802.17 presentations

- Prepared for 802.17, March 2002
- David V. James, PhD
  Chief Architect
  3180 South Ct
  Palo Alto, CA 94306
  Tel: +1.650.494.0926
  Cell: +1.650.954.6906
  Fax: +1.360.242.5508
  Email: dvj@alum.mit.edu

Time-to-live checks

- Prepared for 802.17, March 2002
- David V. James, PhD
Chief Architect
3180 South Ct
Palo Alto, CA 94306
Tel: +1.650.494.0926
Cell: +1.650.954.6906
Fax: +1.360.242.5508
Email: dvj@alum.mit.edu

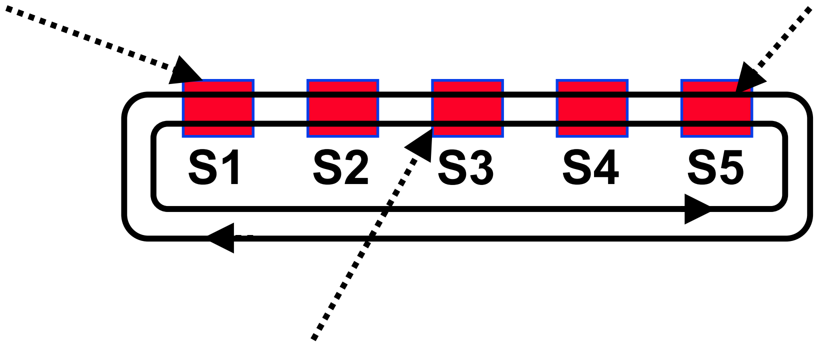
Time-to-live rationale

- Uniform aging protocol
  - Still supports 255 stations
- Simplified strip rules
  - No topology-database dependency
  - Deadlock eliminated
  - Less wrap-invoked discards
- Detailed test
  - dvj_Clause06_timeToLive_02.pdf

Time-to-live checks

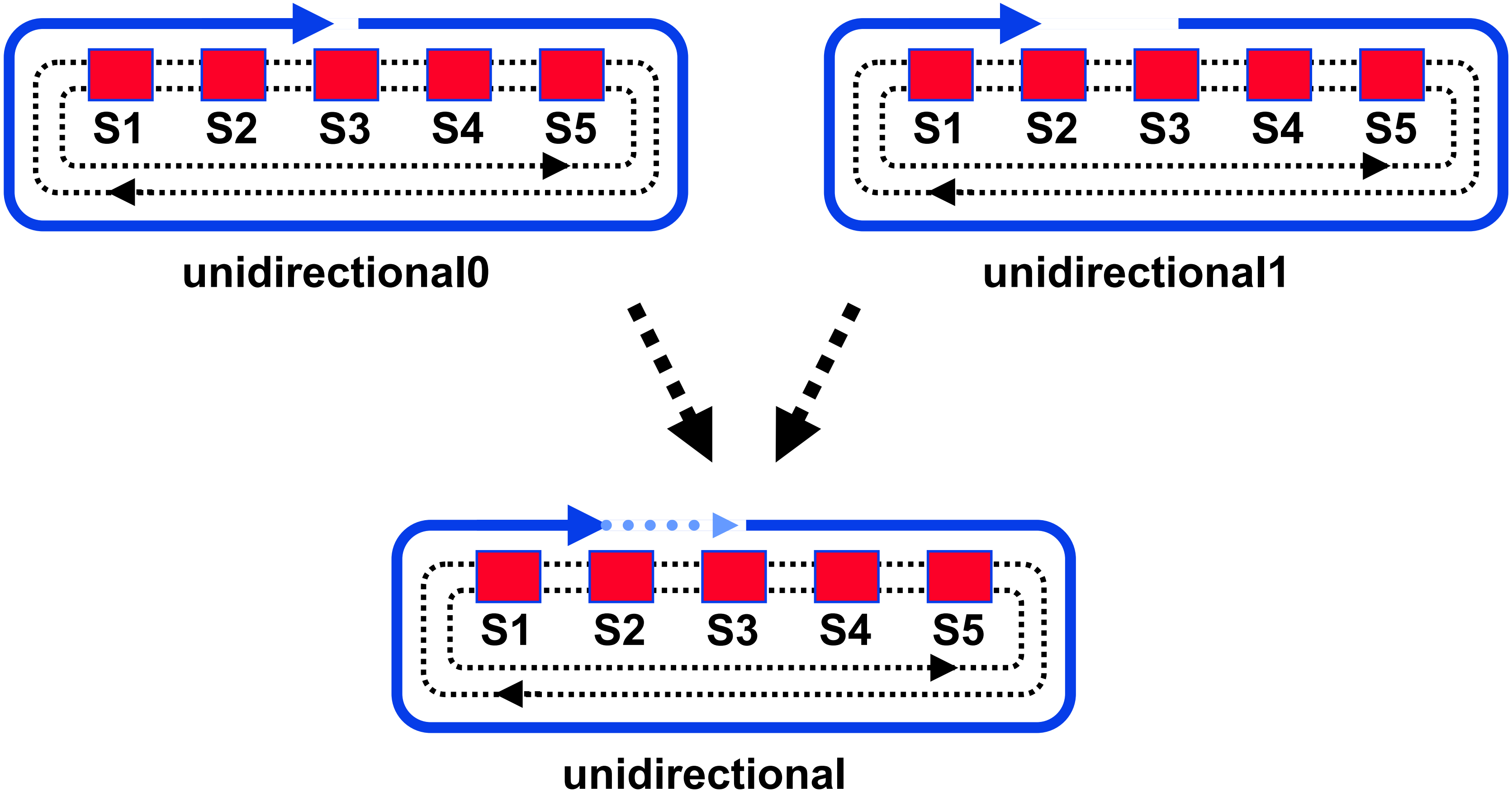
```
// (c) Second-wrap point
if (frame.we==0||frame.ps==0)
    discard= 1;
```

```
// (a) First wrap point
if (frame.we==0)
    discard= 1;
```

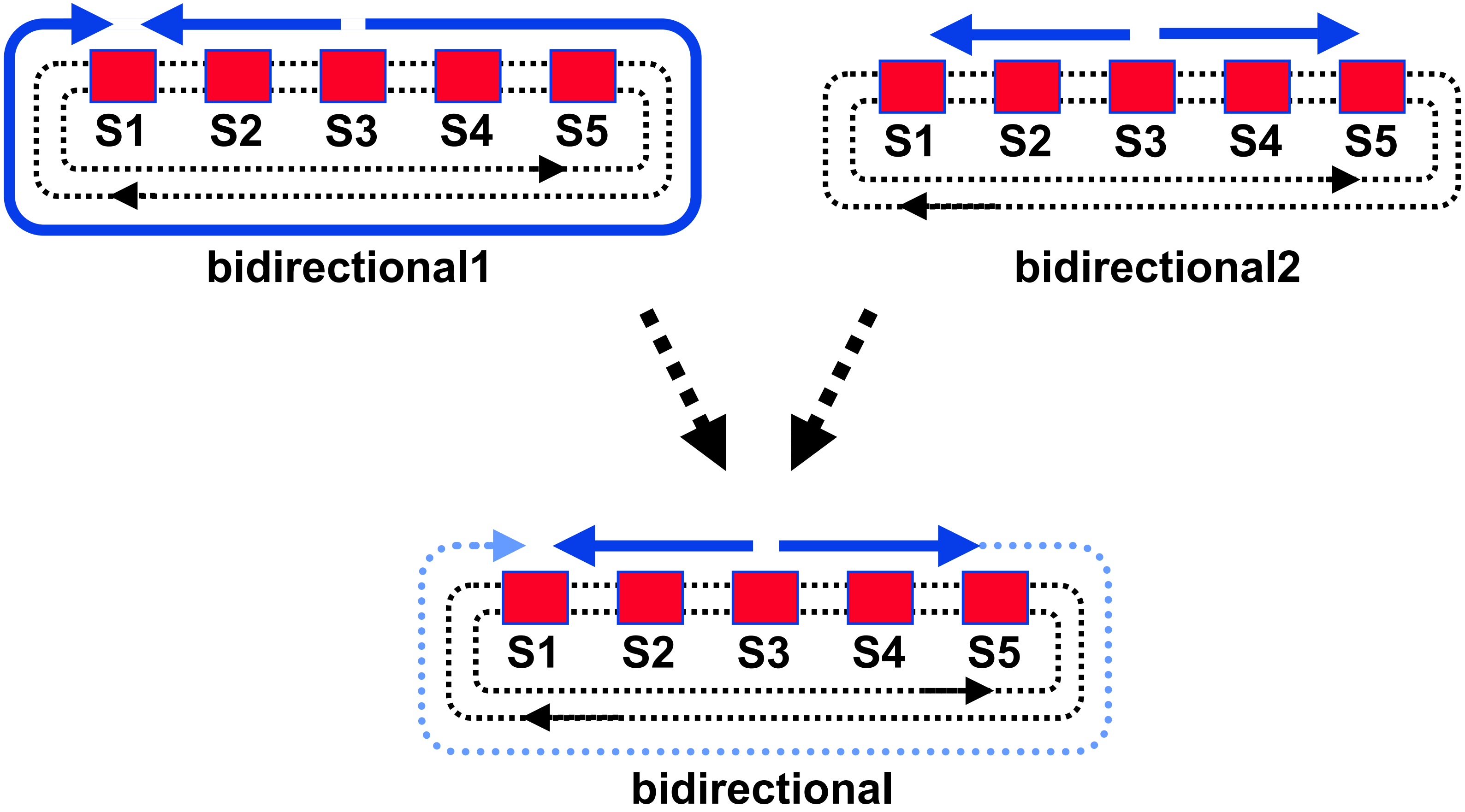


```
// (b) Return-wrap processing
#define AGED (frame.timeToLive==1)
if (frame.we&&AGED) {
        frame.ttl= (stations-AGED);
        frame.ps= 1;
}
if ((frame.timeToLive-= 1)==0)
    discard= 1;
```

