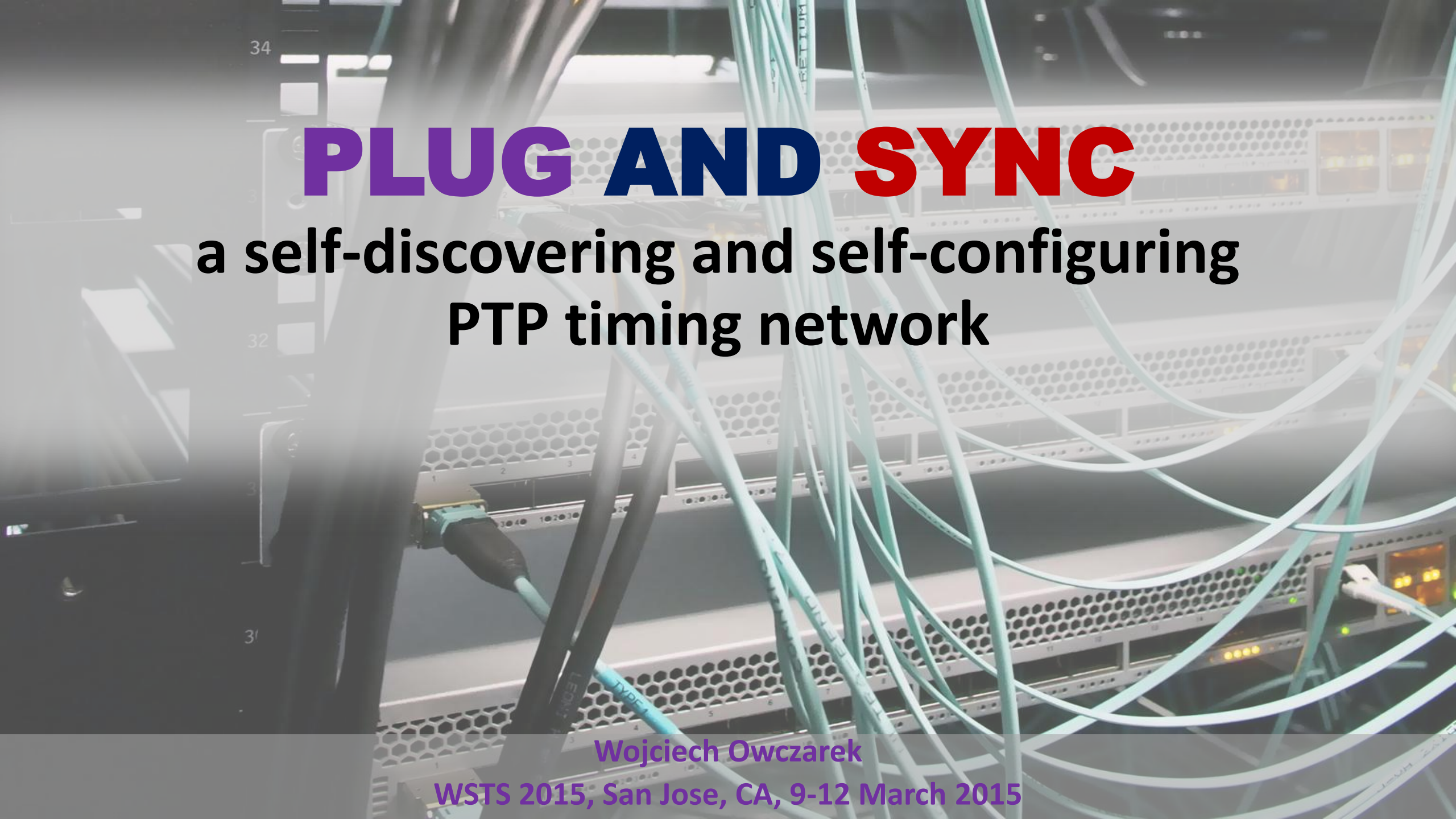


PLUG AND SYNC

a self-discovering and self-configuring PTP timing network

Wojciech Owczarek

WSTS 2015, San Jose, CA, 9-12 March 2015

Background: the need for easier time sync

- **Here be dragons:**
- Many engineers still lack basic understanding
  - Why? Most successfully run NTP
- Most common transport: simple UDP
  - Still many engineers treat it as magic
- Should running PTP require time & frequency knowledge?

