Topics for joint 802.1/802.3 Technical Plenary

Tony Jeffree

March 2004

4 Main Topics:

- Maximum Frame size issues relative to 802.1ad, 802.1AE;
- Relative placement of Link Agg and MACSec;
- Minimum frame size issues relative to 802.1AE;
- 802.1AB issues

Frame Size Issues related to 802.1ad and 802.1 AE

History - 1

- Previous discussion between 802.1 & 802.3 in July 2003 re max frame size impact of these projects
- At that time, 802.1 had not characterized the potential frame size impact of P802.1ad and P802.1AE on 802.3 frames
- 802.3 made a plea for a “one off” request rather than piecemeal requests as new requirements arose/were clarified

History - 2

- 802.1 has been working these issues; the most difficult to characterize has been MAC Sec due to the technical issues/complexities/obscurities surrounding cypher suite choice
- Only now is it possible to make a coherent “pitch” to 802.3 re the requirements for these projects
- A couple of related issues also need discussion

TOPIC 1: Frame Size Expansion Requirements (as currently known)

- MACSec Secure Frame Format – 24 octets (point to point), 32 octets (shared medium)
- Provider Bridge TAG – 4 octets
- Total possible for mandatory secure cipher suite:

  32 (Customer security) plus

  32 (Provider security) plus

  4 (Provider TAG)

Caveats:

- Possible use of cipher suites to meet Federal requirements – 64 octets
- Larger cipher blocks for greater security – 160 octets
- Requests for larger Provider TAG ann duplicate FCS (yet to be resolved)

The wider frame size problem

- Whatever we define in 802, others (e.g., MPLS) will choose to increase the frame size further. There is not a great deal we (ether .1 or .3) can do about that.
- We are assuming that legacy protocols (e.g., Appletalk) will continue to ship 1500 octet frames
- Any “oversize” frames will be limited to IP
  - Therefore any fragmentation/reassembly offered at L2 would not be useful as this would be better handled above L2

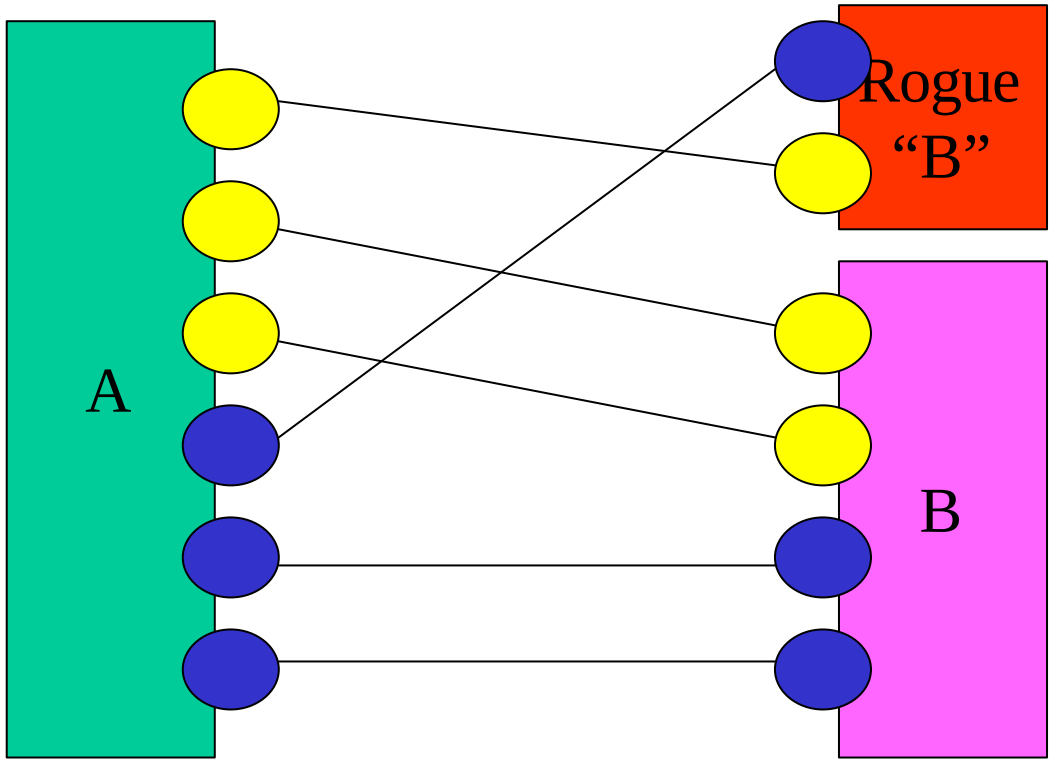
TOPIC 2: Relative Placement of Link Agg & MACSec

