D1.1 Next Generation Ethernet Tests

Ronald van der Pol and Freek Dijkstra

Version 1.1
December 2011

1 Introduction

This document describes the tests that were done with some of the new SURFnet7 Next Generation Ethernet (NGE) switches, the Ciena 3960. The Classes of Service (CoS) features of these switches were investigated in order to determine if they can be used to implement a *lightpath* like guaranteed bandwidth service. Section 2 describes the CoS features of the Ciena 3960. Section 3 describes how the Anritsu MD1230B Ethernet/IP Network Data Analyzer works. It was used as traffic generator and analyzer. Finally, section 4 describes how a *lightpath* service can be configured on the Ciena 3960.

2 CoS on the Ciena 3960

CoS on the 3960 is done both at the ingress and at the egress port. At the ingress port profiling and mapping to classes is done and at the egress port scheduling, shaping and congestion management is done. At the ingress port each frame is given an internal Resolved Class of Service (R-CoS) value and a Resolved Color (R-Color) value. The R-CoS value is in the range 0-7, where 7 has the highest priority. The R-Color is either *green* or *yellow*. Frames at the ingress port are forwarded and put in one of eight queues of an egress port.

Figure 1 gives an overview of the CoS functions of the Ciena 3960. The basic functionality is described in chapter 10 of the Ciena SAOS Software Configuration Guide release 6.9.0 [1], but unfortunately that documentation is incomplete. This section briefly describes each function and clarifies its properties based on our experiments. We used version saos-06-09-00-0242 of the Ciena 3960 firmware.

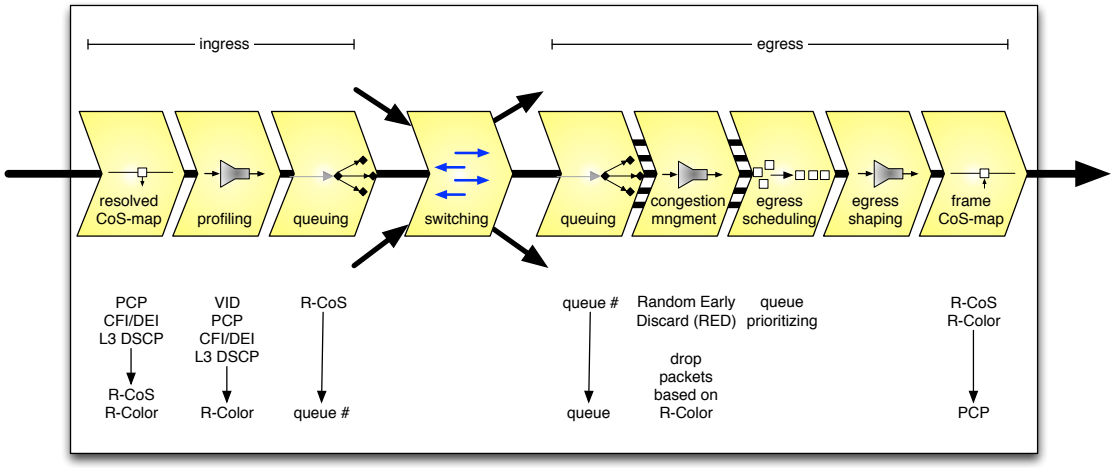

Fig. 1. Overview of CoS functions in the Ciena 3960.

2.1 VLAN Tag Control Information

The IEEE 802.1Q standard defines the VLAN Tag Control Information (TCI). It is a two-byte field containing the Priority Code Point (PCP), the Canonical Format Indicator (CFI) and the VLAN ID (VID). The CFI bit is usually set to zero. It is set to one for media access types such as FDDI. The PCP is a three bit field containing the priority value (0-7). See also figure 2. When

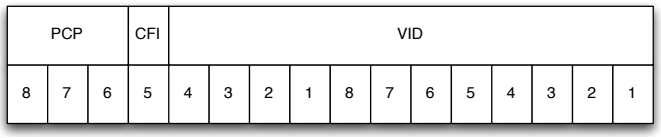

Fig. 2. VLAN Tag Control Information.

using IEEE 802.1AD (QinQ) the CFI bit in the S-TAG is called DEI (Drop Eligible Indicator). Frames with the DEI set to 1 are more likely to be dropped than frames with a DEI of 0.

2.2 Resolved CoS and Color Mapping

At the ingress port each frame is given an internal Resolved Class of Service (R-CoS) value and a Resolved Color (R-Color) value. The R-CoS value is determined by the Resolved CoS Map, while the R-Color value is determined by both the Resolved CoS Map, as well as the optional traffic profiling as discussed in section 2.3. The Resolved CoS Map can be configured based on various attributes of an incoming frame, and can be set for each port. One of these three options can be used:

l3-dscp-cos: Use the L3 DSCP (DiffServ) CoS value
dot1d-tag1-cos: Use the L2 PCP/DEI/CFI values
fixed-cos: Use fixed R-CoS and R-Color values

This is the way to configure it on the 3960:

```
> port set port <PortNameList> [resolved-cos-policy
  <dot1d-tag1-cos|fixed-cos|l3-dscp-cos>]
```

Frames are now internally marked with an R-CoS and R-Color value. The Resolved CoS Map is used to define this mapping. The default Resolved CoS Map can be shown with:

```
> traffic-services cos-mapping resolved-cos-map show cos-map DefaultFcosRcos
```

The default Resolved CoS Map is shown in table 1. Customized Resolved CoS Maps can also be configured and applied to an ingress port.

2.3 Ingress Traffic Profiling

The Resolved CoS Map labels traffic with an R-CoS and R-Color based on the CoS value of a frame. Traffic Profiling sets the R-Color based on the bandwidth rate. Various traffic profiles can be configured, including per-VLAN traffic profiles.

