DetNet Working Group D. Trossen
INTERNET-DRAFT Huawei
Intended Status: Standards Track F.-J. Goetz
Expires: August 11, 2021 J. Schmitt
Siemens
February 11, 2021

DetNet Control Plane Signaling
draft-trossen-detnet-control-signaling-01.txt

Abstract

This document provides solutions for control plane signaling, in accordance with the control plane framework developed in the DetNet WG. The solutions cover distributed, centralized, and hybrid signaling scenarios in the TSN and SDN domain. We propose changes to RSVP IntServ for a better integration with Layer 2 technologies for resource reservation, outlining example API specifications for the realization of the revised RSVP (called RSVP-TSN in the document).

Status of this Memo

Copyright and License Notice

Table of Contents

1. Introduction

The authors in [ID.malis-detnet-controller-plane-framework] provide an overview of the DetNet control plane architecture along three possible classes, namely (i) fully distributed control plane utilizing dynamic signaling protocols, (ii) a centralized, SDN-like, control plane, and (iii) a hybrid control plane.

When investigating the usage of RSVP [RFC2205] for the signaling of deterministic IP connectivity in combination of underlying Layer 2 mechanisms, considerations arise for the development of the detnet-specific RSVP protocol, called RSVP-TSN in the following.

This document will outline use cases spanning the classes of control planes introduced in [ID.malis-detnet-controller-plane-framework], followed by the design rationale and specification for the proposed RSVP-TSN protocol.

1.1. Terminology

This document uses the terminology established in the DetNet Architecture [RFC8655], and the reader is assumed to be familiar with that document and its terminology.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Use Cases

Based on the detnet stack model [RFC 8938], "Resource allocation", located in the forwarding sub-layer, is split into RSVP-TSN IP flow signaling and underlying TSN subnet stream reservation. Stream reservation within TSN subnetworks can be organized with a decentralized, centralized or hybrid configuration model. The notion of TSN in these use cases and the remainder of the document assumes a Bridged-Ethernet LAN with enhancements for time-sensitive networking.

2.1. Distributed DetNet User Network Interface (ddUNI)

The following figure illustrates the principle of a hybrid DetNet using RSVP-TSN for DnFlow signaling in a TSN aware customer network. DetNet/TSN end nodes signal their DnFlows over RSVP-TSN. In parallel, the TSN control plane triggers the stream reservation within a TSN aware customer network, using e.g., LRP/RAP. The control plane solution of a TSN customer network is independent from RSVP-TSN signaling and can cover distributed, centralized or hybrid reservation scenarios.

An RSVP detnet Edge Router supports RSVP-TSN signaling of DnFlows and covers DnFlow signaling supported by the associated detnet aware core network. Although the DetNet control plane within the DetNet core network is without support for RSVP, it still supports the DetNet Flow and Service Information Model [ID-detnet-flow-information-model] and can be organized in a decentralized or centralized (SDN-like) manner.

```
                                   +--------+
           +--------+             +--------+|             +--------+
   +----+  |        |  +-------+  |        ||  +-------+  |        |  +----+
   |RSVP|  |  TSN   |  | RSVP  |  |DetNet  ||  | RSVP  |  |  TSN   |  |RSVP|
   |TSN |--| Cust.  |  |DetNet |  | aware  ||  |DetNet |  | Cust.  |--|TSN |
   |End-|  |Network |--| Edge  |--| Core   ||--| Edge  |--|Network |  |End-|
   |Node|  |        |  |Router |  |Network ||  |Router |  |        |  |Node|
   +----+  +--------+  +-------+  |        |+  +-------+  +--------+  +----+
                                  +--------+
```

Figure 1 : Distributed DetNet UNI

2.2. Fully Distributed Detnet Control Plane (still supports ddUNI)

The following figure illustrates a fully distributed DetNet using RSVP-TSN for DnFlow signaling in TSN aware customer networks and RSVP aware core networks. In difference to the previous scenario, the detnet control plane within the detnet aware core network still supports RSVP to establish detnet end-2-end connectivity.

```
                                   +--------+
           +--------+             +--------+|             +--------+
   +----+  |        |  +-------+  |  RSVP  ||  +-------+  |        |  +----+
   |RSVP|  |  TSN   |  | RSVP  |  | aware  ||  | RSVP  |  |  TSN   |  |RSVP|
   |TSN |--| Cust.  |  | Edge  |  | Core   ||  | Edge  |  | Cust.  |--|TSN |
   |End-|  |Network |--|Router |--|Network ||--|Router |--|Network |  |End-|
   |Node|  |        |  +-------+  |        |+  +-------+  |        |  |Node|
   +----+  +--------+             +--------+              +--------+  +----+
```

