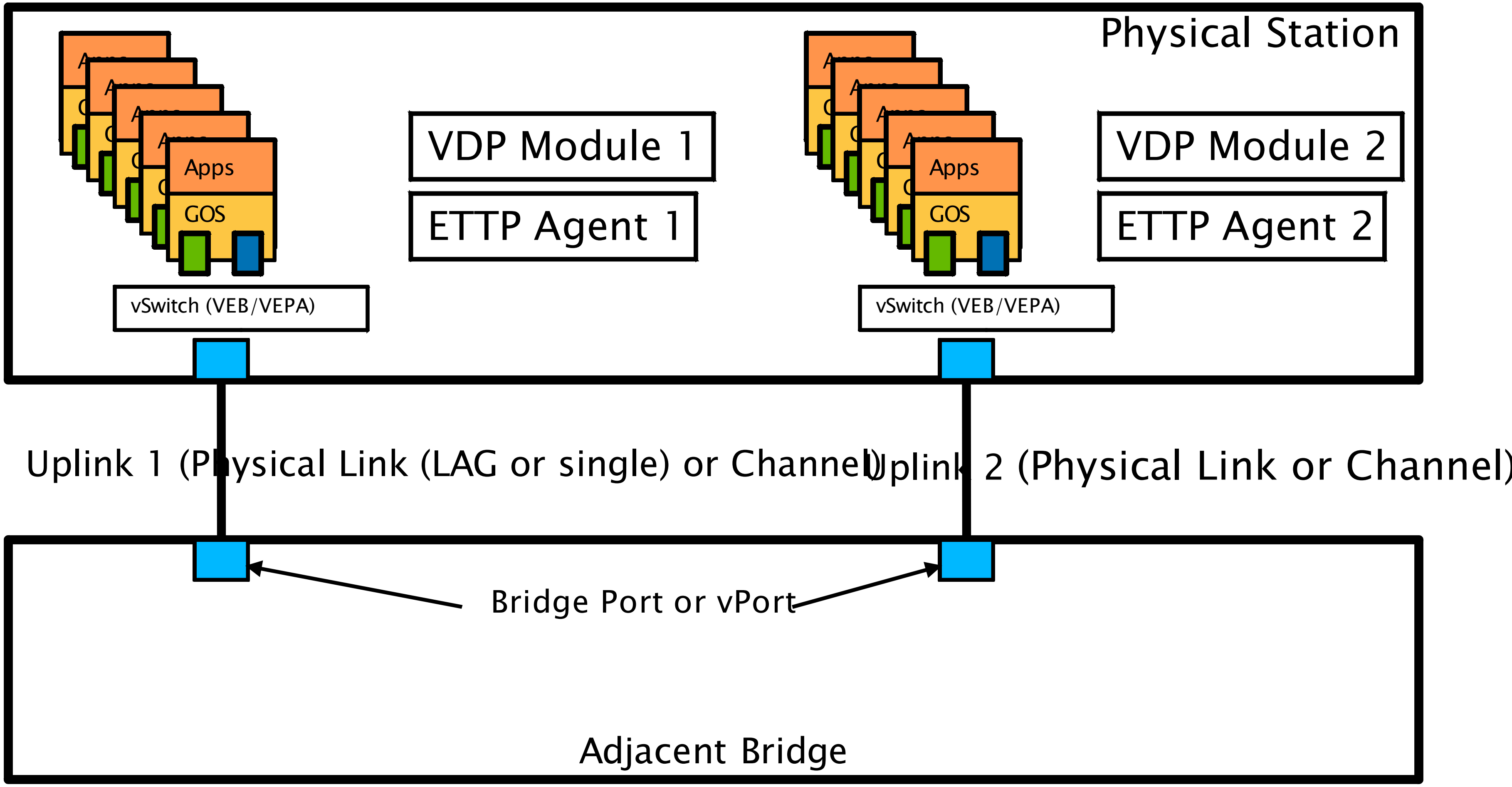
Uplink
(Physical Link
or Channel)

Bridge Port or vPort

vNICs can be
configured for
specific MACs
or promiscuous

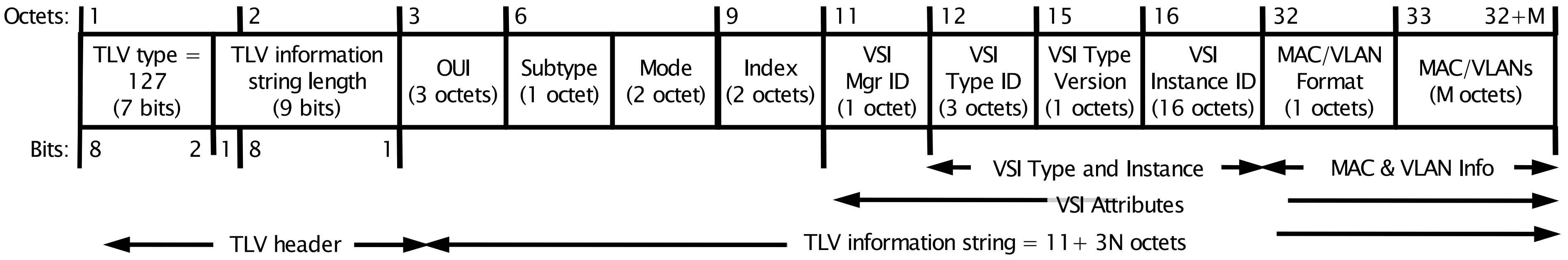
Bridge VSIs

VSI and T3P-R Components Example



Proposed VSI Discover/Configuration TLV

Transport: ETTP



Octets:	1	2	3	6		9	11	12	15	16	32	33 32+M
	TLV type = 127 (7 bits)	TLV information string length (9 bits)	OUI (3 octets)	Subtype (1 octet)	Mode (2 octet)	Index (2 octets)	VSI Mgr ID (1 octet)	VSI Type ID (3 octets)	VSI Type Version (1 octets)	VSI Instance ID (16 octets)	MAC/VLAN Format (1 octets)	MAC/VLANs (M octets)
Bits:	8 2	1 8 1										

VSI Type and Instance

MAC & VLAN Info

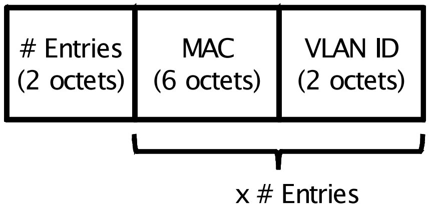
VSI Attributes

TLV header

TLV information string = 11+ 3N octets

- Index – VSI index – Offset in bit-arrays containing state and configuration status of VSIs.
- Mode – Indicates VSI TLV Mode
  - First octet identifies a pre-associate, associate, de-associate, or the corresponding confirmation or rejection for each.
  - Second octet is used during a rejection to indicate the reason for the pre-assoc or assoc rejection.
- VSI Manager ID – Identifies the VSI Manager with the Database that holds the detailed VSI type and or instance definitions. VSI Manager ID can be used to obtain IP address and/or other connectivity and access information for the manager.
- VSI Type ID (VTID)- The integer identifier of the VSI Type.
- VSI Type ID Version – The integer identifier designating the expected/desired version of the VTID.

