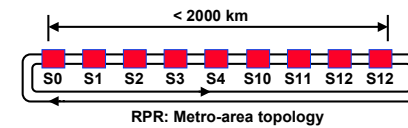


## P1796 Resilient Backplane Ring (RBR) slide material 2004Sep16

David V. James  
djv@alum.mit.edu

An overview of RBR directions, as envisioned by the PAR.

## RPR topologies



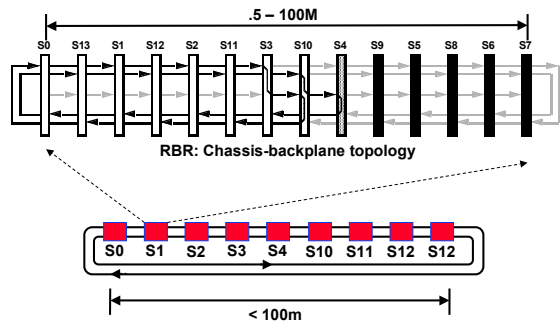
SONET environment applications  
Duplex counter-rotating rings with spatial reuse  
IEEE 802 frames, with ring-routing supplements  
Several product-in-field constraints

IEEE 802.17-2004 Resilient packet ring (RPR) acknowledged the inherent full-duplex point-to-point link topology associated with high speed connections.

While Ethernet switches have effectively done similar things, there are some advantages (protection recovery time, class of service commitments) that benefit from ring topologies.

While the concept of full-duplex rings is not new (see IEEE 1596-1995), this concept has been difficult to acknowledge within 802.3. However, within 802.17, rings are already a given (from SONET), and could therefore be safely assumed within the metropolitan area networks (MANs) topologies.

## RBR topologies



On the backplane, distances are much smaller.

Some vendors desire to cluster multiple backplanes, so longer hops are also necessary. However, there is no desire to extend into the MAN space.

That being said, there are multiple simplifications for RBR that may eventually be found as useful for RPR.

## Current status

- Study group authorized by MSC January, 2004
- PAR approved June 24, 2004
- Scope:
  - Resilient backplane ring (RBR) is a backplane interconnect based on the dual-ring resilient topology of resilient packet ring (RPR) and the 802 MAC addressing structure. RBR includes features appropriate for the low-latency backplane environment: destination-based flow control, low-power short-haul PHY, backplane-to-backplane links, transport of IEEE-1394 isochronous data, and support of IEEE-1596 memory-update operations.
- Purpose:
  - The purpose of this project is to leverage the benefits of network-compatible resilient interconnects within low-latency backplane environment.

The MSC development tends to be different than that of 802.

Detailed straw-man proposals are encouraged in early phases of the standard. This allows for more informed voting during down-selection phases, which may occur throughout the working-group lifetime.

The scope is defined as the backplane environment, wherein low latencies can facilitate features not easily supported over long-haul connections. These features could also be useful within the computer room or enterprise, provided the latency remains less than or comparable to frame sizes.

## Reasons for the RBR standard

- High speed backplanes are oftentimes used within the networking environment, where designs can be simplified by sending network frames and card-to-card communications over the same links.
- Although the resilient packet ring (RPR) has the quality of service (QoS) needed for card-to-card communications, other facilities associated with a low-latency backplane environment are missing.
- When RPR like protocols are supplemented with latency-critical backplane services, the resulting backplane interconnect should be sufficient for many mixed application backplane designs.
- Affected sectors would include enterprise networking and computer server industries; perhaps 100s or hopefully 1000s of companies.

These are the “official” reasons for RBR, as documented in the PAR application. These are new and (I believe) not required to be included in the standard. So, its unclear if these are simply background or official requirements, from a procedural perspective.

## RBR motivation

- **Merged backplanes, for low \$\$**
  - **computer & network services**
  - **point-to-point connectivity**
- **RPR topology is “native” point-to-point, but**
  - **optimized for metro applications**
- **IEEE 1394/1596 have backplane services, but**
  - **network addressing model desired**
  - **refinements from lessons of the past**

The basic concept is that mixed backplanes (network and computer apps) are more cost effective.

With only one backplane to support, economies of scale take effect. This effects the learning curve as well a component costs.

With one backplane being shared, fewer pins are consumed.

