

Interdomain Reservation and Authorization on OSCARS/DCN

Version 0.3

1 Introduction

This paper assumes a basic understanding of multidomain DCN networks. As used in this paper, a DCN network is a Dynamic Circuit Network. Individual organizations run a DCN network, and multiple DCN networks may be interconnected to create a multidomain DCN network.

Interdomain protocol capabilities are implemented in Interdomain Controller (IDC) modules in each network domain. Multiple domains are interconnected allowing a source connected to one network may make a circuit to a destination on another network. Note that source and destination have to do with the request and not with data flow.

The figure below shows three network domains that interconnect with each other. It also shows two hosts that each connect to a network. Each domain has a Data Plane, a Domain Controller, and an inter-Domain Controller. Links on a domain that connect to other networks are E-NNI (interdomain) links (idl-x) and links that connect hosts to the network-domain are UNI (edge) links (el-x) as shown below.

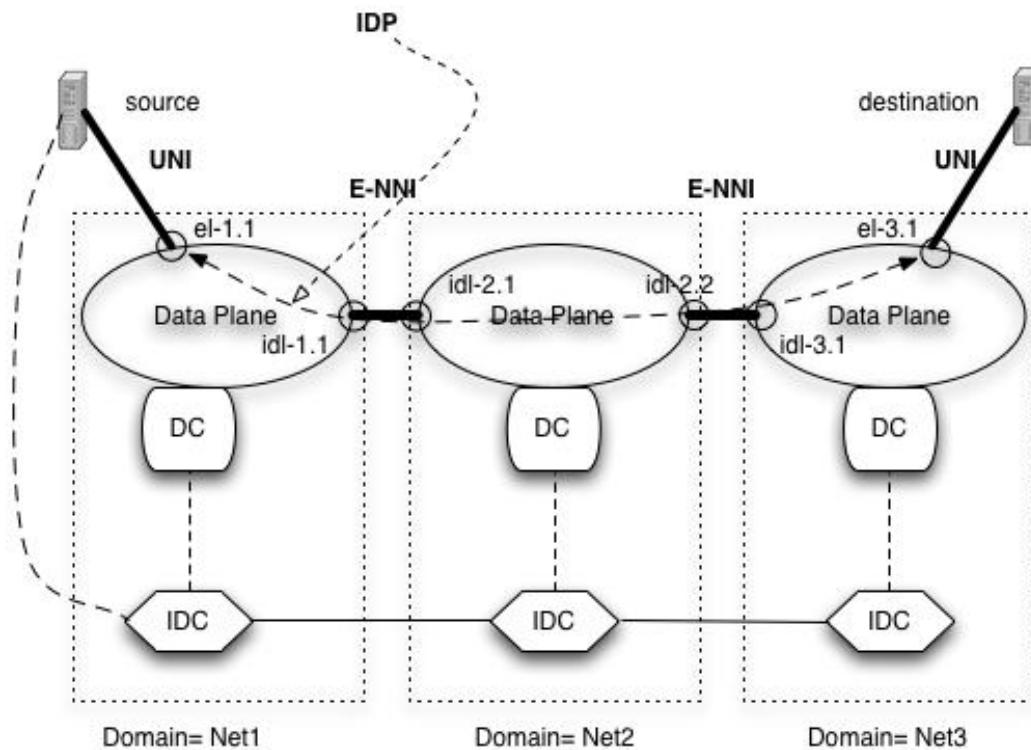


Figure 1 – Interdomain networks example

A user requests a reservation between the source and destination. The result of this request is a “confirmed” path which is an ordered sequence of domain segments which connect the source and destination. Each segment is confirmed by its domain. This confirmed path activated during circuit setup time.

In the environment shown above, the user request

ResReq [source, destination]

returns

ResResp[(el-1.1, idl-1.1), (idl-2.1, idl-2.2), (idl-3.1, el-3.1)]

This paper describes how IDCs interoperate to reserve such a path based on a user request. It gives more exact forms of the request and response and parameters that go with them, but it is not an implementation specification. The algorithm that each IDC uses to perform its part of interdomain circuit reservation is described.

1.1 Definitions

DCN is a dynamic circuit network. This can refer to a particular network-domain or to a collection of domains that are able to interoperate dynamically.

A **circuit** is a connection between two endpoints on a circuit network

A **link** is a connection between nodes.