- VSI Instance ID – A globally unique ID for the connection instance. The ID shall be done consistent with IETF RFC 4122.
- Format – identifies the format of the MAC and VLAN information that follows in the TLV.
- MAC/VLANs – Listing of the MAC/VLANs associated with the Virtual Station Instance (VSI).
- Following is the format for Format = 1

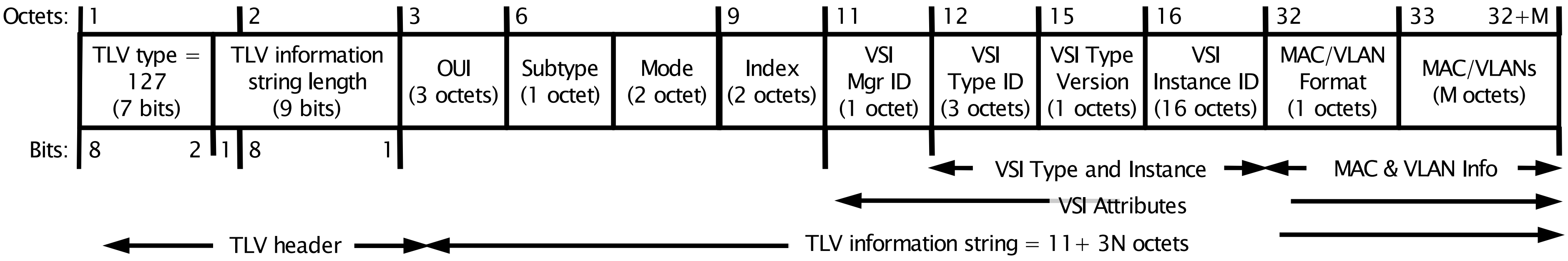


# Entries (2 octets)	MAC (6 octets)	VLAN ID (2 octets)

x # Entries

NOTE: The station and bridge environments and their common understanding of the meaning of a VSI Type ID is outside the scope of this effort.

Notes on VSI TLV



Octets:	1	2	3	6		9	11	12	15	16	32	33 32+M
	TLV type = 127 (7 bits)	TLV information string length (9 bits)	OUI (3 octets)	Subtype (1 octet)	Mode (2 octet)	Index (2 octets)	VSI Mgr ID (1 octet)	VSI Type ID (3 octets)	VSI Type Version (1 octets)	VSI Instance ID (16 octets)	MAC/VLAN Format (1 octets)	MAC/VLANs (M octets)
Bits:	8 2	1 8 1										

VSI Type and Instance

MAC & VLAN Info

VSI Attributes

TLV header

TLV information string = 11+ 3N octets

Normative Notes:

-One or more VSI TLVs are transported in a ETTP PDU.

Informative Notes:

- The station and switch environments and their common understanding of the VTID meaning is outside the scope of this TLV.
-The contents of a VSI Type are outside the scope of this specification. Bridges may use access and traffic controls as part of the contents of a VSI Type.
-LLDP/T3P-R PDUs use Physical Station MAC Address (e.g. Hypervisor MAC).
-LLDP/T3P PDUs carry Chassis ID TLV for the Physical Station (Hypervisor).
- The Physical Station port’s VLAN ID uses the VLAN TLV in the same transport (LLDP or T3P) PDU and is not contained in this TLV.
- Format field - VSI TLV allows multiple formats of this information to optimize frame space usage and functionality. Further, it makes possible extensions in future.

VSI Discovery TLV – Mode and Mode Response

- Mode field purpose: Identifies VSI Discovery TLV type:
  - VSI TLV Request field: 1st octet
    - Pre-Associate: 0x00
    - Pre-Associate with resource reservation: 0x01
    - Associate: 0x02
    - De-Associate: 0x03
  - VSI TLV Response field: 2nd octet
    For all the unsuccessful responses, the bridge reflects the same VSI TLV fields the Requester had sent.
    - Success: 0x00
    - Invalid Format: 0x01
    - Insufficient Resources: 0x02
    - Unused VTID: 0x03
    - VTID Violation: 0x04
    - VTID Version Violation: 0x05
    - Out of Sync: 0x06
    - Reserved 0x07 – 0xFF
- Usage:
  - Used under the control of VDP state machines.

Mode Response Definitions

- Success: 0x00
- Invalid Format: 0x01
  - The VSI Format is not supported by the switch.
- Insufficient Resources: 0x02
  - The switch does not have enough resources to complete the VSI operation successfully.
- Unused VTID: 0x03
  - The VSI referenced by the VSIID does not exist in the VSI Manager database referenced by the VSI Manager Identifier
- VTID Violation: 0x04
  - The VSI referenced by the VSIID is not allowed to be associated with the VTID.
- VTID Version Violation: 0x05
  - The VSI referenced by the VSIID is not allowed to be associated with the VTID Version.
- Out of Sync: 0x06
  - The VTID or one of the VSI List fields used in the Associate is not the same as the corresponding field used in the Pre-Associate.

Mode Requests and Responses

- Requests:
  1. Pre-Associate
     - Pre-associate VSI Instance Identifier to a VTID.
     - Validate parameters.
     - Notify bridge to prep for Association.
  2. Pre-Associate with resource reservation.
     - Same as Pre-Associate. Reserves resources in addition.
  3. Associate
     - Associate VSI Instance Identifier to a VTID.
     - Allow resources are allocated and VSI is active.
  4. De-Associate
     - De-associate a VSI Instance Identifier from the associated VTID.
- Responses:
  - Each of the above Modes has an associated Response.

Pre-associate (0x0000) Semantics

- Pre-Associate VSI Instance Identifier to a VSI Type ID.
- If required, should obtain VSI Type Definition from the VSI Manager Database.
- Validate the request and fail it in case of errors.
- Successful Pre-Association does not enable any traffic from VSI.
  - Note that VSI may still be associated at another station.
- Makes Associate response faster. Important for VM mobility and failover.

Pre-associate Completion (0x00nn) Semantics

- Second Mode octet contains the results of the Pre-Associate requested for the VSI Instance ID (VSIID).
  - Success x0000 - Pre-Associate was successful. The switch shall permit a subsequent Associate or De-Associate by the VSI referenced by the VSI Instance Identifier.
  - The following are all unsuccessful Pre-Associate Completions. For each of these, the switch shall not permit a subsequent Associate or De-Associate by the VSI referenced by the VSIID.
    - Invalid Format.
    - Insufficient PT Resources.
    - Unused VTID
    - VTID Violation
    - VTID Version Violation

Pre-Associate with resource reservation (0x0100) Semantics

- Pre-association of a VSI Instance Identifier to a VSI Type Identifier
  - Same steps as Pre-Associate
  - Additionally:
    - Bridge should validate required resources and place reservation.
    - Enable pre-Associate timer to conserve resources.
- Does not allow any traffic from VSI.

Pre-associate with Resource Reservation Completion Semantics (0x01nn)

- Second Mode octet contains the results of the Pre-Associate with Resource Reservation request performed for the VSIID.
  - Success 0x0100 - Pre-Associate was successful. Prior to issuing this response, the switch shall reserve resources for use in a subsequent Associate or De-Associate by the VSI referenced by the VSIID.
  - The following are all unsuccessful Pre-Associate Completions. For each of these, the switch shall not permit a subsequent Associate or De-Associate by the VSI referenced by the VSIID.
    - Invalid Format.
    - Insufficient PT Resources.
    - Unused VTID
    - VTID Violation
    - VTID Version Violation

Associate (0x02) Semantics

- Associates the VSI Instance ID with the VSI Type ID
  - Allocates required bridge resources for referenced the VSI Type ID.
  - Binds specific MAC/VLAN pairs with the VSI Type ID.
  - Activates the switch configuration for the VSI Type ID.
- For a given VSI Instance ID, a Station may issue an Associate without having previously issued a Pre-Associate or Pre-Associate with Resource Reservation.
  - Same VSI may not be successfully Associated more than once.

