UIT - Secteur de la normalisation des télécommunications
ITU - Telecommunication Standardization Sector
UIT - Sector de Normalización de las Telecomunicaciones

Commission d'études Study Group Comisión de Estudio	13	Contribution tardive Delayed Contribution Contribución tardía	**D.**

Q.19/13 Rapporteur's Meeting, September 2000, Turin

Questions: 19/13	Texte disponible seulement en Text available only in Texto disponible solamente en	**E**

SOURCE*: T1X1

TITLE: ASON – Requirements at the Client API

ABSTRACT

Work already done on ASON has identified several requirements and architectural features. This contribution contains material providing requirements, operational descriptions, and attributes of the ASON Client API, which from a logical perspective is termed the User Network Interface. It represents our understanding of requirements from an ASON customer perspective, connection attributes, and operation phases and constructs.

* Contact Person: G. W. Newsome
101 Crawfords Corner Road
Holmdel, NJ 07733 USA
Tel: +1 732 949 0812
Fax: +1 732 949 3210
Email: gnewsome@lucent.com

Michael Mayer
PO Box 402
Ogdensburg NY 13669
USA
Tel: +1 613 765-4153
Fax: +1 613 763-2388
Email: mgm@nortelnetworks.com

1 . Introduction

Work already done on the Automatic Switched Optical Network (ASON) has identified several requirements and architectural features. In particular several interfaces have been identified, however there are no detailed requirements for these interfaces. This contribution presents our understanding of requirements from the point of view of what an ASON customer can be expected to do with ASON. These are mainly requirements at the Client Application Programming Interface (API). Note that the Client API is the logical interface between a network client and the network server.

The contribution limits itself to customer control aspects, and makes no statements about what may or may not be supported using conventional provisioning protocols. G.ason already has several requirements that are pertinent to this contribution, and those requirements are replicated here.

2 . Discussion

2.1 Potential business models

Before we can consider requirements for ASON services, we must briefly consider the business models that may be applicable and which must be supported. The focus here is on business rather than technical issues.

1. ISP owning all layer 1 infrastructure
   - The only delivered service is IP based. There is no need to offer any other connection services on the infrastructure. As the infrastructure is fully owned, there is no trust issues between the infrastructure and IP layers. There is only a single instance of IP needed.

2. ISP owning or leasing layer 1 infrastructure
   - The only delivered service is IP based. There is no need to offer any other connection services on the infrastructure. As the infrastructure may be leased, there is a trust issue between the infrastructure and IP layers. There is only a single instance of IP needed.

3. Retailer or wholesaler for multi-services
   - In this case the business owns the layer 1 infrastructure and sells services to customers who may themselves resell to others. There is a definite trust issue between this business and the client businesses. As this business cannot make any assumptions about the business of the customers, it is highly likely that several customers may be engaged in the same business. This business offers many different services to many different clients. If the customers are engaged in the IP business (as in case 2) there will be several instances of IP networks running over this network, and each instance will have its own control plane. (Of course, only one ASON control plane instance is required)

4. Carriers carrier, or bandwidth broker
   - Potential clients of such an application are other network operators, once removed from delivering service to an end user. In this case there is again a major trust issue. The customer network is likely to be a circuit network, potentially in the same layer as this network (OCh taking part in another operators network) and again there will be several instances of client networks in the same layer each with its own control plane. In this model the provider has no idea what services are being eventually offered by the client network.

It is clear that all four models are reasonable and exist today. Indeed, a network might be used in all four ways by various organisations within the same network operator. The most stringent requirements are placed by case 3 and 4 as both these cases have trust (or security) issues between the server and its client business, and both can have multiple instances of client networks. This need for multiple instances indicates that there is a greater need for partitioning than by technology alone. For this reason there must be support for any desired general partitioning of the network resources, and case 1 and 2 is handled as a special case where the only partitions are driven by technology and layer boundaries. We note that while ASON is directed primarily to new networks, there is nothing to preclude its use to control existing networks.

These potential business models make it clear that support for all types of protocol and carried services is required. We reaffirm that ASON must not assume a single client type and instance.

