# Abschlussbericht
# Qualitätsmessungen im G-WiN

**(TK 602 - NT 201)**

(G-WiN Labor)

Projektleiter: Dr. Peter Holleczek

Version – 1.1

Laufzeit:

1.4.2002 – 30.6.2004

G-WiN Labor

Regionales Rechenzentrum Erlangen (RRZE)

Martensstr. 1

91058 Erlangen

E-Mail: win-labor@dfn.de

18. Oktober 2004

**Inhaltsverzeichnis**

**In diesem Abschlussbericht befinden sich zum Teil vertrauliche Informationen, die ohne Absprache mit dem DFN-Verein nicht an Dritte weitergegeben werden dürfen.**

# 1 Zusammenfassung

Das G-WiN Labor, ein Projekt des DFN-Vereins, hat sich im Rahmen des Projektes mit den folgenden Aufgabenthemen beschäftigt:

- *Entwicklung eines Verfahrens zur Leistungsbeschreibung für den DFN-Internet-Dienst,*
- *Konzeption und Realisierung der Qualitätskontrolle für die G-WiN SDH-Plattform,*
- *Konzepte und Verfahren zu IP-Verkehrsflussmessungen und*
- *Tests betriebsrelevanter Komponenten.*

Das Labor entwickelte ein IPPM-Messprogramm (IP Performance Metrics), um One-Way Delay, Delay Variation und Paketverluste innerhalb des G-WiNs zu messen und entsprechende Statistiken zu berechnen und grafisch darzustellen. Zur Synchronisation der mittlerweile flächendeckend im Wissenschaftsnetz installierten Messstationen werden Zeitsignale interner GPS- bzw. PZF-Empfänger genutzt.

Zur Qualitätskontrolle der SDH/WDM-Plattform des G-WiNs wurde ein Programmsystem entwickelt, welches aufgrund von Performance-Daten des Betreibers des G-WiN-Kernnetzes und von den Routern gesammelten SNMP-Daten eine Bewertung der Verfügbarkeit der Kernnetzverbindungen innerhalb des Netzes berechnet und grafisch darstellt. Zusätzlich zum Kernnetz werden auch die Zugangsleitungen des G-WiNs zu den Anwendern mit Hilfe von Verfügbarkeitsreports und einem Trouble Ticket System qualitativ bewertet.

Für die Verkehrsflussmessungen im G-WiN sammelt eine Accounting-Workstation am Kernnetzknoten in Erlangen die NetFlow-Daten aller im Netz befindlichen Router. Die Daten werden aggregiert, ausgewertet und bieten in Form einer tabellarischen Übersicht einen Überblick über den im G-WiN zwischen verschiedenen Anwendern erzeugten Verkehr. Die ebenfalls berechneten Level 1- Level 1- Verkehrsbeziehungen können zur Berechnung einer geeigneten Netztopologie herangezogen werden. Zusätzlich wurden Cluster- und Mitnutzeranschlüsse im G-WiN detailliert analysiert.

Zu testende betriebsrelevante Komponenten waren in diesem Projekt verschiedene Cisco-Linecards, die hinsichtlich Funktionalität, Performance und Class-of-Service Eigenschaften untersucht wurden. Die Ergebnisse wurden in entsprechenden Testberichten festgehalten.

# 2 Erläuterungen

## 2.1 Aufgabenstellung

Das G-WiN Labor ist ein Projekt des DFN-Vereins (Verein zur Förderung eines Deutschen Forschungsnetzes). Auf das G-WiN Labor kamen im Rahmen des Projektes folgende Aufgaben zu:

- Neue Verfahren der Leistungsbeschreibung für den DFN-Internet-Dienst:

  Ziel ist die Bereitstellung eines Werkzeugs, um sowohl den Ist-Zustand der IP-Dienstgüte im G-WiN festzustellen und zu überwachen, als auch später, den Anwendern gegebenenfalls angebotene differenzierte Service Level Agreements zu überprüfen.

- Konzeption und Realisierung der Qualitätskontrolle für die G-WiN SDH-Plattform:

  Daten aus der SDH/WDM-Plattform und anderen Informationsquellen sollen ausgewertet, aufeinander abgebildet und für die DFN Geschäftsstelle geeignet dargestellt werden. Damit soll zum einen dem DFN-Verein ein Prüfmittel zur Verfügung gestellt werden, um die vom Betreiber des Kernnetzes zugesagten Dienstgüteparameter (Verfügbarkeit, MTBF) auf der SDH-Plattform überwachen zu können und zum anderen, eine zusätzliche Möglichkeit zur Verfügung gestellt werden, um Probleme auf der vom Betreiber bereitgestellten Plattform möglichst frühzeitig erkennen zu können.

- IP-Verkehrsflussmessungen – Konzepte und Verfahren:

  Das Hauptziel ist die Bereitstellung von Verkehrsdaten zur Netzplanung. Trotz der hohen Bandbreiten muss die genutzte Kapazität immer wieder kontrolliert und die Topologie des Netzes gegebenenfalls an die sich verändernden Datenflüsse angepasst werden. Hierzu sind unterschiedliche Verkehrsmatrizen von Interesse: über die Kernnetzknoten (Level1), über die Level1/Level2 Verbindungen und über die Internet-Dienste (G-WiN Anwenderanschlüsse, Übergänge zu GÉANT, DE-CIX und in die USA u.a.).

- Tests betriebsrelevanter Komponenten:

  Um zusätzliche oder neue innovative Mechanismen (z.B. IPv6 Multicast, Class-of-Service-Dienste) nutzen zu können, die Performanz von Netzwerkkomponenten zu erhöhen oder einfach auch nur bestimmte Leitungen auf eine höhere Kapazität aufzurüsten, kann es nötig sein, die im G-WiN eingesetzten Router/Netzwerkelemente mit neuer Hardware auszustatten. Da es sich hierbei häufig um Komponenten handelt, mit denen der DFN-Verein bislang keine Erfahrung sammeln konnte, sollen diese vor ihrem potentiellen Betriebseinsatz vom G-WiN Labor unter Laborbedingungen auf ihre grundsätzlichen funktionalen Eigenschaften untersucht werden, um etwaige auftretende Probleme im Betrieb zu reduzieren.

Die hier beschriebenen Aufgaben werden ergänzt durch die Aktivitäten im Projekt „Router-Test-Labor“, das Markterkundung und Tests von Routern im Terabit-Bereich durchgeführt hat [Rout02].

## 2.2 Voraussetzungen

Der Auftrag wurde unter Projektleitung und in den Räumlichkeiten des Regionalen Rechenzentrums der Universität Erlangen-Nürnberg (RRZE) durchgeführt. Da das RRZE gleichzeitig Anwender des G-WiNs war, konnte der Auftrag in der Betriebsumgebung und unter Nutzung der G-WiN Anschlüsse des RRZE durchgeführt werden. Der Einsatz von vielen Teilzeitstellen (siehe Kapitel 3.7.7) machte eine Arbeitsplanung aufwändig.

## 2.3 Planung und Ablauf des Auftrags

Grundsätzlich verlief das Projekt entsprechend des Projektplans [PROJ01]. In den folgenden Teilbereichen erhöhte sich jedoch der Arbeitsaufwand:

### 2.3.1 IP-Verkehrsflussmessungen

Die Realisierung der Verkehrsflussmessungen musste neu gestaltet werden: bereits in der Vorphase des Projektes wurde der Betrieb des Messsystems mit dem NetFlow Paket von CAIDA aufgenommen und Daten von den Routern konnten zentral auf der Accounting-Workstation empfangen werden. Für eine Abbildung dieser Daten, von denen nur Eingangs-Router und Eingangs-Routerport bekannt sind, auf Level1-Standorte (L1) ist eine Zuordnung der Routerports bzw. Router zu den jeweiligen L1-Standorten nötig. Dies sollte laut Projektplan durch eine Zuordnung der Anwender-Router (AR) zu den Core-Routern (CR; immer am L1-Standort) im GIS (G-WiN-Informations-System) bereitgestellt werden (Bestimmung der Routertopologie). Ferner muss berücksichtigt werden, dass lokal zugestellter Verkehr, der das Kernnetz und die CRs nicht betritt, unberücksichtigt bleiben muss. Deshalb sollten lokale Verbindungen zwischen den ARs aus dem GIS abrufbar sein. Diese geforderten Informationen wurden dem G-WiN Labor jedoch **nicht** zur Verfügung gestellt, so dass eine Aggregation auf L1-Standorte so nicht realisiert werden konnte. Um dennoch eine L1-L1 Matrix zu bekommen, musste eine Methode entwickelt werden, mit der die Routertopologie automatisch erstellt werden kann. Dies realisierte das G-WiN Labor, indem es per SNMP alle AR und CR Router im G-WiN regelmäßig abfragte, um somit eine textuelle - vom DFN-NOC angegebene - Beschreibung (interface-description) aller PoS-Interfaces auszulesen. Diese Informationen werden in eine vom G-WiN Labor aufgebaute Datenbank (G-WiN Accounting System, GAS) auf der Erlanger Accounting-Workstation eingetragen und ausgewertet. Die Güte dieses Lösungsvorschlages ist somit stark abhängig von der Korrektheit der angegebenen Interface-Beschreibungen.

Die lange Ungewissheit, ob die benötigten Router-Topologie Informationen in der GIS-Datenbank zur Verfügung gestellt werden oder nicht und letztendlich die Implementation eines Lösungsvorschlages ohne diese Informationen führten schließlich dazu, dass die ursprünglich vorgesehenen Meilensteine für dieses Arbeitsgebiet so nicht eingehalten werden konnten und umgestaltet werden mussten.

### 2.3.2 Neue Verfahren der Leistungsbeschreibung für den DFN-Internet-Dienst

Von einigen Anwendern wurde der Wunsch geäußert, IPPM-Messstationen in ihren lokalen Netzen aufzustellen, Messungen (innerhalb ihrer Einrichtung oder zwischen ihrer Einrichtung und den im G-WiN aufgestellten Messpunkten) durchzuführen und bei der Interpretation der Ergebnisse behilflich zu sein. Dies war im Projektplan so nicht vorgesehen. Dennoch bemühte sich das Labor weitgehend die Wünsche zu erfüllen. Generell ist das Labor jedoch personell nicht in der Lage, solche Anforderungen zusätzlich mit zu erledigen. Um in Zukunft den Anwendern die Möglichkeit zu geben, von diesem Service generell Gebrauch machen zu können, stehen derzeit Überlegungen an, dies als zusätzlichen Dienst (mit zusätzlichem Personal) durch den DFN-Verein anzubieten.

Im letzten halben Jahr wurden Vorbereitungen getroffen, um IPPM-Messstationen auch im europäischen Wissenschaftsnetz (Geant1-Netz) aufzustellen. Hierzu wurden die Standorte Frankfurt (DANTE), Rom und Poznan (Polen) ausgewählt. Anfang Juli 2004 wurden die Messrechner an die Standorte verschickt und werden bald ihren Betrieb aufnehmen.

## 2.4 Wissenschaftlicher und technischer Stand

„Das Deutsche Forschungsnetz (DFN) ist das von der Wissenschaft selbst verwaltete Hochleistungsnetz für Lehre und Forschung in Deutschland. Es verbindet Hochschulen und Forschungseinrichtungen miteinander und unterstützt die Entwicklung und Erprobung neuer Anwendungen innerhalb der Internet2-Community in Deutschland. Der nationale Backbone des DFN ist das Gigabit-Wissenschaftsnetz G-WiN. Über den europäischen Backbone GÉANT ist das G-WiN mit dem weltweiten Verbund der Forschungs- und Wissenschaftsnetze direkt verbunden“ (www.dfn.de). Das G-WiN Labor ist seit der Inbetriebnahme des G-WiNs Mitte 2000 (und auch schon vorher im B-WiN) an Aufgaben wie Netzabnahmen, Durchführung von Dienstgütemessungen und weiterem beteiligt. Dem Labor steht eine Reihe von Test-Equipment zur Verfügung, das durch Leihstellungen oder explizitem Neukauf immer wieder an die neue Hardware im G-WiN angepasst wurde, so dass durchzuführende Tests mit aktuellem HW-Stand durchgeführt werden konnten.

Parallel zum ersten Projektabschnitt wurde am RRZE außerdem das von der Telekom finanzierte Projekt „Router-Testlabor“ fortgeführt und abgeschlossen, mit dem an vielen Stellen ein reger Informationsaustausch stattfand.

## 2.5 Zusammenarbeit mit anderen Stellen

Aufgrund der betriebsnahen Laborarbeit hatte das Labor engen Kontakt mit dem DFN-NOC, der Telekom, sowie auch mit den Herstellern der eingesetzten Geräte (wie Agilent, Spirent, Cisco, Imtech, Juniper, Endace, Meinberg, Dell, bee GmbH, Frasch, RCE).

# 3 Eingehende Darstellung

## 3.1 Vorbemerkungen

- Zum voraussichtlichen Nutzen, insbesondere der Verwertbarkeit des Ergebnisses:

  Die angestellten Untersuchungen und die entwickelten Methoden erlauben eine wirksame Qualitätskontrolle des G-WiN. Für eine nachhaltige Kontrolle bedarf es dauerhafter Anstrengungen und einer ständigen Verfeinerung der Methoden.

- Zu den während der Durchführung des Forschungs- und Entwicklungs-Auftrags dem Auftragnehmer bekannt gewordenen Fortschritten auf dem Gebiet des Auftrags bei anderen Stellen:

  In den USA wurde annähernd zeitgleich für das Internet2 auch ein IP Performanzmesssystem entwickelt (OWAMP), das dem IPPM-System des G-WiN Labors sehr ähnelt: http://e2epi.internet2.edu/owamp/overview.html) Linux-PCs, Uhren-Synchronisierung mittels NTP und Hardware-Uhren. Es gibt bei OWAMP ein eigenes Konfigurationsprotokoll, über das sich zwei Messrechner z.B. über die Ports einigen. Die IETF arbeitet derzeit an einem Standard für ein Protokoll für One-Way-Delay-Messungen (OWDP, http://www.ietf.org/internet-drafts/draft-ietf-ippm-owdp-07.txt). Dadurch sollen verschiedene Messboxen Messpakete austauschen können. OWAMP ist praktisch eine Beispielimplementation des OWDP. Dadurch, dass das IPPM-System des G-WiN Labors bereits entwickelt wurde, bevor es den Standard gab, richtet sich dieses System (noch) nicht danach. Dafür ist es nun schon seit einiger Zeit an allen Kernnetzstandorten im G-WiN in Betrieb.

- Die erzielten Ergebnisse werden im Folgenden im Detail beschrieben. Die Gliederung folgt dem Antrag.

- Die erfolgten oder geplanten Veröffentlichungen sind in Abschnitt 3.7.1 zu finden.

## 3.2 Einleitung

Seit Mitte 2000 ist das Gigabit-Wissenschaftsnetz (G-WiN) in Betrieb. Entsprechend der Aufgabenstellung im Rahmen des diesem vorangegangenen Entwicklungsprojektes „Leistungsmessungen im G-WiN“ (TK 602-NT118) war das G-WiN Labor aktiv an der Inbetriebnahme beteiligt [Glab01]. Eine Fortsetzung erfolgte im April 2002 mit diesem Projekt. Das Vorhaben endet vertragsgemäß im Juni 2004. Im Einzelnen sollten vom G-WiN Labor folgende Arbeitsbereiche abgedeckt werden:

- Neue Verfahren der Leistungsbeschreibung für den DFN-Internet-Dienst,
- Konzeption und Realisierung der Qualitätskontrolle für die G-WiN SDH-Plattform,
- IP-Verkehrsflussmessungen – Konzepte und Verfahren,
- Tests betriebsrelevanter Komponenten,
- Begleitende Aktivitäten.

Die Ergebnisse des Projekts wurden halbjährlich protokolliert und liegen in Form von Zwischenberichten vor ([Glab02], [Glab03], [Glab04]). Die dort beschriebenen Aufgaben wurden ergänzt durch die Aktivitäten im Router-Test-Labor, das Markterkundung und Tests von Routern im Terabit-Bereich durchgeführt hat [Rout02]. Aktuelle Informationen über die Tätigkeiten im G-WiN Labor finden sich unter www.win-labor.dfn.de.

Dieser Abschlussbericht liefert einen umfassenden Überblick über das Gesamtergebnis des Projekts TK 602 - NT 201.

Im Folgenden werden die Aktivitäten ausführlicher beschrieben.

## 3.3 IP-Dienstgüteüberwachung

### 3.3.1 Stand der Entwicklungen

Im nationalen und internationalen Umfeld des DFN gibt es einige Ansätze und Konzepte auf dem Gebiet der Messungen zur Dienstgüteüberprüfung:

- Im Projekt *QUASAR* (Quality of Service Architectures, Fraunhofer Gesellschaft Fokus und Universität Stuttgart) wurden Konzepte für die Bereitstellung von unterschiedlichen Dienstklassen im G-WiN erarbeitet [Qu01]. Das Projekt ist abgeschlossen.

- RIPE hat schon 1997 das *Test Traffic Project* gestartet [RI01]. Ziel dieses Projektes ist es, Performance Parameter für das Internet zu messen. Dazu werden mit GPS Empfängern ausgestattete PCs in Netzwerken verschiedener Provider installiert. Von den Testboxen aus werden kontinuierlich das One Way Delay und die Paketverluste zwischen den Messpunkten gemessen. Die Daten sollen Hilfe bei der Suche nach den Ursachen von Problemen im Netzwerk bieten und zu Kapazitätsplanungen herangezogen werden. In dem Projekt werden darüber hinaus neue Techniken für Performance-Messungen untersucht.

- CESNET hat Mitte 2001 ein „*low cost*“ Werkzeug vorgestellt, um aktive Messungen in Netzwerken durchzuführen. Das Werkzeug basiert auf dem RUDE/CRUDE Paketgenerator und –analysator (Real-time UDP Data Emitter/Collector for RUDE) [CR01]. Dabei werden zur Synchronisation der Uhren GPS Empfänger genutzt. Die Weiterentwicklung des Systems erfolgt im SCAMPI-Projekt [SCAMPI00]. „SCAMPI ist primär für passives Monitoring bestimmt, aber es kann auch als programmierbarer Paketgenerator für aktive Messungen eingesetzt werden“ [SCAMPI01].

- Das G-WiN Labor hat bereits frühzeitig Kontakt zur Universität Waikato in Neuseeland und dem National Laboratory for Applied Network Research (NLANR) aufgenommen. Dort wurden für das Projekt „Passive Measurement & Analysis“ (PMA) Netzwerkkarten für Dienstgütemessungen entwickelt [PMA01]. Bei diesem Projekt werden von allen Paketen an verschiedenen Netzpunkten in der USA und Neuseeland über einen optischen Splitter Kopien an ein vor Ort installiertes Monitorsystem, bestehend aus einem Personal Computer mit FreeBSD und der entwickelten Netzwerkkarte geschickt, die den Header aus den IP-Paketen extrahiert und sie mit einem Zeitstempel versieht, bevor sie zur weiteren Auswertung an einen zentralen Server gesendet werden.