Associate Completion (0x02nn) Semantics

- Second Mode octet contains the results of the Associate request performed for the VSI Instance Identifier.
  - Success x0200 – Associate was successful. Prior to issuing this response, the switch shall:
    - For a Format 1 TLV, associates the VSI Type referenced by the VSI Type Identifier and VSI Type Version with the MAC Address, VLAN and VSIID.
  - The following are all unsuccessful Associate Completions. For each of these, the switch shall not permit a subsequent De-Associate by the VSI referenced by the VSIID.
    - Invalid Format 0x0201
    - Insufficient PT Resources – Note: If the Associate was preceeded by a successful Pre-Associate with Resource Reservation, then the switch shall not use this response.
    - VTID Violation
    - VTID Version Violation
    - Out of Sync

De-Associate (0x03) Semantics

- De-Associate VSI Instance Identifier from a VTID.
  - Pre-Associated and Associated VSIs can be De-Associated.
  - De-Associate releases resources and de-activates the configuration associated with the VSIID.
  - A VSI Instance may get De-Associated by bridge due to bridge error situation or management action.

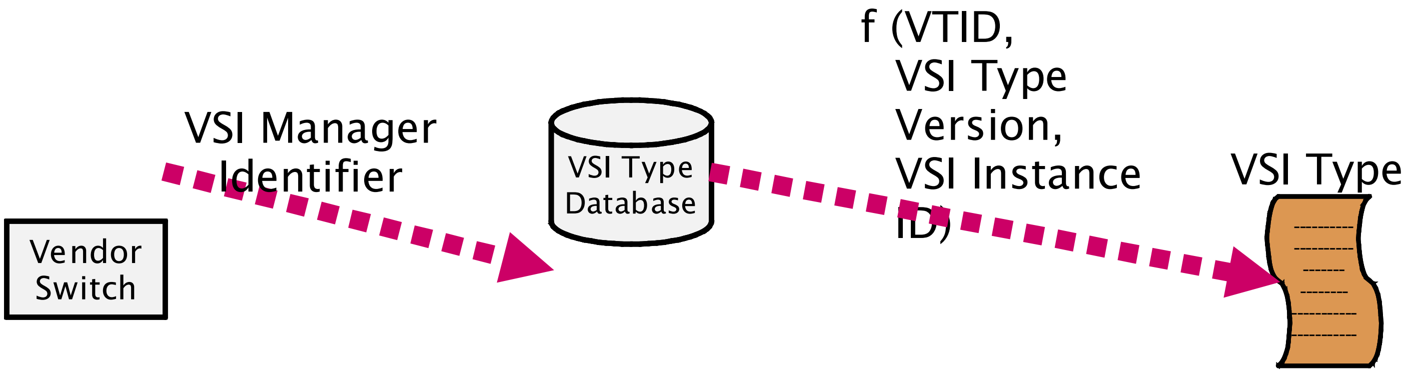
De-Associate Completion (0x03nn) Semantics

- Second Mode octet contains the results of the De-Associate request performed for the VSIID.
  - Success x0300 - De-Associate was successful. Prior to issuing this response, the switch shall:
    - For a Format 1 TLV, de-associates the VSI Type referenced by the VSI Type Identifier from the the MAC Address, VLAN and VSIID.
  - The following are all unsuccessful De-Associate Completions
    - Invalid Format.
    - VTID Violation.
    - VTID Version Violation.

Note: The result of the above semantics is that De-Associate can be issued at any time.

VSI Manager Identifier Semantics

- Definition: Identifies the VSI Manager with the Database that holds the detailed VSI Type and/or VSI Instance Identifier definitions.
  - The contents of the VSI Manager Database are outside the scope of this specification.
  - The VSI Manager Database may use a combination of the following fields to index into the VSI Manager Database:
    1. VSI Type Identifier
    2. VSI Type Version
    3. VSI Instance Identifier



VSI Type Identifier Semantics

- Definition: Integer value field used to identify a pre-configured set of controls/attributes that are to be associated with a set of VSIs.
  - VTID contents/meaning and the database used to contain the VSI Type are outside the scope of this effort.
  - One VTID may be describe the VSI Type configuration of multiple VSIs.
  - The VSI Type content referenced by the same VTID may differ between switches and VEBs. For example:
    - Same VTID is used by switches from two different vendors.
    - Same VTID is used by a VEB and vendor switches.

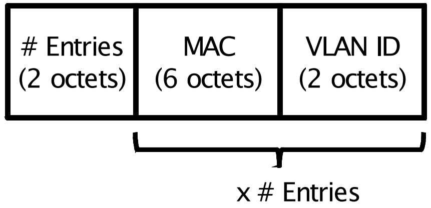
VSI Type Identifier Version Semantics

- Definition: The integer identifier designating the expected/desired VTID.
  - The VTID Version enables a VSI Manager Database to contain multiple VSI Type versions.
  - Allows smooth migration to newer VSI types.

VSI Instance ID

- Purpose: A globally unique ID for the VSI instance. The ID shall be done consistent with IETF RFC 4122 VSI ID is unique.

Format



# Entries (2 octets)

MAC (6 octets)

VLAN ID (2 octets)