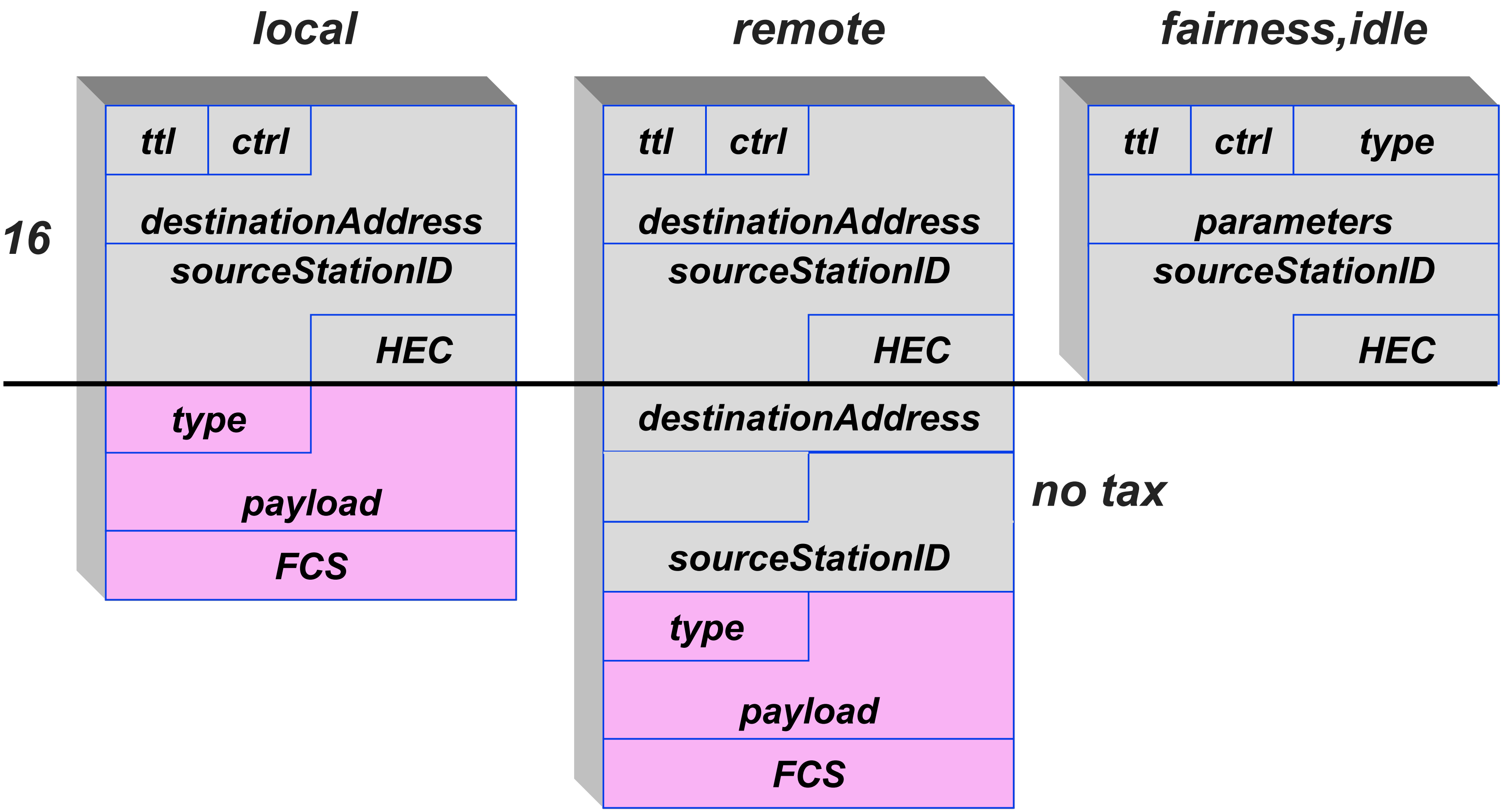
Unidirectional flooding



Bidirectional flooding



Uniform frame formats



local

ttl | ctrl

destinationAddress

sourceStationID

HEC

type

payload

FCS

remote

ttl | ctrl

destinationAddress

sourceStationID

HEC

destinationAddress

sourceStationID

type

payload

FCS

fairness,idle

ttl | ctrl | type

parameters

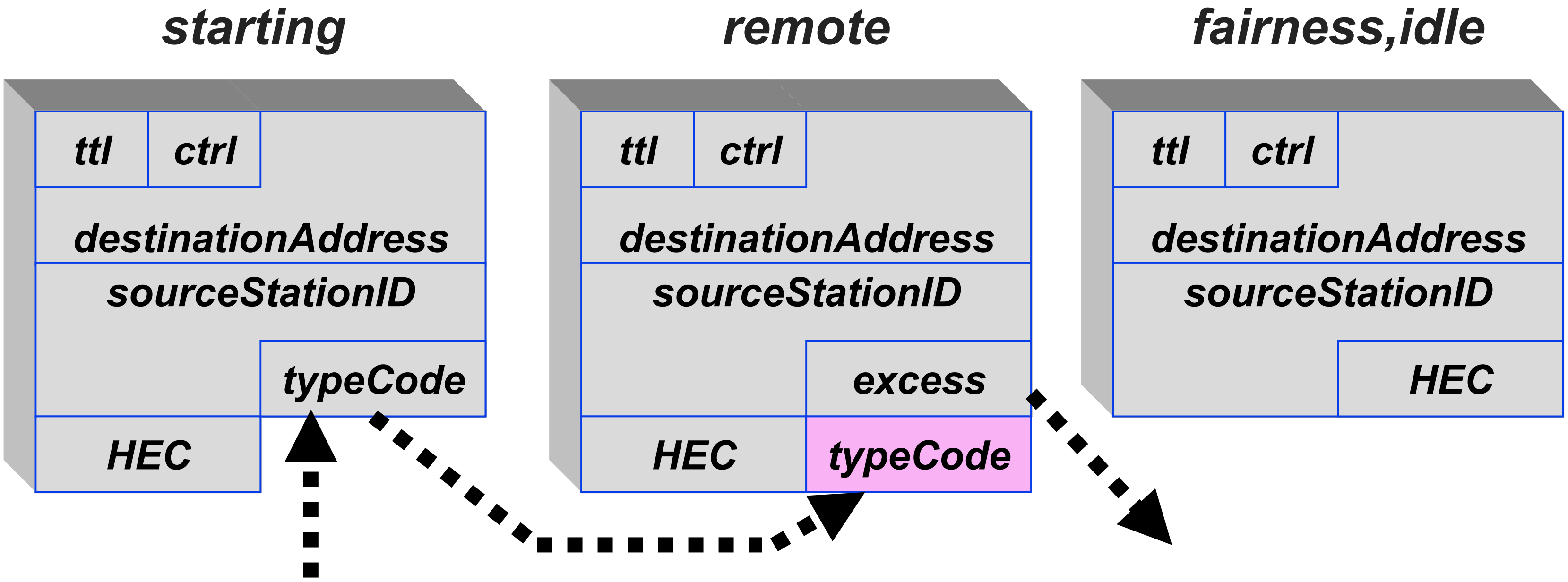
sourceStationID

HEC

16

no tax

Historical evolution



starting

ttl *ctrl*
destinationAddress
sourceStationID
typeCode
HEC

remote

ttl *ctrl*
destinationAddress
sourceStationID
excess
HEC *typeCode*

fairness,idle

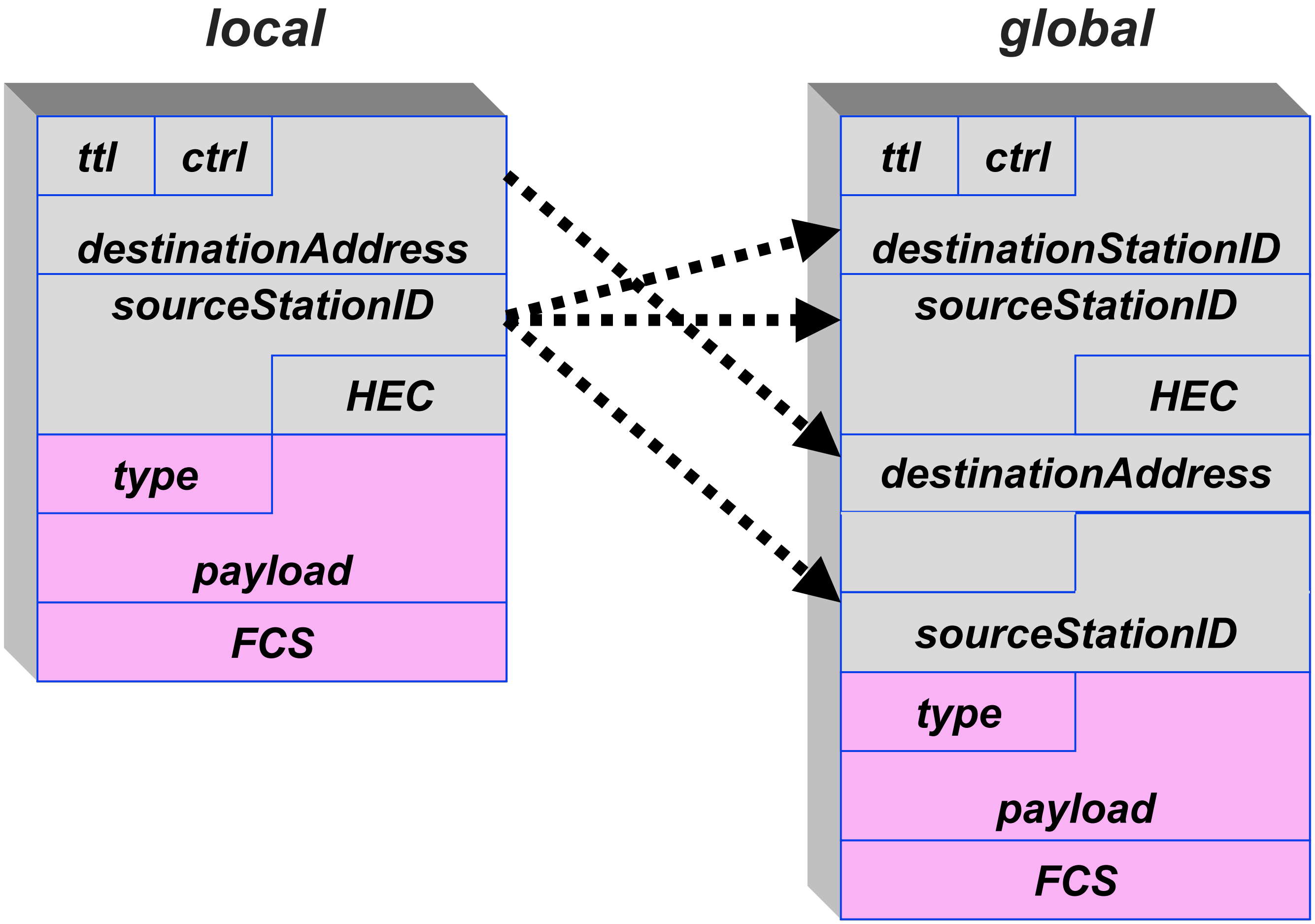
ttl *ctrl*
destinationAddress
sourceStationID
HEC

1) Strictness=>
extra is needed

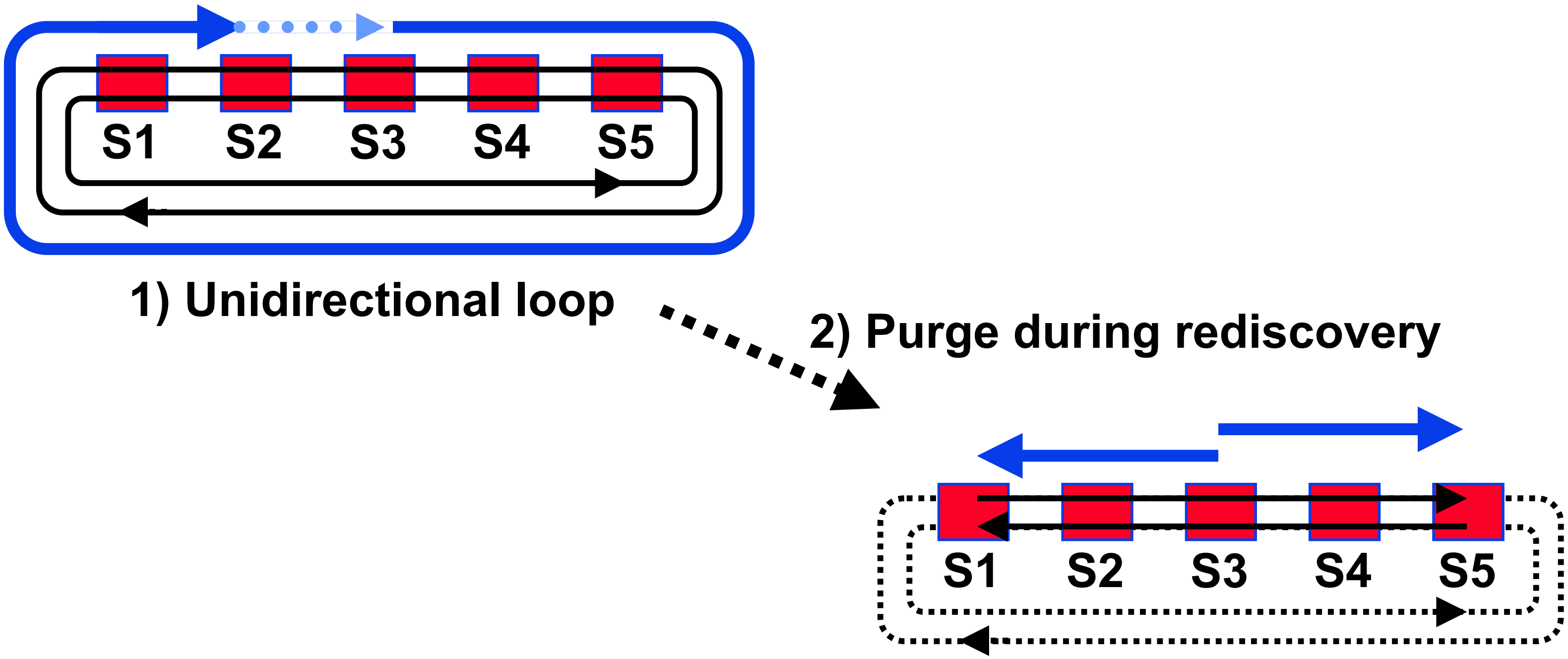
2) Displaced=>
typeCode OK
in the payload

3) Analysis=>
excess not needed
(AD&WH)