A **circuit segment** is the portion of an ete circuit that is carried on a particular link. An interdomain representation of this is the ingress and egress URN of the domain

A **domain segment** is the portion of an ete circuit that is carried over a network domain

An **ete circuit** is a concatenation of circuit segments

A **dynamic circuit** is an ete circuit setup by a request from the user to a DCN controller

An **Inter-Domain Controller (IDC)** controls protocols between domains.

A **Domain Controller (DC)** manages the network within a domain.

The IDC requests information about the domain from the DC and instructs the DC to perform operations within its domain.

An **InterDomain Path (IDP)** connects endpoints of a circuit through one or more domain.

A **Strict InterDomain Path (SIDP)** an ordered list of the ingress and egress links of each domain in the path.

A **Loose InterDomain (LIDP)** an ordered list of at least the end links of the path and may include some or all intermediate links, or domains.

A **Confirmed Strict InterDomain Path (CSIDP)** is an ordered list of path segments that have been authorized and reserved by their respective domains.

A **GRI is a Global Reservation Identifier**. This is an id created by the initial IDC that receives a user request, is carried by all future requests having to do with the requested circuit, and is saved in state information about the reservation, whether it is being reserved, is active, or is completed and logged, or any other state. It is also kept in resource records that are related to this circuit.

UNI (User Network Interface) A UNI connects to a user device
E-NNI's (External Network to Network Interface) connects to a device in another domain.

DCN-name is a unique name for device connected to a DCN

Lookup Server maps a DCN-name for a user device to the link-id URN of the network link to which the user device connects. Every domain must have access to LS functionality. For version 0.3 the plan is to continue to run a global LS that each participating domain can use to insert its user names and port mappings. In the future this might expand to running an instantiation in each domain and sharing information (i.e. “federating”) to offer complete information coverage, using a centralized instance to manage the global view of all domains, or using some other way of providing the service.

A **link-id Uniform Resource Name (URN)** is of the form:
[urn:ogf:network:domain=*:node=*:port=*:link=]. The URN is the form agreed on by the DICE Control Plane working group and originates from work performed in the Open Grid Forum's (OGF) Network Markup Language Working Group (NML-WG). The URN is used in describing topologies and in performing path computation and reservation.

Every domain has a **Topology description**. The Topology description contains information about links from the domain to other domains.

1.2 Examples of definitions

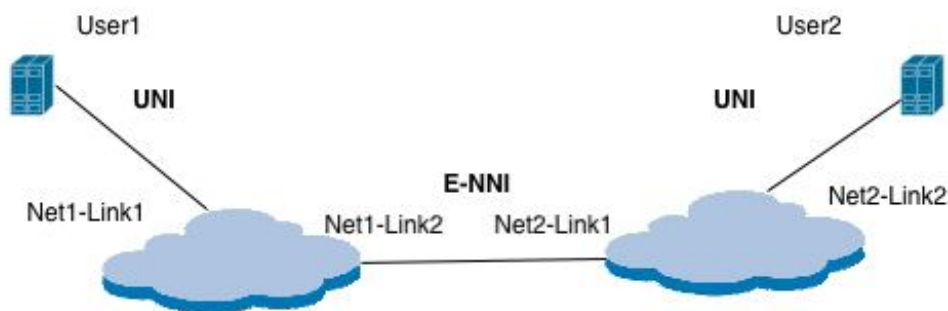


Figure 2 – Interdomain Network Elements

Figure 2 shows two domains, net1 and net2, each with connection to a user. This simple network is used for the following examples.

1.2.1 User names and URNs for the figure 2 example:

User1 name = cluster1.school1.edu

User2 name = cache.school2.edu

Net1-Link1 URN= [urn:ogf:network:domain=net1:node=city1.switch:port=A3:link=1]

Net1-Link2 -> URN=

[urn:ogf:network:domain=net1:node=city2.idsw:port=10.1.1.2:link=1]

Net2-Link1 -> URN=

[urn:ogf:network:domain=net2:node=city2.bigswitch:port=101:link=1]

Net2-Link2 -> URN= [urn:ogf:network:domain=net2:node=city3.switch:port=A1:link=1]

1.2.2 Paths between User1 and User2

A *InterDomain Path (IDP)* is a path that connects two end links.

