AdventNet
Excellence Matters

White Paper

Element & Network Management With

Transaction Language 1 – TL1

Krishna Sanjeeviah
Neil Butani

TABLE OF CONTENTS

Transaction Language 1 – TL1

1.0 TL1 Background

During the mid 80s, Bellcore embarked on specifying a standard man-machine language to manage network elements (NEs.) It was to be based on Z.300 series Man Machine Language (MML) standards. It was designated Transaction Language 1 or simply TL1. This effort encompassed specifying a language as well as message set(s) for managing a variety of Telecom equipment from different domains. During that time, Bellcore was also developing a fault management OSS (Operations Support System) called Network Monitoring and Analysis[1] (NMA) for the RBOCs. As it happened, TL1 was chosen as NMA's element management protocol. Consequently, for NEs to be NMA manageable, TL1 support became mandatory. TL1 made its foray into carrier networks well over a decade ago and during the subsequent years, TL1 deployments became widespread in the RBOC space. Traditional circuit switches were the only ones to not embrace TL1 and continued to operate their proprietary interfaces. Today TL1 is a management protocol of choice for the majority of SONET and access infrastructure in North America.

1.1 TL1 Strengths

TL1 is human decipherable protocol that can be easily guised as a machine-to-machine language. It is a multi-vendor and multi-technology protocol with comprehensive FC(A)PS[2] support. Processes for design and implementation of TL1 interfaces are well established and are relatively straightforward. Since TL1 is human readable, special decoders or debuggers are not necessary and has become the preferred CLI (command line interface) for NE vendors. Finally, TL1 can support next generation network elements and the ever-increasing scalability and extensibility needs.

---

[1] Other OSSs include OPS/INE (Operations Provisioning System/Intelligent Network Element) and TIRKS (Trunk Integrated Record Keeping System) for inventory management and provisioning
[2] Fault, Configuration, Accounting, Performance, Security

1.2 TL1 and OSMINE

One of the challenges NE vendors who target the RBOCs face is obtaining the OSMINE[3] certification. This is a process to ensure that their equipment can reliably interwork with Telcordia[4] OSSs in the targeted service provider network(s). OSMINE is a service offered by Telcordia and can cost NE vendors well over a Million dollars and take over a year to complete. OSMINE comes as an unwelcome surprise for some, especially for those who come from the IP (data) domain. Considering the fact that successful completion of OSMINE is just a stepping-stone to getting access to the golden ring of ILEC market, the associated opportunity cost is pretty high. During the CLEC hey day, it was not uncommon for NE vendors to forego OSMINE efforts (it is worthwhile mentioning some were not even aware of such a process) and pursue the growing CLEC market. However, since the collapse of the CLECs, more NE vendors are embarking on OSMINE. During 2001, established NE vendors like Cisco, Ciena, Fujitsu, Lucent, Tellabs, and Nortel made significant investments on the OSMINE front. The total investment from these companies alone was over 100 MM. Startups like Metro-optix, Turin Networks, Movaz Networks, Actelis Networks and Astral Point (now a part of Alcatel) have undergone OSMINE certification. Many others are in the process.

The OSMINE process has a myth surrounding it. Often NE vendors are led to believe "OSMINE compliance" of management software without context of deployment. Being an independent software vendor, it is not surprising that we are often asked the question – *are you OSMINE compliant?* As clearly specified by Telcordia, the ecosystem determines OSMINE compliance - specifically, the incumbent Telcordia OSSs with which the service provider manages NEs and services.

For example, if NMA is the fault management OSS used by the service provider, NEs need to report alarms and events to NMA in a format mandated by the NMA system. NMA will also issue queries to NEs. So, the NEs need to be able to understand how to interpret and respond to such queries. In case of TL1, it translates to supporting autonomous messages, TL1 commands and responses as defined in NMA system requirements.

---

[3] OSMINE – Operating Systems Modification for the Integration of Network Elements
[4] Telcordia was formerly Bellcore

Telcordia is the single authority to determine the scope of work for OSMINE certification and to subsequently endorse a product as OSMINE compliant.

Quite often, majority of the effort is in updating OSSs to support new NEs. Telcordia identifies the OSSs impacted by the introduction of a new NE and updates the impacted systems so that the service provider can manage the new NE with its existing systems.

