Power-Saving Mechanism for EPoC

Marek Hajduczenia, PhD

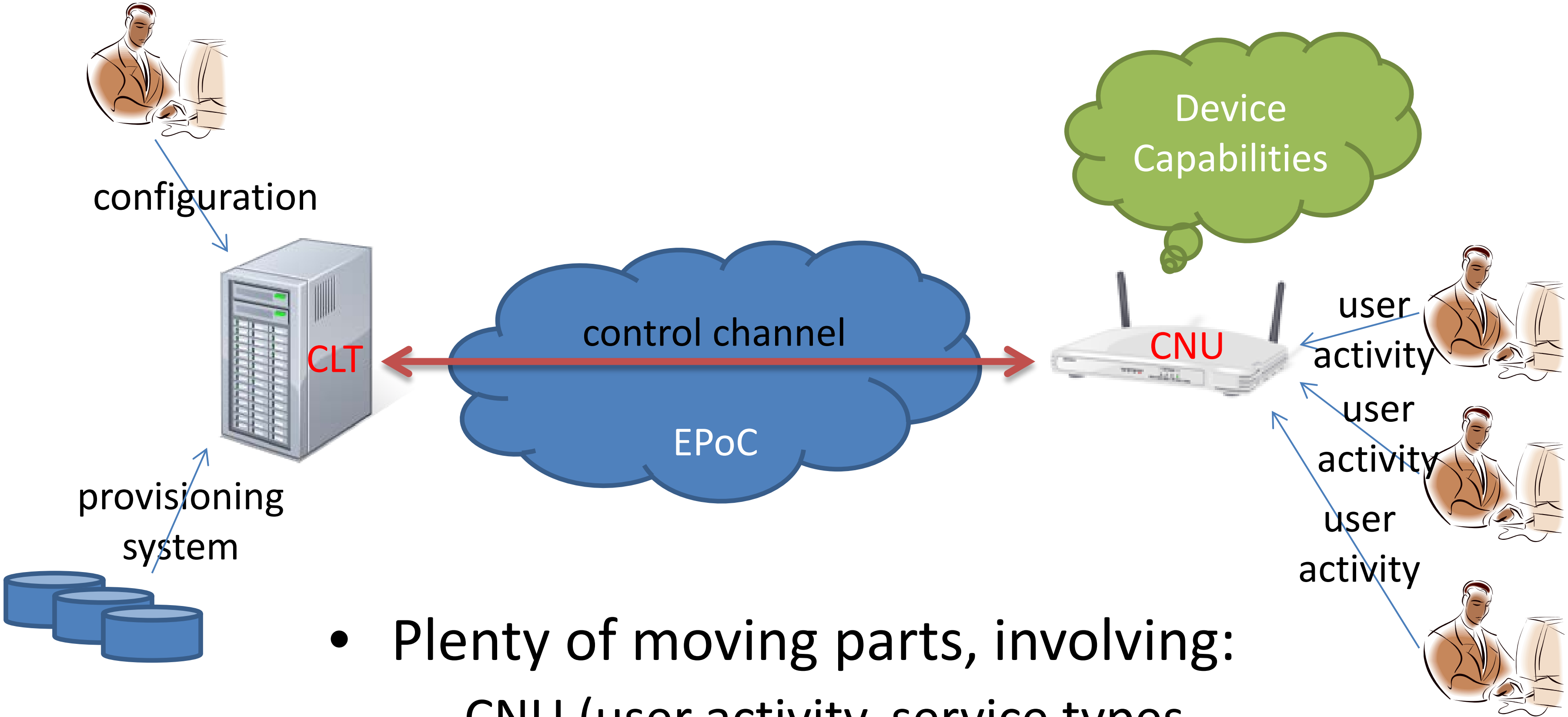
ZTE Corporation

marek.hajduczenia@zte.pt

Summary

- IEEE Std 1904.1 (SIEPON) defines three power-saving mechanisms, addressing specific operational requirements of the given profile
- Each power-saving mechanism provides detailed definitions, state diagrams, and clear description of operation of individual devices
- This presentation suggests a path forward, building on SIEPON-defined mechanism(s), and indicating what needs to be done in EPoC to support power-saving operation

