DSL Forum
Technical Report
TR-046

(Formerly WT-060v4)

Auto-Configuration
Architecture & Framework

February 2002

Abstract:
This document describes the architecture & framework within which Auto-Configuration of the B-NT occurs. The architecture is modular so that new technologies, protocols and solutions can be added to the framework when they become available. It describes architecture requirements and the role of related technical recommendations and working texts within the framework.

DSL Auto-Configuration Architecture

1 Table of Contents

2 Problem Statement & Requirements

Requirements context

The DSL Forum auto-configuration framework serves specific entities in the network, specifically the subscriber, the Network Access Provider (NAP) and the Network Service Provider (NSP). The NAP provides some variation of point to point logical connectivity between a subscriber and an NSP. This is constructed by some concatenation of DSL/copper access loops with a WAN technology. The NSP hosts services.

[Requirement 1]: Minimize customer interaction and truck rolls

The goal of auto-configuration is to eliminate/minimize the effort required of the customer in order configure a DSL B-NT without merely shifting that burden to other craftspeople. This is achieved within this framework via specifying network centric mechanisms such that both initial configuration and life cycle management can be automated and performed after physical installation of the B-NT.

[Requirement 2]: Arbitrary NAP/Multiple NSP/Customer relationships

A subscriber may obtain service from one or more NSPs. A NAP may offer services to multiple NSPs. An NSP may be a client of multiple NAPs. In some scenarios the NAP and NSP may be vertically integrated however this only constitutes a portion of the overall market for deploying DSL services. The more likely scenario is that the provider at the top of the stack, the NSP, may be constructing a network by obtaining DSL from many access providers, each of which may be deploying different technology. This necessitates a number of specific requirements:

- It must work regardless of the relationships between subscriber, network and service providers.
- The division of NAP and NSP must be supported, for example, a NAPs DSLAM providing access services for multiple NSPs
- It must keep the required interfaces between providers as simple as possible.
- It may be required to reach into the customer premises network.
- It must make new service introduction as simple as possible.

[Requirement 3]: NAP/NSP/CPN management must be independent

The framework recognizes the horizontal stratification of the DSL industry and ensures that each provider has the option of direct management connectivity to perform the necessary configuration, monitoring and assurance functions that are part of their service offering.

[Requirement 4]: Multiple services

The framework must be able to address a number of single and multi-service scenarios. The following were identified:

1.Single service from a single NSP
Primary focus on Internet Access including IP address, DNS server using CPE obtained independent of the NSP
2. Sequential access to single service type from multiple NSPs
3a. Configuring ATM layer for multiple ATM endpoints on the premises, at least one of which does not terminate in the xTU-R
3b. Simultaneous access to multiple types of services from a single NSP
3c. Simultaneous access to different types of services from multiple NSP
4. Simultaneous access to single type of service from multiple NSPs

[Requirement 5]: Must support more than ADSL DMT and ATM

- It must work for different flavors of DSL.

The auto-configuration framework needs to be cogniscent of the variety of potential inter-provider relationships that might exist. Therefore it has been designed to holistically fit into an overall architectural framework that encompasses both network based configuration and service management for each business entity / management realm. This business entity and management realm[1] separation (in specific implementations this is sometimes called "layer" separation) is of significant importance for many reasons:

1) DSL encompasses a number of different physical layer technologies (e.g. G.991, S/HDSL, VDSL etc.). These must be interchangeable in our architecture with minimal cross layer impacts[2].

2) DSL encompasses a number of different transport layer technologies (e.g. ATM, HDLC etc.). These must be interchangeable in our architecture with minimal cross layer impacts.

3) DSL providers find themselves in a horizontally stratified industry whereby the business and operational owner of a particular portion of the network does not own or operate other portions of the network. Configuration and management of their service is most easily and appropriately accomplished within their own management realm[3].

[1] See section 4.3.2 Nested Auto-Configuration Architecture for a more in depth description of "business entities" and "management realms".
[2] An example of DSL layer specific architecture elements would be the T1.413 EOC channel combined with the RFC 2662 ADSL MIB. There is layer connectivity and a defined configuration and management MIB. Similar efforts are underway to create similarly consistent layer architectures for G.lite and G.shdsl.
[3] See section 4.3.2 Nested Auto-Configuration Architecture for a more in depth description of "business entities" and "management realms".

[Requirement 6]: Should work in existing networks, regulatory environments

- It must be easily deployed in existing networks.
- It must work in multiple regulatory environments.

DSL providers may wish to provide auto-configuration for their service and do not wish deployment to have to be tightly coupled to that of other providers in the service hierarchy. Similarly regulatory reasons may require that DSL providers function autonomously. Therefore this framework does not require "all or nothing" deployment. DSL layer configuration can be decoupled from transport layer configuration can be decoupled from Service layer configuration. Further, the recommended hierarchy builds upon existing tools and practices.

In this framework, there are either none or minimal dependencies between the management realms / layers and other technologies may be substituted in the mix transparently. This best meets the needs of the forum membership as it is agnostic to both technology, regulatory and business stratification.

Similarly, this framework is designed to not unreasonably complicate outside plant and is deployable whether the DSL is terminated in a central office (e.g., in a DSLAM) or remotely (e.g., in a RAM, NGDLC, etc.).

[Requirement 7]: Support non-disruptive incremental service provisioning

Services changes should be able to be enacted by the provider without affecting the existing service mix.

[Requirement 8]: Allow incremental deployment/upgrade of autoconfig framework

In addition to working in existing networks, and regulatory environments, the framework needs to be able to be enhanced with additional functionality over time.

DSL NAPs and NSPs need to differentiate the service they offer. Therefore the addition of any new service must have minimal impact to the configuration infrastructure such that high velocity of service introduction can be achieved.

The amount of information that needs to be exchanged between the NAP and NSP to configure and manage service to a customer is minimized and appropriate to the exchange of service. The evolution and enhancement of the services offered by the NSP should not be coupled to the configuration/management methods used by the NAP to configure the user connection. Similarly the evolution and enhancement of the services offered by the NAP do not impact existing NSP services.

[Requirement 9]: Must support multiple transparent NSP configuration channels

The nested architecture of this framework permits individual domain associated configuration mechanisms per NSP[4].

[4] See section 4.3.2 Nested Auto-Configuration Architecture for a more in depth description of "business entities" and "management realms".