Weiterhin besteht Kontakt zu Mitarbeitern des AMP-Projektes (Active Measurement Project) [AMP01]. Anders als beim PMA-Projekt werden hier Testpakete zwischen verschiedenen Rechnern verschickt, um Informationen über die RTT (Round-Trip-Time) und Verluste des Netzes zu bekommen. Im G-WiN Labor ist ein passives Messsystem im Einsatz, in Kooperation mit der Universität Leipzig konnten bereits Delaymessungen durchgeführt werden.

- Im G-WiN besteht seit Ende 2003 ein vom G-WiN Labor komplett entwickeltes IPPM-Messsystem (IP Performance Metrics), welches zwischen den Kernnetzstandorten voll vermascht One-Way Delay, One-Way Delay Variation und Packet Loss Daten ermittelt. Derzeit ist kein weiteres System in dieser Größenordnung bekannt.
- INTERNET2 entwickelt im Rahmen der „End-to-End Performance Initiative“ ein ebenfalls mit ntp over GPS basierter Zeitsynchronisation ausgestattetes Performance Messsystem, das bereits im Abilene-Netz eingesetzt wird [OWAMP1]. Dort wird derzeit an einem Internetstandard gearbeitet (A One-way Active Measurement Protocol Requirements [OWDP01]). Kontakte dorthin sind aufgebaut, Austausch von Daten wird praktiziert.
- GEANT2 wird ein Performance Messsystem beinhalten. Das G-WiN Labor bereitet sich derzeit auf eine Installation des hier entwickelten Systems vor.
- Parallel zu den Messungen innerhalb des G-WiN führt das Labor Testmessungen in und zwischen Anwendernetzen durch. Diese Messungen werden mit mobilen Messeinheiten realisiert und befinden sich in einer späten Erprobungsphase.

### 3.3.2 Das IPPM-Messprogramm des G-WiN Labors

Bei diesem Messprogramm handelt es sich um aktive Messungen. Im Gegensatz zu passiven Messungen (siehe Kapitel 3.3.5), bei denen der tatsächliche Betriebsverkehr des Netzes analysiert wird, werden bei aktiven Messungen spezielle Testpakete erzeugt und zusätzlich mit ins Netz eingeschleust. Eine Sendestation (Linux-PC, Pentium 4) erzeugt Gruppen von UDP-Paketen in konfigurierbaren Abständen, versieht jedes einzelne Paket mit einem aktuellen Zeitstempel sowie einer Sequenznummer und sendet sie zu einer Empfangsstation. Diese bestimmt wiederum die aktuelle Empfangszeit und schreibt die gesammelten Daten in eine Logdatei. Daraus lassen sich dann One-Way Delay, Delay Variation und Paketverluste berechnen. Der zeitliche Abstand zwischen den Gruppen, die Anzahl der Pakete pro Gruppe und die Paketgröße sind konfigurierbar. Minimum, Maximum und Median des One-Way Delays zu jeder Gruppe lassen sich bestimmen und grafisch darstellen. Der Median dient insbesondere dazu, einzelne "Ausreißer" in den Messungen zu eliminieren. Die im IP-Header definierbaren Precedence Bits im ToS-Feld werden explizit gesetzt und somit einer speziellen Dienstklasse zugeordnet. Da bislang keine Unterscheidung von verschiedenen Dienstklassen im G-WiN stattfindet, werden die Precedence Bits aller Testpakete derzeit einheitlich auf 0 gesetzt. Bei den aktuellen Messungen wird auf jeder Messstrecke alle 30 s eine Gruppe von 5 Testpaketen verschickt, wobei ein zeitlicher Abstand der Pakete in einer Gruppe von 5 ms eingehalten wird. Dieser Offset ist nötig, um eine Verfälschung der Messung durch eine unbeabsichtigte Wartezeit in der Netzwerkkarte zu vermeiden.

Für jede Messstrecke gibt es einen Sende- und einen Empfangsprozess. Alle Sendeprozesse einer Messstation müssen mit einem Offset gestartet werden, damit sich die Prozesse nicht

gegenseitig behindern und somit alle Pakete zeitlich versetzt erzeugt werden können. Eine analoge Problematik besteht auch auf der Empfängerseite.

Um dort weitgehend sicherzustellen, dass nicht mehrere Pakete gleichzeitig ankommen, werden zusätzlich die Sendeprozesse aller Messstationen, die zur selben Empfangsstation senden, zeitlich so versetzt gestartet, dass deren Pakete unter normalen Netz-Bedingungen – also auf nicht stark überlasteten Wegen – auch zu verschiedenen Zeitpunkten ihr Ziel erreichen.

Um ausreichend genaue One-Way Delay Messungen zu erzielen, ist es wichtig, die Messstationen zeitlich sehr genau miteinander zu synchronisieren. One-Way Delays im G-WiN liegen im Bereich unter 10 ms. Dies macht eine maximale Abweichung der Uhren von weniger als 0,1 ms nötig. Somit reicht eine Zeitsynchronisation mittels NTP (Network Time Protocol) [NTP] über externe Zeitserver nicht aus, so dass auf das Zeitsignal interner GPS- bzw. DCF77/PZF-Empfänger zurückgegriffen wurde. Jedoch zieht GPS einige Nachteile mit sich:

- Durch die zusätzlichen Hardware (GPS-Empfänger und Antenne) und eine zum Teil sehr aufwändige Installation der Antenne entstehen Zusatzkosten.
- Es können nur an geeigneten Standorten GPS-Antennen aufgestellt werden, an denen es möglich ist, die Antenne an einem Ort zu platzieren, der freie Sicht zu den Satelliten hat.
- Es werden spezielle Treiber benötigt, um von den Messworkstations auf die Zeitdaten des GPS-Empfängers zugreifen zu können. Für die im G-WiN Labor für dieses Projekt ursprünglich vorgesehenen SUN-Workstations (SunBlade 100) mit Solaris-Betriebssystem stehen jedoch keine Treiber zur Verfügung. Aus diesem Grund fand ein Wechsel auf Messstationen (PCs) mit Linux-Betriebssystem statt, für die die nötigen Treiber vorhanden sind.

### 3.3.3 Ausbaustand - Ausstattung

Im Sommer 2002 wurde die erste PC-Messstation mit GPS-Antenne in Erlangen erfolgreich in Betrieb genommen. Bis November 2003 wurden schließlich an allen G-WiN Kernnetzstandorten (10 Level 1 und 17 Level 2 Standorte) Messstationen zur Ermittlung von Leistungsdaten in Betrieb. An manchen Standorten wird über eine zweite Netzwerkkarte die lokale Einrichtung mit überwacht. Darüber hinaus besitzt das G-WiN Labor mehrere mobile Messstationen mit PZF-Zeitsynchronisation, die flexibel an variablen Standorten eingesetzt werden können. So ist eine flächendeckende Messungen im Netz möglich. Die erzeugten Daten werden sowohl lokal an den Messstationen gespeichert, als auch zentral auf einem Datenserver im Labor gehalten. Dort werden auch ein Analyserechner und ein Webserver zur Datendarstellung betrieben. Die Anzahl der überwachten Verbindungen beläuft sich derzeit auf ca. 1200.

Jeder IPPM-Messrechner ist über dem ISDN-Telefon in einem der DFN-Schränke bei der Einrichtung am Kernnetzstandort montiert (siehe Abbildung 1).

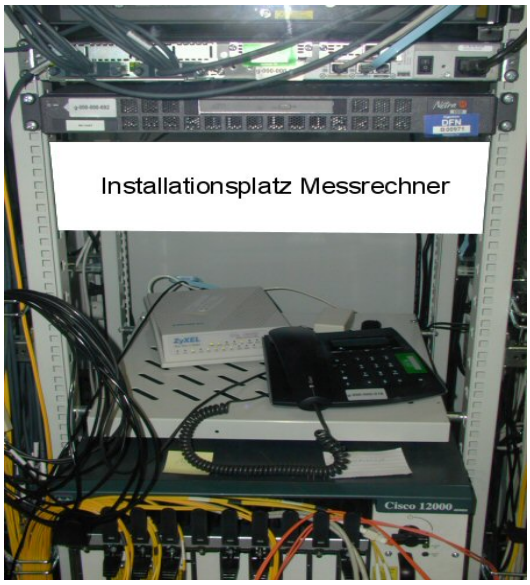



**Abbildung 1: Installation des Messrechners.**

Die Anbindung an das G-WiN wird über die eine Fast Ethernet Karte („Netzkabel DFN“) des PCs ermöglicht, die mit dem Cisco-Switch über der Messstation verbunden wird (siehe Abbildung 2). Die zweite Netzwerkkarte steht zur Verfügung, falls die Einrichtung auch den Zugang vom Anwendernetz zum Kernnetz überwacht haben möchte. Derzeit werden neun dieser Anbindungen bereits genutzt. Die Uhrensynchronisation erfolgt über die GPS-Antenne. Diese ist an der BNC-Buchse des GPS-Empfängers („GPS-Antenne“) angeschlossen. Ein „Verbindungskabel“ verbindet die Weasel-Karte (siehe [Glab02]) des PCs mit der Tastatur, um per Terminal Server remote auf die BIOS-Konfigurationsoberfläche der Messstation zugreifen und diese administrieren zu können.

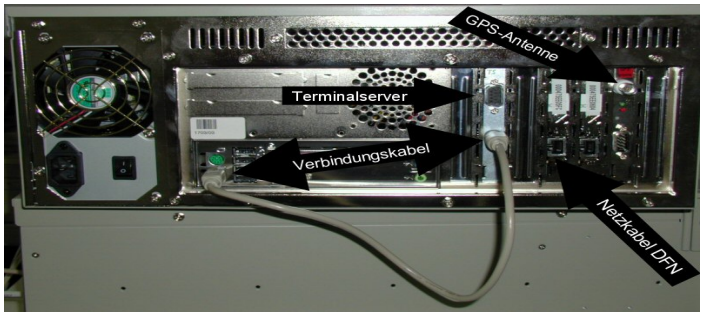



**Abbildung 2: Rückseite einer Messstation.**

Die IPPM-Mess-PCs wurden jeweils in Erlangen vorinstalliert, konfiguriert und getestet. Gut die Hälfte aller Messrechner wurden vom G-WiN Labor selbst zur Einrichtung transportiert und dort angeschlossen. In den anderen Fällen wurden die PCs zur jeweiligen Einrichtung verschickt und dort von der Einrichtung in Eigenregie montiert.

An zwei Standorten konnten die bei der Einrichtung bereits vorhandene GPS-Antennen mit verwendet werden. Für die anderen wurden eigene Antennen gekauft. Zwei Einrichtungen übernahmen dabei selber die Verlegung des Antennenkabels von der Messstation bis zu einem geeigneten Ort, an dem die Antenne freie Sicht zu den GPS-Satelliten hat. Bei den anderen

Standorten wurde hierfür eine Firma beauftragt. Aufgrund baulicher Gegebenheiten wurde ein Standort anstelle mit einer GPS-Antenne mit einem PZF/DCF77-Empfänger ausgestattet.

Einige Einrichtungen möchten die installierten GPS-Antennen mitnutzen. Hierzu ist es möglich, eine eigene Station mit GPS-Karte über einen Splitter an die Antenne anzuschließen. Eine zweite Variante ist die Nutzung der zwei asynchronen, seriellen Schnittstellen der in der Messstation eingebauten GPS-Karte, um das sekündliche *Meinberg Standard Telegramm* oder das automatische *Capture Telegramm* auszulesen.

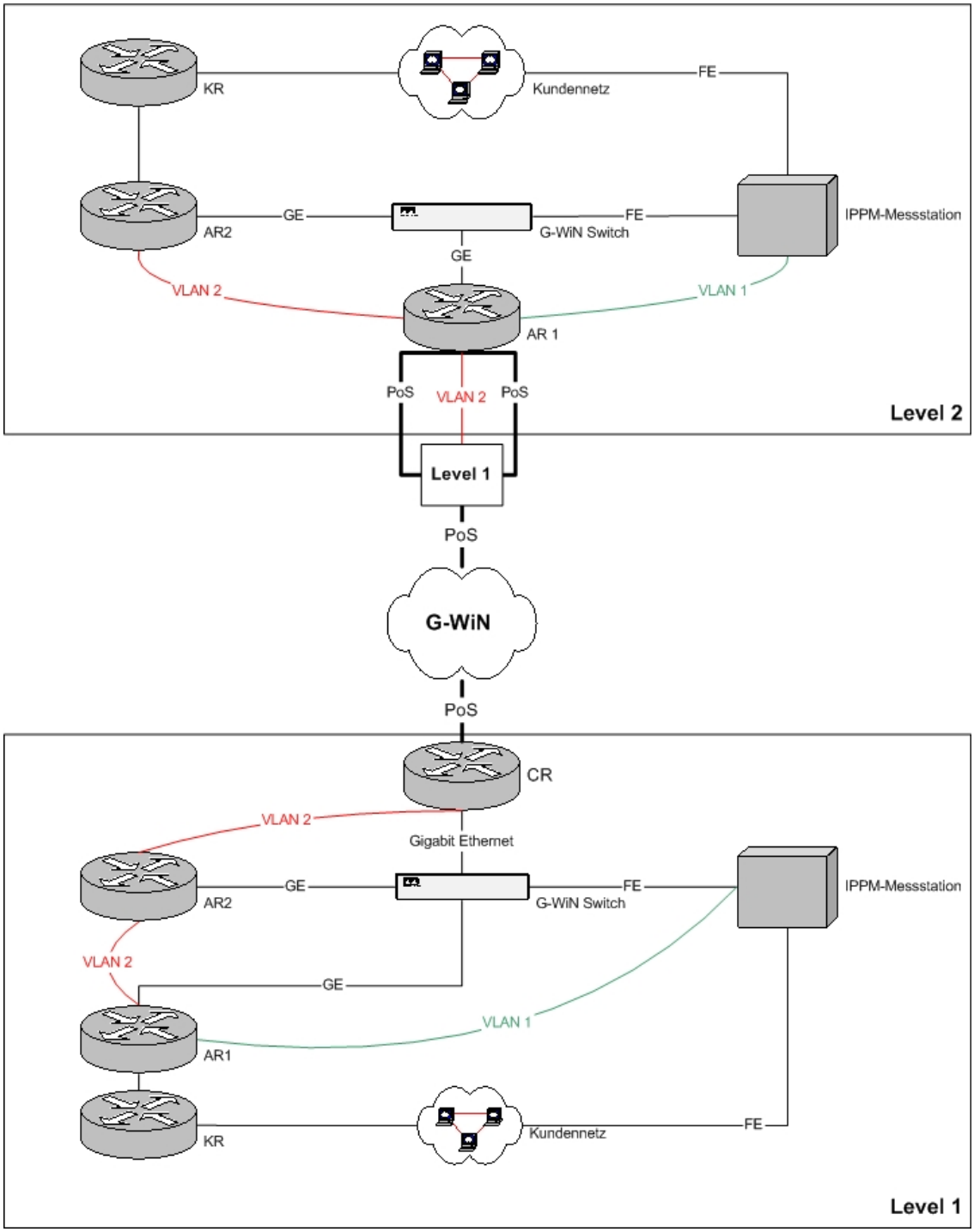



**Abbildung 3: Prinzipieller physikalischer und logischer Aufbau der Kernnetzstandorte (AR: Access-Router, KR: Kunden-Router, CR: Core-Router).**

In Abbildung 3 ist zu sehen, wie die IPPM-Messstationen an den Kernnetzstandorten physikalisch und logisch angebunden sind. Über eine Fast Ethernet Schnittstelle sind die Messrechner an dem DFN-Switch des Kernnetzstandortes angeschlossen. Ein logisches VLAN verbindet sie jeweils mit einem der dort vorhandenen Access-Router (AR).

### 3.3.4 Auswertung und Interpretation der Daten

Die auf Tagesbasis ausgewerteten Messergebnisse sind auf dem Web-Server des Labors (www.win-labor.dfn.de) dargestellt.

Nach der Installation der Messstationen an den verschiedenen G-WiN Standorten, wurde mit der eigentlichen Analyse und Interpretation der Messdaten begonnen. Erste Ergebnisse und Interpretationen wurden in [Pearl03]] zusammengefasst und zeigen, dass z.B. die Qualität von Videoübertragungen empfindlich von der Höhe und Streuung der gemessenen Delays abhängt. Dass das Delay auf einer Strecke zeitlich nicht konstant ist, zeigt folgendes Beispiel:

Bei der Betrachtung einzelner Tagesstatistiken sieht man, dass es auf manchen Strecken zur Hauptverkehrszeit zu stärkeren Schwankungen des Delays kommt. Oft sind es nur einzelne „Ausreißer“ mit erhöhten Delays, die in dieser Zeit zu sehen sind, in ein paar Fällen sieht man jedoch eine deutliche Streuung der Delays. Abbildung 4 zeigt eine Messung zwischen einer Einrichtung und dem nächsten Level 1 Kernnetzknoten. Die Streuung der Delays (oberes Teilbild) ist in den frühen Morgenstunden geringer als zur Hauptverkehrszeit. Der Delay-Anstieg entspricht in diesem Fall dem Anstieg im Verkehrsaufkommen (unteres Teilbild), der aus den Statistik-Daten des CNM des DFN-Vereins gewonnen werden konnte [CNM]. In anderen Fällen lässt sich der Delay-Anstieg nicht auf eine höhere Leitungsauslastung zurückführen.