Background: the need for easier time sync

- **Common error conditions not easily identified:**
- Default behaviour: sync with whatever we're getting
  - masks issues
- Domain mismatch: only packet trace or error counters (if implemented)
- One-way sync (no delay_resp): both state machine and user oblivious
- Receiving Announce but no Sync: still in SLAVE state

- **Interop challenges reach further than IEEE 1588 compliance**
- **Things can get messy when each port can be master or slave**

PTP everywhere: enterprise and more

Use cases for easier sync:

- Initial use case: electronic trading
- Entertainment systems, wireless audio
- Time as a service
- IoT, sensor networks

- some of those use UDP/IPv4/multicast, some do not

Configuring PTP

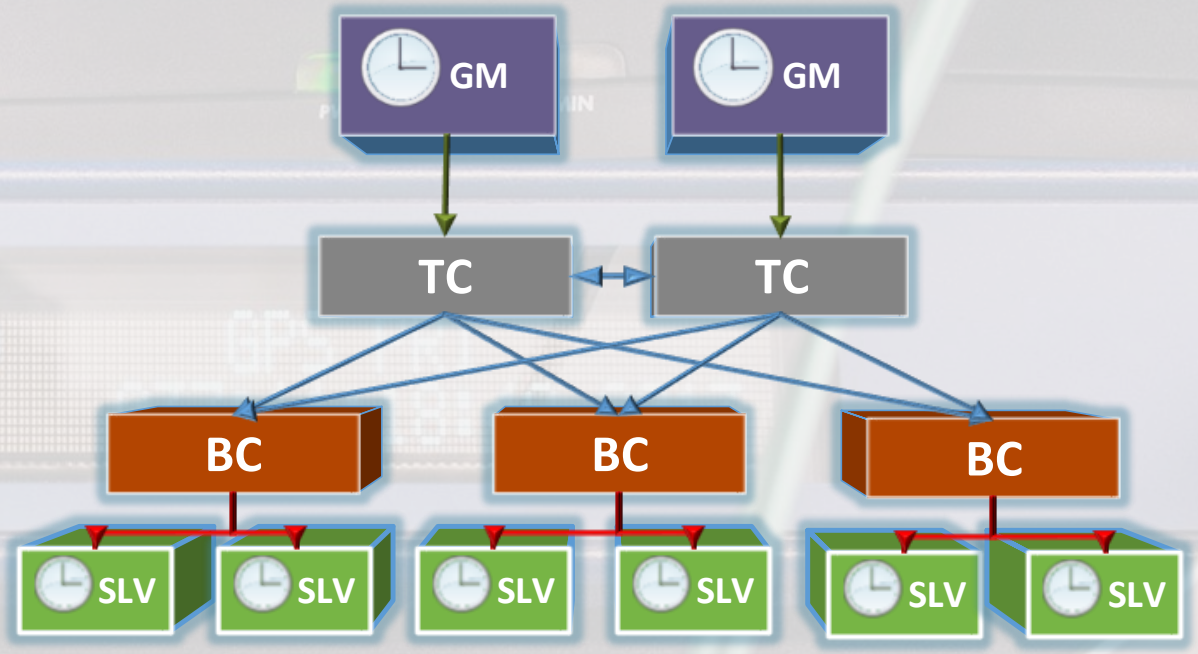
- Domain number
- One-step / two-step
- Host nodes: slave only?
- Profile selection or capability selection?
  - Multicast / unicast / mixed
  - Signalling / no signalling
  - Two implementations of the same profile may not interoperate
- P2P/E2E
- Message rates

AutoPTP: goals

- Sync across whole network
  - hardware permitting – out of scope
- Deterministic (or fixed) topology
  - Sync from North, provide sync to South
  - Sync East-West only when North fails
- Inherent security
  - Eliminate the need for ACLs when facing customers
  - Prevent rogue masters from taking over
- Minimal configuration effort
- Full interoperability with non-AutoPTP devices
- More than what BMCA itself can provide

Deterministic PTP time domain topology

- Avoiding surprises
  - Rogue masters, topology flip, cold start
- GMs only have “downlink” ports
- Hosts only have “uplink” ports
- BCs and TCs have both
- When no GM, BCs can syntonise
- Do we expect GMs on host ports?
  - No. MasterOnly / NotSlave
- Do we expect slaves on uplink ports?
  - No. SlaveOnly / NotMaster



AutoPTP: Configuration steps – 1 2 3

1. GMs:
- Domain number, priorities
- One-step / two-step, message rates

2. TCs and BCs:
- Select which are TCs and which BCs if possible (Default to BC?)
- Enable PTP on selected links, configure link types (optional – otherwise autoconfigure)

