Data Gathering in Optical Networks with the TL1 Toolkit

Ronald van der Pol[1] and Andree Toonk[2]

[1] SARA Computing & Networking Services, Kruislaan 415, 1098 SJ Amsterdam,
The Netherlands
`rvdp@sara.nl`
`http://nrg.sara.nl/`
[2] The University of British Columbia, 6356 Agricultural Road, Vancouver,
BC Canada V6T 1Z2
`andree.toonk@ubc.ca`

Abstract. This paper describes a new tool for gathering (performance) data from optical network devices. During the last few years many National Research and Education Networks (NRENs) have deployed hybrid networks. These hybrid networks consist of a traditional routed IP part and an innovative optical part. On the optical part *lightpath* services are offered. These lightpaths are high speed (up to 10 Gbps) circuits with a deterministic quality of service. Not all optical devices used in these networks have full SNMP support, so the traditional management and monitoring tools cannot be used. Therefore we designed and implemented the TL1 Toolkit which supports management, data gathering and monitoring of optical devices.

1 Optical Networks and Lightpaths

In recent years many NRENs have acquired their own dark fiber infrastructure. The dark fibers are lit with DWDM and SONET/SDH equipment in order to build a circuit switched network that can be used for either transporting traditional IP traffic or so called lightpaths. Lightpaths are transparent point-to-point circuits with deterministic quality of service properties.

These optical networks provide researchers with new possibilities, but there are also significant research efforts in the infrastructure itself. The circuit switched nature of these optical networks is very different from the packet switched nature of the internet. New technologies are being developed to monitor the health of the lightpaths. Another topic of research is investigating which (performance) metrics are interesting to investigate in the optical networks. Section 4 discusses some of these new metrics.

Lightpaths are used by various research projects. Some examples are given in the next sections.

1.1 LHCOPN

The new Large Hadron Collider particle accelerator of CERN in Geneva produces huge amounts of data. This data is distributed to 11 major storage sites in

Europe, North America and Asia. These sites are connected to CERN and each other with 10 Gbps lightpaths. This network is called the Large Hadron Collider Optical Private Network (LHCOPN).

1.2 eVLBI

eVLBI (electronic Very Large Baseline Interferometry) uses lightpaths to connect several radio telescopes to a central site, where the signals are combined. This creates one big virtual telescope with a much higher resolution than each of the individual telescopes. This can be done real-time, which is a big advantage. Previously, this was done by storing the data on disk or tape and shipping these to the central site, which took days to weeks.

2 Dynamic Lightpaths

Currently, most lightpaths are static permanent circuits. But several projects, like DRAGON [7] and Phosphorus [8], are investigating technologies for setting up dynamic lightpaths in networks. Dynamic lightpaths are lightpaths that are setup on-demand and have a relatively short lifetime. End-users will be able to setup these dynamic lightpaths by themselves without needing network operators to do the provisioning. Instead, special reservation and provisioning systems will be deployed to do the lightpath setup.

Dynamic lightpaths have some similarities with calls in telephone networks. Both are circuits that are temporarily setup between two points in the network. There is however a big difference. The probability of failed calls in a telephone network can be described by the Erlang-B function. However, this queuing theory does not apply to dynamic lightpaths as is explained in [6].

The Erlang-B function only applies when the connection requests satisfy the following rules:

- Arrival times between requests for connections must follow a Poisson distribution
- Each connection must use the same capacity
- Hold times for connections must follow an inverse exponential distribution

Dynamic lightpath setup attempts will probably not follow a Poisson distribution. The number of setup attempts will be too small to have sufficient statistical variation. E.g., when a reservation system is used to schedule the setup of dynamic lightpaths, the start times will probably peak at round times (like 8:00 instead of 8:07).

The second rule is also not satisfied. Lightpaths support various speeds, usually multiples of 155 Mbps up to 1 Gbps.

The fact that the Erlang queuing theory does not apply to dynamic lightpath setup makes it is even more important to do measurements in networks that support dynamic lightpaths. Various relevant data can be retrieved form the network. The most important metrics are the amount of static and dynamic

lightpaths and the resources used by each of them. The used resources that are relevant to measure are the amount of timeslots, the total capacity of a link and the devices and interfaces used. This will give a better understanding of the usage of dynamic lightpaths and how many timeslots should be reserved for dynamic lightpaths. Too many timeslots for dynamic lightpaths wastes bandwidth and makes it more difficult to find optimal paths for static lightpaths. Too few timeslots for dynamic lightpaths increases the blocking probability. Section 5 describes how the data described above is retrieved from the SURFnet6 network.