[Requirement 10]: Security – Risk Assessment and Implications

Discussed in the "Security Considerations" section.

[Requirement 11]: Use other existing standards wherever possible

All the components in this framework are based on enhancements to existing practice, and existing standards.

[Requirement 12]: Must support initial service provisioning as well as refresh and reconfiguration of services

Automated life cycle management requires that the B-NT may obtain initial configuration from the network at service take, and life cycle changes to service configuration can be synchronized between the network and the B-NT in a timely manner.

[Requirement 13]: Doesn't preclude bulk provisioning

The operational aspects of auto-configuration can be significantly optimized if the amount of overall system configuration that must be performed at subscriber service take is minimized. This includes such aspects as being able to bind service to provider assigned and administered identification tokens (e.g. PPP user I.D.) and minimizing the need for real time synchronization of back office support systems.

3 Introduction & Scope

The DSL Auto-Configuration Architecture is designed to enable timely and effective service activation. Auto-Configuration achieves effective service activation by enabling the Broadband Network Termination (B-NT) to autonomously obtain configuration information from the network and its service provider(s). If an operator has already deployed flow-through service fulfillment (TR-038) then auto-configuration will occur after flow-through has occurred. In this sense these two processes are highly complementary and each process has a distinct emphasis as described below:

Table 1: Focus of Auto-Configuration and Flow-Through

Auto-configuration focus	**Flow-through service fulfillment focus**
• expected to occur after flow-through provisioning is complete • automate configuration of the B-NT • minimize end-user involvement in configuration • emphasis on primary NSP – B-NT for service configuration • generic enough to include several architectural variations	• minimize provisioning delay • automation of business to business interfaces • applies to various business entities providing DSL service • improve visibility into partner systems • efficient flow-through provisioning

This document does not specify the transport protocol to be used across the U interface to configure a specific service or a finite set of services. The transport protocol recommendations and service-specific based object models will be covered in separate working texts. This recommendation does not rely on the presence of signaling in the network. An informative comparison of provisioning and signaling is given in section 4.4.3 of this framework.

4 Document Overview

This document provides a framework for DSL auto-configuration. It describes the layered architecture interactions and associated features/parameters that will be used to automatically configure DSL B-NT. It also describes how progressive stages of auto-configuration inter-relate and the sequence of steps taken to configure the B-NT. Although primarily aimed at the DSL specific environment it is hoped that many of the tools and techniques described will find applicability for use with other broadband access technologies.

4.1 Guiding Principles

The approach to auto-configuration must satisfy a number of principles in order to be useful to the DSL industry. Among the key features required of the auto-configuration framework are:

1. Enables service providers to evolve smoothly from their existing DSL and dial-up approaches in order to leverage existing systems and processes to leverage this infrastructure without necessitating a "fork-lift" upgrade. Hence today's approaches (PPP, RADIUS etc.) must be considered as the starting points for an evolution in capability.
2. Facilitates the configuration of increasingly sophisticated broadband B-NT as new features and functions are added. Hence the approach should be extensible. The framework should also be able to accommodate all flavors of DSL including non-ATM DSL.
3. The framework must span the range of B-NT that will exist in the market from the simple B-NT used only for best effort Internet access through to complex B-NT capable of acting as a residential gateway delivering multiple services (voice, data, video) simultaneously. However, in doing so the framework should not necessitate the use of a full-capability complex-client protocol stack on B-NT aimed at the simple service sets. This is particularly important if the highest volume of deployed DSL B-NT is expected to be in the residential mass-market. The framework should enable the complexity of the required CPE client protocol stacks to grow as the functionality grows to deliver a wider range of capability and value-added service sets.
4. The framework should leverage existing standards and protocols as much as possible and only prescribe use of new approaches where a gap in capability exists. This should enhance time to market by enabling vendors to use protocols that are already standardized and hence understood. It also means that many implementation and interoperability issues will already have been resolved.
5. The framework should facilitate interoperable implementation whilst at the same time allowing sufficient degrees of freedom for vendor differentiation.
6. The framework must easily integrate with both flow-through provisioning (as defined in TR-038) and facilitate prompt resolution of customer service issues.

4.2 Summary of Approach

The auto-configuration framework is based upon the principle of enabling the B-NT to use a succession of protocols in order to gather its configuration information from the network and associated service providers. The B-NT is not required to sequence through every step in the process, only those necessary to gather the configuration it needs. Hence simple B-NT only requires a simple client.

The main stages employed in the example (in figure 3) of an ATM access based auto-configuration architecture framework are: ILMI (TR-037), PPP & DHCP (TR-044) and Automated Configuration Service (ACS) (e.g.WT-064).

The most common and basic service provided over DSL is today is best-effort fast Internet access. Internet Service Providers predominantly use PPP as part of the B-NT configuration process. Where the service is delivered over ADSL or DSL-lite the associated ATM layer (VPI/VCI etc.) is either preset by the B-NT vendor to be compatible with the network environment in which the service provider will deploy the B-NT or it can be manually configured by the end-user. The DSL Forum has worked with the ATM Forum on extensions to ILMI in order to allow these ATM aspects of the service to be automatically configured (layer 2 ATM features plus layer "2.5" via higher protocol identification). Some service providers employ DHCP in the configuration of aspects of IP services. Extensions within DHCP expand the capability of IP service auto-configuration (TR-044). ACS (e.g. WT-064) enables other higher layer configuration information to reside in a repository (which can be distributed). The use of ACS is designed to enable an extensible approach to auto-configuration. An extensible approach provides a repository for configuration information beyond that provided by PPP or DHCP. This can facilitate configuration of more complex B-NT capabilities than PPP and DHCP can currently configure such as for firewalls, NAT and VPN end-points.

A summary of this stepwise approach is shown in the figure below. The sequence on the left illustrates how ILMI and DHCP can help to automate service configuration in the context of IP services.

