MS2.1.1: CoCo System Architecture

Version 1.4

Rudolf Strijkers, TNO
Ronald van der Pol, SURFnet
Marijke Kaat, SURFnet

1. Introduction

The Community Connection (CoCo) service is a prototype for on-demand multi-domain multipoint L2/L3 VPN instances. The prototype will be build on top of an OpenFlow infrastructure. End-users use a web portal to setup CoCo instances. Typical users are research communities that form a closed user group and that want their e-science resources (servers, VMs, laptops, storage, instruments, etc.) interconnected, but reachable for their closed group only.

The prototype consists of several parts. The core of the network is based on MPLS label forwarding and is implemented over several domains. Each domain runs a web portal for its own end-users. End-users and resources are situated at university campuses, but mobile users will also be able to connect to a CoCo instance via an OpenVPN based client.

2. Definitions

CoCo: Community Connect

CoCo Network: A network that can offer CoCo instance services

CoCo Instance: A CoCo style private network. There can be multiple CoCo instances running at the same time on the CoCo network

CoCo Agent: A stand-alone program that manages a CoCo network within one domain and can offer multiple CoCo instances. The southbound interface talks to the network, the northbound interface talks to the CoCo Service Agent.

CoCo Service Agent: The CoCo orchestrator that manages multi-domain CoCo instances. Its southbound interface talks to the CoCo agents in the various domains. Its northbound interface is a web portal where users can login and setup and tear down CoCo instances.

3. CoCo Design Choices

A couple of design choices have been made in order to make CoCo scalable and deployable. We have based many of the CoCo concepts on RFC 4364 (BGP/MPLS IP VPNs). The amount of forwarding entries, especially in the core, needs to be kept as small as possible in order to not run out of forwarding table space. We have chosen to use MPLS labels to aggregate traffic. E.g. all traffic between each pair of provider edge (PE) switches (see Figure 2) is encapsulated with the same MPLS label. Optionally, a second backup path (or even multiple paths) between each pair of PE switches can be configured. By doing so, the switches in the core (P) forward based on that label only.

The CoCo agents of each domain are responsible for configuring the correct forwarding entries on the core switches of their own domain. Not all domains need to use the same MPLS labels. The CoCo agents exchange information about the MPLS labels used with their peers. This is something like “when I send traffic on the primary path to provider edge switch X, I use MPLS label Y”.

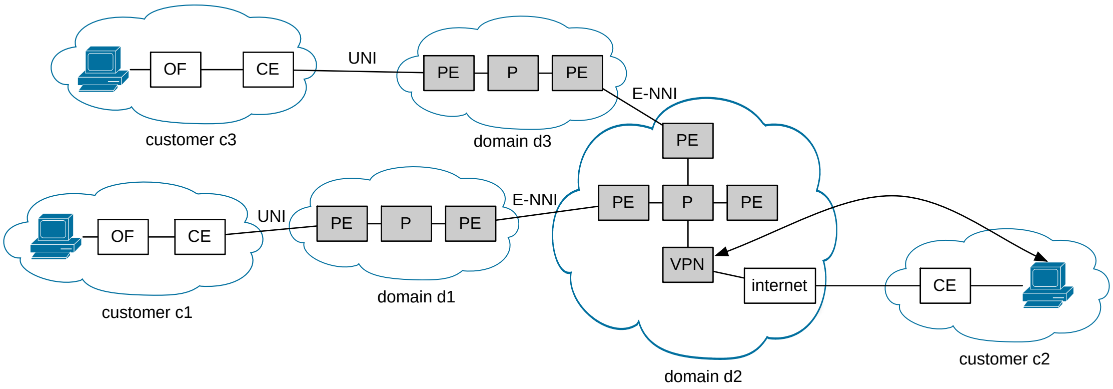

Figure 1 Interdomain Data Plane

Another design choice is to map each CoCo instance to an MPLS label that identifies the instance. This is the second label in the MPLS label stack. This mapping also needs to be exchanged between domains. This is information like: “for CoCo instance X we use MPLS label Y”.

At the PE switches traffic needs to be encapsulated with the correct destination PE and CoCo instance MPLS labels. For L3 service, this is done based on IP prefixes. For L2 service, this is done based on MAC/IP tuples. A L2 service needs additional features like MAC learning at the edges and handling of broadcast and unknown traffic. These features are not needed for the L3 service. That’s why we have chosen to start with implementing the L3 service and if time allows we will add the additional L2 features. This can be done as enhancement of the L3 service because the core MPLS based forwarding stays the same.

When an end node at a site is added to an instance, the CoCo agent (with the web portal that is used to add the site) announces that information to its peers. This announcement contains information about the identifier of the CoCo instance and the prefixes used at the site for that node in that instance.

