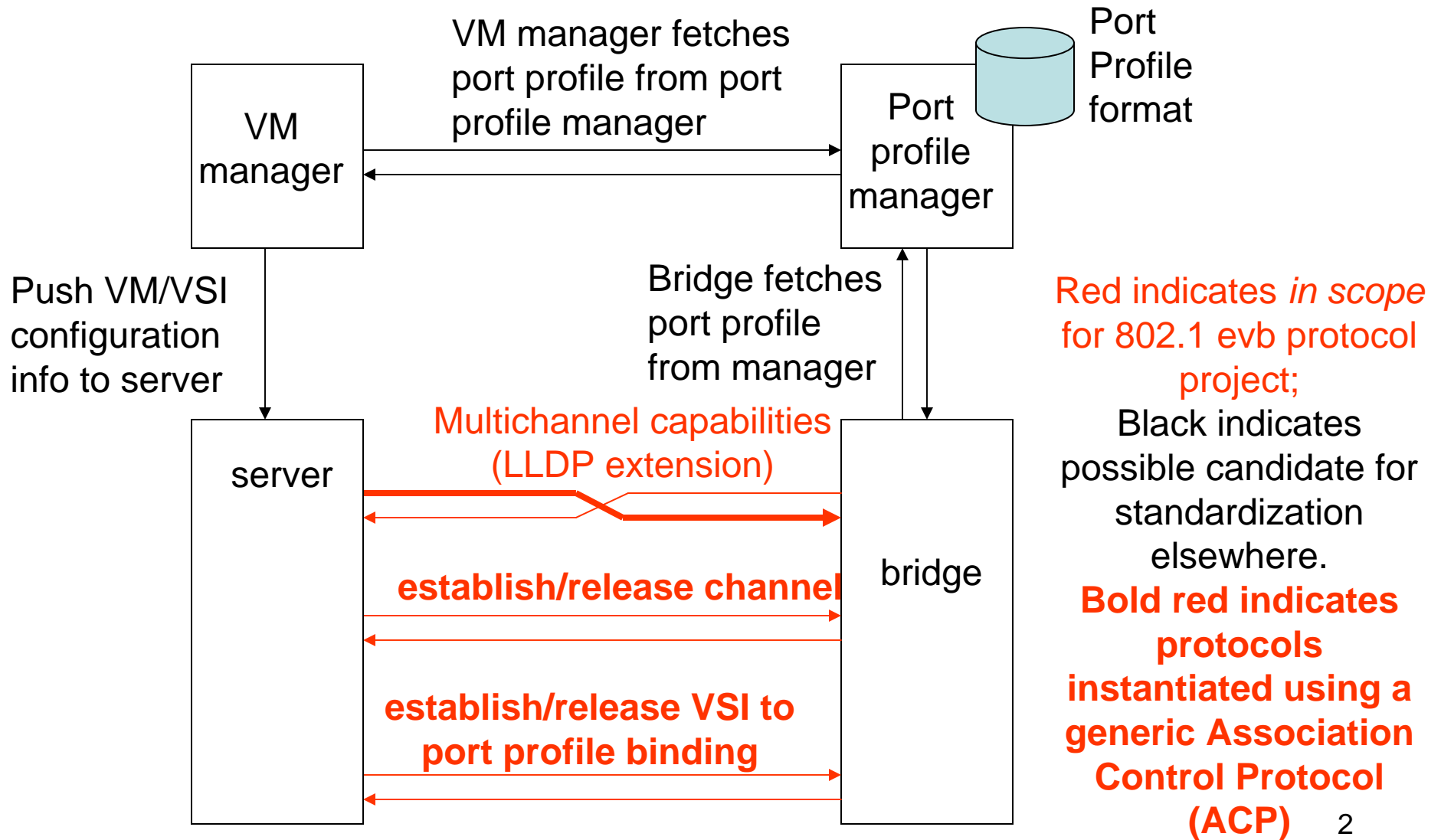


Association Control Protocol (ACP)

Bob Sultan (bsultan@huawei.com)

Ben Mack-Crane (tmackcrane@huawei.com)

