A YANG Data Model for In-Situ OAM

draft-zhou-ippm-ioam-yang-02
Huawei: Tianran Zhou, Jim Guichard
Cisco: Frank Brockners, Srihari Raghavan
Changes Since Last Meeting

• Comments received from
  – Greg Mirsky and Reshad Rahman

• Make the 4 IOAM option profiles as features

• Modify the ACL leafref

• Add a “admin-config” container

• Modify the Security Considerations

• Modify the IANA Considerations
Overview

• Profiles
  – The IOAM model is organized as list of profiles.
  – Each profile associates with one flow and the corresponding IOAM information.
  – Multiple IOAM data types can be encapsulated into the same IOAM header.

```plaintext
module: ietf-ioam
  +--rw ioam
    +--rw ioam-profiles
      +--rw admin-config
        | +--rw enabled? boolean
      +--rw ioam-profile [profile-name]
        +--rw profile-name string
        +--rw filter
          | +--rw filter-type? ioam-filter-type
          | +--rw acl-name? -> /acl:acls/acl/name
        +--rw protocol-type? ioam-protocol-type
      +--rw incremental-tracing-profile {incremental-trace}?
        | ...
      +--rw preallocated-tracing-profile {preallocated-trace}?
        | ...
      +--rw pot-profile {proof-of-transit}?
        | ...
      +--rw e2e-profile {edge-to-edge}?
        ...
```
Preallocated Tracing Profile

- The preallocated tracing option will create pre-allocated space for each node to populate its information.

```
+--rw preallocated-tracing-profile
    +--rw enabled?    boolean
    +--rw node-action? ioam-node-action
    +--rw trace-type?  ioam-trace-types
    +--rw enable-loopback-mode?  boolean
```
The incremental tracing option contains a variable node data fields where each node allocates and pushes its node data immediately following the option header.

```
+--rw incremental-tracing-profile
   +--rw enabled?       boolean
   +--rw node-action?   ioam-node-action
   +--rw trace-type?    ioam-trace-types
   +--rw enable-loopback-mode? boolean
   +--rw max-length?    uint32
```
Proof of Transit Profile

- The IOAM Proof of Transit data is to support the path or service function chain verification use cases.
- It’s imported from “draft-brockners-proof-of-transit-04”

```plaintext
---rw pot-profile
  +---rw enabled?    boolean
  +---rw active-profile-index?  pot:profile-index-range
  +---rw pot-profile-list* [pot-profile-index]
    +---rw pot-profile-index  profile-index-range
  +---rw prime-number  uint64
  +---rw secret-share  uint64
  +---rw public-polynomial  uint64
  +---rw lpc  uint64
  +---rw validator?    boolean
  +---rw validator-key?  uint64
  +---rw bitmask?  uint64
```
**Edge to Edge Profile**

- The IOAM edge to edge option is to carry data that is added by the IOAM encapsulating node and interpreted by IOAM decapsulating node.

```
  +--rw e2e-profile
      +--rw enabled?  boolean
      +--rw node-action?  ioam-node-action
      +--rw e2e-type?  ioam-e2e-types
```
Next

• Comments?
• How about adopting this draft as the starting point for IOAM configurations?
Thanks