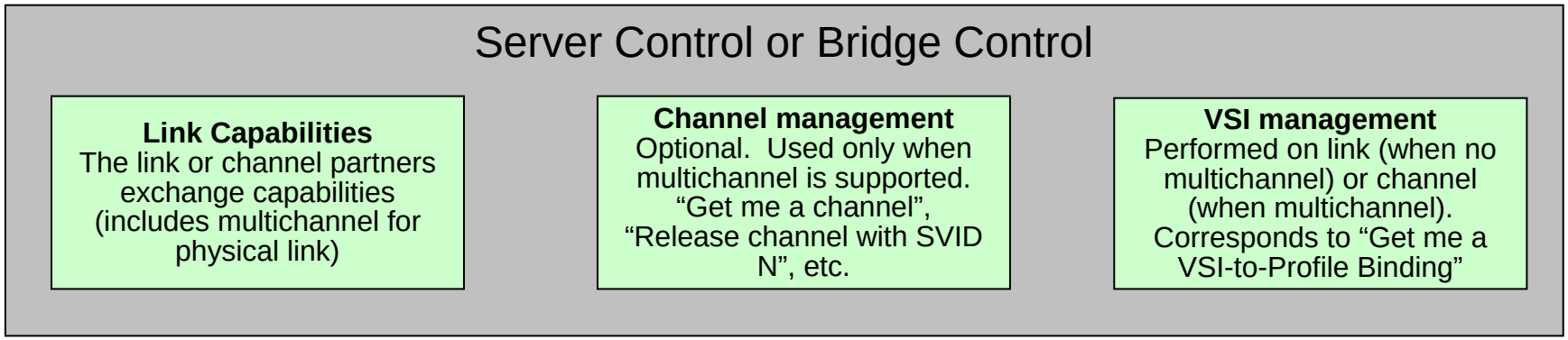
Link Capabilities and a Binding Control Protocol

EVB phone meeting Tuesday 2/2/10

Bob Sultan (bsultan@huawei.com)