x # Entries

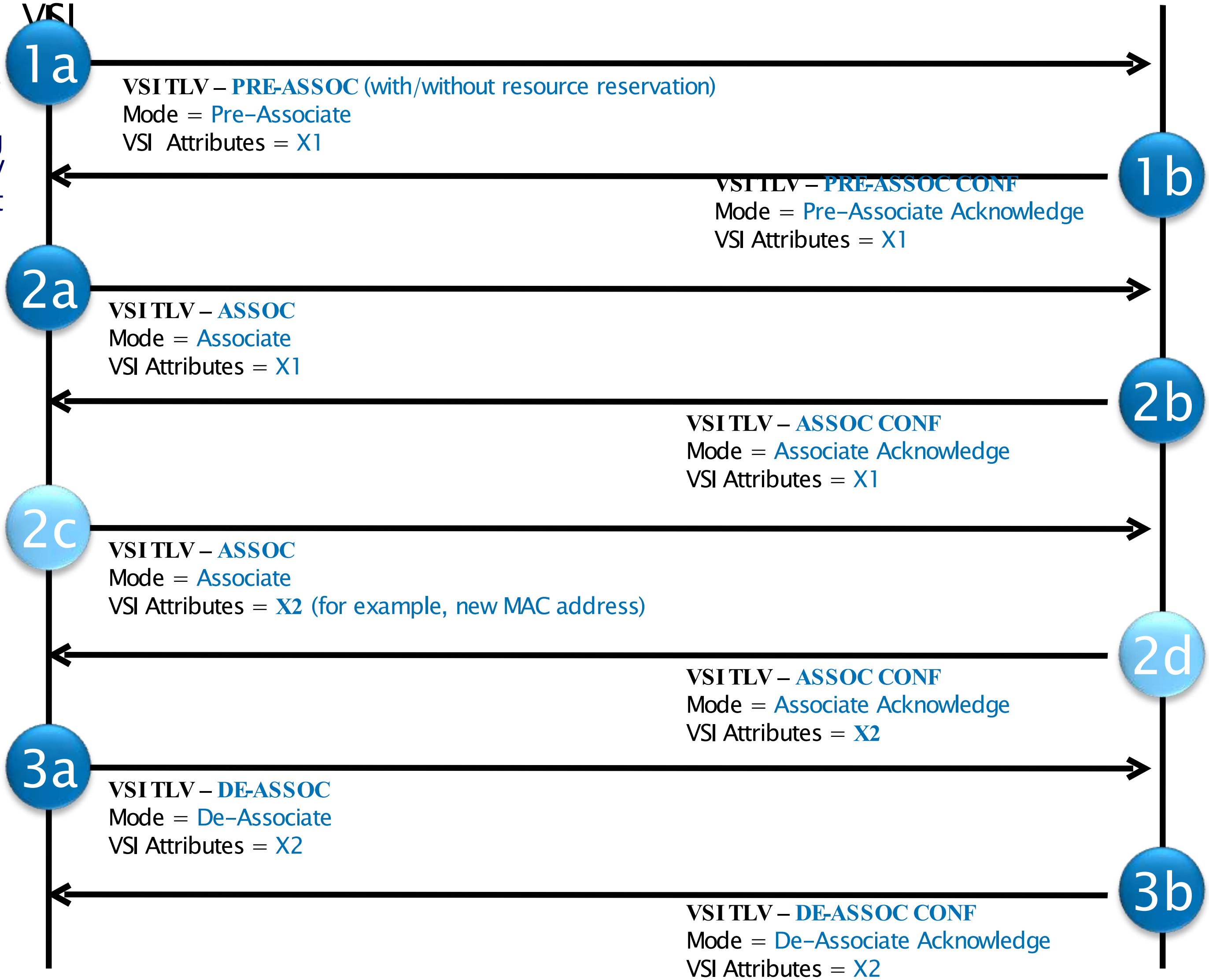
- Format 1
  - Definition: Contains the set of MAC Addresses and VLANs to be associated with the VSI Instance ID.
    - Note the bridge uses MAC+VID to identify traffic from VSI and to steer the frames.
  - Field:
    - #MAC-VLAN pairs: 2 octets

      Per Pair Content:
      - MAC address: 48 bits
      - VID: 12 bits

VSI Discovery/Config. Exchange Example

Station (Hypervisor)/ VSI

Bridge



This exchange is usually done by Station using the T3P-R TLV transport

1a VSI TLV – PRE-ASSOC (with/without resource reservation)
Mode = Pre-Associate
VSI Attributes = X1

1b VSI TLV – PRE-ASSOC CONF
Mode = Pre-Associate Acknowledge
VSI Attributes = X1

Switch fetches the settings from the VSI Type database server or from a local cache.

2a VSI TLV – ASSOC
Mode = Associate
VSI Attributes = X1

Activates the new VSI connection

2b VSI TLV – ASSOC CONF
Mode = Associate Acknowledge
VSI Attributes = X1

Server announces change to configuration

2c VSI TLV – ASSOC
Mode = Associate
VSI Attributes = X2 (for example, new MAC address)

