Determining the Future

Norman Finn

nfinn@alumni.caltech.edu

Introduction

This author is attending the September, 2016, interim meeting of IEEE 802.1 as a private individual, with no support from any company.

These contents of this presentation are personal opinions, based on the author's passionate personal interest in making Deterministic Networking a successful technology.

I will present my opinion of likely subjects for the next round of TSN / DetNet standards.

We've made a good start!

1. Low- and Zero-congestion loss queuing techniques
2. Transmission preemption
3. Topology discovery
4. Fixed-path establishment over bridges and routers
5. Serial numbering of packets for replication and elimination
6. Resource reservation via peer-to-peer protocols or central control

Two big classes of issues to work on

Scaling up and down

1. Scaling up the data plane
   a) Aggregate flows within enclosing flows
   b) Mitigate the burden of per-flow state
   c) Applications are dynamic, not static
2. Scaling down the data plane
   a) Support 2-port devices that are very, very simple
   b) Ensure that hop count doesn't scale out of bounds
   c) Work with routers as well as bridges
3. Integrating DetNet islands with the enterprise
   a) We must not reinvent connectivity

Security

4. Enabling security choices
   a) Authentication and encryption choices
   b) Discovery mechanisms
5. Enforcing security decisions
   a) Generalized model for characterizing allowed flows (not just deterministic flows)
   b) Enforcement of flow characteristics
   c) Provision DetNet with same mechanism as security

And, don’t forget wireless media

(Not my strong suit)

Scaling

What you mean, “we,” Kemosabe?

The author makes a lot of statements about “We need to …” in this presentation. “We” are a number of groups:

IEEE 802, IETF DetNet, and others are creating high-level standards that can be used in arbitrary combinations to achieve diverse goals.

Industry vertical associations, equipment vendors, and large users are creating more specific profiles of these standards to enable the design of specific classes of networks meeting narrower goals.

Vendors and users are putting networks and devices to work in specific applications.

1. Scaling up the data plane

- There is a fundamental conflict between DetNet and enterprise networking that must be addressed.
- The emphasis, for enterprise networking, has long been to reduce the amount of state required in the core of the network by means of address aggregation (802.1ah, LISP, Ether-over-XYZ, IP-over-XYZ, …)
- DetNet has been assuming that we add state for every DetNet flow in the core of the network.
- Adding state could easily lead us to the SDN (Software Defined Networking) fantasy world where there are more control packets, and CPU cycles consumed, than data packets delivered, which would be bad.

1. Scaling up the data plane

- In small networks, we **can** add resources for per-flow state. The DetNet WG is working on standard mechanisms for L2, L3, and above, to:
  a) Export the flow ID and/or serial number to the outermost layers of the packet
  b) Look inside the packet to obtain the flow ID and/or serial number
  c) Perform per-flow forwarding and per-flow resource allocation
- Enterprise wide, we need new standard mechanisms at L2, L3, and above to:
  a) Aggregate flows within enclosing flows
  b) Shape flows at the entrances to and exits from the aggregations
  c) Define additional queuing technologies to make aggregation actually yield zero congestion loss for aggregated flows*

* I've thought a lot about this one – CQF is on the right track, but we're a long way from the useful aggregation of DetNet flows.

1. Scaling up the data plane

- So far, TSN is static. In fact, our world is very dynamic:
  a) Machines (Applications) can be turned on and off
  b) They switch between operating in Mode A and Mode B
  c) Paths and their reservations need to be moved, while they are running, without significantly increasing the chances of packet loss
  d) It takes time – too much time – to set up connections, make reservations, etc.
- We need standard mechanisms at L2, L3, and above to:
  a) Differentiate between potential and running reservations
  b) Define sets of reservations that can operate only at different times
  c) Support requests to turn on and off sets of pre-made reservations

2. Scaling **down** the data plane

- Chains of two-port devices modeled on the Two-Port MAC Relay (TPMR) or a 3-port Bridge (one to a Station) have advantages and disadvantages:
  a) Cable length and weight can be dramatically reduced **(+)**
  b) Port count, size, and weight of forwarding systems can be reduced **(+)**
  c) Hop count, and thus latency, can be dramatically increased **(-)**
- To enable TPMRs, we need standard mechanisms at L2, L3, and above to:
  a) Shift computing tasks (topology discovery, best-effort topology control, address migration, bandwidth reservation, etc.) from the simple devices to the bridges and routers.
  b) Provide queuing and/or software mechanisms to mitigate the hop count / latency problem