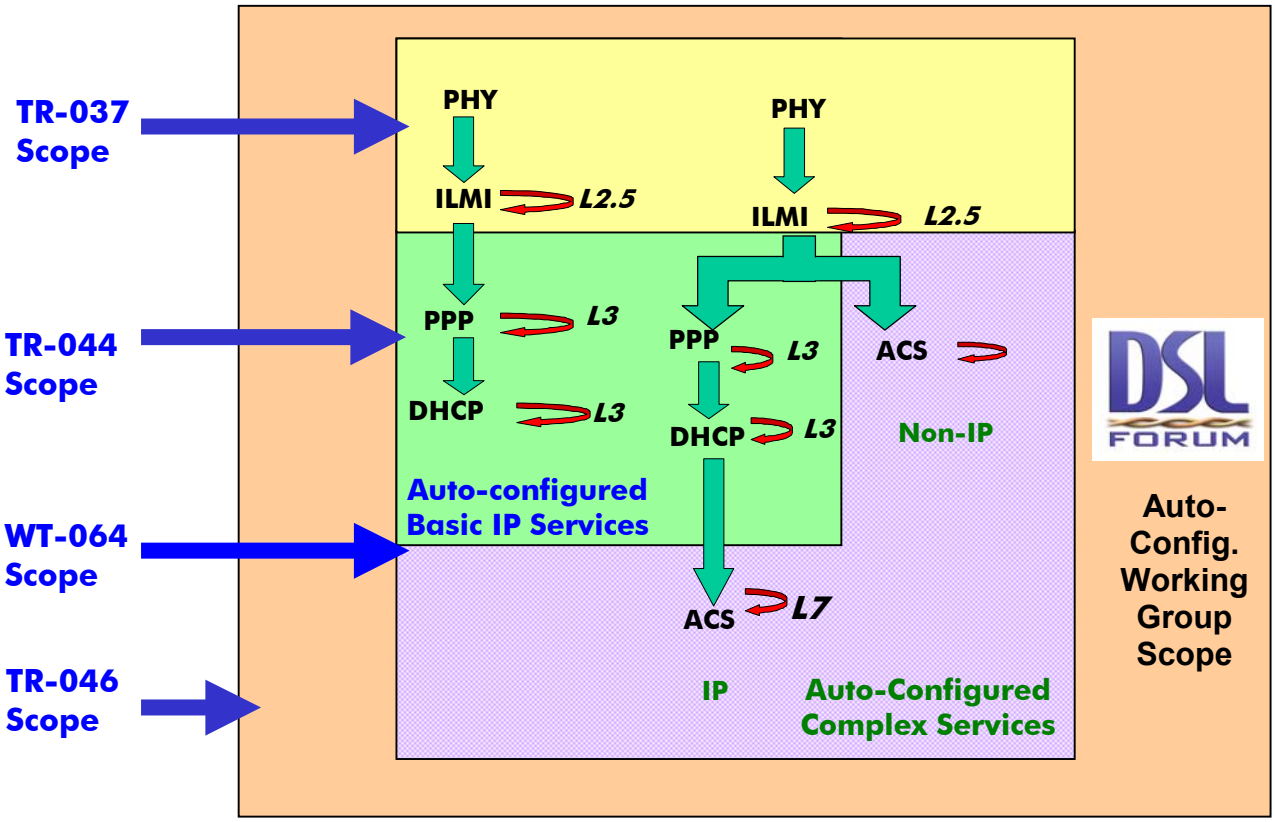

Figure 1: Example of the Stepwise Auto-configuration Approach

The sequence shown in figure 1 illustrates how ACS can also be employed as part of an auto-configuration framework.

4.3 Scope of Auto-configuration of the B-NT

4.3.1 Auto-Configuration and manual configuration

Auto-configuration of the B-NT, as opposed to manual configuration, can be defined as the process of the B-NT autonomously taking actions in order to obtain configuration information from one or more appropriate management realms in the network. This allows the user of the B-NT to make use of the service(s) he has requested previously. Auto-configuration avoids the need for the B-NT user having to configure information manually in order to make use of the requested service. Key advantages of the auto-configuration process are that it makes life easier for the user of the B-NT and minimizes customer service issues caused by mis-configuration.

4.3.2 Nested Auto-Configuration Architecture

Auto-configuration is done in a sequence of management realms. The deeper in the sequence of management realms contacted, the deeper the B-NT can reach in the network. At each stage of the sequence, the B-NT uses a mechanism that is appropriate for that specific management realm. This allows configuration within different management realms to be mutually independent. This independence creates a great deal of implementation flexibility within the framework of the architecture.

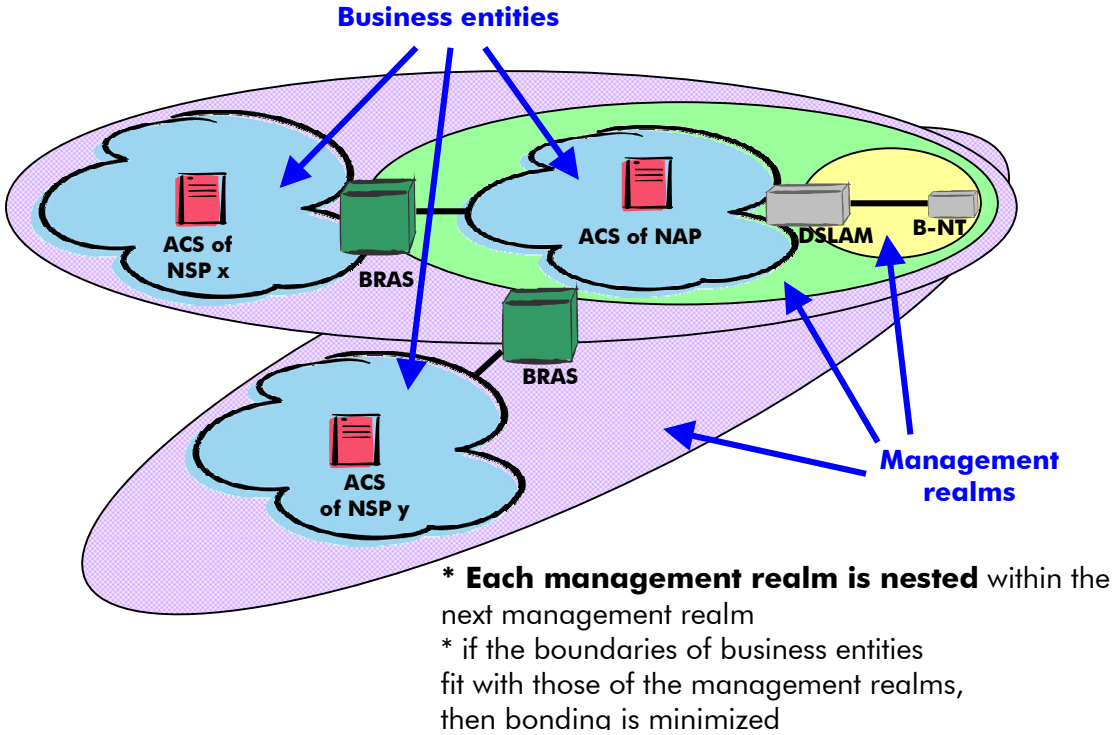

Figure 2: Nested Auto-Configuration Architecture

Based on the previous requirements stated in section 2, two objectives of the architecture can be stated as follows:

- To allow the auto-configuration protocols used within each management realm to be independently appropriate for the services provided by that realm. The only real requirement for the protocol is that it allows access to the configuration parameters required for the services of that realm.
- To minimize the need for bonding — i.e., the need for any management realm to provide configuration information specific to another management realm.

High levels of bonding results in implementation and operational difficulties in flow-through provisioning. If there is not a natural alignment between the management realms and business entities, the need for bonding will increase.

As an example of natural alignment, the regional broadband network provider and the ISP could each have a single management realm. The boundary between the management realms would then be identical to the physical boundary of the regional

broadband network and the ISP network. Each business entity therefore only has to implement the mechanisms that are relevant to the appropriate management realm.

Of course, complete independence between management realms is not possible. It can be desirable to have information of the next stage management realm to bootstrap through the nested structure(s). A table of bootstrap links should be provided in each of the management realms, to enable service branching out of the different management realms. The amount of information in these links should be the minimum to get to the next stage of auto-configuration process.

Although this information will be transferring the boundary between two management realms, it must be clear that configuration information from one realm will only pass a single border.

The concept of separated management realms is graphically summarized in fig 3 below:

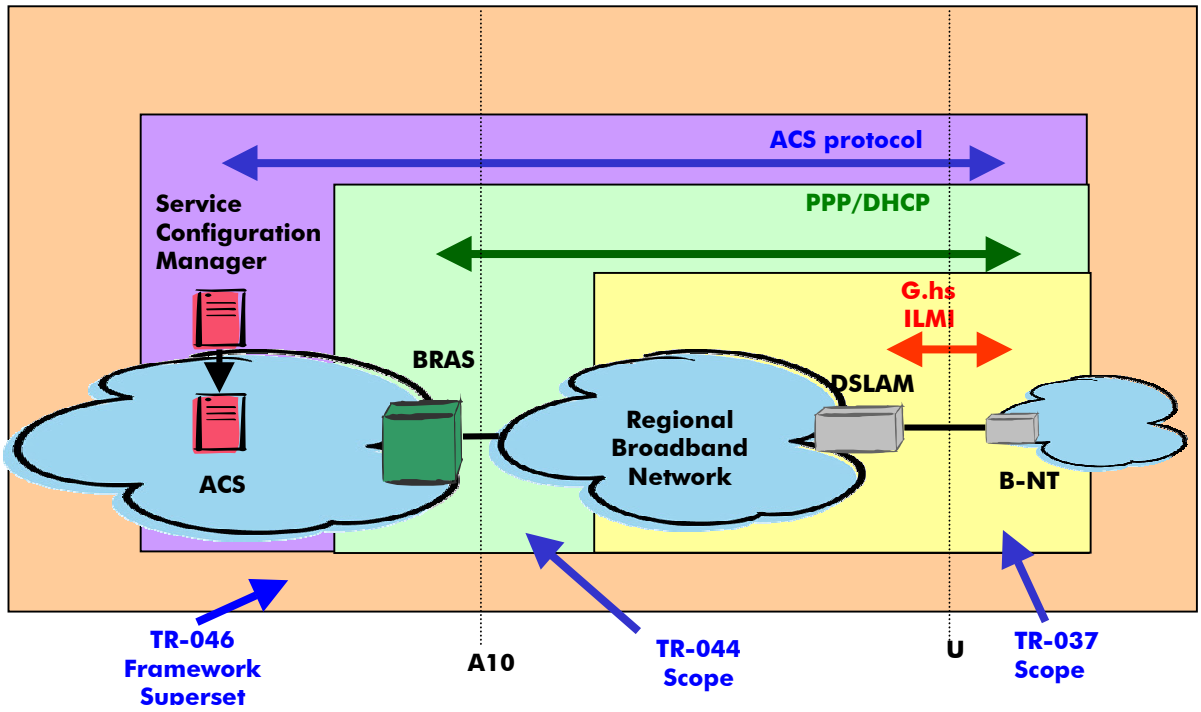

Figure 3: Graphical example of and ATM based auto-configuration architecture with separated management realms and an example of protocol implementation

This architecture is a based on a modular framework. This means that TR-037, TR-044, WT-064 are auto-configuration modules developed for use within the framework. Therefore new WT and TR can be developed for each management realm enabling more future-proof and modular introduction of new auto-configuration protocols and techniques into the framework. For example for purely ATM based services TR-037 is relevant but if no IP parameters need to be configured then TR-044 is not relevant. For non-ATM based access services that are IP based TR-044 is relevant but TR-037 would not be relevant. This approach enables maximum re-use of WT and TR that have been developed within the Auto-Configuration WG.

4.3.3 Roles of Auto-Configuration & Flow Through Service Fulfillment

Auto-configuration of the B-NT is part of a general architecture describing the steps between the first request for a certain service and the service delivery to the customer. In order for a service to be available for a customer, the following needs to be done.

4.3.3.1 Service definition

Services are typically defined by the service provider and offered to the customer. In this case the service provider wants the customers to choose from a fixed and limited number of services that have been pre-defined by the service provider. However, it is also possible for the user to request a customized service that the service provider agrees to deliver. In that case the services will be defined in real-time. In this case the service definition is freely chosen or defined by the customer from a significantly larger set of possible services at the cost of much more fine-grained service definitions. Market forces will influence what the final service definition will look like for fixed and customized services alike. The process of service definition is outside the scope of this document and is mentioned here for completeness.

4.3.3.2 Pre-order and Pre-qualification

This first stage enables the initial viability of service to the end user to be determined. The interactions for this are described in TR-038. Pre-order activities validate location of the service and determine the loop provider organization.