Independent software vendors cannot ascertain OSMINE compliance in any way. OSMINE to some extent is an on-going process. NE vendors who intend to deploy at multiple service provider networks, may be required to undergo OSMINE process multiple times. This could also be the case when a new Telcordia OSS is introduced into an existing service provider network. The scope of OSMINE efforts is also determined by "what" needs to undergo OSMINE. In some cases, there is a need for the EMS supplied with the NE to undergo OSMINE as well since the EMSs (Element Management Systems) and NMSs (Network Management Systems) interact with OSSs.

2.0 TL1 and Other Protocols

Attempts have been made to replace and/or bypass TL1. This section highlights some of the key efforts in that direction. Some of the potential alternatives have emerged independently as well.

2.1 CMIP – Common Management Information Protocol

ITU-T endorsed CMIP[5] was developed in conjunction with the TMN (Telecommunications Management Network) information models to facilitate a common protocol, structure and naming convention that would allow vendor transparent network element integration. CMIP was one of the protocols that emerged from the Open Systems Interconnect (OSI) initiative. Purportedly, this would obviate the need for OSMINE. However, there was not even a single widely deployed OSS that was pinned to CMIP and there was really no pressing need to support it. Besides, the OSI initiative did not prosper as anticipated and TMN and CMIP turned out to be somewhat valiant efforts in the Americas.

2.2 CORBA – Common Object Request Broker Architecture

The industry turned to CORBA after TMN efforts failed. Since it was based on an object model to exchange information between different entities, standardized interface definitions became a critical factor. Lack of industry accepted interoperability suites hindered its adoption to some extent. In the OSS community however, CORBA has been used widely for interoperable management solutions.

2.3 XML – Extensible Markup Language

XML, like TL1 is a human readable ASCII language that enables separation of content of data from its presentation and describes the content in terms of the type of data (unlike HTML which describes the content in terms of how it is to be presented.) XML can be used as a protocol neutral interface to interconnect technologically disparate management systems. XML can be leveraged heavily in interconnecting NEs utilizing different protocols like TL1 and SNMP with network management systems and OSSs. In the Element and Network

---

[5] CMIP unlike TL1 is a machine to machine language

Management arena, XML is emerging as a popular protocol that enables seamless integration.

2.4 SNMP – Simple Network Management Protocol

SNMP has emerged as a popular protocol for managing data centric NEs. Like CMIP, SNMP is a machine-to-machine language. SNMP as the name implies is based on simple get-set messaging. With this simplicity comes lack of sophisticated operations capabilities. SNMP dominates the IP domain and is popular in applications where the management process is simple. Management of enterprise network elements can be performed easily with SNMP. However, SNMP has not been able to displace TL1 although it is common for many NE vendors to support both protocols. Some NE vendors who started off by supporting SNMP (and not TL1) are recognizing the need to support TL1 especially when selling to ILECs. Some vendors choose to use TL1 adapters instead of supporting TL1 natively in their equipment.

3.0 TL1 Operation Fundamentals

Fundamentally, TL1 comprises of command sets (input messages), acknowledgements, responses and autonomous messages. TL1 commands and messages conform to a common syntax. The following table outlines some of the TL1 command set terminology.

Terms	Description
TID	Target Identifier specifies the target NE
AID	Access Identifier specifies the entity on which to perform the command. The entity can be equipment (shelf-slot) or facilities related (Circuit number)
ATAG	A sequence number to track messages from the NE to the manager
CTAG	Correlation Tag is used to associate responses with commands
PID	Private identifier is used to hold the password field

Each command consists of a command code, a staging block, a general block and one or more payload blocks. The command code is composed of a verb and one or two modifiers. Example: ed-eqpt-all – *ed* is the verb which implies edit, *eqpt* is the first modifier which implies equipment and *all* is the second modifier which could imply all the shelf-slot combinations. (It should be clear to the reader why TL1 is a man-machine language.) The staging block comprises of fields to specify the details of the network element. TID, AID and CTAG are a part of the staging block. The general block is usually optional and can be used for additional information. The payload block carries any further information necessary to process the command.

3.1 TL1 Command

A TL1 command is issued to a targeted network element from an OSS, NMS, EMS or craft terminal. The resident TL1 agent[6] in the NE initiates an operation in the NE based on the command. The following example depicts the act-user command:

---

[6] NEs that support TL1 natively have a resident TL1 agent. If not, the commands are funneled to a gateway that mediates between the NE and northbound systems.