Unidirectional address mappings



Purge consistency assistance

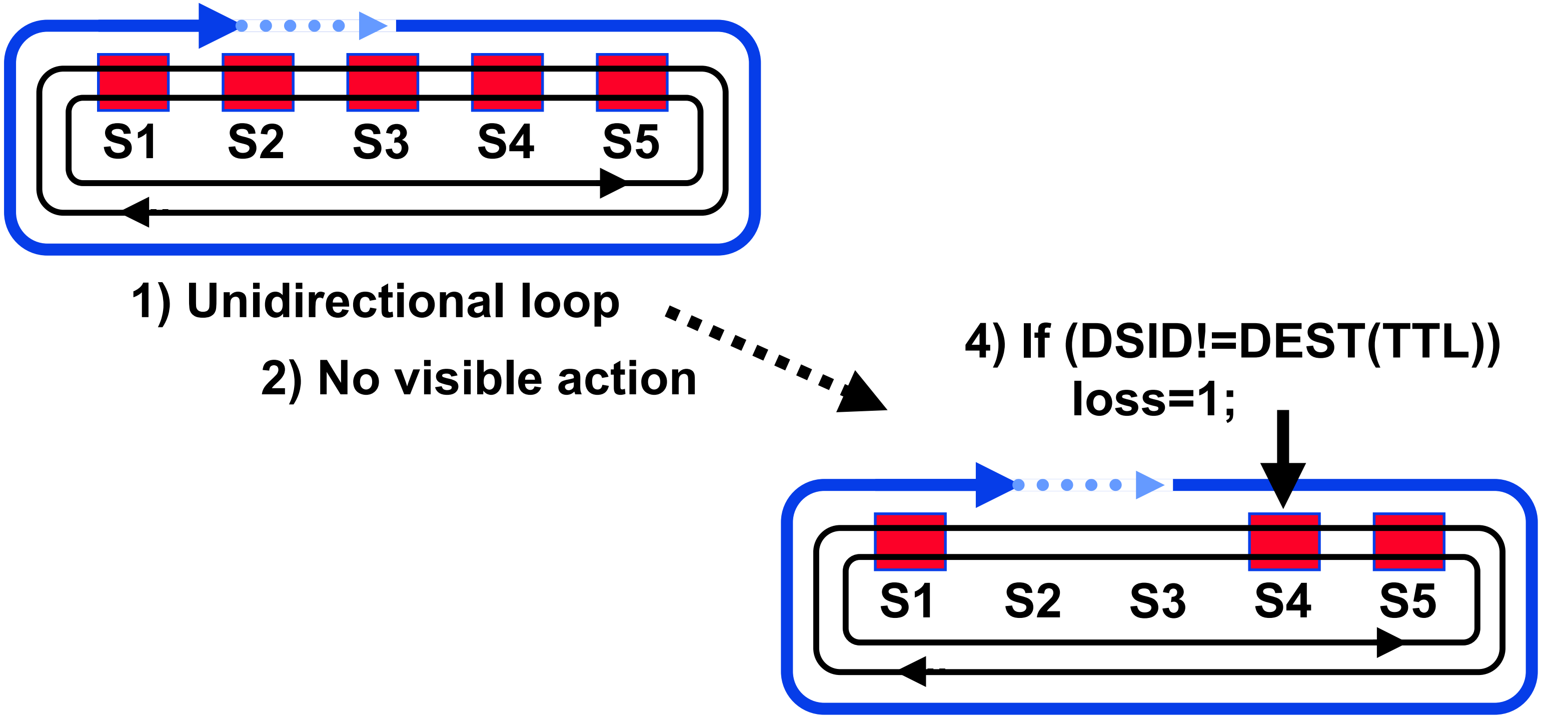


1) Unidirectional loop

2) Purge during rediscovery

3) Unidirectional chain

Destination consistency check (for quiet pass-through)



1) Unidirectional loop

2) No visible action

4) If (DSID!=DEST(TTL))
loss=1;

3) Unidirectional chain

AD flooding conclusions

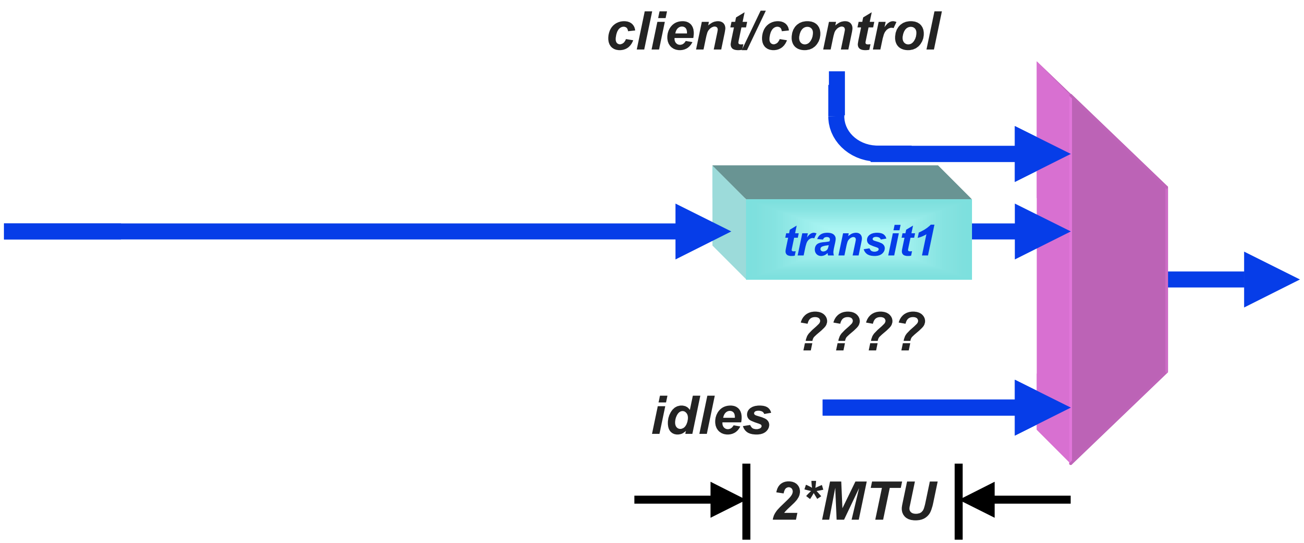
- Fully capable
  - Unidirectional/bidirectional & steer/wrapped
- No frame tax
  - Local router-like traffic
  - Global intermediate transparent bridge
- Four-address equivalents
  - Confidence in completeness
  - Diagnostic monitors are missing nothing
- Defensible header format
  - Minimal 16 bytes, 8-byte “aligned”
  - No “special” parity/HEC/FCS for fairness or idle
- Limitations:
  - Overhead for bidirectional-loop steering
    (but bidirectional w/wrapping is possible)

Transit buffer ad-hoc[3]

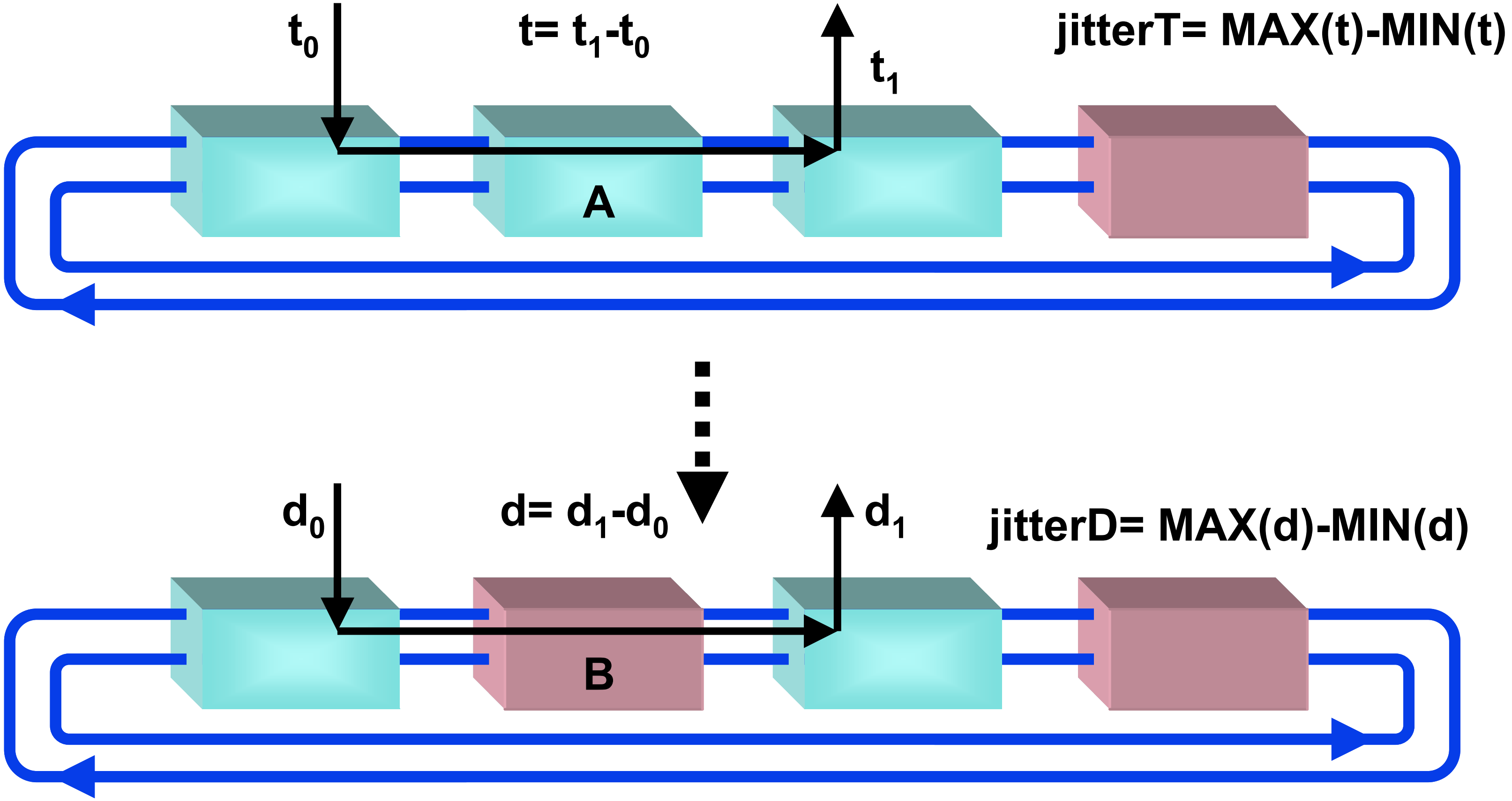
Baseline assumptions

- Multiple options shouldn't complicate the standard
- Lossless transmissions, except for:
  - link failures (cable cuts)
  - transmission errors (noise)
- Cannot mandate large 2nd transit buffer
  - the cost/efficiency set by vendor
  - optimal size depends on link lengths
- Dual-queue stations are uncompromised by others
  - TDM-like bandwidth affects affect only on-path links
  - jitter is unaffected by through-queue replacements
  - (sigh) TDM-like traffic is unclaimable if:
    - Sourced by a small dual-queue station
    - Sourced by a thru-queue station

Arbitration components



Jitter measurements



jitterT= MAX(t)-MIN(t)

jitterD= MAX(d)-MIN(d)

(jitterT-jitterD)= 0

Ad-hoc conclusions

- **Don’t constrain transit designs**
  - **notation “buffer”➔”queue”**
  - **enforced FIFO ordering**
  - **precedence: 1st queue > 2nd queue**
  - ***(any more is controversial)***
- **Vendor flexibility**
  - **any 2nd transit-queue sizing > 2*MTU**
  - **shall maintain jitter behaviors**
  - **don’t complicate the specification**
  - **2nd size of zero ➔ 1st size is *nominal* 1MTU**
  - ***(unclear if 2nd size of zero implies complexity)***

Proposal options

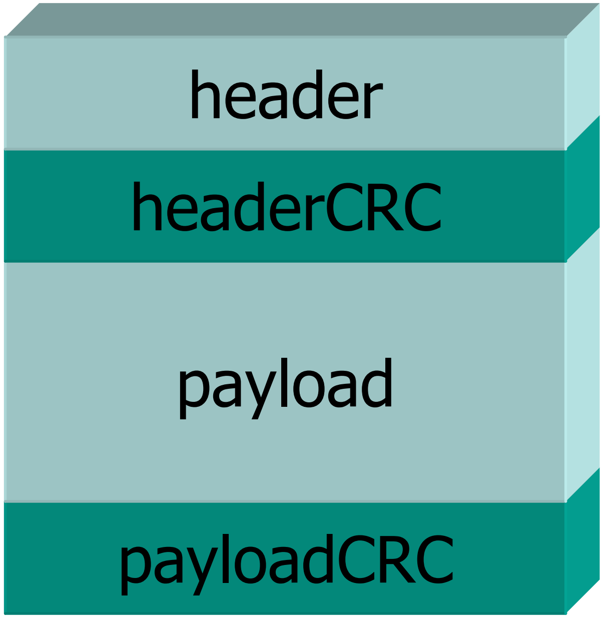
- **All RPR stations shall have two transit queues.**
  The minimum size of both queues is 2 MTUs.
- **All RPR stations shall have either:**
  a) Two transit queues.
  The minimum size of both queues is 2 MTUs
  b) One transit queues.
  The nominal size of this queue is 1 MTU
  (as perceived by normal pass-through traffic)
- ***Expected* decisions would be based on:**
  - **How is specification complexity measured?**
  - **What is the default draft content?**