- 802.3 didn’t want an embedded (within MAC) solution to MACSec
  - Looks like this was the right decision
  - However, some problematic architectural issues: MACSec may well need to operate below LinkAgg

Why below Link Agg?

- Goal of MACSec to confine/localize DoS attacks
- Having Link Agg under MACSec would allow additional attacks (spoofing aggregation membership, for example) as LACP would be in clear
- ➔ MACSec must be placed below LinkAgg to remove these DoS opportunities

DoS attack opportunity…



What 802.1 should do in MacSec with 802.3's explicit knowledge

- Document the placement of MACSec as being below LinkAgg in the Bridge Port's "MAC Stack"

The longer term plan

- Continue to work with 802.3 to converge the 802.1 ISS, the 802.3 MAC service, and the P802.1AC MAC Service definition
  - 802.3 service used to lack the SA; this has now been fixed
  - ISS currently has user priority and access priority – largely a hangover from 802.4 and 802.5 MACs – can and should be reduced to a single parameter
  - Local “return codes” seem to have disappeared from 802.3 some while ago
  - Once we have service convergence, ensure the management view fits together properly
- This may take some time; in the meantime it is clear what the protocols need to do.

TOPIC 3: Minimum frame size problem - 1

- No explicit length in Ethernet (Type interpretation) frames:
  - For large frames, length is recovered from physical frame length
  - For small (minimum frame length) frames, determination of the actual number of user data frames is possible only by the recipient protocol entity
- MACSec protected frames carry an ICV trailer (after the user data):
  - With current MACSec proposal, minimum sized secured frames can contain 0-28 octets of PAD
  - Padding likely to be (but not necessarily) applied after MACSec
  - For some combination of minimum frame length and ICV length, the ICV position may therefore be indeterminate
  - Therefore, need a user data length indication (more strictly, an ICV position indication)

Minimum frame size problem - 2

- Remedy: Indicate the position of the ICV in short frames (less than 63 octets)
  - 6 bits available for this (see frame format earlier)
  - 1 bit to indicate “Get length from physical frame size”
  - 5 bits to indicate explicit length/ICV position

TOPIC 4: Issues related to 802.1AB

802.3-related TLVs

- P802.1AB is heading rapidly in the direction of Sponsor Ballot
- The current draft contains a section that defines 802.3-specific TLV definitions
- We have alerted 802.3 to the existence of these definitions
- We need to be sure that these have had, or will receive, adequate review by 802.3

“EtherType”

- One of the ballot comments on P802.1AB pointed out that this term (which is already in common usage), or an agreed variant of it, needs to be standardized
- We believe the right place to do this is in the 802.3 standard
- It would then be appropriate for 802, 802.1AB, …etc. to reference that definition
- The IEEE Registration Authority web pages also use the term; they should make use of whatever term is agreed for insertion into 802.3

BACKGROUND: 802.1ad

- Current draft makes use of “Q IN Q” tagging – involves addition of a 4-octet Provider tag (will use a new Ethertype, slight differences in field semantics – for example, no need to identify Canonical/Non-canonical format)
- If usable payload is to stay at current size of 1500 octets (highly desirable) then an increase in frame size of a further 4 octets would be necessary

802.1ad contd…

- There has been a proposal for a larger provider tag – discussion on this is as yet unresolved
- Inclusion of an additional FCS has also been proposed; again, discussion on this is as yet unresolved.