2.2 What is required from an ASON "dial-up" service?

ASON has been described in terms of path management, path set up, client discovery, resilience against server faults, address resolution, and addressing. The question of what is required from an ASON "dial-up" service can best be answered using an approach similar to that used by Rec. G.85x in defining connection services. In particular, first define a basic service that sets up connections and then think of features that may be added to the basic service. This allows us to separate the various features, enabling them to be implemented or offered separately. Some key service features are enumerated below:

Connection set-up

> This service sets up an end to end connection between a pair of client names, and may be triggered by either a client end point or a third party. The trigger point may be located anywhere, including at one or other end point. There are three essential operations needed for the service: Set-up, Modify, and Release a connection.

a) Set-up: The ASON UNI provides its users with the necessary mechanisms to signal to the network its desire to create a new connection between two endpoints. Certain attributes are associated with the requested connections. Those attributes are contained in the signalling message.

b) Release: Upon signalling of a connection release by ASON client, the network eliminates all the transport resources associated with this connection, though a record of the connection should be maintained for billing purposes. It is expected that the network will acknowledge the release of the connection back to the client.

c) Modify: This action results in modifying one or more of the attributes associated with a client’s connection. This action allows ASON clients to adjust connection attributes based on his actual usage. For instance a client could invoke this action to modify the connection bit rate (bandwidth) based on measurements of the actual usage.

Connection Resilience

> This service is so named to distinguish it from the protection and restoration mechanisms that are used to provide fault tolerance to the connection. The service adds the ability to tolerate faults to the basic connection set-up service itself. Any parameters to this service must be in terms meaningful to the client and may not specify server layer architectural solutions. In this service we must recognise that various faults may be protected against, and the most important characteristic of the fault is how much physical space is affects. It is likely that such parameters are in terms of fault radius, as well as availability. This is important as the degree of diversity required depends on the size of the fault. Earthquakes, for example, affect a wide area yet a service request may well want resilience in the face of an earthquake.

Connection Constraints

> There must be a means of attaching constraints to a particular connection request.
> Route diversity

It is conceivable that there is a need for multiple connections to be set up in one operation. An obvious application is the provision of two diverse connections so that a client layer can provide fault resilience. This can only be provided by specifying constraints on the connections, and these constraints will be comparable with those needed in the Resilience service. It is not possible to specify these needs in terms of a potential solution as the client has no view of the server name space and connectivity.[1]

Pre-emption

Indicates both the set-up and the holding priorities of the connection. A higher set-up priority connection could pre-empt a lower holding priority connection if network resources are scarce during the connection establishment time.

Adaptation management

Some infrastructures may allow for dynamic change to the client adaptation on a call by call basis. Examples are bandwidth selection in infrastructures supporting virtual concatenation and possible variable client encoding schemes.

Service Policies

While not a client feature, policies are the means by which the service provider enforces the Service Contract agreed to by the client.

Call Acceptance

An essential part of connection oriented networks is call acceptance. It must be possible to control the types of connection requests that are accepted by the network. This allows the traffic intensity to me matched to the available network resources as well as allowing various forms of restricted network access.

Billing method

It may be useful to consider the method of billing to be variable between calls.

Virtual name spaces

It is a requirement to support partitioning of the server network into several independent virtual name spaces. The common application of this is the provision of Virtual Networks. Note that there are several forms of private networks. The form that is applicable to the Client API presents a particular set of end points in the server network in the form of a closed group. Connections may be made between these end points and no others. An expansion of this will allow the server to restrict the number of links that may be used among these end points. As a separate feature, the client may be provided a service to alter the number of links provided. Any other forms of Virtual network, in particular those forms in which the client gains access to transport network equipment, are not covered by this API.

2.3 API Operation Phases

There are several phases in the operation of ASON and it is important to understand to which phase a particular requirement applies. These phases are 1) Contract Negotiation, 2) Equipment Installation and 3) Connection Service usage. Each phase requires interaction between the client and the ASON provider and both one and two must be completed before the connection service is available. There is no constraint on the order of step 1 and step 2, which means that it is quite feasible to have a customer connected to ASON who does not yet have a contract, and who therefore cannot make calls (using the PSTN as a metaphor for terms).

It is obvious that that the client API cannot provide any meaningful functionality during equipment installation (software after all cannot install hardware), and it is not necessary that the API provide any of the functions required during contact negotiation. ASON is about the automation of connection set-up, and

[1] Note that one type of constraint could be "avoid city X".

not primarily about contact automation. However it is expected that the vast majority of clients will be machines, not people, so some degree of automation could be applied.

