802.17 presentations

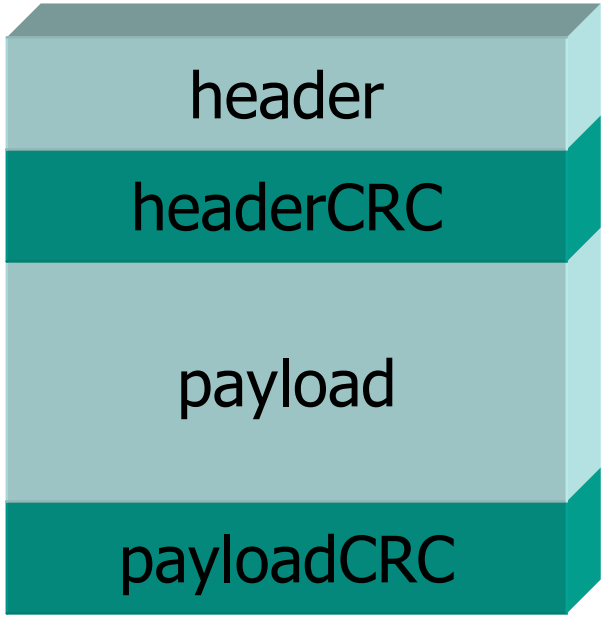
- Prepared for 802.17, March 2002
- David V. James, PhD
Chief Architect
Network Processing Solutions
Data Communications Division
110 Nortech Parkway
San Jose, CA 95134-2307
Tel: +1.408.942.2010
Fax: +1.408.942.2099
Base: dvj@alum.mit.edu
Work: djz@cypress.com

CRC calculations

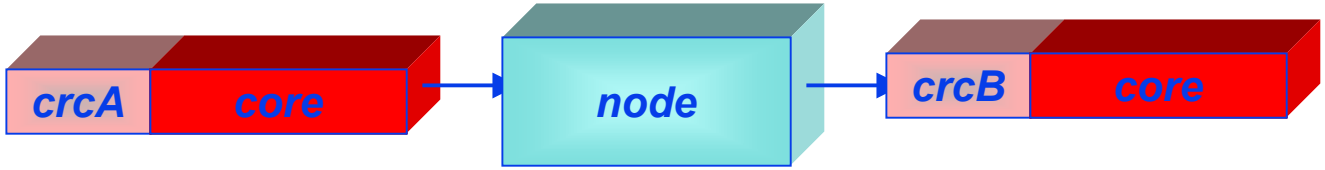
CRC processing

- Store&forward/Cut-through agnostic
- Invalid data is effectively discarded
  - store-and-forward discards
  - cut-through stomps the CRC
- Maximize error-logging accuracy
  - Separate header&data CRCs
  - “most” corruptions hit the data

