Automatic attachment of end stations and network devices

Dan Romascanu

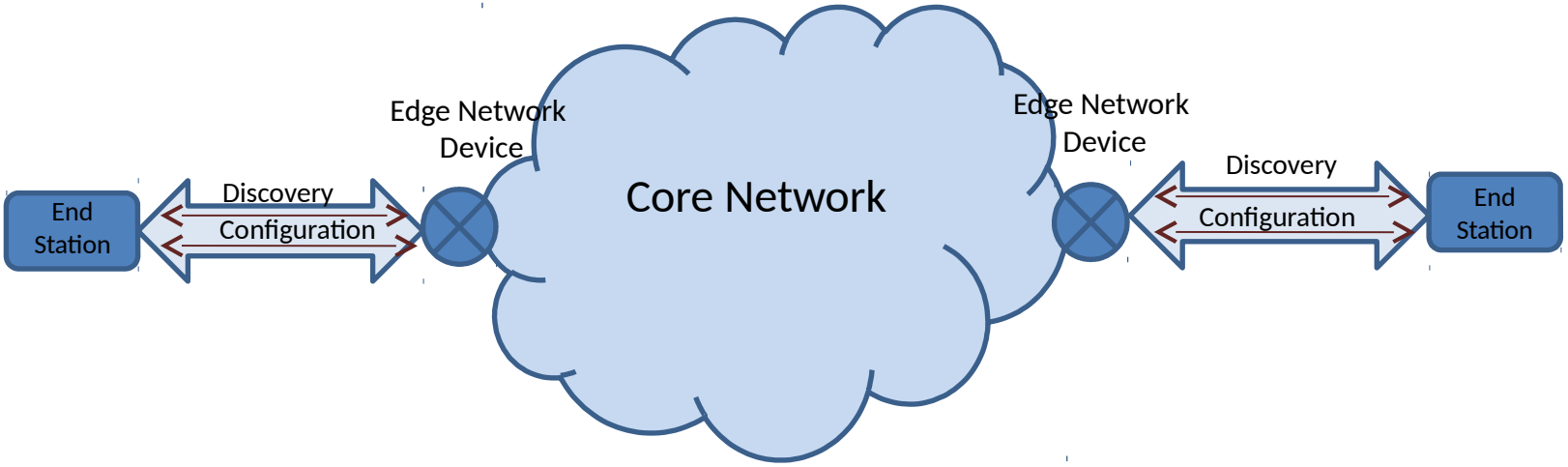
Paul Unbehagen

(draft-romascanu-opsawg-auto-attach-framework-00)

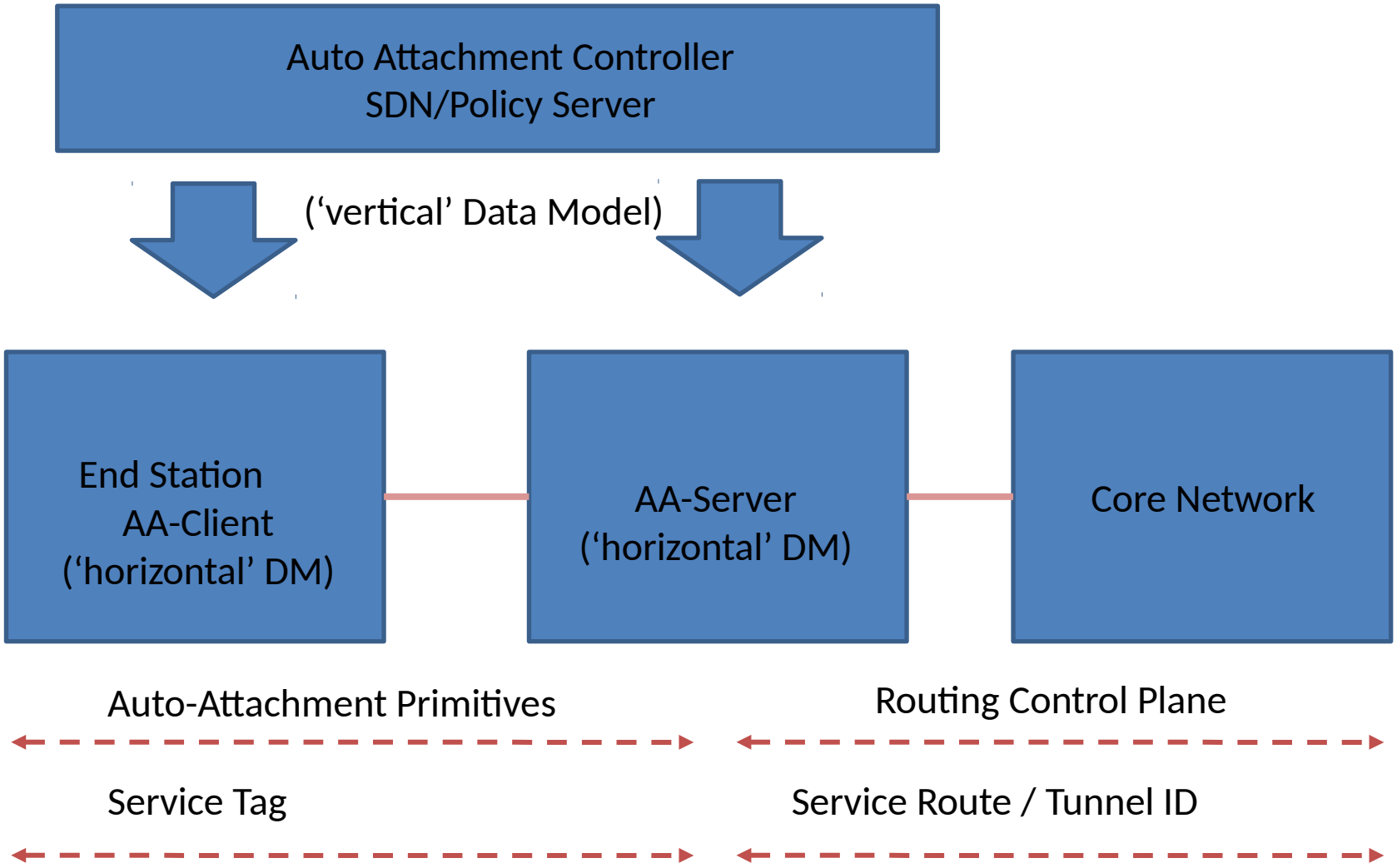
(draft-romascanu-opsawg-auto-attach-lldp-00)