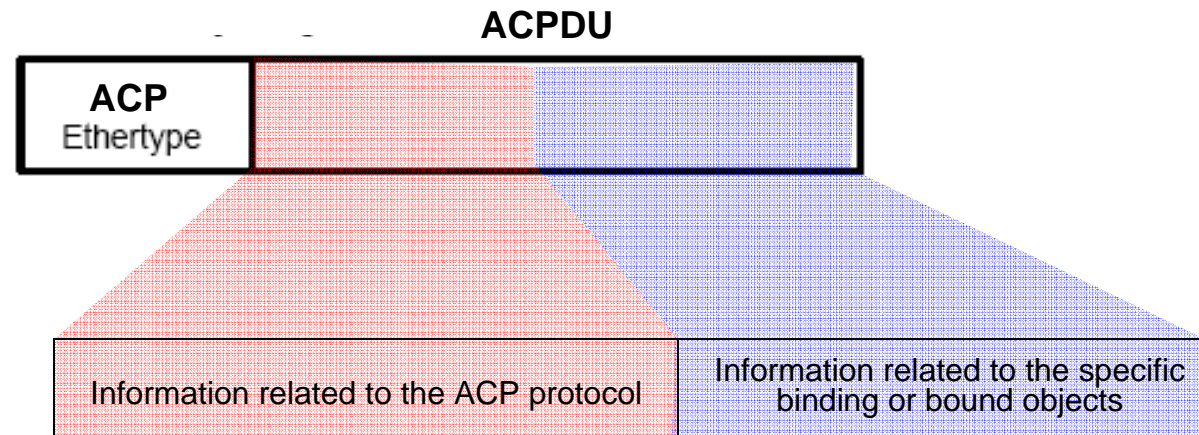
evb Protocols



Purpose/characteristics of a generic ACP

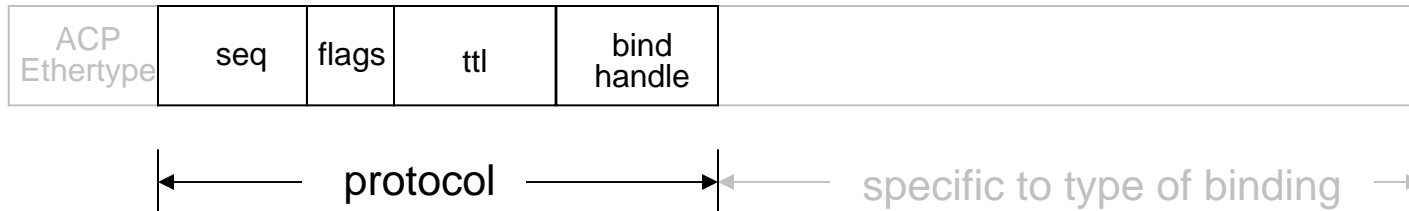
- Requester (server) asks responder (bridge) for a binding to be established or released;
- Responder notifies requestor if an established binding is released;
- The bind establish request, the bind release request, and the notification always requests in the recipient sending an ack or nack;
- All requests are acknowledged by a response;
- A method is specified to correlate a response with a request;
- The bind establishment can be achieved in stages:
 1. The responder assembles information needed to fulfill the bind establishment;
 2. The responder reserves resources needed to fulfill the bind establishment;
 3. The bind establishment is committed;
- The identity of the bound object can be specified by the responder based on *type* information supplied by the requestor;
- A binding is associated with a TTL in order to ensure that resources are always eventually freed (idempotency); the requestor must 'renew' leases periodically;
- The bind establishment can be accompanied by 'auxiliary' information that can be associated with the binding;
- *The ACP is independent of the particular object types that are bound;*
 - E.g., same rules apply to Channel Management (Channel to Physical Link binding) as to VSI to Port Profile binding.