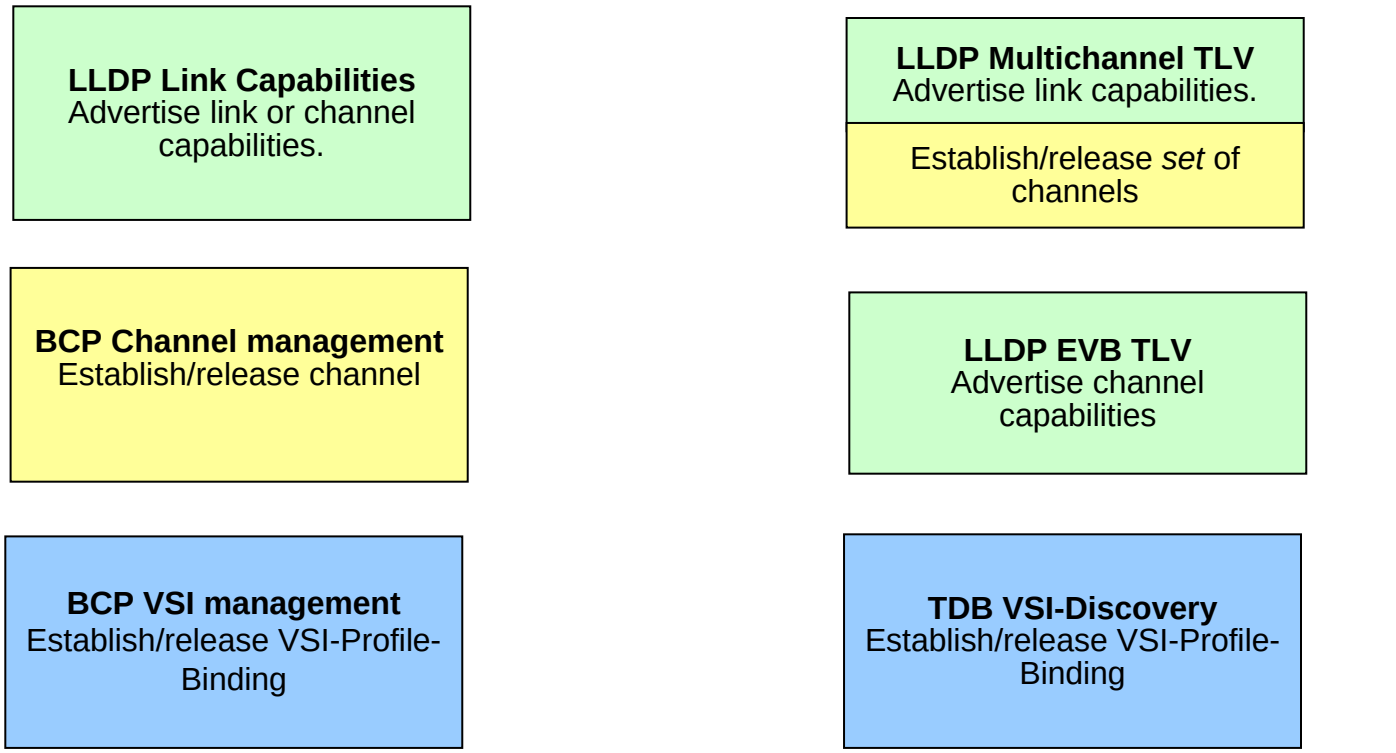
Three EVB protocol requirements



- Assumption: when a physical Link is configured *multichannel*, then each channel behaves like a link, except that the channel cannot be configured as multichannel (ie., multichannel capability is not recursive);
- *Link Capabilities* can use *existing* LLDP with a new TLV;
- *Channel management* provides a channel with identifying SVID and releases a channel identified by SVID;
  - Once provided, a channel behaves like a link and its capabilities are advertised by LLDP, like any other link;
- VSI Management establishes and releases VSI-to-Profile bindings;
  - VSI identified consistent with RFC 4122.
  - Profile is identified by Profile ID, version, database ID, etc.
    - Or (better) by an identifier that uniquely identifies this set of profile information on the bridge;
    - The relationship between the identifier and the profile information is established by a protocol not shown in this slideset;
  - Allows parameters such as a Traffic Stream Identification String (to allow traffic associated with the VSI to be identified by the bridge port);