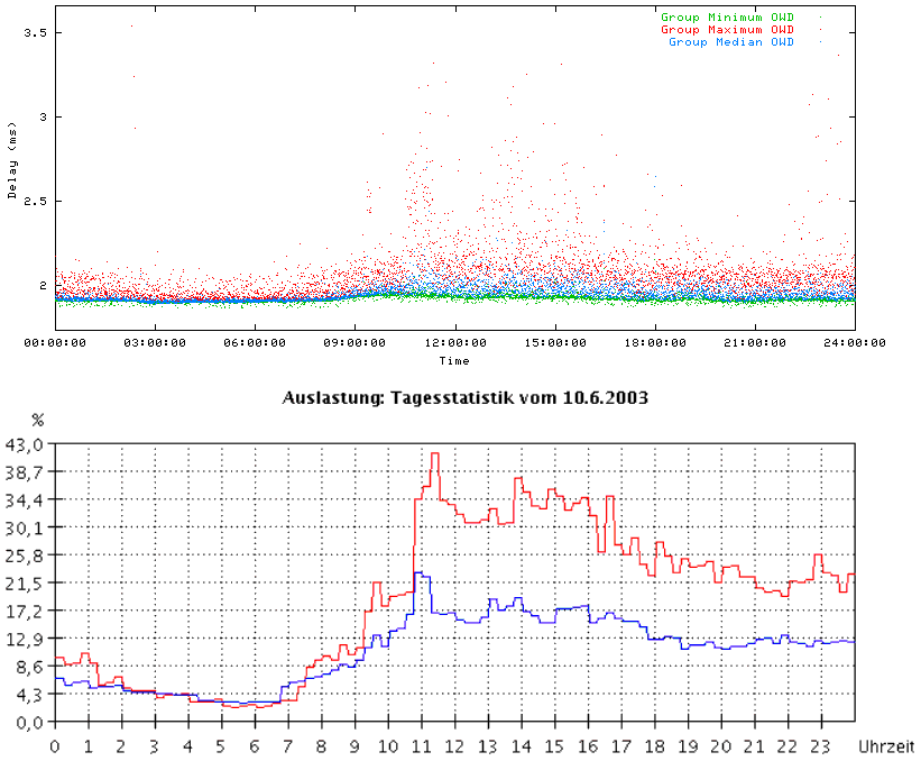



**Abbildung 4: Anstieg des Delays zur Hauptverkehrszeit (oberes Teilbild); vom CNM ermittelte Auslastung der gemessenen Strecke (unteres Teilbild).**

Sobald eine neue Messstation in Betrieb genommen wird, beginnt sie zu allen anderen bereits vorhandenen Messstationen Messungen durchzuführen. Diese Messungen sind sowohl in Form

von abrufbaren Tagesstatistiken als auch in Form einer sich in etwa alle 10 Minuten aktualisierenden Leitungstopologie (sog. „Weathermap“) zu sehen, in der mit unterschiedlichen Farben die aktuellen OWDs der Messstrecken gekennzeichnet sind (www.win-labor.dfn.de). Dabei werden nur Messungen entlang der SDH-Verbindungen berücksichtigt (siehe Abbildung 5).

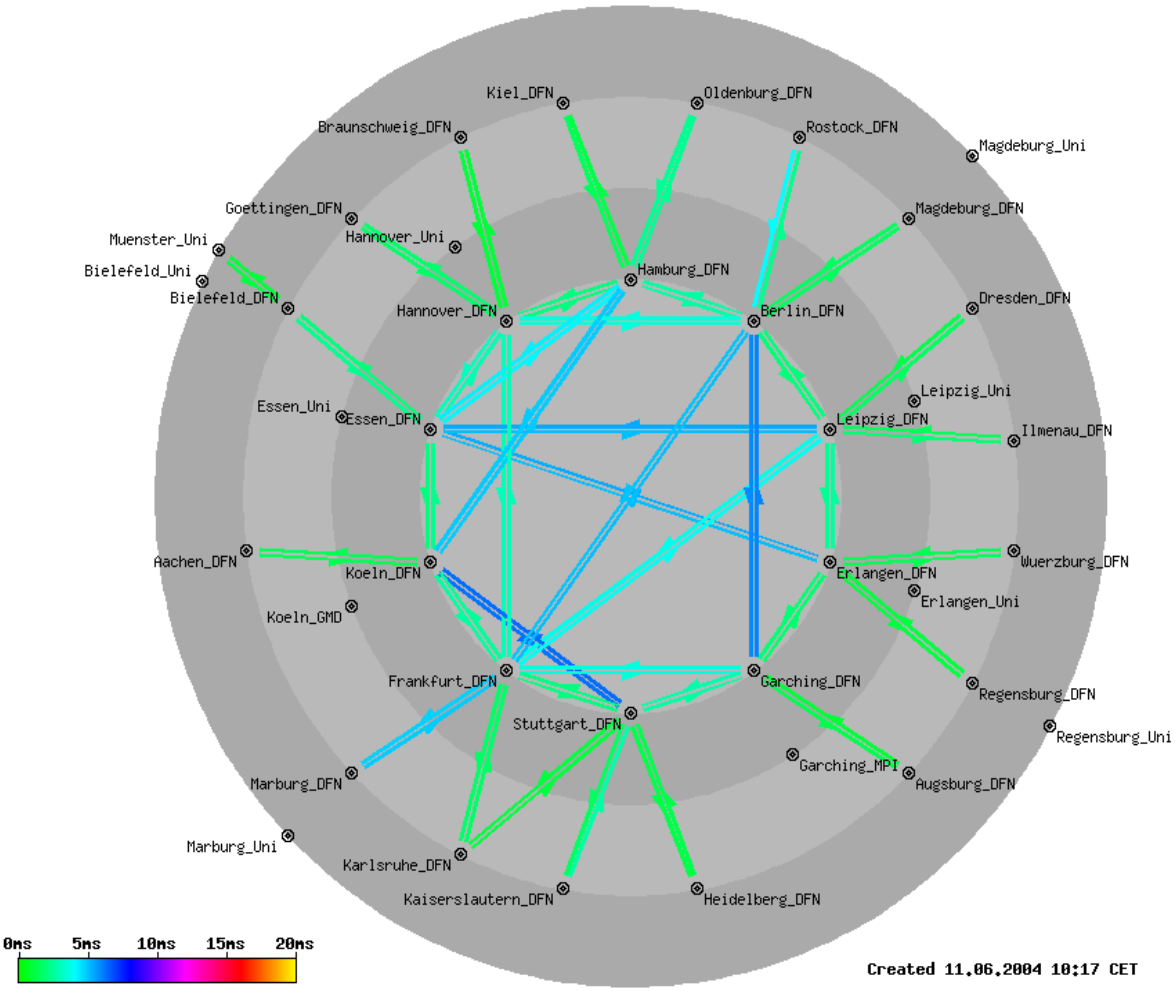



**Abbildung 5: Aktuelle OWDs zwischen DFN-Standorten.**

Das OWD einer Verbindung wird dabei aus dem Median der letzten 21 angekommenen Pakete zum Zeitpunkt des Updates der Kartendarstellung ermittelt.

Im Gegensatz dazu werden bei den abrufbaren Tagesstatistiken alle Pakete analysiert. Hierbei werden die Strecken streng sequentiell ausgewertet. Sobald die Auswertung der letzten Strecke vorliegt, wird wieder mit dem neuen Datensatz der ersten Messstrecke begonnen. Um die derzeitigen Strecken (ca. 1200 Stück) alle einmal abgearbeitet zu haben, dauert es ca. 22 Minuten.

Um mit Hilfe der OWD-Ergebnisse aktuelle Probleme möglichst rasch lokalisieren zu können - z.B. eine mit hohen Delays belastete Teilstrecke während einer Videokonferenz - ist es nötig, die ausgewerteten Informationen möglichst in „Echtzeit“ darzustellen. Aus diesem Grund

wurde das Auswerteprogramm verbessert, in dem die Zwischenergebnisse binär und nicht in Form von ASCII-Texten abspeichert werden und somit der erneute Zugriff zur Weiterverarbeitung der Daten schneller erfolgen kann.

### 3.3.5 Messkonzepte anderer Projekte: NLANR (passive Messungen)

Während des Projekts beschäftigte sich das G-WiN Labor auch mit passiven Messmethoden. Hierfür wurde ein PC mit einer Gigabit-Ethernet-Messkarte der Firma Endace Measurement Systems Ltd. [Endace] beschafft. Diese Messkarte wurde mittels eines Lichtwellenleiter-Splitters in die Gigabit-Ethernet-Verbindung zwischen einem Access-Router und dem Standort-Switch in Erlangen eingeschleift. Die Karte ist mit Hilfe des GPS-Zeitempfängers der IPPM-Messstation Erlangen zeitsynchronisiert. Dies erlaubt es, Teile der Ethernet-Frames zusammen mit hochpräzisen Zeitstempeln auszuwerten. Misst man mit zwei solcher Karten an verschiedenen Stellen im Netz, so können Pakete, die an beiden Messkarten erfasst werden, identifiziert werden. Dazu werden typischerweise die ersten 56 Bytes jedes Paketes aufgezeichnet und dann verglichen. Da nur ein kurzes Zeitintervall in Frage kommt, in dem das Paket zu finden sein muss, reichen die gelesenen Headerinformationen (Quell- und Zieladresse, Checksumme, Paketlänge, ...) zur Identifikation aus. Aus datenschutzrechtlichen Gründen werden die Paketinformationen anonymisiert.

Erste Messungen mit dieser Karte wurden bereits durchgeführt. Es wurde analysiert, wie sich die Auslastung der Gigabit-Ethernet-Strecke in Zeitschlitzen von unter 1 s verhält (siehe Abbildung 6). Dies erlaubt genauere Auslastungsaussagen als die langfristigen Mittelwerte, wie sie z.B. über Portcounter der G-WiN Router ermittelt werden können.

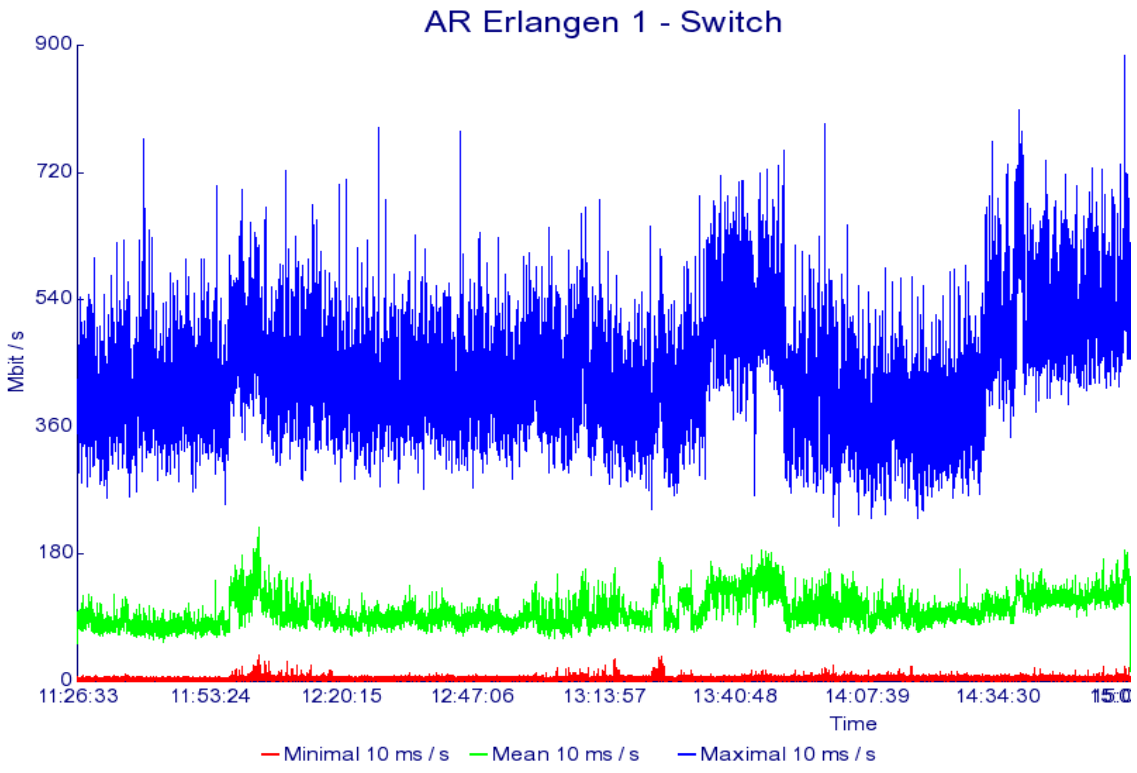



**Abbildung 6: Auslastung der Gigabit-Ethernet Strecke.**

Des weiteren wurde eine Analyse der Paketgrößenverteilung gemacht. Diese Werte sind wichtig für die im G-WiN Labor durchgeführten Tests von Netzwerk-Hardware.

Weiterhin fanden erste Messungen in Zusammenarbeit mit dem Rechenzentrum der Universität Leipzig statt, welches ebenfalls Messkarten von Endace einsetzt [Trace]. Hier wurden unter anderem die Delays der Pakete zwischen den beiden Standorten gemessen (siehe Abbildung 7). In Abhängigkeit der Paketgröße (zwischen 40 Byte und 1500 Byte) sind die Delaywerte mit unterschiedlichen Farben dargestellt. Gleichzeitig wurde mit dem aktiven IPPM-Tool zwischen Erlangen und Leipzig gemessen (siehe Abbildung 8). Die Messung erfolgte hier mit 429 Byte großen Paketen, was in etwa der durchschnittlichen Paketgröße im Wissenschaftsnetz entsprach. Die Pakete der beiden Messmethoden nehmen allerdings nicht exakt denselben Weg. Bei den aktiven Messungen befindet sich ein Router mehr auf der Strecke. Daher ist das Delay, das mit der passiven Messmethode ermittelt wurde, etwas geringer als das der aktiven Messung. Die Streuung der Delays ist bei der passiven Messung ebenfalls etwas geringer. Dies ist nicht anders zu erwarten, da der Zeitstempel per Hardware in der Endace-Karte erzeugt wird. Bei der aktiven Messbox muss das Paket erst vom Betriebssystem empfangen und an die Applikation weitergeleitet werden, bevor der Zeitstempel erzeugt und eingefügt werden kann. Die Ergebnisse der aktiven und passiven Messungen können daher als konsistent betrachtet werden.

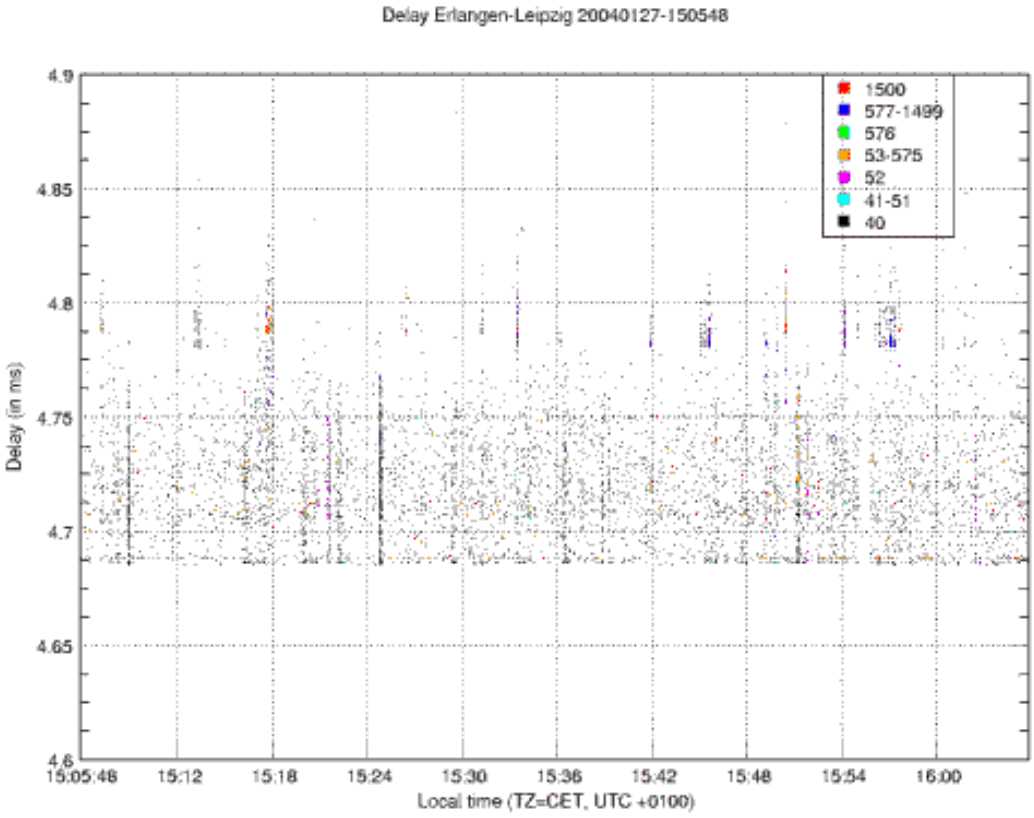



**Abbildung 7: Delay Erlangen-Leipzig (passive Messung).**

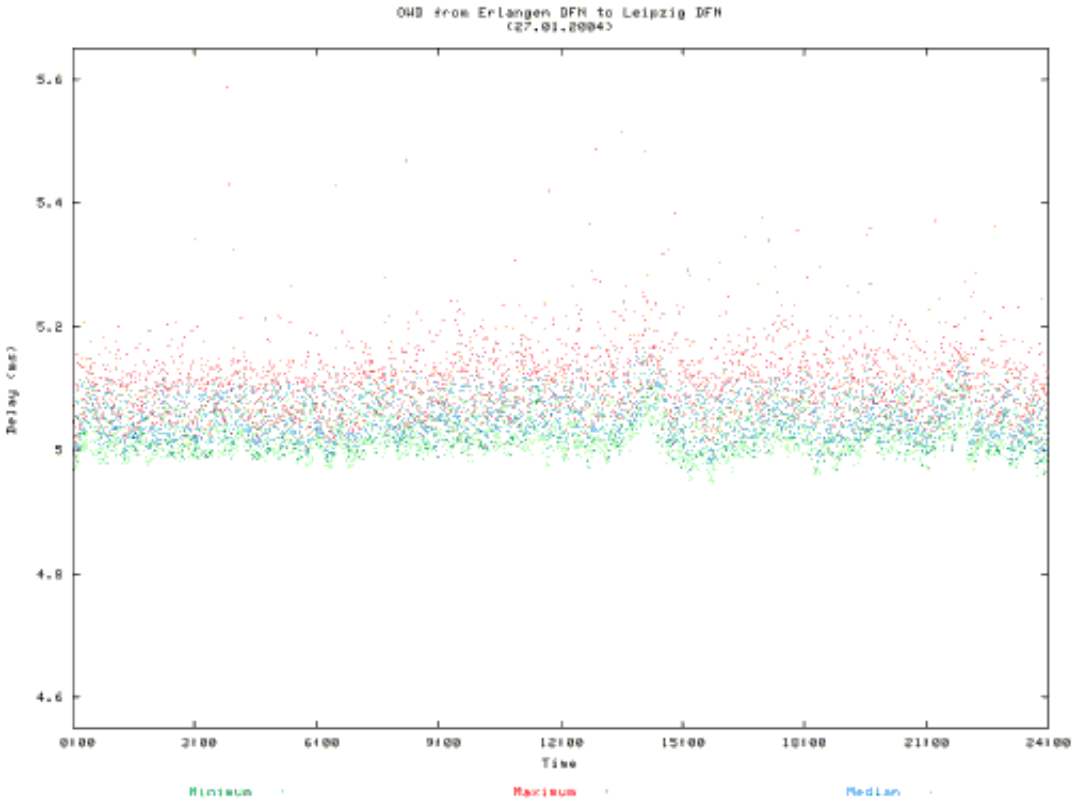



**Abbildung 8: Delay Erlangen-Leipzig (aktive Messung).**

Aufgrund der positiven Erfahrungen mit dieser Messkarte ist der Aufbau eines zweiten passiven Messsystems für das G-WiN Labor geplant. Damit werden dann weitere Analysen von Paketlaufzeiten im G-WiN möglich.

