TSVWG
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

P. Thubert, Ed.
Cisco

October 30, 2017

A Transport Layer for Deterministic Networks
draft-thubert-tsvwg-detnet-transport-01

Abstract

This document specifies the behavior of a Transport Layer operating over a Deterministic Network and implementing a DetNet Service Layer and a Northbound side of the DetNet User-to-Network Interface.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Over last twenty years, voice, data and video networks have converged to digital over IP. Mail delivery has become quasi-immediate and volumes have multiplied; long distance voice is now mostly free and the videophone is finally a reality; TV is available on-demand and games became interactive and massively multi-player. The convergence of highly heterogeneous networks over IP resulted in significant drops in price for the end-user while adding new distinct value to

the related services. Yet, and even though similar benefits can be envisioned when converging new applications over the Internet, there are still many disjoint branches in the networking family tree, many use-cases where mission-specific applications continue to utilize dedicated point-to-point analog and digital technologies for their operations.

Forty years ago, Control Information was first encoded as an analog modulation of current (typically 4 to 20 mA) that can be carried virtually instantly and with no loss over a distance. Then came digitization, which enabled to multiplex data with the control signal and manage the devices, but at the same time introduced latency to industrial processes, the necessary delay to encode a series of bits on a link and transport them along, which in turn may limit the amount of transported information. The need to save cable and simplify wiring lead to the Time Division Multiplexing (TDM) of signals from multiple devices over shared digital buses, each signal being granted access to the medium at a fixed period for a fixed duration; with TDM, came more latency, waiting for the next reserved access time. Statistical multiplexing, with Ethernet and IP, was then introduced to achieve higher speeds at lower cost, and with it came jitter and congestion loss.

A number of Operational Technology (OT) applications are now migrating to Ethernet and IP, but that comes at the expense of additional latency for the flows, to compensate for the degradation of the transport discussed above. This also comes at the expense of additional complexity in particular, applications may need to transport a sense of time, provide some Forward Error Correction (FEC) and include a jitter absorption buffer. for that reason, many applications were never ported and OT networks are still largely operated on point-to-point serial links and TDM buses.

A sense of what Deterministic Networking is has emerged as the capability to make the Application simple again and enable a larger migration of existing applicationsby absorbing the complexity lower in the stack, at the Transport, Network and Link layers. A Deterministic Network should be capable to emulate point-to-point wires over a packet network, sharing the network resources between deterministic and non-deterministic flows in such a fashion that there can no observable influence whatsoever on a deterministic flow from any other flow, regardless of the load of the network.

The generalization of the needs for more deterministic networks have led to the IEEE 802.1 AVB Task Group becoming the Time-Sensitive Networking (TSN) [IEEE802.1TSNTG] Task Group (TG), with a much-expanded constituency from the industrial and vehicular markets. In order to address the problem at the network layer, the DetNet Working

Group was formed to specify the signaling elements to be used to establish a path and the tagging elements to be used identify the flows that are to be forwarded along that path.

The "Deterministic Networking Use Cases" [I-D.ietf-detnet-use-cases] indicates that beyond the classical case of industrial automation and control systems (IACS), there are in fact multiple industries with strong and yet relatively similar needs for deterministic network services such as latency guarantees and ultra-low packet loss. The "Deterministic Networking Problem Statement" [I-D.ietf-detnet-problem-statement] documents the specific requirements for the use of routed networks to support these applications and the "Deterministic Networking Architecture" [I-D.ietf-detnet-architecture] introduces the model that must be proposed to integrate determinism in IT technology.

A DetNet network will guarantee a bounded latency and a very low packet loss as long as the incoming flows respect a certain Service Level Agreement (SLA), as typically expressed in the form of a maximum packet size, a time window of observation and a maximum number of packets per time window.

Outside the scope of DetNet, the IETF will also need to specify the necessary protocols, or protocol additions, based on relevant IETF technologies, to enable end-to-end deterministic flows. One critical element is the Deterministic Transport Layer (DetTrans) that adapts the flows coming from the Application Layer to the SLA of the DetNet Network and provide end-to-end guarantees such as loss, latency and timeliness.

The DetTrans Layer should in particular ensure that:

- the Deterministic Network setup matches the needs of the Application

- the Application flows are presented to the Deterministic Network in accordance to the SLA regardless of the way the data is passed from the application

- the use of the network is optimized so as to ensure that every byte from the application can effectively be transported

- the application flow is delivered reliably and with a bounded latency to the other Transport End Point, which may imply a FEC technique such as Network Coding, Packet Replication and Elimination (PRE), or basic 1+1 redundancy.

- the full of the application flow is served, which may require the use of multiple reservations in parallel, and the reordering of the flows

On the one hand, the Deterministic Network will typically guarantee a constant rate, so the classical Transport feature of flow control will not be needed in a Deterministic Transport. On the other hand, the Application and Transport layers may not reside in the same device as the DetNet Router and/or the IEEE Std. 802.1 TSN Bridge that acts as ingress point to the Deterministic Network. It results that a minimum reliability and flow control must take place over the Local Loop between these devices to ensure that the Deterministic Network is kept optimally fed, meaning that packets are received just in time for their scheduled transmission opportunities.