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



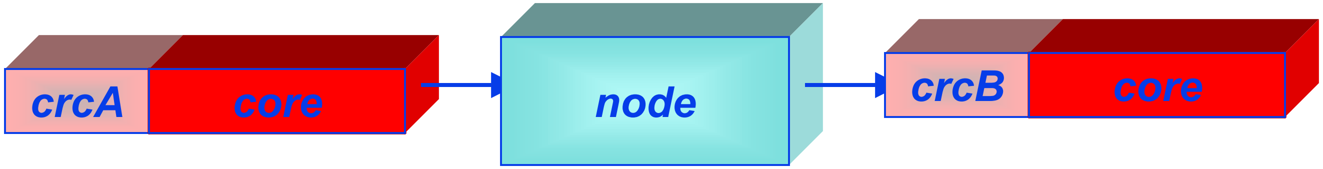
header

headerCRC

payload

payloadCRC

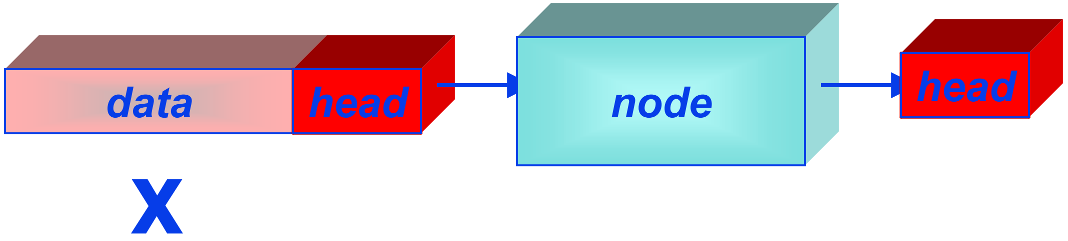
Cut-through CRCs



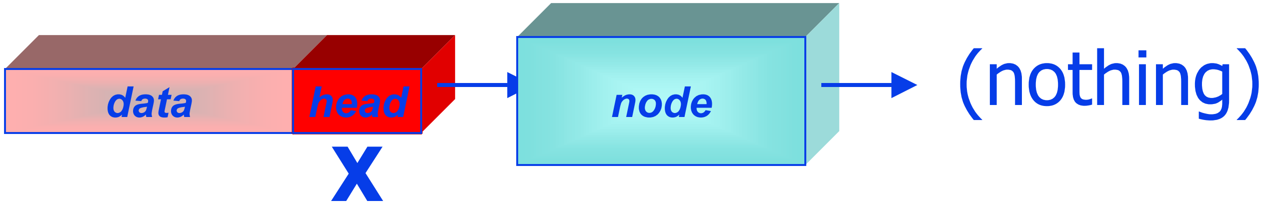
- Corrupted packet remains corrupted
- Error logged when first detected
- if (crcA!=crc) {

```
if (crcA!=crc) {
    errorCount+= (crcA!=crc^STOMP);
    crcB= crc^STOMP;
}
```

