Application-aware Networking (APN) Solution Discussions

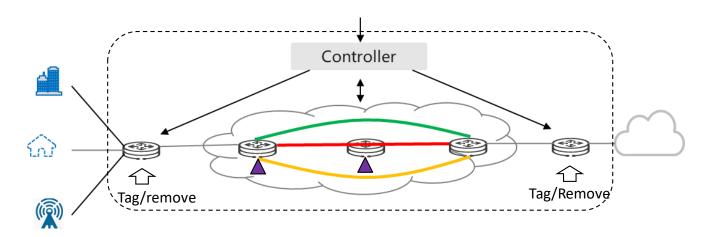
Shuping Peng/Zhenbin Li

Progress summary

- APN Side Meetings @IETF105 & IETF108
- APN Hackathons @IETF108 & IETF110
- APN Demos @INFOCOM2020 & 2021
- APN Mailing List: apn@ietf.org
- APN Wiki: https://datatracker.ietf.org/wg/apn/about/
- The use cases have been discussed extensively in previous IETF meetings
 - SD-WAN, FBB, MBB, etc.
- RTG people would like to start working on the solutions of APN.

Scope & Scenario of APN

- APN works within a service operator's network domain.
 - Typically, an APN domain is defined as a service provider's network domain where MPLS, SR/SRv6, VXLAN and other tunnel technologies are adopted.
 - APN attribute is tagged/removed at the edge of the network domain.
- APN is not about identifying a particular application or user within the network.
- APN is about telling the network what policies to apply to traffic.
- APN attribute is constructed based on the existing information such as 5-tuple presented in the packet header.
- According to the APN attribute, various policies can be flexibly applied to the traffic flow on various nodes along
 the network path, without the need of resolving the 5-tuple at every policy enforcement point in the network.