High-Level View of Power-Saving



- Plenty of moving parts, involving:
  - CNU (user activity, service types, device capabilities, etc.),
  - CLT (and its capabilities),
  - operator configuration, and
  - provisioning systems …

What we need …

1. A mechanism to configure power-saving parameters on CLT and CNU (provisioning system)
2. A control channel between the CLT and the ONU
   - for CLT to control the CNU (sleep / active) state
   - to assure QoS measures are still met
3. A way to discover CNU power-saving capabilities
   - including supported sleep modes, power on/off times …
4. Support for sleep modes in hardware
   - to maximize power-saving potential and guide power-efficient implementations
5. Concise definition of interaction between CLT and CNU when entering and leaving the sleep mode in an organized manner (AKA *power-saving mechanism*)

What we have today …

- Items 1 and 3 from the previous slide are already available to us
  - See the summary slides on SIEPON power-saving mechanisms (hajduczenia_01_0313.pdf)
  - Also, see Clause 10 of SIEPON in our private directory for more details
- These two items guarantee:
  - all necessary definitions for control channel between the CLT and CNU,
  - ways to discover CNU capabilities and react to them
- SIEPON provides us with three distinct ways to handle item 5
  - Do we really need to come up with a new way to do things in EPoC?

CLT – CNU Control Channel

- SIEPON power-saving control mechanisms are
  - based on exchange of extended OAM or MAC Control messages
  - not tightly coupled with the PHY layer and allow greater implementation flexibility
- For EPoC …
  - alternative proposals were presented, suggesting to use PLC to signal the CNU to enter / leave the sleep state
- At some point of time, we will need to take decision on the use of specific control channel

Control Channel – What to Choose ?

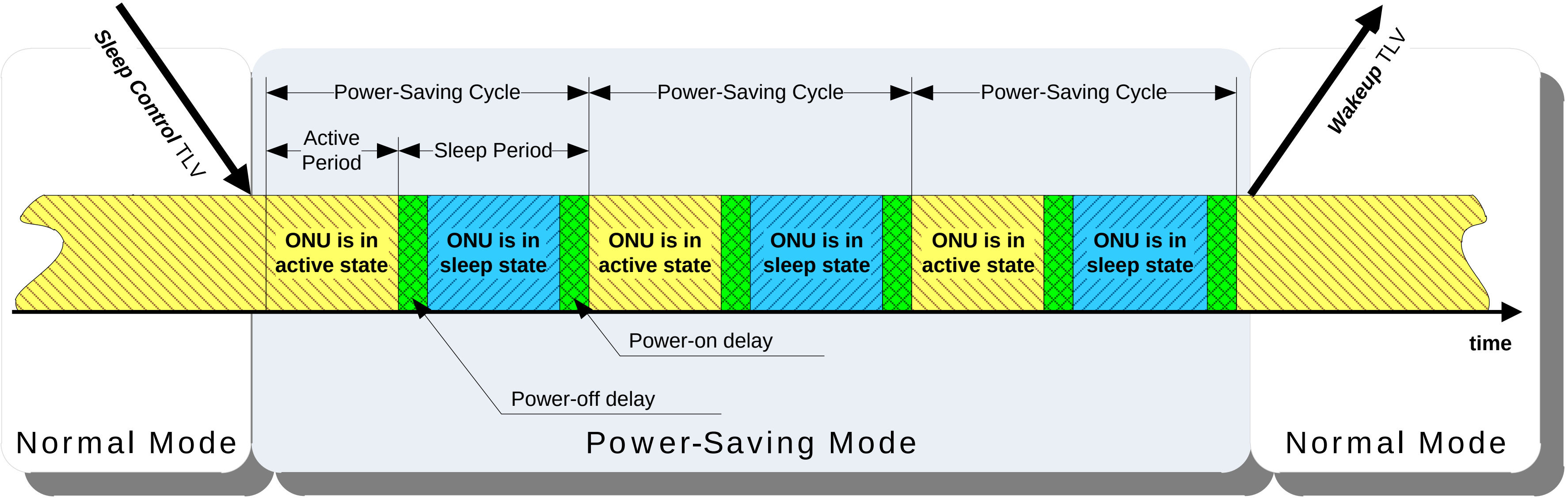
- PLC control channel:
  - Capability to exchange power-saving signalling between peer PCS instances
  - Limited capability to cooperate with scheduler and MPMC layers (XGMII is in the way, all data needs to be pumped through MDIO)
  - Limited bandwidth may become a problem when controlling status of a larger number of CNUs
  - Synchronization of sleep period with scheduler is much more complex and requires quasi-real time information exchange across MDIO. Same applies to synchronized sleep for multiple stations

Control Channel – What to Choose ?