2. 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 [RFC2119].

3. On Deterministic Networking

3.1. Applications and Requirements

The Internet is not the only digital network that has grown dramatically over the last 30-40 years. Video and audio entertainment, and control systems for machinery, manufacturing processes, and vehicles are also ubiquitous, and are now based almost entirely on digital technologies. Over the past 10 years, engineers in these fields have come to realize that significant advantages in both cost and in the ability to accelerate growth can be obtained by basing all of these disparate digital technologies on packet networks.

The goals of Deterministic Networking are to enable the migration of applications that use special-purpose fieldbus technologies (HDMI, CANbus, ProfiBus, etc... even RS-232!) to packet technologies in general, and the Internet Protocol in particular, and to support both these new applications, and existing packet network applications, over the same physical network.

Considerable experience ([ODVA]/[EIP], [AVnu], [Profinet],[HART], [IEC62439], [ISA100.11a] and [WirelessHART], etc...) has shown that these applications need a some or all of a suite of deterministic features.

That suite of deterministic features includes:

1. Time synchronization of all Host and network nodes (Routers and/or Bridges), accurate to something between 10 nanoseconds and 10 microseconds, depending on the application.

2. Support for critical packet flows that:

   * Can be unicast or multicast;

   * Need absolute guarantees of minimum and maximum latency end-to-end across the network; sometimes a tight jitter is required as well;

   * Need a packet loss ratio beyond the classical range for a particular medium, in the range of 10^-9 to 10^-12, or better, on Ethernet, and in the order of 10^-5 in Wireless Sensor Mesh Networks;

   * Can, in total, absorb more than half of the network's available bandwidth (that is, massive over-provisioning is ruled out as a solution);

   * Cannot suffer throttling, flow control, or any other network-imposed latency, for flows that can be meaningfully characterized either by a fixed, repeating transmission schedule, or by a maximum bandwidth and packet size;

3. Multiple methods to schedule, shape, limit, and otherwise control the transmission of critical packets at each hop through the network data plane;

4. Robust defenses against misbehaving Hosts, Routers, or Bridges, both in the data and control planes, with guarantees that a critical flow within its guaranteed resources cannot be affected by other flows whatever the pressures on the network;

5. One or more methods to reserve resources in Bridges and Routers to carry these flows.

Robustness is a common need for networking protocols, but plays a more important part in real-time control networks, where expensive equipment, and even lives, can be lost due to misbehaving equipment. Reserving resources before packet transmission is the one fundamental shift in the behavior of network applications that is impossible to avoid. In the first place, a network cannot deliver finite latency and practically zero packet loss to an arbitrarily high offered load. Secondly, achieving practically zero packet loss for un-throttled (though bandwidth limited) flows means that Bridges and Routers have to dedicate buffer resources to specific flows or to classes of

flows. The requirements of each reservation have to be translated into the parameters that control each Host's, Bridge's, and Router's queuing, shaping, and scheduling functions and delivered to the Hosts, Bridges, and Routers.

3.2. The DetNet User-to-Network Interface (UNI)

The "Deterministic Networking Architecture" [I-D.ietf-detnet-architecture] presents the end-to-end networking model and the DetNet services; in particular, it depicts the DetNet User-to-Network Interfaces (DetNet-UNIs) ("U" in Figure 1) between the Edge nodes (PE) of the Deterministic Network and the End Systems. These UNIs are assumed to be packet-based reference points and provide connectivity over the packet network. The Architecture also mentions internal reference points between the Central Processing Unit (CPU) and the Network Interface Card (NIC) in the End System. The DetNet-UNIs provide congestion protection services and belong to the DetNet Transport Layer.

```
  DetNet                                                       DetNet
End System                                                   End System

   _                                                                _
  / \     +----DetNet-UNI (U)                                      / \
 /App\    |                                                       /App\
/-----\   |                                                      /-----\
| NIC |   v         ________                                     | NIC |
+--+--+   _____    /        \               DetNet-UNI (U) --+   +--+--+
   |     /     \__/          \                               |      |
   |    / +----+    +----+    \_____                         |      |
   |   /  |    |    |    |          \_______                 |      |
   +------U PE +----+ P  +----+             \            _   v      |
       |  |    |    |    |    |              |       ___/ \         |
       |  +--+-+    +----+    |       +----+ |      /      \_       |
       \     |                |       |    | |     /         \      |
        \    |   +----+    +--+-+  +--+PE  |--------         U------+
         \   |   |    |    |    |  |  |    | |     \_      _/
          \  +---+ P  +----+ P  +--+  +----+ |       \____/
           \___  |    |    |    |           /
               \ +----+__  +----+     DetNet-1      DetNet-2
   |            \_____/  \___________/                              |
   |                                                                |
   |      |     End-to-End-Service         |       |         |      |
   <---------------------------------------------------------------->
   |      |     DetNet-Service             |       |         |      |
   |      <-------------------------------------------------->      |
   |      |                                |       |         |      |
```

Figure 1: DetNet Service Reference Model (multi-domain)

A specific hardware is necessary for the time-sensitive functions of synchronization, shaping and scheduling. This hardware may or may not be fully available on a NIC inside the Host system. This specification makes a distinction between a fully DetNet-Capable NIC, and a DetNet-Aware NIC that participates to the DetNet-UNI, but is not synchronized and scheduled with the Deterministic Network.