By default, traffic profiling status is disabled. It can be enabled with the commands:

```
> traffic-profiling enable
> traffic-profiling enable <PortNameList>
```

The first command enables traffic profiling globally and the second enables it on the specified port(s). The status of profiling on each port can be viewed with:

Table 1. Default Resolved CoS Map (*DefaultFcosRcos*)

PCP	DEI/CFI	DSCP	R-CoS	R-Color
0	0	0-7	0	green
0	1	0-7	0	green
1	0	8-15	1	green
1	1	8-15	1	green
2	0	16-23	2	green
2	1	16-23	2	green
3	0	24-31	3	green
3	1	24-31	3	green
4	0	32-39	4	green
4	1	32-39	4	green
5	0	40-47	5	green
5	1	40-47	5	green
6	0	48-55	6	green
6	1	48-55	6	green
7	0	56-63	7	green
7	1	56-63	7	green

```
> traffic-profiling show

+------------ TRAFFIC PROFILING GLOBAL TABLE ---------------+
| Profiling Status     | Enabled                             |
+-----------------------------------------------------------+

+---------------- PORT TRAFFIC PROFILE TABLE ----------------+
|  Port     |       Status        | Mode                    |
|           | Admin    | Oper     |                         |
+-----------+----------+----------+-------------------------+
| 1         | Enabled  | Enabled  | standard-vlan           |
| 2         | Enabled  | Enabled  | standard-dot1dpri       |
| 3         | Enabled  | Enabled  | standard-dot1dpri       |
| 4         | Enabled  | Enabled  | standard-dot1dpri       |
| 5         | Enabled  | Enabled  | standard-vlan           |
| 6         | Enabled  | Enabled  | standard-dot1dpri       |
| 7         | Enabled  | Enabled  | standard-dot1dpri       |
| 8         | Enabled  | Enabled  | standard-vlan           |
| 9         | Disabled | Disabled | standard-dot1dpri       |
| 10        | Disabled | Disabled | standard-dot1dpri       |
| 11        | Disabled | Disabled | standard-dot1dpri       |
| 12        | Disabled | Disabled | standard-dot1dpri       |
+-----------+----------+----------+-------------------------+
```

The mode can be one of these:

none: No individual profiles. Use the Non-Conform Standard Profile for all traffic.
standard-dot1dpri: Finds a matching traffic profile using 802.1D priority. This mode is the default.
standard-ip-prec: Finds a matching profiling using the upper 3 bits of the TOS byte that make up the IP precedence.
standard-dscp: Finds a matching profiling using the DSCP value.
standard-vlan: Finds a matching profiling based upon the VLAN ID.
standard-vlan-dot1dpri: Finds a matching profiling based upon the VLAN ID and outer.1D priority value in the frame.
standard-vlan-ipp: Finds a matching profiling based upon the VLAN ID and ip-precedence value in the frame.

standard-vlan-dscp: Finds a matching profiling based upon the VLAN ID and the DSCP value in the frame.

Traffic profiling can be used to compare ingress traffic to a configured Committed Information Rate (CIR) and a Peak Information Rate (PIR). With traffic profiling each ingress frame is assigned an R-Color:

- Traffic up to CIR will be marked *green* and allowed through
- Traffic above CIR and less than PIR will be marked *yellow* and allowed through
- Traffic above PIR will be marked *red* and dropped

The final R-Color is determined by both the Traffic Profile and the Resolved CoS Map. If the Resolved CoS map has marked it as *yellow*, it will remain *yellow*, even if the data rate is below the CIR. Traffic profiling can be set on a port with the command:

```
> traffic-profiling set port <PortName>
  {[arp-standard-profile <bypass|<Traffic ProfilingStandardName>],
  [meter-pool <TrafficProfilingMeterPoolName>],
  [classifier-mode <narrow|wide>],
  [mode <none|standard-dot1dpri|standard-ip-prec|standard-dscp|
  standard-vlan|standard-vlan-dot1dpri|standard-vlan-ipp|
  standard-vlan-dscp|hierarchical-port|hierarchical-vlan],
  [nonconform-standard-profile <drop|<TrafficProfilingStandardName>>]}
```

For example:

```
> traffic-profiling set port 1 arp-standard-profile bypass
  mode standard-vlan nonconform-standard-profile drop
```

This configures port 1 to use VLAN based traffic profiles, to mark all ARP traffic as *green*, and to drop all traffic which does not match one of the configured traffic profiles. A traffic profile can be created with the command:

```
> traffic-profiling standard-profile create {port <PortNameList>}
  [dot1dpri <NUMBER LIST: 0-7>] [ip-prec <NUMBER LIST: 0-7>]
  [dscp <NUMBER LIST: 0-63>] [dscp-remark-policy <leave | fixed>]
  [fixed-dscp <0-63>] {[cir <NUMBER>], [pir <NUMBER>], [cbs <NUMBER: 0-2048>],
  [ebs <NUMBER: 0-2048>]} [name <TrafficProfilingStandardName[15]>]
  [vlan <VLAN>] [untagged] [statistics <on|off>] [drop <true|false>]
```

For example, to set a CIR of 500 Mbps and PIR of 800 Mbps for VLAN 42:

```
> traffic-profiling standard-profile create port 1
  cir 500000 pir 800000 vlan 42 statistics on name vlan42
```

To display all traffic profiles for all ports:

```
> traffic-profiling standard-profile show

+---------------------------------------- STANDARD PROFILE TABLE ----------------------------------------+
|  Port |      Profile       |    |Parent|        BW (Kbps)    |   Max Burst KB  |       CLASSIFIERS      |
|       | ID | Name          |Role|#Child|    CIR   |    PIR   |   CBS  |   EBS  |                        |
+-------+----+---------------+----+------+----------+----------+--------+--------+------------------------+
|5      |1   |vlan42-400     | N  |    0 |   400000 |   400000 |  20016 |      0 | VLN| 42                |
+-------+----+---------------+----+------+----------+----------+--------+--------+------------------------+
|5      |2   |vlan43-500     | N  |    0 |   400384 |   800768 |     64 | 200000 | VLN| 43                |
+-------+----+---------------+----+------+----------+----------+--------+--------+------------------------+
|6      |1   |rvdp-std-prof  | N  |    0 |   600000 |  1000000 |   1024 |   1024 |                        |
+-------+----+---------------+----+------+----------+----------+--------+--------+------------------------+
|7      |1   |rvdp-std-prof  | N  |    0 |        0 |  1000000 |   1024 |   1024 |                        |
+-------+----+---------------+----+------+----------+----------+--------+--------+------------------------+
|8      |1   |rvdp-test      | N  |    0 |   100032 |  1000000 |   1024 |   1024 |                        |
+-------+----+---------------+----+------+----------+----------+--------+--------+------------------------+
```