## 3.4 SDH/WDM Qualitätskontrolle

### 3.4.1 Stand der Entwicklungen, Ausstattung

Im Rahmen der Qualitätskontrolle im G-WiN für die SDH/WDM-Infrastruktur werden Daten aus der SDH/WDM-Plattform und anderen Informationsquellen ausgewertet und aufeinander abgebildet. Die Auswertung ist derzeit nur für autorisierte Stellen (DFN-GS Berlin) zugänglich.

Realisiert wurde die Bereitstellung von Kenngrößen des WDM/SDH- und IP-Netzes:

- Durch die kontinuierliche, tägliche Berechnung relevanter Leistungsparameter ist die Früherkennung von Problemen auf der vom Betreiberbereitgestellten Plattform möglich.
- Es wurde ein Modell entwickelt, das die Grundlage für die Berechnungen bildet. Die berechneten Daten werden für die Geschäftsstelle aufbereitet.
- Um etwaige Probleme im IP-Dienst richtig interpretieren zu können, muss ein Überblick über die Qualität des darunter liegenden SDH/WDM Dienstes möglich sein.

Zur Realisierung der Qualitätskontrolle bedarf es der Sammlung/Klassifikation von Betriebsdaten und der Entwicklung eines Berechnungsverfahrens sowie einer Archivierungsmöglichkeit. Hierfür steht dem Labor ein Datenbankserver zur Speicherung, ein Analyserechner zur Auswertung und ein Web-Server zur Darstellung der Daten zur Verfügung.

### 3.4.2 Sammlung/Klassifikation von Betriebsdaten

Es liegen monatliche Verfügbarkeitsberichte für einen Teil der Leitungen vor, die von betrieben werden. Zugangsleitungen

Zur Bestimmung der Verfügbarkeiten der Zugangsleitungen liegen die relevanten Informationen im Rahmen der entsprechenden Möglichkeiten (Trouble Ticket System) vor.

Über das Trouble-Ticket-System des G-WiN der T-Systems wird die Güte der Zugangsleitungen im G-WiN bestimmt. Die von der T-Systems zur Verfügung gestellten Trouble-Ticket-Listen werden in Form von monatlichen und jährlichen Diagrammen für die Zugangsleitungen an die DFN-Geschäftsstelle Berlin weitergeleitet. In den monatlichen Diagrammen werden die im Trouble-Ticket-System dokumentierten Ausfälle nach Verursacher und Dauer differenziert. Die Ermittlung des Verursachers erfolgt in wöchentlichen Telefonkonferenzen mit der T-Systems. In der jährlichen Übersicht (siehe Abbildung 9) wird unterschieden zwischen der **T**-Systems als Verursacher, einer **U**nbekannten Ursache (meist sogenannte Kurzzeitausfälle ohne ermittelbaren Grund) und **anderen** Ursachen, die zusammengefasst dargestellt werden. Für Zugangsleitungen, die von anderen Carriern bereitgestellt werden, erfolgt die Bewertung auf der Grundlage monatlicher Verfügbarkeitsberichte. Der Inhalt dieser Reports ist je nach Carrier allerdings sehr unterschiedlich und kann deshalb nicht automatisiert ausgewertet werden. Nicht alle Carrier stellen diese Berichte zur Verfügung. Deshalb werden die vorhandenen Daten vom G-WiN Labor derzeit vorrangig gesammelt und nach Möglichkeit mit eventuell vorhandenen Trouble-Tickets aus dem Trouble-Ticket-System der T-Systems abgeglichen. Eine grobe Bewertung der Carrier-Reports geht an die DFN-Geschäftsstelle in Berlin.

Auf diesen Grundlagen werden die Verfügbarkeiten berechnet und in das GIS (G-WiN-Informationssystem) eingetragen.

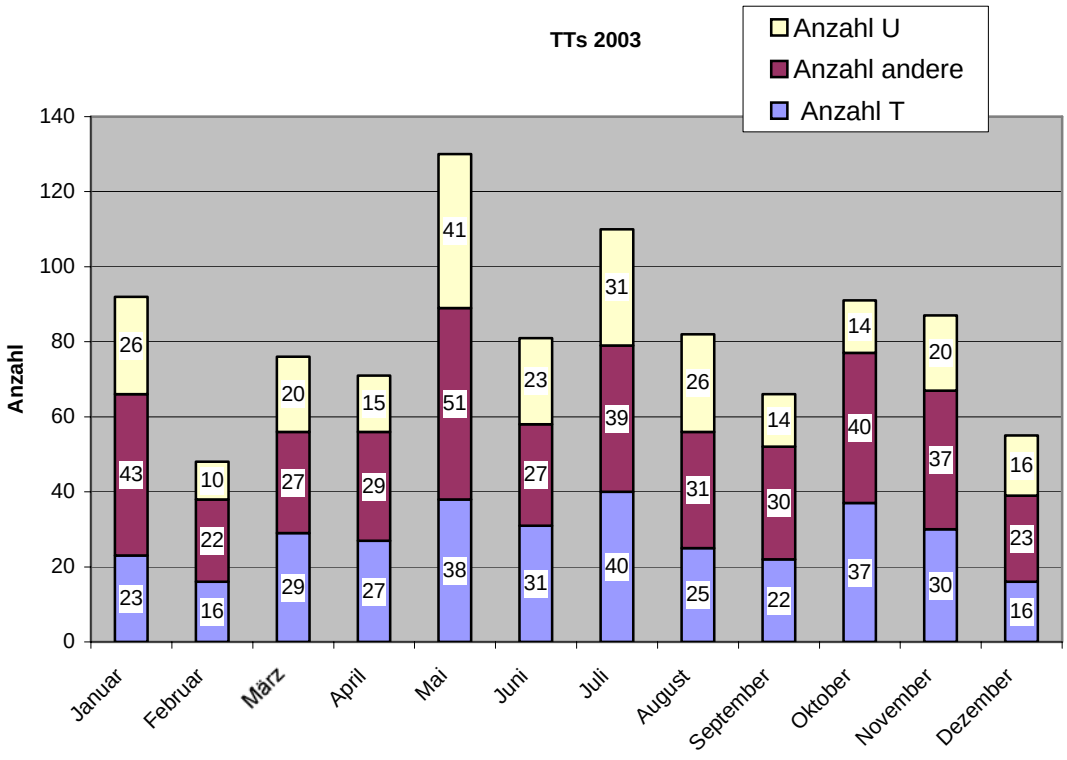



**Abbildung 9: Übersicht über die Anzahl der Trouble Tickets für die im G-WiN vorhandenen Zugangsleitungen im Jahr 2003, wobei „U“ die unbekannte Ursache und „T“ den Verursacher T-Systems bezeichnet.**

### 3.4.2.1 Kernnetzverbindungen

Die T-Systems stellt dem DFN-Verein täglich eine Fehler-Reportdatei über das Telekom-Portal Giovana (G-WiN Integrated Online Service Management) zur Verfügung. Darin werden alle PM- und FaultLog-Daten für das L1- und L2-Kernnetz aus dem Alcatel Management-System dargestellt.

*Erläuterungen:*

PM-Daten: Performance-Daten

FaultLog-Daten: aufgezeichnete UDT-Ereignisse (Up-Down-Traps)

L1: Kernnetzverbindungen der Stufe Level 1

L2: Kernnetzverbindungen Level 2 (doppelte Anbindung „hinter“ Level 1- Standorten)

ES: errored seconds

SES: severly errored seconds

BBE: background block errors

UAS: unavalaible seconds

*Beispiel PM-Daten (Performance-Daten)*

```
PM;403/11002_71102_63100;2004-01-22 03:00:00;2700;ok;0;0;0
PM;403/11002_71102_63100;2004-01-22 03:45:00;900;ok;2;0;2
PM;403/11002_71102_63100;2004-01-22 04:00:00;900;ok;1;0;1
PM;403/11002_71102_63100;2004-01-22 04:15:00;900;ok;0;0;0
PM;403/11002_71102_63100;2004-01-22 04:30:00;900;ok;1;0;1
PM;403/11002_71102_63100;2004-01-22 04:45:00;900;ok;0;834;16
PM;403/11002_71102_63100;2004-01-22 05:00:00;900;ok;0;834;1
PM;403/11002_71102_63100;2004-01-22 05:15:00;900;ok;0;834;0
PM;403/11002_71102_63100;2004-01-22 05:30:00;900;ok;0;834;0
PM;403/11002_71102_63100;2004-01-22 05:45:00;900;ok;0;834;0
PM;403/11002_71102_63100;2004-01-22 06:00:00;900;ok;0;192;0
PM;403/11002_71102_63100;2004-01-22 06:15:00;1800;ok;0;0;0
PM;403/11002_71102_63100;2004-01-22 06:45:00;900;ok;1;0;1
PM;403/11002_71102_63100;2004-01-22 07:00:00;900;ok;1;0;1
PM;403/11002_71102_63100;2004-01-22 07:15:00;900;ok;1;0;1
PM;403/11002_71102_63100;2004-01-22 07:30:00;4500;ok;0;0;0
```

Man findet nach der Kennzeichnung als PM-Datum die die Leitung betreffende Bezeichnung, das betreffende Datum, die diesem Eintrag zugrundeliegende Zeit, also beispielsweise die auf 5:00 Uhr folgenden 900s, das ok dafür, dass die Daten in Ordnung sind und danach die für diese Zeitspanne gefunden Fehler. Der erste Wert zeigt die gefundenen Fehlersekunden (ES – errored seconds), der zweite Wert die nicht verfügbaren Sekunden (UAS – unavailable seconds) und der dritte die für die Verfügbarkeitsberechnung nicht maßgeblichen Block Errors (BBE – background block errors).

*Beispiel Fault-Daten:*

```
Fault;403/11002_71102_63100;2004-01-22 04:46:02;2004-01-22 06:03:14;4632
```

In den Fault-Daten, die nur bei SDH-Verbindungen vorliegen, ist die genaue Zeitspanne des Ausfalls bzw. der Störung angegeben. In diesem Beispiel dauerte der Ausfall von 04:46:02 Uhr bis 06:03:14 Uhr, was einer Anzahl von 4632 Sekunden entspricht.

Das G-WiN Labor holt diese Fehler-Reportdateien regelmäßig ab, überprüft sie auf Vollständigkeit und wertet sie aufgrund der Vorgaben des G-WiN-Vertrages aus. Die Ergebnisse werden in die GIS-Datenbank eingetragen.

Dabei müssen SDH- und WDM-Verbindungen hinsichtlich der auftretenden Fehlerszenarien unterschieden werden, da die T-Systems für die WDM-Verbindungen keine FaultLog-Daten zur Verfügung stellen kann.

**Fall 1:** Die PM-Daten zeigen keine ES- und UAS-Fehler, aber es gibt ein FaultLog-

Ereignis -> der Fehler tritt am Endgerät auf und liegt damit im Verantwortungsbereich des DFN-Vereins.

**Fall 2:** Es gibt sowohl Fehler in den PM-Daten als auch entsprechende FaultLog-Ereignisse -> der Fehler tritt auf der Strecke auf und liegt damit im Verantwortungsbereich der Telekom.

**Fall 3:** Es gibt Fehler in den PM-Daten und kein FaultLog-Ereignis -> der Fehler liegt auf der Strecke, es ist kein Ausfall, aber ES und SES liegen im Verantwortungsbereich der Telekom.

#### 3.4.2.2 Jahresstatistikberechnungen des G-WiN-Kernnetzes

Für die Jahre 2002 und 2003 wurden Statistiken der Verfügbarkeiten und MTBF der Kernnetzleitungen, der Kernnetzknoten und des gesamten Kernnetzes berechnet. Zusätzlich zur Berechnung mit den vertraglich vereinbarten Schwellwerten für die ESs wurden Vergleichsrechnungen mit anderen Schwellwerten durchgeführt und der Geschäftsstelle zur Verfügung gestellt. Die folgende Tabelle ist ein Auszug aus den für die Kernnetzverbindungen für das Jahr 2003 berechneten jährlichen Verfügbarkeiten und MTBF (T: durch T-Systems verursachter Ausfall).

| Leitungsschlüsselzahl (LSZ) | Dauer T in s | Dauer Wartung in s | Dauer DFN in s | Anzahl Ausfälle T | Betriebstage | MTBF in h | VF in % |
|---|---|---|---|---|---|---|---|
| 403/801_69003_72470 | 0 | 4595 | 2162 | 0 | 180 | 4320.000 | 100.000 |
| 403/1_20100_22410 | 0 | 0 | 0 | 0 | 9 | 216.000 | 100.000 |
| 405/1209_69003_22410 | 0 | 0 | 0 | 0 | 9 | 216.000 | 100.000 |
| 401/2_89002_82100 | 0 | 0 | 126 | 0 | 365 | 8760.000 | 100.000 |
| 401/4_89002_82100 | 0 | 0 | 270 | 0 | 365 | 8760.000 | 100.000 |
| 401/11000_71102_72470 | 0 | 0 | 32 | 0 | 62 | 1488.000 | 100.000 |
| 401/11002_71102_72470 | 0 | 0 | 3 | 0 | 62 | 1488.000 | 100.000 |
| 401/11002_71102_62210 | 0 | 0 | 336 | 0 | 365 | 8760.000 | 100.000 |
| 403/500_22410_24100 | 0 | 6195 | 6048 | 0 | 365 | 8760.000 | 100.000 |
| 401/3_40004_43100 | 0 | 0 | 343 | 0 | 365 | 8760.000 | 100.000 |
| 403/7002_51100_53100 | 0 | 115 | 2186 | 0 | 365 | 8760.000 | 100.000 |
| 403/11000_71102_72470 | 619 | 406 | 3149 | 4 | 303 | 1454.366 | 99.998 |
| 401/11000_71102_62210 | 900 | 0 | 33269 | 1 | 365 | 4379.875 | 99.997 |
| 403/11002_71102_72470 | 411 | 4660 | 1859 | 1 | 123 | 1475.943 | 99.996 |
| 403/502_22410_24100 | 2793 | 778 | 11111 | 8 | 365 | 973.247 | 99.991 |
| 403/803_69003_64210 | 3313 | 6950 | 3438 | 2 | 365 | 2919.693 | 99.990 |
| 403/2001_40004_51100 | 3600 | 4500 | 0 | 2 | 365 | 2919.667 | 99.989 |

| Leitungsschlüsselzahl (LSZ) | Dauer T in s | Dauer Wartung in s | Dauer DFN in s | Anzahl Ausfälle T | Betriebstage | MTBF in h | VF in % |
|---|---|---|---|---|---|---|---|
| 403/1_91310_93100 | 3627 | 0 | 601 | 3 | 365 | 2189.748 | 99.989 |
| 403/3_91310_94100 | 7318 | 154 | 406 | 7 | 365 | 1094.746 | 99.977 |
| 403/4_34100_36770 | 8200 | 9314 | 569 | 4 | 365 | 1751.544 | 99.974 |
| 403/3_40004_44100 | 14087 | 2969 | 1335 | 7 | 365 | 1094.511 | 99.955 |
| 403/801_69003_51100 | 14400 | 7200 | 0 | 1 | 365 | 4378.000 | 99.954 |
| 403/701_69003_22410 | 18000 | 6300 | 0 | 2 | 357 | 2854.333 | 99.942 |

**Tabelle 1: Auszug aus den für die Kernnetzverbindungen für das Jahr 2003 berechneten jährlichen Verfügbarkeiten und MTBF.**

### 3.4.3 Erstellung eines Konzepts zur Auswertung der Informationen aus den Routern

#### 3.4.3.1 Zielsetzung

Für ein System zur Qualitätskontrolle der SDH-Plattform müssen unterschiedliche Informationsquellen herangezogen und ausgewertet werden. Die Daten müssen aufeinander abgebildet, ausgewertet und dargestellt werden.

#### 3.4.3.2 Informationsquellen

Als Informationsquellen stehen verschiedene Systeme zur Verfügung. Sie werden zum einen von unterschiedlichen Betriebseinheiten (DFN, DFN-NOC und T-Systems/Telekom) verwaltet und haben zum anderen verschiedene Sichtweisen auf das Netz.

- SDH/WDM-Management-System
- Interface-Logs von den Routern
- Reload-Meldungen der Router
- Sonet-MIBs aus den Routern

Als Beispiel wird hier die Auswertung der Sonet-MIBs dargestellt, mit denen eine Überprüfung der von der T-Systems zur Verfügung gestellten Managementdaten stattfand:

#### 3.4.3.3 Sonet-MIBs aus den Routern

- Allgemeines

Auf den Routern (*SNMP Agent*) stehen zu Managementzwecken verschiedene Datenbanken zur Verfügung. Aus diesen so genannten MIBs (*Managment Information Base*) können verschiedene Variablenwerte von einer Workstation (*SNMP Manager*) abgefragt oder an diese exportiert werden.

Die Router im G-WiN werden vom DFN-NOC verwaltet und konfiguriert. Für das Labor wurden vom NOC verschiedene MIB-Trees / Variablen zum Lesen für bestimmte Labor-PCs und Workstations freigegeben.

Unter Unix gibt es SNMP-Kommandos (**snmpget** oder **snmpwalk**), mit deren Hilfe Variablen oder ganze MIB-Trees abgefragt werden können.

Beispielabfragen:

**snmpwalk –m** *SONET-MIB.my* **–c** *... cr-erlangen1.g-win.dfn.de*

➔ für den gesamten SONET-MIB-Tree

oder

**snmpget** *cr-erlangen1.g-win.dfn.de ...*

*ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifAlias.36*

➔ für den Alias-Eintrag des Interfaces mit dem Index 36

Von den ARs und CRs können die folgenden Variablen ausgelesen werden:
Interface- und IP-relevante Variablen

**RFC1213-MIB:**

ifdescr

**IF-MIB:**

ifmibobjects

**IP-MIB:**

ipadentifindex

*ifdescr* enthält die Zuordnung Interface-Nummer <-> Interface-Typ.

*ipadentifindex* enthält die Zuordnung IP-Adresse <-> Interface-Nummer.

*ifmibobjects* enthält die Zuordnung Interface-Nummer <-> Leitungsbezeichnung.

Aus diesen Variablen lässt sich also eindeutig eine Abbildung der Leitungen zu den Interface der Router generieren.