Example:

```
ACT-USER:NE-apollo:Gates:1234::passwd;
```

tid: NE-apollo – identifies the NE

aid: Gates – identifies the user name

ctag: 1234 – correlation tag

―

":"General Block can be empty but required

pid: passwd - password

3.2 Acknowledgement

TL1 commands are received by the resident agent inside the NE which acknowledges the command. The acknowledgement contains a code to notify the issuer if the command was rejected or being acted upon. It also contains the CTAG to let the user associate the acknowledgment with a command. The following table outlines some of the TL1 acknowledgment codes.

Terms	Description
IP	In Progress
OK	All Right
NA	No Acknowledgement – Can be used when the command is garbled
NG	No Good – Requested action conflicts with current configuration
RL	Repeat Later – Requested action cannot be carried on at the time because of resource constraints etc.

Example:

IP 25

Acknowledgement Code – IP

25 – CTAG

3.3 Response

The agent takes the necessary action(s) and sends a response back to the issuer. The response contains information on whether the command executed successfully. The format of the response comprises of a header, response identification, text block and termination. The header contains the system identifier and time and date stamps. The response identification contains the letter M (indicating it is a response) the CTAG and a completion code. The following table outlines some of the TL1 condition codes.

Terms	Description
COMPLD	Completed successfully
DENY	Denied
PRTL	Partial Success
DELAY	The command execution is delayed
RTRV	Large amount of data needs to be retrieved

```
Example:
GSR12-4 98-01-01 05:03:05
M 12345 COMPLD

GSR12-4 – System ID
12345 - CTAG
```

3.4 Autonomous Messages

Autonomous messages report alarmed or non-alarmed trouble events, PM (Performance Monitoring) data, changes in NE's database etc. The autonomous message contains a header, auto id, text block and terminator. Events and alarms are often conveyed to Telcordia OSSs such as NMA for correlation and trouble ticketing. The header contains the system identifier and time and date stamps. The auto id contains the severity of the alarm, ATAG, verb and modifier. The following table captures the TL1 alarm severity codes. ATAG allows for tracking of autonomous messages. The verb and modifier capture the nature of the autonomous message. The text block contains additional information on the autonomous message.

Terms	Description
*C	Critical Alarm
**	Major Alarm
*^	Minor Alarm
A^	Non-Alarm Message

```
Example:
GSR12k 99-01-01 06:06:06
A 123 REPT ALM
"SP:MJ,INT,NSA,01-01,06-06-06,NEND,NA:\"Disk Full\""
GSR12k – System Id
A – Non Alarm Message Severity Code
REPT ALM – Verb Modifier
```

3.5 TL1 NE Topology

A TL1 NE can serve following functions – Gateway, End-point or Intermediate. A NE operates as Gateway NE or GNE when TL1 commands destined to another NE can be forwarded through the GNE. An End-point NE or ENE processes TL1 commands addressed to it. An Intermediate NE or INE simply acts as a conveyor of TL1 messages.
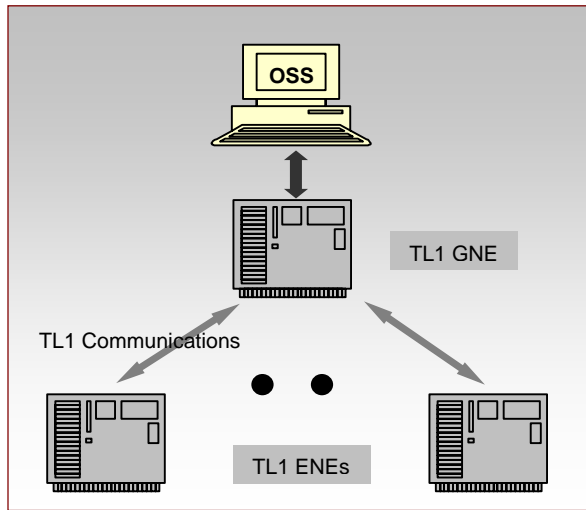