## RBR simplifications

- ~~Steering, edge wrap, and~~ center-wrap protection.
- ~~Single queue and~~ dual queue transit buffers.
- ~~Rate based and~~ credit based shapers.
- ~~Parity, 16-bit hec, 16-bit hec,~~ 32-bit hec, and 32-bit fcs.
- ~~Conservative and aggressive fairness, without~~ adaptive.
- ~~Multichoke “hooks” and~~ single-choke specification.
- Negotiated capabilities:—
  - ~~wrapping mode~~ (→ center wrap)
  - ~~jumbo capable~~ (→ all are transit-capable)
  - ~~conservative fairness~~ (→ all are adaptive)

RPR (a source of many RBR concepts) was a collection of ideas from multiple vendors, some with existing product in the field. As such, a variety of options were accepted into the standard, as the process of achieving consensus. Interoperability issues could be glossed over in RPR, since most rings were expected to consist of equipment from the same supplier.

RBR needed to be simple and fully interoperable, without operator intervention. As such, the elimination of options is highly desirable. And, in the absence of existing product, the elimination of options is an easier task.

## RBR distinctions

- **WG development process**
  - **Straw men proposals (more than slideware)**
- **RBR is within smaller “boxes”**
  - **Reaction times are much smaller**
  - **Destination-congestion: “retry” becomes practical**
  - **Low-powered transceivers are a must**
- **RBR plug-and-play vs RPR single-vendor acquisitions**
  - **Elimination of most configurable “knobs”**
  - **Negotiated real-time bandwidths**
  - **Specified time-of-day synchronization**

Significant revisions are needed, however:

Real time bandwidths must be “real”, perhaps necessitating pre-emption  
Fewer options are needed to ensure correctness and interoperability.  
Obscurity remains in the MAC and fairness, due to incomplete reviews

Extensions are needed, primarily due to the shorter latency or lower costs.  
Destination-assisted flow control avoids loss of frames on  
heavily congested stations, and is feasible due to lower “busy-retry” delays.  
Hard-coded memory-access (read, write, add, set, clear) commands  
simplifies multiprocessor communications  
Accurate time-of-day synchronization reduces accuracy requirements  
of clocks.

PHYs are best done by large formal groups: perhaps Ethernet on Backplane  
efforts can be leveraged.

## RBR protocol summary

- Leveraged RPR values:
  - Ethernet frames with QOS delivery
  - Ring efficiency and resiliency
- QOS enhancements
  - Accurate time-of-day synchronization
  - RPR fixups for calculated classA1/classB guarantees.
  - Reduced sync/sync and sync/async interference.
  - Negotiated access controls.
- Lossless transactions
  - Destination-asserted flow control
  - Hard-coded memory-access commands
  - Request/response queuing options
- Backplane PHY definitions

RBR leverages RPR values, but provide enhancements expected of computer or computer-cluster backplanes.

## Similar technologies

- Infiniband
- HyperTransport
- PCI-express
- Rapid I/O
- Others?
  - Fiber-channel, serial ATA, serial SCSI, FDDI

Has this been done before?

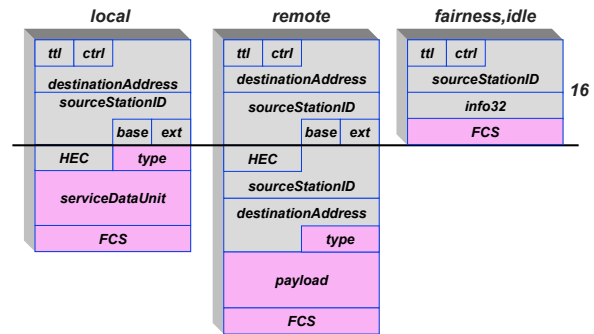
Yes, but mostly with a computer I/O channel perspective.

## Frame formats

Lets be concrete and talk about formats.

The RBR formats closely resemble the RPR formats, so both will be illustrated to show the common and distinctive features.

## RPR format summary



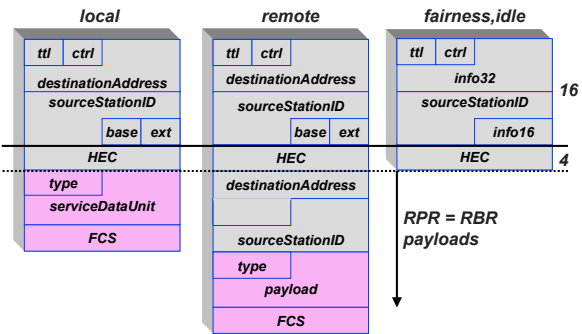
For comparison, consider the RPR frame formats. From a simplistic perspective, these can be thought of as Ethernet frames, with additional header content provided to manage the multidrop nature of the ring.