Compare A B

evb-hudson-tlvoverview-0110-v09

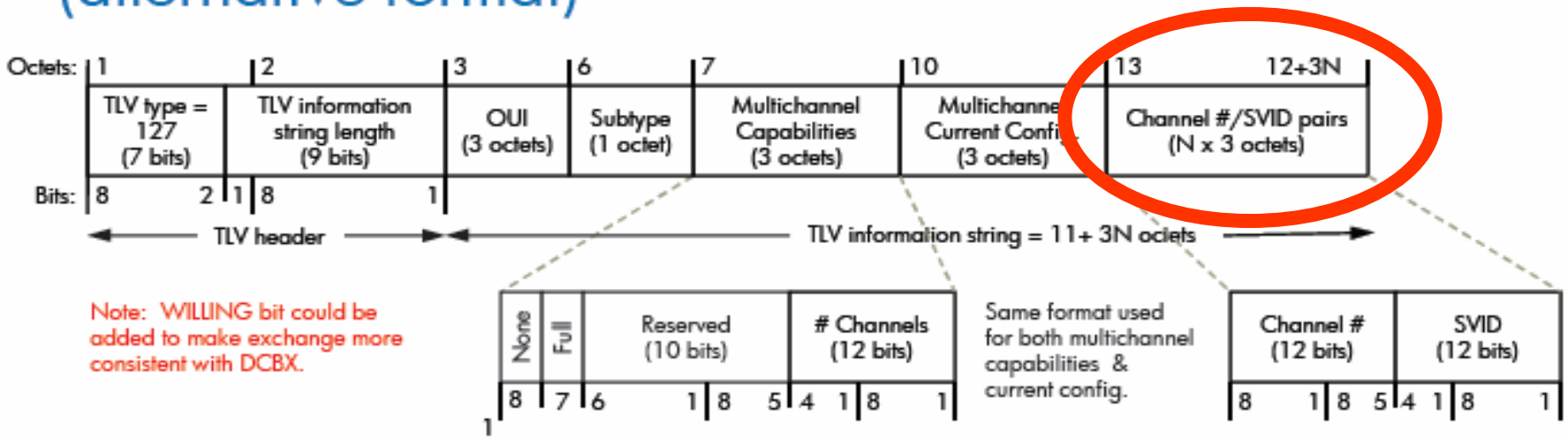


- **A** uses same LLDP TLV for link or channel capabilities; **B** uses two different TLVs for link capabilities (one for the physical link and one for the channel);
- **B** combines link capabilities with channel establish/release in Multichannel TLV;
- Method of channel establish/release in **B** has difficulties
  - You have twenty channels established; how do you release channel 117?
- What you want is the capability of establishing/releasing individual channels in the same way that you establish/release VSI-to-Profile bindings;

Complexity in releasing channel

Proposed Multichannel TLV (alternative format)

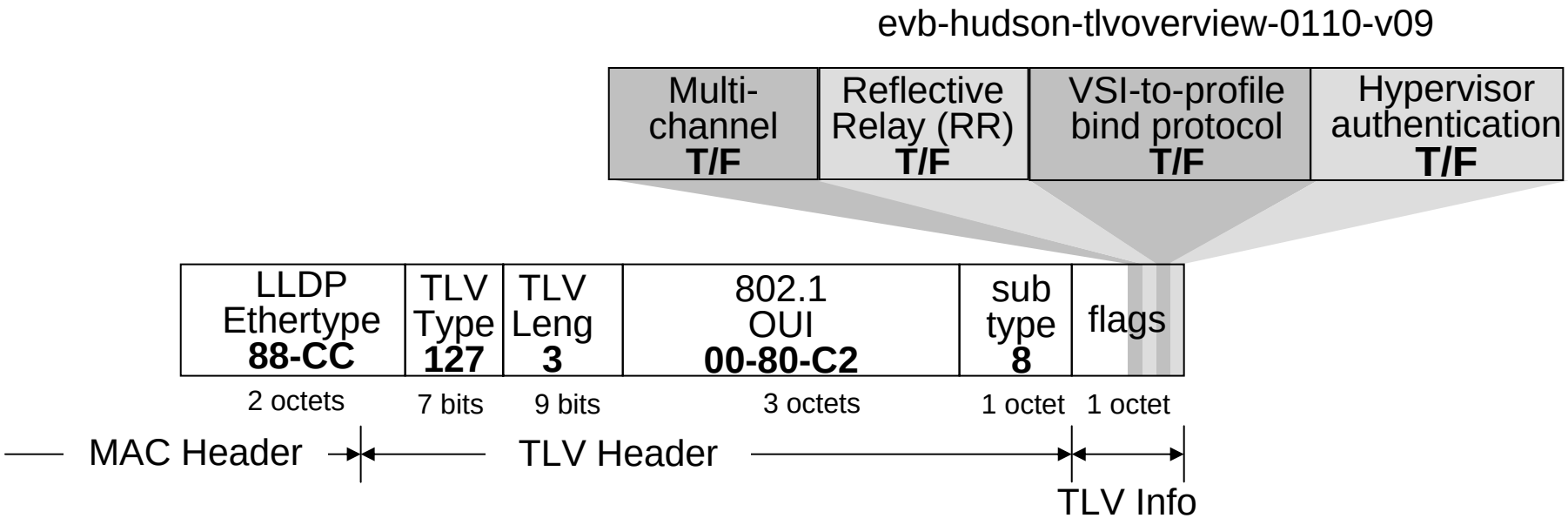
Transport: LLDP



- Multichannel Capabilities– Describes EVB multichannel capabilities that can be supported by the sender.
  - Number Channels Supported – Identifies the number of SVID channels that are supported by the sender.
- Multichannel Current Configuration – Identifies the EVB multichannel capabilities that are currently enabled by the sender. Only one channel mode may be selected.
  - Number Channels Configured – Identifies the number of SVID channels that are configured/desired by the sender.