2d VSI TLV – ASSOC CONF
Mode = Associate Acknowledge
VSI Attributes = X2

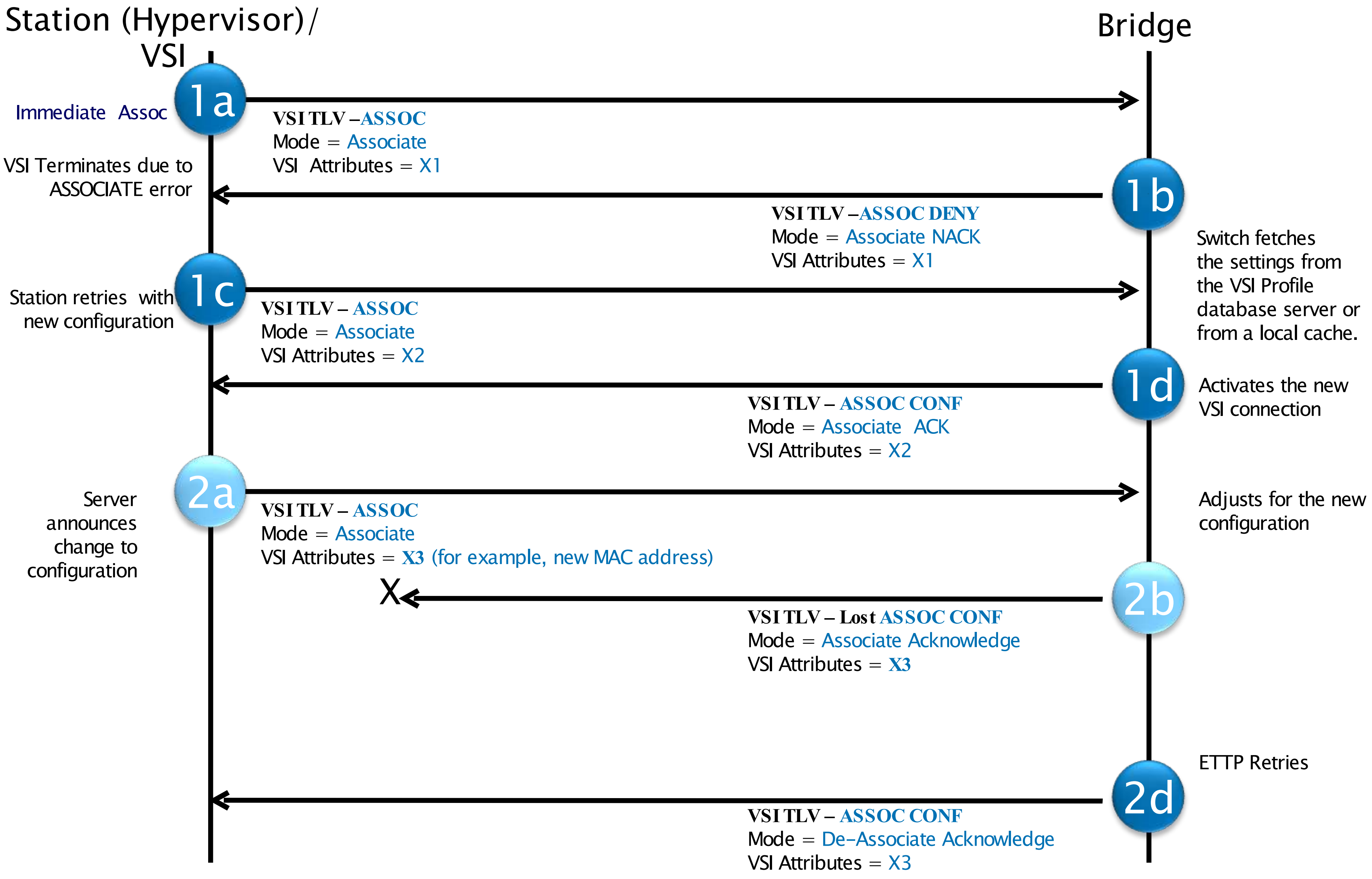
Adjusts for the new configuration

3a VSI TLV – DE-ASSOC
Mode = De-Associate
VSI Attributes = X2

Tears down VSI configuration

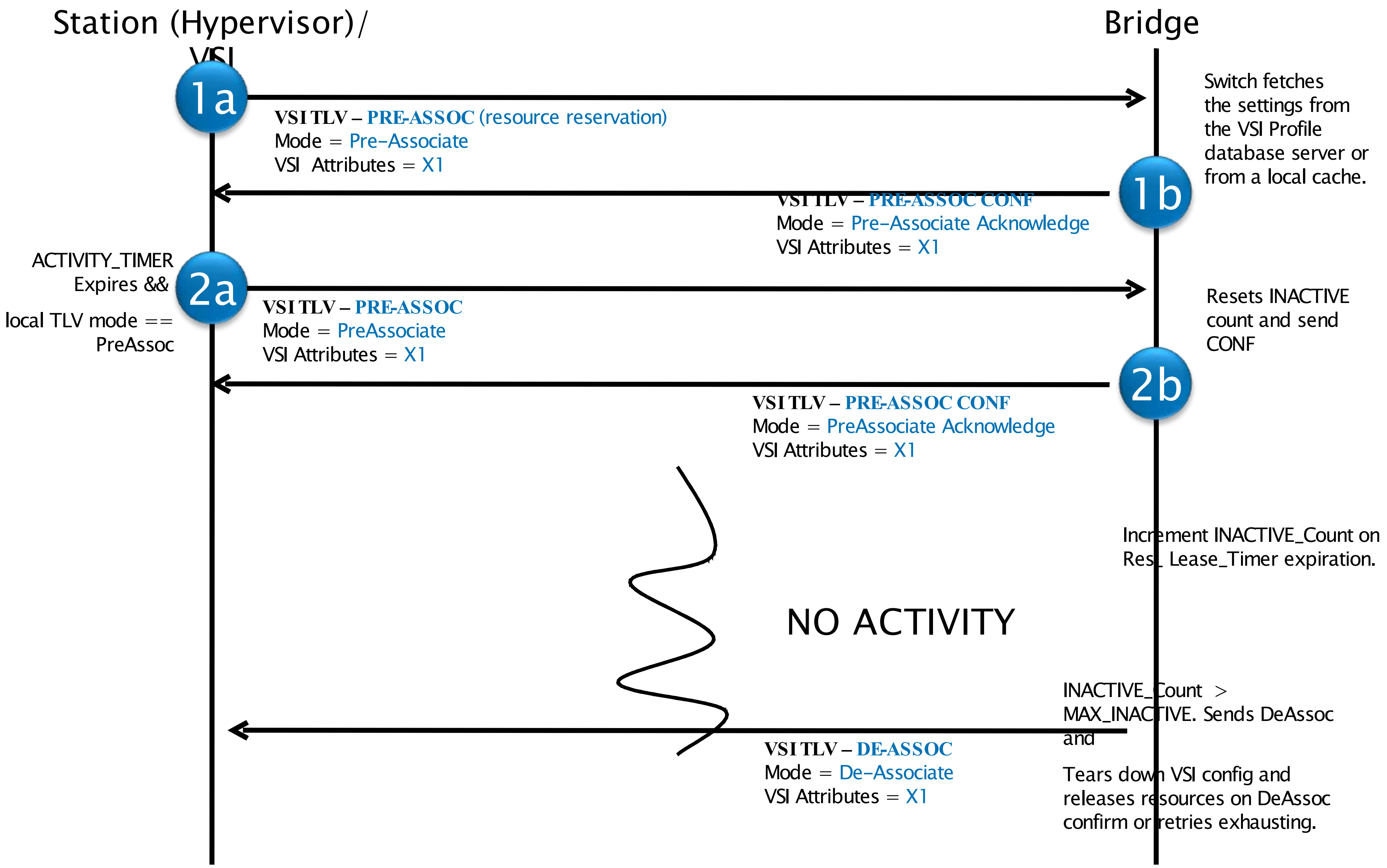
3b VSI TLV – DE-ASSOC CONF
Mode = De-Associate Acknowledge
VSI Attributes = X2

VSI Exchange Example – Direct Associate with Error Situations



Station (Hypervisor)/ VSI
Bridge
1a
Immediate Assoc
VSI TLV –ASSOC
Mode = Associate
VSI Attributes = X1
VSI Terminates due to ASSOCIATE error
1b
VSI TLV –ASSOC DENY
Mode = Associate NACK
VSI Attributes = X1
Switch fetches the settings from the VSI Profile database server or from a local cache.
1c
Station retries with new configuration
VSI TLV – ASSOC
Mode = Associate
VSI Attributes = X2
1d
Activates the new VSI connection
VSI TLV – ASSOC CONF
Mode = Associate ACK
VSI Attributes = X2
2a
Server announces change to configuration
VSI TLV – ASSOC
Mode = Associate
VSI Attributes = X3 (for example, new MAC address)
Adjusts for the new configuration
X
2b
VSI TLV – Lost ASSOC CONF
Mode = Associate Acknowledge
VSI Attributes = X3
ETTP Retries
2d
VSI TLV – ASSOC CONF
Mode = De-Associate Acknowledge
VSI Attributes = X3

VSI Exchange Example – PreAssoc Resource Lease



Station (Hypervisor)/ VSI

Bridge

1a

VSI TLV – PRE-ASSOC (resource reservation)
Mode = Pre-Associate
VSI Attributes = X1

Switch fetches the settings from the VSI Profile database server or from a local cache.

1b

VSI TLV – PRE-ASSOC CONF
Mode = Pre-Associate Acknowledge
VSI Attributes = X1

ACTIVITY_TIMER Expires &&

local TLV mode == PreAssoc

2a

VSI TLV – PRE-ASSOC
Mode = PreAssociate
VSI Attributes = X1

Resets INACTIVE count and send CONF

2b

VSI TLV – PRE-ASSOC CONF
Mode = PreAssociate Acknowledge
VSI Attributes = X1

Increment INACTIVE_Count on Res_Lease_Timer expiration.

