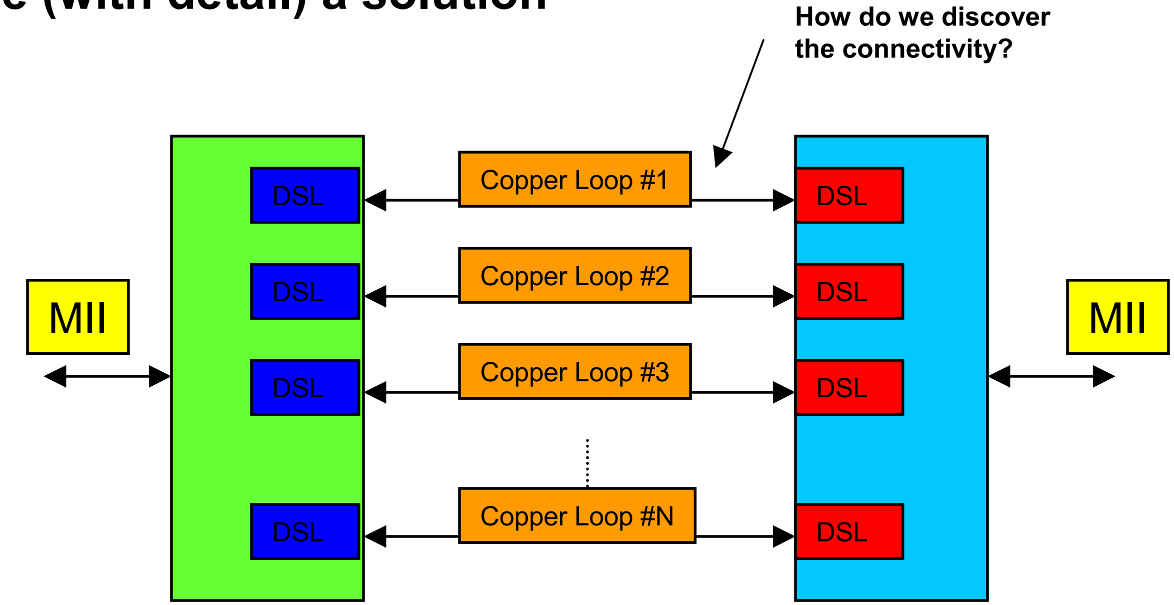
Loop Aggregation Discovery

Hugh Barrass

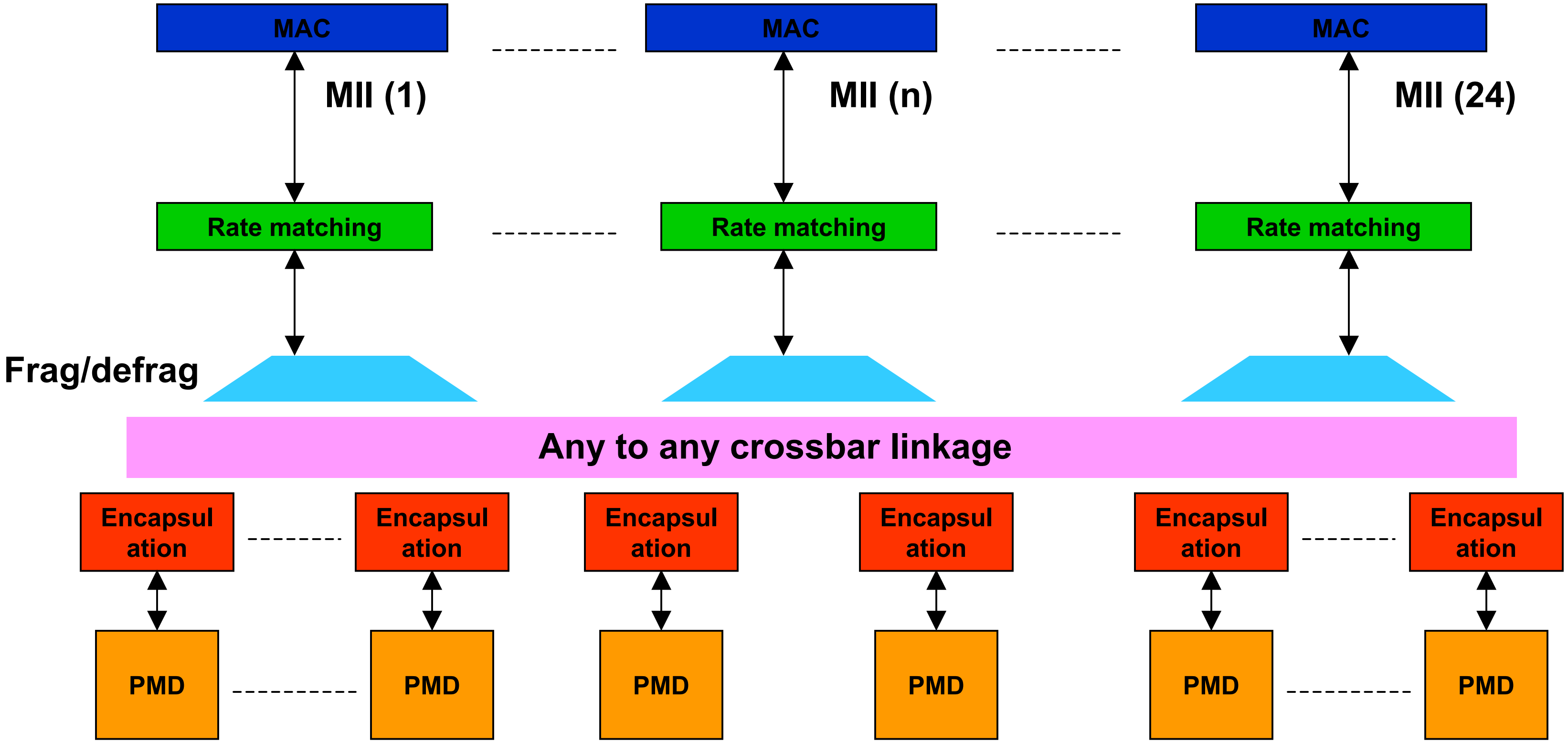
(supplemental to baseline proposed by Klaus Fosmark)

Loop Aggregation Discovery

- Loop aggregation as described by Klaus Fosmark (fosmark_1_0302.pdf) requires a simple and resilient discovery mechanism
- Consider options
- Propose (with detail) a solution



Reminder of “dream system”



- Any connection between MII’s and PMI’s
- Implementation could use arbitrary number of MII’s vs PMI’s
- System vendor could choose optimal ratio

Discovery problems

- Default settings at either end may not permit passage of frames across all loops
  - e.g. # PMI > # MII ➔ some PMI's not used until loop aggregation initialized
- "Loop confusion" means that no assumptions can be made about connectivity
  - Can't assume that PMI-2 at LT connects to PMI-2 at NT (even if PMI-1 connects)
- Possibility that alien devices are performing similar function in same plant
  - More unbundling difficulties…
- Host may address PCS (& thus MII) directly, or may address PMA/PMD (thus PMI) directly
  - Mechanism for explicit access to EOC/VOC for a specific loop