Pre-qualification is used to determine whether a service can be provided at a particular location. Pre-qualification also helps understand the underlying characteristics, which can be used to determine the type of service offered.
In this document these activities are focused on ensuring layer 1 and layer 2 connectivity is available end-to-end. Higher layer (layer3+) validation is considered stage described below.

4.3.3.3 Validation of fixed configuration templates in the ACS (Associated with pre-validation phase of TR-038)

It is expected that fixed configuration templates are pre-validated and placed in the Auto-Configuration Server (ACS). This enables the B-NT to retrieve this information as soon as service activation for the specific B-NT occurs.
Pre-validation typically involves:

- The design of the new service/service bundles that are to be offered.
- Implementation of operational support systems that bill for the service / service bundle (this is outside the scope of this discussion but is important).
- The preparation and validation of generic configuration templates, typically one per service bundle per B-NT capability.
- The pre-positioning of the generic configuration templates in the ACS.

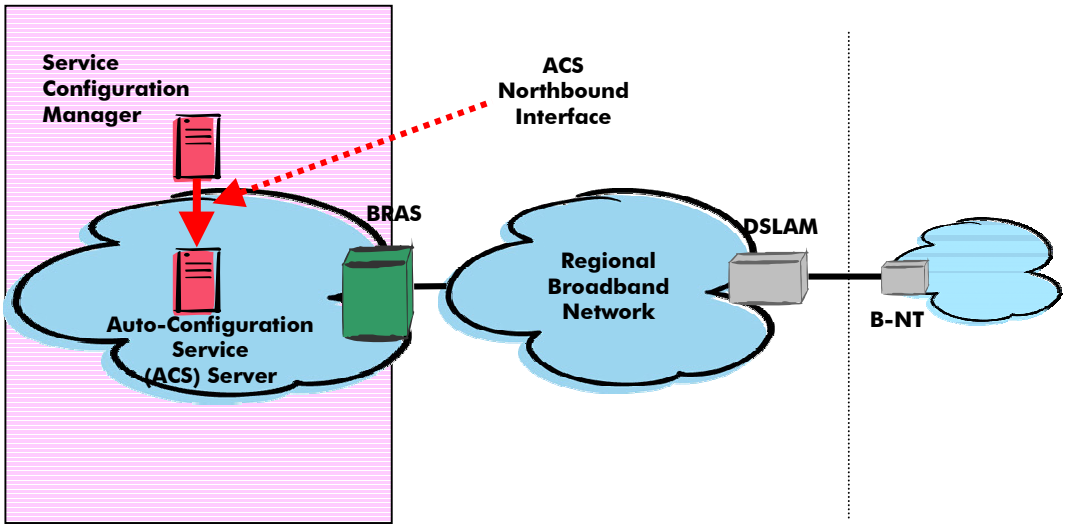

Fig.4 Storage of pre-validated configuration templates in ACS

- Service Configuration Manager populates configuration repository in ACS
- Can evolve gracefully to a more customized environment.

4.3.3.4 Waterfall Flow-Through Provisioning

TR-038 models the information flow interactions between various operational entities in the DSL Service supply chain. An interaction may consist of a pair of request and response transactions between two entities. This means that there is a distinct separation of the information that is sent between the customer, the ISP, CLEC and ILEC. This interaction example with many business entities is shown in figure 5 below. Simpler examples with fewer entities also exist. In this example the customer will request a service from the ISP, typically specifying layer 3 information that is described in the service information model. The ISP will contact the CLEC if necessary in order to request a layer 2 service, specifying layer 2 information. Finally, the CLEC will contact the ILEC, specifying layer 1 information. Each of the parties involved will only send information to the other parties as required [1].

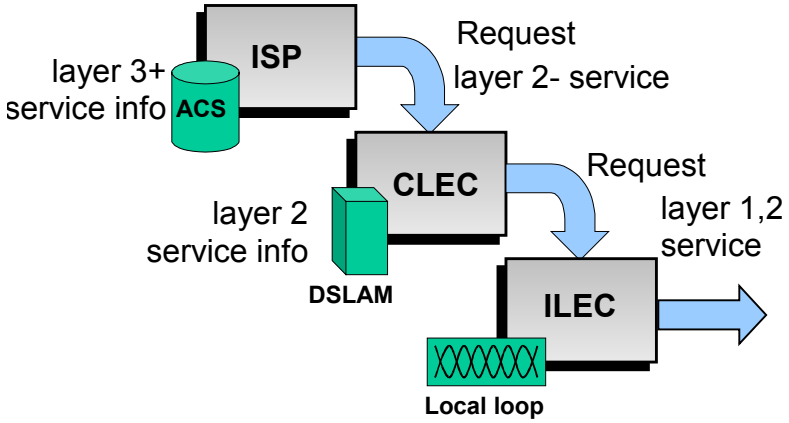

Figure 5: The Waterfall Flow-Through Provisioning Model

In addition in the example of an ATM based access network the initiation of ILMI / TR-037 across the U interface occurs at the end of waterfall provisioning.

For each layer, this data will typically be stored in a different combination of network elements. This action can also be regarded as a configuration action of one of the network elements in the service provider domain. The interface used to store this configuration information, which is needed for B-NT auto-configuration, is called the "ACS-northbound" interface throughout the rest of this document. This process is outside the scope of the auto-configuration architecture but is covered in other DSL Forum documents.