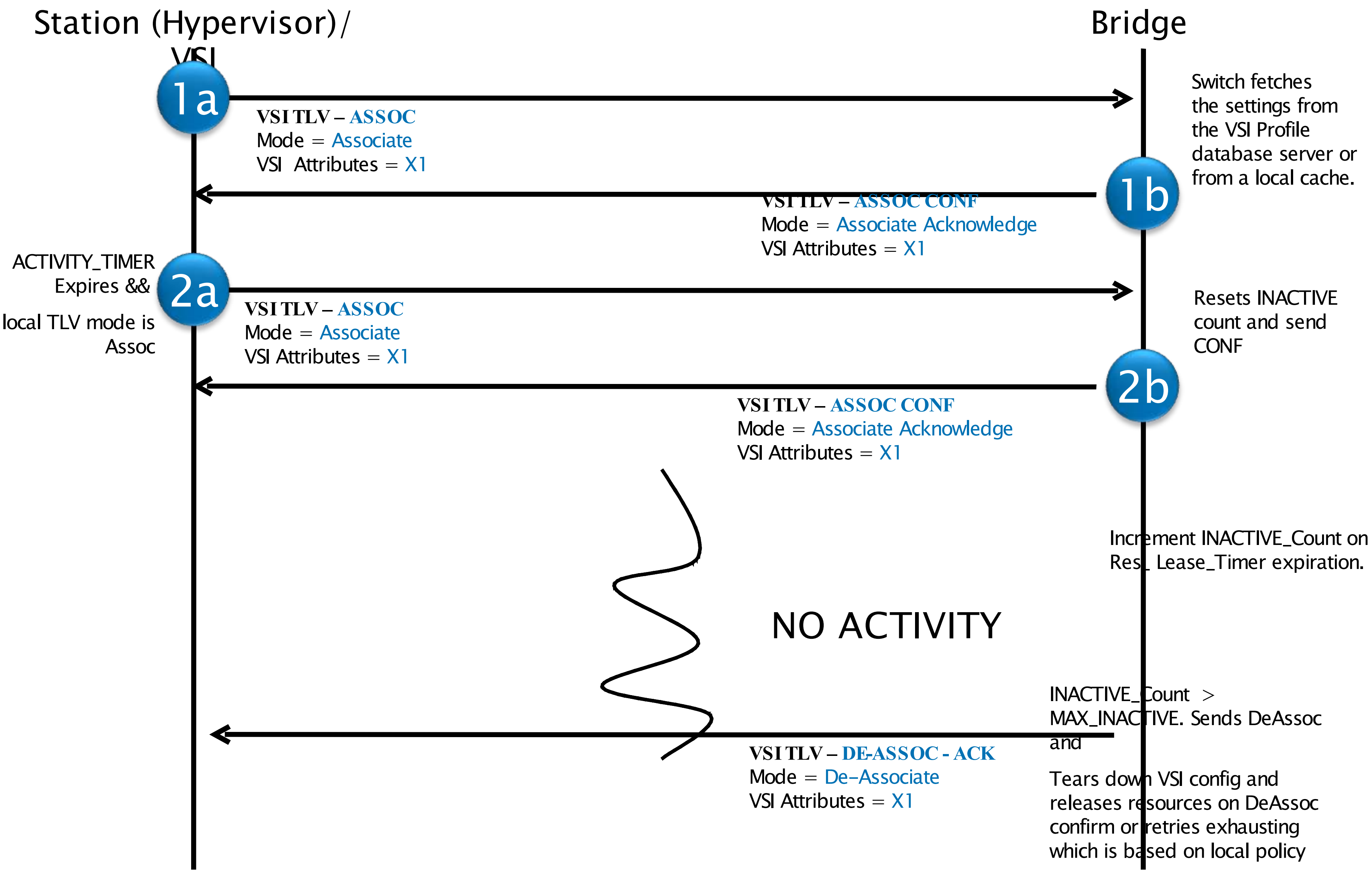
NO ACTIVITY

INACTIVE_Count > MAX_INACTIVE. Sends DeAssoc and

Tears down VSI config and releases resources on DeAssoc confirm or retries exhausting.

VSI TLV – DE-ASSOC
Mode = De-Associate
VSI Attributes = X1

VSI Exchange Example – Assoc Resource Lease Refresh



Station (Hypervisor)/ VSI

Bridge

VSI TLV – ASSOC
Mode = Associate
VSI Attributes = X1

Switch fetches the settings from the VSI Profile database server or from a local cache.

VSI TLV – ASSOC CONF
Mode = Associate Acknowledge
VSI Attributes = X1

ACTIVITY_TIMER Expires &&

local TLV mode is Assoc

VSI TLV – ASSOC
Mode = Associate
VSI Attributes = X1

Resets INACTIVE count and send CONF

VSI TLV – ASSOC CONF
Mode = Associate Acknowledge
VSI Attributes = X1

Increment INACTIVE_Count on Res_ Lease_Timer expiration.

NO ACTIVITY

INACTIVE_Count > MAX_INACTIVE. Sends DeAssoc and

VSI TLV – DE-ASSOC - ACK
Mode = De-Associate
VSI Attributes = X1

Tears down VSI config and releases resources on DeAssoc confirm or retries exhausting which is based on local policy

VSI Discovery and Configuration Requirements

1. Support VSI preAssociate (with and without resource reservations), Associate and deAssociate.
2. ASSOCIATE, PreAssociate and DeAssociate are Idempotent i.e. can be repeated.
3. Capability to Associate skipping PreAssociate.
4. VDP will work both for VEPA and VEB environments.
5. VSI TLVs are defined on slide 6-9.
6. Local or remote event can arrive in any state.
7. VSI state machines ETTP as TLV transport and utilize following capabilities of ETTP
    1. Transport will be transmitting TLVs in-order and are received in-order.
    2. Flow control
    3. Transport error from ETTP and LLDP are indicated to VDP
    4. ETTP provides best effort delivery of TLV. Therefore VDP waits for ETTP waits for ACK_Timeout. If no response is received, VSI exits.
        1. ACK_Timeout = 2*T3P retransmission period * MAX-RETRIES)+locally administered wait described informative text.
8. Timeout mechanism to ensure:
    a. Bridge resources are not reserved too long for inactive VSIs (lease semantics)
    b. Allow removing resources from inactive VSIs with the goal of
        - Conserve bridges resources (Number VSIs being hadled by bridge can be large).
        - Prevent inactive or VMs in error state to continue to hold resources.
    - Timeout out values to be negotiated on per channel between station and bridge. One timeout used for all ULPs on the channel negotiated using EVB TLV.

VSI Discovery and Configuration Requirements

Manageability and Robustness

a. Ensure VSI state and configuration between the Station and the Bridge remains consistent.

b. Hard errors at the Bridge or the Hypervisor that can impact individual VSI or Hypervisor/Bridge as a whole (resetAll)

   - Option 1: All VSI configuration goes away.

c. Bridge and Station Errors are detected through one or more of the following mechanisms.

   - VSI KEEP-ALIVE (periodic transmission of VSI TLV from station and response from Bridge)
   - ACK Timer
   - Transport (ETTP and LLDP) status indications.

d. Support switch/hypervisor administrator actions force VSI deAssociate.

