Technical and economic feasibility for millisecond deterministic delay in Residential Ethernet

Michael Johas Teener

Plumblinks

mike@teener.com

Agenda

- Why millisecond deterministic delay?
- An example technical approach
- Complexity comparison
- Difficulty of implementation
- Summary

Why millisecond delay?

- Interactive control response
  - should be less than 50ms, best if less than 10ms
  - musical instruments need lowest values
- Simplifies multiple-path control and synchronization
  - some devices on networks will be serially connected
  - ultimate source to ultimate sink may take several trips on network
- Expectation of 1394-based devices
  - single 1394 bus has less than 300µs application-to-application latency

Why deterministic delay?

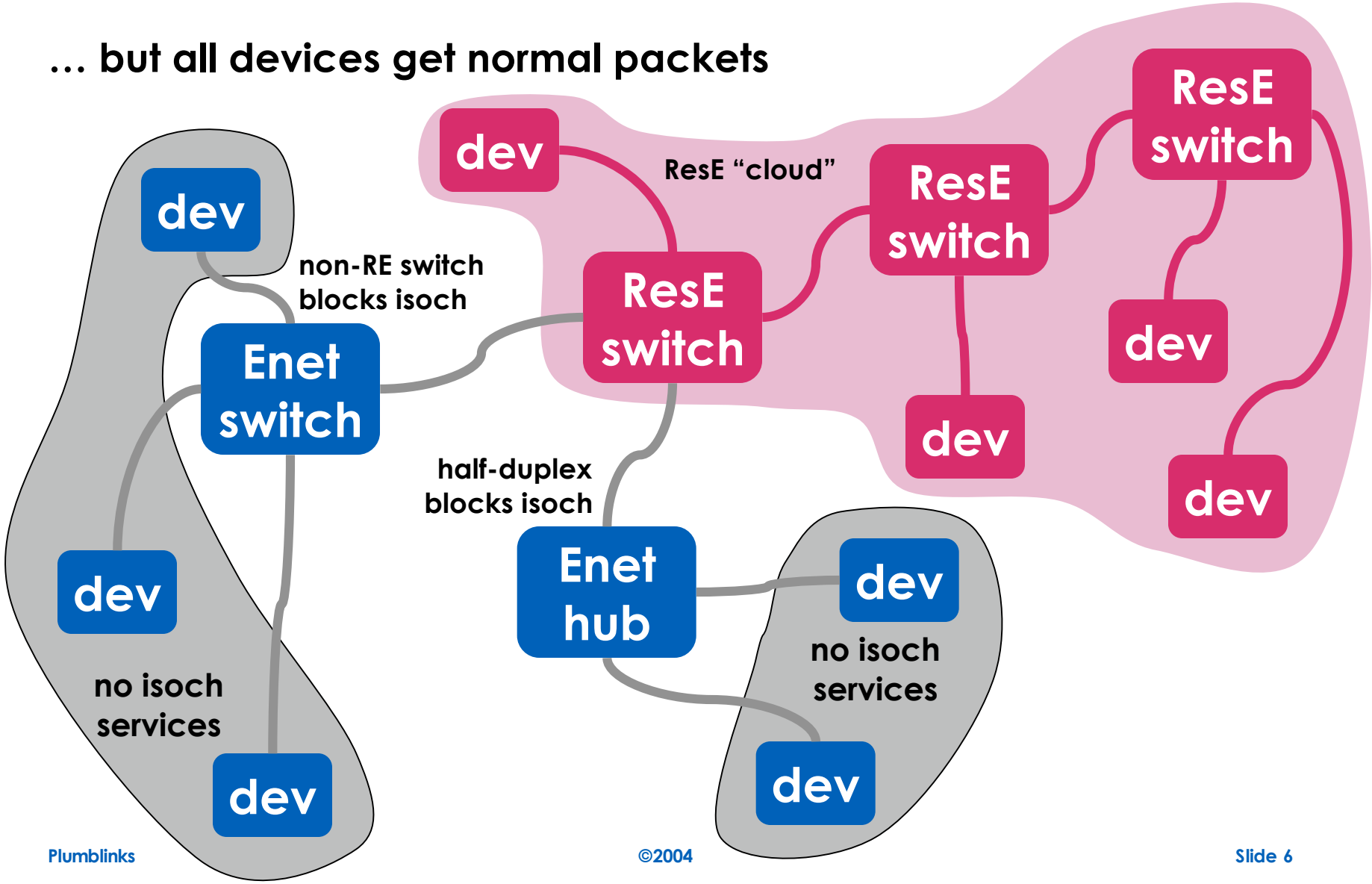
- Deterministic delay means bounds on both maximum *and* minimum!
- Buffer sizes are bounded and small
  - Buffer is sized at difference between max and min delay
- Expectation of 1394-based devices
  - 1394 devices usually have 250µs buffers

