



# Preemptive Transmission advantages

## Is it worth the effort?

Rev. 2

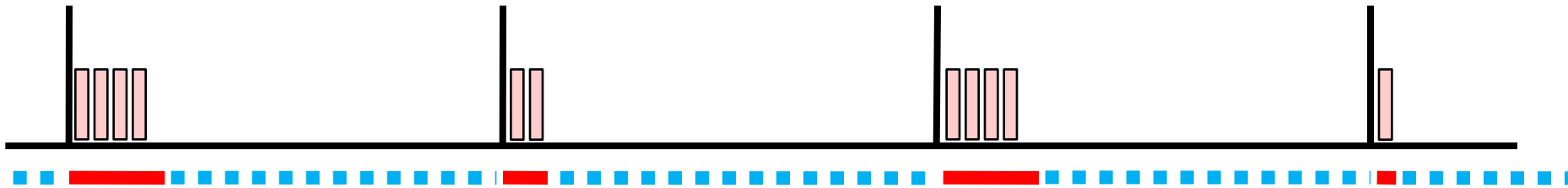
**Norman Finn**

[nfinn@cisco.com](mailto:nfinn@cisco.com)

<http://www.ieee802.org/1/files/public/docs2012/new-avb-nfinn-preempt-advantage-0112-v02.pdf>

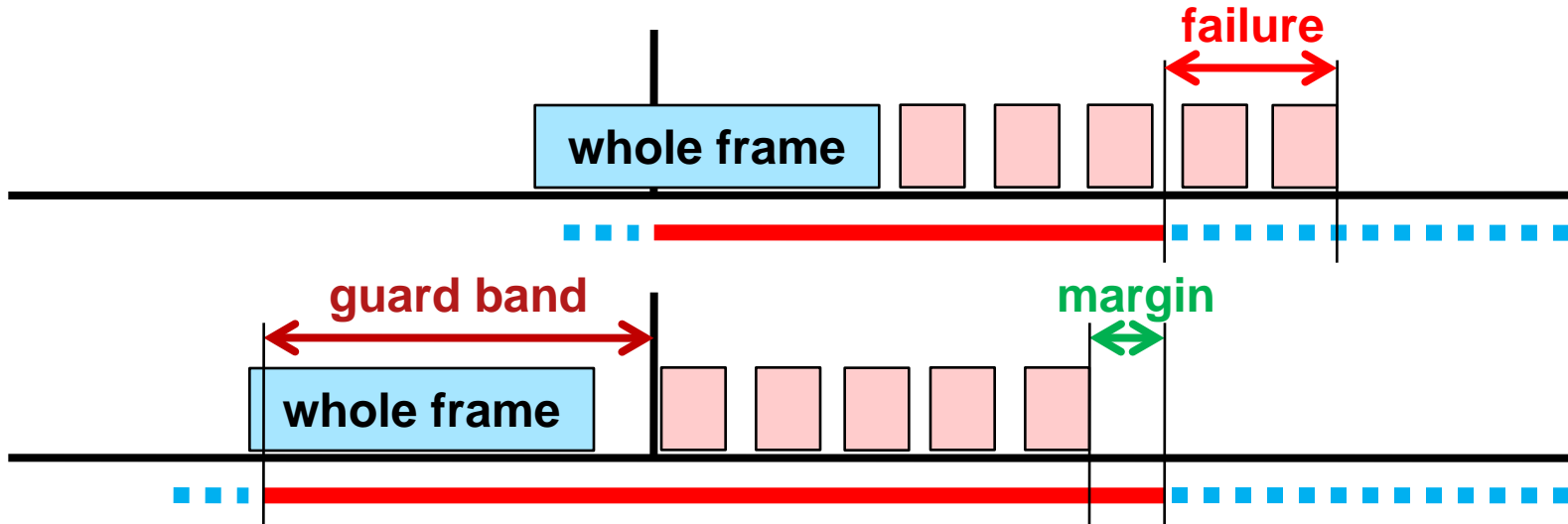
# Scheduling is required for real-time nets

- The real-time network scheduling model is:  
**communicate**, **compute**, **communicate**, **compute**, ...