Figure 2 : Fully Distributed DetNet UNI

3. Design Rationale

This section will explore the design rationale behind the development of RSVP-TSN. The next two sub-sections outline aspects derived from the comparison of RAP, as a Layer2 mechanism, and RSVP, before highlighting key design considerations for the presentation of RSVP-TSN in Section 4.

3.1. RAP Reservation in TSN vs RSVP IntServ Model

Layer2 reservation in TSN-based networks is supported through RAP, providing a maximum of 8 classes of traffic where the frame priority code point (PCP) is used to select the Resource Allocation (RA) class at the ingress bridge. Streams within a single RA class are queued in a single traffic class where the latency of the stream is guaranteed per hop and per RA class.

This model contrasts with the RSVP IntServ [RFC2212] model, which provides a flow bandwidth driven latency model with a separate transmission queue per flow, not a class-based model like in the aforementioned RAP model.

This difference in models poses a number of challenges:

1. Is RSVP IntServ (as defined in [RFC2212]) the right starting point?

2. How to efficiently map the different reservation styles of RSVP-TSN onto RAP?

3. What is the nature of the interface between RSVP-TSN and RAP?

4. How is the binding between L3 signaling (RSVP IntServer) and L2 signaling (RAP) realized, e.g., mapping of Stream-ID?

The following sub-sections elaborate on the various aspects in addressing those challenges.

3.2. Similarities and Differences between RSVP and RAP

The following sub-sections will outline various aspects to be considered when designing the interfaces between RSVP-TSN and RAP, namely the assumptions on network nodes (Section 3.2.1), the mapping of the latency model used in both models (Section 3.2.2), the dealing with latency margins (Section 3.2.3), the dealing with Jitter and non-shaping nodes (Section 3.2.4), and the mapping of resource reservation styles (Section 3.2.5).

3.2.1. Assumptions on Network Nodes

RSVP assumes three different nodes over which a reservation can be done, namely

- Shaping node, which implements the RSVP signaling and shaping on the data plane,

- None shaping node, which implements the RSVP signaling and is capable of estimating the latency caused by this node

- Legacy node, which does neither implement RSVP nor any shaping.

RAP assumes properties common to all nodes within a reservation domain:

- All nodes take part in the signaling process

- Different data plane architectures are supported albeit limited to those defined in IEEE 802.1Q.

- Bridging between different (heterogeneous) data planes is achieved through a peer-to-peer model where every upstream node is a talker for the next downstream node.

3.2.2. Mapping of Latency Model

RSVP assumes a weighted fair queuing (WFQ) at the data plane, where a listener is able to influence therefore the latency through the reserved bandwidth per flow.

RAP assumes one traffic class with given interference per common RA class, resulting in a per hop latency for all stream within a single RA class. The E2E latency is just signaled by accumulating hop latency while the allowed interference determines the amount of allowed flow per RA class. Here, the listener is unable to influence the latency but the stream requirement is signaled upstream.

3.2.3. Dealing with Latency Margins

RSVP provides the notion of slack [RFC2212] per flow, which can be consumed by the processing node in the network to enable additional reservations.

In RAP, every listener of a stream propagates its required latency upstream to the talker. Latency margins are not handled directly by RAP, while the per hop latency of an RA class is preconfigured by management. In each node, the per RA class upstream required latency of all streams can be used to locally calculate the latency margins per hop. The management system can then use this information to adjust the per hop maximum latency at runtime.

3.2.4. Dealing with Jitter and Non-Shaping Nodes

RSVP has two different parameters to propagate the maximum nonconformance to the leaky bucket model introduced through jitter and non-shaping nodes. These can be accumulated by non-shaping nodes, i.e., those which implement the RSVP protocol but are not performing shaping at the data plane.