3.3. The DetNet Stack

The "Deterministic Networking Architecture" [I-D.ietf-detnet-architecture] presents a conceptual DetNet data plane layering model. The protocol stack includes a Service Layer and a Transport Layer and is illustrated in Figure 2.

```
         |   packets going   |             ^   packets coming  ^
         v  down the stack   v             |    up the stack   |
       +------------------------+        +------------------------+
       |         Source         |        |      Destination       |
       +------------------------+        +------------------------+
       |      Service layer     |        |     Service layer      |
       |   -------------------  |        |  --------------------  |
       |       Adaptation       |        |      Presentation      |
       | Packet      | Network  |        | Duplicate   | Network  |
       | sequencing  | Coding   |        | elimination | Coding   |
       | Flow        | Packet   |        | Flow        | Packet   |
       | duplication | encoding |        | merging     | decoding |
       +-------------+----------+        +-------------+----------+
       |    Transport layer     |        |    Transport layer     |
       |  -------------------   |        |  -------------------   |
       |     Encapsulation      |        |      Decapsulation     |
       | Congestion protection  |        | Congestion protection  |
       +------------------------+        +------------------------+
       |      Lower layers      |        |      Lower layers      |
       +------------------------+        +------------------------+
            DetNet End System                  DetNet End System
                    v                                ^
                     \________DetNet Transport______/
```

Figure 2: DetNet-Capable End-System Protocol Stack

3.4. The DetNet Service Model

The "DetNet Service Model" [I-D.varga-detnet-service-model] provides more details on the distribution of DetNet awareness and services.

4. DetTrans Operations

4.1. DetTrans Overview

The DetNet Service Layer mostly operates between the end-points, though it is possible that some operations such as Packet Replication and Elimination are also performed in selected intermediate nodes. The DetNet Transport Layer represents the methods that ensure that a packet is deterministically forwarded hop-by-hop from a Detnet Relay to the next. The term "Transport" in the DetNet terminology must not be confused with the function described in this document. This document defines Detrans as a Layer-4 operation and an IETF Transport Layer; DetTrans provides DetNet End-To-End Services for its Applications, as well as intermediate services in selected points.

Following the DetNet Architecture, DetTrans mostly corresponds to the DetNet Service Layer and its interface with the Detnet Transport Layer for congestion protection services through the DetNet_UNI, as well as for encapsulation and decapsulation services. Compared to a traditional IETF Transport Layer, DetTrans performs similar operation of end-to-end reliability, flow control and multipath load sharing, but differs on how those functionalities are achieved.

Architectural variations are also introduced, for instance:

- Multipath operations are not necessarily end-to-end and a DetTrans function may be present inside the network to relay between N parallel paths and M parallel path, and or perform reliability functionality such as Packet Replication and Elimination.

- The flow control is only needed between the DetTrans Layer and the first Deterministic Transit or Relay Node, for instance a DetNet Router or an IEEE Std. 802.1 TSN Bridge. From that point on, the flow is strictly controlled by the DetNet operation. Architecturally speaking, the flow control does not belong to the DetNet Service Layer but to the DetNet Transport Layer, which means that this specification also defines a sublayer from the DetNet Transport Layer for DetNet-UNI operations.

4.2. Application Requirements

4.2.1. Packet Normalization

A typical SLA for DetNet must be simple, for instance a maximum packet size, and a maximum number of packets per window of time. Smaller packets will mean wasted bandwidth, and excess packets within a time window will be destroyed by the ingress shaping at the first DetNet Bridge or Router.

The way the application layer feed the DetTrans layer may not necessarily match the SLA with the Deterministic Network and in order to provide the expected service, the DetTrans layer must pack the data in packets that are as close to the maximum packet size as possible, and yet make them available for transmission before scheduled time.

4.2.2. Packet Streaming

The DetTrans Layer operates on its own sense of time which may be loosely connected to the shared sense of time in the Deterministic Network.

The DetTrans layer must shape its transmissions so as to ensure that packets are delivered just in time to be injected along schedule in the Deterministic Network.

4.3. Deterministic Flow Services

4.3.1. Deterministic Flows

Deterministic forwarding can only apply on flows with well-defined characteristics such as periodicity and burstiness. Before a path can be established to serve them, the expression of those characteristics, and how the network can serve them, for instance in shaping and forwarding operations, must be specified.