3.6 TL1 Mediation

TL1 mediation is often necessary for service providers to mediate between NMSs/EMSs/OSSs and TL1 NEs. In such cases, XML is a popular northbound interface. There is also a necessity for service providers to manage NEs that don't support TL1 natively with TL1 centric OSSs like NMA. In such cases a TL1 gateway is used at the EMS layer.
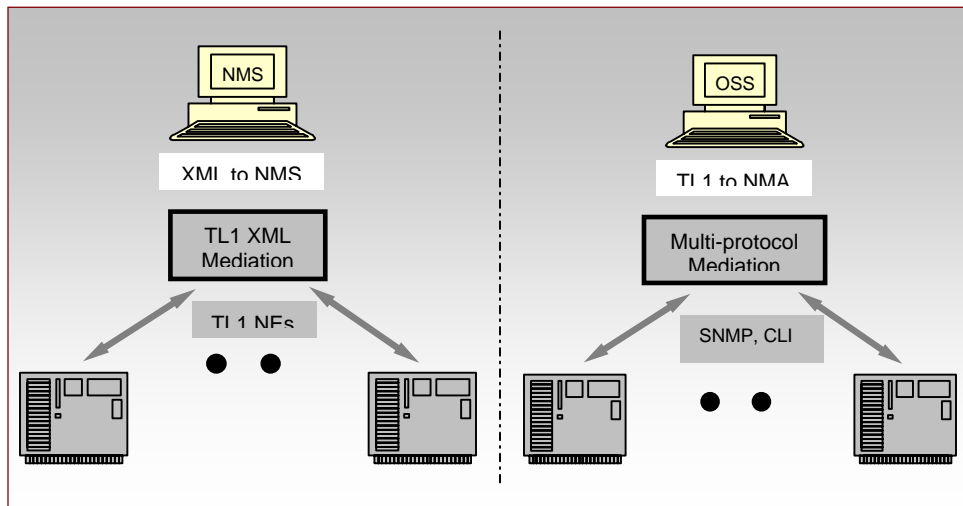
4.0 AdventNet TL1

AdventNet offers a portfolio of TL1 products ranging from Web NMS, an EMS/NMS platform with comprehensive FC(A)PS support to embedded multi-protocol agents and tools. Open standards based architecture allows easy integration into existing systems, applications and processes. Over one hundred EMS/NMS applications have been implemented using Web NMS.
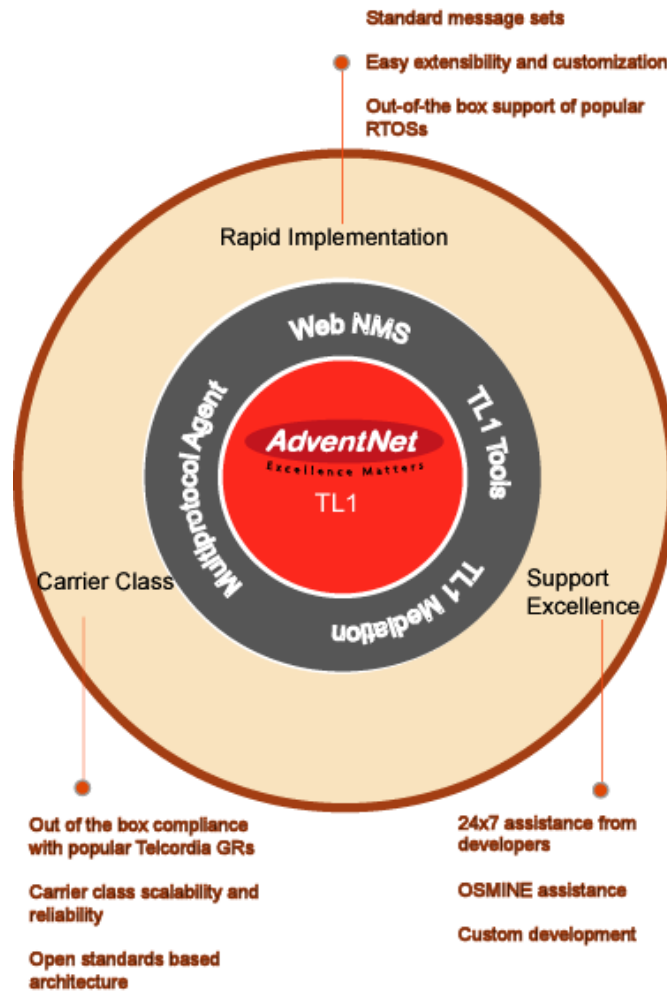
AdventNet's TL1 solutions have undergone OSMINE certification with widespread deployment in large networks including RBOCs and other carriers like Sprint.

4.1 AdventNet TL1 Overview

AdventNet's end-to-end TL1 support includes EMS/NMS platforms, mediation server, management APIs and embedded agents to facilitate easy integration of Network Elements (NEs) and management systems into carrier infrastructures. For Telecom equipment that needs to support TL1 natively, AdventNet offers a choice of embedded agent toolkits that enable rapid implementation of TL1 support on the device.