Within RAP, there is no distinction between shaping and non-shaping nodes since all nodes adhere to the data plane architecture defined in IEEE 802.1Q. Heterogeneous data planes are possible as long as assurances to the next hop can be upheld, while RA class attributes are used to propagate data plane behavior (e.g., shaper) to the next neighbor.

3.2.5. Mapping Resource Reservation Styles

RSVP uses the notion of 'sessions', which are able to maintain different kinds of end-to-end connectivity and resource styles, namely fixed (i) filter style, (ii) shared explicit style, and (iii) wildcard filter style - see [RFC2205] and Figure 3. It is important to note that in RSVP, both sender selection and resource styles are controlled by the receiver; we return to this issue in our next section, discussing the rationale for the proposed design for RSVP-TSN.

The current draft version of RAP supports only distinct explicit mode of reservation, while in principle supporting reservation between one talker and multiple listeners. Bridged Ethernet technology is also able to support the shared resource modes as specified by RSVP. Also a new resource style called Coordinated Shared Resource Style is planned.

Sender Selection	Resource Style		
	Distinct	Shared	Coordinated Shared
Explicit	supported	supported	supported
Wildcard		supported	

Figure 3: Resource Style and Sender Selection [RFC2205]

3.3. Design Considerations for RSVP-TSN

3.3.1. Rationale

Continuing from Section 3.2.5, in RSVP (for IntServ), the receiver initiates resource style and sender selection through the Resv message being sent upstream, while path state being setup through the Path message from the sender to the receiver upon receiving the Resv message.

When looking into an integration with lower layer APIs, such as the TSN API, we identify key differences in WHEN these lower layer APIs decide if a reservation is possible:

1. For a new Announce downstream, each L2 node decides that if this stream was reserved at this port, would there be enough resources available to do so?

2. For a new Attachment upstream, each L2 node will lock the required resources and bandwidth exclusively for this stream. For every L2 node local non-locked Announce, the L2 node will decide the same question as in item 1 and refresh and propagate the necessary states accordingly.

It is important to note that steps 1 and 2 only work if the 'resource style' is already known by the Announce propagation.

3.3.2. Splitting Control over Resource Style and Sender Selection

In order to allow for an efficient resource reservation at the lower network level by implementing the steps 1 and 2 in Section 3.3.1, we propose to split the control over 'resource style' and 'sender selection' in that in RSVP-TSN the sender controls the 'resource style' and the listener controls the 'sender selection'.

3.3.3. Coordinated Shared Resource Style

Independent from the efficient realization of lower layer resource reservation, we have also identified a requirement in industrial use cases to support a large amount of deterministic connections with small data usage. In those cases, separate reservation for each connection could be inefficient.

To address this, we propose to introduce another 'resource style' called 'Coordinated Shared', which would indicate the use of scheduling (of those many deterministic connections) at L2-Listener and L3-Receiver level. A first proposal for a solution in the TSN RAP protocol was presented to the IEEE in [CHEN-IEEE]

3.3.4. DnFlow DestinatinIpAddress Resolution

To support deterministic QoS Bridged Ethernet has introduced Streams. Streams differ from legacy traffic within a Bridged Ethernet sub-network because streams belong to a traffic class which is uniquely identified by a priority value in the range of 0 through 7. Streams within an TSN aware Bridged Ethernet sub-network also need unique destination MAC-address for identification. The priority and the unique destination MAC-address indicates a Stream within a virtual LAN (VLAN). The IEEE 802.1CQ draft for "Multicast and Local Address Assignment" specifies protocols and procedures of locally unique assignment for 48-bit and 64-bit addresses in IEEE 802 networks.

Streams do not use the interface Mac-Address as destination MAC-Address within a Bridged Ethernet. Further enhancements for IP address resolutions are required within TSN and detnet aware end-systems and routers and to map one or multiple detnet IP flows to a stream destination MAC-Address. DnFlows are identified by a "6-tuple" that refers to information carried in IP and higher layer protocol headers. The 6-tuple referred to in this document is the same as that defined in [RFC3290]. Specifically, 6-tuple is DestinationIpAddress, SourceIpAddress, Protocol, SourcePort, DestinationPort, and differentiated services (DiffServ) code point (DSCP).