Distinct CRCs reduces discards

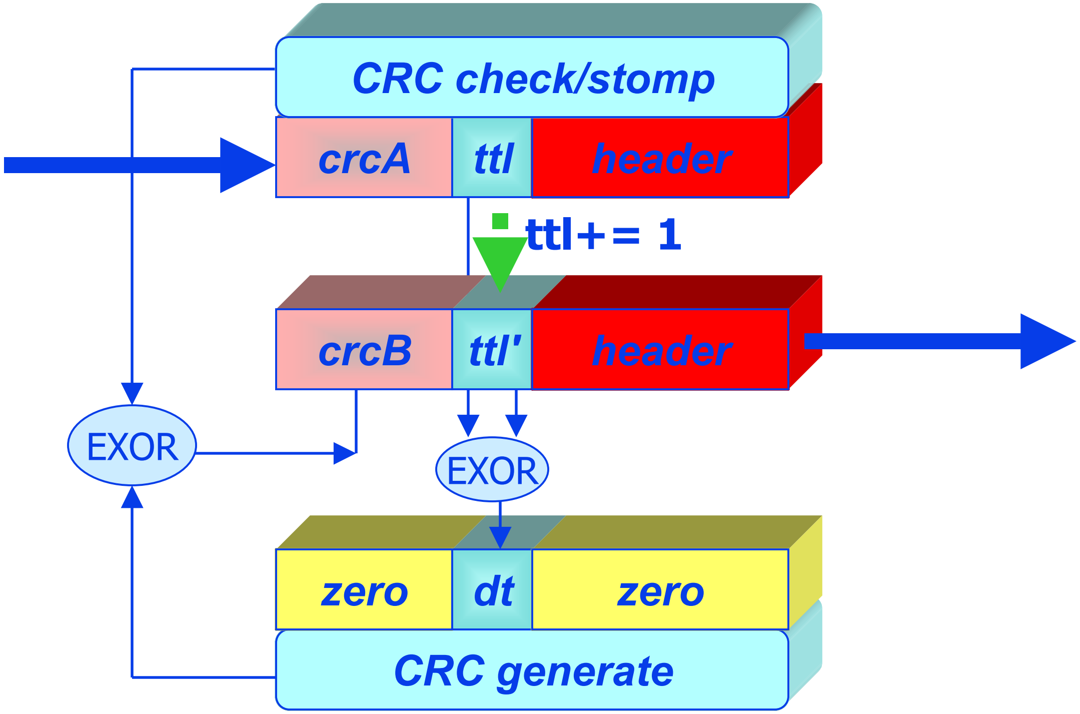


- Discard the corrupted data



- Discard the corrupted packet

End-to-end CRC protected TTL



CRC equation examples

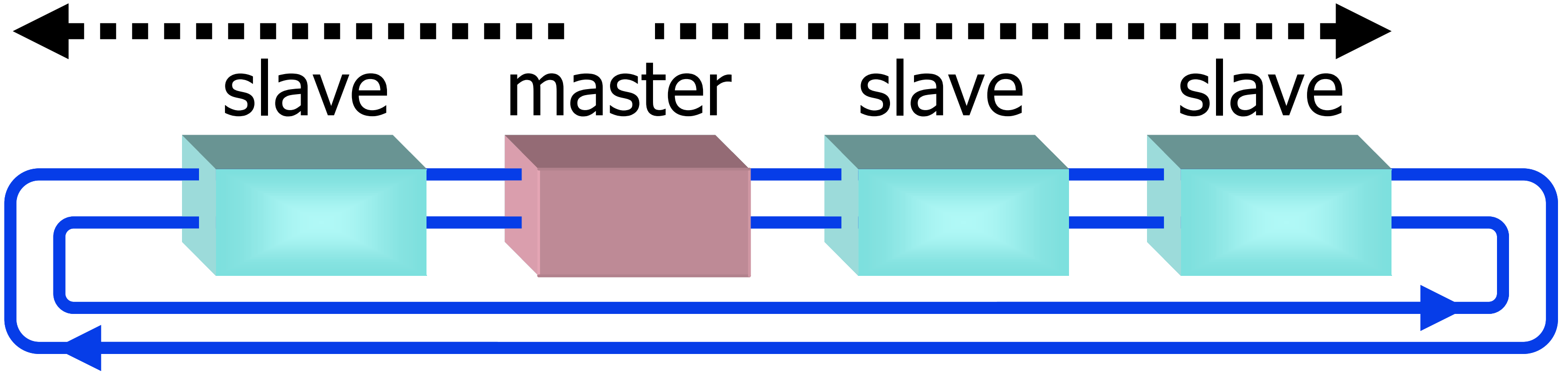
```
a= c00^d00;   b= c01^d01;
c= c02^d02;   d= c03^d03;
// ...
s= c14^d14;   t= c15^d15;
c00= a^      e^  g^h^    m^
c01=   b^      f^  h^j^    n^
c02=     c^      g^  j^k^    p^
c03=       d^      h^  k^m^    r^
c04=         e^      j^  m^n^    s^
c05=           f^      k^  n^p^    t^
c06= a^      e^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

CRC requirements

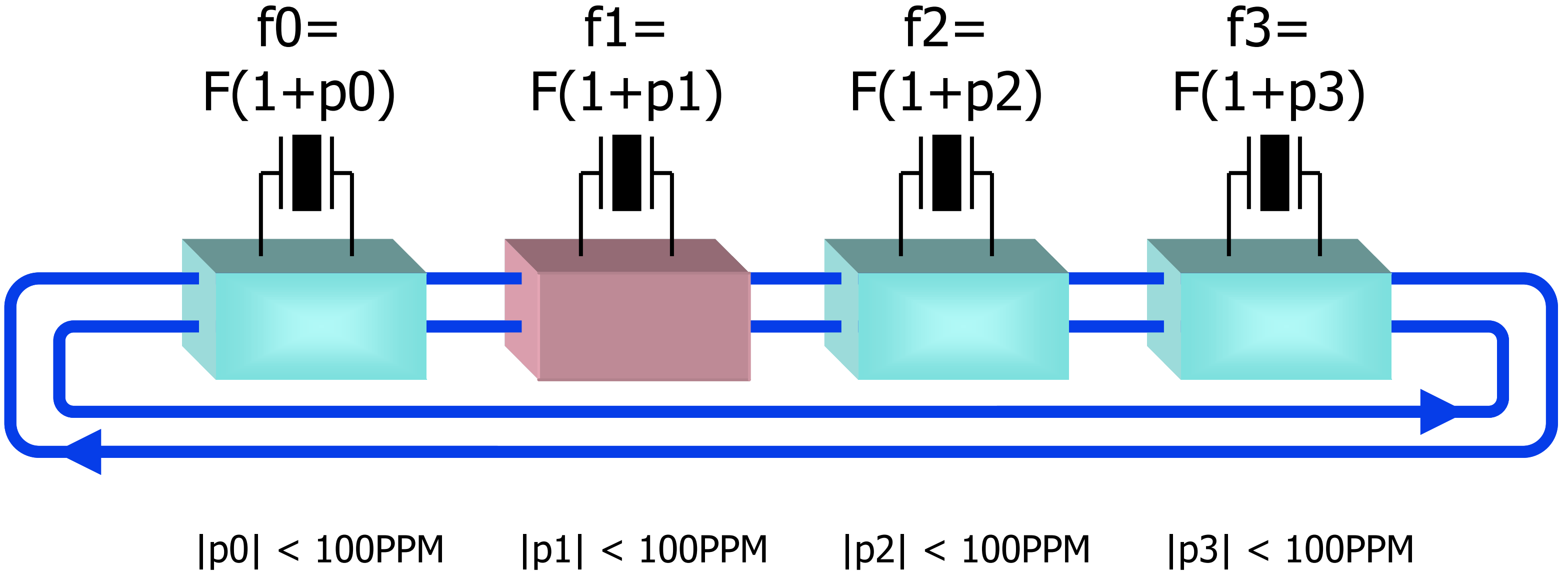
- CRC computation order
  - MAC optimized & PHY independent
  - PHY optimized and 2 MAC orderings
- Define the “stomp” value
  - For HEC as well as FCS
- CRC parallel computations
  - X8, x16, x32 are easily done
  - X64 is harder to print in “portrait”
  - ?? data values as C-code comments ??

Time-of-day synchronization