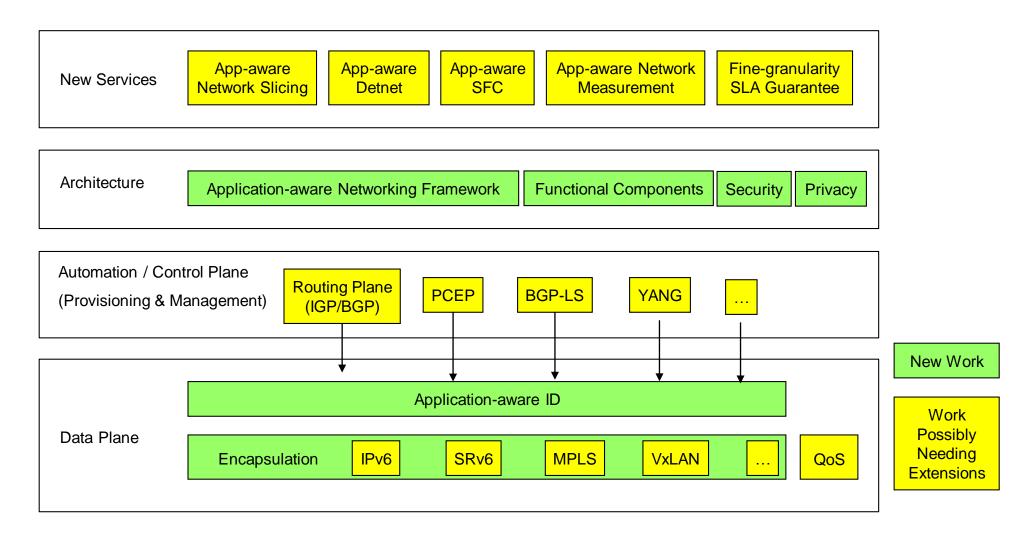


Updates

- Changes we have made to the drafts according to feedback received from the presentations @IETF110
 - Removed the application-side solution, only keep the network-side solution
 - The APN attribute is acquired based on the existing information in the packet header such as 5-tuple and QinQ (S-VLAN and C-VLAN) at the edge devices of the APN domain, added to the data packets along with the tunnel encapsulation.
 - · When the packets leave the APN domain, the attribute will be removed together with the tunnel encapsulation header.
 - APN aims to apply various policies in different nodes along a network path onto a traffic flow altogether, for example, at the headend to steer into corresponding path, at the midpoint to collect corresponding performance measurement data, and at the service function to execute particular policies.
- Drafts have been updated accordingly to reflect the presented contents and concepts
 - Framework
 - https://tools.ietf.org/html/draft-li-apn-framework
 - · Problem Statement
 - https://tools.ietf.org/html/draft-li-apn-problem-statement-usecases
 - Scope & Gap analysis
 - https://tools.ietf.org/html/draft-peng-apn-scope-gap-analysis
 - Security & Privacy consideration
 - https://tools.ietf.org/html/draft-peng-apn-security-privacy-consideration

Supporters would like to form a working group

The potential work items as below,



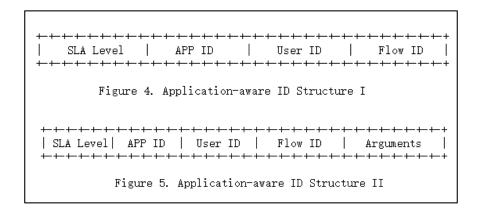
Solutions that need to look into

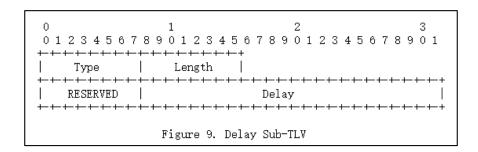
- 1. Design of the APN attribute
- 2. Encapsulation of the APN attribute on the various data planes
- 3. Delivery of the APN attribute through the control plane protocols
- 4. Management of the APN attribute via NETCONF/YANG

Design of the APN attribute

- Do we need a dedicated header for APN?
- What should be the structure of the APN attribute?
- Shall the APN attribute include
 - SLA (Mandatory/Optional?)
 - ✓ The SLA level of the service requirements
 - APP Group ID (Mandatory/Optional?)
 - ✓ The identifier of the application type/group
 - USER Group ID (Mandatory/Optional?)
 - The identifier of the user type/group
 - Session/FLOW ID (Mandatory/Optional?)
 - The identifier of the key session/flow of the traffic flow
 - Requirement parameters?
 - ✓ Bandwidth, Latency, Jitter, Packet loss?
- Total Length
 - Flexible/Fixed?
- Length of each field
 - Flexible/Fixed?

Example of the APN attribute structure



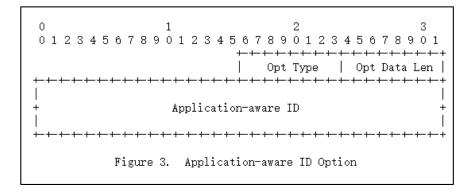


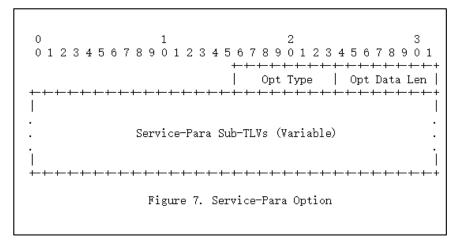
https://tools.ietf.org/html/draft-li-6man-app-aware-ipv6-network-03

Encapsulation of the APN attribute on the various data planes

- MPLS
- IPv6
- SRv6
- VxLAN
- GRE
- ?

Example of the encapsulation of the APN attribute on the IPv6 data plane



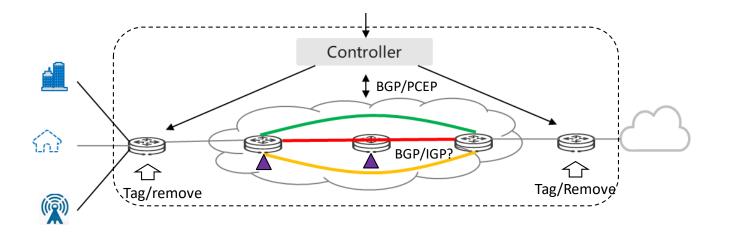


https://tools.ietf.org/html/draft-li-6man-app-aware-ipv6-network-03

Delivery of the APN attribute through the control plane protocols

- BGP
 - Between PE
 - · Between the Controller and the PE
- PCEP
 - · Between the Controller and the PE

IGP?