Table 2. Default Internal R-CoS Mapping

R-CoS	Egress Queue
0	0
1	0
2	1
3	2
4	3
5	4
6	5
7	6
CPU frames	7

2.4 Ingress R-CoS to egress queue mapping

Each egress port supports 8 queues. The internal R-CoS value is used to map each frame to one of the 8 port queues as shown in table 2. Up to 7 custom Queue Map Profiles can be configured. The following example defines a queue map where all traffic ends up in queue 4:

```
> traffic-services queuing queue-map create rcos-map all-in-4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 0 queue 4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 1 queue 4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 2 queue 4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 3 queue 4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 4 queue 4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 5 queue 4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 6 queue 4
> traffic-services queuing queue-map set rcos-map all-in-4 rcos 7 queue 4
```

This queue map can be applied to a port with:

```
> port set port 9 ingress-to-egress-qmap all-in-4
```

The queue map is applied at the ingress port, while the queue is at the egress port. Thus if traffic flows from port 2 to port 3, the queue map of port 2 is consulted to determine the queue at port 3.

2.5 Congestion Management

A congestion avoidance profile is applied to each queue based on (Weighed) Random Early Discard (RED/WRED) of frames. The profile determines the probability that a frame is dropped when entering the queue. The goal of RED is to drop packets even before the queue buffer fills up completely. TCP detects this loss of packets and slows down. This reduces the latency caused by the queue.

The drop probability of a frame is determined by

- The amount of data in the queue
- The R-Color of the frame (*yellow* or *green*)
- The protocol number (TCP or non-TCP)

The Ciena 3960 defines 10 attributes for each congestion avoidance profile:

yellow admit rate: Undocumented, likely the maximum buffer fill for *yellow* frames to enter the queue.

and for each of the categories TCP-green, TCP-yellow and non-TCP:

lower threshold: The percentage of the queue capacity that is reached before frames are eligible to be dropped.

upper threshold: The percentage of the queue capacity that is reached before all frames are eligible to be dropped (not properly documented, likely meaning that all frames will be dropped after this threshold is reached).

drop probability: The percentage of frames that are dropped once the threshold is exceeded.

Unfortunately, the documentation for the Ciena 3960 is incomplete, so it the exact nature of the lower and upper threshold, and the yellow admit limit are unclear. Measurements clearly showed that a change in traffic profile resulted in a different drop rate for *yellow* and *green* traffic, but were inconclusive in the exact behaviour of the drop rate. We suspect the drop rate to follow one of the two functions in figure 3.

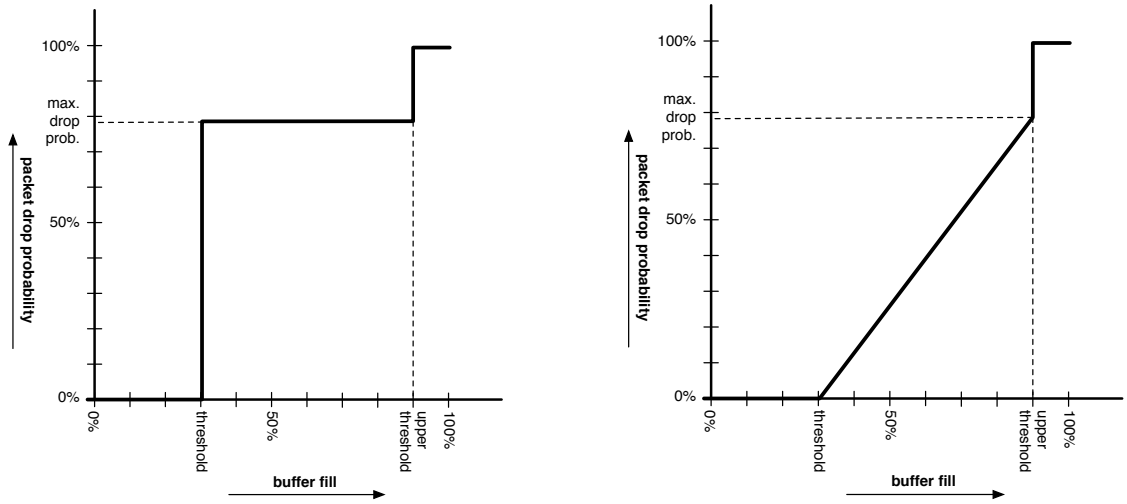

Fig. 3. Frame drop probability as function of how full the queue is. The exact function is unclear due to inconclusive measurements.

The per queue configured congestion avoidance profiles can be shown with:

```
> traffic-services queuing egress-port-queue-group show queue 2 port 5

+---------------------------------- QUEUE DATA ----------------------------------+
|                                                                                |
+--------------------------------+-------------------------------------------+
| Queue Id                       | 2                                         |
| Queue Group Name [Port]        | 5                                         |
| Congestion Avoidance Profile   | Default-S-WRED                            |
+--------------------------------------------------------------------------------+
|     Scheduler      |  Size   |    CIR     |   CBS    |    EIR     |   EBS    |
| Pri Idx | Weight   | (Pckts) |   (Kbps)   | (Kbytes) |  (Kbps)    | (Kbytes) |
+---------+----------+---------+------------+----------+------------+----------+
| 30000   | 40       | 100     | 0          | 0        | 1000000    | 256      |
+---------+----------+---------+------------+----------+------------+----------+
```

The actual congestion avoidance profiles can be shown with:

```
> traffic-services queuing congestion-avoidance-profile show
```

We tested the following three congestion avoidance profiles.