At the time of this writing, the distinction between application layer flows and lower layer flows is not clearly stated in the "Deterministic Networking Architecture" [I-D.ietf-detnet-architecture]. For the purpose of this document, we use the term Determistic End-to-End Service Flow (DEESF), or DetTrans Flow, to refer to an end-to-end application flow, and the term Determistic Service Flow (DSF), or DetNet Flow, to refer to a lower layer deterministic transport. This is illustrated in Figure 3.

```
+---------------+                                      +---------------+
|  Application  |                                      |  Application  |
+---------------+                                      +---------------+
| Service >--------- DetNet End-to-End Service Flow ---------> Service |
+---------------+       +--------+    +--------+       +---------------+
|  Transport    |       |        |    |        |       |   Transport   |
+---------------+       |>-- DetNet Service -->|       +---------------+
| Lower Layers  |       |         Flow         |       | Lower Layers  |
+---------------+       +--------+    +--------+       +---------------+
```

Figure 3: DetTrans vs. DetNet Flows

An application flow is established end-to-end between the DetTrans layers and uses one or more lower-layer deterministic flows either in parallel or in serial modes.

At Application and DetTrans Layers, the characteristics of a flow relate to aggregate properties such as throughput, loss, and traffic shape, and the Traffic Specification (TSPec) is expressed as a Constant Bit Rate (CBR) or a Variable Bit Rate (VBR), burstiness (e.g. video I-Frames), reliability (e.g. five nines), worst case latency, amount of data to transfer, and expected duration of the flow.

At the DetNet Transport Layer (between Relays), metrics are very different, and relate to immediate actions on a packet as opposed to general characteristics of a flow. DetNet Transport Layer characteristics include time sync precision, time interval between packets, packet size, jitter, and number of packets per window of time. This is how the network SLA is defined, but this is not the native terms for the application and a complex mapping must ensure that the path that is setup and the DetNet Transport Layer effectively matches the requirements from the DetNet Services Layer and above.

4.3.2. Deterministic Flow Encapsulation and Stitching

4.3.2.1. Flow Stitching

The DetNet encapsulation and decapsulation of one-in-one, one-in-many and many-in-one Deterministic flows belongs to the DetNet Transport Layer. Direct one-in-one flow stitching also belongs to the DetNet Transport Layer. This happens when a deterministic flow can be directly bridged into another, resource-to-resource, without the need of an upper layer adaptation such as service protection from the Service Layer. A Detnet End-to-End Service flow may be stitched into one Detnet Service flow, or encapsulated in one or multiple Detnet Service flows.

4.3.2.2. Load Sharing

Load Sharing refers to the encapsulation of a DetNet Flow in more than one DetNet flows, for instance using multiple small and more manageable DetNet Service Flows in parallel to carry a large Determistic End-to-End Service Flow, in order to avoid the need to periodically defragment the network. Packets are sequenced at the DetTrans Layer and distributed over the DetNet Transports paths in accordance to their relative capacities. In case of inconsistent jitter and Latency characteristics, packets may need to be reordered at the receiving DetTrans Layer based on the DSF Sequence.

```
+---------------+       +--------+    +--------+       +---------------+
|  Application  |       |>-- DetNet Service -->|       |  Application  |
+---------v-----+   +-------------Flow-------------+   +-----^---------+
|         |     |   |   |        |    |        |   |   |     |         |
|         |     |   |   +--------+    +--------+   |   |     |         |
|         |     |   |                              |   |     |         |
|         | >------- DetNet End-to-End Service Flow -------> |         |
|         |     |   |                              |   |     |         |
|         |   +-----+   +--------+    +--------+   +-----+   |         |
|         +---+ |       |>-- DetNet Service -->|       | +---+         |
|             +-------------------Flow-------------------+             |
|  DetTrans     |   |   |        |    |        |   |   |    DetTrans  |
+---------------+       +--------+    +--------+       +---------------+
     DetNet                 Deterministic                   DetNet
   End System             Routers and Bridges             End System
```

Figure 4: Load Sharing

In order to achieve this function, a Load Distribution function is required at the source and a Re-Ordering Function is required at the destination DetTrans End Point.

4.3.2.3. Flow Aggregation

Flow Aggregation refers to the encapsulation of more than one DetNet flows in one DetNet Flow, for instance using one large and long-lived DetNet Service Flow from a third party provider to carry multiple more dynamic Deterministic End-to-End Service Flows across domains. Packets are sequenced at the DetTrans Layer and distributed over the DetNet Transports paths in accordance to their relative capacities. In case of inconsistent jitter and Latency characteristics, packets may need to be reordered at the receiving DetTrans Layer based on the DSF Sequence.

```
+---------------+         >-- DetNet Service -->        +---------------+
|  Application  |                   Flow                 |  Application  |
+-----v---v-----+       +--------+    +--------+       +-----^---^-----+
|     |   |     |       |        |    |        |       |     |   |     |
|     |   +--------------------------------------------------+   |     |
|     |    >------- DetNet End-to-End Service Flow --------->    |     |
|     +----------------------------------------------------------+     |
|      >----------- DetNet End-to-End Service Flow ------------->      |
|  DetTrans     |   |   |        |    |        |       |    DetTrans  |
+---------------+       +--------+    +--------+       +---------------+
     DetNet                 Deterministic                   DetNet
   End System             Routers and Bridges             End System
```

Figure 5: Flow Aggregation

In order to achieve this function, a multiplexing function is required at the source and a demultiplexing function is required at the destination DetTrans End Point.