A *Strict InterDomain Path* includes ingress and egress link for each domain between the starting link and termination link of the path. A SDP must be described as a set of link-id URNs. The only SDP between user1 and user2 above is:

```
{URN= [urn:ogf:network:domain=net1:node=city1.switch:port=A3:link=1],
  URN= [urn:ogf:network:domain=net1:node=city2.idsw:port=10.1.1.2:link=1],
  URN= [urn:ogf:network:domain=net2:node=city2.bigswitch:port=101:link=1],
  URN= [urn:ogf:network:domain=net2:node=city3.switch:port=A1:link=1]
}.
```

A *Confirmed SIDP* is an SIDP that has been authorized by each domain in the path. A confirmed SIDP is also known as a circuit, and includes time and duration and parameters such as bandwidth.

A *Loose InterDomain Path* must have the initial links and may have some intermediate route objects. The following are examples of LIDP, and show the use of different route objects.

a) LIDP with no intermediate route objects

```
{ URN= [urn:ogf:network:domain=net1:node=city1.switch:port=A3:link=1],
  URN= [urn:ogf:network:domain=net2:node=city3.switch:port=A1:link=1]
}.
```

b) an LIDP with intermediate link

```
{ URN= [urn:ogf:network:domain=net1:node=city1.switch:port=A3:link=1],
      URN= [urn:ogf:network:domain=net2:node=city2.bigswitch:port=101:link=1],
      URN= [urn:ogf:network:domain=net2:node=city3.switch:port=A1:link=1]
}
```

c) an LIDP with intermediate node

```
{ URN= [urn:ogf:network:domain=net1:node=city1.switch:port=A3:link=1],
      URN= [urn:ogf:network:domain=net2:node=city2.bigswitch],
      URN= [urn:ogf:network:domain=net2:node=city3.switch:port=A1:link=1]
}
```

d) an LIDP with intermediate domain

```
{ URN= [urn:ogf:network:domain=net1:node=city1.switch:port=A3:link=1],
      URN= [urn:ogf:network:domain=net2],
      URN= [urn:ogf:network:domain=net2:node=city3.switch:port=A1:link=1]
}
```

1.3 Lookup Service

The Lookup Service contains mapping between names of devices connected to a network and the external links on a particular domain to which the device is connected. The lookup service is used to map names (which may occur in a user request) to URNS which are required in IDC requests.

For the example the Lookup Service will have the following two entries.

```
cluster1.school1.edu <->
    [urn:ogf:network:domain=net1:node=city1.switch:port=A3:link=1]
cache.school2.edu <->
    [urn:ogf:network:domain=net2:node=city3.switch:port=A1:link=1]
```

1.4 Topology Description

Each domain has a topology description that includes, at a minimum, the the URN of each link connecting to another domains and the URN of the link it connects to in the other domain. In the example, the two networks have very simple topology descriptions, each with a single interdomain link.

Net1 topology description

```
[urn:ogf:network:domain=net1:node=city2.idsw:port=10.1.1.2:link=1] <->
    [urn:ogf:network:domain=net2:node=city2.bigswitch:port=101:link=1]
```

Net2 topology description

[urn:ogf:network:domain=net2:node=city2.bigswitch:port=101:link=1] <->
[urn:ogf:network:domain=net1:node=city2.idsw:port=10.1.1.2:link=1]

2 Dynamic Circuit Reservation and Authorization

DCN requests start with a User Request which is the initial request for a circuit, and IDC requests that are used for all subsequent requests between IDCs that are used to setup, teardown, modify or request information about the circuit.

A reservation request, if successful, results in the creation of a reserved circuit with a Confirmed Strict InterDomain Path, at a specific time and with specific parameters. This section describes how a user request is processed to create a CSIDP.

2.1 Request types

2.1.1 User Request

The user request contains the path for the desired circuit. A user request, as all requests, must contain the source and destination link for the requested circuit. The user request is unique in that the source and destination links may be represented by DCN-names or URNs. If either the source or destination is a DCN-name, it must be converted to a URN. This is done by the Lookup Server.

2.1.2 IDC request

An IDC request is used to communicate about a specific circuit. An IDC request is created from the user request by adding a GRI (Global Reservation Identifier). The GRI is a permanently unique identifier, typically created by using the domain of the creating IDC and a unique identifier with that domain.

An IDC request can contain only URNs in its path. Once the user request has been modified to map DCN names to URNs and to add a GRI, the request becomes an IDC-request.