3. Integrating DetNet islands with the enterprise

- The history of packet networks in our space of interest makes integration challenging:
  a) Control networks have been small and physically isolated from the rest of the world
  b) So “anything goes”: duplicate MAC or IP addresses, extensive NAT, non-standard use of spare bits in packet formats, assumptions about or restrictions upon network topology, L2-only application protocols, dependence upon broadcast packets, no thought for security, …
- To integrate, we need standard mechanisms at L2, L3, and above to:
  a) Provide Platform and Function identity mechanisms that can support the integration of these diverse badly-behaved islands (more about Platforms and Functions, later)
  b) Determine the network topology when no one topology protocol spans the enterprise
  c) Maintain the orthogonality of DetNet and ordinary networking as far as possible, and define the necessary dependencies where necessary

Security

What is going on?
What should be going on?

- 10^n devices, the majority of them cheap = stupid
- 100s or 1000s of local application- or domain-specific controllers
- 10^{2n} two-way possible interconnections
- 1000s of smart phones and tablets
- Actors are being created and destroyed daily. Some are mobile.
- Each connection is not simply allowed or not. There are 10^{lots} of potential security decisions.
  - Only certain kinds of information (statistics? commands?) are allowed on certain connections
  - I want the maker of the lathe to collect reliability statistics, but not to know what I'm making, today.
  - Only some connections need/want/are allowed to have TSN QoS.

Ask the hard questions, first. Then, ask about TSN reservations.

- I am the Network Administrator, the person responsible for the network's usefulness.
- The network covers a big factory.
  - This network is much too complex for me to understand.
  - In any simplistic (i.e. 802.1) paradigm, my automatic minions ask me far too many questions that I don't know the answer to. (Should I allow this program to install this update? To have a TSN reservation between these two devices?)
  - My human users may have detailed information about their homemade applications, but they don't really understand in detail the stuff they bought off the shelf.
- Saying that "the network will be centrally controlled" is not an answer, if the central controller must ask a human questions that the human cannot answer.
- So, let's look at the "I know neither what is nor should be going on" problem, and THEN look at TSN.

A (loosey-goosey) model for deciding

- An Application is a set of logical Functions (network management master brain, sensor, machine controller, statistics collector, call-home responder) residing on Platforms (sensors, controllers, iPads, routers, cloud).
- An Application may have multiple Functions on a single Platform.
- Functions need Connections (data paths, bandwidth, TSN QoS, frequencies, bursts) among them to do their jobs. Connections may be inter-Application or intra-Application.
- An Application is obtained from a Store.
- An Application has a Description, obtained from the Application or the Store, with information about the Application's Functions and Platforms and its possible Connections to its own and other Applications' Functions.

A (loosey-goosey) model for deciding

- The Network Administrator, to some degree (more to come) trusts:
  - An Application
  - A Store that provides Applications and their Descriptions.
  - The Platforms that make up the network.
  - Users to make decisions.
- The Network Administrator assigns levels of trust (# of instantiations, TSN allowed, not all Functions permitted, Platform/Function restrictions, User IDs, …), and thus limits, to Applications based on their Descriptions and the planned usage of the network.
- Untrusted Applications have limited connectivity or platform access. Certain Applications on a Platform may invalidate its trust for other, critical Applications. This is all standard IT stuff.
- Within those limits, Users and Applications are free to come and go.

Descriptions

- The Description of an Application includes a Connection model that the Network Administrator's automatic minions need to:
  - Decide whether (*m* instances of) this Application can be supported on the network
  - Program other Applications (e.g. bridge and routers) to police the Application's Connections
  - Assign Functions to Platforms and network resources to Connections
  - Support TSN requirements
  - Help the Application create Connections among instances of Functions on Platforms
  - Characterize the Platforms that are eligible to run an Application's Functions, and what other Applications can or cannot share that Platform
- This information may be parameterized (e.g. by number of stations or user-chosen cycle frequency)