Discovery options

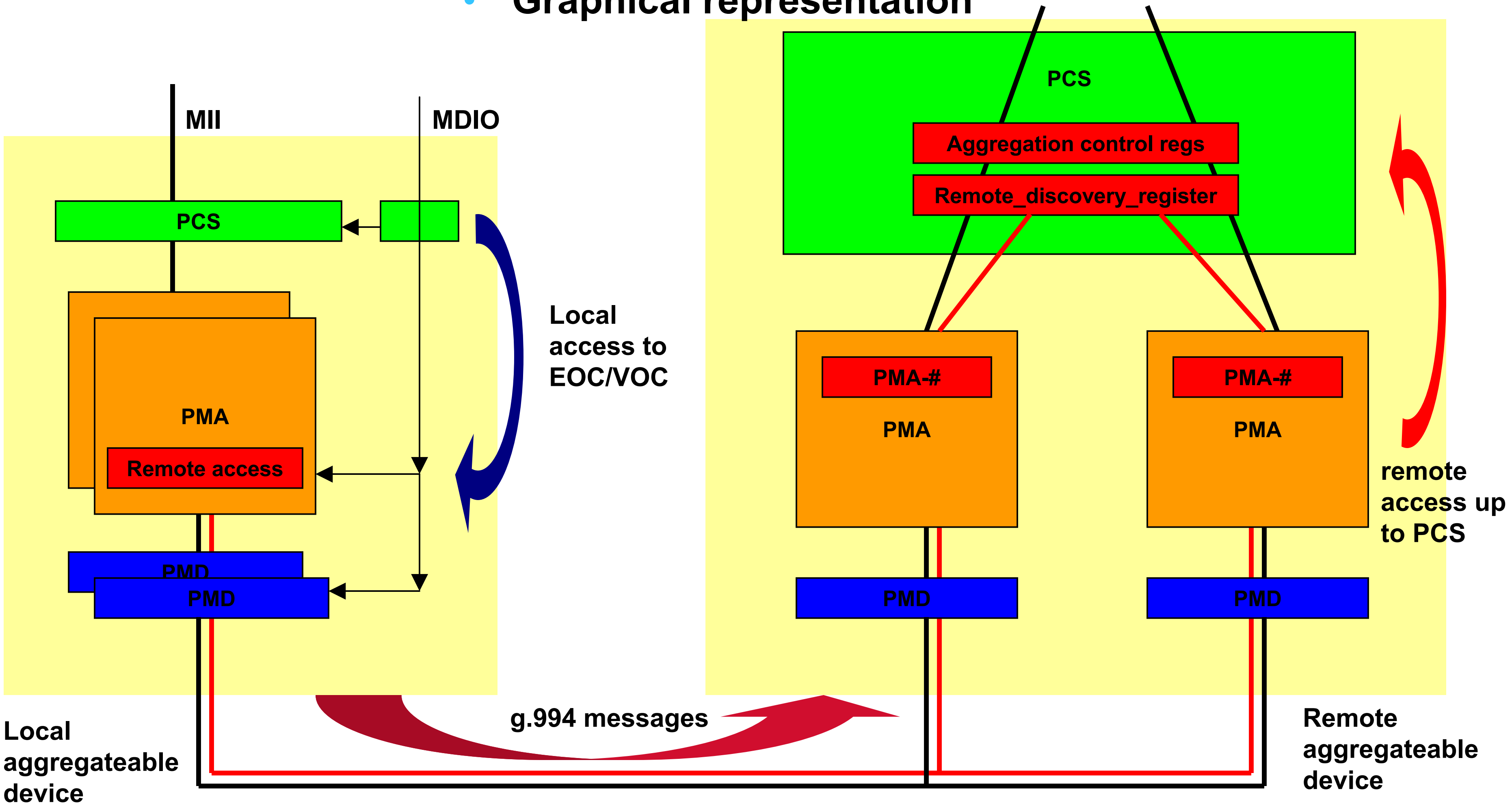
- Extend 802.3ad - LACP
  - Differences to link aggregation too great
  - Danger of overloading existing function (& breaking it!)
- New frame based discovery protocol
  - Collision problem for cross-connected installations
  - No way for MAC control to “direct” a frame out of a particular PMI
  - No way to tell MAC control which PMI received the frame
- Use new PHY layer mechanism
  - Low level, limited flexibility
  - Keeps “architectural purity” of layers
- ==> Propose PHY layer mechanism

Required PHY functions

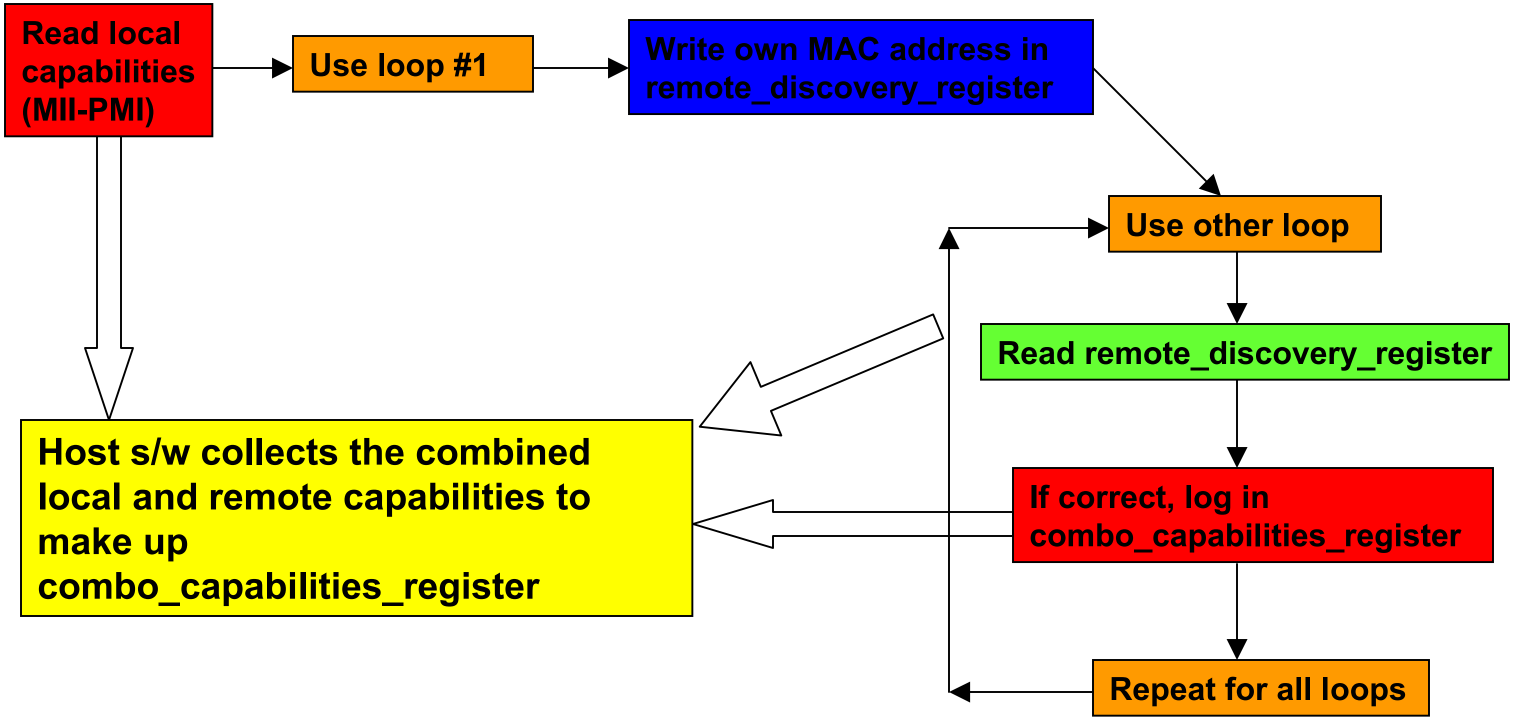
- Mechanism for MDIO access to local & remote PMA/PMD, PCS
  - Already a requirement for OAM, PMD and loop aggregation control
- CPE device must fix its MII to PMI mapping before discovery
  - Each PMI is fixed to one MII - otherwise complexity increases exponentially
  - Limits capabilities, linkage written by CO after discovery
- Remotely writeable and readable register in PCS
  - Ideally >48 bit, may require multiple operations to read/write
  - Only need 1 per MII in any aggregateable PHY device
  - Referred to as “remote_discovery_register”
- Host access to remote register
  - Uses MDIO addressing to access local PMA/PMD (1-32)
  - This gives access to remote PMA/PMD across loop – use g.994 messaging
  - Read/write “remote_discovery_register”