Erläuterungen zu SONET:

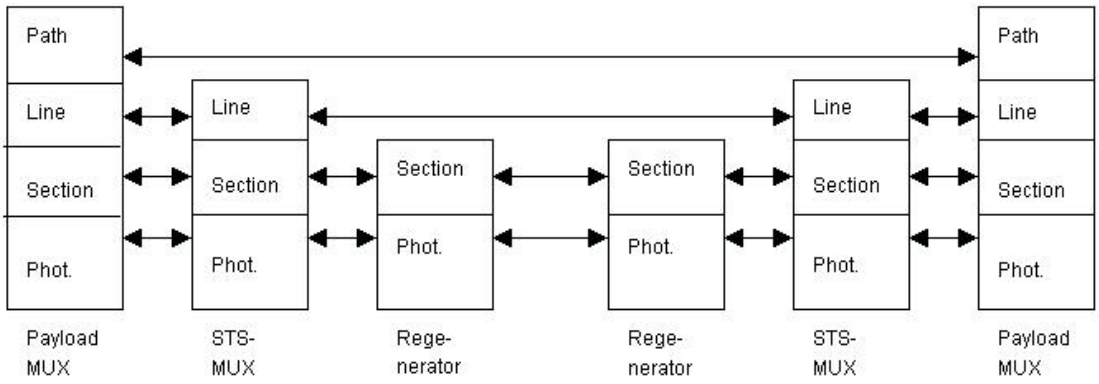



**Abbildung 10: SONET architecture layers [Sonet].**

Abbildung 10 zeigt die Hierarchie von Path, Line und Section. Man unterscheidet bei der SONET-Architektur vier Ebenen (Layer) .

Die unterste Ebene (*Photonic Layer*) bildet die physikalische/optische Ebene.

Die zweite Ebene (*Section Layer*) verweist auf die Regenerator-Section. Bei der optischen Übertragung muss das Signal regelmäßig alle 10 bis 100 km regeneriert werden. Jeder Abschnitt zwischen den Regeneratoren bildet eine Section.

Eine *Line* bildet den Abschnitt zwischen zwei SONET-Einheiten.

Der *Path* umfasst die end-to-end-Übertragung.

Anhand dieser Vorüberlegungen werden die im Abschnitt „Abfragen und Sammeln der SNMP-Variablen“ aufgeführten SNMP-Variablen gesammelt und ausgewertet.

- Erstellen einer Interface-Topologie

Um eine Zuordnung der am Router für verschiedene Interface abgefragten SNMP-Variablen zur entsprechenden Leitung zu bekommen, ist eine Interface-Topologie nötig. Diese Topologie gibt Auskunft darüber, welche Interface an welchen Routern die Endpunkte der Kernnetzleitung, für die mittels SNMP-Variablen eine Bestimmung der Dienstgüte durchgeführt werden soll, bilden.

Ein entsprechendes Script konstruiert mit Hilfe der Variablen ifdescr (RFC1213-MIB) und ifmibobjects (IF-MIB) und unter Verwendung von Router- und Routerinterface-Informationen einer internen Datenbank eine Interface-Topologie des G-WiN-Kernnetzes. Die Topologie wird alle sechs Stunden erneuert.

Somit hat jede im Kernnetz vorkommende Leitung zwei zugehörige Interface, für die im folgenden die SONET-MIBs viertelstündlich abgefragt werden.

- Abfragen und Sammeln der SNMP-Variablen

Um die SNMP-Daten möglichst lückenlos zu sammeln, und da die Router die Daten viertelstündlich aggregieren, wird alle Viertelstunde ein snmpget an die einzelnen Router gesendet. Die zur Berechnung nötigen Daten werden abgefragt:

snmpget router (i bezeichnet hier die Interface-Nummer am entsprechenden Router)

- sonetMediumTimeElapsed.i.1
- sonetFarEndLine
    - sonetFarEndLineIntervalValidData.i.1
    - sonetFarEndLineIntervalESs.i.1
    - sonetFarEndLineIntervalSESs.i.1
    - sonetFarEndLineIntervalCVs.i.1
    - sonetFarEndLineIntervalUASs.i.1
- sonetFarEndPath
    - sonetFarEndPathIntervalValidData.i.1
    - sonetFarEndPathIntervalESs.i.1
    - sonetFarEndPathIntervalSESs.i.1
    - sonetFarEndPathIntervalCVs.i.1
    - sonetFarEndPathIntervalUASs.i.1
- sonetLineInterval
    - sonetLineIntervalValidData.i.1
    - sonetLineIntervalESs.i.1
    - sonetLineIntervalSESs.i.1
    - sonetLineIntervalCVs.i.1
    - sonetLineIntervalUASs.i.1
- sonetPathInterval
    - sonetPathIntervalValidData.i.1
    - sonetPathIntervalESs.i.1
    - sonetPathIntervalSESs.i.1
    - sonetPathIntervalCVs.i.1
    - sonetPathIntervalUASs.i.1
- sonetSectionInterval
    - sonetSectionIntervalValidData.i.1
    - sonetSectionIntervalESs.i.1
    - sonetSectionIntervalSESs.i.1
    - sonetSectionIntervalSEFSs.i.1
    - sonetSectionIntervalCVs.i.1

Alle angegebenen Variablen beziehen sich auf das Zeitintervall einer Viertelstunde, wobei aber jedes Interface des Routers „eigene Viertelstundengrenzen" hat. Die Variable sonetMediumTimeElapsed gibt für jedes Interface die Zeit an, die seit dem Beginn des letzten Viertelstundenintervalls vergangen ist. Fragt man beispielsweise ein Interface um 14:00 Uhr ab, und die sonetMediumTimeElapsed beträgt im Betrachtungsintervall 300 s, dann ist das aktuelle Intervallende 14:00 Uhr – 300 s = 13:55 Uhr. Der Intervallbeginn war demnach um 14:00 Uhr – (300 s + 900 s) = 13:40 Uhr. Somit beziehen sich die abgefragten SONET-MIBs auf den Zeitraum 13:40 Uhr bis 13:55 Uhr.

Daraus ersichtlich ist die zeitliche Unschärfe, die der Auswertung der SNMP-Daten zugrunde liegt. Zum einen lassen sich die Ergebnisse der Auswertung nicht 1:1 auf die „echten" Zeitviertelstunden der T-Systems abbilden. Zum anderen ist der betrachtete Zeitraum nicht an beiden Interface einer Leitung der gleiche. Ein Problem ist auch, dass während der Abfrage der vielen Interface die Grenze von 900 s überschritten wird und damit verschiedene Intervallnummern abgefragt werden. Aus diesem Grund werden Interfaces, deren sonetMediumTimeElapsed 900 s abzüglich einem gewissen Timeoffset überschreitet, nach dem Timeoffset nochmals abgefragt. Momentan ist der Timeoffset auf 42 s festgelegt.

Die abgefragten SNMP-Daten werden in einer eigenen Datenbank mit Bezug zum entsprechenden Interface und dem Startzeitpunkt des Intervalls gesammelt, wenn mindestens eine der Variablen nicht Null ist.

Die Auswerteegeln werden regelmäßig anhand auftretender Ausfälle auf Vollständigkeit und Richtigkeit überprüft und gegebenenfalls ergänzt und korrigiert.

Die Verifikation der mit obigen Regeln ausgewerteten SNMP-Daten mit anderen Daten- und Informationsquellen ergab eine gute Übereinstimmung, abgesehen von der zeitlichen Unschärfe von einer bis maximal zwei Viertelstunden durch das bereits angesprochene „Viertelstundenproblem".

#### 3.4.3.4 Darstellung der SNMP-Daten

Wie im Abschnitt 3.4.3.3 beschrieben, werden die SNMP-Daten der Router den obigen Regeln entsprechend auf Viertelstundenbasis ausgewertet. Das heißt konkret, dass bei Überschreiten der vertraglich vereinbarten Schwellwerte die Viertelstunde als ausgefallen gewertet wird. Anhand der Daten wird ebenso ein Verursacher festgelegt. Wie bei den PM-Daten der T-Systems auch werden Fehler am Rande des Netzes zu „Fehlern des DFN-Vereins" ausgewertet, Fehler auf der Strecke zu „Fehlern der T-Systems". Es treten aber auch Fehlerszenarien auf, die nicht eindeutig einem der beiden Verursacher zuzuordnen sind. Ebenso häufig kommt es auch an „Rändern" größerer Ausfälle zu verfälschten Ergebnissen.

In die Auswertung der SNMP-Daten fließen auch angekündigte Wartungen mit ein. Ausgefallene Viertelstunden („**U**"), zu deren Zeit ein genehmigte bzw. den vertraglichen Regeln entsprechende Wartung angekündigt wurde, werden nicht als echter Ausfall sondern als Wartungsausfall („**W**") gewertet.

#### 3.4.3.5 Vergleich der SNMP-Daten mit den Management-Daten der T-Systems

Exakt die gleiche Darstellung der Fehler und deren Verursacher auf Viertelstundenbasis wird als Folge der Auswertung der PM-Daten der T-Systems erstellt (s. auch Abschnitt 3.4.2.1):

```
*** 403/11002_71102_63100
    Leitungsinformationen (verfueg.pl):
    oid: 618  text: Stuttgart-Kaiserslautern
    avail: 97.618  failcount: 1
    Leitungsinformationen (auswert.pl):
    von: POS5/0 ar-kaiserslautern1 (Interface-ID: 427)
    nach: POS4/1 cr-stuttgart1 (Interface-ID: 1327)
    Auswertungsergebnis (verfueg.pl):
    ..................WWWUUU..................... 12:00
    ..................................................... 24:00
    Auswertungsergebnis (auswert.pl):
    ..................DDWDUU..................... 12:00
    ..................................................... 24:00
```

Obiges Beispiel zeigt die Unterschiede zwischen beiden Auswertungen. Diese können darauf zurückzuführen sein, dass die Zeitproblematik der Interface dazu geführt hat, dass zusammengehörige Fehler an beiden Enden einer Verbindung in verschiedene Viertelstunden „gerutscht“ sind. Die Auswerteregeln interpretieren diesen Fehler dann als „nur an einem Ende der Verbindung vorkommend“, was auf ein Problem am Rande des Netzes hinweist, also auf ein Problem am Router. Da nur die von der Telekom mit Wartung gekennzeichneten und dann auch mit Verursacher Telekom ausgefallenen Viertelstunden als Wartung gelten, findet man in diesem Fall in der Auswertung der snmp-Daten ein „**D**“ und kein „**W**“, da es nicht als Telekom-Fehler erkannt wurde.

#### 3.4.3.6 Darstellung der Ergebnisse im Web

Zur Veranschaulichung der Ergebnisse gibt es eine Webdarstellung:

http://winner.rrze.uni-erlangen.de/cgi-bin/Alex/kernnetz/verbindungen.pl

Wie im folgenden Beispiel gezeigt, sind dort die Kernnetzleitungen in einer Kreisdarstellung erfasst. Im inneren Kreis befinden sich die Level 1-Knoten, auf dem äußeren die Level 2-Knoten. Entsprechend der im G-WiN vorhandenen Kernnetzleitungen existieren Verbindungen zwischen den Knoten.

Es gibt zwei verschiedene Kreise, einen für die Darstellung der PM-Daten, einen für die Darstellung der SNMP-Daten.

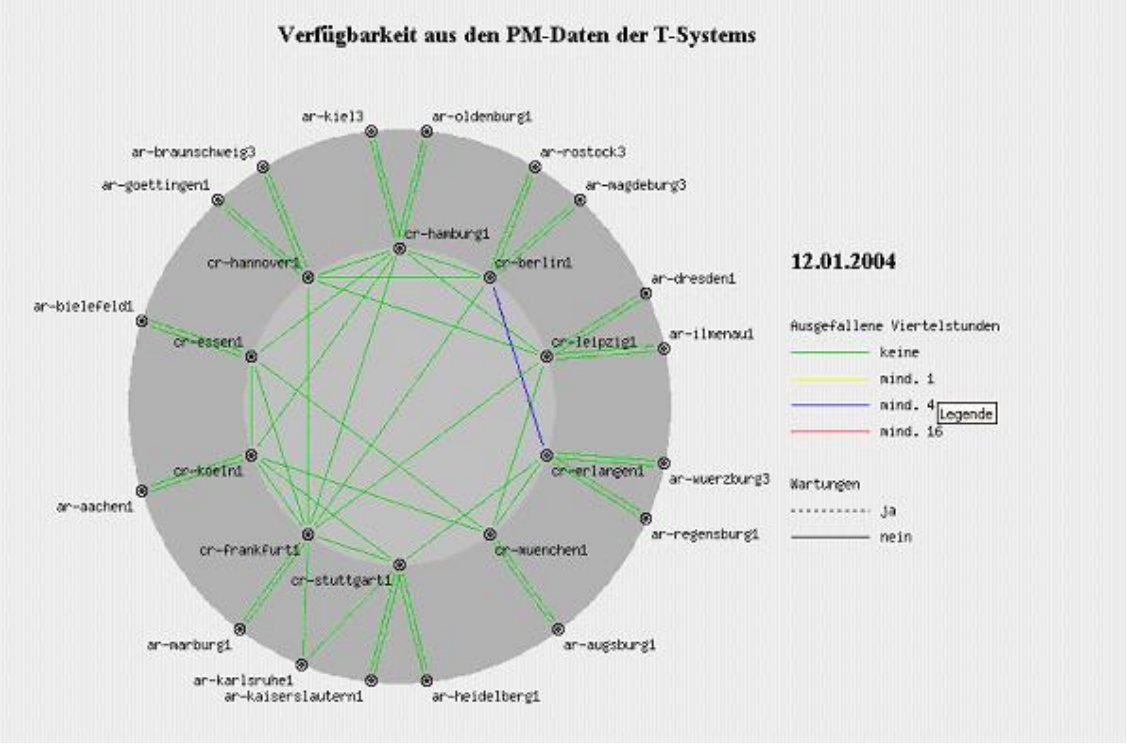



**Abbildung 11: PM-Daten-Darstellung am 12.01.2004.**

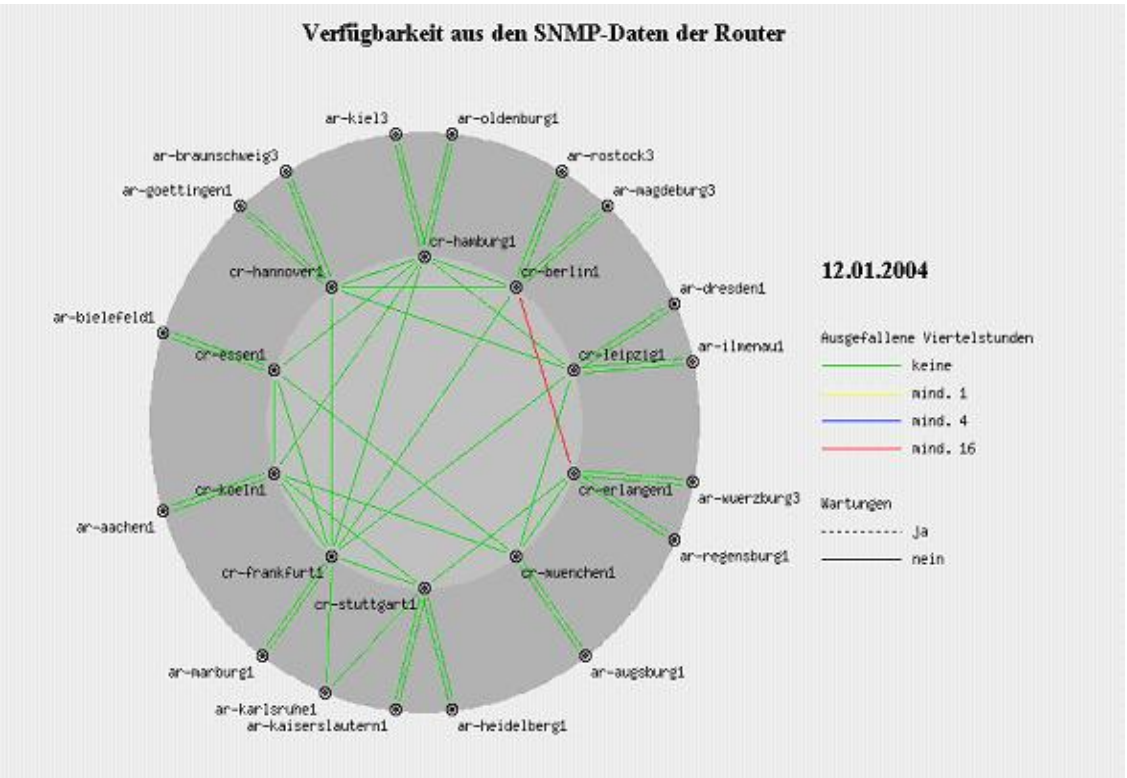



**Abbildung 12: SNMP-Daten-Darstellung am 12.01.2004.**

Die folgende Legende erklärt die Darstellung. Grüne Linien bedeuten, dass es auf diesen Leitungen keine Ausfälle gab. Eine gelbe Linie heißt, dass mindestens eine Viertelstunde des

Tages ausgefallen ist, eine blaue Linie, dass mindestens vier Viertelstunden ausgefallen sind. Die rote Linie bestimmt die Grenze, ab der eine Verbindung laut Vertrag einen ganzen Tag als ausgefallen gilt (ab 16 ausgefallenen Viertelstunden).


**Abbildung 13: Legende**

Sind die Linien gestrichelt gab es für den Tag auf der entsprechenden Verbindung angekündigte (und genehmigte) Wartungen.

Innerhalb dieser Kreisdarstellung wird nicht zwischen den verschiedenen Verursachern der Ausfälle unterschieden.

Zusätzlich zur schematischen Darstellung aller Kernnetzleitungen ist es möglich, die genauen Viertelstundenwerte von einzelnen Verbindungen zu erfragen.

Ein Klick auf die gewünschte Leitung in der Kreisdarstellung der PM-Daten bringt die Auswertung der PM-Daten auf Viertelstundenbasis inklusive ermittelter Verursacher.


**Abbildung 14: PM-Daten auf Viertelstundenbasis**

Ein Klick auf die gewünschte Leitung in der Kreisdarstellung der SNMP-Daten bringt die Auswertung der SNMP-Daten auf Viertelstundenbasis inklusive ermittelter Verursacher.