Time-of-day master and slaves



This is not bit-clock synchronization!



f0=
F(1+p0)

f1=
F(1+p1)

f2=
F(1+p2)

f3=
F(1+p3)

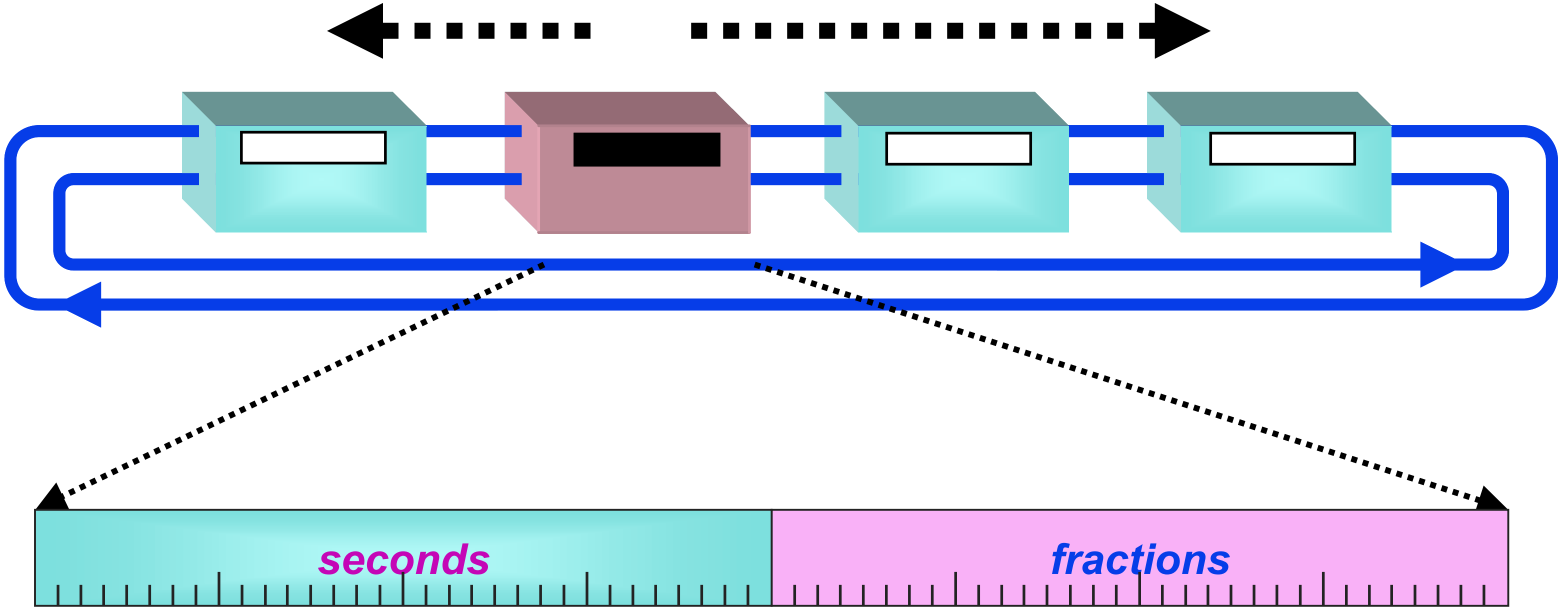
|p0| < 100PPM

|p1| < 100PPM

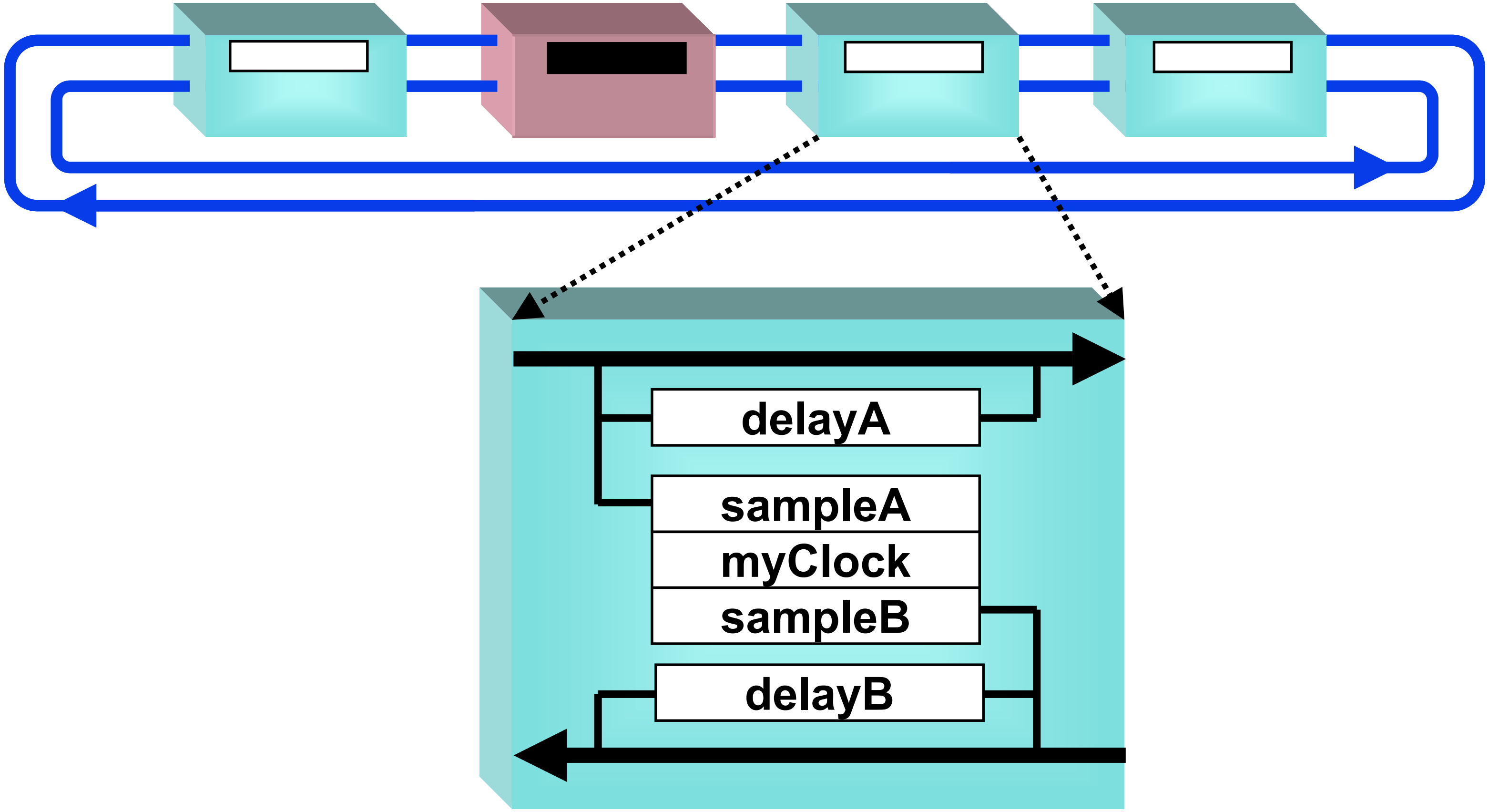
|p2| < 100PPM

|p3| < 100PPM

This is time-of-day synchronization!

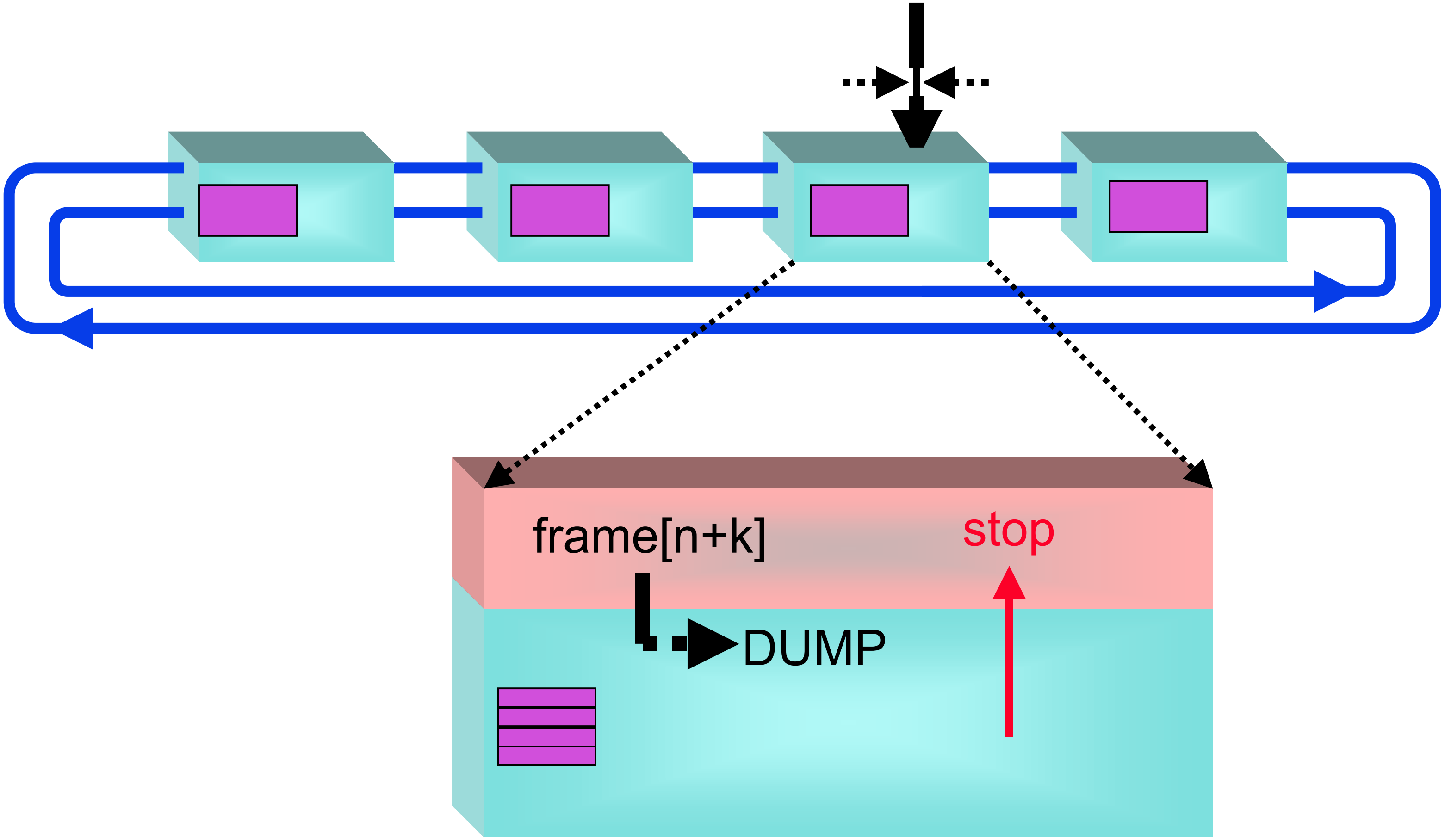


Precise duplex-link synchronization

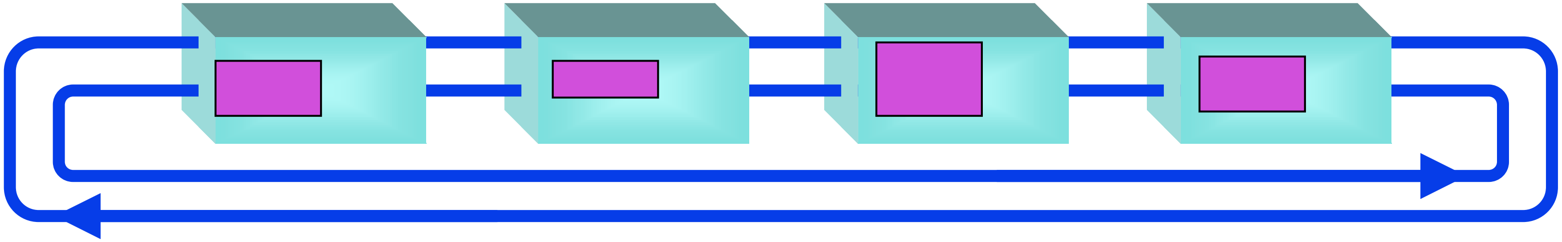


Setting bandwidths

Negotiated bandwidths

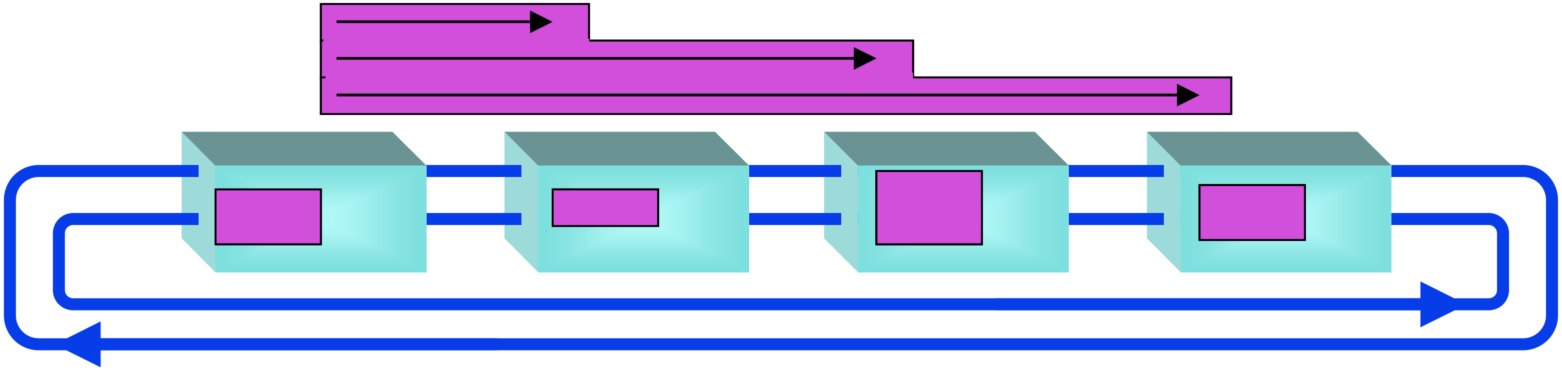


Bandwidth negotiations



- Sum of bandwidths < link capacity
- Independent accounts
  - class-A0 and class-A1 rates
  - class-B rates
  - class-C weightings
- Accounts are distance dependent

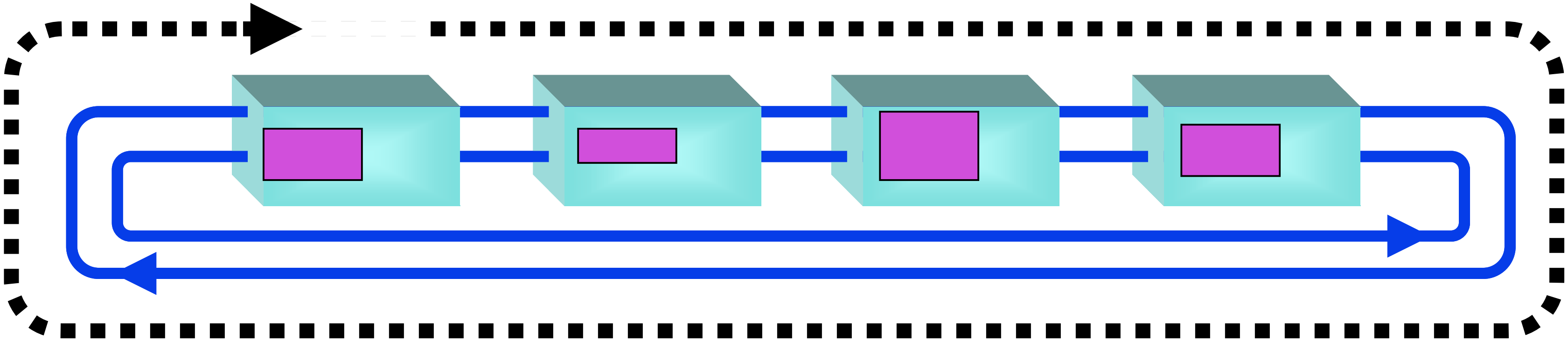
Distance-dependent accounts



- BW accounts have spatial dependencies
- Available BW has monotonic decrease

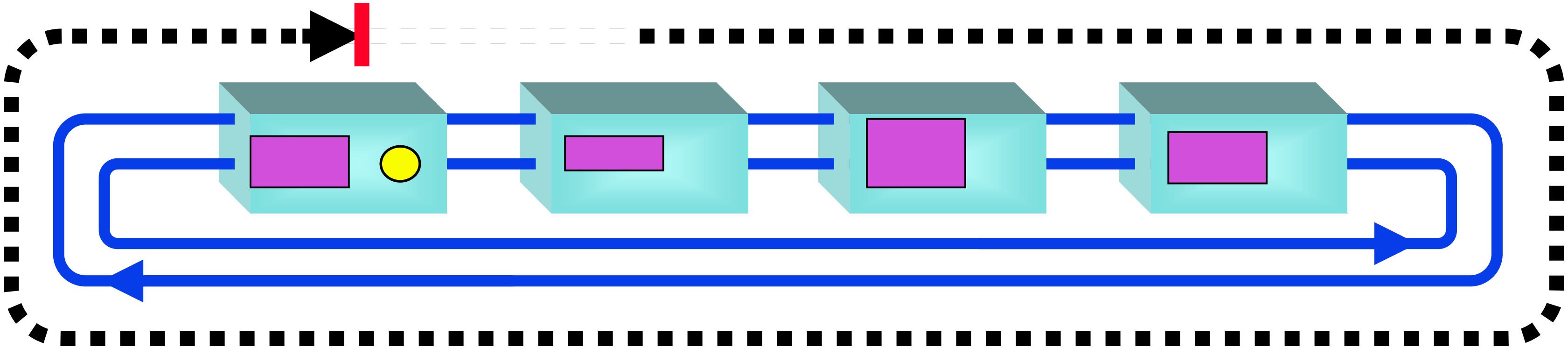
Allocation update sequence

3) Distribute revised accounts

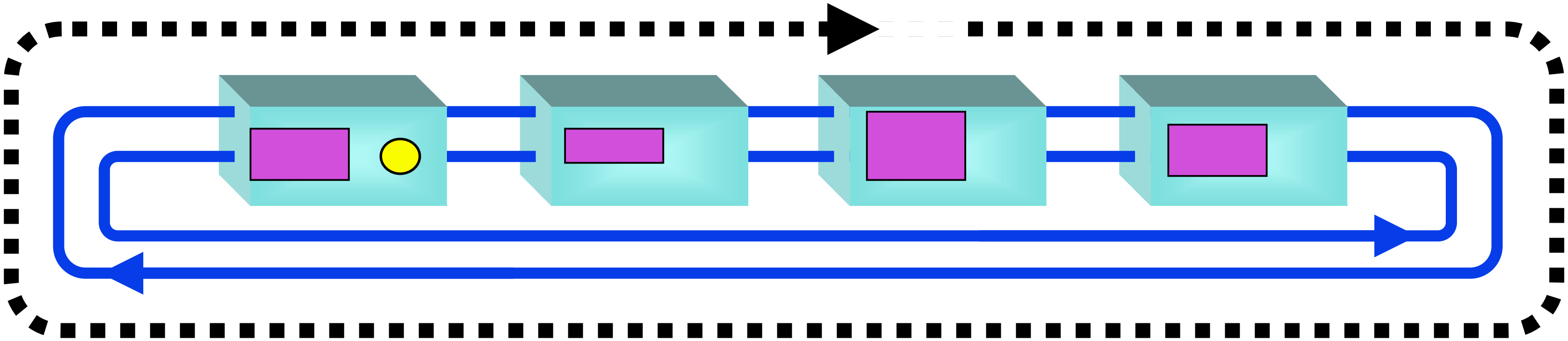