An important part of configuring a client is name binding, the process by which the client name (metaphorically its phone number) is bound to the corresponding network name, and entered into a directory owned by the server layer. This directory must be owned by the server layer to preserve the security of the server layer names and addresses. Suffice it to say here that the server layer should accept client defined names on the connection set-up interface (the main service of the Client API) and the server is responsible for mapping those names into the names of server layer ports between which connections are actually made. A further requirement is to automatically check the client to server connectivity at the same time that these names are bound.

Note that successfully binding the names does not imply that the connection service is available. This depends on all aspects of the contract being completed.

2.4 Requirements on Client Names

From the point of view of the server, there are no useful semantics carried by the client name. Note that the difference between names and addresses is that names have no semantic content, while addresses have semantics designed to assist the server layer with routing. Names generally exist for longer than addresses, and this allows for name portability.

At first sight it might seem that the client name should be unique within server, but in fact the only server requirement is that a name must be registered before it is used. Allowing the client to register the same name for several different link connection end points allows the client to be in full control of its links as the links are effectively created by the identical names. Because these names have no useful semantics to the server, the management of these names is best delegated to an organisation outside of the server operator.

Client names must of course be unique among all clients in the network, and the scope of ASON is global. There are several ways that this uniqueness may be ensured. The first is the mechanism chosen for user labels in Rec. G.85x – the client provides a name, which is checked for uniqueness by the server. This method is attractive as it allows the client to provide any name that is convenient. The drawback is that the global network will not exist on day one, so growing the network by interconnecting ASON islands can give problems when names that are initially unique are no longer unique. The second method is to choose names that are managed by an authority external to ASON. As an example, client names could be DNS names, or IP addresses, or an ATM access point identifier, or E164 telephone numbers, or ISO object identifiers, or an encryption public key, or anything which is unique over a large enough space. A client name can name an access point or access group, within one or more network partitions. This is to allow calls to be set up between clients who have several links connections available that are equivalent for the purposes of setting up the call. When an access group name is specified, the server will choose which access point to use and will return the choice made to the client.

A question that we may pose is "can/should client names be unique in the ASON or only within a particular client network partition"? As suggested earlier, the client can control the composition of its links by the names it registers. An access group is the set of all access points that are taken to be equivalent for the purposes of routing, and may be created by registering the same name for more than one access point. Note that each access point must also have a unique name itself. When an access group name is provided, the server may choose which access point is chosen. There is thus no need for an API parameter specifying the layer network as this is already determined by the end point name. Such parameters would identify the bandwidth or the technology, as is being proposed in other fora are therefore not required.

2.5 API contract set-up phase operations

This phase must be completed before any calls can be made. It involves creating the bindings between the client names and network names that was referred to earlier. It is also possible that a client may wish to register more than one name as an alias (equivalent to having several directory entries for the same telephone number), as it is possible that the client will wish to register the same name for several end points. (to create access groups). This feature need not be available on all infrastructures. Example applications involve clients that want names in several name spaces or simple aliases in the same name space. The service must therefore provide primitives to Register, Modify and Remove name bindings:

- The Client API shall provide a service to Register the Client name with the network. This service may be provided as either a client or a server initiated operation. It shall have been used before the connection set-up service can be used, and may be used at any later time to register new client names (aliases).
- The Client API shall provide a service to modify a registered client name. This may be used at any later time to change the client name and shall have no effect on any connections that are established to or from this client at the time that the name is changed.
- The Client API shall provide a service to Remove a registered client name.

At the same time that the names are being bound, the service shall verify the connectivity between the client and the server. A mechanism to accomplish this is required. Note that this operation does not of itself enable the connection service.

It is necessary that the client be able to connect to an appropriate server before the ASON dial-up service has been established. It is expected that this will be accomplished by a well-known server address (possibly a URL) with security established manually as part of the contract. This model is widely used today, as many contracts supply unique information allowing the client to access online services after the contract has been established.

2.6 API connection set-up Operations

The main operations supported by the client API are associated with the setting up and tearing down of Optical Channel "calls." In addition to the Set-up and Release operations, there may be a need for a "dial-tone" operation. This is to signal to client equipment that there is a connection service available. This covers the case where hardware is installed and the client is connected to ASON but there is no contract established. It seems reasonable that the client can discover the availability of the service without having to make a test call.

As previously mentioned, calls are made between two client names, traditionally called the A end and the Z end. It must be possible for both participants and third parties to set-up calls, and it must be possible to have a control plane that is not associated with the connection being set-up. These requirements mean that the set-up operation must have both the A and Z end client name in the set-up request.