Required PHY functions (2)

- Graphical representation



Discovery process


Read local capabilities (MII-PMI)
Use loop #1
Write own MAC address in remote_discovery_register
Use other loop
Read remote_discovery_register
If correct, log in combo_capabilities_register
Repeat for all loops
Host s/w collects the combined local and remote capabilities to make up combo_capabilities_register

NB: could use other unique code instead of MAC. Also could add pseudo-random extension to MAC to extend function or increase security

- Note that this process is implemented in the host system

  It is assumed that the host system will optimize serial vs parallel implementation

- This process must be repeated for each "aggregation domain"

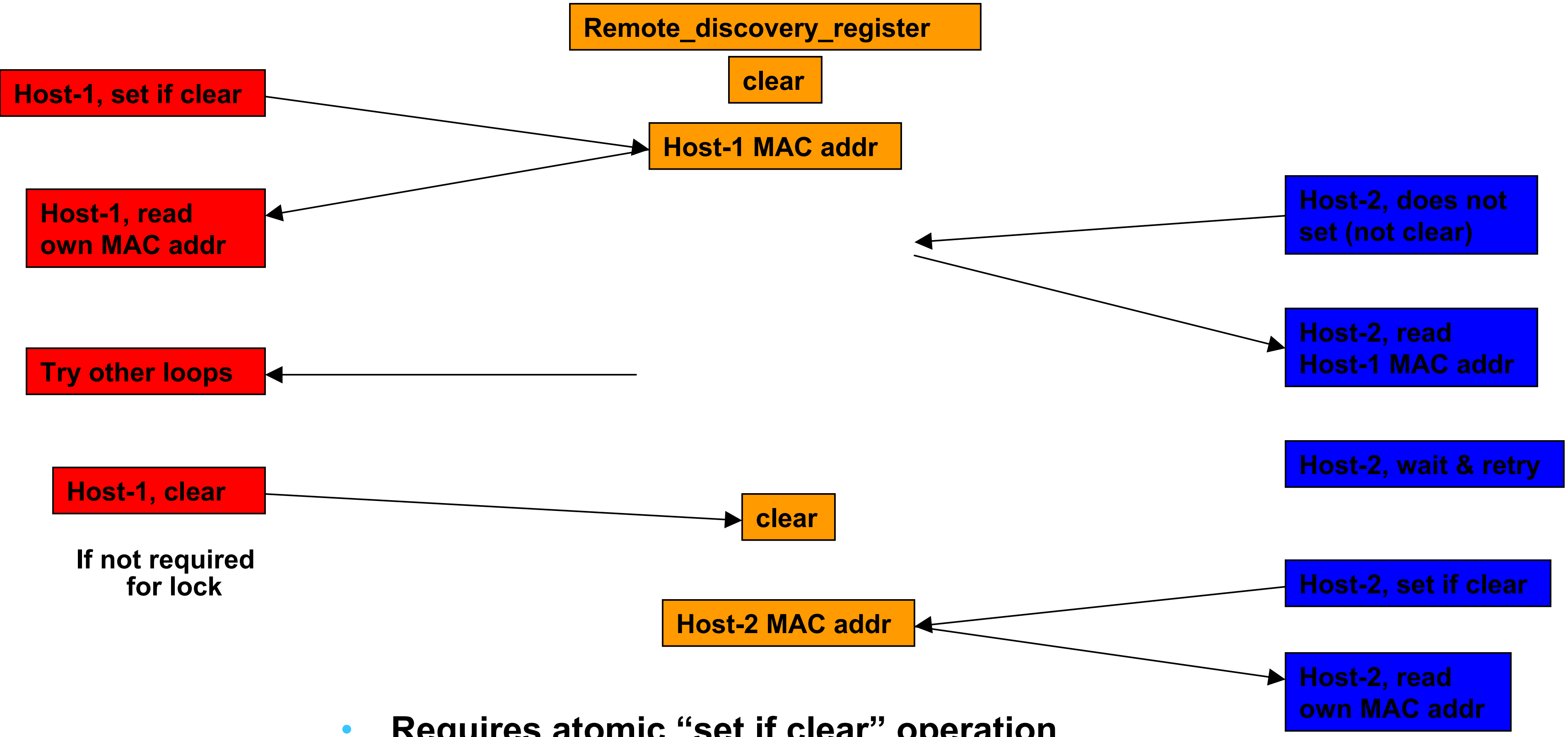
After discovery

- Host has a map of all possible aggregation possibilities
- 802.3ah does not specify how to use aggregation
  - Manual, automatic etc.
  - Locking is a problem
- Control of loop aggregation is master-slave
  - Similar approach to PMD control
  - Program remote aggregation control registers (linkage)

Problem!

