PFC round trip timing protocol

Design requirements and goals
Protocol and algorithms
Addressing, identification, and data
State machines
Packet formats, decoding and validation

Mick Seaman
mickseaman@gmail.com

1. Idea/suggestions, follow up from Lily Lv’s prior work and discussion.
Design requirements and goals

• Determine PFC headroom upper bound
  — Max data octets to be stored after PFC transmission
  — = (Data reception cessation delay * link speed) + max frame size
  — = Measured round trip delay*link speed + max frame size
  — Accuracy ~ max frame size

• Include PFC transmitting system and peer interface stack delays, even when not known to system implementor.¹

• Not disruptive to network operation if neighbor does not implement without need for management (protocol always confined to single link, not uselessly persistent).

• Not dependent on 802.3 Reserved Address use.

• Operation independent of link delay, w/o need to manage timers.

• Do not constrain local timer/time base/time representation.

¹ A real system can comprise multiple component’s from different sources, and may be configured in a number of ways. PFC headroom optimization should not depend on complete system knowledge.
Protocol and algorithms

Protocol, a set of rules that communicating participants obey and depend on to exchange information and operate. For this ‘Query/Response’ round trip measurement protocol:

- Respond promptly (~ PFC recognition and tx cease stop time) to each Query
- Copy peer’s time data in Query to Response (not need to understand time)
- If capable of sending Queries will Respond
- Limit Query frequency (minimum interval between Queries)
- Send (some number of, within bounded time) Queries when initialized

Additional algorithms (for this protocol), agreement not required for protocol operation:

- Smoothing of round trip measurement delays.
- Administrative values (initial round trip assumption, acceptable upper and lower bounds as reflected in buffer headroom allocation.)
Addressing, identification, and data

- **MAC DA**: nearest-bridge (TPMR) Reserved Address
  - PFC (and Pause) operates physical link-by-link (scoped by 802.3 MACs)
  - Back-pressure pausing between C-VLAN Bridges connected by TPMRs is concatenated: Bridge to TPMR, TPMR to TPMR, TPMR to Bridge
  - No 802.3 Reserved MAC address - no 802.3 type allocation reqd.

- **EtherType**: 802.1Q Congestion Isolation (CIM) [Clause 49], followed by 4-bit Version (send >1, ignore on receipt), 4-bit subtype (1, indicates round trip measurement msg, RTM).

- **Query Time Stamp** (present iff the RTM is a Query, not interpreted by recipient, in some format).
  - + Query Adjustment (optional, not interpreted by recipient)

- **Reflected Time Stamp** (copy of rcved Query Time Stamp, present iff the RTM is a Response).
  - + Reflected Adjustment
  - + Responder’s Response Delay Adjustment in nanoseconds (?)
State machines (1)

BEGIN || !portEnabled || !rtmEnabled

 INIT

  myStamp = 0; myAdj = 0; myAdjReflected = 0; rcvdResponseDelay = 0;
  peerStamp = 0; peerAdj = 0; responseDelay = 0;
  queryAfterTicks = jitterTicks; queryWhenTicks = jitterTicks; retryCount = 3;

  ((queryWhenTicks == 0) || (peerStamp != 0))

 TXQ

  myStamp = localTime; myAdj = localAdjustment(); responseDelay = localResponseDelay();
  txQuery(myStamp, myAdj, peerStamp, peerAdj, responseDelay);
  peerStamp = 0; peerAdj = 0; responseDelay = 0;
  queryAfterTicks = minQueryIntervalTicks; queryWhenTicks = maxQueryIntervalTicks;
  retryCount = retryCount - 1;

  (retryCount != 0) && (queryWhenTicks == 0) && (peerStamp != 0) && (queryAfterTicks == 0)

  (retryCount != 0) && (queryWhenTicks == 0)

 TXR

  txResponse(peerStamp, peerAdj, peerStamp, ResponseDelay);
  initializeTimeStampData();

  (retryCount != 0) && (queryWhenTicks == 0)
State machines (2)

BEGIN || !portEnabled || !rtmEnabled

INIT

defaultHeadroom();
myStampReflected = 0; myAdjReflected = 0; rcvdResponseDelay = 0;

myStampReflected != 0

RXR

calculateHeadroom(localTime, myStampReflected, myAdjReflected, rcvdResponseDelay);
retryCount = 3;

Response Processing Machine