2.7 Operations

2.8 Set-up

End points

It must be possible to set-up a connection between two access points or access groups specified by pair of client names.

Scheduled service

It must be possible to request the connection to be made at some time in the future. Periodic schedules are also required. (normal operation is "now")

Scheduled duration

It must be possible to specify a duration for the connection (normal operation is "hold until explicitly released")

Resilience

The connection requested must be resilient to server layer faults. The degree of risk shall be specified, and not the mechanism to be used to mitigate that risk. This requirement can be met by either protection or restoration techniques. Constraints shall be provided that specify various needs for QoS as this relates to resilience. These constraints shall be specified in terms of client needs; not in terms of server mechanisms. An example could be to specify resilience as "on" or "off", while a more detailed constraint could specify a radius of failure. (This is the connection to shared risk groups, which determine the radius of a failure)

Pre-emption

Indicates both the set-up and the holding priorities of the connection. A higher set-up priority connection could pre-empt a lower holding priority connection if network resources are scarce during the connection establishment time.

Adaptation management

Some infrastructures may allow for dynamic change to the client adaptation on a call by call basis. Examples are bandwidth selection in infrastructures supporting virtual concatenation and possible variable client encoding schemes.

Return values

Connection name

To allow for the client names to refer to access groups or more than one layer network, each completed connection must be provided with a server handle which will be used for subsequent interactions with the connection. The client name pair is not sufficiently unique. This name will allow the client to query the connection, and the connection data must be able to persist for longer than the connection itself.

Connection Status

It must be possible to signal success or failure to create the requested connection.

The API shall provide a service to tear down a previously set up connection

2.9 Release

Connection name

The connection to be torndown shall be indicated by the Connection name obtained from a previous set-up operation.

Return values

Connection status

It must be possible to signal successful disconnection.

2.10 Modify Connection

A service is required that allows certain connection characteristics to be changed.

2.11 Security issues

This contribution has mentioned the need for security mechanisms to enforce separation between the various parties connected via the client API. While today's internet security is still not perfect, it is very much better than the security models which were available to the TMN designers. As an example, security can be provided using public key encryption methods for session authentication and triple DES encryption of transmitted data. The weakest link in this method is the management of the authorisation certificates involved and in the degree of confidence in the certifying authority. These are the usual people issues of any security scheme.

Rather than handle security on a per API basis, it seems more reasonable to handle security as a property of the control plane infrastructure and to use this means to set up secure associations to handle the security of the Client API. In this way various security levels can all be handled by providing multiple interface instances, each one controlled by a different security policy. The need for security fields in each message is therefore removed.

2.12 Things not supported in the Client API

There is some confusion in what exactly is covered by the client API, and this confusion can best be avoided by understanding that the API can contain more than one interface. An interface is associated with one or more operations. The interface described here is responsible for initiating client operations and allowing control of the connection service. There are NO operations on this interface that allow for any more than that which is specified here.

There are some requirements to be provided by querying the created connection using the connection name as a parameter. As mentioned, a connection may be set-up between a pair of access groups. In this case the full client name of each end point chosen must be made available to the client. In order to support client query of (say) call duration and cost after the call has been made, the connection data must be more persistent than the connection. These and other queries are operations on a link object and are for further study.

3 . Proposal

The following text should be added to G.ason as a new section.

3.1 Requirements at the Client API

3.2 Requirements on Client Names

From the point of view of the server, there are no useful semantics carried by the client name.

The client will manage access group composition by registering the same name multiple times.

The management of client names is best delegated to an organisation outside of the server operator.

As an example, client names could be DNS names, or IP addresses, or an ATM access point identifier, or E164 telephone numbers, or ISO object identifiers, or an encryption public key, or anything which is unique over a large enough space. A client name can name an access point or access group, within one or more network partitions.

There is no need for an API parameter specifying the layer network as this is already determined by the end point name.

3.3 API contract set-up phase operations

This phase must be completed before any calls can be made. It involves creating the bindings between the client names and network names that was referred to earlier. It is also possible that a client may wish to register more than one name as an alias (equivalent to having several directory entries for the same telephone number), as it is possible that the client will wish to register the same name for several end points. (to create access groups). This feature need not be available on all infrastructures. Example applications involve clients that want names in several name spaces or simple aliases in the same name space. The service must therefore provide primitives to Register, Modify and Remove name bindings:

- The Client API shall provide a service to Register the Client name with the network. This service may be provided as either a client or a server initiated operation. It shall have been used before the connection set-up service can be used, and may be used at any later time to register new client names (aliases).
- The Client API shall provide a service to modify a registered client name. This may be used at any later time to change the client name and shall have no effect on any connections that are established to or from this client at the time that the name is changed.
- The Client API shall provide a service to Remove a registered client name.