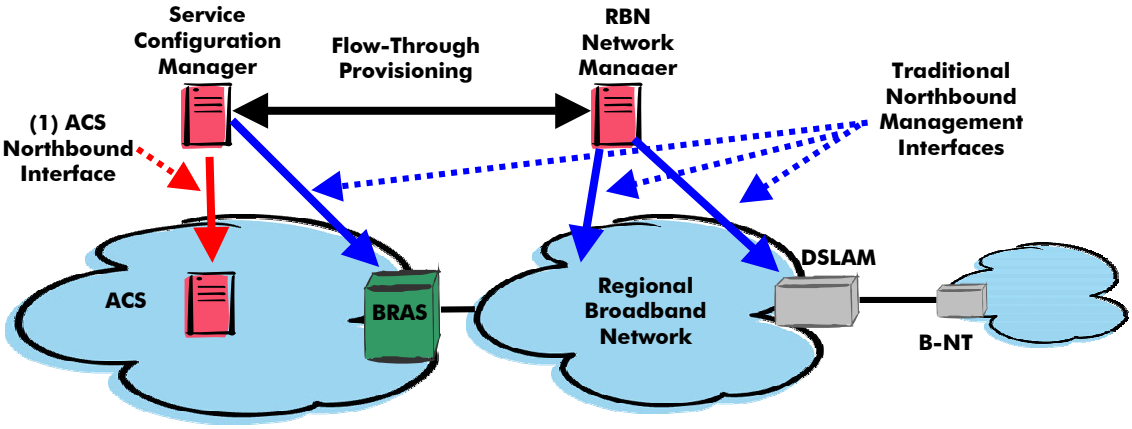

Figure 6: Northbound Interface locations

The main northbound interfaces indicated in Figure 6 above are:

1. ACS-Northbound interface (Auto-Configuration related)
2. DSLAM Northbound Management Interface (Flow-Through related)
3. ATM Switch Northbound Management Interface (Flow-Through related)
4. BRAS Northbound Management Interface (Flow-Through related)

The combination of service negotiation and service installation over the different layers is also called "Flow-through Service Fulfillment". The process is described in TR-038.

4.3.3.5 Initiating Service Configuration

The B-NT needs to be aware when the configuration information is available in one or more of the service provider's servers. Either it has the intelligence to know this itself, or it needs to get informed using some sort of trigger. There are many mechanisms by which the B-NT can know that configuration data is available. This step is an important start to the auto-configuration process, and will be explained in the following sections of this document.

4.3.3.6 Service Retrieval

After the successful conclusion of the flow-through provisioning process, the B-NT needs to retrieve the configuration information that has been prepared by the service provider. This is the auto-configuration process. The interface used to retrieve this configuration information from the ACS is called the “ACS-southbound” interface throughout the rest of this document.

When all these steps have been completed, all the relevant network elements are configured and the customer will be able to make use of the requested service. Whenever a customer alters the requested service, analogous steps will be taken. Also, if for some reason some information, relevant to the B-NT, is changed, it needs to be aware of this and receive the changed configuration information.

In the next sections, the two steps of the auto-configuration architecture are described in more detail.

4.4 The auto-configuration architecture

At each layer of the protocol stack, the auto-configuration architecture will consist of two concepts that need to inter-work. Firstly there is the configuration information that needs to be conveyed to the B-NT. This information will be provided by the flow-through provisioning process and is therefore at the boundary of the ACS-northbound and ACS-southbound interface. In an end-to-end architecture the issues that will arise here are:

- how the data is represented;
- how the data is prepared for the B-NT, this refers to the validation of the configuration templates as previously explained in section 4.3.3.3
- how the data is stored on the "ACS-server" in the service provider domain;

Secondly there is the process and transport protocol that is used to convey the configuration information towards the B-NT. This is also known as the ACS-southbound interface. Issues that must be addressed here include:

- appropriate timeliness of update
- how the data is coded and organized into the chosen transport protocol;

How the configuration information is used within the B-NT itself —leading to service activation— is device specific and is not covered in this architecture. What is important is that service activation should occur when required. If the B-NT wants to change or translate some part of the information, then it should have the freedom of doing so.

At each layer of the protocol stack, these issues can be mapped upon one or more protocols. Throughout the rest of this document, specific protocols for the ACS-northbound and ACS-southbound interfaces are not proposed. Rather, the required functionality that the architecture must (or may) support is described. Recommendations for the transport protocols for the ACS-northbound and ACS-southbound interfaces may be defined in other working texts/technical recommendations of the DSL Forum. It is then up to the service provider (NAP or NSP) to determine which technical recommendations and protocol(s) are preferred at each layer.

4.4.1 Generic data representation in the ACS

In this section, the architecture for representing configuration information is explained. As a first step, the data needs to be represented somehow. Preferably in a way that is independent of the protocol or encoding technique used. In this way it is possible to easily validate and access the configuration information using a particular protocol. If a fixed data store or protocol was chosen, then performing a 'horizontal' mapping onto a different data store or protocol (instead of a 'vertical' mapping from technology independent to technology dependent information) could lead to loss of information or result in non-trivial semantic changes.

Once the configuration data is represented , a mapping can be performed, based upon the specific encoding guidelines of the protocol. The configuration information is then ready to be conveyed to the B-NT, using the chosen protocol.

Configuration data can be represented based upon a number of rules that are high level descriptions of the behavior of the system.

4.4.1.1 Effective data representation in the ACS

An important aspect of auto-configuration is the way the necessary configuration information is represented in the service provider domain. Indeed, this needs to be done in a way that enables service providers to easily extend their data, quickly define new services and reduce incremental costs when defining new services.
For representing the configuration data, an object oriented model is typically preferred. It enables a clean design of the different configuration parameters involved. Parameters can simply be grouped into a number of objects that have object relations with each other. One advantage of this object-oriented design is that it is possible to very quickly define new services, as they can be regarded as 'children' of already defined 'parent' services. This inheritance quality is an important architectural advantage of using a layered object oriented architecture.

Another advantage of this model is that it is possible to standardize parts of the object oriented models, and let the vendors and operators define new objects that inherit the standardized parameters. Service provider and vendors will then be able to quickly create unique parameters and services to enhance these standardized parameters if desired. This rationale is in line with standardization efforts at the IETF [2][3][4][5].

4.4.1.2 Storing data in the ACS

Once the data has been properly represented within the object model, the next step for the operator is to decide which data store is to be used. As there are several possibilities, the operator should consider all options and carefully determine which option would be most beneficial. Note that the choice of configuration repository may influence the protocol to be used to convey the information to the B-NT.

It will often be the case that configuration information, needed by the B-NT to get auto-configured for a certain management realm, will be stored in a different repository associated with another realm. This is the case when different realms will be controlled by different business entities that want to operate with a degree of independence from each other. Therefore, data link layer configuration information will typically be stored in the first device, accessible by the B-NT (i.e. the DSLAM). For network layer configuration, the data repository will most probably be deeper in the network. As a general rule, it can be said that the configuration information needed to auto-configure the B-NT for subsequent management realms can be deeper in the network than repositories containing the information for the previous realms.