An IDC reservation request is of the form

ResReq {[GRI, Transaction-id, path], [time, duration], [circuit parameters], [setup parameters]}

2.2 IDC Reservation request processing

The ResReq is used to communicate between IDCs to come up with a Confirmed circuit satisfying the initial request. There are two areas of negotiation between IDCs, path selection and capabilities negotiation.

Path selection come in when a LIDP is in the request. When it receives an LIDP an IDC **may** modify the LIDP to make it a SIDP. The IDC **must** be sure that the LIDP includes the path segment for the IDC domain.

Capabilities negotiation occurs when the request allows some variation in what is required. For example, the [time, duration] field in the request might say [start=00:30-06:00, duration = 0:45] meaning that the request is for 45 minute anytime between 00:30 and 06:00. In this case an IDC might modify the field to say [start= 02:00-05:00, duration = 0:45].

2.2.1 Path processing

IDC path processing depends on the path that is received in the request. There are three possible path types that can be received

1. SIDP – contains path segments of all domains in the path
In this case the IDC does not modify the path
2. LIDP which contains the path segment for this domain
In this case the IDC **may** modify the path, but does not have to
3. LIDP which does not contain the full path segment for this domain
In this case the IDC **must** modify the path to include the path segment for this domain and **may** modify the path further, as in case 2

In case 1 no path determination is required. In steps 2 and 3 the IDC requests path information (PI) from its Topology server. The PI request sends the current path and receives a modified path. The received path may be either a SIDP or LIDP.

Description of the Topology server is in a separate document.

2.2.2 Circuit capabilities processing

When the IDC receives a Reservation request, it selects the path segment which is the the IDCs domain. It then asks its Domain controller/scheduler whether the resources requested for this segment are available.

If the resources are not available, the DC denies the request.

If the request can be satisfied,

1) resources needed to support the proposed circuit are requested, scheduled, and authorized and put in reserved state. [Note if the request allows multiple choices the IDC may reserve all the choices which it supports and return the revised choices]

2) The IDC returns a reservation accepted

2) the IDC waits for a Reservation Confirmation message and then marks the reservation as confirmed.

2.2.3 Reservation Confirmation at IDC

Once the IDC of each domain in a Reservation Request has accepted the reservation for its domain, then a Reservation Confirmation is created and sent to each domain. There are alternative methods for sequencing this, as described in section 3 below.

The Reservation Confirmation contains a specific set of resources that are required for the reservation. Each IDC takes the information for its domain, marks it as confirmed, and releases any resources that were tentatively reserved.

3 Interdomain Circuit requests using IDC protocol

3.1 Multi domain Circuit creation and confirmation

There are two general approaches to implementing interdomain dynamic circuits using IDC protocols described in section 2 above.

1. Daisy Chain
2. Tree scheduler

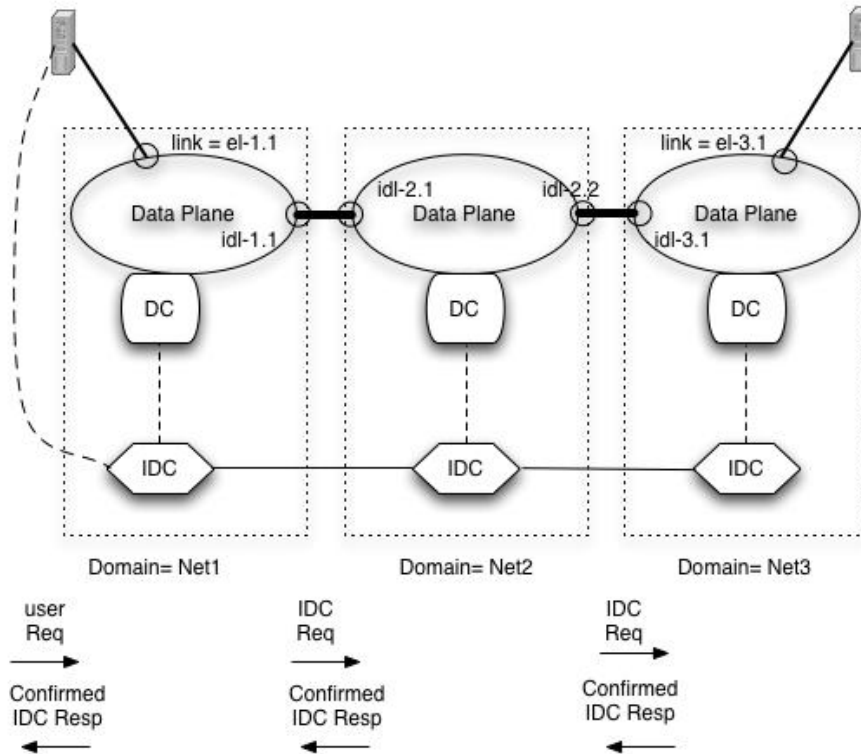
Each of these is described in a section below.

