Dyncast Architecture
(dynamic anycast)

draft-li-dyncastr-architecture

Yizhou Li; Luigi Iannone; Dirk Trossen; Peng Liu; Cheng Li
Overview of Dyncast service model
Terms definition

- **Service**: a monolithic functionality that is provided by an endpoint according to the specification for said service. A composite service can be built by orchestrating monolithic services.

- **Service instance**: a running environment (e.g., a node) that makes the functionality of a service available. One service can have several instances running at different network locations.

- **D-Router**: Dyncast Router, a node supporting Dyncast functionalities as described in this document.

- **D-MA**: Dyncast Metric Agent, a Dyncast specific agent able to gather and send metric updates but not performing forwarding decisions.

- **Dyncast Service Identifier (D-SID)**: An anycast address identifying a service in Dyncast. All service instances running the same service are identified by the same D-SID.

- **Dyncast Binding Identifier (D-BID)**: A unicast address to reach a service instance for a given D-SID. The same Service instances running on different nodes are bound to node identifier D-BID.
**Dyncast architecture: Distributed and Centralized modes**

**Distributed mode:**
- Dyncast nodes are aware of status of computing resources
- Metric will be distributed among the Dyncast capable nodes

**Centralized mode:**
- CRMP collects status of computing resources and send to network controller
- Network controller compute the forwarding policy based on the network information and computing resource status information.
• **Dyncast Metric Agent (D-MA):** A Dyncast specific agent able to gather and send metric updates but not performing forwarding decisions. May run on a D-Router, but it can be also implemented as a separate module (e.g., a software library) collocated with a service instance.

• **D-Router:** A node supporting Dyncast functionalities. Namely it is able to understand both network-related and service-instances-related metrics, take forwarding decision based upon and maintain instance affinity, i.e., forwards packets belonging to the same service demand to the same instance.
In order to supporting Dyncast, a metrics associated to D-SID and D-BID should be calculated and advertised. The Service instance and metrics mapping shows below:

<table>
<thead>
<tr>
<th>D-SID</th>
<th>D-BID</th>
<th>Metrics</th>
</tr>
</thead>
</table>

- 2001:db8:A:A is an anycast IPv6 address identifying a service
- 2001:db8:1:1 is a unicast IPv6 address identifying a service instance
- X is a metric, it can be a number or a tuple with multiple numbers. This will be discussed in the future.
Dyncast metric distribution

Service/Metrics Information
- (D-SID 1, D-BID 21, <metrics1>)
- (D-SID 2, D-BID 22, <metrics2>)

Service/Metrics Information
- (D-SID 1, D-BID 31, <metrics3>)
- (D-SID 2, D-BID 32, <metrics4>)
Flow handling: routing based on Dyncast metrics

- **N_s**: Network Metrics (congestion, latency,...)
- **I_s**: Instance Service Metrics (load, resources available, ...)
Potential work: Flow Affinity & Binding Table

- Flow affinity is one of the critical features that Dyncast SHOULD support
  - Group all the packets for a single service request to go to the same destination.
- Flow binding table allows to determine the most appropriate egress and service
- A flow entry in the flow binding table can be identified using the classic 5-tuple value
  - different services may have different granularity of flow identification

![Flow Identifier Table]

Figure. Example of a Binding Table in Ingress D-router

Note: The discussion of affinity is still open since last side meeting, we need to work on it more.
Comments are welcome

Any questions or comments?
Thanks