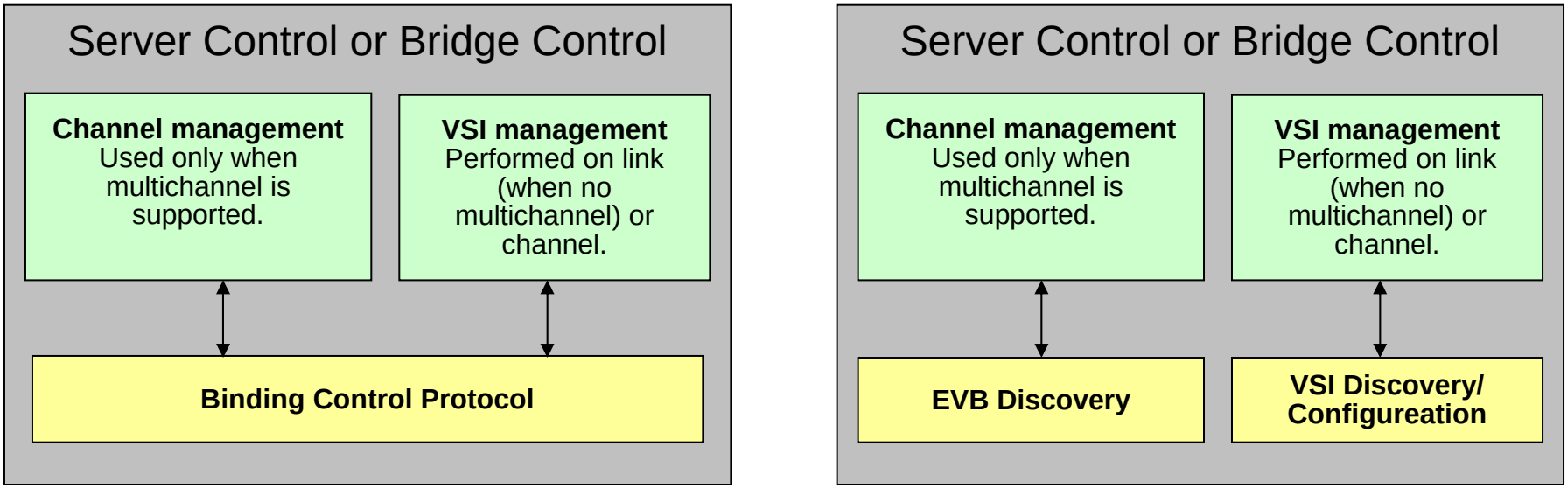
- Channel #/SVID Pairs
  - **Channel # – indicates the index number of the** channel. Allows insertion or deletion of specific channels while only listing the currently configured channels.
  - **SVID – The S-Tag VLAN ID assigned to the channel.** This is identified by the bridge. SVID of 0 means that no VLAN ID has been assigned.