3 TL1 Toolkit

The TL1 Toolkit is an open source Perl module that makes it easy to write Perl scripts that retrieve data from optical devices. Many optical devices use the Transaction Language 1 (TL1) [1] as command line interface. Unfortunately, the syntax of TL1 is complex and the returned answers are difficult to read. Therefore, we designed and implemented a Perl module that hides the difficult syntax from the user and returns the results in easy to use Perl data structures. With the TL1 Toolkit it becomes easy to write Perl scripts that extract all kinds of data from optical network devices.

3.1 TL1 Session Example

A typical TL1 session consists of logging in to the network device, executing a command, processing the output and logging out of the device. Figure 1 shows an example of a TL1 session.

The first step is logging in to the network device. This is done with the ACT-USER command in line 1. The parameters are the name of the network device, the login name, a unique string (ctag) and a password (not echoed). TL1 parameters must be given in the correct order, separated by a colon. The command is ended with a semi-colon. After the semi-colon is typed in, the device responds with IP, followed by the ctag. The next lines (2 to 5) are the result of the ACT-USER command. It starts with a line with a less-than sign only, followed by a line with the network device name, date and time. Next is a line starting with an M, followed by the ctag and the result code, in the case COMPLD (completed), indicating that the login was successful.

After a successful login, additional TL1 commands can be executed. In this example the RTRV-OM-ETH command is used to retrieve various Ethernet performance data, like inoctets and outoctets, for an interface. The parameters of the RTRV-OM-ETH command are the network device name, the interface name (ETH-1-1-1) and the ctag. The command is again ended with a semi-colon. The performance data starts on line 11. Various Ethernet statistics for the interface are returned.

Finally, on line 20 the CANC-USER command is used to log out of the device. This command has the network device name, the login name and the ctag as parameters. Line 23 indicates that the logout was successful.

```
--------------------------------------------------------------
 1 < ACT-USER:"Asd001A_OME4T":USER:42::::;IP 42
 2 <
 3     "Asd001A_OME4T" 08-09-23 19:34:17
 4 M   42 COMPLD
 5 ;
 6 < RTRV-OM-ETH:"Asd001A_OME4T":ETH-1-1-1:42::;IP 42
 7 <
 8
 9 "Asd001A_OME4T" 08-09-26 12:39:25
10 M   42 COMPLD
11 "ETH-1-1-1::INFRAMES=610791,INFRAMESERR=0,
12 INOCTETS=925839022,INDFR=0,INFRAMESDISCDS=0,
13 INPAUSEFR=0,INCFR=0,FRTOOSHORTS=0,FCSERR=0,
14 FRTOOLONGS=0,FRAG=0,JAB=0,SYMBOLERR=0,
15 OUTFRAMES=1198121,OUTFRAMESERR=0,
16 OUTOCTETS=930912996,OUTFRAMESDISCDS=0,
17 OUTPAUSEFR=0,OUTDFR=0,AUTONEGCYCLES=2,
18 INTERNALMACRXERR=0,INTERNALMACTXERR=0"
19 ;
20 < CANC-USER:"Asd001A_OME4T":USER:42;IP 42
21 <
22     "Asd001A_OME4T" 08-09-23 19:35:58
23 M   42 COMPLD
24 ;
--------------------------------------------------------------
```

Fig. 1. This figures shows a typical TL1 session. The session starts with connecting and logging in with ACT-USER. The RTRV-OM-ETH command is used to retrieve various Ethernet statistics. Finally, the CANC-USER command is used to log out of the system.

The TL1 session in figure 1 shows a number of things. Logging in and logging out requires several parameters. It is hard to remember which parameters are required and in which order. The TL1 Toolkit hides these details from the user. How this is done is described in the section 3.2.