In congestion avoidance profile-1 (table 3) all traffic is treated equally. The result is that all traffic is passed through based on the bandwidth of the incoming streams, up till the capacity of the egress port. No distinction is made between *green* or *yellow* traffic.

Table 3. Congestion Avoidance Profile Data for `profile-1`

Name			profile-1					
Id			2					
Type			WRED-Simple					
Yellow Admit Limit			100%					
Tcp-Green			Tcp-Yellow			Non-Tcp		
Lower Threshold	Upper Threshold	Drop Probability	Lower Threshold	Upper Threshold	Drop Probability	Lower Threshold	Upper Threshold	Drop Probability
100%	100%	0pct	100%	100%	0pct	100%	100%	0pct

Table 4. Congestion Avoidance Profile Data for `profile-2`

Name			profile-2					
Id			3					
Type			WRED-Simple					
Yellow Admit Limit			100%					
Tcp-Green			Tcp-Yellow			Non-Tcp		
Lower Threshold	Upper Threshold	Drop Probability	Lower Threshold	Upper Threshold	Drop Probability	Lower Threshold	Upper Threshold	Drop Probability
100%	100%	0pct	50%	75%	100pct	100%	100%	0pct

In profile-2 (table 4) TCP-yellow frames are dropped if more than half the queue is filled, to prioritise TCP-Green traffic. The result is that for TCP streams, *yellow* frames are dropped well before *green* frames are dropped. The queue is never completely filled, and thus no *green* frames are dropped, as long as UDP traffic does not fill the queue and the total CIR does not exceed the available egress capacity. The disadvantage is that this only works for TCP traffic. The profile does not distinguish between *green* and *yellow* non-TCP traffic. E.g., *green* TCP or UDP frames may get dropped in favour of *yellow* UDP frames.

Table 5. Congestion Avoidance Profile Data for `profile-3`

Name			profile-3					
Id			4					
Type			WRED-Simple					
Yellow Admit Limit			30%					
Tcp-Green			Tcp-Yellow			Non-Tcp		
Lower Threshold	Upper Threshold	Drop Probability	Lower Threshold	Upper Threshold	Drop Probability	Lower Threshold	Upper Threshold	Drop Probability
100%	100%	0pct	100%	100%	0pct	100%	100%	0pct

In profile-3 (table 5) the yellow admit limit is set. The result is that *yellow* frames are dropped well before *green* frames are dropped. The queue is never completely filled, and no *green* frames are dropped, even with high bandwidth UDP traffic.

Since ARP traffic is always marked as *green,* it is theoretically possible to use that for a denial of service attack which can impact other guaranteed traffic. Since ARP traffic is not forwarded by routers, it can only be exploited by hosts directly connected to the switch port.

2.6 Egress Scheduling

Scheduling determines the order in which the physical queues are processed. The queue scheduler supports various options:

– Strict priority (queue 7 the has highest, 0 the lowest priority)

– Round-Robin
– Weighed Round Robin (each queue has a weight)
– Weighed Deficit Round Robin (also uses pool size)

The scheduling can be set with the command:

```
> traffic-services queuing egress-port-queue-group set port
  <PortQueueGroup> {[scheduler-algorithm
  <strict|round-robin|weighted-deficit-round-robin|weighted-round-robin>],
  [wdrr-scheduler-granularity <NUMBER: 50- 1600>]}
```

The per port configured congestion scheduling algorithm can be shown with:

```
> traffic-services queuing egress-port-queue-group show port 8
```

```
+------------------------------ QUEUE GROUP DATA -------------------------------+
|                                                                               |
+--------------------------------+----------------------------------------------+
| Name [Port]                    | 8                                            |
| Id                             | 8                                            |
| Queue Count                    | 8                                            |
| Scheduling Algorithm           | round-robin                                  |
| Shaper Bandwidth (Kbps)        | 1000000                                      |
| Shaper Burst Size (Kb)         | 10240                                        |
+-------------------------------------------------------------------------------+
|   |     Scheduler    |  Size   |    CIR    |   CBS    |    EIR    |   EBS    |
| Q | Pri Idx | Weight | (Pckts) |  (Kbps)   | (Kbytes) |  (Kbps)   | (Kbytes) |
+---+---------+--------+---------+-----------+----------+-----------+----------+
| 0 | 10000   | 20     | 100     | 0         | 0        | 1000000   | 256      |
| 1 | 20000   | 30     | 100     | 0         | 0        | 1000000   | 256      |
| 2 | 30000   | 40     | 100     | 0         | 0        | 1000000   | 256      |
| 3 | 40000   | 50     | 100     | 0         | 0        | 1000000   | 256      |
| 4 | 50000   | 60     | 100     | 600000    | 256      | 1000000   | 256      |
| 5 | 60000   | 70     | 100     | 0         | 0        | 1000000   | 256      |
| 6 | 70000   | 80     | 100     | 0         | 0        | 1000000   | 256      |
| 7 | 80000   | 0      | 100     | 1024      | 256      | 1000000   | 256      |
+---+---------+--------+---------+-----------+----------+-----------+----------+
```

2.7 Egress Shaping

Both shaping and burst sizes can be configured on an egress port queue. This is done with the following command:

```
traffic-services queuing egress-port-queue-group set port <PortQueueGroup>
shaper-rate <NUMBER: 0-40000000> burst-size <NUMBER: 4-10240>
```

For each egress queue the following parameters can be configured:

CIR: (in Kbps) Committed Information Rate; guaranteed traffic
CBS: (in Kbytes) CIR Bucket Size
EIR: (in Kbps) Excess Information rate; non-guaranteed traffic above CIR
EBS: (in Kbytes) EIR Bucket Size

This is configured with the following command:

```
traffic-services queuing egress-port-queue-group set queue <NUMBER: 0...7>
port <PortQueueGroup> cir <NUMBER: 0...40000000> cbs <NUMBER: 0...262144>
eir <NUMBER: 0...40000000> ebs <NUMBER: 0...262144>
```

