MPLS Architectural Principles

George Swallow

(swallow@cisco.com)

MPLS Guiding Principles

- Co-existence with IP
- IP based Control Planes
- Reuse of IP protocols
- Extendable protocols
- Pragmatism

MPLS Architectural Principles

- Separation of Control Plane and Data Plane
- Label semantics (except reserved) belong to application
    - Forwarding Equivalence Class
- Compact Header
- Multiple, Independent Control Planes
- Label Stack

Some contextual background

- Ratio of processor speeds to data-link speed much lower
- Cost of a bit orders of magnitude higher
- Getting links was very slow
- IP community did not like ATM and particularly the “cell tax”

Dataplane Requirements

- Very fast (i.e few gates and/or few cycles per packet)
- Very wide range of equipment
- Simple Data Plane
- Stable Data Plane

  Needs to work for lifetime of equipment (>10 yrs)

  Including boxes for which

  - New linecards cannot be cost justified
  - Software has become “functionally stable”

MPLS Components

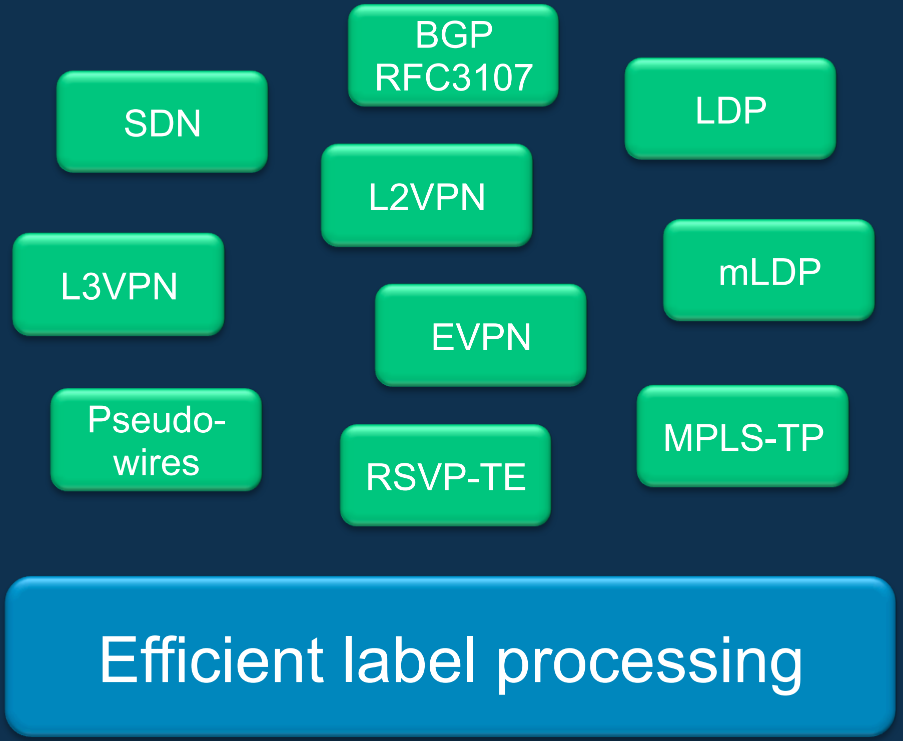
- Control components

  Applications directly define and manipulate label bindings

  Essentially they use labels to program data plane

- Forwarding component

  Simple label-swapping paradigm



Simple Forwarding Plane

Label Stack Entry

- Label determines simple forwarding operations

  Push, Pop, Swap, Outgoing interface(s)

- Class of Service + Label defines queuing, drop priority

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Label                  | COS |S|       TTL     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

COS = Class of Service

S = Stack Bit

TTL = Time to Live

Label Semantics

- Defined by the application
- Need only be known by LSRs that Push/Swap/Pop that label
- Full semantics need only be known to the consumer of the label
- Forwarding Equivalence Class
  - Simple powerful concept
  - Single label assigned to a set of packets that are to be treated identically
    - Note that CoS bits are not part of this definition

Label Space

- One Million Labels
- 16 Reserved Labels

  Intentionally a limited number

  New values impact dataplane

- Label Spaces per
  - LSR
  - Interface

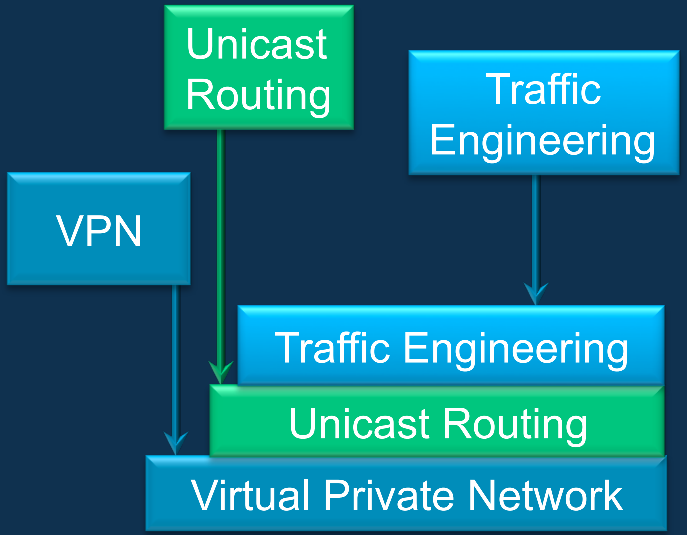
Context Labels

- One label can define a new label space
- Powerful means of expanding the available labels
- If you need more labels, this is the proper way to do it!

Label Stack

- Arbitrary number of labels
- Multiple control planes act on packets
- Control planes are totally independent
- Control plane at one level can be replaced

  No impact on levels above or below



The single most important element in MPLS’ success

“No idea is so bad that it won’t be proposed over and over again”

Ideas proposed over and over

- Global Labels
- Source Labels
- Special encodings of labels
  - Special meanings to label bits
- Special semantics for double labels
  - PoP top label; process next label; push PoP’ed label back on
  - 8 byte label stack entry
- Special uses for CoS and / or TTL bits
- Multiple EOS bits set in the label stack

Other rejected / abandoned ideas

- Encoding labels into VLAN-IDs
- Encoding labels into the IPv6 flow label
- Label Stack Header
- Labels in PIM
- I’m sure I’ve missed some