Canova Tech

The Art of Silicon Sculpting

PIERGIORGIO BERUTO
ANTONIO ORZELLI

IEEE802.3cg WG

PHY-Level Collision Avoidance

rev. 1.0

August 2nd, 2017

Overview

- Idea for half-duplex *multi-drop* short-reach PHY
  - Media access multiplexing protocol
    - Collision-Avoidance scheme
- Objectives
  - Interworking with standard CSMA/CD MAC
    - No modifications to MAC, everything done at PHY level
  - Beat CSMA/CD performance, especially at high bus loads
  - Lower complexity than existing TDMA systems (e.g. EPON)

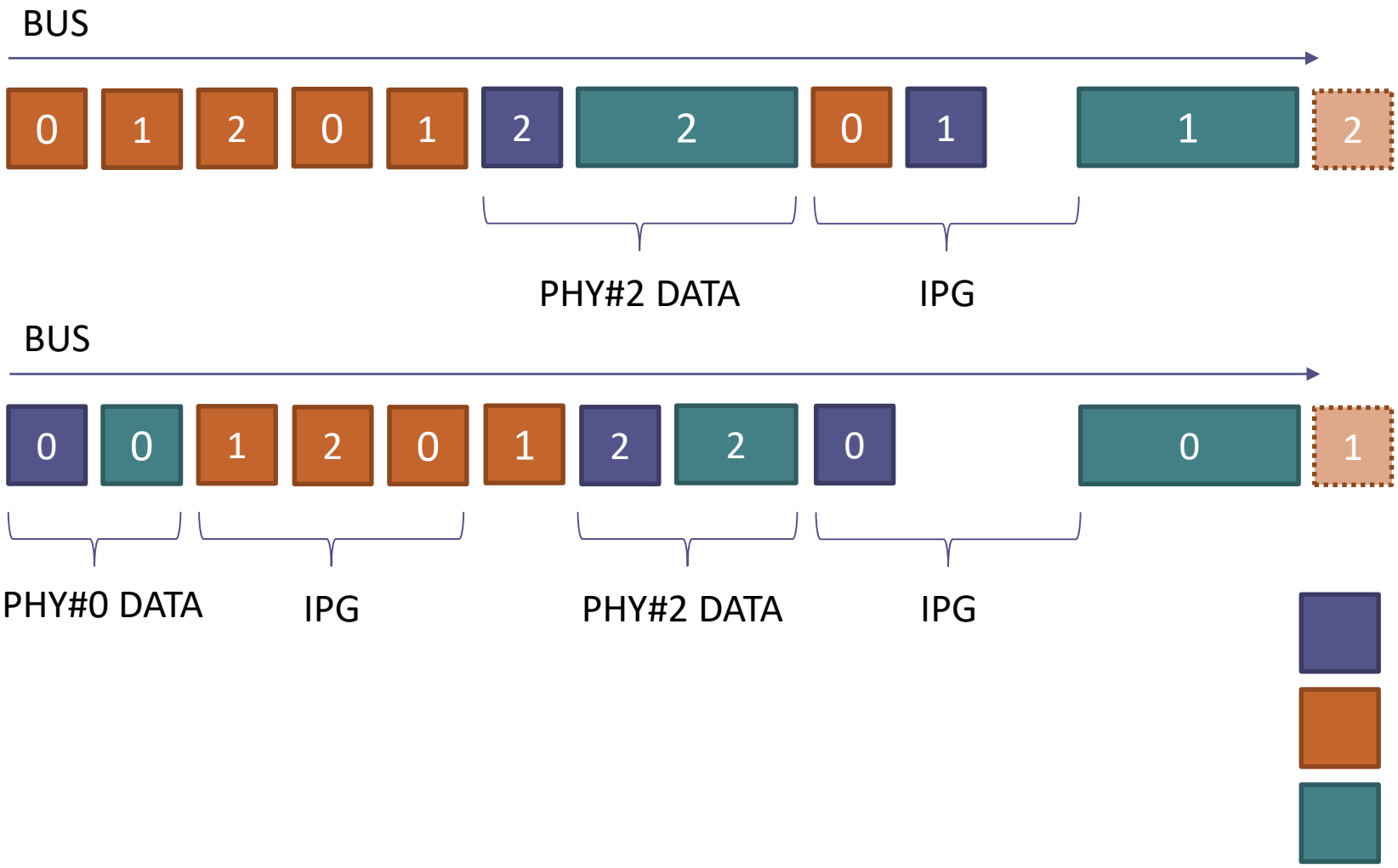
MIX of CSMA and TDMA

-exploiting limited number of nodes-

In a nutshell

- Assumptions
  - Up to 6 nodes with pre-configured unique IDs (0-5)
    - ID = 0 is the "master" node
  - Interface to 10Mbit half-duplex CSMA/CD capable MAC
  - PCS encoding shall allow more signaling (SSD, ESD, ACK, NACK)
- Basics
  - PHYs send packet in sequence (no collisions)
    - Also good for fairness! (no starvation)
  - Starting from master, each PHY sends ACK + DATA or NACK in turn, following unique ID order.
    - no additional overhead as long as ACK/NACK < IPG *(see next slide)*
    - no waste of bandwidth (PHYs with nothing to send just "skip the turn")
  - Implicit NACK after timeout to handle absent/link-down nodes
    - Constraint on maximum allowed TX/RX latency
      - Trade-off with max achievable throughput

Example with 3 PHYs



What's the catch?

Devil is in the details: how does a PHY know whether the MAC has something to send?

- It's not possible via MII to have such information **in advance**
  - [ugly] Buffering the packet? → increase latency, relative cost
  - [bold] Modify the MAC? → against objectives
  - Exploiting existing CSMA/CD (COL, CRS) → let's see…!

Interface with CSMA/CD MAC #1

- CSMA/CD MAC transmit process (from IEEE 802.3, clause 4)
  - If line is busy (CRS = 1) → wait (defer transmission)
  - Wait IPG (at least 96 bits)
  - Start transmitting, despite line becoming busy again
    - If a collision is detected (COL = 1) → backoff:
      - Send jam for 32 bit times, stop transmission
      - Retry after random(0, ATTEMPTS) * 512 bit times
      - If ATTEMPTS > attemptLimit → give up (discard packet)
- CRS / COL can be used to have the MAC defer transmission until next handshaking
  - Use CRS to have the MAC defer transmission
  - Use COL at most once and only at beginning of a packet
    - MAC is ready to re-send in at most 32 + 512 = 544 bit times
      - Less than minimum packet size (576 bits)
      - COLLISION AVOIDED, NO WASTE OF BUS TIME!