4. RSVP-TSN

In this section, we specify the APIs for RSVP-TSN, the message formats, as well as outline the layer and node interactions in an RSVP-TSN based system

4.1. Layer Interactions between RSVP and TSN

Figure 4 provides an overview of the interactions between L2 and L3 elements in a network deployment as an elaboration of the elements in Figure 1, also illustrating the various interfaces described in the following sections.

The application utilizes a generalized API for deterministic QoS (dQoS) that controls and signals the establishment of deterministic end-to-end DnFlow via the upper API of RSVP-TSN (uRSVP).

RSVP-TSN end nodes utilize RSVP-TSN to signal DnFlows to a detnet aware edge router. This L3 network interface is called "Distributed DetNet User Network Interface" (ddUNI).

The lower API of RSVP-TSN (lRSVP) interacts with the TSN control plane to trigger the establishment of streams in an TSN aware (e.g. customer) sub-network. The TSN control plane for the establishment of streams in a TSN sub-network can be organized decentralized, centralized or hybrid for stream reservation. For stream establishment based on centralized scheduling, a third-party protocol like RESTCONF is typically used.

```
+-----------+
|Application|
+-----------+
  | dQoS    |
  |         |
  |C       S|
  |         |
  | uRSVP   |
+-----------+                                       +-------------+
| RSVP-TSN  |<------------------------------------->|  RSVP-TSN   |
+-----------+                                       +-------------+
  | lRSVP   |                                         |   |   |   |
  |         |                                         |   |   |   |
  |M&A     S|                                         |M S|   |M S|
  |         |                                         |   |   |   |
+-----------+      +---------------------------+     +-------------+
| RSVP-TSN   |<===>| TSN aware                 |<===>| RSVP/DetNet |
| End-Node  |      | Customer Subnetwork       |     | Edge Router |
+-----------+      +---------------------------+     +-----+ +-----+

<--->   RSVP-TSN DNFlow Signaling
<===>   TSN Stream Reservation
dQoS    API for deterministic QoS
uRSVP   upper API of RSVP-TSN
lRSVP   lower API of RSVP-TSN
C Controls        S Signals        M&A Maps and Aggregation
```

Figure 4: Layer Interactions between RSVP and TSN

4.2. API for Deterministic QoS (gQoS)

The description of a generalized API to support deterministic QoS is not part of this document.

4.3. RSVP-TSN upper API (uRSVP)

The definition of the upper and lower APIs of RSVP-TSN is based on the DetNet flow information model [ID-detnet-flow-information-model].

This interface is oriented on the interface specified by RSVP-IntServ (RFC 2205). Most of the changes are due to mapping resource reservation styles (see Section 2.4.5).

Sender

Call: Open Session (oriented to the RSVP-IntServ interface)

Request parameter (make use of pieces from the DnFlowSpecification)

- DestinationIpAddress, Protocol, DestinationPort

Response parameter:

- SessionID

Call: Add DnFlow

Request parameter (make use of pieces from the DnFlowSpecification)

- SessionID, SourceIpAddress, SourcePort, DSCP

- DnTrafficSpecification: Interval, MaxPacketsPerInterval, MaxPayloadSize, MinPayloadSize

- DnFLowRank

- Select one of the Resource Style: Distinct, Shared, CoordinatedShared

- Data TTL, PATH MTU size, LossRate

Notes for new parameter:

The DSCP is required to map DnFlows according their service class to offered service classes of the lower layer.

The resource style for an DnFlow is announced by the sender within the path message.

The LossRate is accumulated per DnFlow from Sender to Receiver.

Upcall: DnFlow

- Session ID

- One of the Info_type: RESV_EVENT; PATH_ERROR

Receiver

Call: Open Session

Request parameter (make use of pieces from the DnFlowSpecification)

- DestinationIpAddress, Protocol, DestinationPort

Response parameter

- SessionID

Call: Join DnFlow

Request parameter

- SessionID

- Select one of the DnFlow Source Selection: Wildcard, List of explicit sources with SourcePort

- MaximumPacketSize

- Extended Traffic Specification: MaximumExpectedLatency

Notes for new parameter:

The Source Selection is split from the RSVP-IntServ Reservation Style but still follows the rules defined by RSVP-IntServ.