4.2 AdventNet TL1 Strengths

AdventNet offers a compelling set of TL1 solutions by bringing a wide range of business and technical merits through seasoned network management experience. Some of these values include:

Business Values

1. **Usability:** With a complete support of the TL1 standards such as GR-833, GR-199 and GR-835, AdventNet TL1 usage is across any technology or network deployment. NEs and management systems can be enabled with TL1 functions rapidly with no re-engineering efforts.

2. **High-availability:** Well-entrenched in the networks around North America, AdventNet TL1 solutions have proven their reliability and scalability in large carrier environments. Currently, RBOCs (SBC and others), ILECs (Sprint and others) and other carriers have successfully deployed AdventNet TL1 Solutions

3. **Improves Network Profitability:** AdventNet TL1 solutions hold the key to streamlining operations and increasing profitability from network resources.
    - Integrated network management and service creation
    - Simplifying operational complexity in multi-service, multi-vendor networks
    - Easy migration to next generation networks and technologies

Technical Values

1. **Completeness:** End-to-end TL1 support from embedded agent to EMS/NMS systems and mediation

2. **Performance:** AdventNet TL1 solutions can easily scale to handle the performance needs of large carriers. AdventNet TL1 architecture is based on transport provider framework that can enable user-based protocol out-of-the-box.

3. **Testability:** TL1 NEs can be simulated with AdventNet simulation tools for a thorough validation of core functionality before entering Lab trials. This enables development teams to deliver tested solutions and reduces the burden of discovering issues in the field.

4. **Multi-platform Support:** AdventNet TL1 agent products support popular RTOSs like VxWorks, OSE, Linux and other RTOSs. Web NMS, Mediation Server and API products support Windows 2000/NT, Solaris, Linux etc.

5. **Built in commands:** SONET and ATM specific commands are supported out-of-the-box to accelerate development.

4.3 AdventNet TL1 Management

There are two dimensions to the management side of TL1 - the ecosystem and management applications. The ecosystem comprises of OSSs on the northbound side and Network Elements on the southbound side. Management applications comprise of Service management functions like Fault management, Configuration Management and Performance management.

AdventNet's solution provides an open interface in the northbound side to interface with popular Telcordia OSSs like TIRKS and NMA. Support for multi-level pass-through feature to bridge OSSs and NEs and GNE management are supported out-of-the-box. AdventNet offers rich FCAPS functionality with out-of-the-box support for standard TL1 command sets and messages for SONET NEs, Multi-service platforms etc.

4.4 AdventNet Multi Protocol Agent (MPA)

Embedded agents reside within an NE and serve as a peer to the manager. The agent processes TL1 commands from the manager and sends acknowledgements, responses and autonomous messages. Out-of-the-box support for standard TL1 command sets compliant with Telcordia definitions and

easy extensibility to support custom commands accelerates agent implementation. MPA conforms to GR 831, 835, 815 standards. Security functionality is implemented as per GR 815 & 835 Standards. GNE functionality is supported as per GR199.

MPA offers Telecom NE vendors a common instrumentation for multiple protocols like TL1, SNMP and HTTP. It is portable across a wide variety of platforms. For applications where TL1 is the only southbound protocol used to manage NEs, AdventNet TL1 agent can be used instead of the MPA.

4.5 AdventNet TL1 Mediation

AdventNet's multi-protocol mediation server helps manage heterogeneous networks [IP, Cable, Optical, DSL] where multi-protocol support is needed. The mediation server offers a protocol neutral XML interface to northbound systems while supporting popular southbound protocols like SNMP, TL1, CLI and CORBA.

4.6 AdventNet TL1 Tools

AdventNet's TL1 tools include the simulation toolkit to simulate TL1 NEs, message builder, compiler, browser and TL1 libraries to provide TL1 south bound interface.

4.7 AdventNet TL1 Standards Support

AdventNet TL1 solutions support the following Telcordia standards:

1. GR-199-CORE: *Operations Application Messages - Memory Administration Messages - Issue September 5, 2002*

   GR-199-CORE provides the messages and data dictionary information required for the Memory Administration functions for Transport, including Synchronous Optical Network (SONET) — and switching, including ISDN and Advanced Intelligent Network (AIN) NEs.

2. GR-831: *Operations Application Messages - Language For Operations Application Messages – Issue November 1996*

GR-831 provides Transaction Language 1 (TL1) messages used in the input and output between Operation Support Systems (OSSs) and Network Elements (NEs) for Operations domains such as surveillance, memory administration, and access and testing functions between the OSS and NE.

