IEEE 802.3as
Frame Expansion forward compatibility

May, 2005

Boris Fakterman, Intel

boris.fakterman@intel.com

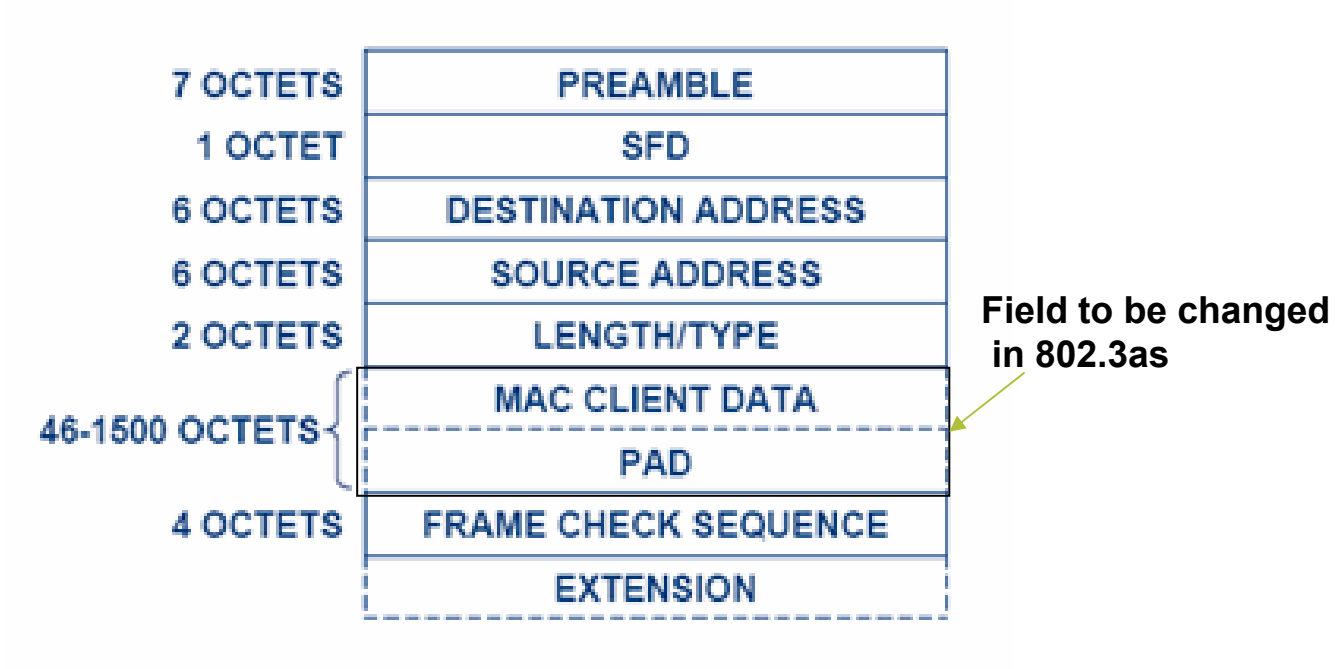
Forward compatibility problem statement

- The lack of format definition of the 802.3as extension field may create forward compatibility issues and accelerate obsolescence of MAC Hardware
- In the next slides
  - The forward compatibility problem is presented
  - Solution is proposed

Layer 2 payload – A practical status

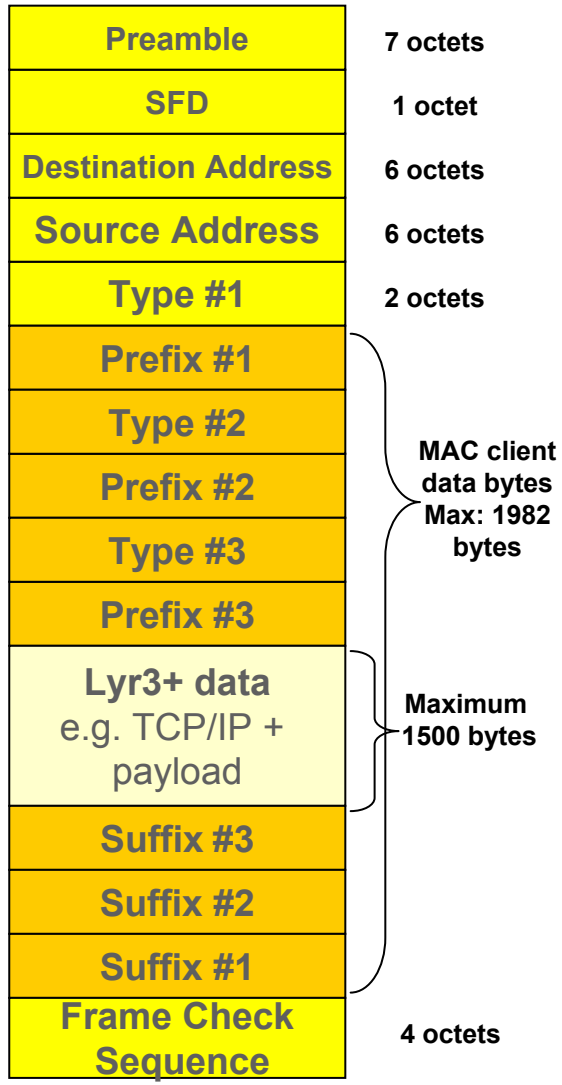
- End node applications *assume* Layer3-4 payload is visible and processed in the NIC. E.g.
  - S/W/ H/W system optimization:
    - e.g. Receive Side Scaling (RSS) – required by Microsoft –
    - Directs packets using 5-tuple information to different process threads
  - Out-of-band functions –e.g. manageability:
    - Sanctioned by industry standards: IPMI, ASF, etc, in the DMTF
    - Direct packet to OOB channel using MAC or Layer 3,4 filtering toward "always-on" management processor. e.g. BMC's.
    - Allow system management in any OS health or power down state.
  - System stack optimizations
    - Acceleration under Microsoft Chimneys;
    - NIC Performance enhancements implemented in OS use for a decade - E.g. Checksums, header/data splitting, TOE
    - Assume Layer 3,4 process on the NIC.
- Ethernet = low cost
  - Implementation at line speed in Firmware is possible but expensive;
    - May be used in some generations of products–
    - But should <u>not</u> be assumed by standards as a practical pre-requisite