Unfortunately, there are some irregularities that complicate designs.

The HEC differs from the FCS size, due to unclear (perhaps product-in-field) constraints.

Furthermore, the first 2 bytes in the fairness frame are not covered by the FCS, but yet-another very-weak parity check, due to unclear (perhaps product-in-field) constraints,

## RBR format summary



Early inputs requested more uniform frame formats.

The RBR frame formats have uniform/complete 32-bit CRC checks, on the header and the payload.

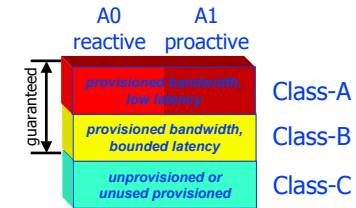
The header is thus always 32-bit aligned.

Better error coverage on the header is associated with the 32-bit HEC.

The HEC and the FCS use the same CRC computation, with resulting savings in conceptual and physical design complexity.

This has minimal impact on a bridge, since this is normally recalculated due to ttl (time-to-live) decremented values. However, it provides the freedom to extend and/or revise some defined features.

## Arbitration classes



Traffic classes are labeled as 'A', 'B', and 'C'.

Class A: provisioned low-latency bandwidth, for telephone.

Supports "synchronous-like" transfers.

Class B: provisioned bounded-latency bandwidth, for SLA provisioned BW.

Class C: unprovisioned or unused provisioned bandwidth.

Supports ensured forward progress and fairness for the residual BW.

The current RPR doesn't really support classB and subclassA1, since the levels of classB that can be safely supported are dependent on STQ sizes and a set of complex unsolvable equations.

The backplane RBR fixes these limitations, and provides synchronized clocks, so that end-to-end TDM like flows can be readily supported.

Much of the design knowledge exists within the 1394 domain, although some changes in technique are necessary do to differences in transport details, as well as knowledge gained from the good&bad experiences in the field.

P1796

## IEEE Std 802.17 RPR precedents

- Point-to-point, because
  - Fiber rings are dominant in the MAN
  - Destination stripping for spatial reuse
- Resilient to link faults
- Service classes
  - ClassA: TDM like services
  - ClassB: prioritized delivery
  - ClassC: weighted fairness
- Some congestion management

dvjRbrSlides

September 16, 2004, page 15

802.3 and 802.17 are both rings, its just that 802.17 allows more than 2 stations!  
The advantage is resiliency: any one cable can be “cut”.  
Class of service are a clear value and distinction.

P1796

## IEEE Std 1394 Serial Bus precedents

- Synchronized clocks
- Isochronous has priority.
- Admission control limits usage
  - local: shared register
  - remote: path setup messages
- Isochronous BW is limited to one’s allocation
  - transmissions are once-per-125 $\mu$ s
  - bandwidth includes header overhead

dvjRbrSlides

September 16, 2004, page 16

Leverage is better than NIH and 1394 is the “within home” precedent for isochronous (synchronous) delivery services.

## IEEE Std 1596 SCI precedents

- Memory mapped transactions
  - primitives: read, write, swap
  - request/response deadlock concerns
  - coherence: directory based is feasible
- Congestion management
  - busy status, with local retries
  - fairness is the difficult issue

Leverage is better than NIH and IEEE 1596 is the “computer” precedent for bus-like transactions on ring topologies.

However, the arbitration protocols were largely classless, so RPR derived enhancements are desired.