The extended traffic specification MaximumExpectedLatency is propagated and merged to a minimum upstream from receiver to sender.

Upcall: DnFlow

- SessionID

- SourceIpAddress (Sender)

- SourcePort

- One of the Info_type: RESV_EVENT; PATH_ERROR

General

Call: Close Session

Request parameter

- SessionID

4.4. RSVP-TSN lower API (lRSVP)

Sender

Call: Add DnFlow

Request parameter

- SessionID, Interface, DnFlowID, DestinationIpAddress, DSCP

- DnTrafficSpecification: Interval, MaxPacketsPerInterval, MaxPayloadSize, MinPayloadSize, MinPacketsInterval

- One of the Resource Styles: Distinct, Shared, Coordinated Shared

Response parameter

- TransportFlowID (TSN StreamID)

Notes for new parameter:

The DnFlowID is a local parameter to correlate DnFlows to transport flows (e.g., TSN Stream).

The TransportFlowID correlates the DnFlow to the lower layer transport flow, e.g., TSN Stream ID.

Upcall: DnFlow

Response parameter

- SessionID

- TransportFlowID

- One of the Info_type: RESV_EVENT, RES_MODIFY_EVENT

Receiver

Call: Join DnFlow

Request parameter

- SessionID, Interface, DnFlowID, TransportFlowID

- MaximumPacketSize

- Extended Traffic Specification: MaximumExpectedLatency

Notes for new parameter:

(see notes above)

Upcall: DnFlow

Response parameter

- SessionID, TransportFlowID

- One of the Info_type: ANNOUNCE_EVENT, ANNOUNCE_MODIFY_EVENT

4.5. RSVP-TSN Message Formats

TBD

5. Security Considerations

Editor’s note: This section needs more details.

6. IANA Considerations

N/A

7. Conclusion

This draft outlines the possible control plane signaling in deterministic networking environments for distributed, centralized and hybrid deployments.

For this, changes to the RSVP signaling have been proposed in the form of RSVP-TSN for a better alignment of the Layer 3 signaling with that of emerging Layer 2 solutions, together with suggested API specifications for the realization of the L3 to L2 interfaces in endpoints.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

[RFC2212] Shenker, S., Partridge, C., and Guerin, R., "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

[RFC2205] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jasmin, " Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC8938] B. Varga, Ed, J. Farkas, L. Berger, A. Malis, S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC8938, November 2020.

8.2. Informative References

[ID.malis-detnet-controller-plane-framework] A. Malis, X. Geng, M. Chen, F. Qin, B. Varga, "Deterministic Networking (DetNet) Controller Plane Framework", draft-malis-detnet-controller-plane-framework-05 (work in progress), 2020.

[ID-detnet-flow-information-model] Balazs Varga, Janos Farkas, Rodney Cummings, Yuanlong Jiang, Don Fedyk, "DetNet Flow and Service Information Model", draft-ietf-detnet-flow-information-model-14 (work in progress), 2021

[CHEN-IEEE] F. Chen, F.J. Goetz, M. Kiessling, J. Schmitt, " Support for uStream Aggregation in RAP (ver 0.3)" (work in progess), Jan 2019, <http://www.ieee802.org/1/files/public/docs2019/dd-chen-flow-aggregation-0119-v03.pdf>

[RAP_IEEE] IEEE, "P802.1Qdd - Resource Allocation Protocol", (work in progress), <https://1.ieee802.org/tsn/802-1qdd/>

Authors' Addresses

Dirk Trossen
Huawei Technologies Duesseldorf GmbH
Riesstr. 25C
80992 Munich
Germany

Email: Dirk.Trossen@Huawei.com

Franz-Josef Goetz
Siemens AG

Gleiwitzer-Str. 555
90475 Nuremberg
Germany

Email: franz-josef.goetz@siemens.com

Juergen Schmitt
Siemens AG
Gleiwitzer Str. 555
90475 Nuremberg
Germany

Email: juergen.jues.schmitt@siemens.com