- What if two hosts are trying to discover in the same binder?
- Needs a mechanism for collision detection or avoidance
- Write host MAC address into remote_discovery_register
  - Gives means of collision detection
  - Needs atomic operation for guaranteed collision avoidance
- Explore collision scenarios with and without atomic operation
  - Atomic - “read register and write xxx if zero”
  - Similar to operations used in parallel computing

Collision detection/avoidance



Remote_discovery_register

clear

Host-1, set if clear

Host-1 MAC addr

Host-1, read own MAC addr

Host-2, does not set (not clear)

Host-2, read Host-1 MAC addr

Try other loops

Host-2, wait & retry

Host-1, clear

clear

If not required for lock

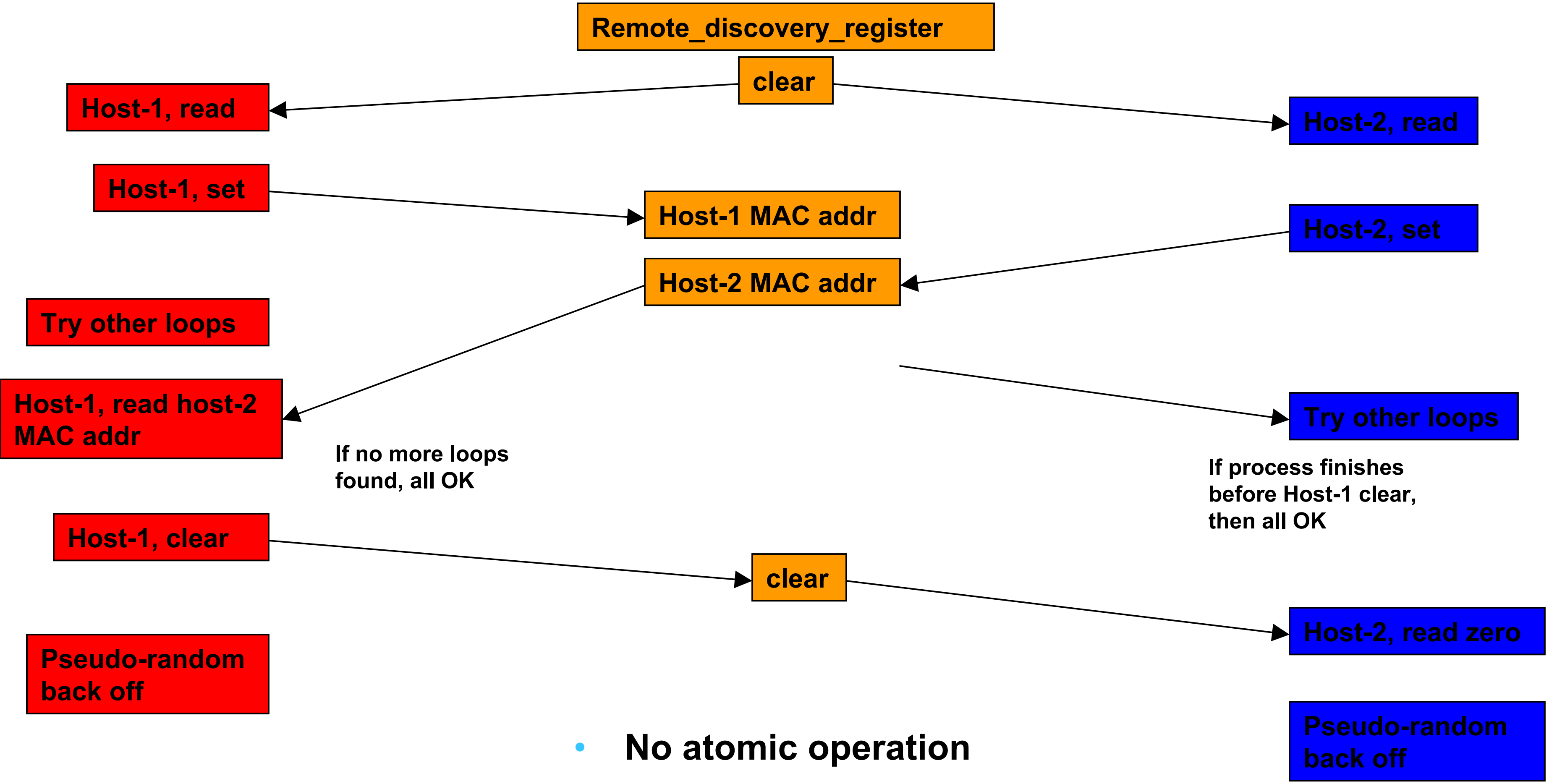
Host-2, set if clear

Host-2 MAC addr

Host-2, read own MAC addr

- Requires atomic “set if clear” operation
- CPE register must have self-clear mechanism

Collision detection/back off



Remote_discovery_register

clear

Host-1, read

Host-2, read

Host-1, set

Host-1 MAC addr

Host-2, set

Host-2 MAC addr

Try other loops

Host-1, read host-2 MAC addr

Try other loops

If no more loops found, all OK

If process finishes before Host-1 clear, then all OK

Host-1, clear

clear

Host-2, read zero

Pseudo-random back off

Pseudo-random back off

- No atomic operation
- In most cases, read then set is not interrupted by other host => same as for atomic operation

Summary – so far…

- Method for Loop Aggregation discovery
- Compatible with Klaus' baseline
  - fosmark_1_0302.pdf
- Uses PHY layer communication for control of PHY layer aggregation
  - Architectural purity!
- Minimal overhead
  - R/w pathway through remote PMA/PMD to remote PCS
  - 48 bit register
  - Larger register will ease parallel operation for large concentrator
- Special requirements
  - Atomic operation
  - Auto reset timeout

Proposal

- Add remote_discovery_register
  - 48 bit width
  - One per MII (aggregateable PHY)
  - Accessible through EOC/VOC
- Atomic operation
  - “set if clear”
  - Trivial implementation – preferable to pseudo random back off
- Only register is implemented in PHY, all else in host s/w
  - Algorithm may be proposed for reference
- CPE linkage registers must be remotely writeable
  - To allow CPE to know the results of the discovery