**Abbildung 15: SNMP-Daten auf Viertelstundenbasis**

#### 3.4.3.6.1 Ausblick

- Die Auswertung der Informationen aus den Routern und der Vergleich der Ergebnisse mit den Ergebnissen der ausgewerteten PM-Daten der T-Systems befindet sich momentan noch in einer Testphase. Zum weiteren Test müssen verschiedene Fehlerszenarien überprüft, muss den Ursachen dieser Fehler nachgeforscht und müssen davon abhängig Bewertungen dieser Fehler gegebenenfalls korrigiert werden.
- Das gesamte Programmsystem sowohl zur Auswertung der PM-Daten als auch zum Sammeln und Auswerten der SNMP-Daten soll mit Hilfe einer webbasierten Bedienoberfläche transparenter und einfacher bedienbar gemacht werden.
- Als weitere Automatisierung ist eine Generierung von Online-Statistiken für die monatliche und jährliche Verfügbarkeit der Kernnetzverbindungen, der Kernnetzknoten und des gesamten Kernnetzes gedacht.
- Für die genaue Feststellung der Fehlerursache wird es nötig, zur Auswertung zwischen SDH- und WDM-Verbindungen zu unterscheiden. Nur dann kann den verschiedenen Fehlerszenarien Rechnung getragen werden.

## 3.5 IP-Verkehrsflussmessungen

### 3.5.1 Realisierung der Verarbeitung der Accountingdaten

Bereits seit Anfang 2002 sammelt die Accounting-Workstation in Erlangen die NetFlow-Daten aller Router im G-WiN und speichert sie in voraggregierter Form in ARTS-Dateien [ARTS] ab [Acc02]. Die Daten liegen in der Form von Tabellen vor, welche einen Eintrag von Zahl der Bytes und IP-Pakete für jede Kombination von IP-Quellnetz und IP-Zielnetz haben. Pro Router wird alle 5 Minuten eine solche Datei erzeugt. Diese Daten mussten nun weiter zusammengefasst werden. Ziel war eine Zusammenfassung auf die Ebene von Anschlüssen des G-WiN. Bei Einrichtungen mit mehrfachem Anschluss (z.B.T-InterConnect) sollte die Zusammenfassung auf einen virtuellen Knoten (z.B. DTAG) erfolgen.

Zur Datenerfassung, Aggregierung und Auswertung steht eine SUN Enterprise 6500 (16 Prozessoren, 10 GByte RAM, 200 GByte HW-Raid5) zur Verfügung, die auch die Webdarstellung der Daten übernimmt. Ein Backup-Server wird derzeit aufgebaut.

Für die Realisierung dieser Aggregation sind die folgenden Schritte nötig:

### 3.5.2 Bestimmung der aktuellen Routerkonfiguration

Um die Konfiguration der Accounting-Workstation aktuell zu halten, wurde ein Programm (router2gas.pl) geschrieben, welches aus dem GIS die Informationen über die Router im G-WiN liest und in die Datenbank des G-WiN Labor (G-WiN Accounting System, GAS) auf der Erlanger Accounting-Workstation einträgt. Hierbei werden die Hostnamen der Router ausgelesen. Über SNMP werden die Interfacenummern und -bezeichnungen aller Router abgefragt und ebenfalls in die GAS eingetragen. Diese Einträge werden auch für das SDH-QoS verwendet (vgl. Kapitel 3.4.3.3). Die in der GAS gespeicherten Informationen über die Topologie des G-WiN werden benutzt, um für jeden Accessrouter (AR) den zugehörigen Corerouter (CR) zu bestimmen. Danach richtet sich der UDP-Port, der auf der Accountingworkstation für den Empfang der NetFlow-Daten konfiguriert werden muss.

Per SNMP wird dann aus den Routern ausgelesen, ob es sich um Cisco Router der 7500er oder 12000er Reihe handelt. Dies entscheidet, ob die NetFlow-Daten mit oder ohne Sampling gespeichert sind.

Für das IP-Accounting dürfen nur die NetFlow-Daten der Interface ausgewertet werden, an denen Anwender oder Peerings angeschlossen sind. Für Analysezwecke aktiviert das NOC allerdings zeitweise auch an anderen Interface das NetFlow-Accounting. Daher wird für jedes Interface im GIS der zugehörige Anwender-Übergabepunkt gesucht. Fehlt dieser, handelt es sich um ein kernnetzinternes Interface und dieses Interface wird nicht in die Konfiguration der NetFlow-Datensammlung aufgenommen.

### 3.5.3 Bestimmung der IP-Adressbereiche der Anwender

Die IP-Adressbereiche aller G-WiN-Anwender sind im GIS gespeichert. Es wurde ein Programm (net2auoid.pl) entwickelt, welches täglich diese Einträge ausliest, zusammen mit den dazugehörigen Anschluss-Übergabepunkten.

### 3.5.4 Bestimmung der Anschlüsse von Peerings

Aus dem GIS wird für jedes Peering-AS der entsprechende Anwender-Übergabepunkt bestimmt. Die Daten aus 3.5.2 und 3.5.3 werden dann auf die Accounting-Workstation kopiert und dort historisiert gespeichert. Dies wird täglich automatisch durchgeführt.

### 3.5.5 Auswertung der BGP-Tabellen

Die IP-Adressbereiche, die nicht zu G-WiN-Anwender gehören, werden mittels der BGP-Tabellen der G-WiN-Corerouter den Peering-ASen zugeordnet, über welche sie geroutet werden. Mit Hilfe der AS-Anwenderübergabepunkt-Daten aus 3.5.4 lässt sich dann der entsprechende Anwenderübergabepunkt finden.

Die BGP-Tabellen holt die Accounting-Workstation täglich von einem Rechner des NOC ab. Ein vom G-WiN Labor entwickeltes Programm (bgp2hash.pl) entnimmt jeder BGP-Tabelle die Information, über welches AS der fragliche Adressbereich geroutet wird. Über die Daten aus 3.5.4 lässt ich dann der Anschluss bestimmen.

Eine Ausnahme bilden die Peerings am ir-frankfurt2. Über diesen Router ist das G-WiN mit sehr vielen Peering-Partnern verbunden, deren IP-Adressbereiche und ASe sich häufig ändern, u.a. dem DE-CIX, GEANT und Tiscali. Das NOC sieht sich daher nicht in der Lage, das GIS in dieser Beziehung aktuell zu halten. Daher werden alle IP-Adressbereiche, die auf dem CR den ir-frankfurt2 als NextHop-Router haben, nicht über das AS sondern über die BGP-Tabelle des ir-frankfurt2 zugeordnet. Aus dieser BGP-Tabelle wird der NextHop aus Sicht des ir-frankfurt2 bestimmt. Da es sich hierbei nur um 11 verschiedene ISPs handelt, können diese fest konfiguriert auf Anschlüsse umgesetzt werden.

Um bei den Peerings mit mehr als einem Übergabepunkt ins G-WiN (z.B. DTAG) unterscheiden zu können, über welchen der Anschlüsse ein Flow geroutet wurde, muss ebenfalls die BGP-Tabelle herangezogen werden. Über die IP-Adresse des NextHop-Routers kann der Exit-Router bestimmt werden.

### 3.5.6 Aggregation der ARTS-Dateien

Die ARTS-Dateien (Dateien im Standardformat für Daten des cflowd-Systems) können nun anhand der obigen Zuordnungstabellen bzw. Datenbankeinträge weiter aggregiert werden. Das Aggregationsprogramm (acc.pl) liest folgende Daten ein:

- IP-Netz -> Anwenderübergabepunkt
- Peering-AS -> Anwenderübergabepunkt
- Router-IP und Interface-ID der NetFlow-exportierenden Interface

- Samplingraten der RouterInterface

Um die 16 Prozessoren der Accountingworkstation ausnutzen zu können, wurde das Aggregationsprogramm parallelisiert. Nach dem Einlesen der Konfiguration wird eine konfigurierbare Anzahl Child-Prozesse gestartet, welche die eigentliche Aggregation durchführen. Derzeit werden 10 Childs verwendet. So bleiben 6 Prozessoren für andere Aufgaben übrig, wie z.B. das Entgegennehmen, Voraggregieren und Abspeichern der NetFlow-Datenströme. 10 Childs erwiesen sich im Test als ein guter Wert, da hiermit stets die zuletzt vom cflowd übermittelten Accountinginformationen rechtzeitig weiterverarbeitet werden konnten, bevor die nächsten Daten eingetroffen sind. Eine weitere Erhöhung der Zahl Childs bringt keinen weiteren Geschwindigkeitsvorteil, weil dann mehr Prozesse gleichzeitig auf die Festplatten zugreifen und sich dabei gegenseitig behindern.

Die Child-Prozesse bekommen von dem Parent-Prozeß automatisch je 1/10 der ARTS-Dateien zur Aggregation zugeteilt. Die von den Childs erzeugten Summentabellen werden nach Ende aller Berechnungen vom Parent-Prozeß wiederum zu einer Tabelle zusammengefasst und abgespeichert.

Derzeit wird eine solche Tabelle jeweils für 24 h erzeugt. Diese Tagestabellen können dann nach verschiedenen Gesichtspunkten ausgewertet und dargestellt werden. Neben der vollen Matrix über die Anwender-Übergabepunkte lassen sich beispielsweise Quellsummen, Zielsummen u.a. realisieren. Auch eine weitere zeitliche Summierung z.B. für Wochensummen ist einfach machbar.

Die volle Tabelle umfasst etwa 70.000 Zeilen bzw. Verkehrsbeziehungen.

### 3.5.7 Darstellung der Ergebnisse

Um rasch eine erste Nutzung der erzielten Daten zu ermöglichen, wurde ein DFN-interner Zugang via HTTP geschaffen. Dort können Tages- und Wochensummen sowohl für die volle Anwenderübergabepunkt-Matrix als auch die Quell- und Zielsummen abgerufen werden.

Ein Beispiel für die Top-10 der Anwenderübergabe-Verbindungen, mit dem größten Gesamtvolumen am G-WiN Verkehr, ist folgender Abbildung zu entnehmen:

2004/06/16

Total: 4.515404e+13 Bytes
Unknown: 6.618064e+11 Bytes ( 1.5 %)
Internal: 4.101192e+11 Bytes ( 0.9 %)
188.1.0.1: 1.702695e+09 Bytes ( 0.0 %)
0.0.0.0: 3.313352e+07 Bytes ( 0.0 %)

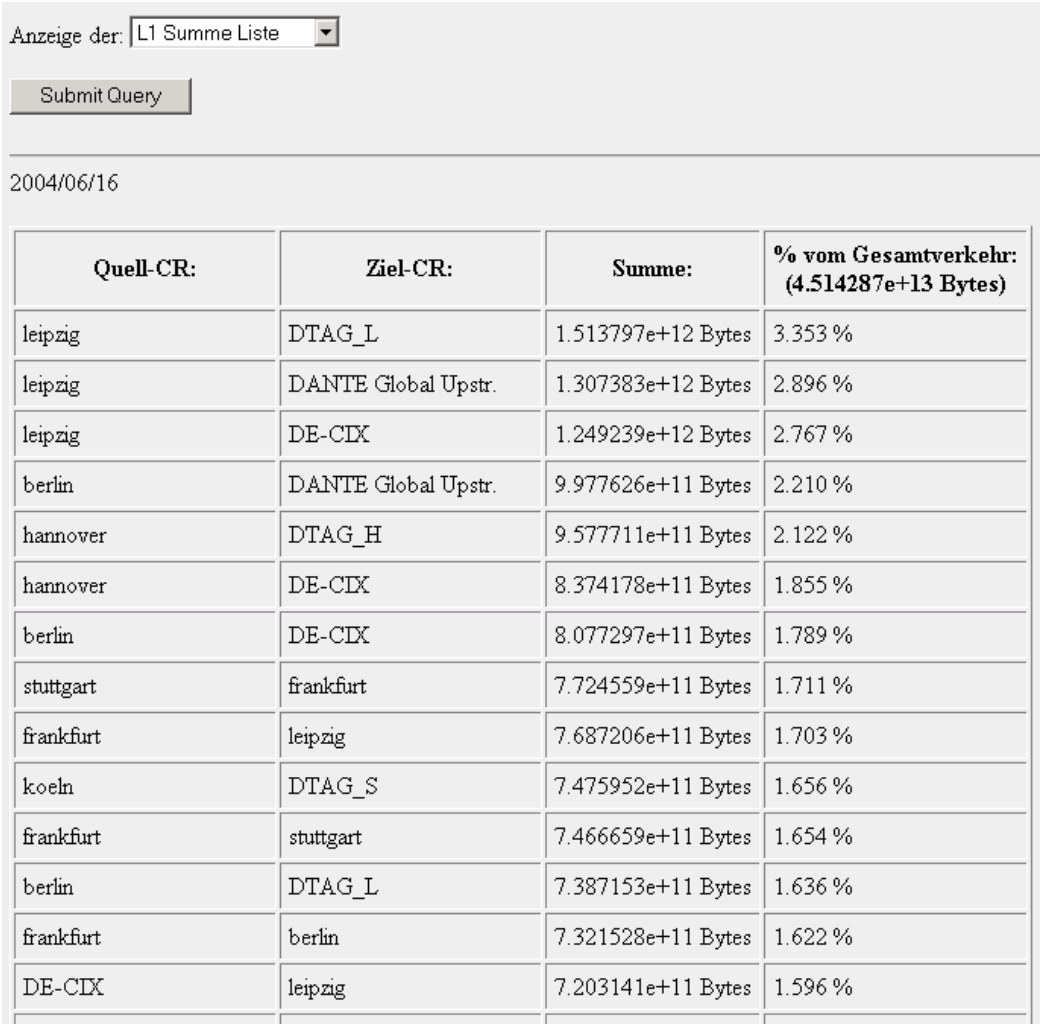
| SRC: | | DST: | | | |
|---|---|---|---|---|---|
| AU-OID: | Name: | AU-OID: | Name: | Bytes: | |
| 759 | GWD Goettingen | 1105 | DTAG (Hannover) | 6.474304e+11 | ( 1.434 %) |
| 759 | GWD Goettingen | 967 | DE-CIX | 5.596535e+11 | ( 1.239 %) |
| 717 | TU Dresden | 1104 | DTAG (Leipzig) | 5.295143e+11 | ( 1.173 %) |
| 717 | TU Dresden | 967 | DE-CIX | 4.879890e+11 | ( 1.081 %) |
| 1075 | RWTH Aachen | 1232 | DTAG (Stuttgart) | 4.655302e+11 | ( 1.031 %) |
| 1075 | RWTH Aachen | 967 | DE-CIX | 3.815127e+11 | ( 0.845 %) |
| 759 | GWD Goettingen | 1223 | Telia International Carrier GmbH | 3.632628e+11 | ( 0.804 %) |
| 1260 | Forschungszentrum Karlsruhe GRID | 4731 | GEANT (DANTE) | 3.557942e+11 | ( 0.788 %) |
| 4731 | GEANT (DANTE) | 1328 | Uni Karlsruhe | 3.500380e+11 | ( 0.775 %) |
| 1075 | RWTH Aachen | 1115 | DANTE (Global Upstream) | 3.375447e+11 | ( 0.748 %) |

**Abbildung 16: Top-10 Liste der Anwenderübergabepunkt-Verbindungen vom 16.6.2004**

*Total* entspricht der Gesamtsumme des Verkehrs, der über das G-WiN gesendet wurde. *Unknown* ist der Verkehr, der keinem Ziel aus der GIS-Datenbank oder der BGP-Tabelle zugeordnet werden konnte (z.B. unvollständige oder nicht aktuelle Einträge). Bei *Internal* handelt es sich um Datenflüsse, deren Ziel sich im G-WiN befindet und somit kein Anwenderübergabepunkt ist. Die beiden Adressen *188.1.0.1* und *0.0.0.0* sind Adressen, die vom DFN-NOC verwendet werden, um bestimmten Verkehr zu verwerfen.

### 3.5.8 Erstellen der Level1-Matrix

Zur Bestimmung einer geeigneten Netz-Topologie fürs G-WiN, ist es wichtig die Level1 zu Level1 Verkehrsbeziehungen zu kennen. Aus diesem Grund wurde hierfür vom G-WiN Labor ein Programm entwickelt, das anhand der GAS-Topologie feststellt, welcher Anwender-übergabepunkt zu welchem Level1-CR gehört. Mit Hilfe der Verkehrsmatrix für die Anwenderübergabepunkte kann daraus die Level1-Verkehrsmatrix berechnet werden.




| Quell-CR: | Ziel-CR: | Summe: | % vom Gesamtverkehr: (4.514287e+13 Bytes) |
|---|---|---|---|
| leipzig | DTAG_L | 1.513797e+12 Bytes | 3.353 % |
| leipzig | DANTE Global Upstr. | 1.307383e+12 Bytes | 2.896 % |
| leipzig | DE-CIX | 1.249239e+12 Bytes | 2.767 % |
| berlin | DANTE Global Upstr. | 9.977626e+11 Bytes | 2.210 % |
| hannover | DTAG_H | 9.577711e+11 Bytes | 2.122 % |
| hannover | DE-CIX | 8.374178e+11 Bytes | 1.855 % |
| berlin | DE-CIX | 8.077297e+11 Bytes | 1.789 % |
| stuttgart | frankfurt | 7.724559e+11 Bytes | 1.711 % |
| frankfurt | leipzig | 7.687206e+11 Bytes | 1.703 % |
| koeln | DTAG_S | 7.475952e+11 Bytes | 1.656 % |
| frankfurt | stuttgart | 7.466659e+11 Bytes | 1.654 % |
| berlin | DTAG_L | 7.387153e+11 Bytes | 1.636 % |
| frankfurt | berlin | 7.321528e+11 Bytes | 1.622 % |
| DE-CIX | leipzig | 7.203141e+11 Bytes | 1.596 % |

**Abbildung 17: Auszug aus Level1-Matrix**

Da die Verkehrsmatrix stark von den Standorten der Peering-Aufpunkte (wie DTAG, DE-CIX usw.) abhängt, werden solche Anschlüsse zusätzlich zu den normalen Level1 Standorten als sogenannte Pseudoknoten in die Level1-Matrix mit aufgenommen. Würde sich ein Peering-Aufpunkt von A nach B verschieben, muss mit diesem Konzept keine neue Level1-Matrix erstellt werden, sondern kann direkt berücksichtigt werden. Abbildung 17 zeigt einen Auszug der berechneten Level1-Matrix vom 16.6.2004. Dabei sind von oben nach unten die stärksten Verkehrsbeziehungen aufgelistet.

### 3.5.9 Analyse der Clusteranschlüsse im G-WiN

Zusätzlich zu den bisher geplanten Anforderungen ans IP-Accounting im G-WiN wurde von der DFN Geschäftsstelle an das G-WiN Labor der Wunsch herangetragen, die Clusteranschlüsse im G-WiN detaillierter zu analysieren.

Da bei einem Clusteranschluss mehrere Anwender an einem Anwenderübergabepunkt angeschlossen sind, kann man mit dem normalen IP-Accounting nur die Summe des Traffics aller Anwender an diesem Anschluss ermitteln. Gleiches gilt für die vom DFN-NOC ermittelten Portcounter. Da die einzelnen Clusterteilnehmer aber jeweils eigene Verträge mit

individuellen Verkehrsgrenzen mit dem DFN haben, ist es wichtig, die Verkehrsflüsse der Clusterteilnehmer einzeln zu ermitteln.