Link Capabilities LLDP TLV



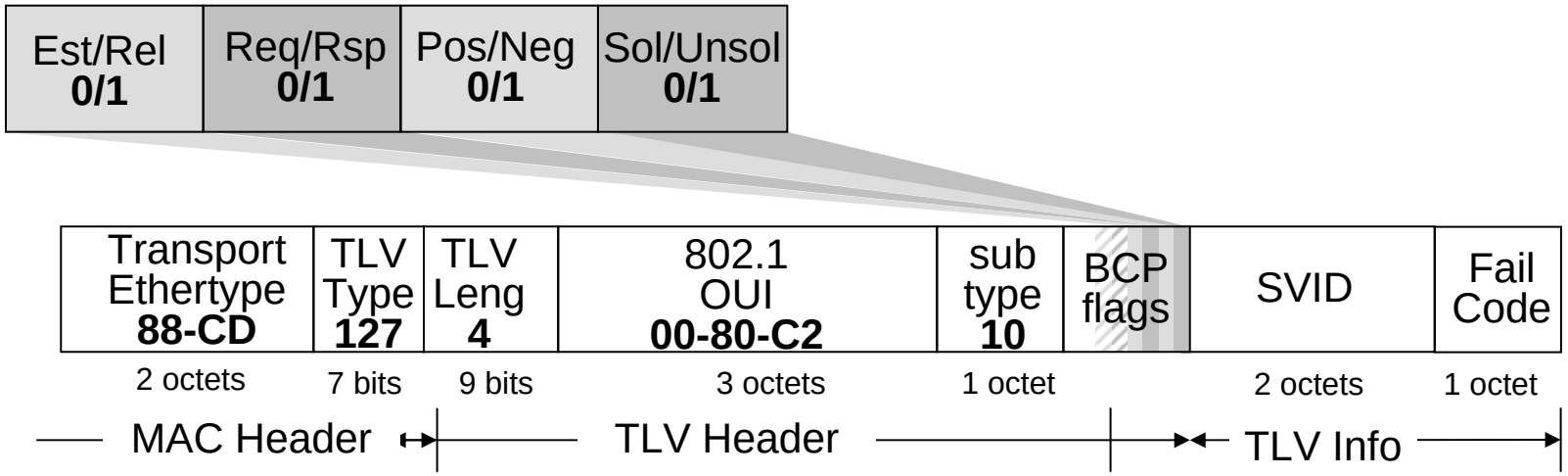
- Could contain additional link attributes;
- Multichannel must be **F** when advertised on channel;
- Values advertised by server are ‘requested’ values;
- Values advertised by bridge are ‘capabilities’;
- Could have separate ‘flag’ fields for ‘requested values’ and ‘capabilities’ as in evb-hudson-tlvoverview-0110-v09 bit this is not shown in this figure.

Binding Control Protocol



- BCP provides common protocol processing to support both Channel Management and VSI Management;
  - Establish/release, request/response, positive/negative response, unsolicited release, periodic renewal, etc. are common to both;

BCP Channel Management TLV



type	direction	BCP flags
Establish request	S →B	xxxx0000
Establish positive response	S →B	xxxx0100
Establish negative response	S →B	xxxx0110
Release request	B →S	xxxx1000
Release positive response	B →S	xxxx1100
Release negative response	B →S	xxxx1110
Unsolicited release	B →S	xxxx1001

NOTE: 'hatched' protocol flags are reserved when used with channel management; these represent 'pre-fetch' and 'reserve' when used with VSI Management TLV

- *SVID* significant in all messages except establish request;
- Fail code significant only in negative responses and unsolicited release;
- The channel management request TLV for a given channel is sent periodically by the server as a method of allowing unused channels to be reclaimed by the bridge;
  - Receiving the channel management request TLV corresponding to an existing channel has no impact on the channel (idempotency);

When Multichannel *not* supported on Link

LLDP Link Capabilities; server is not configured to use multichannel or bridge does not support multichannel, or both; Bridge adopts values of RR/NRR, VSI-to-Profile binding, and Hypervisor Authentication specified by server but flags configuration error if different from locally configured values.