The management infrastructure within a particular realm typically invokes a hierarchy of building blocks. In alignment with the Telecommunications Management Network (TMN) building blocks, a representation or abstraction of information would be maintained by a centralized repository, while more detailed information would be stored in a more distributed repository closer to the network elements (e.g. DSLAM).

4.4.2 Preparing the information for the B-NT

There are two models that can be considered as to how configuration information is prepared for the B-NT. They are also referred to as the “data synchronization models”:

1. The static approach;
2. The dynamic approach.

Both are now discussed.

1. The static approach

The static approach suggests that the ACS is a simple repository of pre-positioned and pre-validated service configuration templates (typically one per NT capability definition) and a per-subscriber list of the services relevant to the subscriber’s service bundle. Typically at startup the B-NT is required to obtain knowledge of its basic configuration. When the B-NT learns that it requires configuration update, it simply obtains a standard configuration template from the network appropriate to its capability and the service or service mix currently active. This makes the ACS pretty much stateless at the application layer.

One inference is that configuration templates are pre-validated. E.g. potential conflicts and mis-configuration have been resolved in non-real time and pre-positioned in the directory. There are no race conditions or synchronization issues between the application and the ACS as the B-NT can obtain useful configuration information at any time. The ACS is not required to be a mirror of the ideal NT state.

2. The dynamic approach

The dynamic approach places intelligence in the ACS that is aware of the B-NT service state and is able to modify generic templates and policies uniquely for the B-NT. This inserts a time element between when the ACS becomes aware of a service activation, and when the ACS has a valid configuration available for the B-NT.

The dynamic approach generates the configuration information to be presented to the B-NT in real time. This inserts a slight wrinkle into the "pull" model we have adopted. Some mechanism is required to ensure that the B-NT only activates a configuration that has been validated for it. This could require some "configuration ready" indication, or a polling mechanism from the B-NT that would not succeed until the ACS responded to the poll with a valid configuration.

Both the static approach as well as the dynamic approach must be possible within the framework.

4.4.3 B-NT configuration and service activation

The B-NT configuration is the process of updating[5] in a timely fashion the B-NT parameters or a subset of them (for all relevant management realms) for service delivery.

A pull model is used for the auto-configuration process. In this case, the B-NT (the client) initiates the action of retrieving data from a server (typically not based on a fixed frequency). Note that the devices in the service provider domain will typically be configured using a push model, i.e. network management will initiate the configuration action. The pull model is a clean solution for providing configuration information to devices that are outside the service provider domain. As part of this process, the B-NT needs to authenticate itself before it can access the configuration information. Also, the B-NT has to know when it should perform the auto-configuration process.

The following are some examples of the areas of B-NT configuration covered in this framework:

- The configuration of the link layer. The primary instantiation of this is addressed by TR-037, which provides methods procedures for configuring an ATM UNI. All ATM layer and encapsulation (layer 2.5) parameters must be configured using TR-037.

[5] It includes of course B-NT initialization

- The configuration of devices that do not support a network layer. Specific examples would include 802.1d bridges, or B-NTs that have ATM switch capability.

- The configuration of devices that support IP based services:
  Network layer configuration
  Application layer configuration for intermediate systems
  Application layer configuration for end systems

Where signaling touches auto-configuration

Provisioning of the B-NT involves configuring the device to enable use of a service. Signaling involves those actions that the user (or an application) initiates which allow the service to be utilized. Provisioning constrains the service by changing the state of the B-NT; Signaling involves the transfer of information to or from the B-NT and throughout the access network and the environment at the remote end to allow the service to be used. Management oriented events occur, for example when a user places a new order, and takes place during B-NT initialization.

Signaling oriented events are a result of the user requesting a service on-demand. Launching an SVC is an event of this kind, as the ATM layer parameters are changed as a result of the SVC connection establishment. Service Selection is another type of control event that may require updating the B-NT parameters.

Although the difference between signaling and provisioning is not always sharply delineated (and arguments could be made that instant fulfillment of provisioning could be considered a form of signaling), there are some 'rules of thumb' that allow us to limit the scope of the effort described in this document

Signaling typically	**Provisioning typically**
Is often acknowledged	Is often not acknowledged
Affects and synchronizes resource commitments in the network that are required to use a service	Affects network elements on an element by element basis to constrain the elements use of the service
Has a very short time scale compared to provisioning	Has a comparatively longer time scale than signaling
Is triggered by the user (or the other end point)	Is triggered from a management system
Is required to be available if the service is to be operational.	Can fail, and the service is still operational although its parameters may not be able to be changed.
Sets up a particular use of the service based on the existing constraints set by the configuration.	Sets constraints on the service, including on the signaling.
Is often in-band.	Is often out-of-band.
Often require redundancy and ultra high availability in its implementation.	Has lower reliability requirements in its implementation.
Is within a layer.	Can provision multiple layers.
Modifies the current state.	Defines the states that are possible.
High frequency of change	Low frequency of change

Auto-configuration of the B-NT, as discussed in this document, specifically refers to provisioning of the B-NT from information delivered via the access network over the DSL 'U' interface.

It is important to have a clear understanding of when to use the term provisioning and when to talk about signaling. For a DSL-based TV-like service, setting up the ATM parameters and filters (i.e. parental controls) for channel changing could be considered configuration. However, the act of changing the channels in real time is a signaling action.

This recommendation specifically excludes configuration of the B-NT over the S/T interfaces local to the user's premises. However, the interfaces based on this framework do not preclude use of such interfaces for provisioning the B-NT. The following figure illustrates the scope of auto-configuration[6].

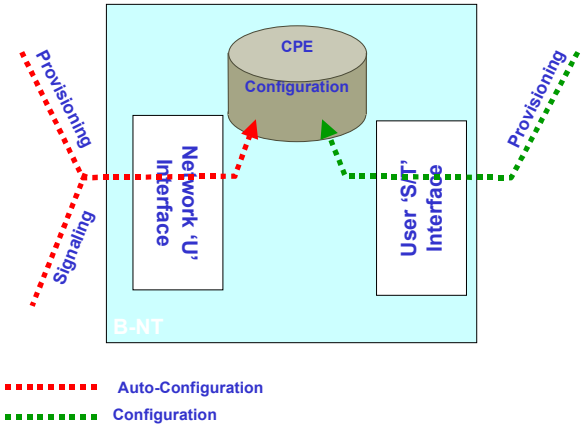

Figure 1 Scope of Auto-configuration

.