4.3.3. Deterministic Service Protection

4.3.3.1. PRE vs. 1+1 Redundancy

The DetNet Flows may also be used for Packet Replication and Elimination, in which case an elimination function is required at the DetTrans Termination.

1+1 Redundancy refers to injecting identical copies of a packet at the ingress of two non-congruent paths, and eliminating the excess copy when both are received at the egress of the paths. Packet Replication and Elimination extends the concept by enabling more than 2 paths, and allowing non-end-to-end redundant paths with intermediate Replication and Elimination points.

4.3.3.2. Network Coding

Redundancy and Load Sharing may be combined with the use of Network Coding whereby a coded packet may carry redundancy information for previous data packet and cover the loss of one, in which case the recovery function is required at the other DetTrans End Point. Network Coding provides a Forward Error Correction between multiple packets or multiple fragments of a packet. It may be used at the DSF layer to enable an efficient combination of redundancy and load sharing.

4.3.3.3. Multipath DetTrans Services

A DetTrans Flow may leverage multiple DetNet Flows in parallel in order to achieve its requirements in terms of reliability and Aggregate throughput. The "Deterministic Networking Architecture" [I-D.ietf-detnet-architecture] clearly states that the capability of Replication and Elimination is not limited to the DetNet End Systems. DetNet Relay Nodes that operate DetTrans but then relay the packets are needed when the DetTrans operations are not end-to-end.

It may be that the DetTrans flow may need to traverse different domains where those Services are operated differently, e.g. controlled by different controllers or leveraging different technologies. It may also be that the bandwidth that is required is only available one segemnt at a time, and that for each segment, a different number of DetNet flows must be setup to transport the full amount of the DetTrans flow.

Figure 6 illustrates an example of the latter case, whereby The DetTrans Flow is distributed over two DetNet Flows, maybe operating PRE, then over three DetNet Flows, for instance operating Network Coding between them but using a smaller banswidth for each flow, and then two DetNet Flows again.

DetTrans is needed at the interconnection points to adapt the flows, recover losses and reinject the appropriate rates in the next segment.

```
+---+-----------+       +--------+    +--------+
| C |Application|   +------------------------------+
| P +-----------+   |   |>- DetNet Transport ->|   |   +---------------+
| U | DetTrans  |   |   +--------+    +--------+   |   | DetTrans  /   |
+---+-----------+   |                              |   |    -------    |
| N | Packet  +------+  +--------+    +--------+   |   |   /  DetTrans |
| I | Queues -+ |       |>- DetNet Transport ->|   +--------+---+---+  |
| C |         +---------------------------------------------+---+---+  |
+---+-----------+       +--------+    +--------+       +----|---|---|--+
                                                            |   |   |
                                                    DetNet  |   |   |
                                                  Transport |   |   |
                                                            |   |   |
+---+-----------+       +--------+    +--------+            |   |   |
| C |Application|   +------------------------------+        |   |   |
| P +-----------+   |   |<- DetNet Transport -<|   |   +----|---|---|--+
| U | DetTrans  |   |   +--------+    +--------+   +--------+---+---+  |
+---+-----------+   |                              +--------+---+---+  |
| N |         +------+  +--------+    +--------+   |   |   \  DetTrans |
| I |       <-+ |       |<- DetNet Transport -<|   |   |    -------    |
| C |         +------------------------------------+   | DetTrans  \   |
+---+-----------+       +--------+    +--------+       +---------------+
  DetNet-Aware               Deterministic               Deterministic
  Host Systems               Transit Nodes                Relay Nodes
```

Figure 6: Intermediate Systems

5. The DetNet-UNI

Figure 7 illustrates a simple example of classical networked devices implementing the DetNet architecture. In that example, applications reside on Host systems and run on main CPUs; DetTrans is collocated with its applications and provides them with a Deterministic Service through DetTrans APIs. NICs provides the connectivity to the Deterministic Routers or Bridges acting at DetNet Edge and Relay Nodes - say as an example that they are IEEE Std. 802.1 TSN Bridges.

```
                             Deterministic
   Host System           Routers and Bridges          Host System
+---+-----------+   |   +--------+    +--------+   |   +---+-----------+
| C |Application|   |---|        |----|        |---|   | C |Application|
| P +-----------+   |   |<- DetNet Transport ->|-------+ P +-----------+
| U | DetTrans  |   |   +--------+    +--------+   |   | U | DetTrans  |
+---+-----------+ <------------ DetTrans ------------> +---+-----------+
| N | Lower     |   |   +--------+    +--------+   |   | N | Lower     |
| I | Layers    |---|   |<- DetNet Transport ->|   |---| I | Layers    |
| C | (queues)  |   |---|        |----|        |---|   | C |           |
+---+-----------+   |   +--------+    +--------+   |   +---+-----------+

                 <-UNI-> <-- DetNet Services -> <-UNI->

     <------------- DetNet End-To-End Services ------------------->
```