3.2 TL1 Toolkit Script Example

Figure 2 shows how the TL1 session of figure 1 can be done using a TL1 Toolkit [2] Perl script.

```
--------------------------------------------------------------
  1   #!/usr/bin/perl -w
  2
  3   use tl1;
  4
  5   my $tl1 = tl1->new(
  6       hostname => 'node',
  7       username => 'user',
  8       password => 'passwd',
  9       peerport => 23,
 10       verbose => 0,
 11   );
 12
 13   if ($tl1->open() == 0) {
 14       print "Could not connect to $tl1->{hostname}\n";
 15       exit 1;
 16   }
 17
 18   $tl1->{slot} = 1;
 19   $tl1->{port} = 1;
 20   print $tl1->retr_inoctets(), "\n",
 21
 22   $tl1->close();
--------------------------------------------------------------
```

Fig. 2. This is an example of a Perl script that uses the TL1 Toolkit to connect, login and retrieve Ethernet inoctets data from an optical device.

First, in line 5 to 11 an instance of the TL1 Toolkit module is created. It expects a couple of parameters, such as a hostname, the login name and a password. In line 13 the *open* method is called. This takes care of connecting and logging in to the device. Line 18 and 19 set the slot and port numbers of the interface. Line 20 calls the *retr_inoctets* method which returns the number of inoctets. Finally, in line 22 the *close* method is called. This takes care of logging out of the device and disconnecting.

This example shows that it is easy to retrieve performance data from an optical device with a relatively simple Perl script. The TL1 Toolkit supports a variety of functions for frequently used TL1 commands; in addition there is also a generic *cmd* function that can be used to send any TL1 command to a device and parse the output with Perl.

The TL1 Toolkit is open source and released with an Apache2 license. It can be downloaded from `http://nrg.sara.nl/TL1-Toolkit` [2]. It is device agnostic and has been used on various devices of various vendors. Among them are Adva, Ciena, Cisco and Nortel.

4   Metrics in Optical Networks

The (performance) metrics in optical networks are different from the metrics used in the traditional routed Internet. Traffic in optical networks is transported over circuits instead of via packets. Parameters like RTT and jitter are not very interesting because of the deterministic nature of circuits. But on the other hand new parameters become important, like the timeslot usage on links. Timeslot usage is an important metric used in capacity planning for optical networks. Section 5 describes how this is done in the Dutch NREN (SURFnet6).

Other possibly interesting metrics are the parameters involving the laser equipment, like voltages, power transmitted and received, optical loss, etc. This may give early warnings about degrading of the equipment or changes in optical impairments of the fibers.

5   Lightpath Data retrieval in SURFnet6

The TL1 Toolkit is used to retrieve data from the SURFnet6 network. SURFnet6 is the Dutch NREN and is a hybrid network. It has about 200 lightpaths. Various data is retrieved from the SURFnet6 network. This is done by running TL1 Toolkit Perl scripts periodically. The retrieved data is stored in a database. A website was built to show the information from the database in various ways.

Section 5.1 describes how this data is used for static lightpath planning in SURFnet6. Section 5.2 describes how the data is used for capacity management and what can be done when dynamic lightpaths will be introduced in SURFnet6.

5.1   Lightpath Planning

The SURFnet6 lightpath part of the network consists of around 100 devices. It soon turned out that with a network of this size manual planning of lightpaths was impossible. Therefore a planning tool [4] was written for SURFnet6. This uses the TL1 Toolkit in various ways. In order to find a path from A to B, the topology of SURFnet6 must be made available to the planning tool. The Network Description Framework [5] of the University of Amsterdam is used to describe the topology. The NDL topology file is generated automatically by running a

TL1 Toolkit script that discovers the topology information from the network using the SDH section trace information. The current SURFnet topology file consists of more than 14,000 lines.

Another script retrieves the used resource data of all the lightpaths from the network. This data is stored in the database. For each lightpaths the devices, interfaces and timeslots are stored.

The planning tool uses the NDL topology file to generate a mathematical graph of the network. In this graph a constraint based shortest path algorithm (Dijkstra) is used to find paths between two points in the network. The constraints are the timeslots already in use in the network and various links weights that represent network usage policies. Typically, the paths found are first reserved in the database so that no other lightpaths can use those resources. At a later stage, when all the formalities are completed, the lightpath is provisioned in the network.