Separate header and data CRCs

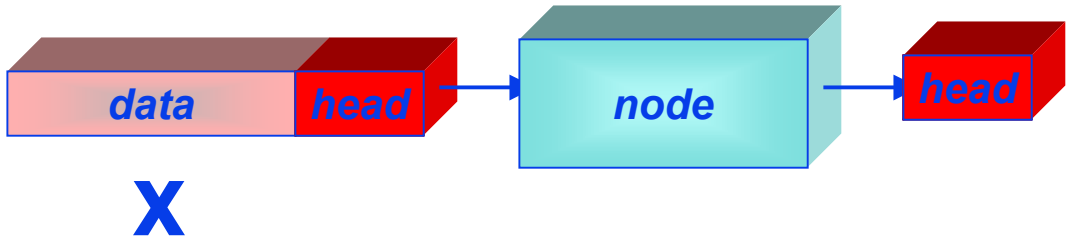


Cut-through CRCs

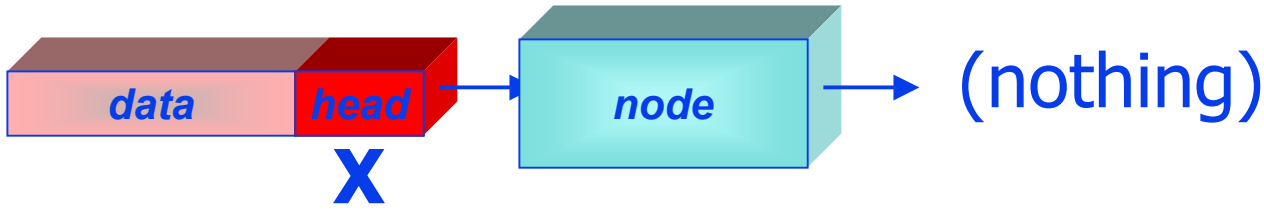


- Corrupted packet remains corrupted
- Error logged when first detected
- if (crcA!=crc) {
  errorCount+= (crcA!=crc^STOMP);
  crcB= crc^STOMP;
  }

Distinct CRCs reduces discards

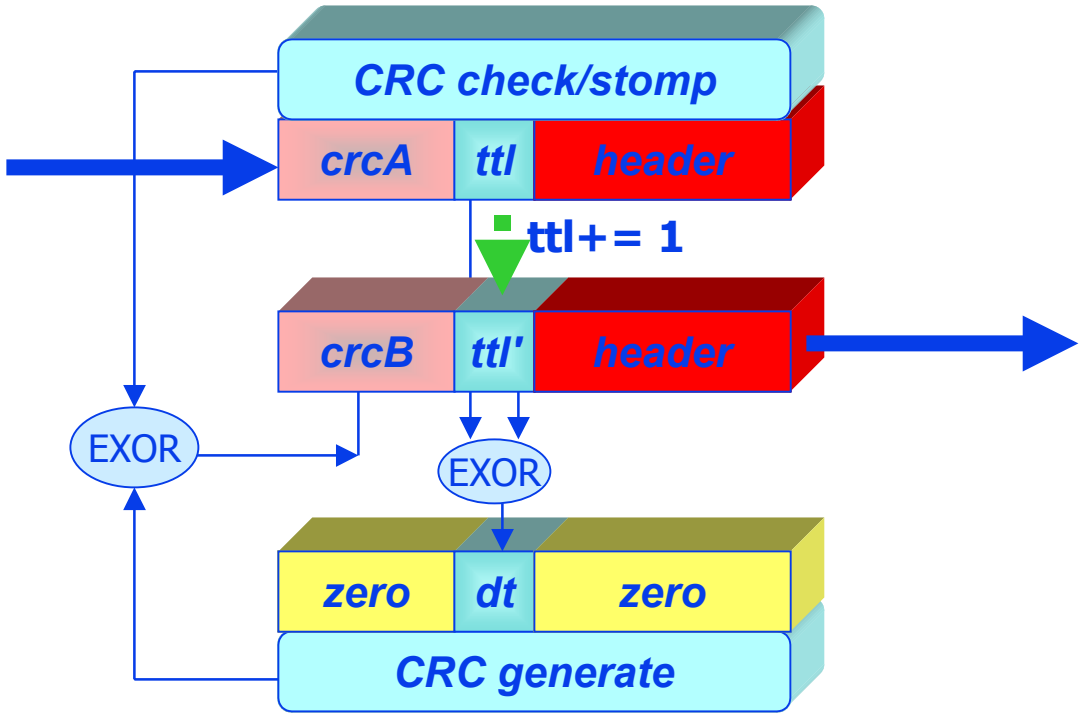


- Discard the corrupted data



- Discard the corrupted packet

End-to-end CRC protected TTL



CRC equation examples

```
a= c00^d00;   b= c01^d01;
c= c02^d02;   d= c03^d03;
// ...
s= c14^d14;   t= c15^d15;
c00= a^      e^  g^h^    m^
c01=   b^      f^  h^j^    n^
c02=     c^      g^  j^k^    p^
c03=       d^      h^  k^m^    r^
c04=         e^      j^  m^n^    s^
c05=           f^      k^  n^p^    t^
c06= a^      e^    h^        p^r^
c07=   b^      f^    j^        r^s^
```

CRC requirements

- CRC computation order
  - MAC optimized & PHY independent
  - PHY optimized and 2 MAC orderings
- Define the “stomp” value
  - For HEC as well as FCS
- CRC parallel computations
  - X8, x16, x32 are easily done
  - X64 is harder to print in “portrait”
  - ?? data values as C-code comments ??