An IPv6 Flow Label Specification Proposal

<draft-rajahalme-ipv6-flow-label-00.txt>

Jarno Rajahalme, Nokia

Alex Conta, Transwitch

IETF #52, Salt Lake City

NOKIA

Outline

- Flow Support in IPv6
- Why Bother Now?
- Brief History of the IPv6 Flow Label
- Requirements
- Problems with the RFC 2460 Definition
- Proposed New Definition
- <Source Address, Flow Label, Destination Address>
- The Flow Label Value
- The Way Forward

NOKIA

Flow Support in IPv6

- **A flow** is a sequence of packets that should receive specific non-default handling from the network
- **Flow state** is established in a subset of the IP nodes on the path
  - Defines the “handling” the packets should receive
  - Not likely set up on every hop end-to-end
  - But useful at the network edges (access networks)
    - mapping to a specific Behavior Aggregate for the “core” routers
- The **Flow Label field** should enable classification of packets belonging to a specific flow
- Without the flow label the classifier must use transport next header value and port numbers
  - Less efficient (need to parse the option headers)
  - May be impossible (fragmentation or IPsec ESP)
  - Layer violation hinders introduction of new transport protocols (e.g. SCTP, DCP)
  - More state in the router forwarding plane

NOKIA

Why Bother Now?

- Flow support is an IPv6 feature, but “still experimental and subject to change”
  - Non-normative appendix in a Draft Standard
  - Appendix A is inadequate even for RSVP/Integrated Services
- Soon there will be a lot of incomplete IPv6 stacks out there
  - Only the default (zero) flow label support
- Customers are waiting for this:
  - “On short term there is no reason to use IPv6 because of QoS - waiting the IPv6 flow label to be better defined and used for QoS in the future”
    - Operator at the “Deploying IPv6 Networks” event (Paris 20.-23.11.2001)
- If IPv6 Flow Label is now standardized, NSIS and other WGs can pick it up and develop flow state establishment methods using it
- An item in the proposed new IPv6 WG charter

NOKIA

Brief History of the IPv6 Flow Label

- SIPP (draft-ietf-sipp-spec-01.txt): 28-bit Flow Label
  - Contained the “TClass” field (the first 4 bits), semantics of which optionally flow specific
  - Flow identified by the source address and the “Flow ID”
  - Pseudo-random flow label as a hash key to locate flow state
- RFC 1883: 24-bit Flow Label
  - “Priority” field separated (4 bits)
  - “Opportunistic mode”
    - The rest of the IPv6 headers were to be used as the flow state establishment method
    - Strict rules for routing headers, hop-by-hop options, and destination address: Must remain the same for all packets in the flow
    - Allowed use of cached next-hop, header contents
    - Max lifetime of 6 seconds for “opportunistically” created flow state
- RFC 2460: 20-bit Flow Label
  - IPv4 compatible Traffic Class separated
  - Opportunistic state setup abandoned, but rules kept in the Appendix A

NOKIA

Requirements

- The IPv6 flow label should enable flow classification without dependencies on higher layer protocol headers
- The semantics-free nature of the flow label, when out of context of the source and destination addresses, should be maintained
- Should provide end-to-end transparency for labeled flows
- The semantics of the flows should to be set up with explicit flow state establishment methods
  - All specifics out of scope
  - As few restrictions as possible on such methods
- The IPv6 specification must enable co-existence of different methods in both hosts and routers
- Minimal changes

NOKIA

Problems with the RFC 2460 Definition

- Inadequate support for RSVP
  - RFC 2205:
    - “IPv6 inserts a variable number of variable-length Internet-layer headers before the transport header, increasing the difficulty and cost of packet classification for QoS.”
    - “Efficient classification of IPv6 data packets could be obtained using the Flow Label field of the IPv6 header. The details will be provided in the future.”
  - Current Session object definitions require transport header lookup, even if Flow Label based filter is used
  - Three reservation styles: Fixed-Filter, Shared-Explicit (SE), Wildcard-Filter (WF)
  - WF style applicable on e.g. audio conference on a multicast group
    - Shared resource reservation for up to e.g. 3 speakers simultaneously
    - 100 members would require 100 classifier rules with the SE reservation style
    - Only one classifier rule with the Wildcard-Filter style
  - WF style has only the Session object, no Filter Specs
    - Classification without port numbers not possible
    - Could be solved if the same flow label value could be used with different flows to different destinations (Flow Label in the Session object)
    - But a flow is uniquely identified by the source address and the flow label, AND the destination address must remain constant
  - The above confirmed with RFC 2205 authors

NOKIA

Problems with the RFC 2460 Definition (cont.)

- Problematic requirements for non-signaled flow state establishment methods
  - Use of pseudo-random values
  - Constraints for the hop-by-hop and routing headers
    - The history of the “opportunistic” flow state establishment
- Too restricting requirement for Flow Label value re-use
  - Not needed if flow state is appropriately set up for the new flow
- Ambiguity on the end-to-end nature of the Flow Label
  - No IPsec AH protection

Proposed New Definition

- The draft contains new text proposed to be included into the next revision of the IPv6 specification
- Main points:
  - A flow is uniquely identified by the <source address, flow label, destination address> triplet
  - Labeled flows have a non-zero Flow Label value
  - Non-zero Flow Label value is end-to-end immutable
  - The special handling for a flow is defined by a flow state establishment method, out of scope
    - The methods can include rules for the hop-by-hop and routing headers, if needed
  - The host must keep track of the Flow Label values in use by any flow state establishment method
- RFC 2460 Appendix A to be removed
- Draft proposes a set of requirements for flow state establishment methods
  - Spanning from the definition above

NOKIA

<Source Address, Flow Label, Destination Address>

- Allows the same flow label value to be used with different destinations
  - Enables definition of flow label based Session object in RSVP
- Efficiency (276 bits to compare)?
  - The real benchmark is against the 5-tuple classifier
    - Less bits than in the 5-tuple (296 bits), and in fixed and always available positions
    - Current implementations already implement 5-tuple based classification (when it is possible)
  - HW implementation performance better than 5-tuple
    - Don’t need to assemble bits from variable offsets
  - Robust classifier would check the destination address even with the <source address, flow label> pair
- The added flexibility is worth it

The Flow Label Value

- Any non-zero value
  - No specific format
  - Values can be re-used at will, as long as proper state is established
  - Flow state establishment methods free to choose stricter rules
  - Pseudo-randomness not required for efficient classification implementation
    - Routers can utilize CAMs, search trees, or compute a hash key over the whole classifier triplet
  - Router implementations should not depend on the distribution of the Flow Label values
- A Flow Label value is meaningless by itself
- The non-zero value guaranteed to be received by the destination host
  - “end-to-end immutable”
  - Temporary changes possible, if so defined by the flow state establishment method (out of scope)

NOKIA

The Way Forward

- Any specific issues to be discussed/resolved?
- How to proceed?
  - Revision?
  - What is the message to other WGs?
    - nsis, ipfix, mmusic, ...

NOKIA