802.3 Basic frame



802.3as frame

- MAC client data bytes may include encapsulation of several protocols / extensions
- Each protocol may include different length prefix & suffix ; the length may vary from frame to frame
- Some protocols, like MACSec may change the encapsulated data
- Current frame proposal does not contain information about the encapsulation:
  - Lyr3 payload is not locatable & readable without knowing all types & prefixes



Implications of the loose formatting

- Current frame proposal does not contain information about the encapsulation:
  - Lyr3+ payload is not locatable & readable without knowing all types & prefixes
  - New protocol/extensions must be implemented in H/W to allow Lyr3+ payload location – and allow all Lyr3 dependent MAC based applications
  - There is no easy option to retrofit installed based, 802.3as MAC's with new protocols in Software.
  - Deployment of new protocols becomes an unnecessarily complex problem for IT.
- **A solution that allows forward compatibility of the 802.3as installed base to future protocols is essential for the success of the new protocols.**

IT VLAN experience

- Intel IT describes VLAN installation experience
  - IT did not encounter special problems, because
    - they made sure they used single vendor equipment
    - VLAN use was, and in most cases still is, not reaching end nodes

Solution – small extension of 802.3as scope

Current 802.3as

- 482 additional bytes
- General structure (prefix/suffix)
- 1 Ethertype in extension field

New Proposal

- 482 additional bytes
- General structure (prefix/suffix)
- Standardized 802.1 handle:
  - Extension Ethertype
  - Extension Descriptor

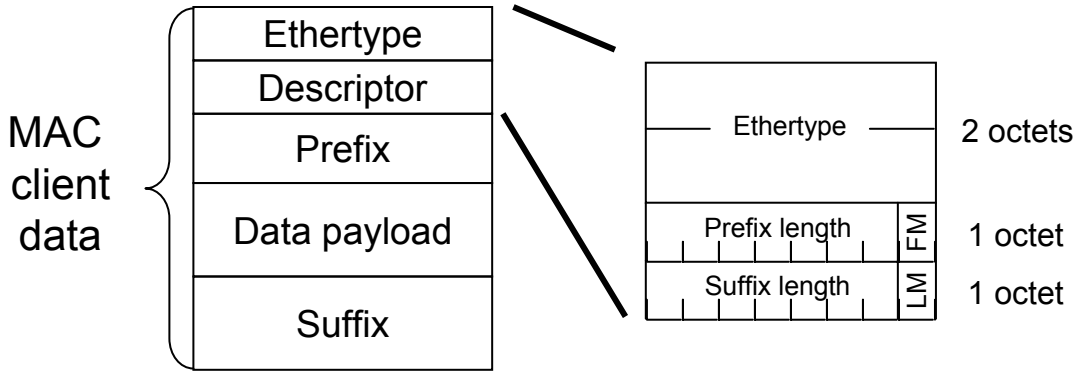
Note:
• Ethertypes are the “handle” between 802.1 and 802.3 today.
• Defining that all extensions start with an Ethertype is merely standardizing the de-facto status

Handle general structure

- Proposed extension handle general structure
  - Ethertype
  - Prefix length
  - Suffix length
  - Bit indicating that extension function modifies fields encapsulated by the extension
  - Bit indicating that extension function modifies the length of the fields encapsulated by the extension.