4. CoCo Core Network Data Plane Forwarding

MPLS based forwarding is used in the core of the network in order to keep the forwarding tables small as described in section 3. Provider edge (PE) OpenFlow switches take care of encapsulating the user traffic received from customer edge (CE) equipment with the proper MPLS tags. When sending traffic from the backbone to the CE the PE removes the MPLS tags. Two MPLS labels are used. The outer MPLS label is used to identify the PE to which a frame must be sent. The inner MPLS label is used to identify the CoCo instance. A site can be present in multiple CoCo instances at the same time.

The CoCo network core consists of *PE* and *P* OpenFlow switches as is shown in Figure 1. The P switches are internal core network switches. The PE switches connect to either customers (via UNI interfaces) or other domains (via E-NNI interfaces). The CoCo agents are responsible for topology discovery within a domain and do intra- and inter-domain path calculation. The intra-domain path calculation is based on the domain topology only. The inter-domain path calculation is based of BGP path information that is exchanged with neighbor domains. In the inter-domain case the CoCo agent configures a path from a PE with UNI ports to the PE that is connected to the E-NNI port to the inter-domain link that has been chosen by the BGP path selection process. There can be one or more paths between each pair of PEs. For simplicity, we start with one shortest path between each pair of PEs.

5. Encapsulation and decapsulation at the UNI interfaces

At the edges of the domain on the UNI interfaces towards the CE of the customer encapsulation and decapsulation takes place. We use VLAN based port services on the UNI interfaces. The VLAN ID maps to a particular CoCo instance. The customer is responsible for putting traffic of nodes in the correct VLAN. The P switches match on the VLAN ID and pop the VLAN header. Before forwarding the packet the P switches add the two MPLS labels corresponding to the destination PE and CoCo instance. The CoCo agent that install the flow forwarding rules on the PE switches need to know what is the destination PE for each IP prefix. The CoCo agent learns this by running BGP and exchanging BGP/MPLS IP VPN information (RFC 4364). The CoCo agent also need to know about the mapping between customer VLAN ID and CoCo instance and the IP prefixes that the customer uses in that CoCo instance. In the first implementation of the CoCo prototype, the person adding a site to a CoCo instance configures this manually on the web portal.

6. Setting up and tearing down a CoCo instance via the web portal

Each domain runs a web portal as is shown in Figure 2. This portal is used to setup and tear down CoCo instances and to retrieve information about active CoCo instances. The user connects to the web portal in his own domain and authenticates himself. He can now perform actions for which he is authorized. The possible actions are:

1. Create CoCo instance

   Allowed by: superuser

This creates an empty CoCo instance without any sites or nodes connected to it. The web portal asks for a user friendly name that will be associated with the instance. This is just a string. The web portal returns a secret token that needs to be used to add or remove sites to the instance, or to terminate the instance. The person that created that instance distributes the token in a secure manner (e.g. encrypted email) to other members of the closed community.

2. Add site to a CoCo instance

   Allowed by: anyone who has the security token

Users that posses the security token can add their site to the CoCo instance. They need to provide some information, like:

- Security token
- VLAN ID used for this instance
- One or more IPv4 prefixes
- One or more IPv6 prefixes

The CoCo agent associated with the web portal can now setup flow forwarding rules in the OpenFlow switches so that traffic from the site with the VLAN ID entered on the web portal is sent on the correct CoCo instance. The CoCo agent can also announce the entered prefixes in its BGP announcements for the CoCo instance. The secret token is sent in the announcement so that other domains can associate the information with the correct CoCo instance.

3. Remove site from CoCo instance

   Allowed by: anyone who has the security token

Users that posses the security token remove their site from the CoCo instance. They need to supply the security token to the web portal. The CoCo instance associated with the web portal withdraws the BGP announcements of the prefixes of that site and also removes the flow forwarding rules from the switches.

7. CoCo Inter-Domain Architecture

The final CoCo prototype will consist of multiple domains. Each domain represents an NREN. The implementation will use several testbeds (e.g. the SURFnet OpenFlow testbed and the GÉANT OpenFlow testbed), each representing a separate domain. Figure 2 shows the inter-domain architecture of CoCo. Each domain has an

OpenFlow based infrastructure and each domain runs its own CoCo agent. In each domain there is an end-user portal that communicates with the CoCo agent. End-users interact with the CoCo agents via the web portal in their own domain (NREN). These CoCo agents and the web portal are the control plane of the architecture. The CoCo agents in various domains need to exchange information (IP prefixes, CoCo instance identifier, authorization token, etc) to offer the CoCo service. The CoCo agents peer with each other to exchange information. BGP will be used to exchange the information based on RFC 4364. There are BGP peering relationships between neighbour CoCo agents. The CoCo agents also work with the concept of *transit*. E.g., CoCo agent a1 has a peering with CoCo agent a2 only. CoCo agent a1 exchanges information with CoCo agents a3 and a4 via CoCo agent a2, which in this case acts as a transit agent.