Es wurde daher im G-WiN Labor ein Programm ähnlich zum normalen Analyseprogramm entwickelt, welches speziell die Clusteranschlüsse unter die Lupe nimmt. Die Daten über die IP-Subnetzbereiche der Clusteranschlüsse werden aus dem GIS importiert. Es werden wie unter 3.5.6 erläutert Quell- und Ziel-Tabellen ermittelt, aber anstelle der Anwenderübergabepunkte werden die einzelnen IP-Subnetze der Clusterteilnehmer betrachtet.

In Abbildung 18 ist die Aufteilung des Verkehrs für die Cluster FZ Jülich, GKSS FZ Geesthacht GmbH und GWD Göttingen dargestellt.

2004/06/16

**Domains:**

| Domain: | SRC Bytes: | DST Bytes: |
|---|---|---|
| | 2.058265e+11 | 1.146705e+11 |
| FZ Juelich GmbH (schule) | 1.315586e+08 | 5.155381e+07 |
| FZ Juelich GmbH (tz-juelich) | 7.427191e+08 | 1.570197e+09 |
| GKSS FZ Geesthacht GmbH (epc.izm.fraunhofer) | 7.534770e+07 | 1.094357e+08 |
| GKSS FZ Geesthacht GmbH (teltow.gkss) | 1.384647e+10 | 1.611507e+09 |
| GWD Goettingen (brzn) | 1.428416e+10 | 5.904428e+09 |
| GWD Goettingen (gwdg) | 2.445570e+12 | 5.120926e+11 |
| GWD Goettingen (iwf) | 2.259153e+09 | 2.992264e+09 |

**Abbildung 18: Auszug aus den Clusteranschlüssen**

### 3.5.10 Analyse der Mitnutzer im G-WiN

Eine ähnliche Analyse wie für die Clusteranschlüsse im G-WiN wurde auf Wunsch der DFN-Geschäftsstelle auch für die Mitnutzer realisiert.

### 3.5.11 Backupsystem für Accounting-WS

Da im Falle eines Absturzes der vorhandenen Accounting-WS in Erlangen (Sun Enterprise 6500) oder der Isolierung dieser WS vom G-WiN, bedingt durch den Ausfall der Kernnnetzverbindung nach Erlangen, Accountingdaten verloren gehen können, wird ein Backupsystem installiert. Um den Fall der Isolierung auszuschließen, wird diese WS an einem anderen Ort im G-WiN (Stuttgart oder Berlin) aufgestellt werden. Sie soll zusätzlich zu der vorhandenen Accounting-WS alle von den Routern zur Verfügung stehenden Daten sammeln und für einige Zeit zur Sicherheit zwischenspeichern.

## 3.6 Tests betriebsrelevanter Komponenten

Abnahmen und Tests von Komponenten, die entweder in Betrieb gehen oder im Betrieb eingesetzt werden, sind erforderlich, wenn damit die Qualität der bisher angebotenen Dienste verbessert werden kann oder neue, innovativere Mechanismen, die in Netzwerkkomponenten implementiert sind, vor Inbetriebnahme unter Laborbedingungen auf ihre grundsätzlichen funktionalen Eigenschaften untersucht werden sollen. Im abgelaufenen Projekt wurden die folgenden Cisco-Linecards primär auf Funktionalität, Performance und Class-of-Service (CoS) Eigenschaften hin untersucht. Die Tests erfolgten in enger Zusammenarbeit mit dem Hersteller und dem DFN-NOC. In Klammern ist der jeweilige Testbericht angegeben, indem die durchgeführten Test ausführlich beschrieben sind:

- 4OC3X/POS-LR-LC-B Engine 3 [Test4OC3]
- Cisco 12000 Series Four-Port GE ISE Line Card, with Multimode Gbics (GLC-SX-MM) [TetraGE04]
- 2 fach E3 Port Adapter (PA) auf VIP2-50 mit RSP4 [Flat02]
- 6 fach E3 Linecards (LCs), Engine0 [Flat02]
- 4 fach OC3 Linecards, Engine0 [Flat02]

Für die Untersuchungen steht im G-WiN Labor eine geeignete Testumgebung zur Verfügung. Sie besteht aus

- zwei Cisco 12416,
- ein Cisco 12410 (Leihstellung vor Inbetriebnahme im G-WiN)
- zwei Cisco 7507,
- zwei Cisco GSR 12008-Routern,

die mit entsprechenden Interfacekarten ausgestattet sind, sowie verschiedenen Verkehrs-generatoren

- Agilent RouterTester (OC3/OC12 POS, OC48 POS, Fast Ethernet, Gigabit Ethernet)
- Spirent Smartbits (OC3/OC12 POS),

Die Testergebnisse zu den oben durchgeführten Cisco-Linecard Untersuchungen sind in Form von Testberichten zusammengefasst und dem Abschlussbericht beigelegt.

## 3.7 Begleitende Aktivitäten

Zur Unterstützung der unter 2.3 bis 2.6 aufgeführten Facharbeit bedarf es einiger begleitender Aktivitäten. Es handelt sich dabei um

- Öffentlichkeitsarbeit (Berichtswesen, Veröffentlichungen, Vorträge, Web-Präsenz)
- Administration Server und Arbeitsplatzrechner (Solaris, Linux, Windows)
- Administration Testumgebungen
- Verwaltung, Controlling

- „händische“ Tätigkeiten (Versand, Dokumentenbearbeitung)

Sie wurden auf alle beteiligten MitarbeiterInnen aufgeteilt.

### 3.7.1 Veröffentlichungen des WiN Labors

- R. Kleineisel, I. Heller, S. Naegele-Jackson, *Messung von Echtzeitverhalten im G-WiN*, Verteilte Echtzeitsysteme (PEARL 2003) der GI-FG.4.4.2 Echtzeitprogrammierung, Boppard, 27./28.11.2003.
- G-WiN Labor, *Qualitätsmessungen im Deutschen Forschungsnetz*, DFN-Mitteilungen Heft 64, März 2004

### 3.7.2 Vorträge des WiN Labors

Das G-WiN Labor hat Vorträge auf den folgenden Veranstaltungen gehalten:

- DFN Betriebstagungen
- RRZE-Kolloquien und Ausbildung
- Spring 2004 Internet2 Member Meeting: *The DFN IPPM measurement system*, 19. April 2004, Arlington USA
- Transatlantic Performance Monitoring Workshop 2004, *The DFN IPPM measurement system*, 17. März 2004, Genf
- Aktive Teilnahme an mehreren Projektbesprechungen für das zukünftige europäische Geant2-JRA1 Projekt in Amsterdam und Erlangen

### 3.7.3 Teilnahme an Seminaren

Die MitarbeiterInnen des Labors nahmen an den folgenden Veranstaltungen teil:

- Cisco-LAB in London, April 2003, Funktionstests der NSF-SSO Features und redundante PRPs

### 3.7.4 Web-Server

Die vom Labor zur Verfügung gestellten Informationen wurden auf dem Web-Server des Labors für die DFN-Geschäftsstelle und die Anwender dargestellt (www.win-labor.dfn.de).

Außerdem wurden auf dem Web-Server die Folien von den Betriebstagungen interessierten Anwendern bereit gestellt. Auch Testberichte wurden auf dem Web-Server eingebunden, sofern die betroffenen Firmen der Veröffentlichung zugestimmt haben.

### 3.7.5 Telefonkonferenzen

Zwischen der DFN-GS Berlin, dem DFN-NOC und dem G-WiN Labor wurden regelmäßig Telefonkonferenzen über die DFN-Gatekeeper im G-WiN durchgeführt (PC mit NetMeeting oder Telefon). Diese Konferenzen tragen dazu bei, den Informationsfluss zwischen den beteiligten Einrichtungen zu verbessern.

### 3.7.6 Nutzerberatung

Das G-WiN Labor wurde des öfteren von Anwendern bezüglich der Darstellung der Messdaten im Web sowie zu deren Aufbereitung befragt, insbesondere im Bereich der IPPM-Messungen. Einige Anwender äußerten auch speziell den Wunsch, One-Way-Delay Messungen bei ihnen durchzuführen, um z.B. spezielle Probleme in ihren lokalen Netzen aufzudecken. Soweit es dem Labor möglich war, versuchte es, die Wünschen der Anwender zu erfüllen.

### 3.7.7 Personalia

Im G-WiN Labor waren im Laufe des Projekts folgende Mitarbeiterinnen und Mitarbeiter ganz- bzw. halbtags beschäftigt. Die meisten MitarbeiterInnen waren hierbei nicht für die gesamte Projektlaufzeit tätig, sondern nur für einen bestimmten Zeitraum (Mitarbeit im Routertestlabor, Wechsel zu anderen Firmen, Erreichung der max. Beschäftigungszeit gemäß dem Hochschulrahmengesetz, ...):

- Iris Heller (halbtags)
- Dr. Ursula Hilgers
- Roland Karch
- Jochen Kaiser
- Dr. Harald Kerscher
- Ralf Kleineisel
- Birgit König (halbtags)
- Dr. Stephan Kraft
- Christina Putsche (halbtags)
- Brigitte Schmalfeld (halbtags)
- Evelyn Weyrich (halbtags)

# 4 Erfolgskontrollbericht

## 4.1 Wissenschaftliches Ergebnis

Es wurde ein weit fortgeschrittenes System entwickelt, um One-Way-Delaymessungen im G-WiN durchzuführen. Von unterschiedlichen Seiten kamen bereits Anfragen, dieses System auch außerhalb des G-WiNs einzusetzen.

## 4.2 Erfindungs- und Schutzanmeldungen

Im Verlauf des Projektes gab es keine Erfindungs- und Schutzanmeldungen.

„Geistiger Eigentümer“ der Basisentwicklungen des IPPM-Messprogramms ist das RRZE der Universität Erlangen-Nürnberg.

## 4.3 Eventuelle wirtschaftliche Erfolgsaussichten nach Auftragsende

Derzeit ist eine Vermarktung des IPPM-Messprogramms nicht geplant.

## 4.4 Eventuelle wissenschaftliche und/oder technische Erfolgsaussichten nach Auftragsende

Die gewonnen Ergebnisse können dazu verwendet werden, das G-WiN zukunftssicher weiterzuentwickeln. Insbesondere finden speziell die erzielten Ergebnisse der IP-Dienstgütemessungen (Erfahrung und Messprogramm) voraussichtlich auch Anwendung im zukünftigen, europäischen GÉANT-II Projekt, das noch im Herbst 2004 starten soll. Dort soll das Messprogramm an die europäischen Gegebenheiten angepasst und durch etwaige neue, innovative Erkenntnisse verbessert werden.

## 4.5 Eventuelle wissenschaftliche und wirtschaftliche Anschlussfähigkeit für eine nächste Phase

Die entwickelten Verfahren zur IP-Dienstgüteüberwachung, zum Accounting und zur SDH/WDM-Qualitätskontrolle dienen der Überwachung des G-WiNs und können durch Weiterentwicklung, Kombination und Ergänzung auch im Nachfolgenetz zur Qualitätssicherung genutzt werden bzw. Aufgaben eines zeitnahen Alarmsystems übernehmen.

## 4.6 Arbeiten, die zu keiner Lösung geführt haben

Es gab im Verlauf des Projektes keine Arbeiten, die zu keinem Ergebnis geführt haben.

## 4.7 Präsentationsmöglichkeiten für mögliche Nutzer

Die Nutzer wurden bei Tagungen regelmäßig über den Stand der Untersuchungen informiert.

### 4.8 Einhaltung der Kosten- und Zeitplanung

Der Kostenrahmen wurde exakt eingehalten. Da das diesem vorangegangene Entwicklungsprojekt TK 602-NT118 später endete als geplant, startete dieses Projekt nicht zum 1.11.2001, sondern erst mit Verzögerung zum 1.4.2002. Außerdem endete es auch erst zum 30.6.2004, da Sachmittel eingespart bzw. Stellen kurzzeitig unbesetzt waren.

## 5 Literatur


[Acc00] *IP-Accounting im G-WiN*, Konzept, G-WiN Labor, Erlangen, Juni 2001

[Acc02] *IP-Accounting im G-WiN*, Konzept, Version 2.0, G-WiN Labor, März 2003.

[ARTS] *http://www.caida.org/tools/utilities/arts/*, Stand September 2002.

[AMP01] *Homepage des Projektes AMP* http ://watt.nlanr.net/. Stand Oktober 2001.

[BI2003] G-WiN-Labor, *Qualitätsmessungen im deutschen Wissenschaftsnetz*, Benutzer-Information des RRZE der Universität Erlangen-Nürnberg, Heft 70, Oktober 2003.

[BiK01] F. Dressler, U. Hilgers, P. Holleczek, *Voice over IP in Weitverkehrsnetzen?* Betrieb von Informations- und Kommunikationssystemen, Tübingen BIK2001, April 2001

[CNM] http://www.cnm.dfn.de.

[CR01] *Homepage von RUDE und CRUDE*, http://www.atm.tut.fi-rude Stand Oktober 2001..

[CT00] *CAR und DTS im G-WiN*. Bericht des G-WiN Labors, März 2000.

[DFN01] U. Hilgers, S. Naegele-Jackson, P. Holleczek, R. Hofmann, *Bereitstellung von Dienstgüte in IP- und ATM-Netzen als Voraussetzung für die Videoübertragung mit Hardware Codec*, 15. DFN-Arbeitstagung über Kommunikationsnetze, „Innovative Anwendungen in Kommunikationsnetzen“, Düsseldorf, Juni 2001.

[Endace] http://www.endace.com

[EUNIS01a] S. Naegele-Jackson, U. Hilgers, P. Holleczek, *Evaluation of Codec Behavior in IP and ATM Networks*. In: J. Knop, P. Schirmbacher (Hrsg.), Proceedings EUNIS 2001, März 2001.

[EUNIS01b] U. Hilgers, R. Hofmann, *QoS-ATM versus Differentiated Services*. In: J. Knop, P. Schirmbacher (Hrsg.), Proceedings EUNIS 2001, März 2001.

[Glab00] *Planungspapier: Abnahme des G-WiN*, Bericht des G-WiN Labors, Juni 2000.

[Glab01] *Abschlussbericht des G-WiN Labors*, G-WiN Labor, Erlangen, Juni 2001.

[Glab02] *1.Zwischenbericht: Qualitätsmessungen im G-WiN (TK 602 - NT 201)*, Erlangen, September 2002.

[Glab03] *2.Zwischenbericht: Qualitätsmessungen im G-WiN (TK 602 - NT 201)* , Erlangen, März 2003.

[Glab04] *3.Zwischenbericht: Qualitätsmessungen im G-WiN (TK 602 - NT 201)* , Erlangen,

September 2003.

[Hil98] U. Hilgers, *QoS von IP-Verbindungen unter Realzeitbedingungen*. In: Peter Holleczek (Hrsg.), Pearl 98, Springer 1998.

[Hof01] G. Hofmann, *Implementation eines Programms zur Bestimmung der Dienstgüte in IP-Netzen*. Diplomarbeit, RRZE 2001.

[Net01] *Weltweit niedrigste Verzögerungszeit*, in NetworkWorld, 6. April 2001.

[NTP] Network Time Protocol, http://www.ntp.org.

[Pearl03] R. Kleineisel, I. Heller, S. Naegele-Jackson, *Messung von Echtzeitverhalten im G-WiN*, Verteilte Echtzeitsysteme (PEARL 2003) der GI-FG.4.4.2 Echtzeitprogrammierung, Boppard, 27./28.11.2003

[Pik00] U. Hilgers, R. Hofmann, P. Holleczek, *Differentiated Services – Konzepte und erste Erfahrungen*. In: Praxis der Informationsverarbeitung und Kommunikation, Februar 2000.

[PMA01] *Homepage von PMA* http://pma.nlanr.net/PMA/. Stand Oktober 2001.

[PROJ01] Dr. Peter Holleczek, G-WiN Labor, *Projektantrag: Qualitätsmessungen im G-WiN, Laufzeit: 1.11.2001 bis 31.10.2003*

[QU01] *Abschlußbericht QUASAR*

*http://www.ind.uni-stuttgart.de/Content/Quasar/publications/M6.pdf*

[RI01] *Homepage von RIPE* http://www.ripe.net. Stand Oktober 2001.

[Rout01] *Router-Test-Labor*, Antrag des G-WiN Labors, Erlangen April 2001.

[Rout02] *Abschlußbericht Router-Test-Labor*, Erlangen, Februar 2003.

[SCAMPI00] *http://www.ces.net/doc/2003/research/scampi.html*

[SCAMPI01] Sven Ubik, Pavel Cimbal CESNET, *Debugging end-to-end performance in commodity operating systems*

[Trace] *http://kea.informatik.uni-leipzig.de/traces/erlangen/*

[OWAMP1] *http://e2epi.internet2.edu/owamp/*

[OWDP01] *http://www.internet2.edu/~shalunov/ippm/draft-ietf-ippm-owdp-reqs-04.txt*

# 6 Anhang

[Flat02] *Testergebnisse Flatrate 2003*, Version 1.0, G-WiN Labor, September 2002.

[Test4OC3] *Testbericht: Komponententests Cisco 4OC3X/POS-LR-LC-B*, G-WiN Labor, 2003

[TetraGE04] *Testergebnisse Tetra GE-SFP-LC*, Version 1.0, G-WiN Labor, Mai 2004.