3. Slaves:
- Nothing: just enable AutoPTP

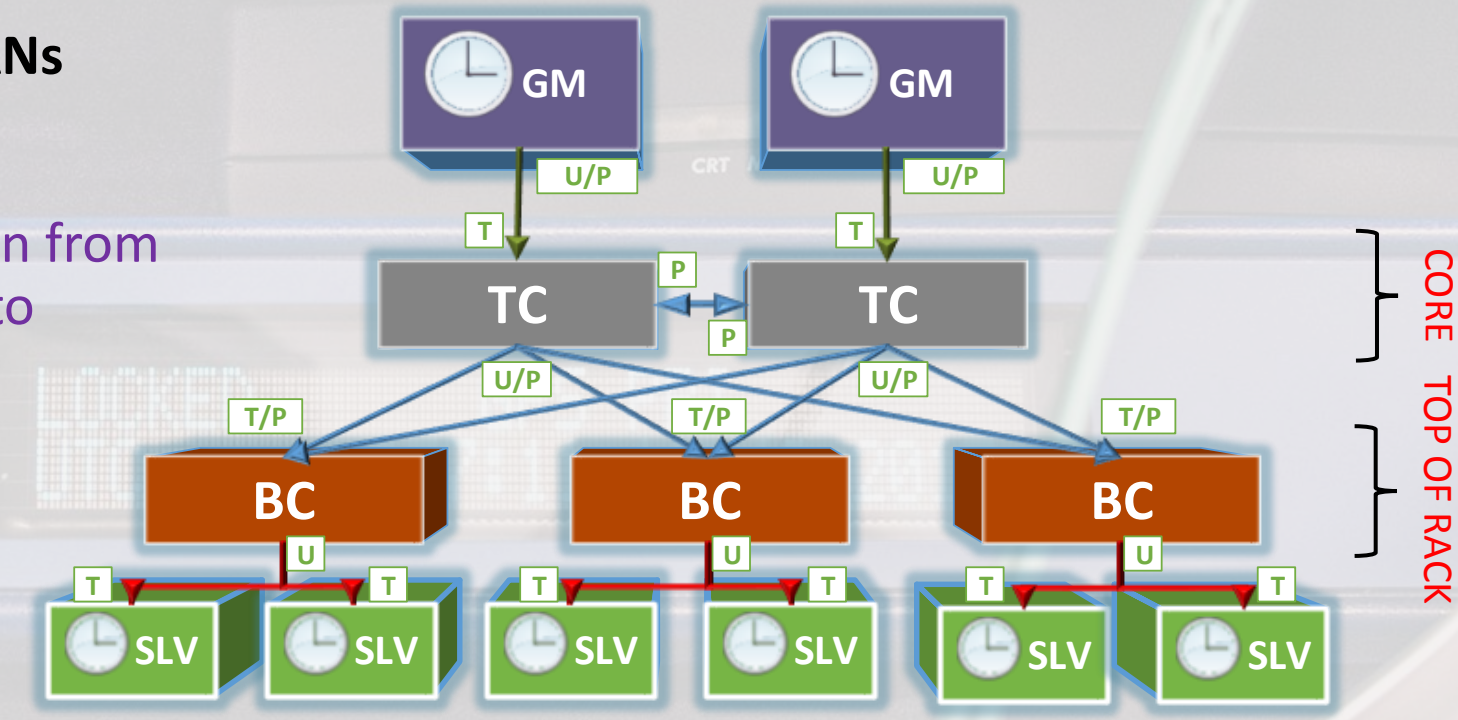
AutoPTP: link types

Similar to DHCP snooping and PVLANs

1. Trusted (uplink):
- Accept announce and configuration from
- Never send or forward announce to

2. [Optional] Promiscuous
- *BC and GM send announce to*
- *BC accepts announce from*
- *TC forwards announce between*
- *Help syntonise when top GMs fail*

3. Untrusted (downlink):
- Always ignore announce from
- Send announce to



- Slave-only clocks: all links trusted unless disabled
- GMs ant TCs: choice of promisc or untrusted downlinks
- BCs: Choice of promisc or trusted on uplinks
- **Topology loops out of scope**

AutoPTP: discovery mechanism

Advertise to adjacent port:

- Clock type (GM, BC, Slave), topology info: has-slaves, has-masters
- Capabilities (Profile, One-step / Two-step, more?)
- GMs and BCs advertise:
  - Domain number, min and max message rates, unicast IPs
  - Required capabilities: one-step/two-step, E2E/P2P
  - Topology summary: has-slaves, has-GMs, is-slave-only, is-GM
- TC:
  - forwards advertisement from trusted links
  - and/or generates advertisements (because they can have unique capabilities)

Choice of protocol:

- LLDP
  + Not forwarded
  + Could be used for discovery only
  - Different protocol
- PTP – management messages with new TLVs (new TLV type: INFO)
  + Native PTP mechanism
  - Should not leave the point to point links (use layer 2? Use link-local multicast MAC?)