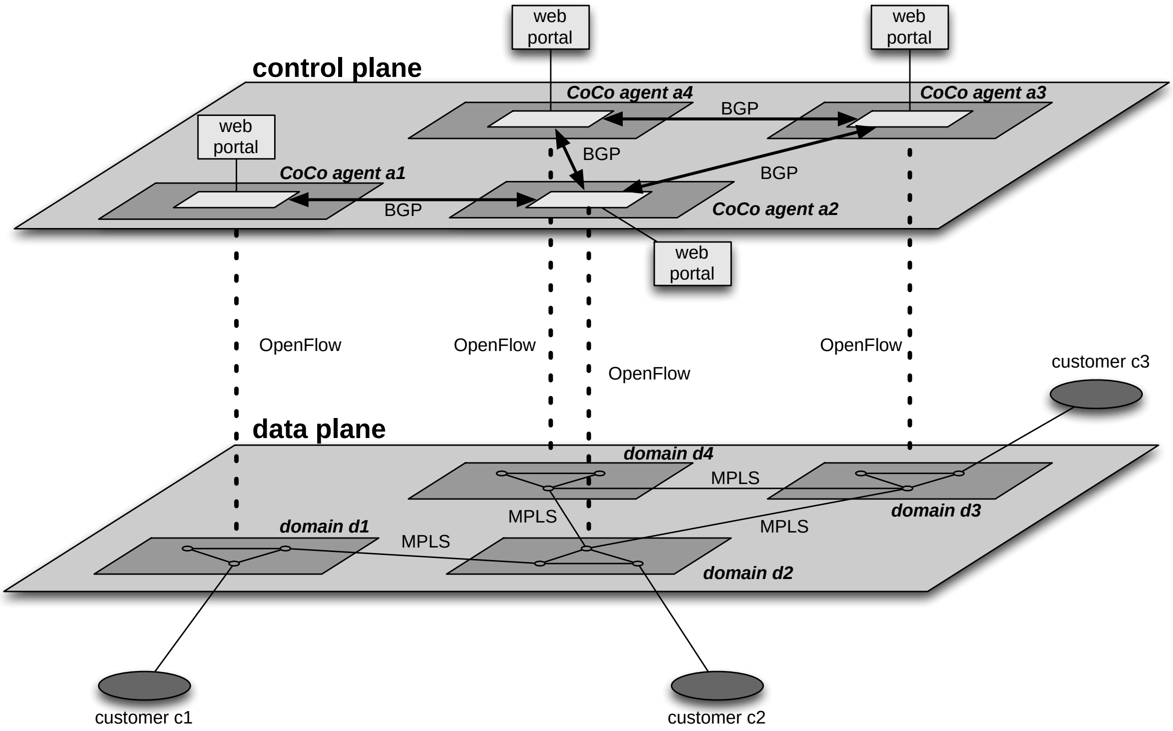

Figure 2 CoCo Inter-Domain Architecture

8. Information exchange between CoCo agents

A CoCo agent has several tasks. On task is to control the OpenFlow switches in its domain by doing topology discovery and configuring flow forwarding rules on the switches. The other task is the participating in the inter-domain control plane of CoCo. Here BGP is used to exchange information between the CoCo agents. This will be done similar to RFC 4364 “BGP/MPLS IP Virtual Private Networks (VPNs)”. For each PE in its domain, a CoCo agent sends the following information to its CoCo BGP peers:

- VPN-IPv4 address family – 12 bytes (RFC 4364)

The 12 bytes consist of an 8 byte Route Distinguisher (RD) and a 4 byte IPv4 address. An RD is encoded as a 2 byte Type and a 6 byte Value. We will use Type 2 RDs that consist of a 4 byte AS number followed by a 2 byte value. This value is managed by each domain, so by each CoCo agent. The CoCo agent manages a list of free values.

- VPN-IPv6 address family – 24 bytes (RFC 4659)

The 24 bytes consist of an 8 byte Route Distinguisher (RD) and a 16 byte IPv6 address. The RD will be the same value as for IPv4.

- Next hop (VPN-IPv4 route with RD == 0)

The next hops in the route announcements should point to the PE that has the prefixes behind it. The CoCo agent assigns fake IPv4 addresses (from 10.0.0.0/24) to each PE in its domain to be used as next hop.