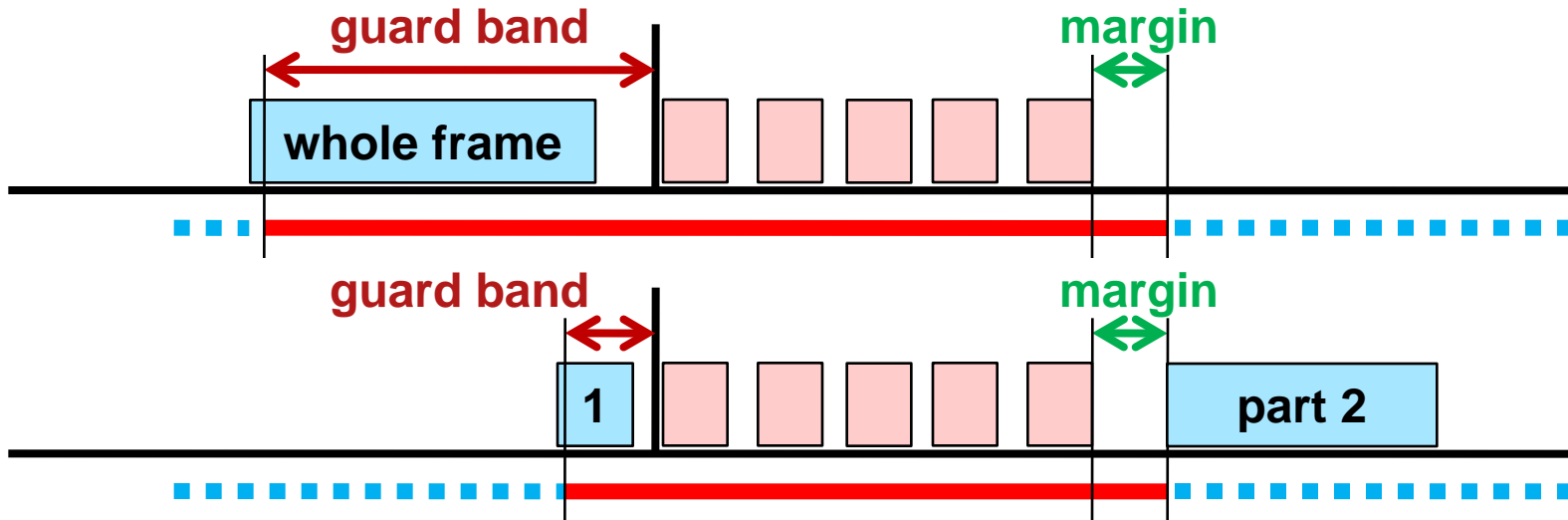
- Communication occurs at specified times.
- The timing is driven by the requirements of the critical application.
- Only by **strict scheduling** can we guarantee, **no matter what happens**, that we will respond to external events in a timely manner.

# Guard bands are necessary



- If an interfering frame starts transmission just before the start of a reserved time period, it can extend critical transmissions outside the window.
- Therefore, a **guard band** is required before the window starts, equal in size to the largest possible interfering frame.

# Preemption shrinks the guard band



- If preemption is used, the guard band need only be as large as the largest possible interfering fragment, instead of the largest possible interfering frame.
- It is easy to see that **the smaller the size of the time-reserved windows, the larger the impact of preemption.**

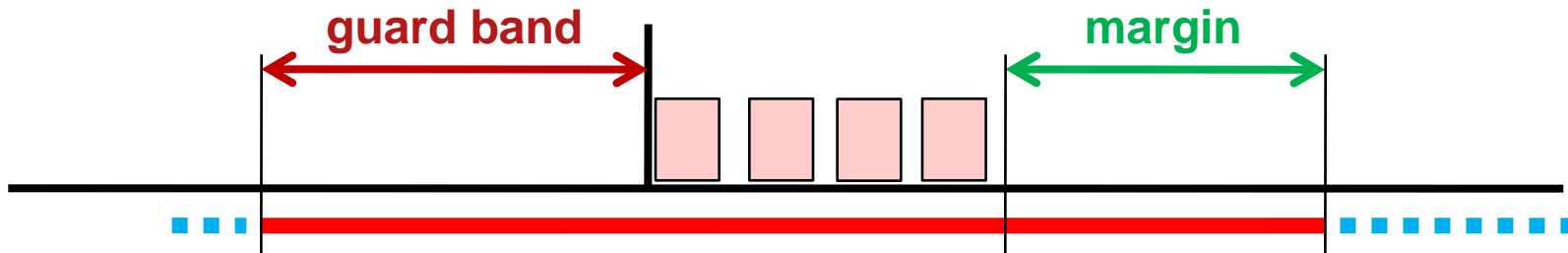
# Some numbers

- Let us assume that the time-critical data frames are typically small, say 128 bytes.
- Let us assume that standard 1522-byte data frames are permitted for all other traffic.
- We will use the standard 20 bytes for preamble, start of frame delimiter, and inter-frame gap.
- Let us assume that preempting a frame adds only an extra 20 bytes; this is the minimum practical penalty.
- We will assume that the worst case frame size is 127 bytes, which cannot be preempted. A 128-byte frame could be preempted and separated into two 64-byte fragments.

# Example 1

- In the first example, the time window is sized for **four 128-byte frames** with a margin of 4 more such frames (50% utilization of the window).
- The basic window size is  $8 * (128 + 20) = 1184$  byte times.
- Without preemption, we require a  $(1522 + 20)$  byte guard band, for a total window size of 2726 bytes.
- With preemption, we require a  $(127 + 20)$  byte guard band, for a total window size of 1331 bytes.
- Thus, whatever percentage of the total bandwidth is allocated to time-critical traffic, it requires **more than twice as much time** ( $2726/1331$ ) be reserved for that traffic if **preemption is not utilized**.