Scope of the problem

- Similar problem as with the original Q tag:
  - Potential incompatibility with existing equipment
  - Difficulty of reducing payload to allow for increased tag size
- However:
  - Unlike the Q Tag, the Provider tag doesn’t affect existing (customer) LAN installations
  - Provider tags added/removed by boundary devices at the customer/ provider network boundary
  - Therefore, problem is confined to those boundary devices and any equipment used in the core of the provider network

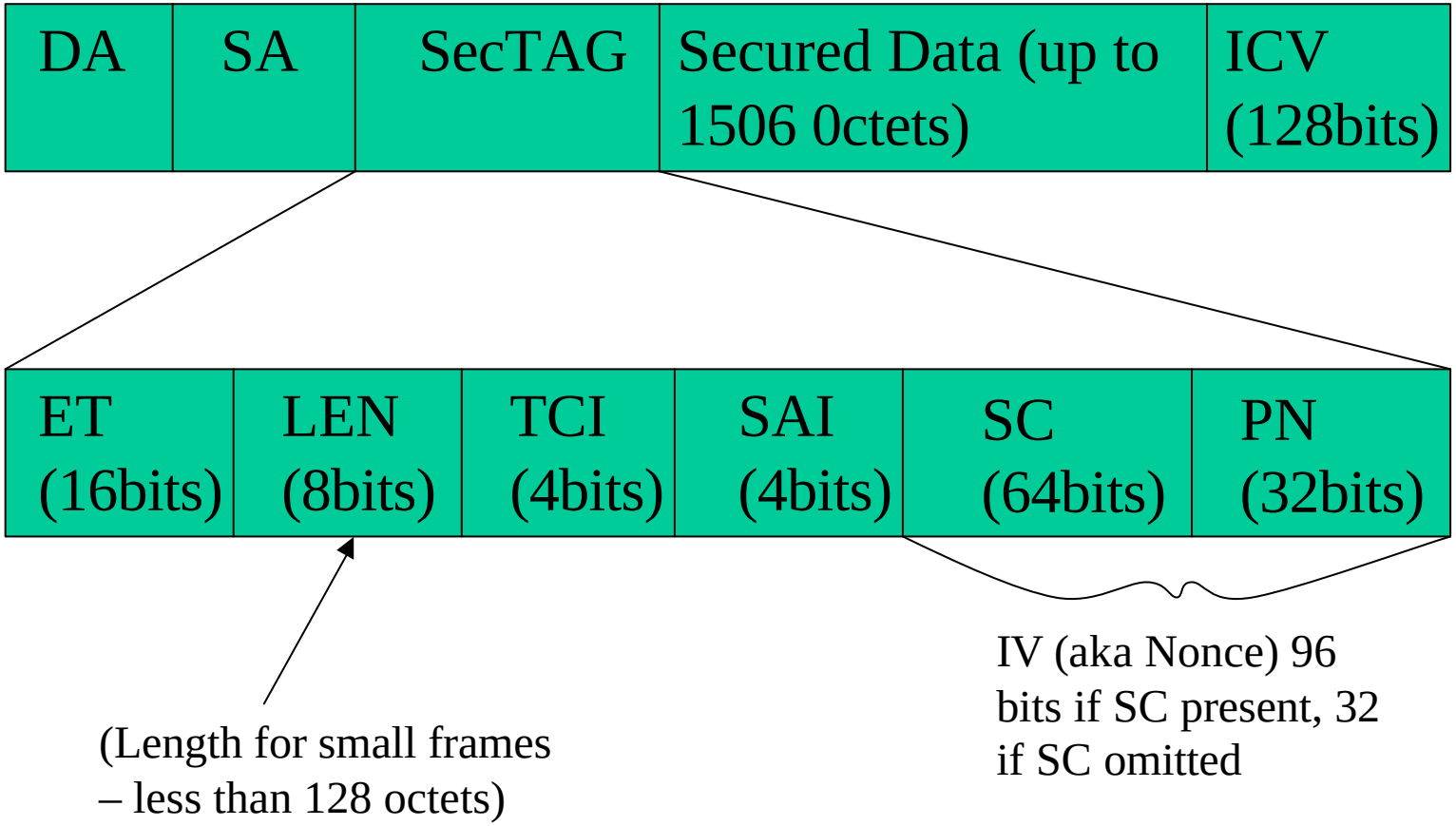
BACKGROUND: 802.1AE

- Three issues:
  - Inclusion of security header etc. information leading to need for increased max frame size
  - Minimum (Ethernet) frame size problem
  - Relative placement of Link Aggregation and MAC Security

Secure frame format - 1

- Current draft calls for a secure frame format consisting:
  - A security tag (SecTAG) prepended to the secured data
  - The secured data
  - An integrity check value (ICV) appended to the secured data

Secure frame format - 2



Secure frame format - 3

- Bottom line – frame grows by 24 octets for point to point and EPON, 32 octets for shared media.
- Sizes assume use of GCM as the cryptographic suite (current assumption); may be necessary to extend the PN field to support other suites for Federal use.
- Some other cryptographic suites (not CGM) result in the data field growing – possibly by the encryption block size +1 (grows to nearest 16 octet boundary + a further 16 octets)
- Some block cyphers (AES-512) that are already specified would require further extension – potentially 160 octets

Scope of the problem

- Again:
  - Potential for incompatibility with existing equipment
  - Difficulty of reducing payload to allow for increased tag size
- BUT use of this frame format is negotiated between stations on a single LAN segment:
  - If the negotiation fails, normal frames are used
  - Unlikely to be used as a software add-on to existing equipment for performance reasons
  - Therefore problem is therefore largely restricted to existing equipment “seeing” oversized frames on shared media segments, which they can/should discard anyway

BACKGROUND: Minimum frame size problem