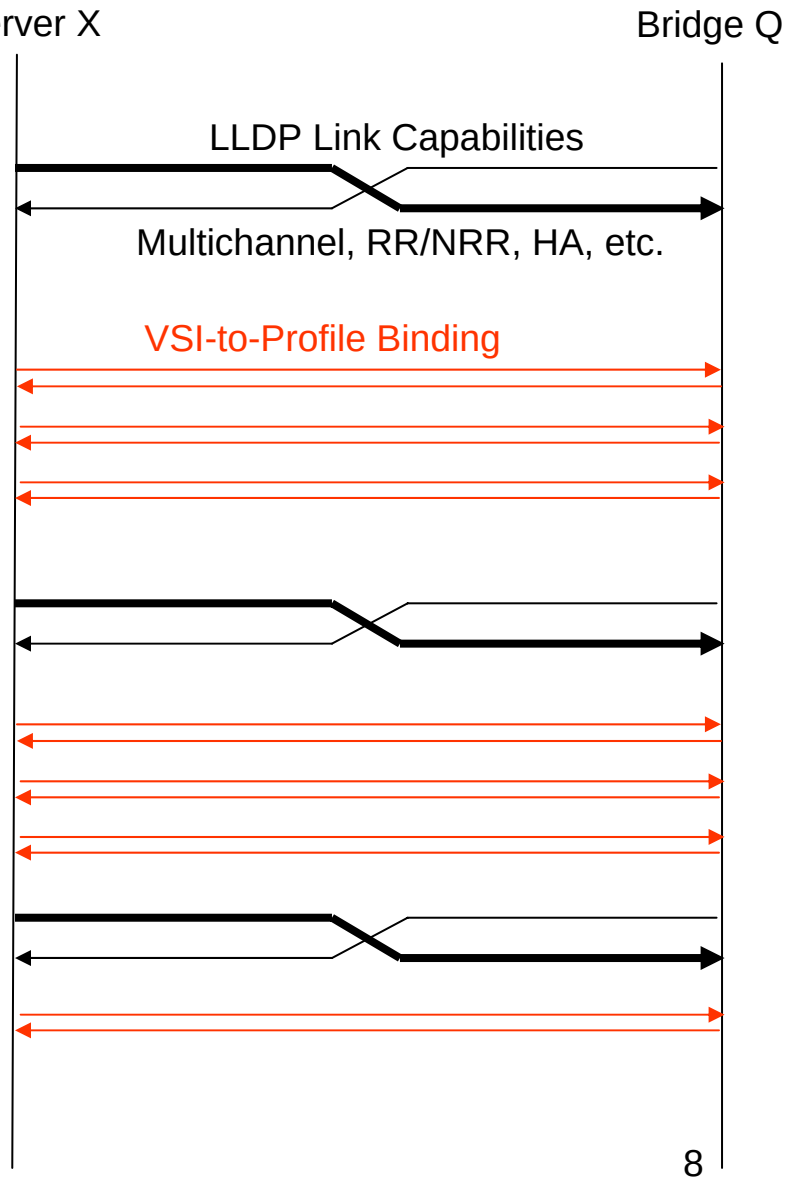
Sequence of VSI Management requests/responses associated with the link (including pre-fetch and reserve, and periodic 'renewals')

LLDP Link capabilities repeated periodically.