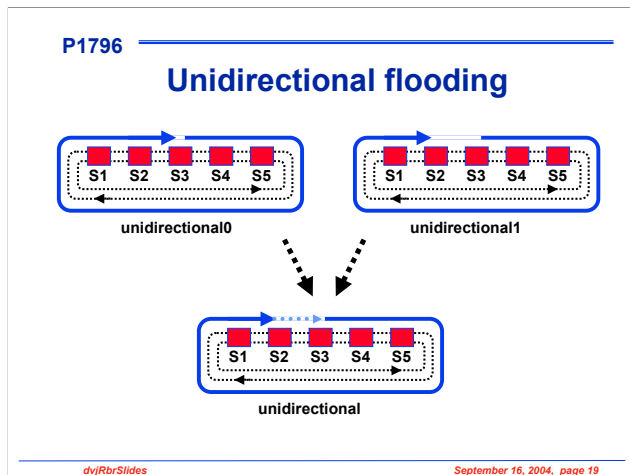
Also, IEEE 1596 busy-retry protocols have the possibility of consuming indefinite bandwidth, as undesirable (and difficult to plan with) property.

## Bus emulation

Buses support broadcast, and RBR is a ring.

How is that possible?

The answer is flooding.



The primary need is to support flooding!

The simplest flooding is unidirectional; send frames to one's self.

At first, two options were thought to be necessary:

unidirectional0—send a frame to yourself

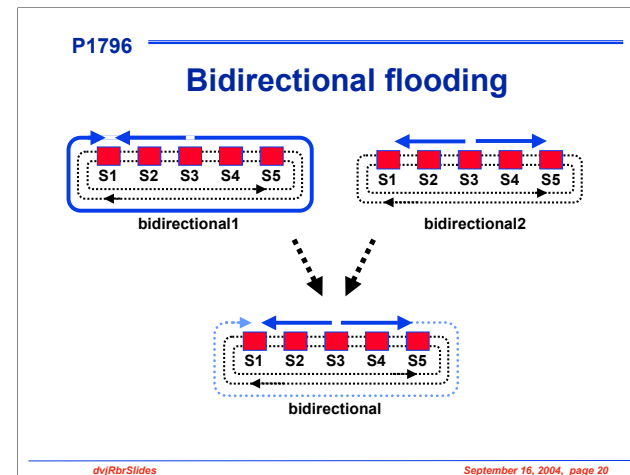
this was simple, but wasted the last-link bandwidth

unidirectional1—send the frame to your upstream neighbor

this was efficient, but complicated the transmit protocol

After reflection (a benefit of constructive competition), only one is needed:

unidirectional—send frame to yourself, strip at the upstream neighbor



A secondary need is to support bidirectional flooding!

Easier to load-balance link utilization

Fundamental for enhanced bridging efficiency

At first, two options were thought to be necessary:

bidirectional1—send a frame to a common midpoint

this was simple, but wasted the last-link bandwidth

bidirectional2—send the frame to adjacent mispoints

this was efficient, but complicated the transmit protocol

After reflection (again, due to constructive competition), only one is needed:

bidirectional—send frame to the midpoint, but

ringlet1 strips at the upstream neighbor

## Flow control

Flow control, a nontrivial problem:

ClassA: Flow is prenegotiated

ClassB: Flow has precedence

ClassC: Flow has weighted fairness

## Opposing arbitration

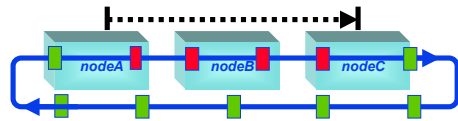


- Data packets flow in one direction
- Arbitration control flows in the other\*

Arbitration information from one link affects the transfers on the opposing link.

So, these packets are separated from the *dataPath* packets and their contents used to control the proposing of the opposing run *dataPath* processing.

## Proactive class-A0 partitions

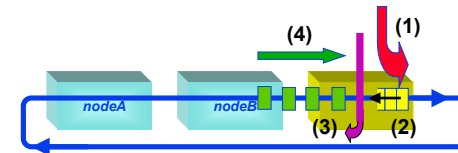


- Data packets go source-to-destination
- Residue returns destination-to-source to provide subsistence for transmissions

For subclassA0 traffic, space is statically “reserved” by carving up the bandwidth into subclassA0 and non-subclassA0 components. If not used by classA0, those reserved bandwidths represented wasted opportunities.

The advantage of subclassA0 is that any amounts of classA traffic can be supported, regardless of buffer sizes and cable lengths. The disadvantage is that unused subclassA0 bandwidths are wasted.