At the same time that the names are being bound, the service shall verify the connectivity between the client and the server. A mechanism to accomplish this is required.

It is necessary that the client be able to connect to an appropriate server before the ASON dial-up service has been established. It is expected that this will be accomplished by a well-known server address (possibly a URL) with security established manually as part of the contract. This model is widely used today, as many contracts supply unique information allowing the client to access online services after the contract has been established.

3.4 API connection set-up Operations

The main operations supported by the client API are associated with the setting up and tearing down of Optical Channel "calls." In addition to the Set-up and Release operations, there may be a need for a "dial-tone" operation. This is to signal to client equipment that there is a connection service available. This covers the case where hardware is installed and the client is connected to ASON but there is no contract established.

Calls are made between two client names, traditionally called the A end and the Z end. It must be possible for both participants and third parties to set-up calls, and it must be possible to have a control plane that is not associated with the connection being set-up.

3.5 Operations

3.5.1 Set-up

End points

It must be possible to set-up a connection between two access points or access groups specified by pair of client names, A and Z.

Scheduled service

It must be possible to request the connection to be made at some time in the future (normal operation is "now")

Scheduled duration

It must be possible to specify a duration for the connection (normal operation is "hold until explicitly released")

Resilience

The connection requested must be resilient to server layer faults. The degree of risk shall be specified, and not the mechanism to be used to mitigate that risk. This requirement can be met by either protection or restoration techniques. Constraints shall be provided that specify various needs for QoS as this relates to resilience. These constraints shall be specified in terms of client needs; not in terms of server mechanisms. An example could be to specify resilience as "on" or "off", while a more detailed constraint could specify a radius of failure. (This is the connection to shared risk groups, which determine the radius of a failure)

Pre-emption

Indicates both the set-up and the holding priorities of the connection. A higher set-up priority connection could pre-empt a lower holding priority connection if network resources are scarce during the connection establishment time.

Adaptation management

Some infrastructures may allow for dynamic change to the client adaptation on a call by call basis. Examples are bandwidth selection in infrastructures supporting virtual concatenation and possible variable client encoding schemes.

Return values

Connection name

To allow for the client names to refer to access groups or more than one layer network, each completed connection must be provided with a server handle which will be used for subsequent interactions with the connection. The client name pair is not sufficiently unique. This name will allow the client to query the connection, and the connection data must be able to persist for longer than the connection itself.

Connection Status

It must be possible to signal success or failure to create the requested connection.

The API shall provide a service to tear down a previously set up connection

3.5.2 Release

Connection name

The connection to be released shall be indicated by the Connection name obtained from a previous set-up operation.

Return values

Connection status

It must be possible to signal successful disconnection. (what's it mean to fail to disconnect)

3.5.3 Modify Connection

An operation is required that allows certain connection characteristics to be changed.

3.6 Security issues

While today's internet security is still not perfect, it is very much better than the security models which were available to the TMN designers.

Security will be handled as a property of the control plane infrastructure, rather than on a per API basis, to provide secure associations for the Client API. In this way various security levels can all be handled by providing multiple interface instances, each one controlled by a different security policy. There is no need for security fields in each operation.

References

Note that T1X1 documents can be freely obtained from the T1X1 web site at http://www.t1.org/. ITU documents can be obtained from http://www.itu.ch/.

1. T1X1/2000-011, *Work on the Automatic Switched Optical Network,* T1X1.5 (January 2000).

2. T1X1.5/2000-143, *Some Routing Constraints*, AT&T, T1X1.5 (July 2000).

3. T1X1.5/2000-154, *Further Attributes of the ASON User Network Interface*, Nortel Networks, T1X1.5 (July 2000).

4. G.85x refers to the G.85 series of recommendations, most important are :-

- ITU-T Recommendation G.851.1 (11/96) - Management of the transport network - Application of the RM-ODP framework

- Recommendation G.852.1 (11/96) - Management of the transport network - Enterprise viewpoint for simple subnetwork connection management

- Recommendation G.853.2 (11/96) - Subnetwork connection management information viewpoint

5. T1X1.5/2000-128, *First Draft of Rec G.ason, "Architecture for the Automatic Switched Optical Network"*