Figure 3 shows the timeslot usage for one particular network device. For all its interfaces the timeslot usage (free, reserved and used) is shown.
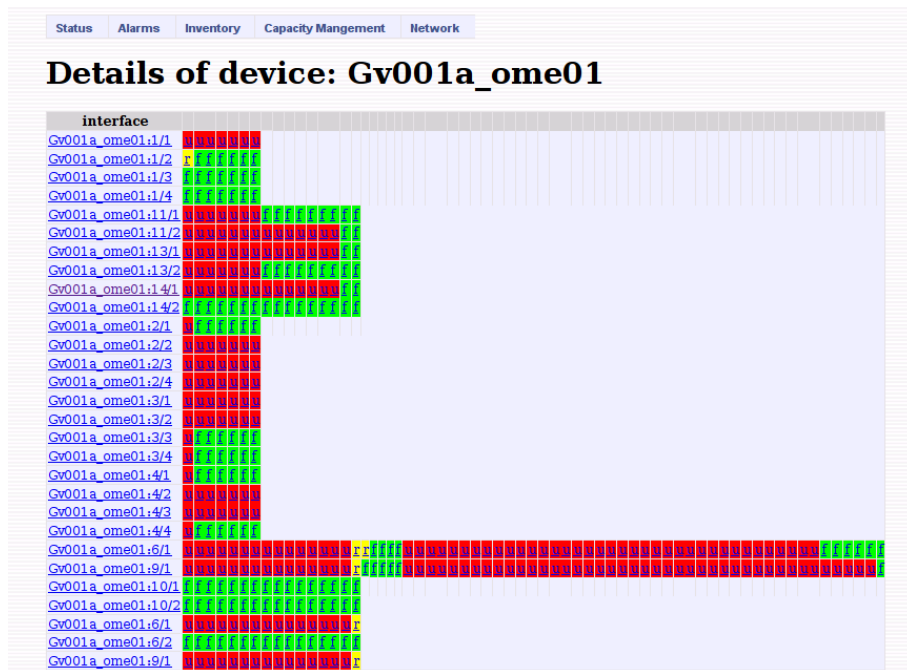

Fig. 3. This figure shows how timeslot usage is represented. The first column is the network device interface. This is followed by a row of timeslots. The timeslots are given in VC4s (7 for 1GE, 16 for OC48 and 64 for OC192). Each timeslot can have three possible values, u for *in use*, f for *free* and r for *reserved*.

5.2 Capacity Management

Section 5.1 described that the resources used by lightpaths are stored in a database. At the time of writing SURFnet6 did not support dynamic lightpaths yet. This dynamic lightpath service is planned for December 2008. So the database contains information about reserved and provisioned permanent lightpaths only. However, when dynamic lightpaths are introduced, the same software can be used to retrieve information about dynamic lightpaths. Each lightpath has an identifier and dynamic lightpath will get an identifier that distinguishes them from static lightpaths.

This information can be used in several ways. We are currently working on algorithms to optimize routing of static lightpaths in an optical network. From time to time it may be beneficial to re-route some of the static lightpaths in order to make more efficient use of the network.

Another research topic will be network simulations. The SURFnet6 network is constantly expanded with new interfaces, links and devices. In order to make good decisions about where increasing the capacity has the most benefit, it might be useful to first run some simulations of the expanded network first.

Figure 4 shows a representation of timeslot usage in the complete network. This web page shows all devices in the network. For each device the card type of all its slots is shown. For SDH cards the timeslots used and the total amount of timeslots of that interface are shown.