Figure 7: Example Physical Network

The DetTrans Layer aggregates the data coming from the application up to a maximum frame size that is part of the SLA with the DetNet Transport. Packets thus formed can be distributed over any of multiple DetNet Transport sessions that are defined to accept that packet size. Packets formed at the DetTrans Layer are queued and ready to be delivered through the DetNet-UNI either with a Rate-Based or a Network-Pull mechanism.

If the NIC is DetNet-Aware then the queue can be offboarded to the NIC and it can be drained with a time gate (Rate-Base) or a message-driven gate (Network-Pull). Else, the queue is handled by the CPU and hopefully it can be drained within an interrupt, either for a timer (Rate-Base) or for a message (Network-Pull).

The DetNet-UNI protocol enables the DetNet transport ingress point to control when the DetTrans Layer transmits its Data packets. It may happen that the DetTrans Layer has not formed a fully-sized packet when time comes for sending it, in which case the packet will be sent with a size below the maximum.

The DetNet UNI uses ICMPv6 to carry its protocol elements. Data Packets across the UNI are encapsulated in order to carry DetNet-UNI control information to identify the reason of a loss or a delay, and determine the action to be taken in case of a packet lost or delayed over the interface.

5.1. Local Loop Flow Control

5.1.1. Dichotomy of a DetNet End System

The logical DetNet End System depicted in Figure 2 comprises several elements which may implemented in one or separate physical Systems. The example dichotomy in Figure 3 segregates ingress shaping and DetNet Relay functions, which are performed by IEEE Std. 802.1 TSN Bridges, from a DetNet-Aware Host.

Hosts and Edge Bridges are connected over Ethernet and together they form a DetNet End System. As it goes, this example introduces a further dichotomy within the Host, between the CPU and the NIC, across a local bus such as PCI, as illustrated in Figure 8.

```
  +-------------+           +-------+                      +-----------+
  | Application |           |  MAC  |                      | Ingress   |
  +-------------+ <--PCI--> +-------+ <---- Ethernet ----> | Shaping   |
  |  DetTrans   |           |  PHY  |                      | and Relay |
  +-------------+           +-------+                      +-----------+
       CPU                     NIC                           IEEE Std.
                  <------------- DetNet-UNI ------------->     802.1
  <-----------  Host System  ------------->                  TSN Bridge
  <-------------------------- DetNet End System ------------------------>
```

Figure 8: Chained Functions

The NICs in the Host System may not participate to the network time Synchronization and may not be aware of the DetNet protocols running between the Deterministic Routers and Bridges, and the associated scheduling rules. In that situation, the DetNet-UNI operates on a Local Loop to ensure that a packet that leaves the Transport reaches the Router or Bridge just in time for injection into the Deterministic data plane and to provide a flow control that avoids congestion loss at the interface.

It is also possible that the NIC participates to the Deterministic Network but still has asynchronous communication with DetTrans Layer running on the the CPU. Either way, a flow control over a local loop must be implemented to drain the queues from the DetTrans layer and feed the network just in time for the deterministic transmission.

Depending on the level of support by the NIC, the loop may be placed on a different interface but remains functionally the same.

5.1.2. Local Loop Location

If the NIC is not aware at all of DetNet, then it is a plain pipe for the Deterministic Traffic. The Local Loop operates between the Edge TSN Bridge and the CPU as illustrated in Figure 9.

```
+---+-----------+           +---+-------+                  +-----------+
| C |Application|           | N | MAC   |                  | Ingress   |
| P +-----------+ <--PCI--> | I +-------+ <-- Ethernet --> | Shaping   |
| U |DetTrans   |           | C | PHY   |    (Bridged      | / Transit |
+---+-----------+           +---+-------+     or P2P)      +-----------+

                  <------------- Local Loop ------------->   Edge 802.1
<-----------  Host System  ------------->                    TSN Bridge
```

Figure 9: DetNet Unaware NIC

If the NIC is fully DetNet-Capable and participates to the deterministic Network including time synchronization and scheduling, then the local loop operates between the CPU and the NIC as illustrated in Figure 10.

```
 +---+-----------+           +---+-----------+              +----------+
 | C |Application|           | N | Ingress   |              |  DetNet  |
 | P +-----------+ <--PCI--> | I + Shaping   | <--DetNet--> |  Transit |
 | U |DetTrans   |           | C | / Transit |   Transport  |   Node   |
 +---+-----------+           +---+-----------+              +----------+
                   <-Local-
                    -Loop ->
 <--------------  Host System  -------------->               TSN Bridge
```