# Example 1



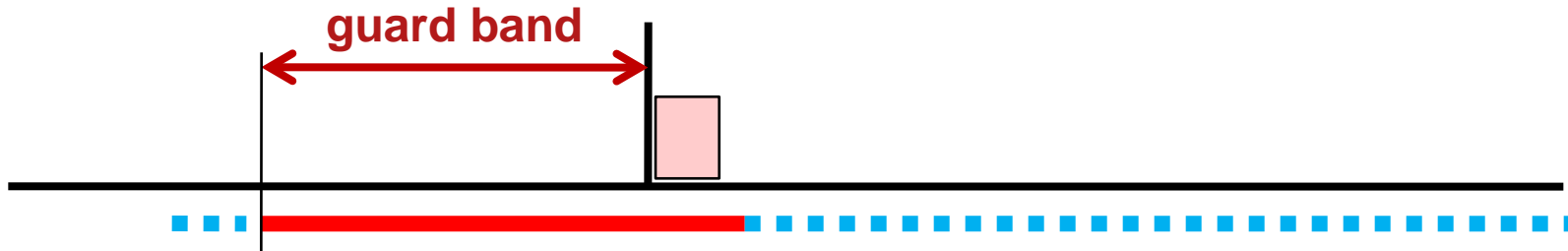
- 1184 byte window, including margin, plus guard band.
- Bandwidth is scaled up by increasing windows per second, not by making windows larger, because the applications determine the window size, not the available bandwidth.

## Example 2

- In the second example, the time window is sized for **one 128-byte frame** with no margin. This is perfectly possible if we assume that the switch will store a time-critical frame very briefly, and then transmit it at the appropriate moment.
- The basic window size is  $(128 + 20) = 148$  byte times.
- Without preemption, we require a  $(1522 + 20)$  byte guard band, for a total window size of 1690 bytes.
- With preemption, we require a  $(127 + 20)$  byte guard band, for a total window size of 295 bytes.
- Thus, whatever percentage of the total bandwidth is allocated to time-critical traffic, it requires **more than 5.7 times as much time**  $(1690/295)$  be reserved for that traffic if **preemption is not utilized**.



## Example 2



- 148 byte window, no margin, plus guard band.
- Bandwidth is scaled up by increasing windows per second, not by making windows larger, because the applications determine the window size, not the available bandwidth.

# Percent of time reserved for critical data

Link speed	Mbits/sec required for critical data alone	(no guard band)†		with preemption		no preemption	
		1184 B* window	148 B* window	1184 B* window + guard	148 B* window + guard	1184 B* window + guard	148 B* window + guard
100 Mbits	0.1	0.23%	0.12%	0.26%	0.23%	0.53%	1.32%
	1	2.31%	1.16%	2.60%	2.30%	5.32%	13.20%
	10	23.13%	11.56%	26.00%	23.05%	53.24%	132.03%
	30	69.38%	34.69%	77.99%	69.14%	159.73%	396.09%
1 Gbit	1	0.23%	0.12%	0.26%	0.23%	0.53%	1.32%
	10	2.31%	1.16%	2.60%	2.30%	5.32%	13.20%
	100	23.13%	11.56%	26.00%	23.05%	53.24%	132.03%
	300	69.38%	34.69%	77.99%	69.14%	159.73%	396.09%

\* Half of 1184-byte window is reserved for margin; none of 148-byte is margin.

† “No guard band” shows wastage from margin, preamble and inter-frame gap.

# Summary

- The window size for scheduled transmissions cannot be much larger than a very few full-sized frames, or the ability to support bandwidth reserved streams is compromised. In many applications, the window size corresponds to a single frame.
- Using single frame windows, it is impossible, without preemption, to allocate even 10% of the bandwidth to scheduled transmissions.
- **Scheduling transmissions requires wasting bandwidth.** The amount of **bandwidth wasted is less if preemption is allowed** than if preemption is not possible, the difference being a **factor of 1.5 to 6**, depending on the size of the scheduling window.

# Derivation of numbers in table

- A. “Mbits/sec required for critical data alone” is bits per second including frame from MAC addresses through CRC, but not preamble or inter-frame gap.

All subsequent columns show the percentage of time dedicated to the transmission windows in order to achieve the bit rate in **A**.

- B. “No guard band” columns are for the two window examples without guard band.  $(\mathbf{A} / (\text{line rate})) * ((\text{window size}) / (\text{MAC-to-CRC bytes per window}))$
- C. “With preemption” columns uses same calculation as **B**, but adds 127 + 20 byte time guard band to window size.
- D. “Without preemption” columns uses same calculation as **B**, but adds 1522 + 20 byte time guard band to window size.