Management of the APN attribute via NETCONF/YANG

- YANG model for the NBI of the controller
- YANG model for the SBI of the controller

```
https://datatracker.ietf.org/doc/html/draft-sun-opsawg-sdwan-service-model-04
          +--rw application* [app-id]
            +--rw app-id syc-id
            +--rw ac* [name] =application criteria
               +--rw name
                                                string
               +--rw (match-type)?
                   +-+: (match-flow)
                      +--rw match-flow
                        +--rw ethertype?
                                                 uint16
                        +--rw cvlan?
                                                 uint8
                        +--rw ipv4-src-prefix?
                                                 inet:ipv4-prefix
                        +--rw ipv4-dst-prefix?
                                                 inet:ipv4-prefix
                        +--rw 14-src-port?
                                                  inet:port-number
           Match
                        +--rw 14-dst-port?
                                                  inet:port-number
                        +--rw ipv6-src-prefix?
                                                inet:ipv6-prefix
                        +--rw ipv6-dst-prefix?
                                                 inet:ipv6-prefix
                        +--rw protocol-field?
                                                 union
                  +--: (match-application)
                      +--rw match-application? identityref
           --rw application-group* [app-group-id]
            +--rw app-group-id
                                  svc-id
                                  -> ../../application/app-id
            +--rw app-id*
           -rw policy* [policy-id]
            +--rw policy-id
                                    svc-id
            +--rw policy-package
               +--rw encryption?
                                       enumeration
               +--rw public-private?
                                       enumeration
               +--rw local-breakout?
                                       boolean
               +--rw billing-method?
                                       enumeration
               +--rw backup-path?
                                       enumeration
               +--rw bandwidth
                  +--rw commit?
                                 uint32
                                                    Policy
                  +--rw max?
                                  uint32
          +--rw endpoints* [endpoint-id]
            +--rw endpoint-id
                                          svc-id
            +--rw site-role?
                                          identityref
            +--rw site-attachment
               +--rw site-id? -> /sdwan-svc/sites/site/site-id
            +--rw endpoint-policy-map
               +--rw app-group-policy* [app-group-id]
                  +--rw app-group-id
                                        leafref
                  +--rw policy-id?
                                        leafref
               +--rw app-policy* [app-id]
                  +--rw app-id
                                      leafref
                                     1eafref Policy
                  +--rw policy-id?
```

https://tools.ietf.org/html/draft-wu-idr-flowspec-yang-cfg-01

```
+--rw flowspec-cfg
   +--rw flowspec-policy* [policy-name]
      +--rw policy-name
      +--rw vrf-name?
                              string
      +--rw address-family? identityref
      +--rw flowspec-rule* [rule-name]
         +--rw rule-name
         +--rw flowspec-component* [component-type]
            +--rw component-type
            +--rw (component)?
                ---:(destination-prefix)
                  +--rw destination-prefix?
                                              inet:ip-address
               +--: (source-prefix)
                  +--rw source-prefix?
                                              inet:ip-address
               +--: (ip-protoco1)
                  +--rw ip-protocol* [op value]
                     +--rw op
                                    numeric-operator
                     +--rw value
                                   uint16
               +--: (port)
                  +--rw port* [op value]
                     +--rw op
                                    numeric-operator
                                   uint16
                     +--rw value
                --: (destination-port)
                  +--rw destination-port* [op value]
                                    numeric-operator
                     +--rw op
                     +--rw value
                                   uint16
               +--:(source-port)
                  +--rw source-port* [op value]
                                    numeric-operator
                     +--rw value
                                    uint16
               +--: (icmp-type)
                  +--rw icmp-type* [op value]
Match
                     +--rw op
                                    numeric-operator
                     +--rw value
                                    uint8
              +--: (icmp-code)
                  +--rw icmp-code* [op value]
                     +--rw op
                                    numeric-operator
                     +--rw value
                                    uint8
               +--: (tcp-flags)
                  +--rw tcp-flag* [op value]
                                    bitmask-operator
                     +--rw op
                     +--rw value
                                    uint16
               +--: (packet-length)
                  +--rw packet-length* [op value]
                     +--rw op
                                    numeric-operator
                     +--rw value
                                    mint16
               +--: (dscp)
                  +--rw dscp* [op value]
                     +--rw op
                                    numeric-operator
                     +--rw value
                                    dscp-type
               +--:(fragment)
                  +--rw fragment* [op value]
                     +--rw op
                                    numeric-operator
                     +--rw value
                                    fragment-type
         +--rw flowspec-action* [action-type]
            +--rw action-type
                                  action-type
            +--rw (action)?
                +--:(traffic-rate)
                  +--rw rate?
                                        float
               +--: (redirect)
 Action
               +--rw route-target?
                                        string
               .
+--:(traffic-marking)
                  +--rw remark-dscp?
                                        dscp-type
```