3.1.1 Daisy Chain

In the Daisy Chain method, the IDC of each domain is asked in sequence to reserve the path. An IDC receives a request, processes it and tentatively reserves the resources needed in its domain.

When the last IDC in the chain receives the request, it recognizes that the destination is in its domain. It reserves its part of the circuit, and creates a Reservation Confirmation. It then modifies reservations in its domain if necessary, and returns a Reservation Confirmation message to the previous IDC. This is repeated until the Confirmation reaches the originating IDC.

Figure 3 below shows such a path



3.1.2 Tree Scheduler

In the Tree Scheduler method, the initial request goes to a “Tree Scheduler” IDC. Figure 4 show such an implementation. In this model, the initial request goes to a Tree Scheduler.

The tree scheduler determines the IDC path and then sends a request to each IDC, perhaps in parallel. Once the tree scheduler gets a response from each domain, it use those to create a Confirmation message which it sends to each domain. When it receives an acknowledgement from each domain it returns a success (or failure) to the requestor.

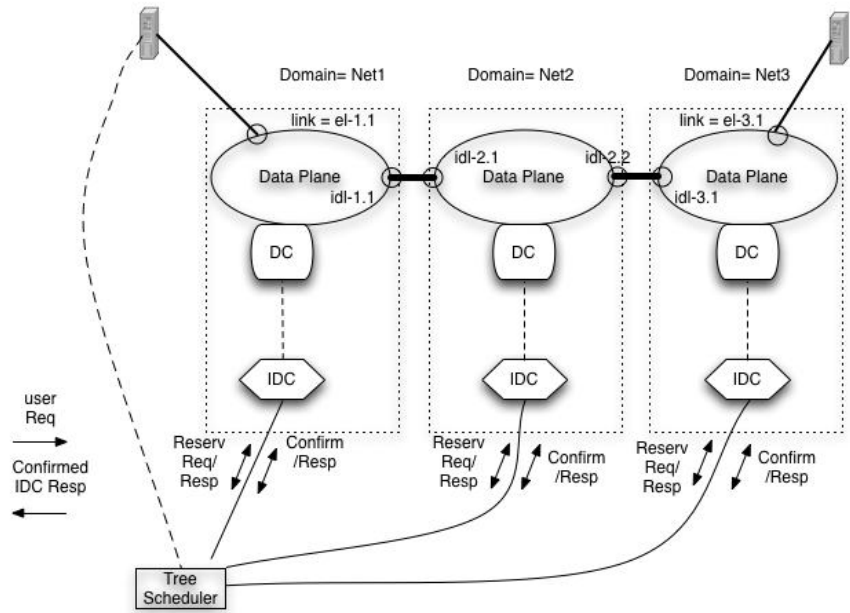


Figure 4

3.1.3 Selecting Daisy Chain vs Tree Scheduling

In order to use either the daisy chain or tree mechanism, when making an IDC request, one must include information about whether the IDC is to forward the request to the next domain in the path, or is to send it to the requestor – i.e. the Tree scheduler. The direct-return field in the request request indicates that the IDC should return directly to the caller.

There are a variety of reasons for choosing Daisy Chain or Tree Scheduling. The choice of which method to use is an implementation issue. It seems possible that an infrastructure could be constructed where some of it used daisy chain methods and some used tree method; this is left for future study.

Some issues that might affect which method to use:

1. Timing might be different with different methods. The tree method allows reservations to proceed in parallel
2. Resource reservation might churn more in tree method than daisy chain since a failure in a domain would halt further requests for reservation
3. Trust relations are different.

4 Summary

This describes the high level architecture to support implementing circuit requests across multiple domains. This needs to be discussed, modified and corrected. The intent is that this will form one basis for deciding what will be implemented in the next version of IDC protocol and the Internet2/ESnet DCN software suite.