Settings can be shown with this command as shown in section 2.6:

```
> traffic-services queuing egress-port-queue-group show port <PortQueueGroup>
```

2.8 Frame CoS Maps

If frame CoS mapping is enabled, the R-CoS and R-Color associated with a frame are mapped to specific S-VLAN L2 PCP values in the frame when it is transmitted on the wire. By default, ports are set to ignore frame CoS mapping, and retain the PCP values from the original frame. To enable frame CoS mapping, use:

```
> port set port <PortNameList> frame-cos-map <Frame CoS Map Name>
  egress-frame-cos-policy rcos-to-l2-outer-pcp-map
```

To disable from CoS mapping, use:

```
> port set port <PortNameList> egress-frame-cos-policy ignore
```

The default Frame CoS Map is shown in table 6.

Table 6. Default Frame CoS Map

R-CoS	R-Color	PCP	DEI/CFI
0	green	0	0
0	yellow	0	0
1	green	1	0
1	yellow	1	0
2	green	2	0
2	yellow	2	0
3	green	3	0
3	yellow	3	0
4	green	4	0
4	yellow	4	0
5	green	5	0
5	yellow	5	0
6	green	6	0
6	yellow	6	0
7	green	7	0
7	yellow	7	0

The Ciena 3960 does not only allow remarking (setting) of the frame PCP value with the Frame CoS Map, but also with the Resolved CoS mapping. To enable that feature, use:

```
> port set port <PortNameList> resolved-cos-remark-l2 true
```

To set the layer 3 (DSCP) field of a frame, configure the traffic profiling to do so:

```
> traffic-profiling standard-profile set port <PortName>
  profile <ProfileNumber> dscp-remark-policy <leave | fixed>
```

2.9 Measurements

The above description of CoS features in the Ciena 3960 was created by a study of chapter 10 of the Ciena SAOS Software Configuration Guide release 6.9.0 [1]. We verified the behaviour of these features, and did additional measurements to verify a particular feature if the documentation fell short.

We examined four methods to display the throughput statistics and packet counters for our measurements. The most basic counter is:

```
> port show port 8 statistics

+---------- PORT 8 STATISTICS ----------+
| Statistic          | Value            |
+--------------------+------------------+
```

```
| RxBytes            | 15072000         |
| RxPkts             | 15072            |
| RxCrcErrorPkts     | 0                |
| RxUcastPkts        | 15072            |
| RxMcastPkts        | 0                |
| RxBcastPkts        | 0                |
| UndersizePkts      | 0                |
| OversizePkts       | 0                |
| FragmentsPkts      | 0                |
| JabbersPkts        | 0                |
| RxPausePkts        | 0                |
| RxDropPkts         | 0                |
| RxDiscardPkts      | 0                |
| RxLOutRangePkts    | 0                |
| RxInErrorPkts      | 0                |
| 64OctsPkts         | 0                |
| 65To127OctsPkts    | 0                |
| 128To255OctsPkts   | 0                |
| 256To511OctsPkts   | 0                |
| 512To1023OctsPkts  | 15072            |
| 1024To1518OctsPkts | 0                |
| 1519To2047OctsPkts | 0                |
| 2048to4095OctsPkts | 0                |
| 4096to9216OctsPkts | 0                |
| TxBytes            | 1463680212362    |
| TxPkts             | 1463692152       |
| TxExDeferPkts      | 0                |
| TxDeferPkts        | 0                |
| TxGiantPkts        | 0                |
| TxUnderRunPkts     | 0                |
| TxCrcErrorPkts     | 0                |
| TxLCheckErrorPkts  | 0                |
| TxLOutRangePkts    | 0                |
| TxLateCollPkts     | 0                |
| TxExCollPkts       | 0                |
| TxSingleCollPkts   | 0                |
| TxCollPkts         | 0                |
| TxPausePkts        | 0                |
| TxUcastPkts        | 1463677538       |
| TxMcastPkts        | 14614            |
| TxBcastPkts        | 0                |
| Tx64OcPkts         | 0                |
| Tx65To127OcPkts    | 0                |
| Tx128To255OcPkts   | 14614            |
| Tx256To511OcPkts   | 0                |
| Tx512To1023OcPkts  | 1463677538       |
| Tx1024To1518OcPkts | 0                |
| Tx1519To2047OcPkts | 0                |
| Tx2048To4095OcPkts | 0                |
| Tx4096To9216OcPkts | 0                |
+--------------------+------------------+
```

Unfortunately, this data showed per-port statistics, which was not very relevant for our measurements.

The throughput can be shown with the command:

```
> port show throughput active count 1 delay 10
```

```
+----------------- PORT THROUGHPUT SUMMARY   10 SECOND SAMPLE -----------------+
|   Port   |         Bit Rate (Mbps)         |        Pkt Rate (Mpps)           |
|          | Tx             | Rx              | Tx             | Rx              |
+----------+----------------+-----------------+----------------+-----------------+
| 1        |          0.000 |                 |          0.000 |                 |
| 2        |        267.293 |           0.000 |          0.522 |           0.000 |
| 3        |          0.000 |         280.636 |          0.000 |           0.548 |
| 4        |          0.000 |                 |          0.000 |                 |
| 5        |          0.000 |         513.971 |          0.000 |           1.004 |
| 6        |          1.847 |                 |          0.004 |                 |
| 7        |          1.311 |                 |          0.003 |                 |
| 8        |        280.667 |           0.000 |          0.548 |           0.000 |
+----------+----------------+-----------------+----------------+-----------------+
```

While this is a useful tool to detect per-port traffic streams, we found a few bugs in the implementation:

– The first measurement is wrong. After unplugging a SFP from a port, the first throughput measurement after would display a throughput of > 100 Mbps for this (disconnected!) port. A possible explanation is that the counters are not reset when the throughput measurement is started.
– A dip of roughly 3.4% (487 Mbps down from 504 Mbps) while taking multiple 30-second samples was observed about once every 5 measurements, while the other measurements were within 0.3% of each other. A possible explanation is that the Ciena 3960 only samples the counters once per second and the slower measurement effectively only sampled 29 seconds.