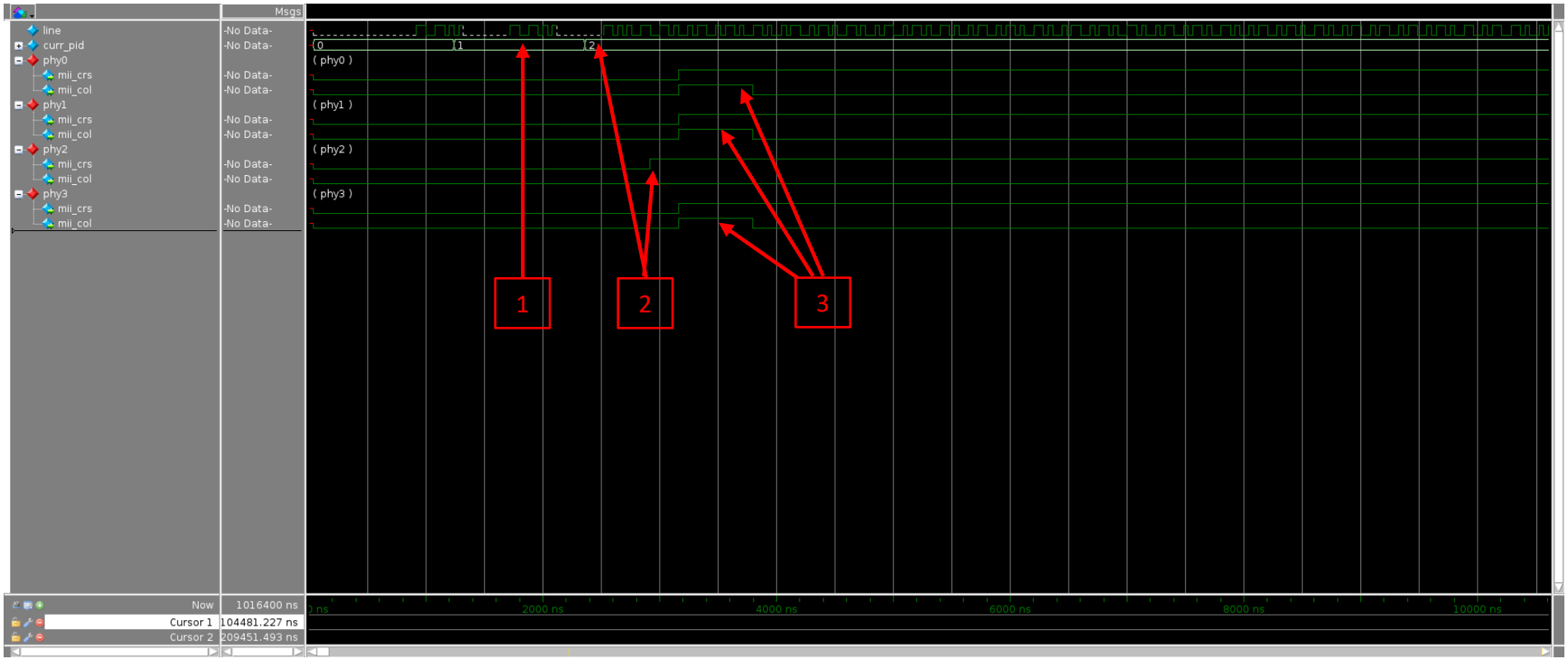
Interface with CSMA/CD MAC #2

- PHY could be designed as follows:
  - CRS shall indicate when data (not ACK/NACK) is on the line → standard behavior
  - If TX_EN = 1, start buffering (**small** buffer, i.e. 2 x handshaking time at most).
  - If it's our own handshake time → send ACK and DATA (DONE)
  - If another PHY's ACK is received instead
    - Set COL = 1 (force the MAC to backoff, ATTEMPTS = 1)
    - Keep CRS = 1 (force MAC to keep the packet, avoid further ATTEMPTS)
    - Flush the buffer (discard JAM)
    - At next own handshake time, set CRS = 0 (allow MAC to send), send ACK + DATA