Next step

• Work on the Charter of the APN working group, who wants to get involved?

• Work on the drafts on solutions, who wants to get involved?

Thank you!

APN Activities

- **Side Meetings** @IETF105 & IETF108
- Hackathons @IETF108 & IETF109 & IETF110
- **Demos** @INFOCOM2020 & 2021
- **APN Mailing List** Discussions apn@ietf.org









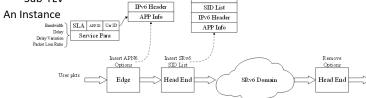
https://github.com/APN-Community

https://www.ietf.org/blog/ietf109-bofs/ https://www.ietf.org/blog/ietf110-bofs/

Application-aware G-SRv6 network

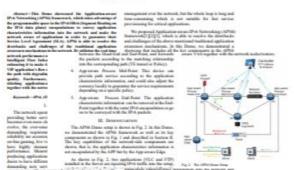
- Champions
 - Jianwei Mao (maojianwei@...)
 - Cheng Li (c.l@...)
 - Shuping Peng (pengshuping@...)
- - Develop functions of Generalized SRv6 (G-SRv6
 An Instance
 - Combine G-SRv6 with APN6, to achieve Applica
- Specifications
 - draft-lc-6man-generalized-srh
 - → draft-cl-spring-generalized-srv6-np
 - → draft-cl-spring-generalized-srv6-for-cmpr
 - → draft-li-6man-app-aware-ipv6-network
 - → draft-li-apn-framework

- Application-aware traffic control.
 - Make use of the IPv6 extensions header to convey the service requirements, in the form of APN6 options and optional Sub-TLV.
 - Determine the SRv6 SID List based on the encapsulated options and Sub-TLV



https://trac.ietf.org/trac/ietf/meeting/wiki/110hackathon https://trac.ietf.org/trac/ietf/meeting/wiki/109hackathon https://trac.ietf.org/trac/ietf/meeting/wiki/108hackathon

Demo Abstract: APN6: Application-aware IPv6 Networking



Application-aware G-SRv6 network enabling 5G services

Cheng Li, Jianwei Mao, Shuping Peng, Yang Xia, Zhibo Hu, Zhenbin Li Huawei Technologies, Beijing, China (c.l. maoijanwei, pengshuping, volanda xia, huzhibo, lizhenbin) @huawei.co

bstract—This demo showcased how application-aware G-SRv6 network provides fine-grained traffic steering with more economical IPv6 source routing encapsulation, effectively supporting 5G eMBB, mMTC and uRLLC services. G-SRv6, a

ompressed SID with G-SRv6, allowing most me upport up to 10 SIDs processing with ecirculation, significantly mitigating hardware processing overhead and facil deployments. Furthermore, for the first nd save the transmission overhead in th