- MPLS label to reach that PE (RFC 3107)

This is done in the NLRI by using a AFI of VPN-IPv4 and a SAFI of 4. The NLRI is encoded as one or more triples of the form <length, label, prefix>. The length is in bits and includes prefix and label(s). Each label is encoded as 3 octets, where the high order 20 bits contain the label value, and the low order bit contains “Bottom of Stack”. The prefix field contains address prefixes followed by enough trailing bits to make the end of the field fall on an octet boundary.

- CoCo instance identifier (2 bytes) – encoded in Route Target (RFC 4360)

A Route Target is sent via BGP Extended Communities (8 bytes) (RFC 4360) and is structured the same as a Route Distinguisher. We will use a Type 2 RD again with a 4 byte AS number and a 2 byte value. The value identifies the CoCo instance and will be used as inner MPLS label in the data plane for all traffic in that CoCo instance.

- Security token (8 bytes) – sent as BGP Extended Community

Security tokens are generated by the CoCo agents. It is a random value sent with the VPN-IPv4/VPN-IPv6 announcements. They are sent as BGP Extended Communities with no type/value structure.

9. Identity Federations (eduGAIN / SURFconext)

The security token is used for authentication to add sites/node to and remove them from CoCo instances. If time allows, we want to investigate including identity federations. A possible architecture for this is shown in Figure 3. A service plane is added that handles user and group administration and the authentication of users and the authorization to create, modify or delete CoCo instances. Similar scenarios that use identity federations typically use a centralized entity that is dedicated for that service.

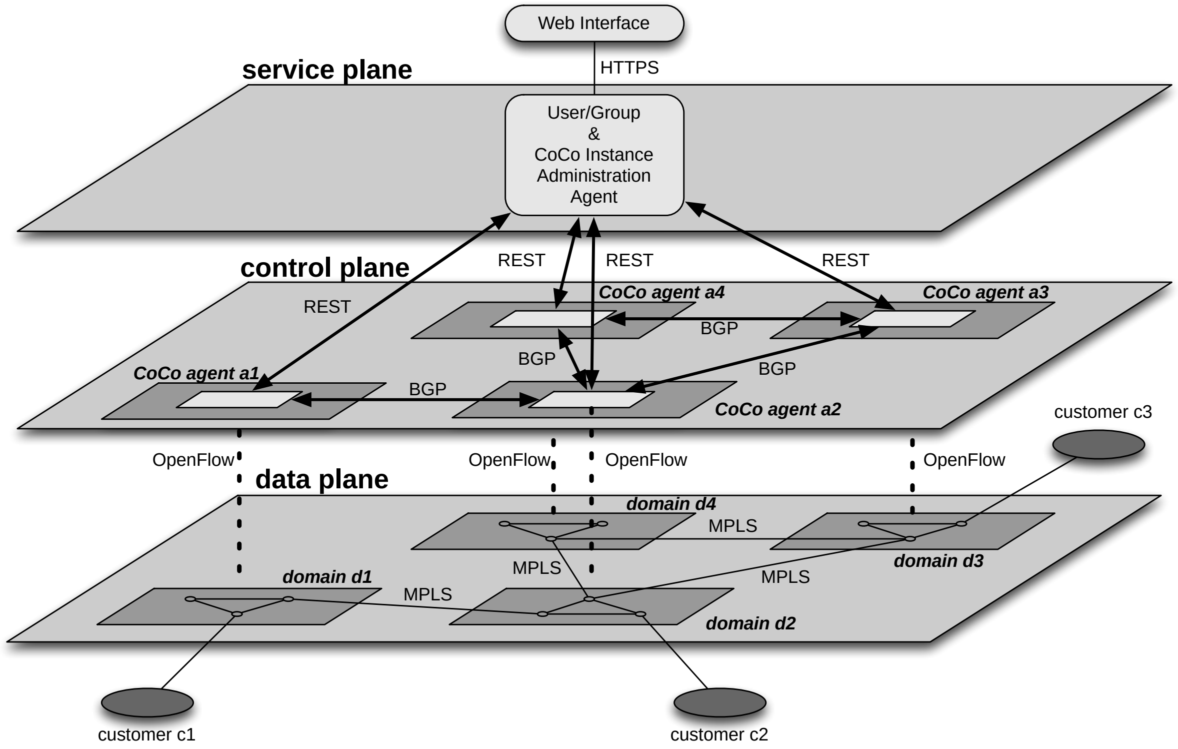

Figure 3 inclusion of identity federations

10. References

RFC 4364, BGP/MPLS IP Virtual Private Networks (VPNs)
RFC 4659, BGP-MPLS IP Virtual Network (VPN) Extension for IPv6 VPN
RFC 3107, Carrying Label Information in BGP-4
RFC 4360, BGP Extended Communities Attribute