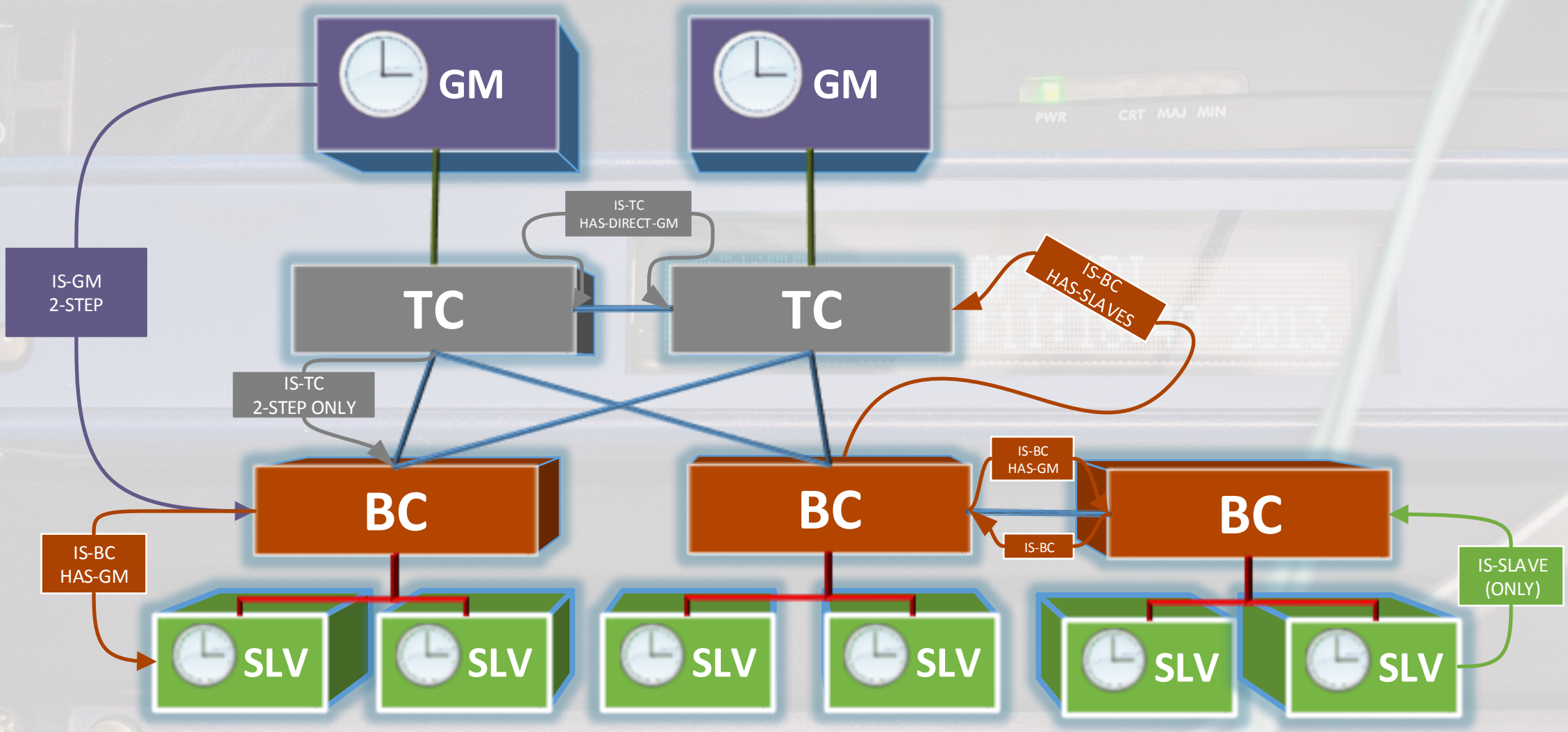
AutoPTP: autoconfiguration

- Starts with GMs on trusted ports, cascades down
- BC or slave accepts from trusted port:
  - Domain number, delay mechanism, message rates
  - Other? Unicast IPs? Profile?
- Dynamic BC priorities on uplinks:
  - $Priority1_{BC} = Priority1_{parent} + 1 + stepsRemoved$
  - $Priority2_{bc} = Priority2_{parent} + 1 + has\text{-}slaves - has\text{-}direct\text{-}masters$