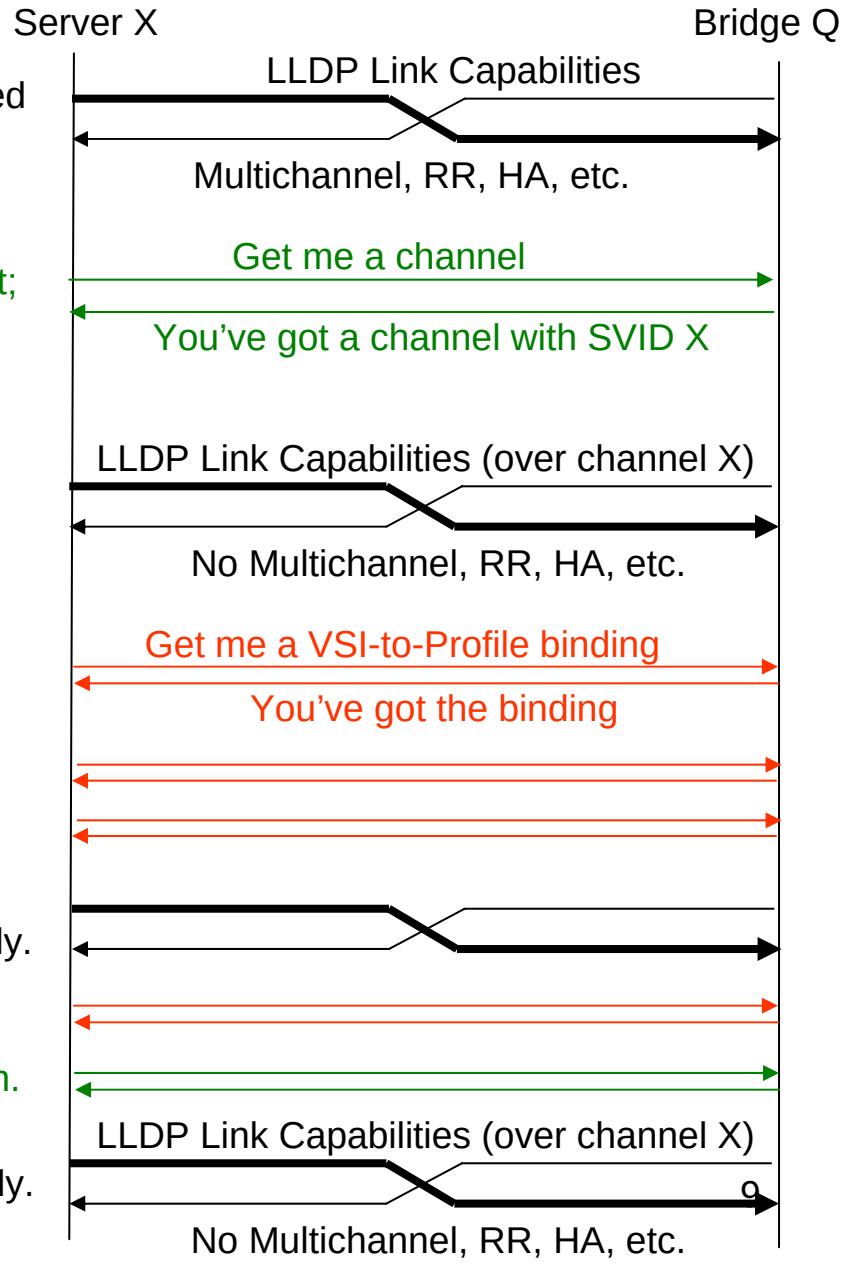
VSI Management messages

LLDP Link capabilities repeated periodically.

VSI Management messages



Multichannel example



LLDP Link Capabilities; case in which server is configured to use multichannel *and* bridge supports multichannel;

Channel management request for channel establishment;

Reply for established channel with SVID;

LLDP Link Capabilities advertised over channel.