The Auto-Attachment Concept

- Simplify operations procedures by automating the configuration of network devices and end-stations to individual services in a core network
- Run a simple one-hop / two steps protocol that
  - Discovers and identifies end-stations to edge devices
  - Configures the link and connects the station to the available core network resource (L2VPN, MPLS, etc.)



Conceptual Auto Attach Model



Changes from IETF 95

- Splitting draft-unbehagen-lldp-spb-02 into two separate I-Ds
  - draft-romascanu-opsawg-auto-attach-framework – generic framework
  - draft-romascanu-opsawg-auto-attach-lldp-00 – the LLDP auto-attachment solution in an SPB network
- Clarification of terminology
- Functional description of auto-attachment processes
  - Element discovery
  - Services configuration
- Protocols instantiating the Discovery Process
  - LLDP (a.k.a. IEEE 802.1AB)
  - Protocols that implement the MUD Framework (the MUD URL DHCP Option and the MUD URC X.509 Extension)
- Alignment with the evolution of IEEE 802.1Qcj (now at D0.2)

Comments

- 1. ‘The AA controller has connection to both AA client and AA server … it’s not practical in many cases. AAC and AAS are usually in different administrative domains. For example, AAC installed on a CE, and the AAS installed on the PE. Usually, CE and PE are controlled by different domain controllers. It’s hard to deploy a super controller for CE and PE. Another example is, as you showed, the IoT. It’s not practical to let all the end devices connect to the AA controller. The suggestion is to have the AA controller only connect to the AAS. AAC can pull additional information from the path: AA controller->AAS->AAC
- 2. If both AAC and AAS connect to the AA controller, that means the AAC and AAS can be synchronized and configured by the management plane. Some of the control plane function between AAC and AAS can be reduced

Lack of clarity in the framework description that needs to be corrected. The AA controller in the diagram should not be regarded as a ‘super-controller’ across administrative domains, but rather as a policy function that applies to the network and may be instantiated in different ways. The ‘vertical’ data model (6) in the diagram does not carry the same information to AACs and AASs.