  } Example only!
- Link types could also be autoconfigured:
  - Add authentication to discovery – or use one outside PTP (different credentials = different topologies)
  - Link to is-slave or has-slaves = untrusted
  - Link to is-master or has-direct-masters = trusted

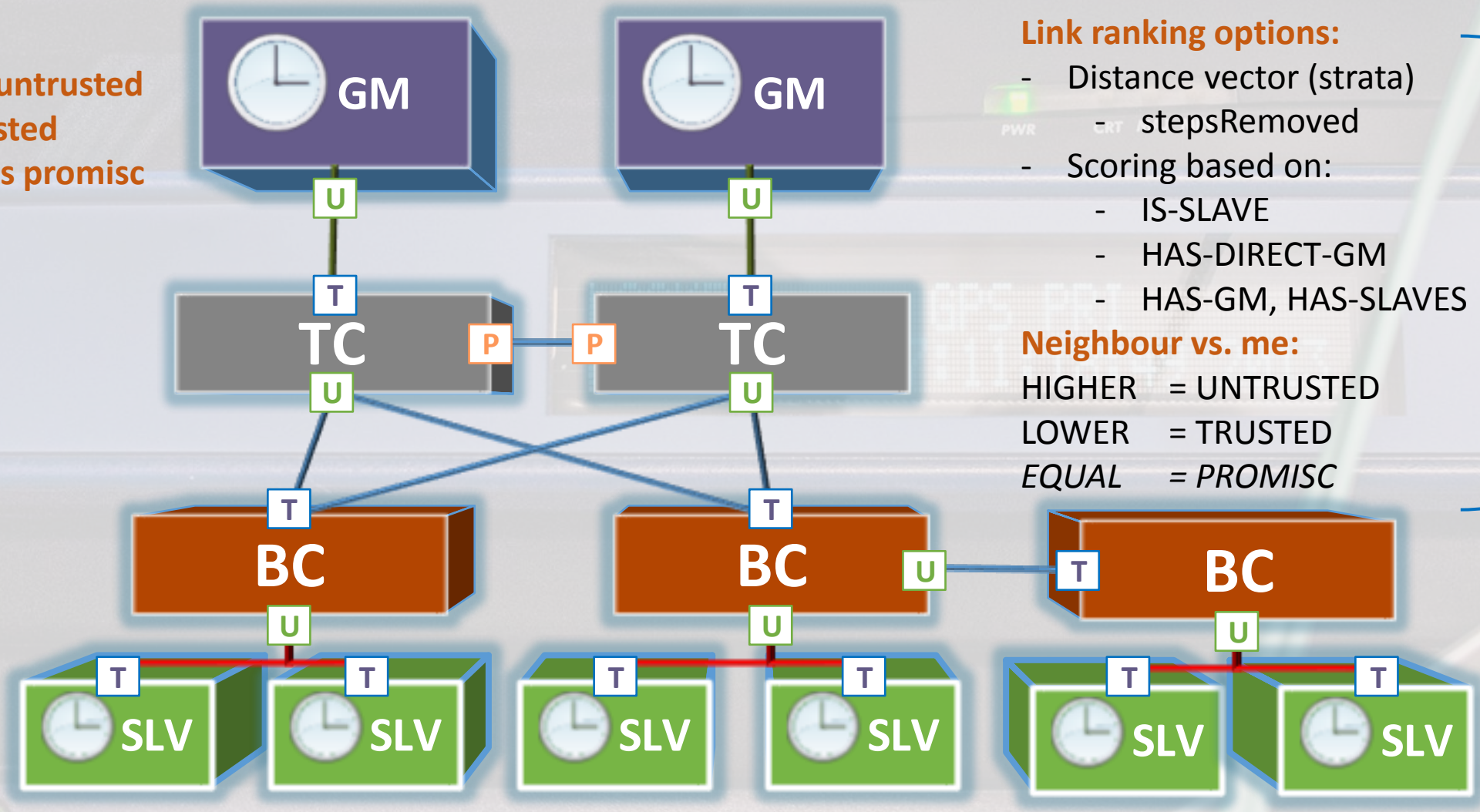
  } Evaluation algorithm – example only!