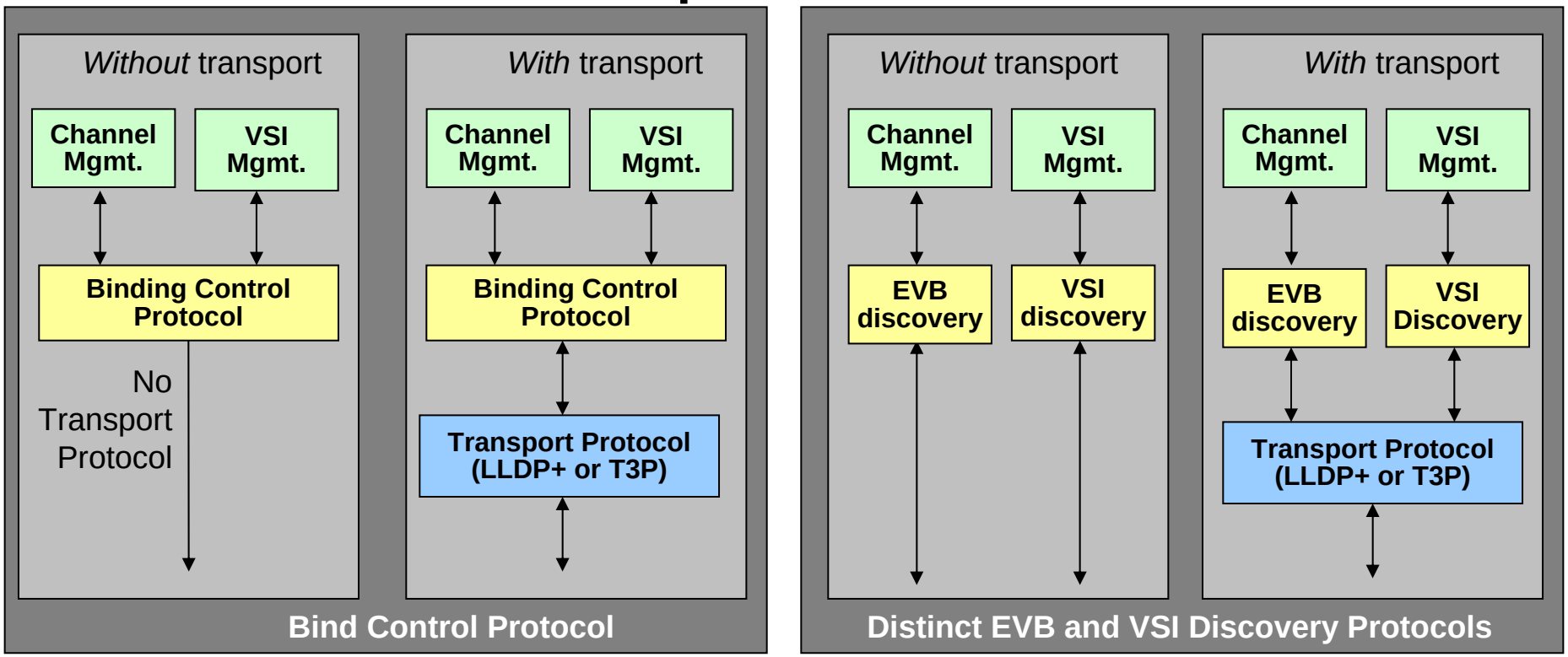
Sequence of VSI Management requests/responses associated with the channel (including pre-fetch and reserve, and periodic 'renewals')

LLDP Link capabilities over physical link repeated periodically.

Channel establish repeated periodically to refresh.

LLDP Link Capabilities over channel repeated periodically.

Transport Protocol



- The question of whether or not to deploy a transport protocol is *completely independent* of whether you deploy
  - the Binding Control Protocol ***or***
  - distinct EVB and VSI discovery protocols.

Transport Protocol

- For those not at the Austin meeting, I argued that the benefit of a transport protocol has *not* been demonstrated;
- It has been argued that a transport protocol is useful because it can provide bandwidth efficiency and prevent buffer overrun when multiple bind/unbind requests are processed within a short time window;
- It seems, however, that this is *exactly* the behavior that a hypervisor (or other controller) would want to avoid;
- In what case does it benefit the hypervisor to start multiple VMs on a *single* physical server when it could distribute these requests to multiple physical servers?
- Sending multiple requests to a single physical server serializes the start-up of the VMs and creates significant latency;
  - This would *certainly* be the case in recovery scenarios which have been cited as the key motivation for the transport protocol;
- Simple fixed-window flow control can be deployed in BCP to avoid buffer overrun in those cases where requests are received within a short time interval;
- Thus, it continues to be unclear to my why people are anxious to introduce a new transport protocol;
- The consensus view is that a transport protocol should be deployed; while I disagree with this consensus view I will not argue the point further (as I said in Austin);
- It should be understood that the question of whether or not to deploy a transport protocol is completely orthogonal to the question of whether (a) BCP or (b) the combination of EVB discovery and VSI discovery is deployed;

Key points

- LLDP Link Capabilities TLV used on link and on channel (which behaves like link);
- *Channel management* and *VSI management* share many common features
  - Bind establish/release, request/response, positive/negative response, unsolicited release;
  - Leverage common features using ‘Bind Control Protocol’ (BCP)
- The issue of using, or not using, a ‘transport’ protocol is independent of whether evb Channel Management and VSI Management are deployed (a) using distinct protocols (EVB Discovery and VSI Discovery) or (b) using a common Bind Control Protocol;