## Reactive class-A1 control



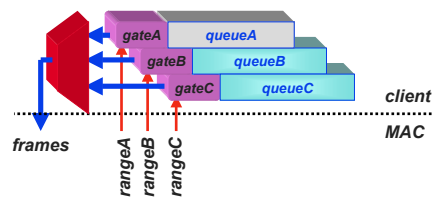
- Transmission of packets causes
- Backup of passBC FIFO that
- Returns flow-control information that
- Provides consumable idle packets

For subclassA1 traffic, space is dynamically “reserved”. Stations send up to their allowed subclassA1 bandwidth, even though this additional traffic may exceed the ringlet bandwidth.

As a station’s queues fill, congestion indications are sent upstream within small frames. This “eventually” stifles sending of upstream classB/classC traffic.

The advantage of subclassA1 is that unused subclassA1 bandwidths can be reclaimed by lower-class traffic or other currently nonoverlapping subclassA1 traffic. The disadvantage is that subclassA1 traffic levels are limited by the  $\text{bufferSize/linkLength}$  ratio. As such, big buffers or limited cable lengths may be needed.

## MAC-Client interface signals

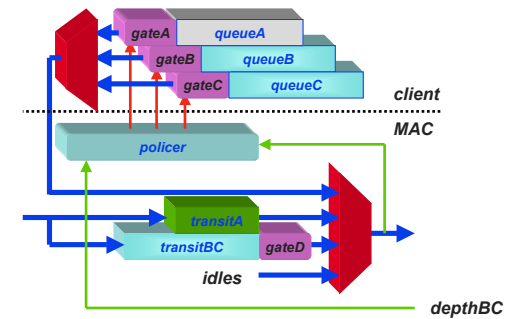


A basic attach point has policing at the ingress.

The intent is to limit the short-term class-A transfer rate, so that class-B bandwidth can be ensured. Also, a short-term bandwidth averaging requirement is used to bound the worst-case jitter to the 125us averaging interval.

The queues are serviced in a given precedence order, but the apparent “presence” of any class effects whether this class can be sent based on policing history.

## Arbitration components

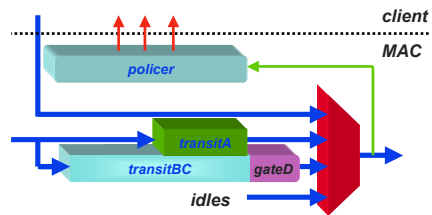


A basic attach point has policing at the ingress.

The intent is to limit the short-term class-A transfer rate, so that class-B bandwidth can be ensured. Also, a short-term bandwidth averaging requirement is used to bound the worst-case jitter to the 125us averaging interval.

The queues are serviced in a given precedence order, but the apparent “presence” of any class effects whether this class can be sent based on policing history.

## Small-to-large transitBC



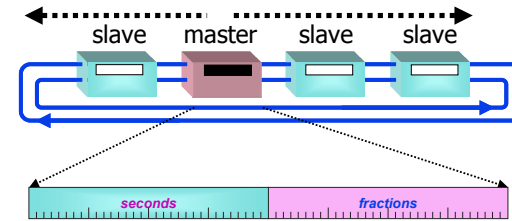
- 1) Small => proactive classA0
- 2) Medium => mixed classA0/classA1
- 3) Large => reactive classA1

The size of the transitBC buffer constrains the levels of supportable subclassA1 bandwidths.

For efficiency, transitBC should be larger than the number of bits "on the wire".

This isn't all that hard, on the backplane or within the computer room.

## Time-of-day synchronization (not bit-clock synchronization)



Synchronized time-of-day involves updating clocks in all stations, so that data from the A/D converted on the microphone is transmitted at the same rate that this data is played on the speakers.

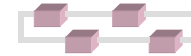
This is a well-understood concept from 1394, but generally discredited by 802 long-haul folks. However, value is perceived for the support of TDM-like services.

## Operations, administration, and maintenance (OAM)

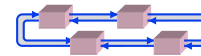
- Recovery from failed components
  - “Protection” via wrap&steer
- Identify failing components
  - MIB counts
  - stomped CRCs

Things don't always work, so...

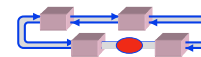
## Supported topologies



- A physical ring



- Dual ringlets



- Duplex ringlet

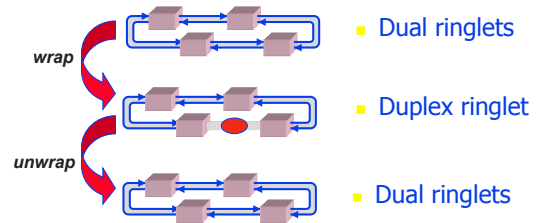
The physical ring is the base cabling model.