AutoPTP: Example interaction… discovery



AutoPTP: Example interaction… link setup

Manual setup:
- Downlinks untrusted
- Uplinks trusted
- Backup links promisc



Link ranking options:
- Distance vector (strata)
  - stepsRemoved
- Scoring based on:
  - IS-SLAVE
  - HAS-DIRECT-GM
  - HAS-GM, HAS-SLAVES

Neighbour vs. me:

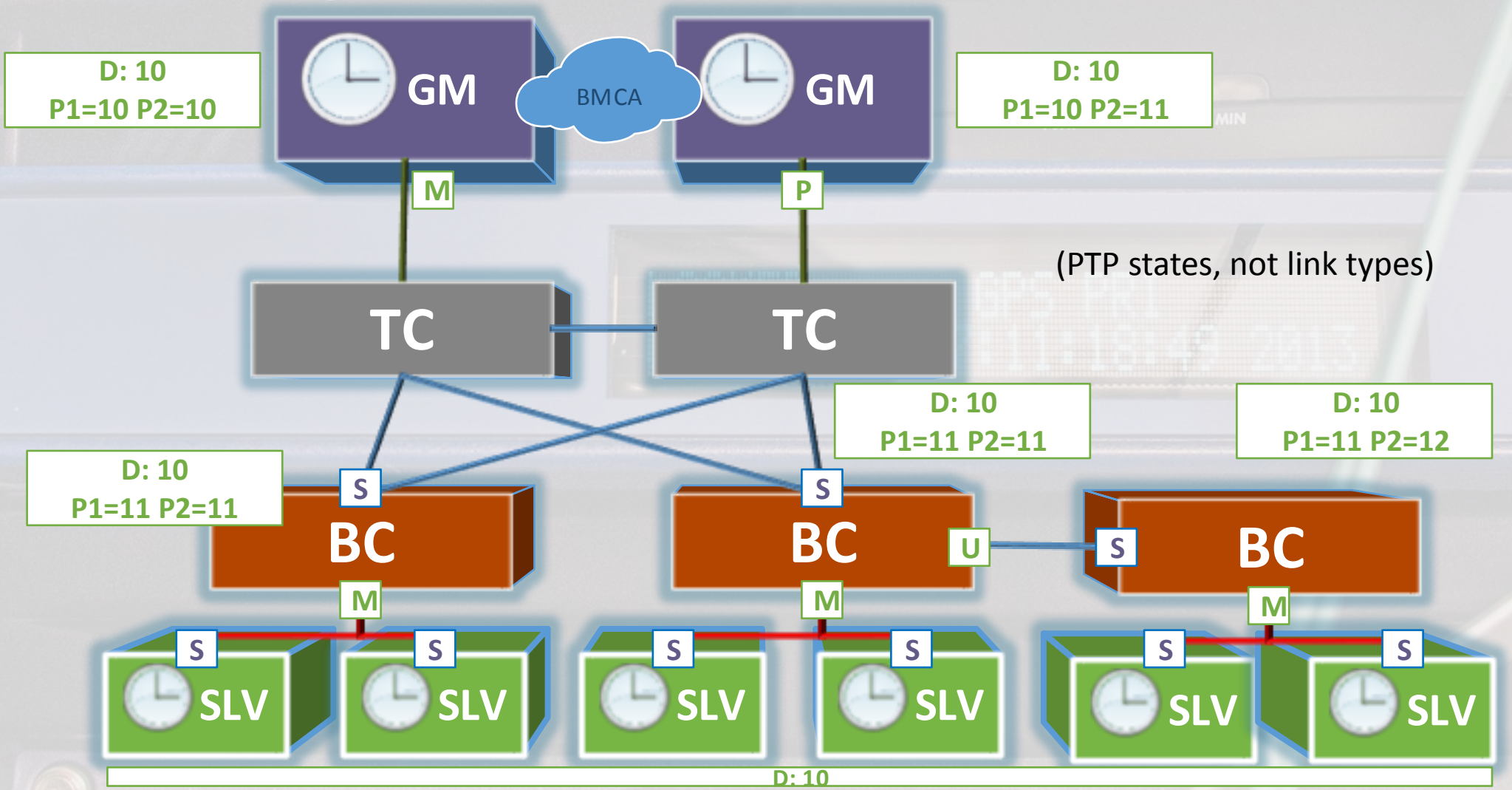
HIGHER = UNTRUSTED

LOWER = TRUSTED

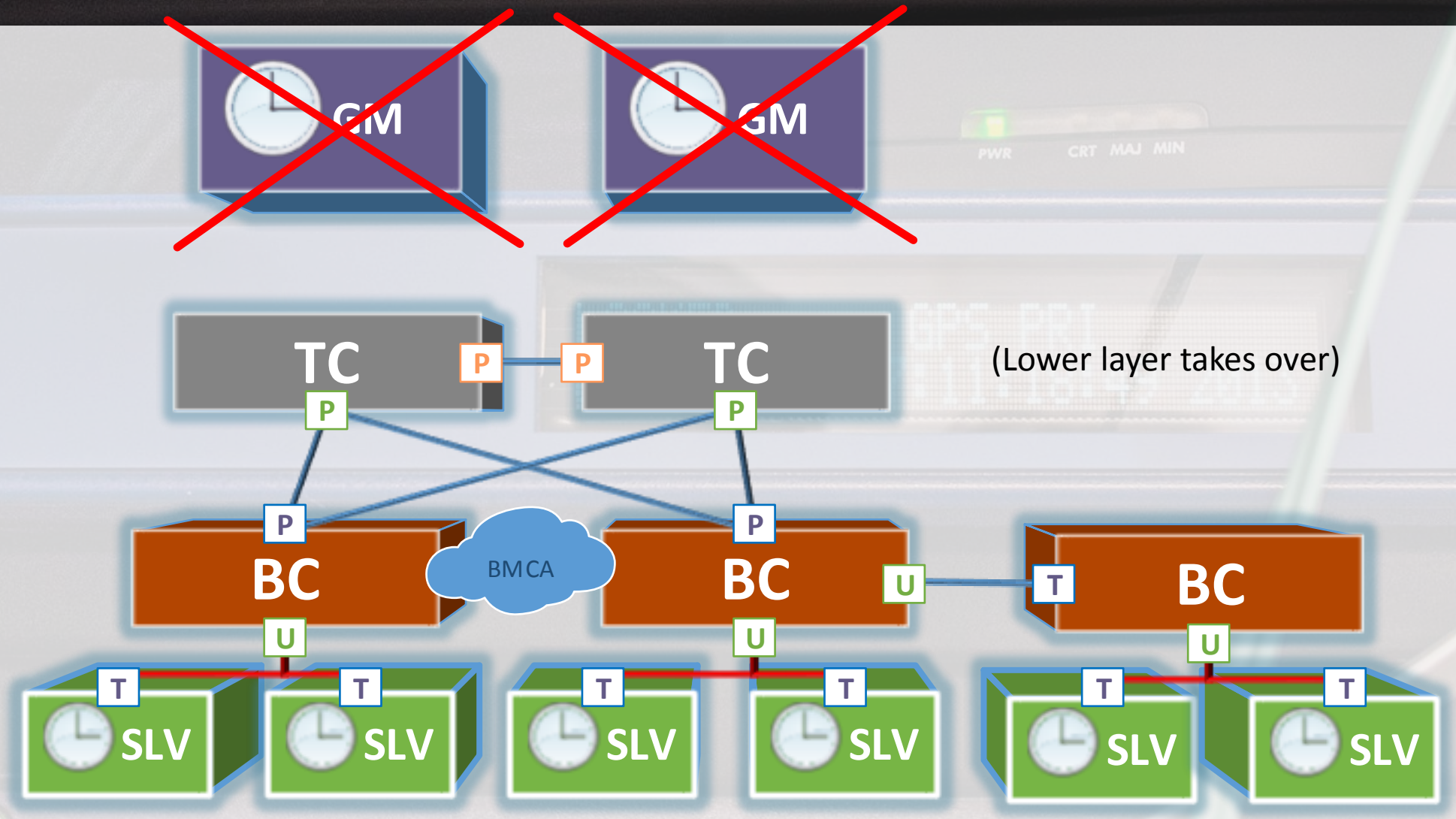
EQUAL = PROMISC

Automatic setup

AutoPTP: Example interaction… autoconfiguration



AutoPTP: Example interaction… promisc – GMs fail



AutoPTP: Implementation options

- Modify the FSM and / or BMCA?
- Discovery during INITIALIZING or LISTENING?
- What to do if we cannot support offered capabilities:
  - **Go into FAULTY to flag this – preferred: instant fault-finder**
  - **Ignore advertisements?**
- Discovery protocol?
- Promiscuous links or maybe community links?
- Alternative 1: No link types: discovery only – LLDP?
  - **Self-configuration optional: would still make deploying PTP easier**
- Alternative 2: No link type discovery – fixed link types only
  - **Would still make deploying PTP easier – and more secure**

AutoPTP: Potential issues

- Which domain to accept when different ones seen? Lower?
- Add domain number to BMCA evaluation?
- Become domain-agnostic?
- Fallback mechanism when discovery fails?