Structure of the ACPDU



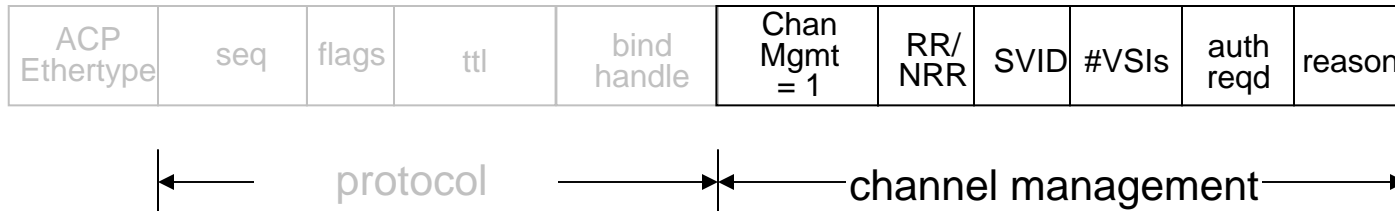
- One portion of payload specifies information related only to the ACP protocol;
- The other portion specifies information associated with the specific elements being bound;

ACPDU: protocol portion



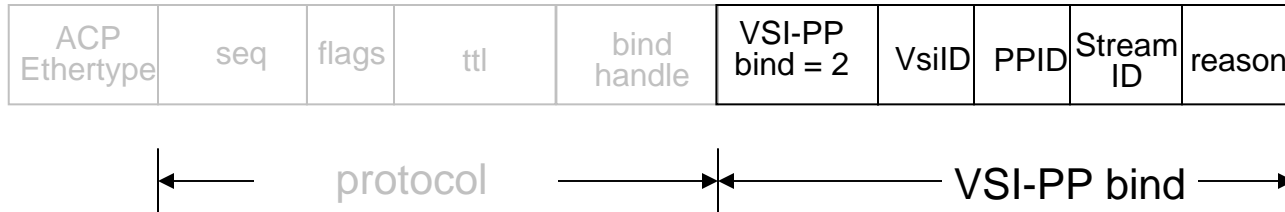
- **Sequence number (2B):** allows correlation of request with response;
- **Flags (1B):**
 - **req/rsp (1b):** identifies request or response;
 - **ack/nack (1b):** indicates whether a reply is positive or negative;
 - **est/rel/nfy (2b):** indicates whether message is related to establish, release, or notification of unsolicited release;
 - **fetch/rsrv/cmit (2b):** indicates whether the establishment is to proceed (1) only as far as fetching information required to initiate establishment, (2) only as far as reserving resources required for establishment as specified by the fetched information, or (3) to completion;
- **ttl (2B):** Time after which responder releases association (supplied on establish request);
- **bind handle (2B):** Unique identifier by which an established binding can be referenced; this value allows correlation of establish and release (for discussion at a later time; we could do without this);

ACPDU: channel mgmt. specific portion



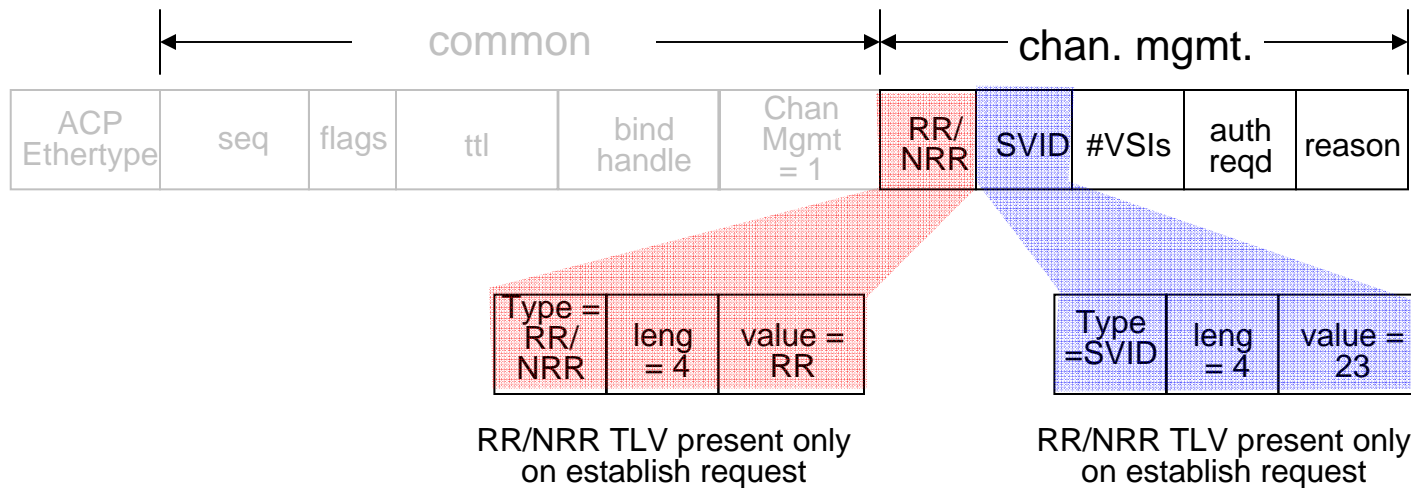
- **RR/NRR:** specifies whether requested channel should or should not be set for reflective relay;
- **SVID:** identifies the allocated channel; specified on establish response;
- **#VSIs:** number of VSIs that can be associated with the channel (under discussion);
- **reason:** present in negative response;