Because of these flaws, we didn't use throughput measurements for our tests.

We mostly used the per-queue counters for our measurements. First we reset the counters, did the measurement and showed the counters:

```
> traffic-services queuing egress-port-queue-group clear port 8 statistics
... perform measurement ...
> traffic-services queuing egress-port-queue-group show port 8 statistics

+----------------------------- QUEUE GROUP STATISTICS -----------------------------+
|------------------------------------------------------------------------------|
|                                                                              |
| Port Queue Group: 8                                                          |
|                                                                              |
| Q ID |  Dropped Bytes  |  Dropped Pkts   | Transmitted Bytes | Transmitted Pkts  |
|------+-----------------+-----------------+-------------------+-------------------|
|    0 |               0 |               0 |                 0 |                 0 |
|    1 |      2666364000 |         2666265 |       11963445000 |          11963445 |
|    2 |      2643012000 |         2643013 |       11859976000 |          11859976 |
|    3 |               0 |               0 |                 0 |                 0 |
|    4 |        22597000 |           22479 |         103477000 |            103477 |
|    5 |               0 |               0 |                 0 |                 0 |
|    6 |               0 |               0 |                 0 |                 0 |
|    7 |               0 |               0 |                 0 |                 0 |
|------+-----------------+-----------------+-------------------+-------------------|
+------------------------------------------------------------------------------+
```

Finally, we also used the traffic profiling statistics to see how many traffic was marked *red* and immediately dropped:

```
> traffic-profiling standard-profile clear statistics
... perform measurement ...
> traffic-profiling show port 5 statistics

+---------------------- STANDARD PROFILE TABLE ----------------------+
|  Port |      Profile            | Statistics                       |
|       | ID | Name               | Type       Bytes                 |
+-------+----+--------------------+----------------------------------+
|5      |1   |vlan42-500          | Accepted   565684527000          |
|       |    |                    | Dropped    3674318000            |
+-------+----+--------------------+----------------------------------+
|5      |2   |vlan43-500          | Accepted   908082121000          |
|       |    |                    | Dropped    0                     |
+-------+----+--------------------+----------------------------------+
```

A minor annoyance is that it is not possible to reset the traffic profile counters per-port, but only globally.

A major drawback is that it seems impossible to show individual statistics for the *yellow* and *green* R-Cos values. We overcame this limitation by defining a Frame CoS map that set the L2 PCP value based on the R-Color, and measured these PCP values with an Anritsu Network Data Analyzer, as described in the next section.

To create and apply such Frame CoS map:

```
> traffic-services cos-mapping frame-cos-map create cos-map test-rcolor
> traffic-services cos-mapping frame-cos-map set cos-map test-rcolor
  r-cos 0 r-color green dot1dpri-cos 5 dot1dpri-dei 0
> traffic-services cos-mapping frame-cos-map set cos-map test-rcolor
  r-cos 1 r-color green dot1dpri-cos 5 dot1dpri-dei 0
> traffic-services cos-mapping frame-cos-map set cos-map test-rcolor
  r-cos 2 r-color green dot1dpri-cos 5 dot1dpri-dei 0
...
> traffic-services cos-mapping frame-cos-map set cos-map test-rcolor
  r-cos 7 r-color green dot1dpri-cos 5 dot1dpri-dei 0
> traffic-services cos-mapping frame-cos-map set cos-map test-rcolor
  r-cos 0 r-color yellow dot1dpri-cos 4 dot1dpri-dei 0
...
> traffic-services cos-mapping frame-cos-map set cos-map test-rcolor
  r-cos 6 r-color yellow dot1dpri-cos 4 dot1dpri-dei 0
> traffic-services cos-mapping frame-cos-map set cos-map test-rcolor
  r-cos 7 r-color yellow dot1dpri-cos 4 dot1dpri-dei 0

> port set port 8 frame-cos-map test-rcolor egress-frame-cos-policy
  rcos-to-l2-outer-pcp-map
```

3 Anritsu MD1230B Ethernet/IP Network Data Analyzer

The Anritsu analyzer can stream and capture Ethernet and IP traffic. The user can define the length and content of frames to be sent and also the timing between frames. Various counters are present at the sending and receiving side and captured frames can be displayed and the content of the frames can be decoded. The Anritsu can decode many protocols.

3.1 Transmitting Stream Settings

Figure 4 shows how streams are built. A stream consists of one or more bursts. The inter bursts gap (IBG) defines how much time there is between two bursts. A burst consists of one or more frames. The inter frame gap (IFG) defines how much time there is between frames. Streams can be sent continuously. When this setting is used, the same stream is sent over and over again. Each stream is separated from the next by the inter stream gap (ISG).

It is also possible to configure many different streams and these streams can be sent one after the other. These are separated again by the ISG. Most of our tests were done with a simple continuous stream with equal packets separated by equal distance gaps. This was done by using the *Continuous Distribution* setting. With this setting streams with 1 burst and 1 frame per burst are used and these are sent continuously, separated by the IFG. The Anritsu uses this formula to calculate the frames per second rate:

$$fps = \frac{MediaSpeed}{PreambleSize + FrameSize + GapSize} \tag{1}$$

For 1000BASE Ethernet the MediaSpeed is 125,000,000 bytes/s. The FrameSize is including the 4 byte FCS (Frame Check Sequence), but excluding the 8 byte preamble. The rate in bits per second is simply: $bps = fps \times FrameSize \times 8$. The following table shows some examples.

frame size	gap size	rate	
bytes	bytes	fps	bps
1000	659	74,985	559,880,024
1000	1492	50,000	400,000,000
1000	3992	25,000	200,000,000
1000	8992	12,500	100,000,000