      - IPG honored by the MAC!
  - The backoff can only occur once, at the very beginning of the frame!
    - Mission accomplished!

Simulation

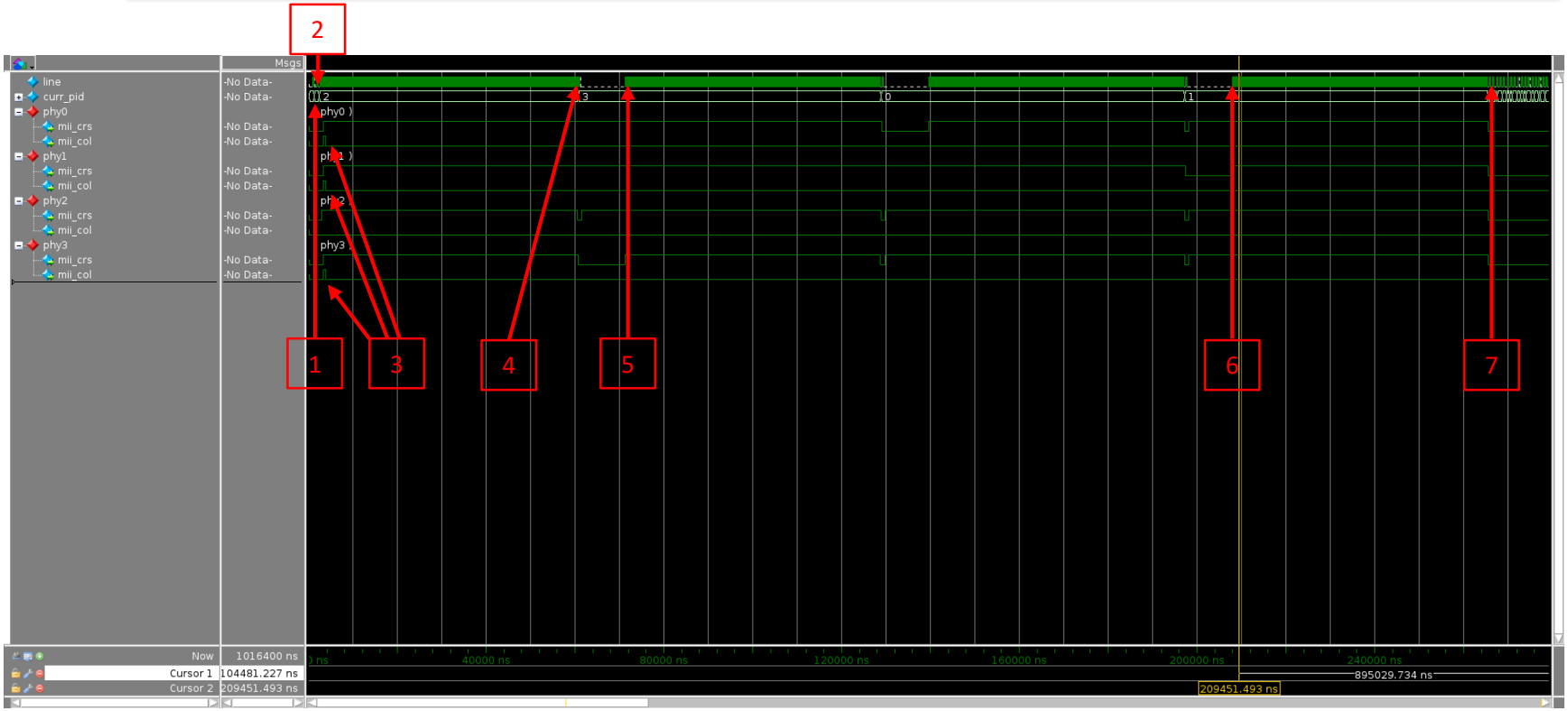
- Digital simulation of up to 6 PHYs
  - Verilog behavioral model of MAC and multidrop PHY
  - Encoding 4b5b + DME (25Mhz BW)
    - Just as an example: not the focus of this presentation
    - Use 5b S/R commands to represent ACK/NACK
  - PHY buffer of 12 bytes
- Proof-of-concept of handshaking protocol
  - Measure of throughput under different load conditions
    - Comparison with point-to-point full-duplex PHY
      - LOAD = 100% each MAC sends data as soon as it can