Trusting Stores

- The Network Administrator trusts the Store to supply Applications and Descriptions
- The NA verifies the credentials of Applications, Functions, and/or Platforms with the Store
- The NA needs to be satisfied of the validity of a Platform ← Store download

Trusting Applications

- Remember the "10**bunches**" problem? The Network Administrator trusts the Application to put the right data on the right Connection to/from the right Function.
  - I trust the maker of the lathe to send the preventative maintenance data from the PM Collection Function (on the machine) to the Call-Home Local Collector Function (in my enterprise data center), and **not** send (or at least, for the lathe maker to not interpret at such) data about what I'm making.
  - That is, the NA trusts the Applications to have primary responsibility for "kinds of data sent" security.
  - The network performs its secondary responsibility by enforcing the Descriptions' limits.
- In return, the NA needs to help the Application: For example, to provide the right environment for the Application, and/or help the Application, to establish connections among the instances of its Functions on Platforms.
- Platforms (or Platforms' Security Applications) will have limitations on what Application Functions they trust.

Trusting Platforms

- The Network Administrator trusts the Platform to ensure that Applications don't have illegal conversations. That is, Connections can be intra-Platform as well as inter-Platform, and are subject to the same bandwidth/burst/Function ID, etc. limitations as inter-Platform Connections.
- Devices can be limited to certain combinations of Applications/Functions. Simple devices can be limited to a single Function.
- IT departments, today, use Security Applications to help enforce these things. Devices too simple for this approach get more limitations, of course. Platforms can come from Stores and can be verified, as well.
- Applications' Functions will have requirements for what Platforms they trust. Applications often bring their own Platforms to the network with the Application.
- Of particular interest to the NA are what data flows (if any) that a given Platform is allowed to participate in that do **not** have Descriptions. This is critical to stopping intrusions.

Downloading

- The Network Administrator needs to be sure that a Platform is running unadulterated copies of specific Functions. Again, a Security Application helps, here.
- Please note that this is one of the most difficult problems to solve, and a major area where we (IEEE, IETF) can be of service.

Call Home

- Ultimately, a Call Home model may be the best means for initial discovery of Functions. Broadcasts and Bonjour Servers will also be used. All techniques have limitations.
- For Call Home scenarios, associating the sensors of a machine with that machine's controller, when the controller resides in the cloud, may ultimately depend on a username/password account with the manufacturer, and the manufacturer's listing of serial numbers of associated Platforms or sets of devices bought by that customer.
  - Lots of room for games, here, involving cell phones with manufacturers' Applications introducing Functions and Platforms to each other.
  - The Description is important for permitting and limiting the Call Home function.
- There are some fundamental identity issues to be solved, and IEEE 802 may be able to help.
  - The network operates in terms of L2/L3/L*n* addresses, which are often not unique (local MACs, NAT)
  - Unique Function or Platform identities may or may not be tied to a MAC address used on the wire
  - So, connecting Application/Description information to Call Home info to the Network is a problem

It's not just TSN/DetNet traffic that must be monitored, policed, etc.

- The network must support enforcing the limitations set by the Network Administrator using the Descriptions
- In general, the network must classify all flows and enforce connectivity, bandwidth, and similar limitations on all of them, in order to have a network that is moderately trustworthy.
- The network must do a good job of showing the usage of all TSN reservations so that:
  - Network growth can be planned
  - Resources can be shifted to needed places
- IEEE 802 may be able to help (and certainly IETF can help) by defining a useful set of parameters describing Connections that are give constraints that are:
  - Loose enough to not give false indications of misbehaviors
  - Tight enough to detect a useful class of misbehaviors

In Summary

Scale up. Scale down. Integrate with the enterprise.

It's all about who you trust. The Network Administrator grants privileges to allow automatic configuration insofar as he or she trusts the source (the Store) of the Application, its Description, the Platforms on which the Applications run, and the Users who control them.

TSN/DetNet reservations are a relatively minor part of the Descriptions that support both automatic configuration, and automatic network supervision.

If all this sounds familiar …

- This model is, to some degree, followed in BYOD enterprise networks today.
- If there is an Internet of Things (IoT) framework out there that more-or-less matches this model, let's seize it!
- If not, let's start one!
- In IEEE 802.1 and in IETF, we need to supply the pieces required for this model to work.