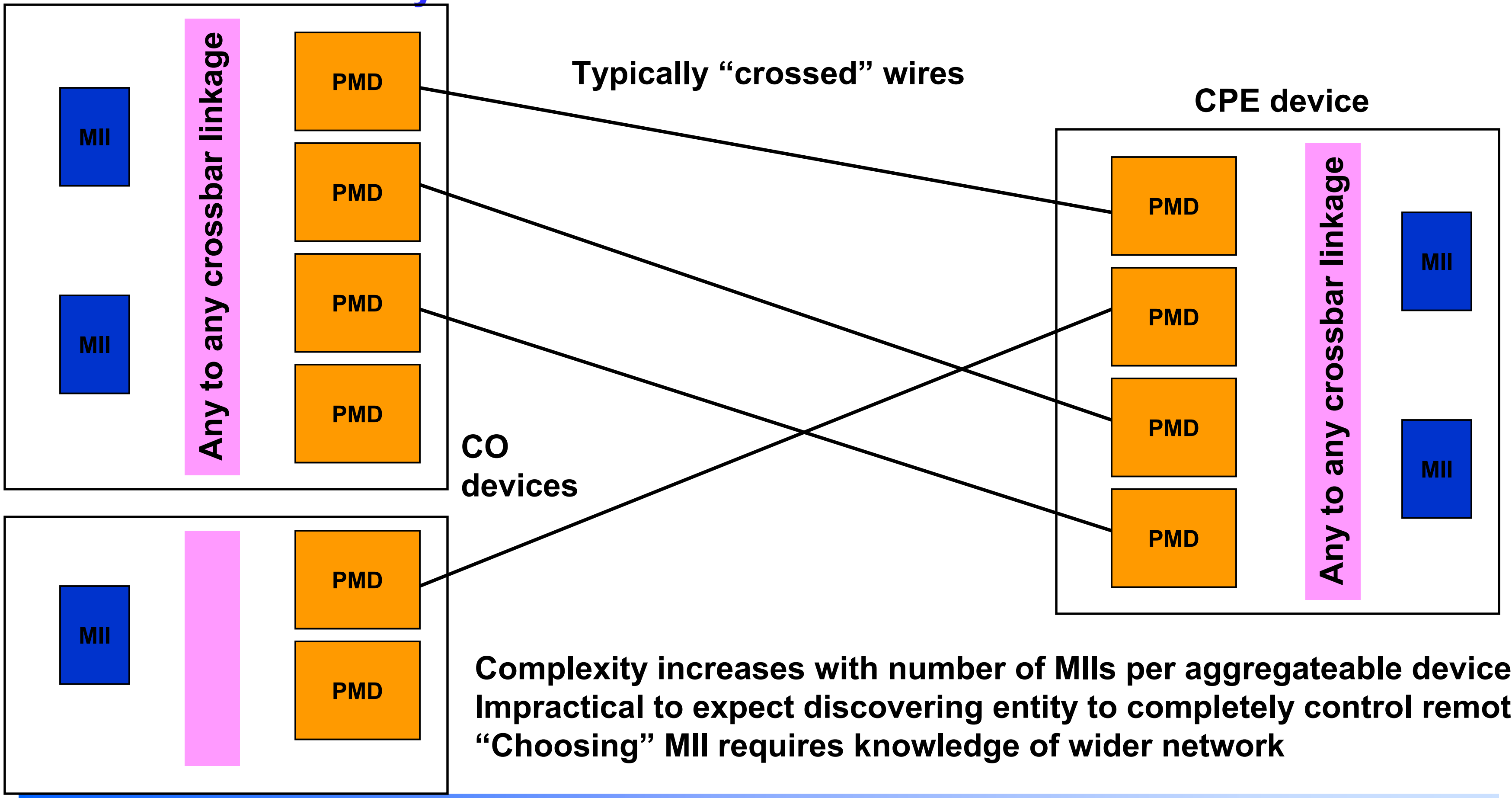
Further complexities

- "Dream System" at both ends
  - Mind-blowing complexity
  - Restrictions and simplifications
  - Limitations and practical applications
- Multiple aggregateable hosts cross-connected
  - Necessity for locking
  - Levels of separation…
- Other issues
  - Remote_aggregation_register self clear
  - Implications for fast failover

High-end CPE

- CPE capable of multiple MIIs & flexible MII to PMI connections

i.e. “dream system”

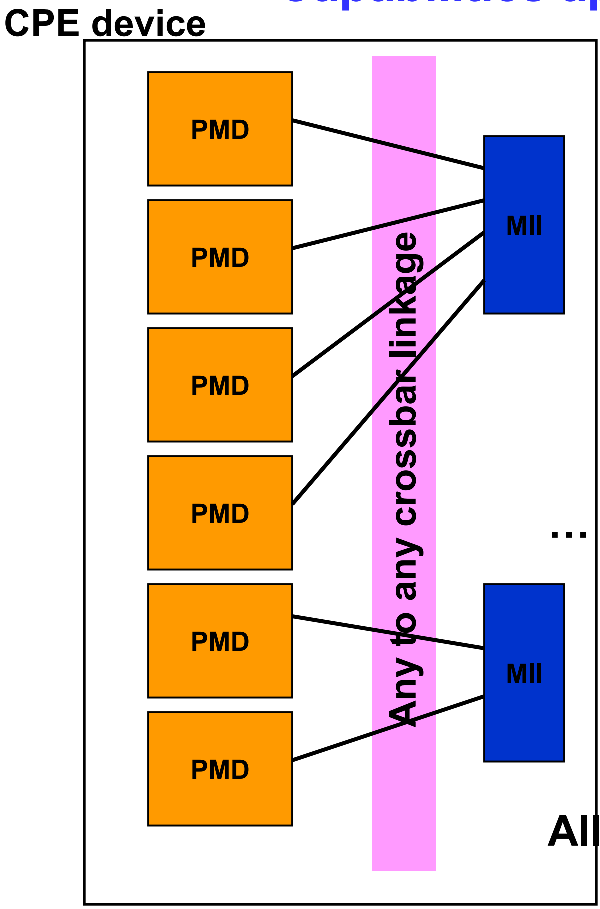

MII
MII
Any to any crossbar linkage
PMD
PMD
PMD
PMD
MII
PMD
PMD
Typically “crossed” wires
CPE device
PMD
PMD
PMD
PMD
Any to any crossbar linkage
MII
MII
CO
devices

Complexity increases with number of MIIs per aggregateable device
Impractical to expect discovering entity to completely control remote xbar
“Choosing” MII requires knowledge of wider network

Simplification

- CPE chooses PMI to MII mapping before enabling links

Capabilities appear fixed to discovering entity



MII to PMI mapping could be set by any means…

Software, DIP switches, fixed by system PCB etc.