However, each cable is assumed to provide a full-duplex connection. Thus, the dual ringlet model is fundamental for the normal operation.

A component (e.g. laser) failure may force this into a single ringlet configuration. This should be supported, as the other (nonfailed) link remains useful.

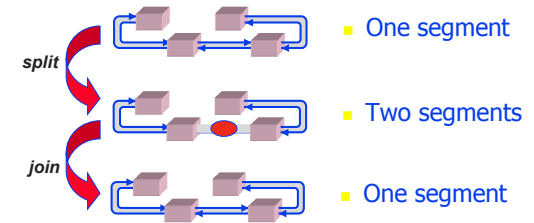
The duplex ringlet, of course, is necessary to support communications after a cable (e.g. backhoe cut) failure.

## Link failures: wrap & unwrap



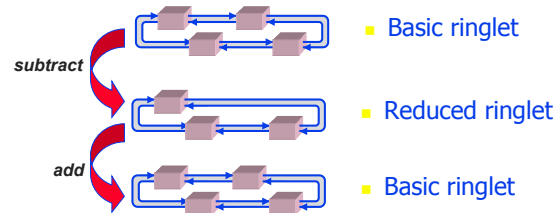
Cables can be cut, although in the computer room backhoe failures are rare(>).

## Link failures: split&join



While less likely, a ring can be severed in two places, yielding two distinct segments.

## Link failures: subtract & add



Stations can be added or removed. This typically happens when a station switches between operational and non-operational modes, typically when severe internal failures are detected. A fiber optic switching matrix could also act like this.

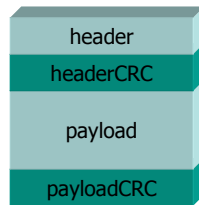
Thus, discovery protocols need to deal with losses of stations, without a visible loss of link connectivity.

## 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

CRC processing with errors isn't as simple as toss-and-forget: a frame may be partially transmitted before the failure is sensed.

## Separate header and data CRCs



Having distinct headerCRC and payloadCRC values simplifies header updates, since the headerCRC is at a nearby fixed location.

Separate CRCs also make it less likely to “lose” the header, allowing flows to be better monitored.

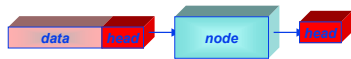
## Cut-through CRCs



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

A distinct value is used to mark previous-detected CRC errors, so that the error can be logged when first detected.

## Distinct CRCs reduces discards



X

- Discard the corrupted data

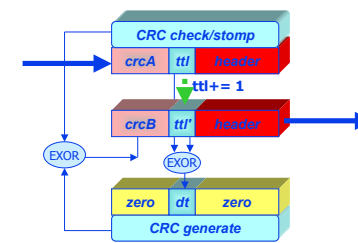


X

- Discard the corrupted packet

Even if the payload is corrupted, the error can be reliably logged. However, header errors (much less likely, with only 20 bytes) cause everything to be discarded.

## End-to-end CRC protected TTL



What about the time to live (ttl) field, which is updated within each station. How is the CRC updated?

Its actually quite simple, since there are only 8 distinct values corresponding to CRC changes, which can be found through a lookup table. By EXOR'ing in the change, rather than recomputing, the frame retains its CRC coverage while being updated.

## 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^
```

Hard time getting those CRC computations correct?  
Confused with 802.3 bit-swapping?  
Well, specific tables, code, and examples are provided.  
The guess work is eliminated!

## Evolving slides

These slides are being refined.

P1796

## Queue depth feedback

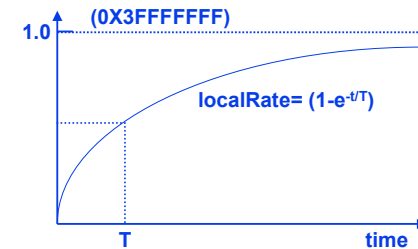


dvjRbrSlides

September 16, 2004, page 41

P1796

## DSP perspectives



dvjRbrSlides

September 16, 2004, page 42

The fairness flow rates are represented as normalized unsigned binary fractions. The largest unsigned value corresponds to “one”, so that only fractions can be represented. This is simpler than floating point, and well understood by DSP designers.