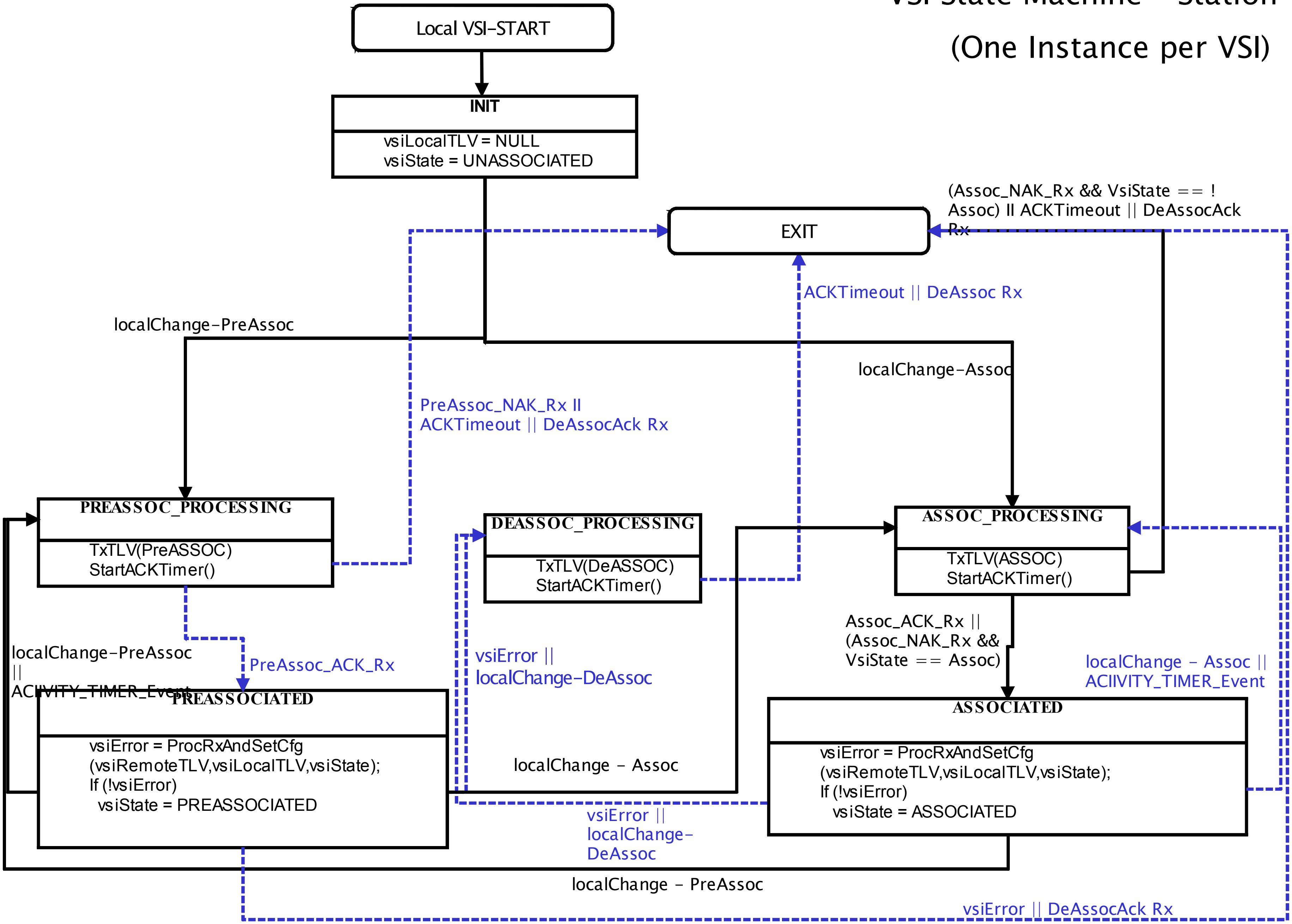
e. Statistics and logging support (need specific proposal)

Lost Digest Sync Semantics for State Machine, for each VSI Instance,

a. TLV Digest based sync is future enhancement and is not covered here.

VSI State Machine – Station

(One Instance per VSI)



Local VSI-START

INIT

vsiLocalTLV = NULL
vsiState = UNASSOCIATED

EXIT

(Assoc_NAK_Rx && VsiState == ! Assoc) II ACKTimeout || DeAssocAck Rx

ACKTimeout || DeAssoc Rx

localChange-PreAssoc

localChange-Assoc

PreAssoc_NAK_Rx II
ACKTimeout || DeAssocAck Rx

PREASSOC_PROCESSING

TxTLV(PreASSOC)
StartACKTimer()

DEASSOC_PROCESSING

TxTLV(DeASSOC)
StartACKTimer()

ASSOC_PROCESSING

TxTLV(ASSOC)
StartACKTimer()

Assoc_ACK_Rx ||
(Assoc_NAK_Rx &&
VsiState == Assoc)

localChange-PreAssoc
||
ACIVITY_TIMER_Event

PreAssoc_ACK_Rx

vsiError ||
localChange-DeAssoc

localChange – Assoc ||
ACIIVITY_TIMER_Event

PREASSOCIATED

vsiError = ProcRxAndSetCfg
(vsiRemoteTLV,vsiLocalTLV,vsiState);
If (!vsiError)
vsiState = PREASSOCIATED

localChange – Assoc

ASSOCIATED

vsiError = ProcRxAndSetCfg
(vsiRemoteTLV,vsiLocalTLV,vsiState);
If (!vsiError)
vsiState = ASSOCIATED