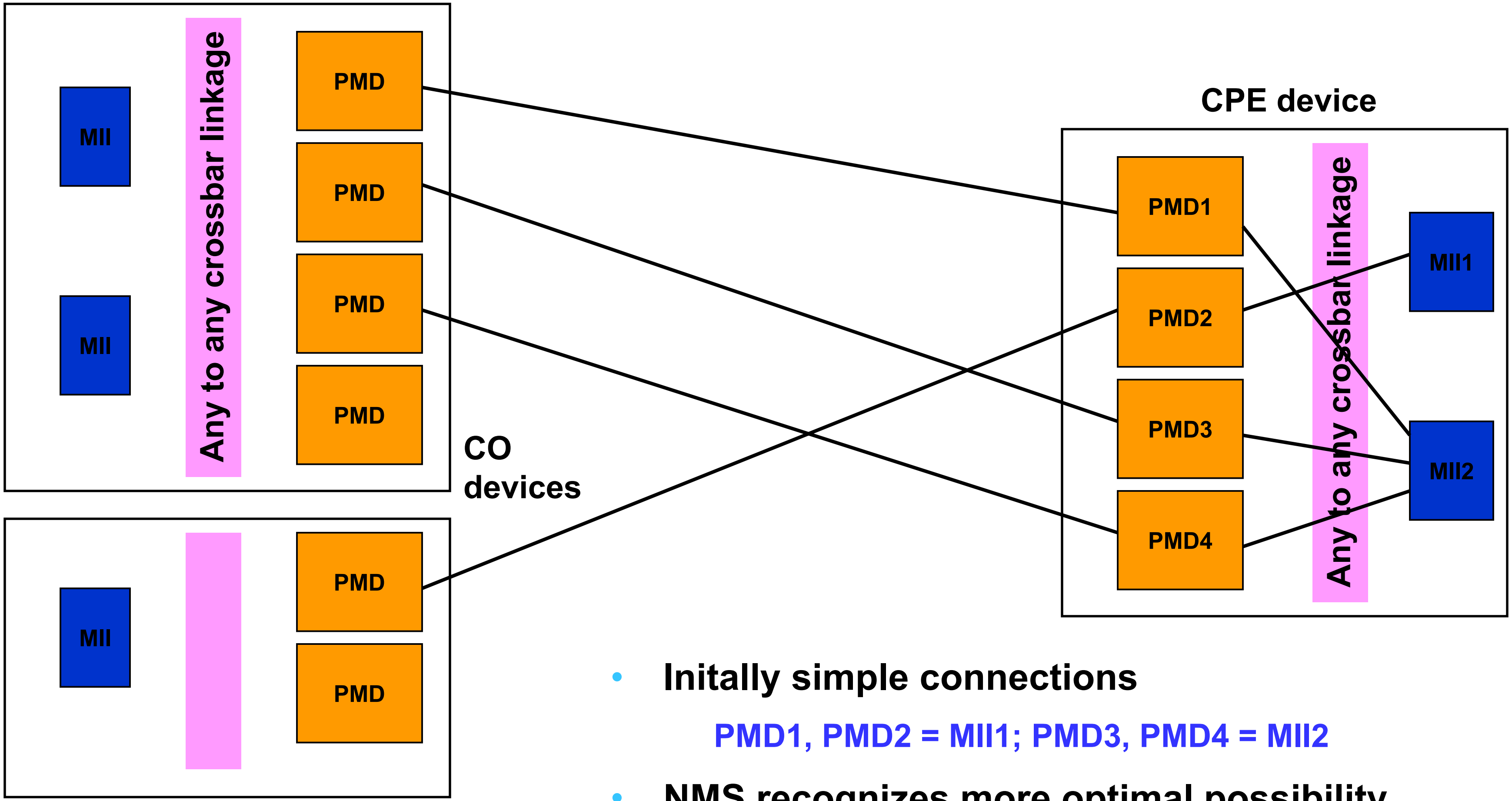
Lack of flexibility may mean non-optimal configuration…

… unless a higher layer entity (e.g. NMS) changes CPE settings

First phase discovery at physical layer
NMS changes CPE MII-PMI settings
Rerun discovery (or not!) to set optimal config

Allows optimal solutions without burdening simple operation

After optimal configuration



- Initally simple connections
  PMD1, PMD2 = MII1; PMD3, PMD4 = MII2
- NMS recognizes more optimal possibility
- Reconfigures CPE (as shown)

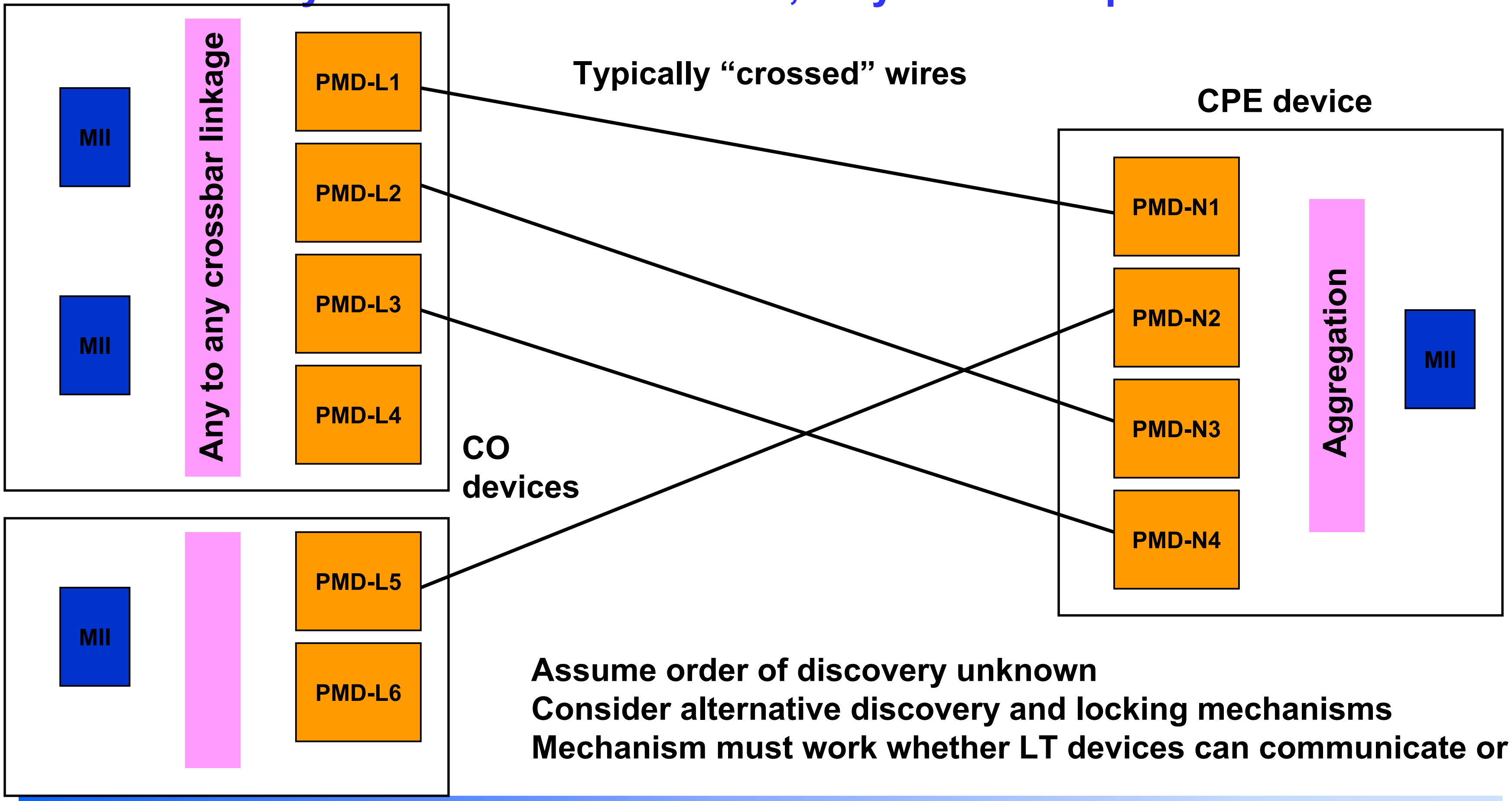
Summary of restrictions

- NT device must fix PMI to MII mapping before enabling links
  - Could be software, firmware, permanent fix etc.
  - Each set of PMIs (with one MII) behaves autonomously
  - One remote discovery register per MII
- PMI to MII mapping may be changed
  - Scope for optimization by other entities
  - All links must be taken down during PMI to MII change
- Reduced complexity for discovery
  - Ultimate optimization still possible