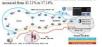
I. INTRODUCTIO As 5G and industry verticals evol services with diverse but demanding re latency and high reliability are acce Different applications have differentiate Agreement (SLA). For instance, on-l demanding requirements on latency, l high requirements on both latency and b traffic mainly requires more bandwidth latency. However, in current network naware of the traffic type traversing t network infrastructure essentially di this issue, Application-aware IPv6 networking(ArN6) [1] is

proposed, which takes advantage of the programmable space in he IPv6/SRv6 packet encapsulations to convey application aware information into the network layer, and makes network aware of applications and their requirements in order to provide fine-grained application-aware services. SRv6 [2], as the underlying network protocol supporting APN6, enables the ingress node to explicitly program the forwarding path of packets by encapsulating/inserting ordered

Segment ID (SID) list into the Segment Routing Header (SRH) at the ingress node, where each SID is 128-bit long. The SLA can be satisfied by steering the application packets into an explicit SRv6 programmable forwarding path. However, in some scenarios such as strict Traffic Engineering(TE), many SIDs will have to be inserted in the SRH, resulting in a lengthy SRH which imposes big challenges on the hardware processing, and affects the transmission efficiency especially for the small size packets in 5G uRLLC or mMTC scenarios. For instance, the size of an SRv6 encapsulation with 10 SIDs is 208 bytes, which exceeds the parser window of most merchant silicon chipsets (e.g., Jericho2) and causes expensive packet

recirculation. This has become a big obstacle for SRv

encode Generalized SIDs (G-SID) in the Generalized SRH (G SRH), where a G-SID can be a 128-bit SRv6 SID, a 32-bit



Switching based on IPv6 DA

Inserting APN6 option

II. APPLICATION-AWARE G-SRV6

Normally SRv6 SIDs are allocated from an address block within an SRv6 domain, so the SIDs share the common prefix (CP) of the address block[5]. An SRv6 SID has the format



Fig. 1. Format of the 128-bit SRv6 SID and 32-bit G-SII

among the SIDs in a SID list, while the common prefix and argument parts are redundant. Removing the redundant parts of the SID list can reduce the overhead. Generalized SRv6 (G-SRv6) realizes this idea. It only carries the compressed SID consisting of node ID and Function ID in the SRH, so that the size of the SRH is compressed. Theoretically, up to 75% overhead of the SRv6 SID list can be reduced. The SID size

https://ieeexplore.ieee.org/abstract/document/9162934 https://www.youtube.com/watch?v=ONgwxKVmPp0

Implemented Functions

- We've implemented the demo based on P4, and conducted some simulations based on BMv2.
- Functions in Demo

 - The encapsulation of APN6 Options and Serice-Para Sub-TLV, support 2 types of APN6 Options and 4 types of Sub-TLV
 - 2. The encapsulation of the SRv6 SID List according to IPv6 DA and APN6 options
 - Basic SRv6 END SID processing

Performance Evaluation

Getting Ingress times

Processing Latency

Result

- Send 50,000 packets in each experiment.
- The interval between 2 packets is 1ms.
- All results are in nanoseconds

etermining to insert APP awar nserting an APP aware ID option · Without processing of APN6 Switching based on IPv6 DA

Switching based on IPv6 DA

Experiment Mean STDEV 0.56514087 364.07436 0.611774343 0.046633473

References – Drafts have been updated

Please find the APN BoF proposal in the IETF wiki for more information.

https://trac.tools.ietf.org/bof/trac/wiki/WikiStart

The archived discussions in this APN mailing list can be found here.

https://mailarchive.ietf.org/arch/browse/apn/

To subscribe the APN Mailing list,

https://www.ietf.org/mailman/listinfo/apn

Here are some relevant drafts and materials for your reference.

Scope & Gap analysis

• https://tools.ietf.org/html/draft-peng-apn-scope-gap-analysis

Problem statement & Use cases

- https://tools.ietf.org/html/draft-li-apn-problem-statement-usecases
- https://tools.ietf.org/html/draft-liu-apn-edge-usecase
- https://tools.ietf.org/html/draft-zhang-apn-acceleration-usecase
- https://tools.ietf.org/html/draft-yang-apn-sd-wan-usecase

Framework

https://datatracker.ietf.org/doc/draft-li-apn-framework/

Security & Privacy

https://datatracker.ietf.org/doc/draft-peng-apn-security-privacy-consideration

APN Community

https://github.com/APN-Community