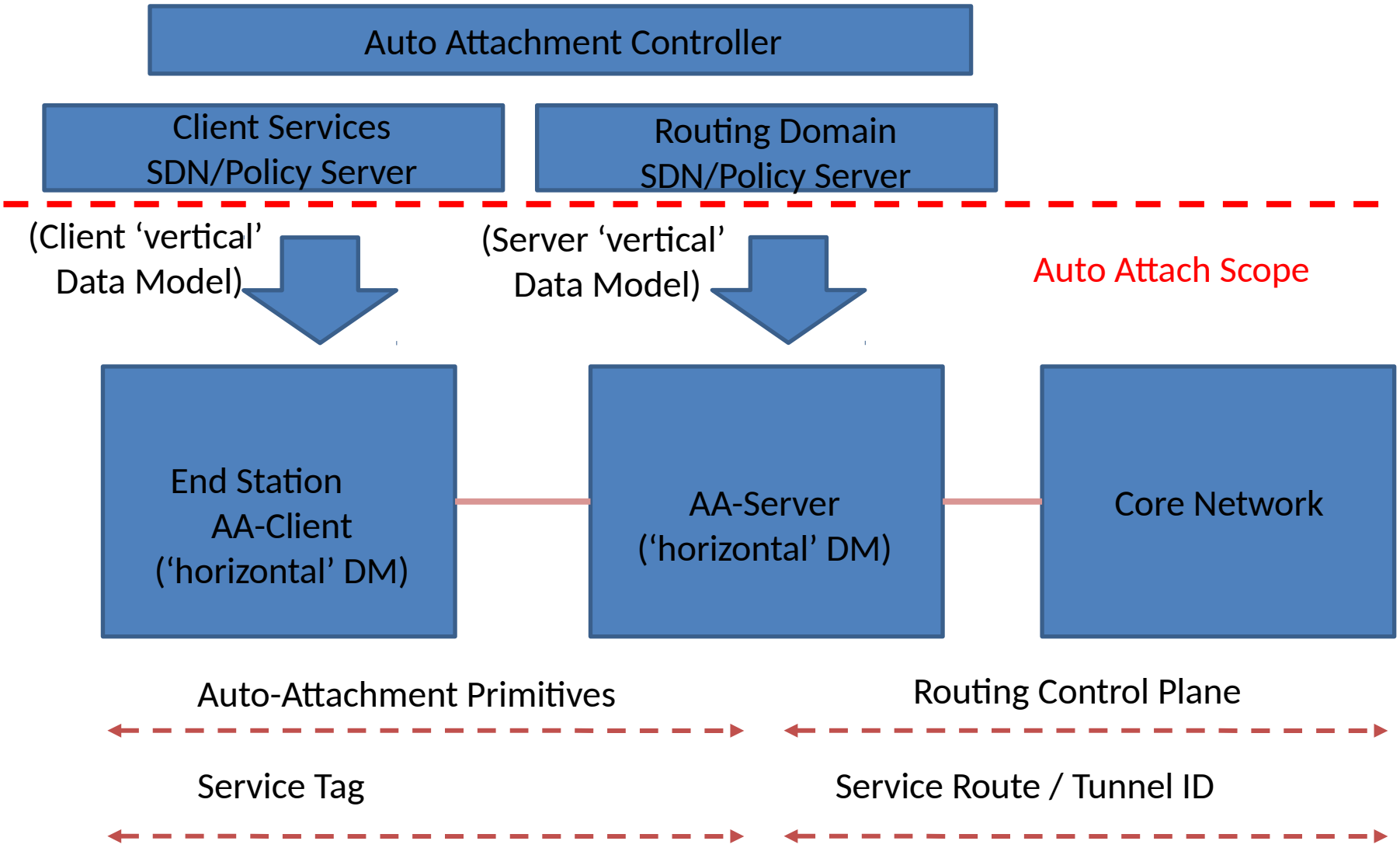
Two use cases:

- *The AA controller connects only to the AAS and configures the mapping policies on the network devices. Clients are configured mainly by the horizontal interaction – this will be true also in most of the IoT scenarios*
- *Both AAS and AACs connect to the AA controller function. The clients get authentication and local security and local services information from the controller (what services are available, how you configure to them). Servers perform same functionality as in #1. The implementation of the controller may be distributed, possibly with an ‘orchestration’ function in-between. This fits for example the BYOD use cases.*

- 3. Possible new use case: cloud VPN solution. The customer provisioned virtual private cloud can automatically join (attach to) the existing enterprise VPN.

To be considered. Need to assess the changes required by a client that is virtualized. Also look into I2NSF work in this space.

Evolution of the Conceptual Auto Attach Model



Status and Next Steps

- Status of the project
  - L2 implementation using extensions of the Link Layer Discovery Protocol (LLDP – IEEE 802.1AB) and core IEEE 802.1aq (Short Path Bridging) networks
  - Deployed
  - Open source code available in ODL, OVS
    - https://github.com/auto-attach/aa-lldpd
    - https://github.com/auto-attach/aa-sdk
    - http://openvswitch.org/releases/openvswitch-2.4.0.tar.gz
  - IEEE 802.1 / LLDP extensions defined in the IEEE 802.1Qcj project
- Next Steps
  - YANG data model for policy and service configuration
  - I-D describing other protocols instantiations (MUD, …)
  - Possible OPSAWG work item if there is interest and willing contributors

Reading List

- https://datatracker.ietf.org/doc/draft-romascanu-opsawg-auto-attach-framework/
- https://datatracker.ietf.org/doc/draft-romascanu-opsawg-auto-attach-lldp/
- http://www.ieee802.org/1/pages/802.1cj.html