ACPDU: VSI-bind specific portion



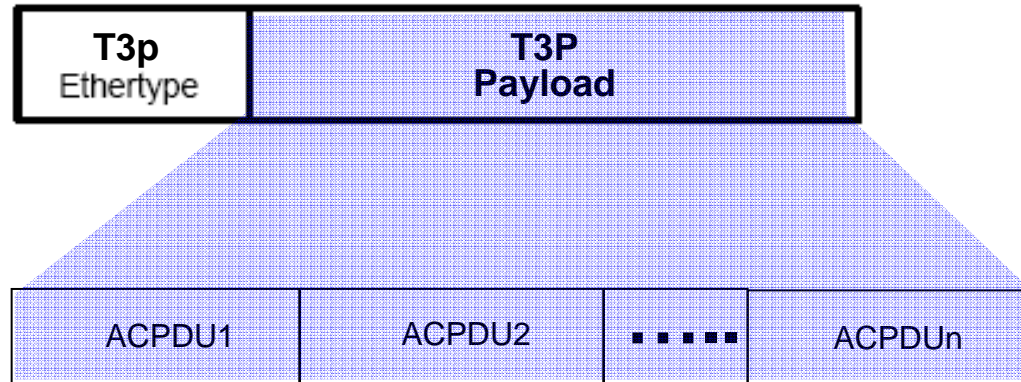
- **VsiID:** identifies the VSI to be bound;
- **PPID:** identifies the Port Profile to which the VSI is bound;
- **Stream ID:** identifies the stream associated with the VSI (optional);
- **reason:** present in negative response;

Where do TLVs come in?



- The question of whether you represent fields within the PDU by simple values or by TLVs is entirely orthogonal to the protocol;
- Some of the type-specific fields appear only in, for example, establish request, or negative response (e.g., reason);
- Some fields are optional (e.g., stream ID);
- Any field, or set of fields, that appears *conditionally* can be represented by a TLV in order to reduce the size of the PDU;
- Further, a variable length field can be represented by a TLV to simplify parsing;

T3P allows multiple PDUs per frame



- We never put multiple data or control PDUs in a single Ethernet frame;
- Why would we do this for ACP?

For Discussion

- What, exactly, is the benefit of putting multiple transactions into the same PDU?
 - The saving in MAC header overhead is tiny;
 - There is extra processing overhead in ‘packing’ the transactions;
 - A ‘reply’ will in general not contain the same transactions as the ‘request’;
 - Except at startup, the frequency of transactions is low enough to make it unlikely that multiple transactions will be packed into a single PDU;

Proposal

- A generic protocol for establish/release of bindings (ACP)... related info in protocol portion of PDU;
- Specific instances of binding activity... related info in bind specific portion of PDU;

Comparison

LLDP	MRP	ACP
One-way advertisement; no reply supported	May be able to use this	Allows each binding to be ack'd individually
All info must be in single frame	But probably results in a very complex MRP application for a relatively simple function.	Many PDUs at startup or restart
Can get around single frame limitation by using T3P (but adds complexity)		Exact fit for evb binding protocols and could be used for other protocols

NOTE—The LLDP protocol is designed to advertise information useful for discovering pertinent information about a remote port and to populate topology MIBs. It is not intended to act as a configuration protocol for remote systems, nor as a mechanism to signal control information between ports. During the operation of LLDP, it may be possible to discover configuration inconsistencies between systems on the same IEEE 802 LAN. LLDP does not provide a mechanism to resolve those inconsistencies. Rather, it provides a means to report discovered information to higher layer management entities.... From .1AB