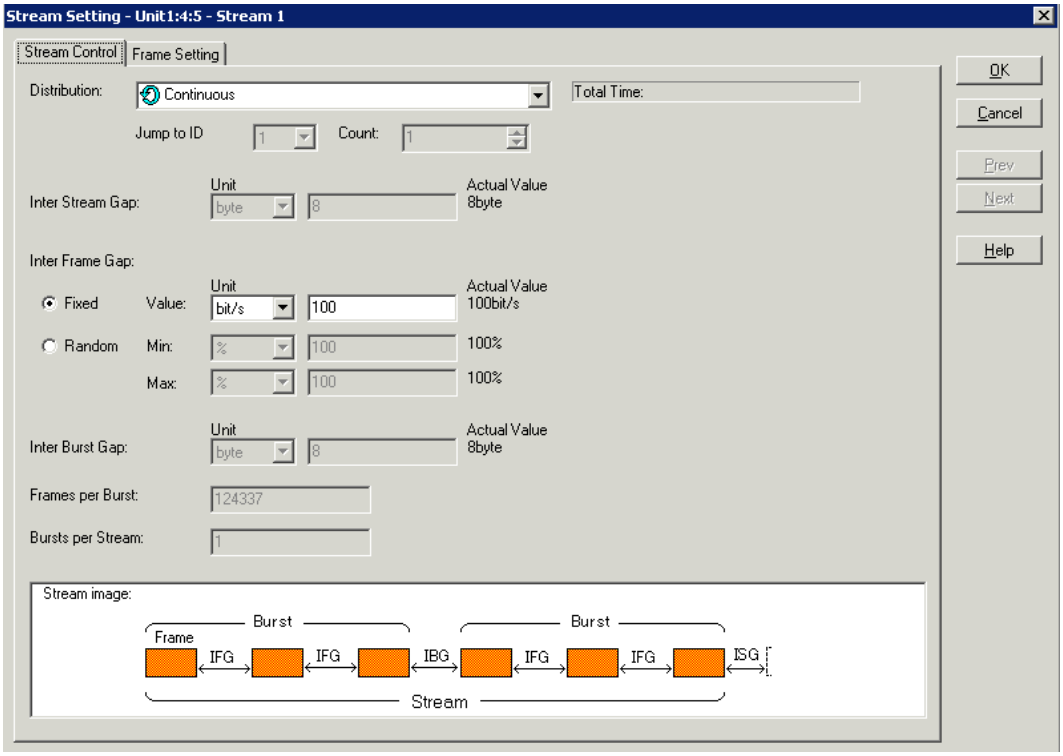

Fig. 4. Anritsu Stream Timing.

Streaming two flows (different MAC/IP addresses) on the same interface is more difficult. The easiest way to do this is to send both flows with the same frames per second rate. This can be done by alternating between the two flows, i.e. send a frame of flow one, followed by a frame of flow two, followed by a frame of flow one, etc. Both streams need to be configured with their own frame content (Ethernet and IP addresses, etc.). Each stream consists of 1 burst and 1 frame per burst, so effectively 1 frame per stream. The ISG defines the gap between the two flows.

When both flows need to be sent at different rates, it gets more complicated. The Anritsu interface does not give the user the option to send two flows and state that one flow should be sent with a rate of X fps and the other with Y fps. The user need to choose multiple frames per burst and also calculate the proper gaps between all frames.

3.2 Counters

The 1230B analyzer supports various counters on each port. These can be viewed by choosing a port in the left-hand side tree menu ([1] in figure 5) and selecting the *Counter* tab [7] in the main window. Counting on a port is enabled by clicking on the *Counter* button [3] at the top of the window. When counting, the *Counter* button [3] shows a red square. Otherwise a green arrow is shown. Counters can be cleared by stopping the counting and clicking the third icon from the top left (*Clear counters* [5]) of the main window.

There are two user defined counters, *User Defined 1* and *User Defined 2*. Up to four patterns can be defined to choose which traffic should be counted. This is done by stopping the counting and clicking on the hammer icon (*Counter setting*) [4] at the top left of the main window. At the left, *Counter 1* and/or *Counter 2* can be enabled by selecting the *On* selection box. The four match patterns can now be defined to select the traffic that should be counted.

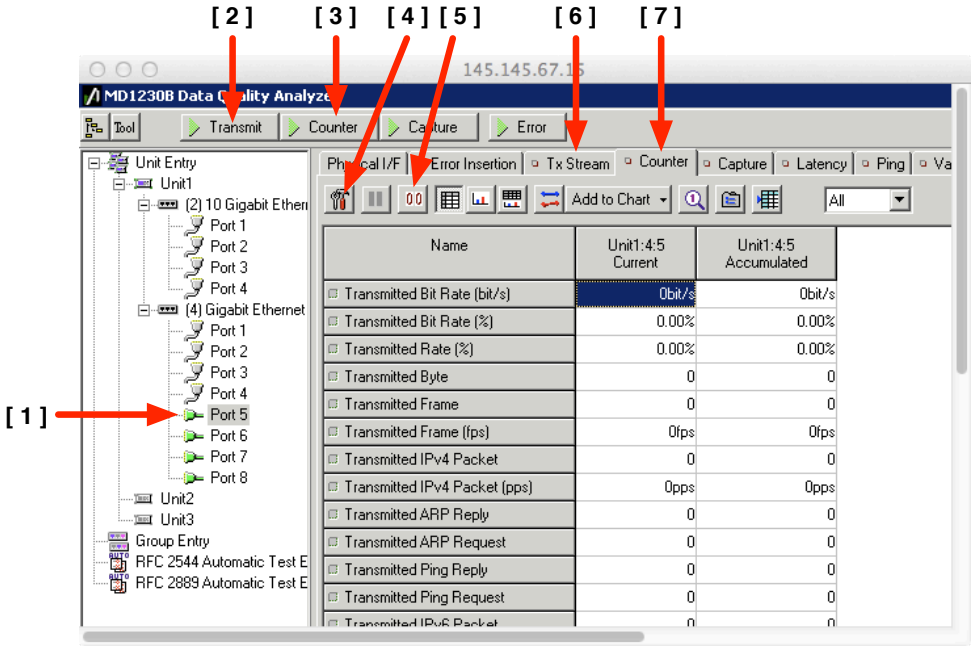

Fig. 5. Anritsu Main Window.