Concurrent conflicts

in.source < me.source



in.source > me.source

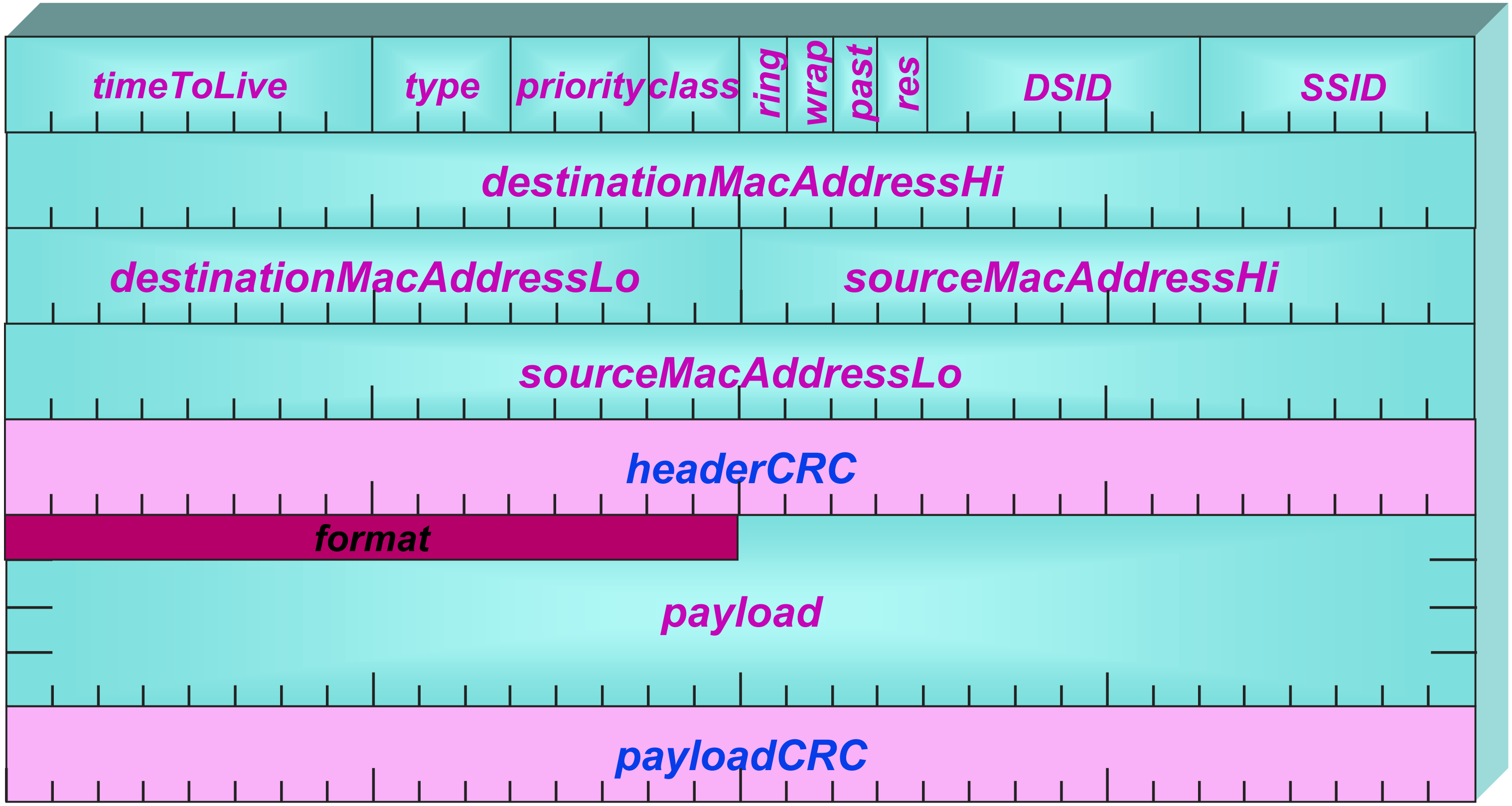


BW allocation conclusions

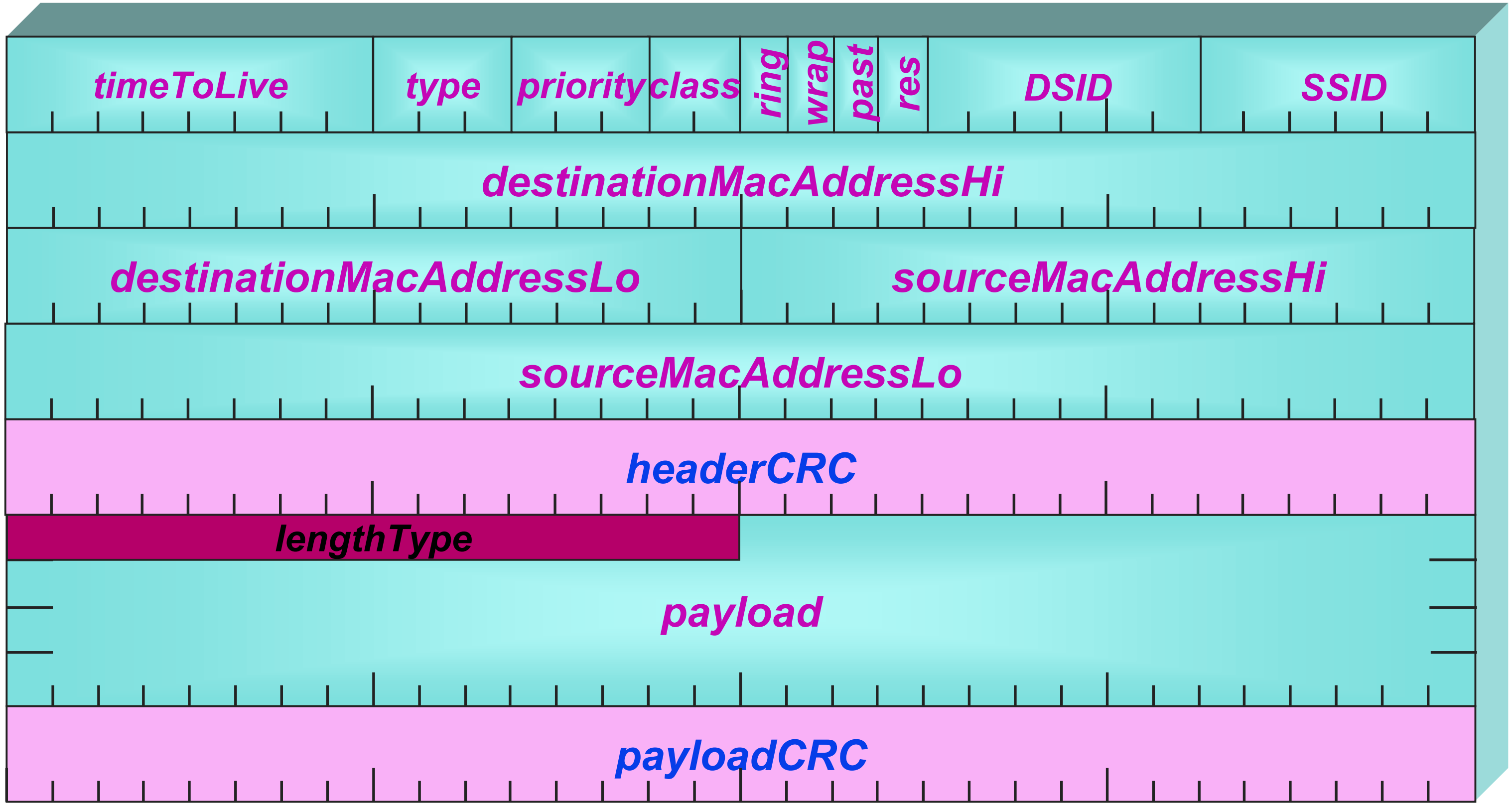
- BW allocation is necessary to ensure consistency
- BW allocation should include spatial reuse
  Worst-cast hop-count OK for simple stations
- No central tables are required
- MAC-identifier suffices for tie resolution
- Some error-recovery details may be necessary
- We need philosophy, not technical, agreement.
  (sufficient detail exists for first-round inclusion)

Frame formats

RPR Frame Format



Ethernet Frame



timeToLive | type | priority | class | ring | wrap | past | res | DSID | SSID

destinationMacAddressHi

destinationMacAddressLo | sourceMacAddressHi

sourceMacAddressLo

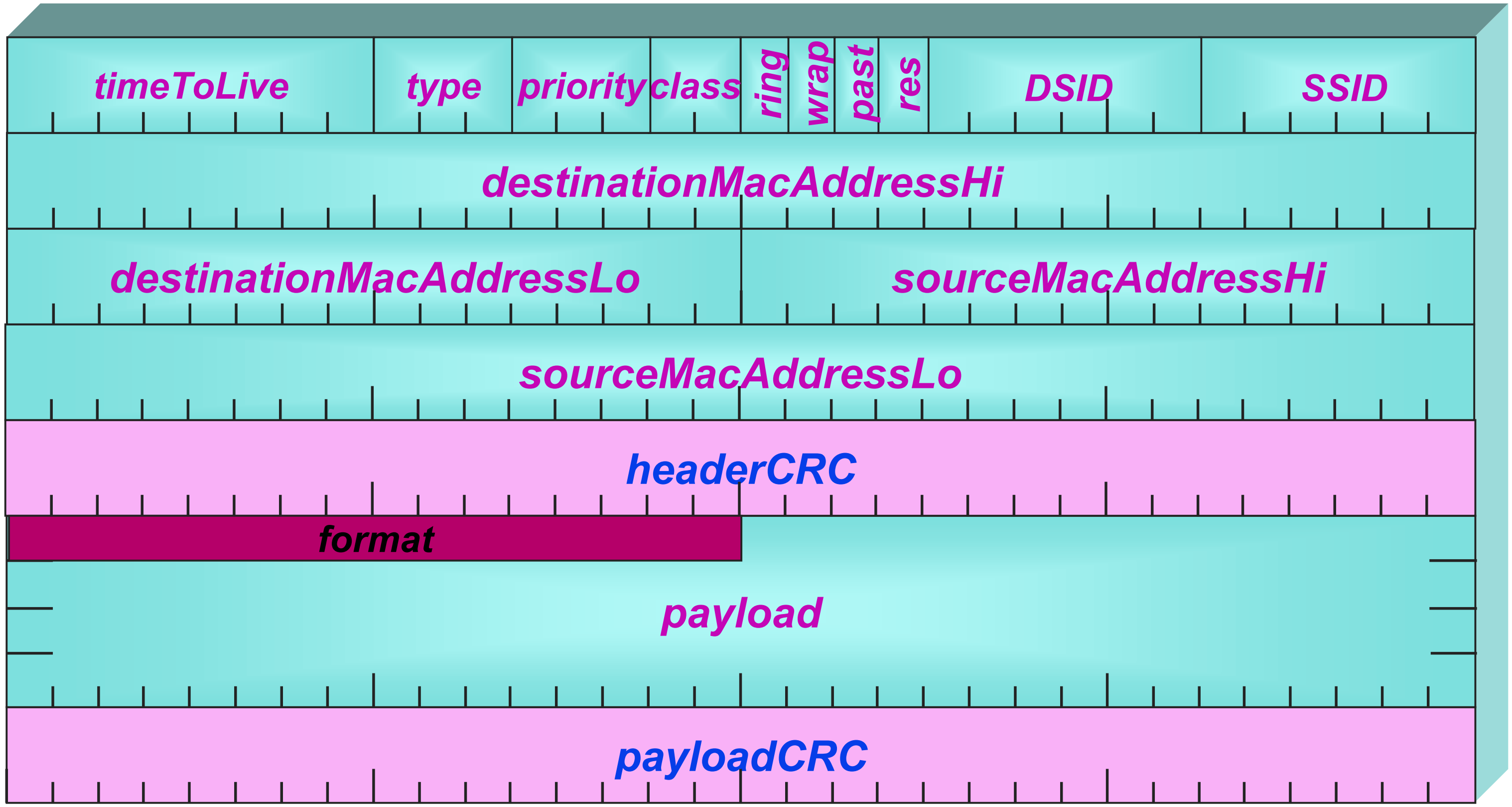
headerCRC

lengthType

payload

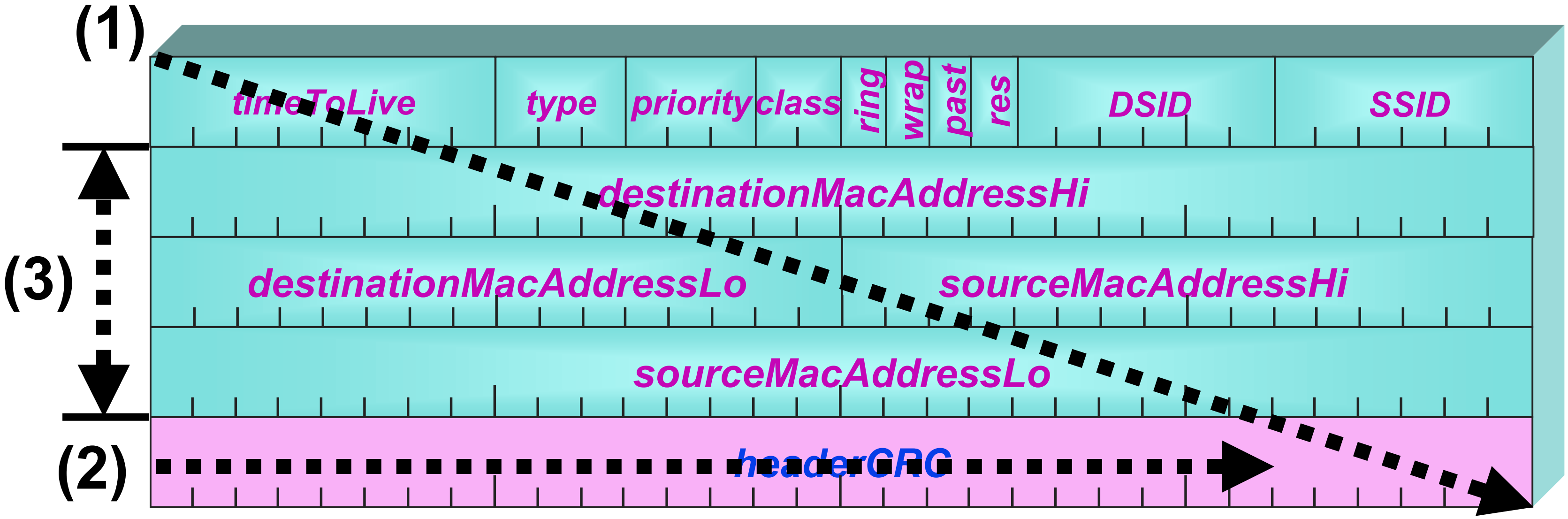
payloadCRC

Control Frame



Control Field Functionality

RPR Frame Format

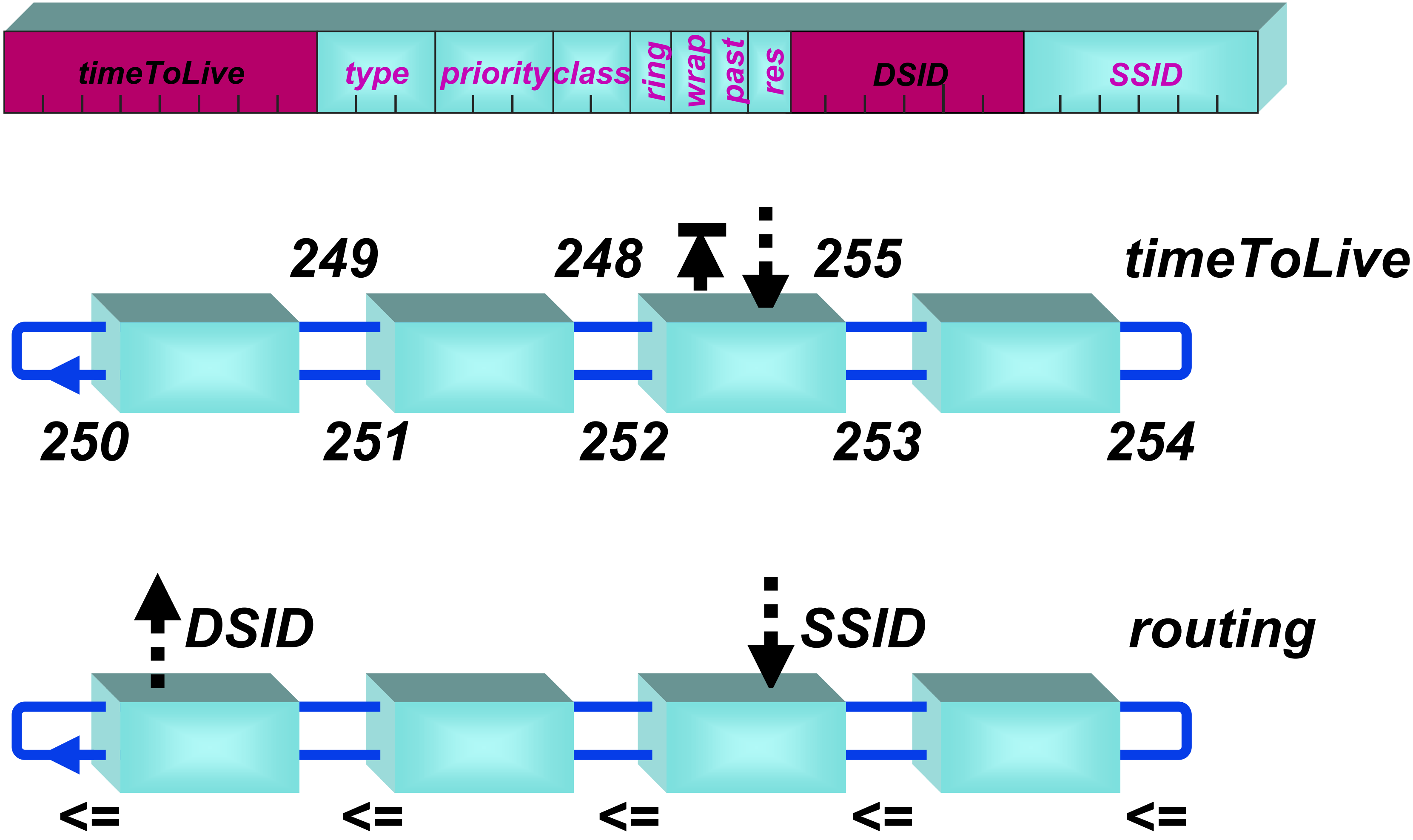