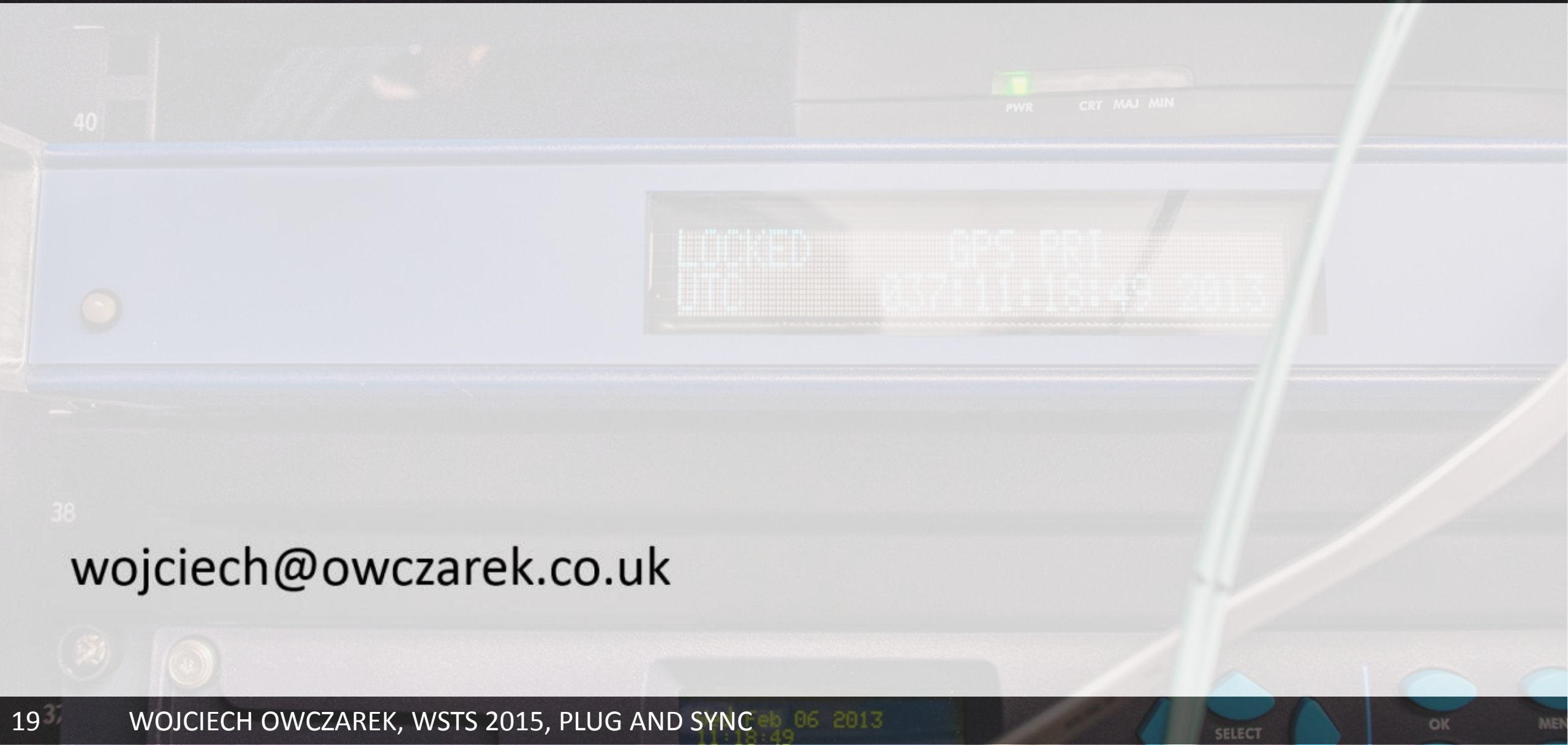
- Interoperability:
  - What to do with PTP traffic from “foreign” domains?
  - Integrating unaware implementations
  - What to do before discovery?

AutoPTP: future work and open questions

- Is it worth it? Re-inventing BMCA?
- RFC material? Maybe too broad for IEEE 1588
- Extend beyond PTP? NTP, SyncE?
- Do some vendors already do this or something similar?

- **Open to PoC work and standardisation**

Questions? Suggestions?



wojciech@owczarek.co.uk