

Revised Definition of The GMPLS Switching Capability and Type Fields

[draft-berger-ccamp-swcaps-update-00](#)

Lou Berger

lberger@labn.net

Julien Meuric

julien.meuric@orange.com

Background

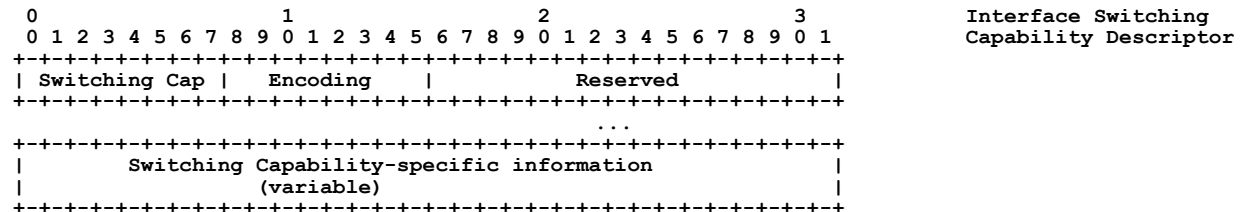
- GMPLS supports
 1. Multiple switching types (e.g., packets and TDM)
 2. Multiple technologies within each type (e.g., SDH and OTN)
 3. Multiple levels of switching/multiplexing within a technology (e.g., PSC1-N, ODUs, Ethernet/PBB)
- Representation of above is a bit haphazard
 - Common *Switching Type* values used in signaling and routing
 - Routing carries values in the *Switching Capability* (or *Switching Cap*) field
 - See <http://www.iana.org/assignments/gmpls-sig-parameters/gmpls-sig-parameters.xml#gmpls-sig-parameters-3>
 - Always represents 1, sometimes 2 and 3
 - 2 sometimes must be inferred
 - 3 has multiple solutions
 - PSC/MPLS signals via hierarchy and separate routing instances
 - Note PSC-2 → N are **not** used!
 - SDH / OTN (pre v3) signals via label+traffic parameters, and **lacks** standardized representation of technology-specific routing information

Value	Name
1	Packet-Switch Capable-1 (PSC-1)
2	Packet-Switch Capable-2 (PSC-2)
3	Packet-Switch Capable-3 (PSC-3)
4	Packet-Switch Capable-4 (PSC-4)
30	Ethernet Virtual Private Line (EVPL)
40	802_1 PBB-TE
51	Layer-2 Switch Capable (L2SC)
100	Time-Division-Multiplex Capable (TDM)
125	Data Channel Switching Capable (DCSC)
150	Lambda-Switch Capable (LSC)
200	Fiber-Switch Capable (FSC)

Background (Continued)

- The *Switching Capability* (or *Switching Cap*) field serves an additional purpose in routing

- It also indicates ISCD SCSI field format



- When technology is inferred, SCSI format must also be inferred
- Current discussion triggered by OTNv3 SCSI format discussions
 - Reminder: Current SDH or OTN RFCs do not include SCSI formats

Objective of Draft

- Address two issues for future definitions:
(i.e., no impact to existing RFC-based implementations)
 - A. Current representation of (1) switching types, (2) technologies, and (3) multiplexing is inconsistent
 - B. Current definition of Switching Capability (and types) is overloaded
- Alternatives considered – two extremes:
 - Assign Switching Type per *potential* SCSI format
 - i.e., different switching type per switching technology
 - Switching Type represents (1), (2) and (3)
 - i.e., type per technology & multiplexing level (ala PSC-N)



Selected Approach

Main Proposal

- Simplify definition of *Switching Capability*
 - Only indicate switching technology
 - Remove overload – no intra-technology significance
 - Remove SCSI format ambiguity
 - Different SCSI formats MUST use different values
- Deprecate unused PSC values
 - PSC 2→4
- Keep values used in routing and signaling aligned
 - No change: keep using IANA Switching Type registry
- No substantive change for signaling
- *Matches discussion on list*
 - Aligned with current OTN drafts
 - Any comments?

Open Question

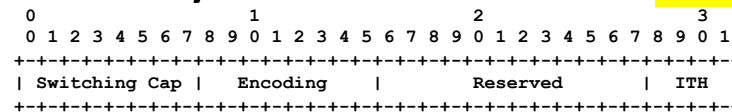
- Should there be a *Generalized* indicator in routing of intra-technology hierarchy/multiplexing?
 - Some reminders:
 - this draft only applies to future CCAMP work, not current (OTN) RFCs or drafts
 - the purpose of GMPLS (and CCAMP) is to define common control plane mechanisms for different technologies
 - multiple technologies support intra-technology hierarchy/multiplexing
- Authors' conclusion is: "yes"
- We propose a specific solution, but want WG input on objective before focusing on specifics.
 - Any comments?

Specific Proposal

- Introduce *Intra-Technology Hierarchy* field into ISCD

– Note: planned name change ILH → ITH

– Using 4 previously reserved bits



Interface Switching
Capability Descriptor

- 0 = ignore field
- Other values are to have technology-specific values
- Some open questions:
 - Is there any impact on MLN?
 - We think no
 - Are there crankback implications?
 - If need SwCap-based XRO, yes
 - Is same for per-technology specific (SCSI-based) solutions
 - If only label-based XRO, no

Next steps

- Solicit feedback
- Update draft based on discussion & comments
- Progress document