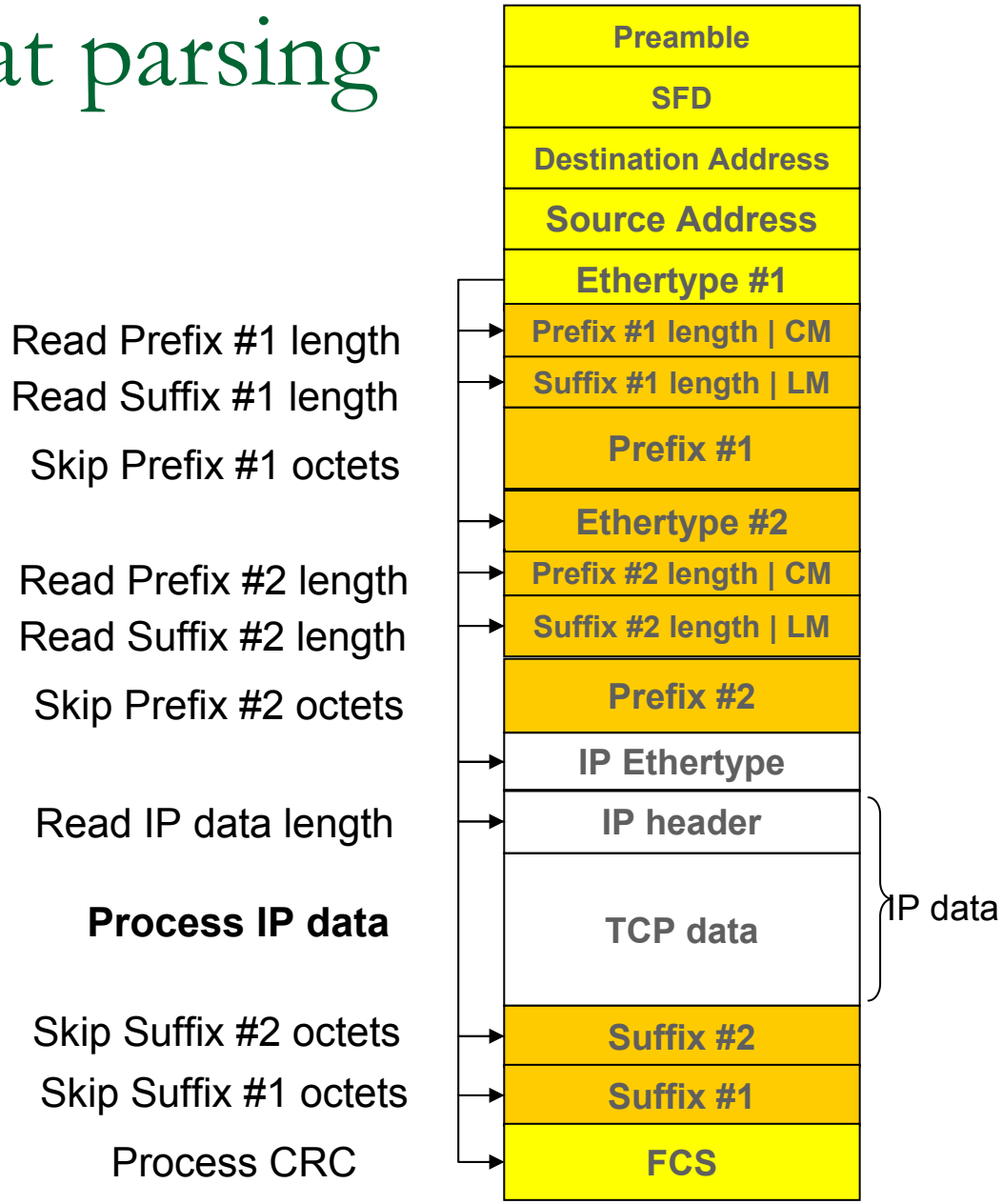
Extension general structure

- Each separate extension shall start with an Ethertype and a two octet descriptor field adjacent to it. The extension shall contain even number of octets.
- Following format shall be applied for the Ethertypes inside a data field



Prefix length – 7 bits indicating number of prefix two-octets in the Ethertype, starting after the suffix length indication octet.
Suffix length – 7 bits indicating number of suffix two-octets in the Ethertype
FM – when set, this bit indicates that extension function modifies fields encapsulated by the extension
LM – when set, this bit indicates that extension function modifies the length of the fields encapsulated by the extension

Proposed format parsing



Back Up

Current data field definition indicates data field contents

- Current chapter 3.2.7 Note

NOTE—The envelope frame is intended to allow inclusion of additional prefixes and suffixes required by encapsulation protocols such as P802.1ad, P802.1ah, P802.1AE and MPLS. These frames will contain at least one Ethertype within the data field in addition to the type field. However, these additional types and encapsulation details are not visible to this standard. If present, they follow the MAC Type field and are carried as MAC Client Data. Note that the original client data must not exceed 1500 bytes which is its size in the basic frame.

- "Length/type field inside data field" comment

Cl 03 *SC* 3.2.7 *P* 12 *L* 24 *Comment #* 18

Squire, Matt Hatteras Networks

Comment Type E *Comment Status* X

Maybe just a nitpick, but do the encapsulation protocols have to contain at least one Ethertype within the data field? Example: MPLS may encapsulate TDM via pseudowires - can it use a 2K frame size? MPLS may encapsulate a frame thats length encoded on the inside - it won't have an inner Ethertype (though it will have an inner length / type field).

Suggested Remedy

Clean up that sentence a little.

Response *Response Status* W

Accept. This is a nitpick. Reword to indicate that another length/type field is in the data field and it is typically an Ethertype.