| Status | Alarms | Inventory | Capacity Mangement | Network |

Available Timeslots

device	slot 1	slot 2	slot 3	slot 4	slot 5	slot 6	slot 9	slot 10	slot 11	slot 12	slot 13	slot 14
Ah001a_ome01	4xGE	4xGE	4xGE		186/192	30/192	30/192	183/192				L2SS
Alr001a_ome01	4xGE					84/192	108/192	168/192				L2SS
Amf001a_ome01	4xGE					108/192	126/192				L2SS	
Amr001a_ome01	48/48	4xGE	4xGE		69/192		114/192	192/192				L2SS
Ap001a_ome01	4xGE	4xGE			183/192	30/192	30/192	186/192			L2SS	L2SS
Asd001a_ome01	171/192				21/192	30/192	66/192	48/192	51/192	24/192	60/192	147/192
Asd001a_ome02	192/192		54/192		126/192	72/192		57/192		120/192	12/192	81/192
Asd001a_ome03	4xGE	48/192	4xGE	4xGE	24/192	54/192	48/192	21/192	81/192	96/192	138/192	75/192
Asd001a_ome04	4xGE	4xGE	4xGE	147/192	69/192	105/192	90/192	30/192	12/192	57/192	69/192	108/192
Asd001a_ome05	4xGE	4xGE	4xGE	4xGE	54/192	105/192	39/192	66/192	120/192	102/192	171/192	
Asd001a_ome06	4xGE	4xGE	4xGE	4xGE	81/192	57/192	114/192	126/192	60/192	180/192		
Asd001a_ome07	4xGE	4xGE	4xGE	4xGE	120/192	51/192	108/192	72/192	84/192	51/192	18/192	
Asd001a_ome09	4xGE	4xGE	4xGE	4xGE	84/192	108/192	108/192	6/48				
Asd001a_ome10	4xGE	4xGE	4xGE	4xGE	48/192	108/192	132/192				4xGE	4xGE
Asd002a_ome01	171/192	162/192			66/192	15/192	12/192	48/192	24/192		57/192	
Asd002a_ome02	192/192				18/192	105/192		57/192	54/192	60/192	48/192	108/192
Asd002a_ome03	4xGE	4xGE			6/192	18/192	39/192	12/192	60/192	117/192	138/192	132/192
Asd002a_ome04	4xGE	4xGE			69/192	126/192	27/192	15/192	48/192	144/192		
Asd002a_ome05	4xGE	4xGE	4xGE		87/192	51/192	99/192	66/192	108/192	171/192		
Asd002a_ome06	4xGE	4xGE	4xGE		84/192	33/192	156/192	105/192	57/192	87/192		
Asd002a_ome07	4xGE	4xGE	4xGE	4xGE	81/192	171/192	108/192	18/192	126/192	162/192		
Asd003a_ome01	4xGE	4xGE				108/192	108/192					

Fig. 4. This figure shows a representation of timeslot usage. In the overview the slots of all devices are shown. For each slot, the card type is shown and for SDH cards the number of used and total available timeslots is shown.

Finally, when dynamic lightpaths are introduced we can investigate the optimal way to divide timeslots between static and dynamic lightpaths. And when the dynamic reservation system provides lightpath request statistics like successful and failed requests, we can correlate that information with the resource usage in the network.

6   Conclusions and Future Work

The TL1 Toolkit has proven to be a very powerful tool for writing Perl scripts with minimal effort to extract all kind of data from optical TL1 equipment.

Our own work with the TL1 Toolkit has been focused on monitoring operational status of lightpath [3]. But because there are many interesting metrics to investigate in optical networks, we will start tracking those too.

The introduction of dynamic lightpaths will introduce a whole new line of research. There is little operational experience with dynamic lightpaths and assigning timeslots is the most efficient way will be an interesting topic. The TL1 Toolkit is very well suited to retrieve and store data about dynamic lightpaths. This will produce a whole new set of network data that has not been available yet.

References

1. TR-NWT-000835, Telcordia Standard, Issue Number 03, (1993)
2. TL1 Toolkit download section, `http://nrg.sara.nl/TL1-Toolkit`
3. van der Pol, R., Toonk, A.: Lightpath Planning and Monitoring. eChallenges 2007 (2007)
4. van der Pol, R., Toonk, A.: Lightpath Planning and Monitoring in SURFnet6 and NetherLight. TNC2007 (2007)
5. van der Ham, J., Grosso, P., van der Pol, R., Toonk A., de Laat, C.: Using the Network Description Language in Optical Networks. 10th IFIP/IEEE International Symposium on Integrated Network Management, pp. 199-205 (2007)
6. GLIF Technical Report: Measuring the use of Lightpath service (2007), `http://www.glif.is/working-groups/tech/lightpath-measurement-0.9.pdf`
7. DRAGON website, `http://dragon.maxgigapop.net/`
8. Phosphorus website, `http://www.ist-phosphorus.eu/`