- Obvious solution would be a length for the entire Ethernet user data
  - Consumes 2 octets
  - Might be possible to add these without increasing SECTag size (but unlikely)
  - Might result in arbitrary constraints, e.g., that never would Ethernet frames exceed 4096 octets
  - Would throw user data off the 4 octet boundary
- Alternative remedy 1: Indicate the length of PAD
  - Assumes knowledge (in MACSec) of behaviour of lower layers
  - Would be problematic if/when more shims get invented

BACKGROUND: Why MACSec should remain MAC independent?

- Potential fragmentation of effort
- Now reaching “critical mass” in 802.1 to address this problem
- Media access independent mechanisms will be important in Provider Bridging scenarios

Why Link Agg should remain in 802.3?

- Link Agg was a significant effort
- In-built assumptions on timings when it comes to flow redistribution
- Costly in time, effort, and market confusion to remove it and re-label it as something else
- Time-consuming to ensure that nothing gets broken in transit

How 802.1 approached this with 802.1X & why we need a similar approach for MACSec

- 802.1X operates on individual links
- If links not aggregateable for security reasons, 802.1X tells system management to tell LACP to change the keys appropriately
- This behaviour is invisible to 802.3 as 802.1X doesn’t do encryption or modify data frames
- With MACSec, secure frames will need decryption
- Depending on placement of MACSec, this needs different numbers of SAs (Secure Associations) and keys
- Preferred method is LA over MACSec; consequence is that LACP frames (but not PAUSE frames) are encrypted
- Explicit choice is a requirement for interoperability

What we should NOT do with Link Agg

- Link Agg interfaces are for practical purposes media access independent (802.1 terminology) interfaces
- Shouldn’t attempt to make it more so by requesting a change in committee/document placement - small practical benefit in doing this
- Should not change how a Bridge uses Link Agg – the status parameters etc. were carefully constructed, we don’t need to introduce any errors here, and sliding MACSec underneath shouldn’t affect this.