4.4.4 Protocol requirements for B-NT configuration

At each layer in the protocol stack, a protocol needs to be used by the B-NT to retrieve the configuration information that will lead to the activation of the customer's service(s).

Because the lower layers have been setup previously, the higher layers can use the services offered by them. This will impact the choice of protocol. For instance, if no link layer connectivity exists, it is not possible to retrieve configuration data from a server that is deeper in the network. On the other hand, if network layer connectivity is established, it is possible to make use of this when contacting a server that is located at a remote service provider (offering a service on top of this already existing network layer service).

[6] Configuration from the user premises could also be considered as a configuration process that is triggered by a management event.

The general requirements for the protocol(s) to be used for configuration retrieval are the following:

- The protocol must allow the B-NT to request the configuration data and confirm successful retrieval or give an indication of any error during the process. This is known as a "pull" model.
- The protocol may allow the B-NT to send information regarding its capabilities to the configuration server.
- The protocol must allow the configuration server to send an asynchronous notification of configuration updates to the B-NT.
- The protocol may support secure communication between the B-NT and the configuration server (e.g. authenticated or encrypted). Some services or realms may require this.

Some of these requirements will be more important than others, depending on the management realm. For instance, encryption of data between the B-NT and the DSLAM for layer 2 data will be less critical than encryption of data, crossing the Internet in order to provide the B-NT with service information from a remote service provider. Depending on the determined use of the protocol, a detailed analysis needs to be made by the provider and B-NT vendor.

4.5 Service Management Models for the B-NT

The basic service deployment scenario for a service can be either NAP centric or NSP centric depending on the relationship between the providers. Service Management Models for the B-NT overlay on the basic service deployment scenarios the following sources of information to allow provisioning of the B-NT:

1. NSP Centric Service Configuration
2. NAP Centric Service Configuration
3. Third Party Service Configuration
4. Application Provider Centric Service Configuration
5. User Centric Service Configuration

The NSP centric and NAP centric models are explicitly covered by this work. Third Party Service Configuration occurs when configuration of the B-NT is based on information provided via a third party and not by the NAP or the NSP. The Application Provider centric model is a particular case where the applications at the far end of the network provide the provisioning information required to configure the B-NT. User Centric Service Configuration is the case when the user configures the B-NT. This last case is excluded from this framework, since we concentrate on configuration mechanisms that run, from the network, over the U interface. Different service management scenarios can coexist, i.e. the basic IP address info comes from the NAP or NSP while the remaining configuration of B-NT is handled by local applications.

Service Configuration Model of the B-NT	**Description**
User Centric Service Configuration	In this model of the service management of the B-NT, the user is largely responsible for the configuration of the B-NT functions above those needed to establish the connection. Configuration of Router or Bridge parameters, Security attributes of functions such as Firewalls, or other aspects of the service are set from applications that run within the users premises. *The precise nature of these applications is outside the scope of the auto-configuration of the B-NT from the network.*
Application Provider Centric Service Configuration	In this model of service management of the B-NT, applications at the far end of the network from the user's premises manage the service functionality on the B-NT. Such a service management model could occur when the B-NT is used to connect to a central corporate environment from a branch office. In this case the B-NT may be completely controlled by central management tools of the corporate entity. Once the connection is established, in-band management tools may be used to manage the services on the B-NT remotely over the access the network. After the connection is established neither the NAP nor the NSP are involved in the configuration of the services on the B-NT.
NAP Centric Service Configuration	In this model Service functionality is configured on the B-NT via an application that is resident in the NAP. Once the connections are configured between the B-NT and the Network, the B-NT can interact with a NAP resident management application that configures the service functionality. The NAP may gather information from either NSP or Application Providers, via network management bonding interfaces, to configure the services on the B-NT.
NSP Centric Service Configuration	In this model, service functionality is configured on the B-NT via applications that are resident in the NSP. Once the connections are configured between the B-NT and the Network, the B-NT can interact with an NSP resident management application that configures the service functionality.
Third Party Service Configuration	A specialized NSP provides a configuration service that the B-NT interacts with to receive the configuration of the services on the B-NT. This option can be thought of as a specialized version of an NSP Centric Configuration. What makes this service configuration model unique is that the third party is a service provider dedicated solely to Service Management.

The auto-configuration models described in this framework cover all of these models except the "User Centric Service Configuration" which is out of scope for this effort.

4.6 Optional functions supported from the ACS

Other functions that an ACS could provide, depending upon the business model are image management and B-NT vendor and model identification. This could be important because over the lifecycle of the B-NT, depending on the ownership model, the Service Provider may need to update and maintain the firmware and software of the B-NT for an indefinite period of time.

5 Glossary

This document makes use of terms defined in DSL Forum TR-012. In addition to those definitions, this document uses the following terms:

5.1 Access Protocol

An access protocol is all of the encapsulation and layer 2 protocols necessary to access a service.

5.2 Connection

In this recommendation, a "connection" is an ATM virtual channel connection from the B-NT, through the NAP, to the first-level aggregation point in a network including any access protocol.

5.3 Service

In this document, a service is any layer 3 protocol capability that exists above the connection that is required to provide access to an NSP. The term "service" in the context of this document implies layer 3 service connectivity.

6 References

[1] Umesh Bellur, Krishna Garimella, "Automating the ISP to CLEC Interaction of the xDSL Supply Chain", DSL Forum contribution adslforum99.200, August 1999
[2] "Policy Core Information Model -- Version 1 Specification" RFC 3060, B. Moore, E. Ellison, J. Strassner, A. Westerinen, February 2001
[3] "Policy Core LDAP Schema", Internet Draft, draft-ietf-policy-core-schema-11.txt, J. Strassner, A. Westerinen, E. Ellison, B. Moore, R. Moats, May 2001
[4] "Policy QoS Information Model", Internet Draft, draft-ietf-qos-policy-info-model-03.txt, Y. Snir, Y. Ramberg, J. Strassner, R. Cohen, April 2001
[5] "Policy Terminology", Internet Draft, draft-ietf-policy-terminology-03.txt, A. Westerinen, J. Strassner, S. Herzog et al, April 2001