Simulation results #1



- Scenario: 4 PHYs, each needs to send one packet at the same time
  1. All PHYs start buffering during NACK of PHY #1
  2. PHY #2 is the first to send ACK + DATA
  3. All other PHYs assert COL to force the MAC to backoff (NOTE: this is NOT a real collision!) and keep CRS high
  4. End of PHY #2 packet, PHY #3 sends ACK and de-assert CRS.
  5. PHY #3 sends data after IPG
  6. Last PHY (1) sends data
  7. All PHYs send NACK (idle) as they have nothing more to send

Simulation results #1



- Scenario: 4 PHYs, each needs to send one packet at the same time
  1. All PHYs start buffering during NACK of PHY #1
  2. PHY #2 is the first to send ACK + DATA
  3. All other PHYs assert COL to force the MAC to backoff (NOTE: this is NOT a real collision!) and keep CRS high
  4. End of PHY #2 packet, PHY #3 sends ACK and de-assert CRS.
  5. PHY #3 sends data after IPG
  6. Last PHY (1) sends data
  7. All PHYs send NACK (idle) as they have nothing more to send

Simulation results #3

TOT PHYS	XMIT PHYs	THROUGHPUT %
2	1	-1.18
3	1	-2.35
4	1	-3.53
5	1	-2.35
6	1	-5.88
2	2	-0.59
3	3	-0.59
4	4	-0.59
5	5	-0.59
6	6	-0.59

Relative throughput loss, compared to point to point full duplex case

PKT_SZ = 60 B payload (i.e. no preamble, CRC)
LOAD = 100%

TOT PHYS	XMIT PHYs	THROUGHPUT %
2	1	-0.1
3	1	-0.19
4	1	-0.29
5	1	-0.19
6	1	-0.48
2	2	-0.05
3	3	-0.05
4	4	-0.05
5	5	-0.05
6	6	-0.05

PKT_SZ = 1500 B payload
LOAD = 100%

Next Steps

- More simulations
  - Any interesting parameter to measure?
    - e.g. Latency?
  - More use-cases?
- Add logic for nodes joining / leaving the BUS
  - e.g. use different signaling for master's ACK/NACK to allow other PHYs to achieve synchronization
- Robustness
  - Errored ACK/NACK handling
    - Possible idea: use robust coding of ACK/NACK so that the chance of losing **all** of the bits is negligible → in case of error, don't transmit and re-sync on master ACK.
  - Corner cases study
- Optional
  - Auto-negotiation of IDs (instead of static config)?
    - e.g. use plain CSMA/CD for electing the master
  - On-the-fly election of new master for failover?

Thank You !