Example technical approach

- Provide synchronous services only on full-duplex links
  - but only normal 802.1D restrictions on number of bridges
  - some form of negotiation used to indicate if neighbor on a link is ResE-capable
- If neighbor is ResE-capable:
  - Provide well-known clock for synchronization
  - Synchronous packets are standard 802.3, but distinctly labelled
  - Admission controls for synchronous traffic on a per-link basis
  - Dynamic queue priority based on synchronous pacing

Full-duplex links only

… but all devices get normal packets



Clock distribution

- All ResE devices periodically exchange “current time” information
  - only with link neighbor, probably using something like MAC control services
- One station in ResE cloud is selected as clock master
  - other stations follow it using a very “stiff” filter
- Very accurate synch is possible
  - less than a microsecond
- Important part is every station knows the current “cycle”
  - cycle is a 8kHz counter

Synchronous packet labelling

- All packets are normal 802.3 format
  - unique length/type
- Use GMRP to handle multicast channel selection?
- Extra "talker channel" field to handle multiple streams from a single station
- Cycle time stamp
  - indicates the cycle that the packet was scheduled to transmit
  - perhaps a second one that indicates when scheduled for transmission from original source
    - … useful for delay calculations, but this could be done other ways
- Other fields to aid 1394 bridging
  - "synch", etc.