3.3 Simple Streaming Example

Most of our testing was done with equally spaced frames all having the same content. This can be done with the following steps.

(1) To use a port reserve it by right clicking on that port in the left-hand side tree menu [1].
(2) Left click on a port [1] in the left-hand side tree menu. Select the *Tx Stream* tab [6] in the main window and configure one or more streams of the sending port according to the description given in section 3.1.
(3) Start streaming by selecting a port in the left-hand side tree menu and clicking on the *Transmit* button [2] at the top of the window.
(4) (Optionally) Select a port [1] in the left-hand side tree menu and enable counting by selecting the *Counter* button [3] at the top of the window. The counters can be examined by selecting the *Counter* tab [7] of the main window.

4 Lightpath Setup

4.1 Problem Statement

SURFnet wants to use Next Generation Ethernet to provide dedicated circuits (*lightpaths*) to its users. The investigations described in this document show that the Ciena 3960 is able to provision bandwidth limitations for lightpaths to a limited extend.

4.2 Considerations

1. The Ciena 3960 has 8 queues per egress port, so not enough to assign one queue per lightpath.
2. Queues are selected based on R-CoS value, the R-Color is not used for queue selection.

3. The R-CoS value can be a fixed value per port or it can be based on traffic content like the PCP, CFI/DEI, VID or L3 DSCP. The R-CoS value is used to enqueue packets to one of the 8 egress queues.
4. CIR and PIR can be used at the ingress to give packets a *green* R-Color (traffic below CIR threshold) or a *yellow* R-Color (traffic between CIR and PIR).
5. The congestion management profile can be used to prioritise *green* traffic over *yellow* traffic.
6. The congestion management profile should use the (undocumented) yellow admit rate, and not the distinction between TCP-Green and TCP-Yellow. Otherwise it is not possible to distinguish between *green* and *yellow* non-TCP traffic (including UDP traffic), which may cause *green* traffic to be dropped too.
7. The distinction between *green* and *yellow* traffic must be made at each switch, not just at the first (UNI) switch where the lightpath enters the SURFnet network. E.g. The first link may be uncongested, so the first switch allows all traffic (*green* and *yellow*), while the second link is congested, so the second switch must drop some of the (*yellow*) traffic.
8. It is possible to set the priority fields (e.g. the PCP or CFI/DEI) of *yellow* traffic at the egress of the first switch, and use this value in subsequent switches further along the path.

4.3 Configuration

A possible setup is to give each lightpath its own VLAN. The traffic profiling mode at the ingress port can be set to *standard-vlan*. A CIR equal to the guaranteed lightpath bandwidth and a PIR equal to the total capacity of the ingress port can be configured for each VLAN that is used for a lightpath. Other VLANs will get a CIR of 0. Doing so will tag guaranteed traffic with a *green* and all other traffic with a *yellow* R-Color.

All traffic is sent to a single queue, where the yellow admit rate of the congestion management profile prioritises *green* traffic over *yellow* traffic. The network operator must pre-calculate the total of CIR values at all egress ports that send traffic to that egress port and make sure that this value does not exceed its total capacity. While the Ciena 3960 does give warnings for CIR total for an ingress port, it does not seem to give warnings if the total CIR value for these circuits exceeds the available capacity of the egress port.

Resolved CoS Mapping R-Cos Map is irrelevant (as everything ends up in same queue), as long as all traffic is marked green. The default *DefaultFcosRcos* map will do fine.

Ingress Traffic Profiling Ingress Traffic Profiling should be enabled globally and per port:

```
> traffic-profiling enable
> traffic-profiling enable port 1-12
```

For each port, non-confirming traffic (not belonging to a configured lightpath) should be dropped to avoid excess green traffic:

```
> traffic-profiling set port 1 arp-standard-profile bypass nonconform-standard-profile drop
```

For each port, the classifier which identifies a lightpath should be set, and for each classifier, the CIR and PIR should be set (in kbps):

```
> traffic-profiling set port 1 mode standard-vlan

> traffic-profiling standard-profile create port 1 vlan 42
  name vlan42-port1 cir 500000 pir 1000000 statistics on
...
```

Ingress R-CoS to egress queue mapping Each ingress port should be configured to use only one egress queue, e.g. queue 1:

```
> traffic-services queuing queue-map create rcos-map all-in-1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 0 queue 1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 1 queue 1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 2 queue 1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 3 queue 1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 4 queue 1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 5 queue 1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 6 queue 1
> traffic-services queuing queue-map set rcos-map all-in-1 rcos 7 queue 1

> port set port 1 ingress-to-egress-qmap all-in-1
> port set port 2 ingress-to-egress-qmap all-in-1
...
> port set port 12 ingress-to-egress-qmap all-in-1
```

Congestion Management The yellow admit limit should be used on the queue to prioritise *green* packets. The default WRED congestion avoidance profile will do fine (it has a yellow admit limit of 30%):

```
> traffic-services queuing egress-port-queue-group set queue 1 port 1
  congestion-avoidance-profile Default-S-WRED
> traffic-services queuing egress-port-queue-group set queue 1 port 2
  congestion-avoidance-profile Default-S-WRED
...
> traffic-services queuing egress-port-queue-group set queue 1 port 12
  congestion-avoidance-profile Default-S-WRED
```

Egress Scheduling Queue scheduling is irrelevant, as there is only one queue in use.

Egress Shaping Not used by setting it to the maximum egress bandwidth.

References

1. Ciena SAOS Software Configuration Guide release 6.9.0, MAN-0235-001 Standard Revision A, October 2011
2. Anritsu MX123001A Data Quality Analyzer Control Software Operation Manual, 22th edition, Document No. M-W1928AE-22.0, November 2009