1) 32-bit aligned

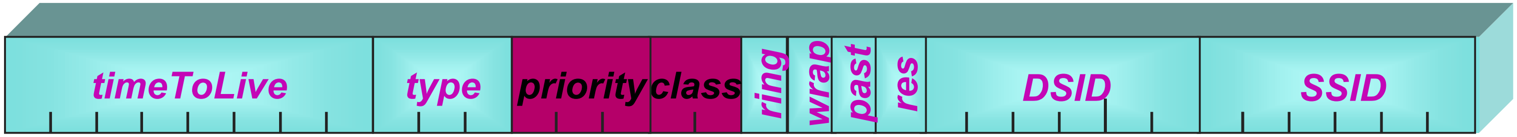
2) 32-bit checksum

3) Global MAC addresses (not local)

Robust TTL accounting



Global and local priorities



timeToLive

type

priority

class

ring

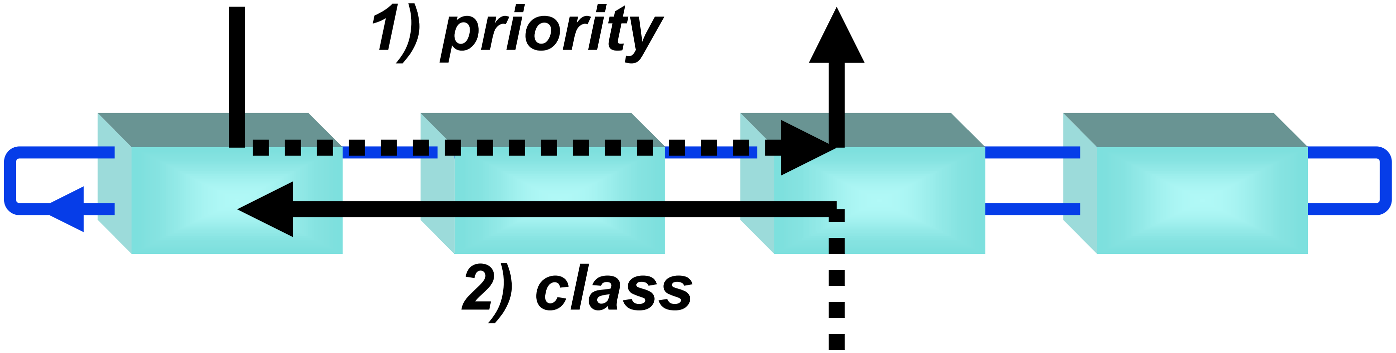
wrap

past

res

DSID

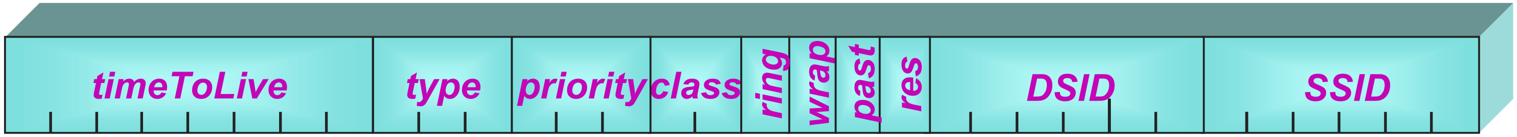
SSID

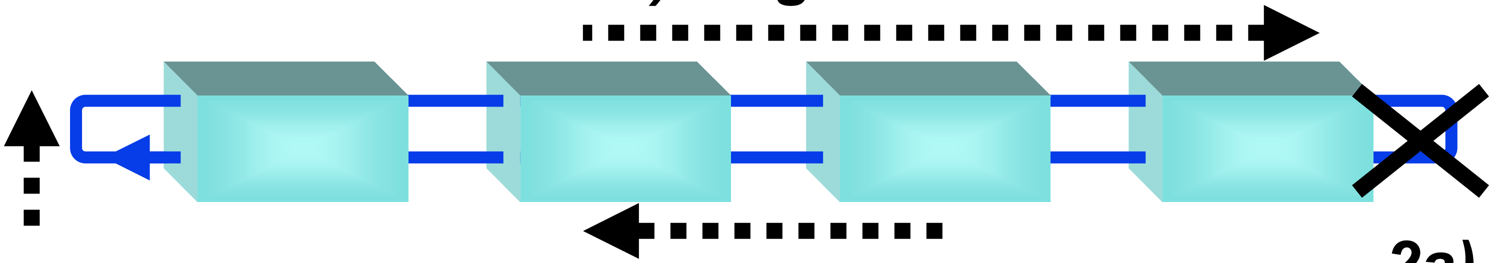


1) priority

2) class

Ring&wrap flags





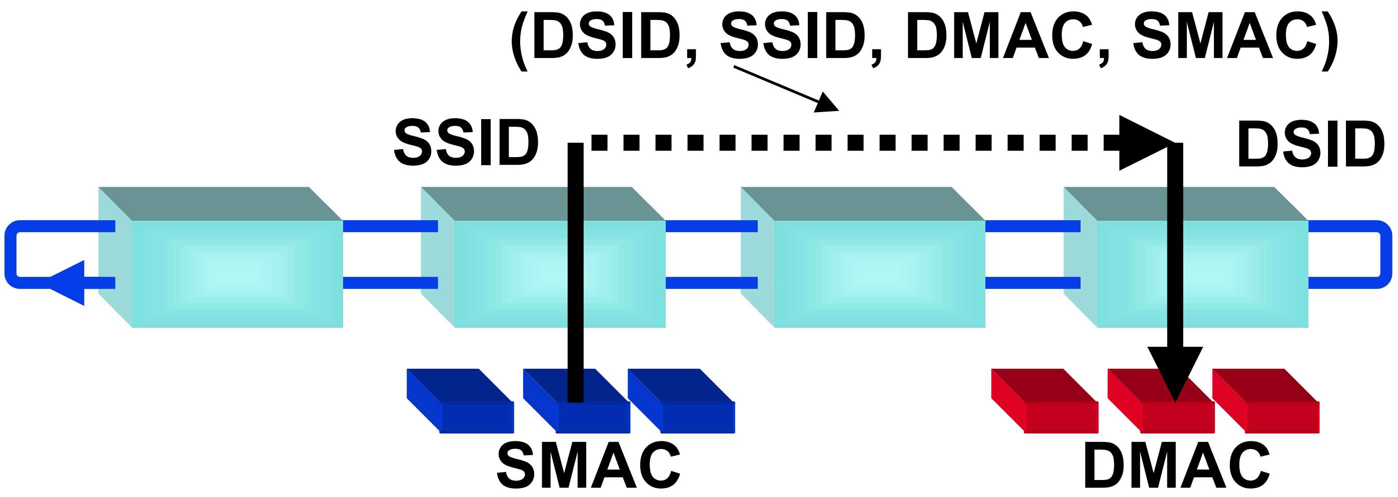
1a) ring=0

2a) wrap=1

1b) ring=1

2a) wrap=0

Source/Destination Coding



Fixed

TTL:8
DMAC:48
SMAC:48
DSID:48
SSID:48

+12 bytes

Stable

TTL:8
DSID:8
SSID:8
DMAC:48
SMAC:48

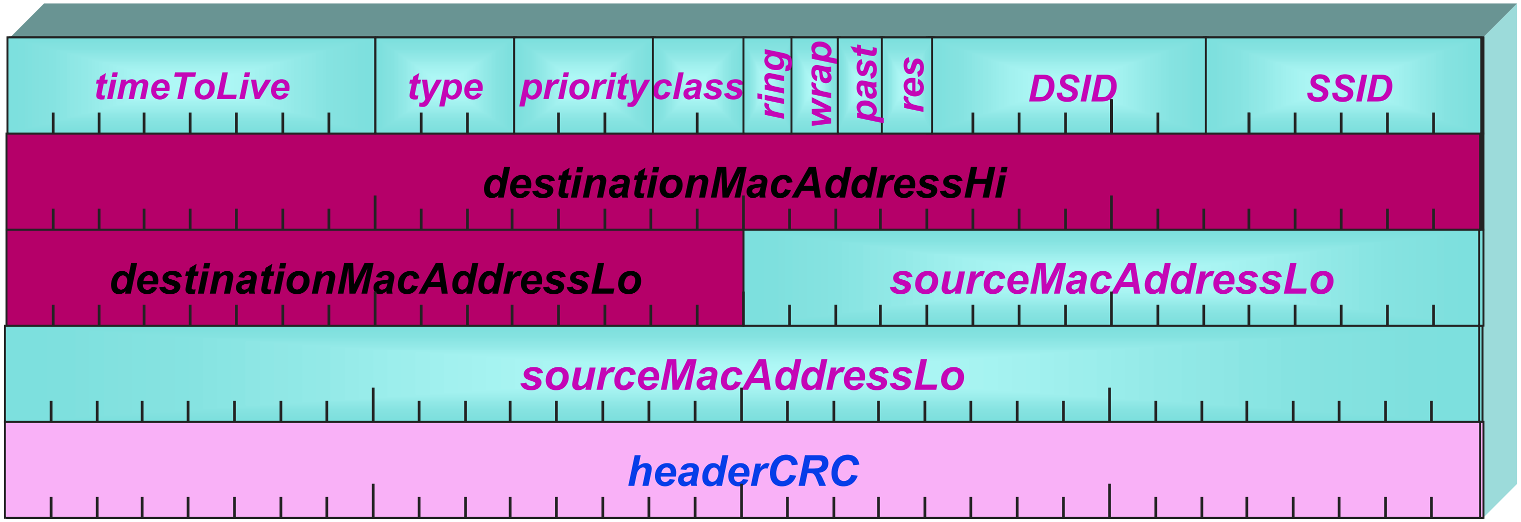
+2 bytes

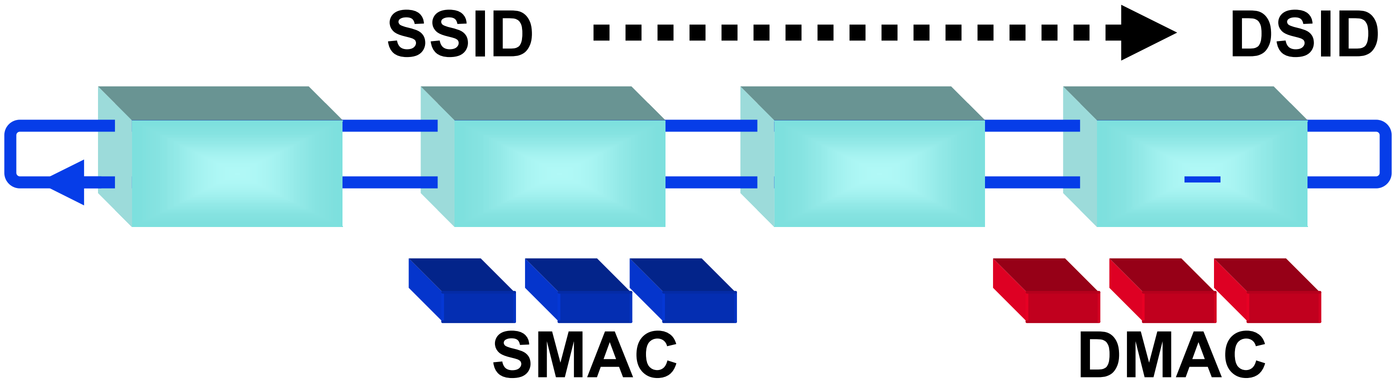
Relative

DSID:8
SSID:8
DMAC:48
SMAC:48

(+1 byte)

Ethernet Bridging


timeToLive
type
priority
class
ring
wrap
past
res
DSID
SSID
destinationMacAddressHi
destinationMacAddressLo
sourceMacAddressLo
sourceMacAddressLo
headerCRC


SSID
DSID
SMAC
DMAC