vsiError ||
localChange-DeAssoc

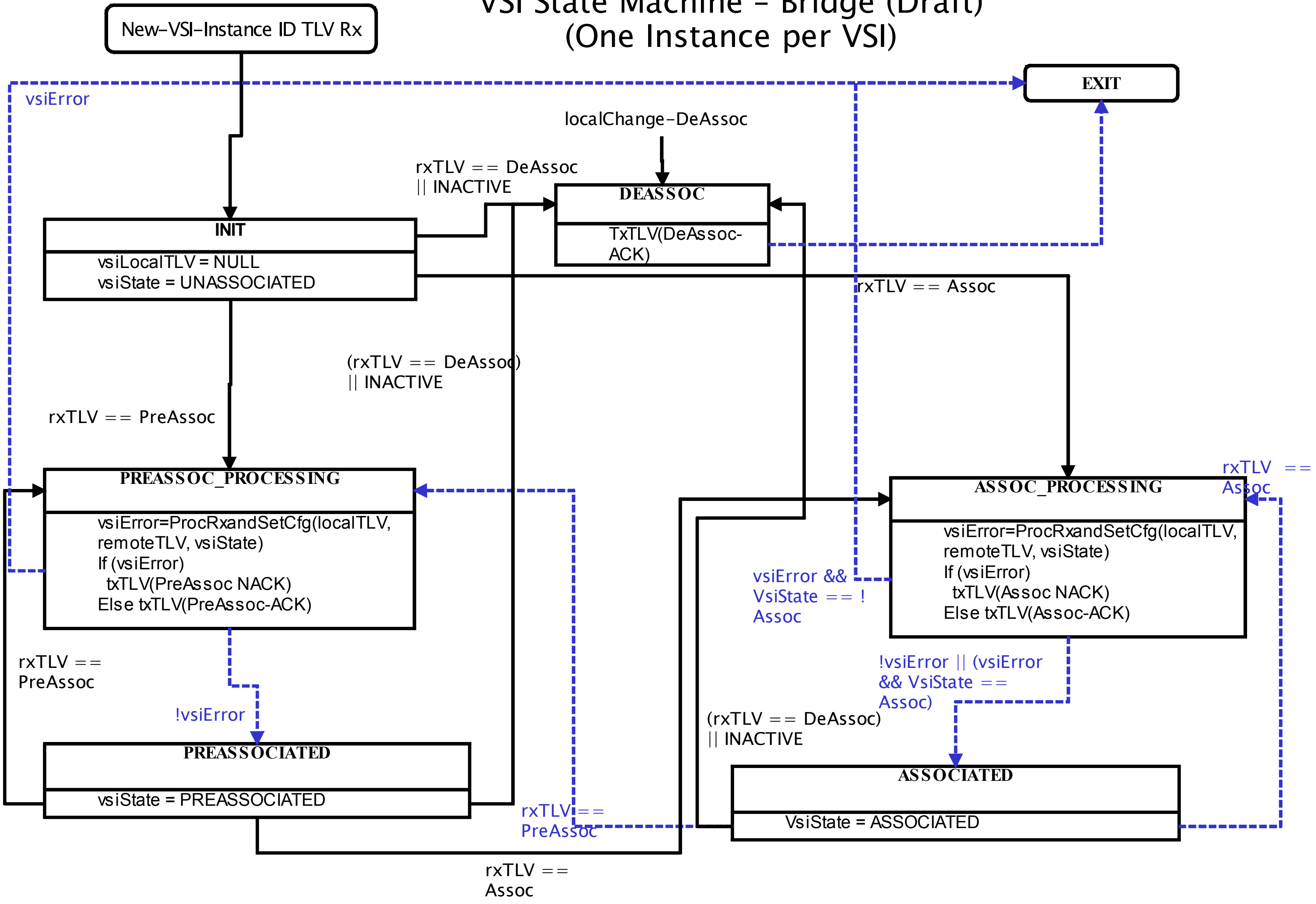
localChange – PreAssoc

vsiError || DeAssocAck Rx

Notes on VSI State Machine

1. **vsiState:** Local variable for current state.
2. **localTLV:** Current local (active) TLV (configuration)
3. **AdminTLV:** TLV from local administration. In addition appropriate localChange variable is set. It allows mode change
4. **RemoteTLV:** TLV received from remote.
5. **TxTLV(*vsiTLV*):** Transmits AdminTLV using TLV transport (ETTP) service interfaces. Includes support for aggregation of VSI TLVs.
6. **ProcRxandSetCfg(***vsiRemoteTLV,vsiLocalTLV,vsiState***):** Processes receive TLV and Sets local TLV variable based on received Remote TLV and vsiState. In case of error, returns error. This function handles PreAssociate with and without resource reservation case as well as accessing VSI Type definition fetch, if required
7. **StartACKtimer**(): Resets ACKTimeout local variable to FALSE and Starts ACK timer. Response (ACK or NACK) is expected before timer expires.
8. **ACKTimeout**: This local variable is set to true, if ACK timer expires
9. **vsiErrorPerm**(*vsiRemoteTLV*): processes the vsiRemoteTLV and returns TRUE is response code is unrecoverable (permanent) error.

VSI State Machine – Bridge (Draft)
(One Instance per VSI)



New–VSI–Instance ID TLV Rx

vsiError

EXIT

localChange–DeAssoc

rxTLV == DeAssoc
|| INACTIVE

DEASSOC

TxTLV(DeAssoc-
ACK)

INIT

vsiLocalTLV = NULL
vsiState = UNASSOCIATED

rxTLV == Assoc

(rxTLV == DeAssoc)
|| INACTIVE

rxTLV == PreAssoc

PREASSOC_PROCESSING

vsiError=ProcRxandSetCfg(localTLV,
remoteTLV, vsiState)
If (vsiError)
  txTLV(PreAssoc NACK)
Else txTLV(PreAssoc-ACK)

rxTLV ==
Assoc

ASSOC_PROCESSING

vsiError=ProcRxandSetCfg(localTLV,
remoteTLV, vsiState)
If (vsiError)
  txTLV(Assoc NACK)
Else txTLV(Assoc-ACK)

vsiError &&
VsiState == !
Assoc

rxTLV ==
PreAssoc

!vsiError

PREASSOCIATED

vsiState = PREASSOCIATED

!vsiError || (vsiError
&& VsiState ==
Assoc)

(rxTLV == DeAssoc)
|| INACTIVE

ASSOCIATED

VsiState = ASSOCIATED

rxTLV==
PreAssoc

rxTLV ==
Assoc

Backup

Station VSI State Machine Definitions

Function	Return	Description
TxTLV(vsiTLV)		Transmits TLV using DBA/T3P-R service i/f
rxTLV()		Receive TLV from DBA/T3P-R service i/f
vsiErrorPerm(vsiTLV)	{TRUE,FALSE}	Returns TRUE TLV has unrecoverable (perm) error.
ProcRxTLV()	vsiError	Bridge function to process receive TLV and generate appropriate return code. This function handles PreAssociate with and without resource reservation case as well as accessing VSI Type definition fetch, if required.

Station VSI State Machine Definitions

Local variable	Type	Range	Description
vsiError	Enum	{OK,TEMP-ERR,PERM-ERR}	One instance per VSI
vsiLocalTLV	Struct	VSI TLV fields structure (see next chart)	VSI local configuration updated by state machine. Active VSI configuration.
vsiAdminTLV	Struct	VSI TLV fields structure (see next chart)	VSI admin configuration
vsiRemoteTLV	Strucure	VSI TLV fields structure (see next chart)	VSI Remote configuration (received TLV)..
<vsiTLVmode>-ACK	Boolean	{TRUE, FALSE}	VSI local variable. TRUE indicates the response code in the TLV is 0x00 (SUCCESS)
<vsiTLVmode>-NACK	Boolean	{TRUE, FALSE}	VSI local variable. TRUE indicates the response code in the TLV is NOT 0x00
localChange	Boolean	{TRUE, FALSE}	Local configuration change i.e. vsiAdminTLV is changed.
TX-RETRY-DELAY	Integer	1 - 200	Tx retry timer value with granularity of 500 msec. Specified as integer representing 500 msec periods. Fro example, A value of 6 means 3 seconds.
retryCount		Integer, min: 0 , max: MAX_RETRY_COUNT	TLV send retry counter. If retry count exceeds, MAX_RETRY_COUNT, error flag is raised. MAX_RETRY_COUNT is locally administered value

Station VSI State Machine Definitions

Local variable	Type	Range	Description
TX_RETRY_DELAY	Integer.	1 – 200	Tx retry timer value with granularity of 500 msec. Specified as integer representing 500 msec periods. For example, A value of 6 means 3 seconds.
ACTIVITY_TIMER	Integer	1 – 16k	Used for resource lease to be maintained. It has granularity of 10 seconds. For example, A value of 6 means 60 seconds

Notes:

1. Station VSI local variables instance are per VSI, VDP/vSwitch. VDP can access all VSI and T3P-R variables.

Station T3P Definitions

Local variable	Type	Range	Description
t3pAdminStatu s		{enabled, disabled}	T3P-R agent is enabled or disabled. VDP and VSI have read access
t3pStats		T3P statistics	T3P statistics array
t3pStatus		{OK(0),ERROR}	T3P operational status

Notes:

1. Station VSI local variables instance are per VSI, VDP/vSwitch. VDP can access all VSI and T3P-R variables.

VSI Discovery and Configuration Protocol (VDP) Module (Implementation Note)

VSI	VSI TLV Fields	VSI state	VSI Timer Ticks	VSI statistics
VSI0	VSI0 TLV fields	VSI0 state	timerTicks	TBD
VSI1	VSI1 TLV fields	VSI1 state	timerTicks	TBD
VSIn	VSIn TLV fields	VSIn state	timerTicks	TBD

- VDP module implementation can be single module that supports all VSIs that
  - Contains table of VSI variables
  - timerTicks updated by single timer with long duration (for example, every 10 or more seconds
  - VDP module get requests (create VSI, pre-Assoc, Assoc and de-Assoc) from VDP user (Hypervisor/Bridge OS) and sends events to the user.
  - Transmits TLVs by queueing to T3P module and receives TLVs from T3P for processing.