3. GR-833-CORE: *Network Maintenance: Network Element and Transport Surveillance Messages – Issue June 2000*

    GR-833 presents TL1 application messages for NE and transport surveillance functions

4. GR-835: *Network Element and Network System Security Administration Messages - Issue January 1993*

    GR-835 provides TL1 messages related to the management of security parameters, messages related to security log management, and messages related to querying security status.

5. GR-253-CORE: *SONET Transport Systems: Common Generic Criteria Messages - Issue 4*

Appendix 1 – Solution Snapshot: TL1 OSS Integration
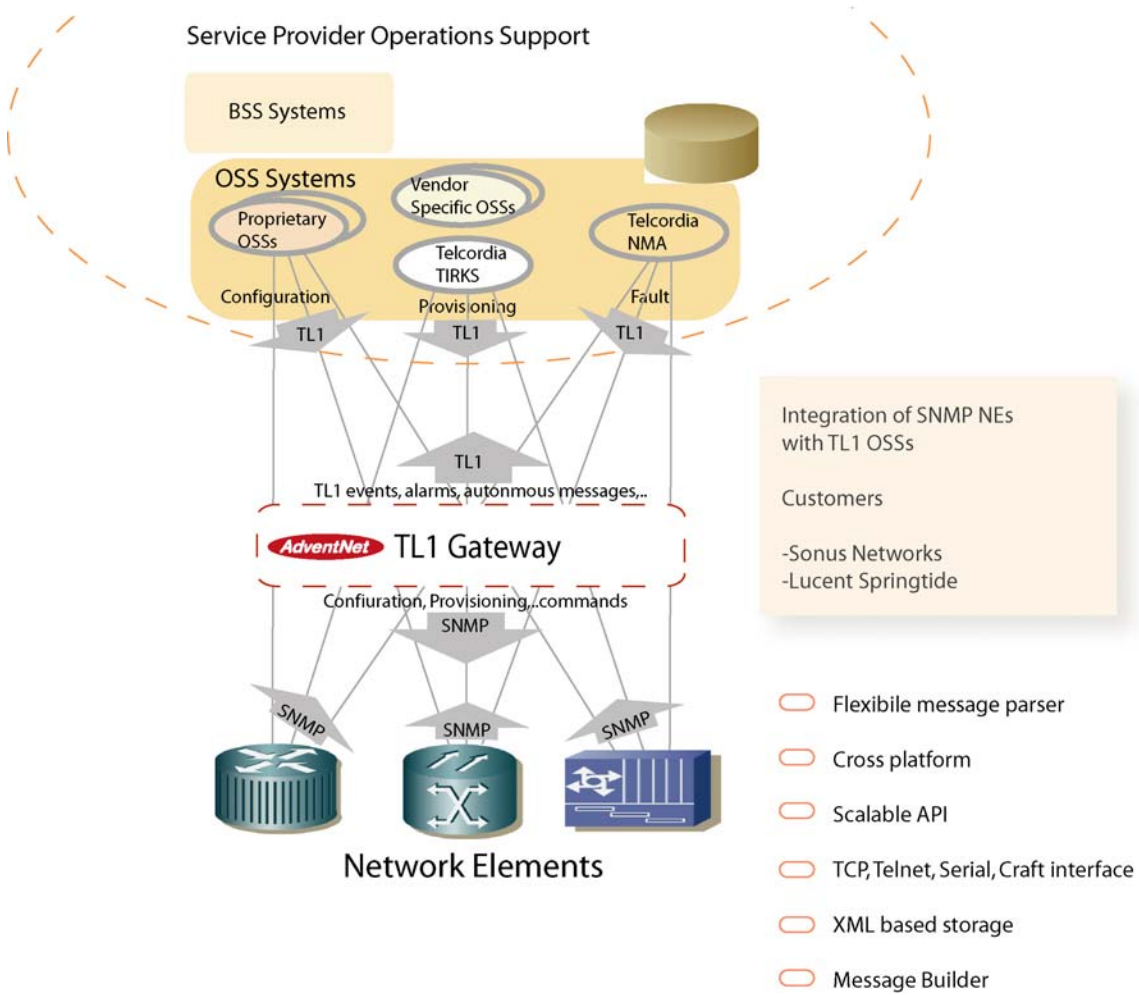
Appendix 2 – Solution Snapshot: TL1 NE Enablement