Problems of Cross Connection

- **Multiple LT devices connected to NT device**

Maybe from same head end, maybe from separate devices



Assume order of discovery unknown
Consider alternative discovery and locking mechanisms
Mechanism must work whether LT devices can communicate or not

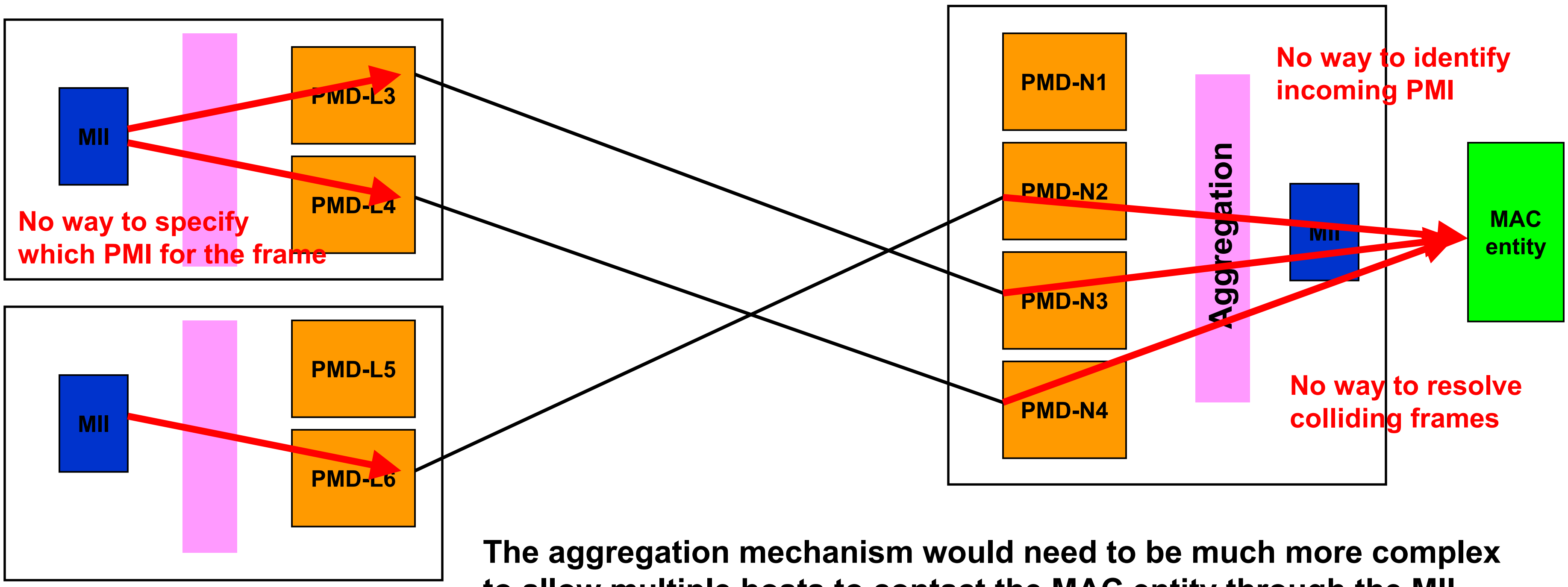
MAC level discovery

- Only PHY-layer mechanisms will work for discovery
  - MAC layer (or any other) does not work
  - (this includes OAM frames)

CO devices

Simplified problem

CPE device



No way to specify which PMI for the frame

No way to identify incoming PMI

No way to resolve colliding frames

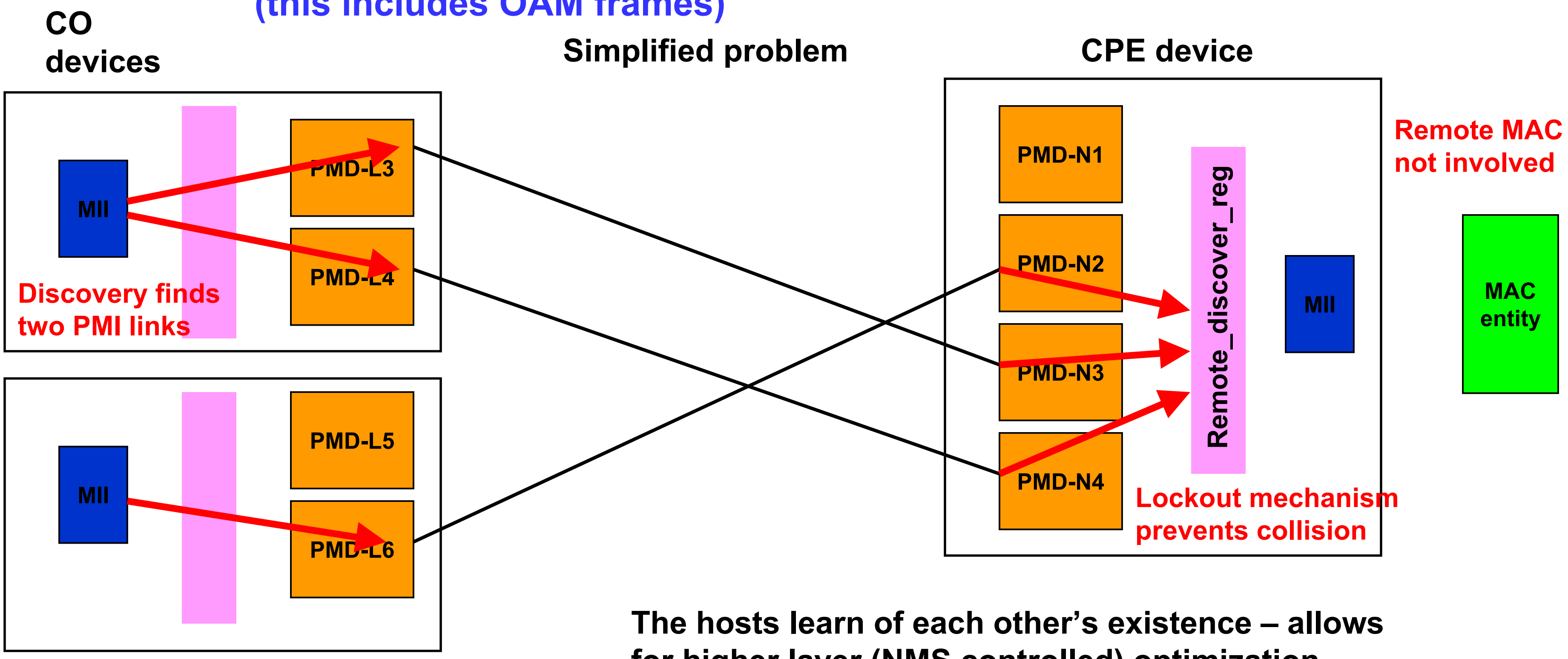
The aggregation mechanism would need to be much more complex to allow multiple hosts to contact the MAC entity through the MII