Figure 10: DetNet Capable NIC

If the NIC is DetNet-Aware and does not participates to the deterministic Network including time synchronization and scheduling, then there are two local loops, one that operates between the CPU and the NIC and one that operates between the NIC and the Edge TSN Bridge as illustrated in Figure 11.

```
+---+-----------+           +---+-------+                  +-----------+
| C |Application|           | N | MAC   |    Ethernet      | Ingress   |
| P +-----------+ <--PCI--> | I +-------+ <-- (Bridged --> | Shaping   |
| U |DetTrans   |           | C | PHY   |      or P2P)     | / Transit |
+---+-----------+ <-Local-  +---+-------+                  +-----------+
                   -Loop ->              <-- Local Loop -->    Edge
<-----------  Host System  ------------->                 DetNet Transit
```

Figure 11: DetNet Capable NIC

5.1.3. Network Pull vs. Rate Based Flow Control

The flow control at the DetNet-UNI can take any of two forms:

Network Pull In that Model, the DetNet Edge node drains the DetTrans queue by sending a DetNet-UNI "More" command some estimated amount of time ahead of the scheduled time of transmission for each packet; in case of load sharing, multiple DetNet Edge nodes may drain a queue at their own rates; in case of a high jitter on the UNI Local Loop (e.g. there is a non-deterministic Bridge in between, or the NIC is not DetNet-Aware and the flows suffer from the more erratic response time of the CPU), the DetNet Edge node may need to pull a window of packets to maintain its own transmission queues fed at all times

Rate Based In that model, the NIC is aware of the rate of the deterministic transmission and is drained by its internal timers. Since the NIC is not synchronized with the Deterministic Network, the Bridge uses a DetNet-UNI "Time-Correction" command asynchronously to move forward or backward the next timeout of the NIC for that flow, in order to keep the Rate-Based transmission by the NIC in rough alignment with the scheduled transmission over the DetNet network.

if the NIC is DetNet-Aware, it is expected that it maintains the DetTrans queues in order to provide a deterministic response to the DetNet-UNI, and in that case another control loop between the NIC and the CPU is needed to ensure that the queue in the NIC is always fed in time by the DetTrans Layer; this second loop may be of a different nature than the DetNet-UNI one and may for instance be operated within an interrupt to limit the asynchronism related to message queueing.

5.2. DetNet-UNI Protocol Exchanges

5.2.1. the "More" Message

The "More" message enables a DetNet Transport Edge to pull one packet from the DetTrans Layer in Network-Pull mode. The message is associated with a future transmission opportunity for a packet. The "More" messages are indexed by a wrapping More Sequence Counter (MSC). The Transport Edge also maintains wrapping counters of Successful Packet Transmissions (SPT) and Missed Transmit Opportunities (MTO). The current value of these counters is placed in the "More "message.

Upon reception of a "More" message, the DetTrans Layer, or the NIC on behalf of the DetTrans Layer, sends the next available packet for

that session. The packet is encapsulated and the encapsulation indicates the MSC. This enables the DetNet Transport Edge to correlate the packet with the transmission opportunity and drop packets that are overly delayed.

5.2.2. the "Time-Correction" Message

The "Time-Correction" message enables a DetNet Transport Edge to adjust the timer associated to the DetNet-UNI session in Rate-Based mode. In that mode, the DetTrans Layer sends a packet and restarts a timer at a period that corresponds to the transmission opportunity of the DetNet Transport Edge. If the clock in the CPU drifts, the DetNet Transport Edge will start receiving packets increasingly ahead of expected time or behind expected time. It is expected that the DetNet Transport Edge is protected against a minimum drift by a guard time, but if the drift becomes too important, then the DetNet Transport Edge issues a "Time-Correction" message indicating a number of time units (e.g. microseconds) by which the DetTrans Layer should advance or delay is next time out.

5.2.3. Loss of a Control Message

The loss of a packet beween the DetTrans Layer and the DetNet Transport Edge will correspond to a missed Transmission Opportunity but this does not mean that packets are piling up at the DetTrans Layer. OTOH, if a "More" message is lost, then one packet will not be dequeued and the Detrans queue might grow, increasingly augmenting the latency. It is thus important to differentiate these situations, and in the latter case, discard an extraneous packet to restore the normal level in the DetTrans queue for that session.

In order to do so, the DetTrans Layer maintains the record of the Number of Packets Sent (NPS), that it compares with the variation of the MTO and SPT counters in the "More" message. Upon a "More" message, the DetTrans Layer computes the variation of NPS (dNPS=NPS2-NPS1) and the variation of SPT (dSPT=SPT2-SPT1) since the previous "More" Message . dNPS is typically 1 if the transport always has data to send. Packets in flight when the "More" message is sent are considered lost since they will be received after their scheduled transmission opportunity, so the Number of Packets Losses (NPL) is (NPL=dNPS-dSPT). The DetTrans Layer also computes the variation of MTO since the previous "More" Message (dMTO=MTO2-MTO1). Since a packet loss implies a missed transmission opportunity, there cannot be more packets losses than missed opportunities, so we have dMTO>=NPL. dMTO-NPL represents the number of missed opportunities that are not due to a packet lost or late arrival, thus this is the sub-count of MTOs due to the loss of a "More" message.