Admission controls

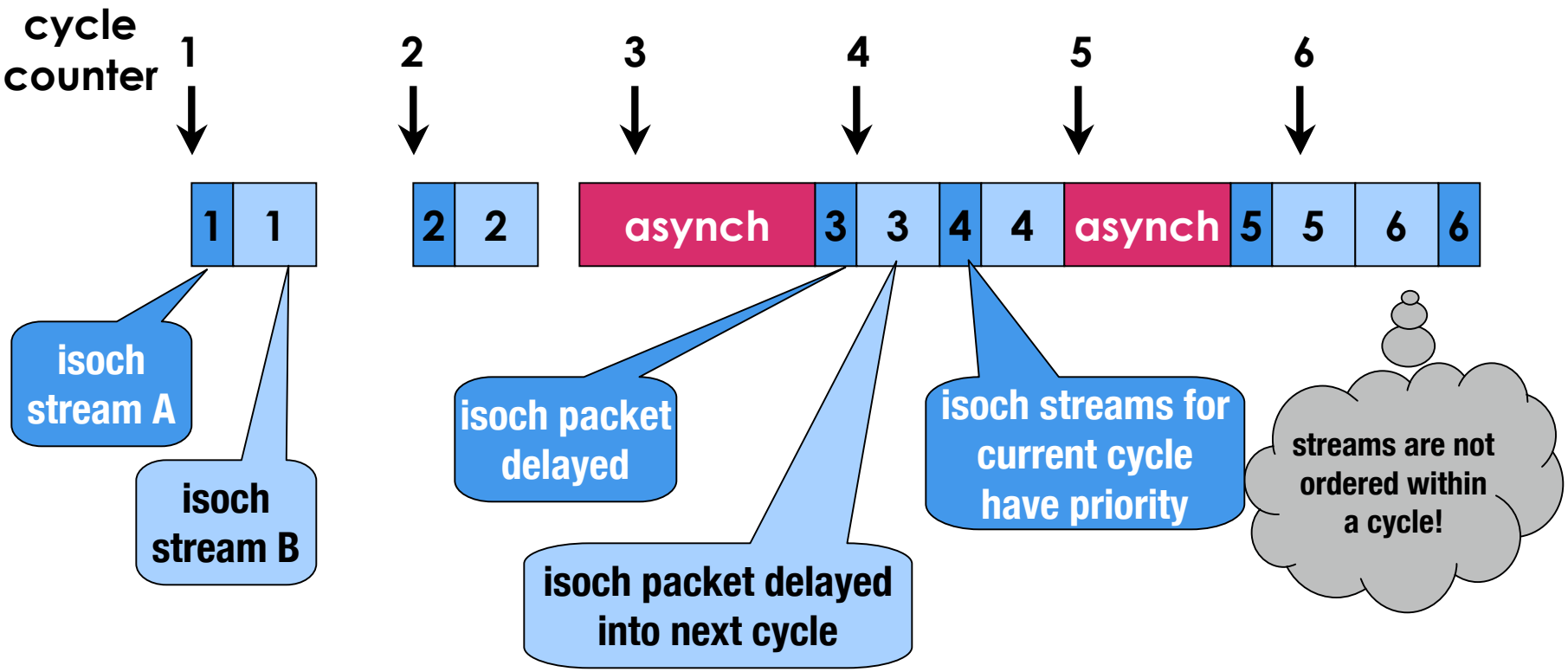
- Listener must confirm resources available along entire path to destination
  - sends “join request” control packet to talker with amount of bandwidth needed (in bytes/cycle)
  - intermediate bridges make reservation, update delay count and pass on control packet
  - if resources not available, packet is stamped as “unavailable”, but still sent to talker
  - talker returns “join response” packet to listener with status
    - status includes resource available (or not), and delay
- Obviously, various timeouts and disconnects affect this
  - fun future work!

Admission controls (2)

- Additional listeners also send “join” request
  - but this time an intermediate bridge can respond if it is already routing the stream
- Listener sends “leave” when done
  - only gets to talker if this is the last listener, bridges intercept all others
- Talker is required to send one packet every cycle
  - may be zero length
  - if missing for more than “x” cycles, can be used to take down the connection

Synchronous pacing

- Transmitter sends all packets labelled with cycle “n” as soon as possible in that cycle



Complexity comparison

- Worst case client MAC will be no more complex than 1394 "OHCI" (Open Host Controller Interface)
  - typically 150k gates including multiple buffers and powerful scripted DMA engine to perform RDMA-type transactions
  - will usually be much simpler for non-computer applications
- Bridge will need extra priority level and required management functions
  - Synchronous priority queue per output port
    250 µs of data per port max
  - Separate routing table for synchronous streams
    same structure as existing 802.1D
  - Resource manager
    new
  - Local clock
    new

Difficulty of implementation

- Example implementations of IEEE 1394.1 bridges exist
  - much more difficult that proposed RE bridge
    - 1394 has its own legacy to deal with!
  - early examples running at 800Mbit/sec were commercially shipped in 2000
    - using FPGAs!
- 802.1D bridges are a great basis to start from
  - many integrated examples, wide range of capabilities

Summary

- Millisecond deterministic delay is technically feasible
  - Example approach is not the only one
  - 1394-based systems exist now
- Millisecond deterministic delay is economically feasible
  - Very minor changes to client are needed
  - Modest changes to bridge are also required
    - Will be much smaller than 1000baseT PHY!

Therefore, bridging 1394 buses should be one of the requirements for Residential Ethernet

Thank you!