- SIEPON-like control channel:
  - Relies on existing control channels
    - OAM / MAC Control, depending on the selected package
  - Allows for coordination with scheduling operations for upstream and simpler synchronized sleep for multiple subtended customer devices
  - Covers power-saving operation of PCS elements, MAC Control and MAC Clients
  - Allows for implementation flexibility while assuring interoperability in multi-vendor environment
  - Bandwidth is not a problem
    - OAM is not limited to 10 frames/second per link any more – it is operator configurable now
    - MAC Control does not have any frame rate limitations at all

Hooks in Hardware

- To optimize power-saving gains:
  - – EPoC PHY must be ready to deal with switching between active state and sleep state
  - – Power on and power off delay should be minimized, and time the CNU stays in sleep state must be maximized



PHY Features

- To put receiver to sleep:
  - CNU must synchronize quickly to received (downstream) data stream to reduce power on delay
  - Fast re-sync mechanism is needed to avoid searching through all possible subcarrier positions every time CNU goes to sleep
- To put transmitter to sleep:
  - Really, no special new requirements are needed, apart from a short power on time
  - CNU will operate in burst mode in upstream anyway in either TDD or FDD modes.

Higher Layer Features

- To improve power efficiency, elements other than Tx/Rx can be also put to sleep
  - (power-hungry) FEC is one of typical candidates
  - Selected MAC Clients and associated packet queues not receiving / sending data could be also powered down
  - Transmit / receive direction on XGMII could be powered down as well when not used
- With SIEPON-like control mechanism, it is also possible to signal higher layer functions (application) about PHY entering sleep mode

What do we need to do in EPoC?

- Two scenarios are possible going forward
  - Both are examined in the following slides
- The presented scenarios differ primarily in the amount of work for EPoC TF and amount of reuse from existing specs
- Since EPoC is intended to look like EPON-on-coax, reuse of EPON-like power-saving mechanisms is naturally recommended to speed-up EPoC development

Scenario 1 – Less Work Involved

- Reference power saving mechanism(s) defined in SIEPON for EPON
- SIEPON-defined management entities are generic enough to be reused by EPoC as well
  - To be confirmed by TF
- In EPoC, specify:
  - Fast re-sync for CNU Rx to minimize power on delay and data re-synchronization process
  - Consider extending the duration of keep-alive period for MPCP to maximize duration between active periods for CNU

Scenario 2 – Contain work in EPoC

- Define the whole power-saving mechanism within EPoC, limiting its applicability to 802.3 layers only (EEE-like mechanism)
- Define operation of the power-saving control channel over PLC, including associated SDs
- Capability / Control registers and new managed objects (in cooperation with 802.3.1) will have to be defined
- Fast re-sync mechanism (see previous slide) should be still specified in EPoC to optimize power-saving capabilities of our PHY

Summary and Suggestions

- EPoC project has already plenty of challenge areas to work on (see lengthy emails …)
- Power-saving mechanism through the reuse of SIEPON-defined control channel and management entities can be added to EPoC PHY with minimum effort, with very limited limited new functions to PHY
- Cooperation with SIEPON will be needed to examine whether any changes are required to SIEPON power-saving definitions to apply them to EPoC
  - The time is right *now* given advanced status of SIEPON
- Power-saving functions in PHY will be proposed at the later date, when more details on PHY operation are available.

Straw Poll

- For power-saving mechanism in EPoC, I prefer to take the following approach:
  - __1_ Approach 1 (slide 13)
  - __0_ Approach 2 (slide 14)
  - __9_ Need more time to study both proposals
  - __0_ I am too confused about what you’re really asking – need more time to digest the slides