For each loss of a "More" message, a packet in the DetTrans queue should be discarded. In order to simplify that operation and outboard it to the NIC, the Transports marks some packets as "Discard Eligible" (DE). A packet can be marked DE if there are enough alternate transmissions of non-DE packets to recover this. For instance, in case of Packet Replication and Elimination only one copy can be marked DE, and the marking should alternate between the sessions to cover a loss on either one rapidly.

6. Security Considerations

The generic threats against Deterministic Networking are discussed in the "Deterministic Networking Security" [I-D.ietf-detnet-security] document.

Security in the context of Deterministic Networking has an added dimension; the time of delivery of a packet can be just as important as the contents of the packet, itself. A man-in-the-middle attack, for example, can impose, and then systematically adjust, additional delays into a link, and thus disrupt or subvert a real-time application without having to crack any encryption methods employed. See [RFC7384] for an exploration of this issue in a related context.

Packet Replication and Elimination of done right can prevent a man-in-the-middle attack on one leg to actually impact the flow beyond the loss of an individual packet for lack of redundancy. This specification expects that PRE is performed at the transport level and provides specific means to protect one leg against misuse of the other.

7. IANA Considerations

This document does not require an action from IANA.

8. Acknowledgments

The authors wish to thank Patrick Wetterwald, Leon Turkevitch, Balazs Varga and Janos Farkas for their various contributions to this work. Special thanks to Norm Finn for being a (if not the) major thought leader to the whole deterministic effort, and for some text that is inlined here from other IETF documents, for the convenience of the reader.

9. Informative References

[AVnu] http://www.avnu.org/, "The AVnu Alliance tests and certifies devices for interoperability, providing a simple and reliable networking solution for AV network implementation based on the IEEE Audio Video Bridging (AVB) and Time-Sensitive Networking (TSN) standards.".

[EIP] http://www.odva.org/, "EtherNet/IP provides users with the network tools to deploy standard Ethernet technology (IEEE 802.3 combined with the TCP/IP Suite) for industrial automation applications while enabling Internet and enterprise connectivity data anytime, anywhere.", <http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00138R3_CIP_Adv_Tech_Series_EtherNetIP.pdf>.

[HART] www.hartcomm.org, "Highway Addressable Remote Transducer, a group of specifications for industrial process and control devices administered by the HART Foundation".

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-03 (work in progress), August 2017.

[I-D.ietf-detnet-problem-statement]
Finn, N. and P. Thubert, "Deterministic Networking Problem Statement", draft-ietf-detnet-problem-statement-02 (work in progress), September 2017.

[I-D.ietf-detnet-security]
Mizrahi, T., Grossman, E., Hacker, A., Das, S., Dowdell, J., Austad, H., Stanton, K., and N. Finn, "Deterministic Networking (DetNet) Security Considerations", draft-ietf-detnet-security-00 (work in progress), October 2017.

[I-D.ietf-detnet-use-cases]
Grossman, E., Gunther, C., Thubert, P., Wetterwald, P., Raymond, J., Korhonen, J., Kaneko, Y., Das, S., Zha, Y., Varga, B., Farkas, J., Goetz, F., Schmitt, J., Vilajosana, X., Mahmoodi, T., Spirou, S., Vizarreta, P., Huang, D., Geng, X., Dujovne, D., and M. Seewald, "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-13 (work in progress), September 2017.

[I-D.varga-detnet-service-model]
Varga, B. and J. Farkas, "DetNet Service Model", draft-varga-detnet-service-model-02 (work in progress), May 2017.

[IEC62439]
IEC, "Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR) - IEC62439-3", 2012, <https://webstore.iec.ch/publication/7018>.

[IEEE802.1TSNTG]
IEEE Standards Association, "IEEE 802.1 Time-Sensitive Networks Task Group", 2013, <http://www.ieee802.org/1/pages/avBridges.html>.

[ISA100.11a]
ISA/IEC, "ISA100.11a, Wireless Systems for Automation, also IEC 62734", 2011, < http://www.isa100wci.org/en-US/Documents/PDF/3405-ISA100-WirelessSystems-Future-broch-WEB-ETSI.aspx>.

[ODVA] http://www.odva.org/, "The organization that supports network technologies built on the Common Industrial Protocol (CIP) including EtherNet/IP.".

[Profinet]
http://us.profinet.com/technology/profinet/, "PROFINET is a standard for industrial networking in automation.", <http://us.profinet.com/technology/profinet/>.

[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>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[WirelessHART]
www.hartcomm.org, "Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART - IEC 62591", 2010.

Author's Address

Pascal Thubert (editor)
Cisco Systems, Inc
Building D (Regus) 45 Allee des Ormes
MOUGINS - Sophia Antipolis
FRANCE

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com