PHY level discovery

- Use of the remote_discovery_register simplifies everything

  Remote MAC is oblivious (➔ can re-use existing silicon)

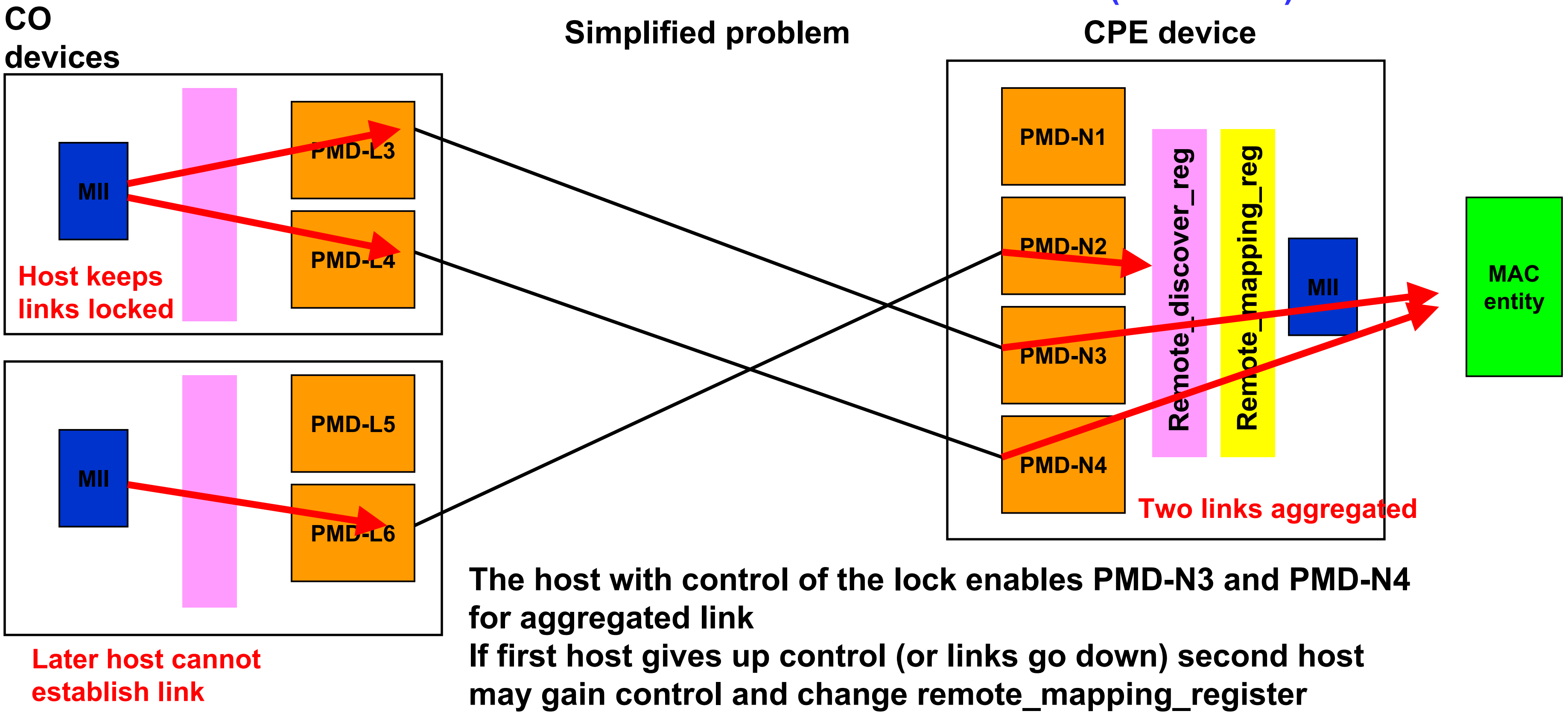
  (this includes OAM frames)



The hosts learn of each other’s existence – allows for higher layer (NMS controlled) optimization.

After discovery

- The first host to discover the NT keeps it locked – prevents collisions/confusion
  - Later hosts will see locked register contents (other entity may optimize)
  - Host releases lock itself or NT releases lock after linkdown (+ timeout)



The host with control of the lock enables PMD-N3 and PMD-N4 for aggregated link

If first host gives up control (or links go down) second host may gain control and change remote_mapping_register

Self-clear / timeout

- If a host is unexpectedly lost (e.g. links go down)…
  - NT must clear remote_discovery_register
  - Otherwise redundant links won’t come up
- Timeout must be based on PMI-MII mapping
  - Already set by controlling host
  - When links corresponding to all enabled PMI’s are down – timeout…
  - … then clear remote_discovery_register
- Why not immediate clear?
  - Must allow for “micro-interruptions” – no unnecessary reconfigure
  - After first discovery, PMI enables must be set (allow 30 seconds)
  - Fast changeover may be achieved by clearing registers directly

Altogether…

- New register “remote_discovery_register
  - 64 (or 48) bit, write/readable from remote, one per MII
  - Atomic “set-if-clear” operation:
    - writes only if currently zero, returns contents with ACK/NACK
- Multiple MII, NT must set MII to PMI mapping
  - Before enabling links (hard, soft, firm – whatever!)
- LT host will set PMI enables for aggregation
  - Gated with link state to control Tx fragmentation
- NT auto-clear remote_discovery_register when…
  - All links corresponding to PMI enables are down and…
  - Wait 30 second timeout
- All the rest should be out of scope…

Cisco Systems
®
Empowering the
Internet Generation℠

Definitions and assumptions

- Define “aggregateable PHY”

  Physical layer interface for multiple PMI’s which supports aggregation across those PMI’s. This may be one or more devices. There may be multiple separate “aggregateable PHYs” within one high density device.

- Define “dumb CPE”

  A CPE device which does not have any capability (or intention) to control any part of the physical layer connection. Such a device may not have a processor capable of communicating across the MDIO interface. It may not be programmed with a MAC address or be capable of responding to MAC control frames.

- Assume aggregation control as described by Klaus

  At least within each “aggregateable device”

- Assume host (LT) is smart, CPE may be dumb

  Mechanism must work in the presence of this asymmetry

- Assume only default connectivity with remote PHY devices

  At least one PMI will be connected to an MII, no more can be assumed