Stream Control Transmission Protocol (SCTP) in Container Environments

Michael Tüxen (tuexen@fh-muenster.de)
Motivation

• SCTP based services are going to be deployed in container based environments.
• Solutions currently being looked at are influenced by solutions used for TCP based services. This does not cover multihoming.
• Depending on the requirements, a simple solution might be sufficient.
• Not being bound to a particular container technique.
Architecture A

```
+-------------------+-------------------+-------------------+-------------------+
|                   |                   |                   |                   |
|                   |                   |                   |                   |
|Cluster            |Cluster            |Cluster            |
|Element            |Element            |Element            |
|  1                |  2                | ...               | N                  |
+-------------------+-------------------+-------------------+-------------------+
```

```
Distributor

+-------------------+-------------------+-------------------+-------------------+
```

```
Network

\/
```

```
+-------------------+-------------------+-------------------+-------------------+
|                   |                   |                   |                   |
|                   |                   |                   |                   |
|                   |                   |                   |                   |
+-------------------+-------------------+-------------------+-------------------+
```

```
+-------------------+-------------------+-------------------+-------------------+
|                   |                   |                   |                   |
|                   |                   |                   |                   |
|                   |                   |                   |                   |
```

```
+-------------------+-------------------+-------------------+-------------------+
```

```
Distributor

+-------------------+-------------------+-------------------+-------------------+
```

```
+-------------------+-------------------+-------------------+-------------------+
```

```
Distributor

+-------------------+-------------------+-------------------+-------------------+
```
Architecture A using NAT

- Distributors are NAT functions
- How to distribute incoming connection request?
- How much state is held?
- Requires specific code in endpoints and NAT.
- Scalability and fault tolerance.
Architecture B
Architecture B using SCTP Proxy

• Distributors are terminating SCTP associations.
• How to distribute incoming connection request?
• Requires no specific code in endpoints and proxy.
• Scalability and fault tolerance.
Architecture C
Architecture C using a Protocol Extension

• The client sends an INIT chunk with a new parameter (INIT-FORWARDING-ALLOWED) to one of the distributors.
• The distributor selects a cluster element, replaces the INIT-FORWARDING-ALLOWED parameter with another new parameter (INIT-FORWARDED) containing the address of the distributor and sends the INIT chunk to the selected cluster element.
• The cluster element processes the INIT chunk and sends an INIT-ACK back to the client. It copies the INIT-FORWARDED parameter in the INIT-ACK chunk.
• After the client receives the INIT-ACK, the client communicates directly with the selected cluster element.
Pros an Cons

• Pros
  – The distributor only needs to handle connection requests.
  – Easy to implement (at the endpoints and the distributor).
  – Very good scalability and fault tolerance.

• Cons
  – Each cluster node needs IP addresses, which can be used to communicate with the peer.
  – The IP addresses of the cluster nodes are exposed to the peer.