# 7 Abkürzungen

| | |
|---|---|
| ACL | Access Control List |
| API | Application Programming Interface |
| AR | Access-Router |
| AS | Autonomous System |
| ATM | Asynchronous Transfer Mode |
| AU-OID | Objetct-Identifiert für Anwenderübergabepunkt |
| B-WiN | Breitband Wissenschaftsnetz |
| BGP | Border Gateway Protocol |
| CAR | Committed Access Rate |
| CNM | Customer Network Management |
| CoS | Class of Service |
| CR | Core-Router |
| DCF77 | Deutschlandweites Zeitsignal, Frequenz 77,5 kHz |
| DE-CIX | Deutsche Commercial Internet Exchange |
| DFN | Deutsches Forschungsnetz |
| DFN-GS | Geschäftsstelle des DFN-Vereins |
| DFN-NOC | Network Operating Center des DFN-Vereins |
| DNS | Domain Name Service |
| DTS | Distributed Traffic Shaping |
| DVMRP | Distance-Vector Multicast Routing Protocol |
| eTTS | Einheitliches Trouble-Ticket-System |
| GAS | G-WiN Accounting System |
| GE | Gigabit Ethernet |
| GEANT | pan-European Gigabit research network |
| GIS | G-WiN Informations System |
| GMD | Gesellschaft für Mathematik und Datenverarbeitung |
| GPS | Global Positioning System |
| GSR | Giga-Switch-Router |
| G-WiN | Gigabit Wissenschaftsnetz |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| IPPM | IP Performance Metrics |
| IR | Interconnect-Router |
| ISP | Internet Service Provider |

| | |
|---|---|
| JRA1 | Joint Research Activity 1 |
| ITU | International Telecommunication Union |
| KR | Kundenrouter |
| L1 | Level1 |
| L2 | Level2 |
| LAN | Local Area Network |
| LC | Line Card |
| LSZ | Leitungsschlüsselzahl |
| MRTG | Multi Router Traffic Grapher |
| MPLS | Multiprotocol Label Switching |
| MTBF | Mean Time Between Failure |
| MTU | Maximum Transfer Unit |
| NMS | Network Management System |
| NMC | Network Management Center der Telekom |
| NTP | Network Time Protokoll |
| OC | Optical Carrier |
| OID | Object Identifier |
| OSPF | Open Shortest Path First |
| OWD | One-Way-Delay |
| PM | Performance Management |
| PoS | Packet over SONET |
| PzP | Punkt-zu-Punkt |
| QoS | Quality of Service |
| RED | Random Early Detection |
| RP | Route Prozessor |
| RRZE | Regionales Rechenzentrum Erlangen |
| SDH | Synchronous Digital Hierarchy |
| SLA | Service Level Agreement |
| SONET | Synchronous Optical Network |
| STM | Synchronous Transport Module |
| STS-MUX | Synchronous Transport Signal - Multiplexer |
| TT | Trouble Ticket |
| TCP | Transmission Control Protocol |
| ToS | Type of Service |
| UDP | User Datagram Protocol |

| | |
|---|---|
| VoIP | Voice over IP |
| WAN | Wide Area Network |
| WRED | Weighted Random Early Detection |

# Testergebnisse Flatrate 2003

(Version 1.0)

(G-WiN-Labor)

G-WiN-Labor
Regionales Rechenzentrum Erlangen (RRZE)
Martensstr. 1
91058 Erlangen
e-Mail: g-lab@rrze.uni-erlangen.de

18. Oktober 2004

# Inhalt

# 1 Motivation

Der DFN-Verein plant unter dem Arbeitstitel „Flatrate2003“ den DFN-Internet Dienst ohne Volumenbegrenzung nach bisherigem Muster anzubieten. Statt dessen soll der Dienst zwei Grundsätzen folgen:

1. Pro Kategorie soll statt des bisherigen monatlichen Datenvolumens (GByte/Monat) eine mittlere IP-Bandbreite (Mbps) festgelegt werden. Die mittlere Bandbreite entspricht einer Umrechnung eines mittleren monatlichen Datenvolumens in eine mittlere sekündliche Datenrate. Vorgesehen sind die in *Tabelle 1* aufgeführten mittleren Bandbreiten.
2. Der DFN-Verein – und nicht wie bisher der Anwender – stellt die vertragsmäßige Nutzung der Kategorie sicher. Dazu stellt der DFN-Verein bei Überschreiten des Monatsvolumens - entsprechend der mittleren Bandbreite - die Bandbreite auf die vertragsmäßige Bandbreite ein.

Der Vorteil für den Anwender besteht darin, dass z.B. mit einer 34 Mbps Zugangsleitung und einer vertraglichen IP-Bandbreite von 17 Mbps in Spitzenzeiten durchaus die volle Bandbreite von 34 Mbps für kurze Zeit genutzt werden kann.

| Bandbreite der Zugangsleitung (Mbps) | Mittlere Bandbreite (Mbps) |
|---|---|
| 2 | 1 |
| 2 | 2 |
| 34 | 4 |
| 34 | 8 |
| 34 | 17 |
| 34 | 34 |
| 155 | 77 |
| 155 | 155 |
| 622 | 311 |
| 622 | 622 |
| 2480 | 1240 |
| 2480 | 2480 |

*Tabelle 1: Einteilung der mittleren Bandbreiten.*

Zu testen ist nun, ob mit den im G-WiN eingesetzten Interfaces und Routern die notwendigen Begrenzungen für den Verkehr, den die Anwender empfangen, eingestellt werden können, ohne dass auf den Geräten Performanceprobleme entstehen. Die Tests sollen nur Regulierungen des Verkehrs untersuchen, der vom Kunden empfangen wird, da dies dem aktuellen Entgelt-Modell des G-WiN entspricht.
Des weiteren soll untersucht werden, welche Auswirkungen die Begrenzungen haben, z.B. auf andere Verkehrsflüsse.

## 2 Vorbemerkungen

Zu Beginn und auch während der Tests ergaben sich einige Schwierigkeiten. Sehr viel Zeit wurde benötigt, um herauszufinden, auf welchen Interfacekarten Shaping, auf welchen Policing möglich war und wie diese Verfahren auf den jeweiligen Karten konfiguriert werden. Die hierzu vorhandenen Informationen auf dem WWW-Server von Cisco waren leider viel zu ungenau und zum Teil unvollständig. Nachdem wir einige organisatorische Hürden überwunden hatten, bekamen wir von Cisco eine sogenannte Feature-Matrix für den GSR, auf der ersichtlich war, für welche Karten Distributed Traffic Shaping (DTS), Policing und MQC möglich ist. MQC steht für *Modular Quality of Service Command Line Interface*. Vom DFN-NOC erfuhren wir, dass Cisco dazu rät, Policing und Shaping mit Hilfe von MQC zu konfigurieren, da dies die zukünftige Konfigurationsmethode sein wird. Fälschlicherweise nahmen wir an, wenn aufgrund der Feature-Matrix von Cisco für einen Linekartentyp sowohl Policing als auch MQC (analog zu Shaping und MQC) unterstützt werden, dass sich dann auch Policing via MQC (bzw. Shaping via MQC) auf dieser Karten konfigurieren lässt. Diese Annahme war jedoch falsch. Die irreführende Fehlerausgabe beim Versuch Shaping via MQC auf der 4fach OC3 LC zu konfigurieren, trug ebenfalls zur Verwirrung bei:

```
gsr82(config-if)#service-policy output shape
shaping is not supported by platform
```

Nicht Shaping im allgemeinen wird auf dieser Karte nicht unterstützt, sondern nur Shaping via MQC. Nach vielen Mailwechseln mit Cisco, Tests auf den Routern, wie sich die Verfahren konfigurieren lassen, und dem Studium vieler Dokumente auf dem Cisco-WWW-Server kamen wir zu folgendem Ergebnis:

| | **Policing** | | **Shaping** | |
|---|---|---|---|---|
| | ohne MQC (CAR[1]) | via MQC | ohne MQC (traffic-shape rate) | via MQC |
| 75xx: 2fach E3 PA | supported | supported | supported | supported |
| GSR: 6 fach E3 (Engine 0[2]) | supported | not supported | not supported | not supported |
| GSR:4 fach OC3 (Engine 0) | supported | not supported | supported | not supported |

Policing via MQC wird in der Cisco Literatur auch als *Class-Based Policer* bezeichnet. Im Unterschied zu CAR werden beim *Class-Based Policer* im 75xx unter anderem zwei *Token Buckets* verwendet, bei CAR hingegen nur einer. Somit werden ankommende Pakete unterschiedlich behandelt (*Advanced Network Services: QoS Guidelines for Cisco 7500/7200 and Lower-End Platforms*).

[1] CAR: Committed Access Rate

[2] Die maximale Paketrate einer Engine 0 Karte liegt nach Aussage von Cisco bei ca. 250000 p/s und ist somit im Vergleich zu höheren Engines nicht so leistungsstark.

Auch Hardwareprobleme führten zur Verzögerung der Tests. Auf dem 75xx stellte sich heraus, dass eines der vom DFN für die Tests gestellten VIP Boards defekt war. Es kostete einigen Aufwand, um die genaue Ursache (ein defekter Speicher) festzustellen, und das Problem zu beseitigen. Weiter Probleme ergaben sich durch die gemeinsame Nutzung des Testequipments mit dem Router-Test-Labor. Zwar ließen sich die Tests zeitlich gut koordinieren, durch die Verschiedenheit der Tests waren jedoch immer wieder Umkonfigurationen nötig.

# 3 Tests am GSR

Am GSR wurden die 6 fach E3 Linecard und die 4 fach OC3 Linecard untersucht.

## 3.1 6 fach E3 Linecard

Diese Linecard war bisher noch nicht im G-WiN im Einsatz. Aus diesem Grund wurden zunächst einige allgemeine Tests (Durchsatztests, Kompatibilitätstests, Überlasttests) durchgeführt, anhand denen man die grundsätzliche Leistungsfähigkeit dieser Karte ergründete. Die Testergebnisse werden noch in einem separaten Testbericht zusammengeschrieben. Zusammenfassend lässt sich sagen, dass die Karte bei großen Paketen (4096 B) Linerate schafft, bei mittleren (429 B) jedoch nur 9690 p/s (= 33.25 Mbps) von 9706 p/s (= 33.31 Mbps) theoretisch maximal möglichen. Die theoretisch maximale Paketrate (*HDLC_max_pps)* für die jeweilige Paketgröße berechnet sich nach einer Formel von Cisco wiefolgt:

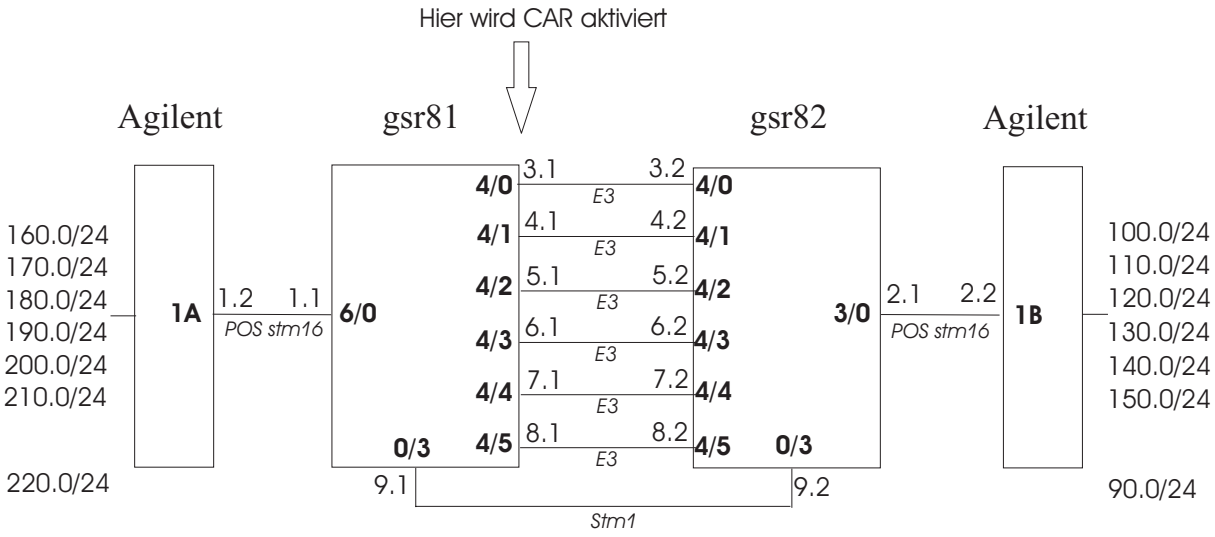
Theoretische max. Paketrate auf E3:

*HDLC_max_pps = 34010000 / ((IP_size + PPP_ovh_byte/pkt)*8)*
*PPP_ovh_byte/pkt für CRC32 = 9 bytes*

Daraus ergibt sich bei einer IP-Paketgröße von 429 B die theoretisch maximal möglichen 9706 p/s.

Für die Flatrate-Tests an der 6 fach E3 LC wurde stets der folgende Testaufbau verwendet:

IP-Adressen: 192.129.x.y




Die IP-Adressen wurden aus dem Bereich 192.129.x.y genommen. Aus Übersichtsgründen wurden jeweils nur die x- und y-Werte in der Zeichnung dargestellt. Als Verkehrsquelle und Senke diente der Routertester von Agilent, der über jeweils eine STM16 Leitung mit den beiden GSRs verbunden ist. Die beiden GSR (gsr81 und gsr82) sind vom Typ 12008. In den

folgenden Tests waren (es sei denn es wird explizit erwähnt) die beiden GSRs nur mit den nötigsten Eigenschaften in einer Laborumgebung konfiguriert. *Multicast, Netflow-Accounting, Accesslisten, Routingprotokolle* die im Betrieb eingesetzt werden, waren also nicht eingeschaltet. Als IOS wurde die Version IOS 12.0.21(S) (gsr-p-mz.120-21.S1.bin) verwendet, die auch zur Zeit im G-WiN aktiv ist. Insgesamt erzeugt der Routertester 12 Flows, 6 in Richtung gsr82 (hin), 6 in Richtung gsr81 (rück). Dabei geht jeder der 6 Hinströme (analog zu den Rückströmen) über einen anderen E3-Link. Die Flows (UDP-Pakete, *Precedence Bit* 0) werden auf der anderen Seite vom Routertester wieder empfangen und analysiert. CAR wird an unterschiedlich vielen E3 Links am gsr81 (Slot 4) aktiviert und zwar in ausgehender Richtung, also beim Verkehr vom gsr81 in Richtung gsr82. Somit entspricht der gsr81 einem Accessrouter AR im G-WiN und jeder E3-Link am gsr82 je einem zugehörigen Kundenrouter KR. Über einen STM1 Link ( 192.129.220.0/24 -- gsr81 --stm1-- gsr82 -- 192.129.90.0/24) wurde ebenfalls Verkehr geschickt, um zu verifizieren, ob CAR Einfluss auf andere Interfaces hat.

Gesendet werden auf allen 6 Links duplex mit je 1035 p/s (bei 4096 Byte Paketen -> Linerate) bzw. mit 9620 p/s (bei 429 Byte Paketen -> etwas unterhalb der Linerate) außer auf den Links an denen CAR eingeschaltet ist. Da wird mit der jeweiligen Überschreitungsrate (s.u.) gesendet. Diese gibt an, um wieviel Prozent der Kunde das mit dem DFN-Verein vereinbarte Verkehrsvolumen überschreitet. Mit Hilfe verschiedener Überschreitungsraten soll festgestellt werden, welchen Einfluss die Verkehrslast und somit die Anzahl der durch CAR zu verwerfenden Pakete auf die CPU und somit auf andere Verkehrsströme hat. Alle Pakete sind *instrumented*, d.h. der Agilent-Routertester schreibt in jedes Paket einen Zeitstempel (und Sequenznummer) anhand dessen die Laufzeit eines Paketes bestimmt werden kann.

Überschreitungsrate:

Bei 4096 B Paketen:

| Überschreitungsrate in Abhängigkeit des vereinbarten, erlaubten Verkehrsvolumens | 10% | 20% | 50% | 100% | 120% |
|---|---|---|---|---|---|
| 17.00 Mbps ( ~ 517 p/s) | 569 p/s | 621 p/s | 776 p/s | 1035 p/s | -- |

Bei 429 B Paketen:

| Überschreitungsrate in Abhängigkeit des vereinbarten, erlaubten Verkehrsvolumens | 10% | 20% | 50% | 100% | 120% |
|---|---|---|---|---|---|
| 17.00 Mbps (~ 4851 p/s) | 5336 | 5821 | 7277 | 9620=98.3% | -- |
| 8.00 Mbps (~2283 p/s) | 2511 | 2739 | 3424 | 4566 | 5022 |
| 4.00 Mbps (~1141 p/s) | 1255 | 1369 | 1712 | 2283 | 2511 |

Wird beispielsweise das vertraglich vereinbarte Verkehrsvolumen auf 8 Mbps festgelegt und möchte man testen, wie der Router sich verhält, wenn der Kunde dieses Verkehrsvolumen um ca. 20% überschreitet, so muss im Test mit 429 B Paketen eine Senderate von 2739 p/s und die Policingrate auf dem Interface auf 8 Mbps eingestellt werden.

Erklärung einiger Tabellenwerte:

| | |
|---|---|
| *delay[us]* | Hier werden jeweils das Minimum und Maximum der Paketlaufzeiten aller 12 Ströme dargestellt |
| *ref/xLink/**diff*** | *ref*: Referenzmessung ohne CAR<br>*xlink*: CAR ist an x Links aktiviert<br>*diff*: CPU-Erhöhung durch Einschalten von CAR (Differenz aus *xLink* und *ref*)<br>Da je nach Anzahl der Links, an denen CAR aktiviert ist, die Senderate variiert s.o., muss hier jeweils eine erneute Referenzmessung gemacht werden, um das Verhalten des Routers, mit exakt derselben Senderate, mit und ohne CAR vergleichen und daraus die CPU-Erhöhung berechnen zu können |

Bei allen folgenden Tests konnte festgestellt werden, dass die Cisco-Router den Verkehr jeweils gut knapp unter der eingestellten Policingrate begrenzen. Wird es in der Testbeschreibung nicht explizit erwähnt, so kommt es bei allen folgenden Tests jeweils zu keinen Paketverlusten auf den gesendeten Verkehrsflüssen, außer natürlich auf denjenigen, an denen Policing eingeschaltet ist.

### 3.1.1 Policing auf 17 Mbps:

1) Tests mit 4096 Byte Paketen (Senderate nicht gepolicter Verkehr: 1035 p/s (33.9 Mbps))

10 % Überschreitung:

| | ref/1Link/**diff** | ref/2Link/**diff** | ref/3Link/**diff** | ref/6Link/**diff** |
|---|---|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 7-8/9/**1-2** | 8/9/**1** | 7-8/9/**1-2** | 7-8/10/**2-3** |
| Delay[us] | 1168-1547/<br>1168-1568 | 1168-1570/<br>1170-1576 | 1167-1590/<br>1169-1641 | 1167-1572/<br>1169-1564 |

Sendet man mit ca. 10 % mehr Verkehr als 17 Mbps und polict diesen auf 17 Mbps, so erhöht sich dadurch die CPU am E3 Interface (Slot 4) um ca. 1-3%. Die Haupt-CPU des Routers erhöht sich nicht. Aus Zeitgründen konnte weitere Überschreitungsraten leider nicht getestet werden.

2) Tests mit 429 Byte Paketen (Senderate nicht gepolicter Verkehr: 9620 p/s (33.02 Mbps))

Bei diesen Tests mit 429 Byte Paketen konnte leider nicht mit voller Linerate auf den nicht gepolicten Strömen gesendet werden, da die E3-Karte nicht leistungsfähig genug ist (siehe Anfang Kapitel 3.1). Um unerwünschte Paketverluste (die zu Verwirrungen bei den Flatrate Tests führen) zu vermeiden, wurde die Senderate auf einen sicheren Wert von 9620 p/s (33.02 Mbps) eingestellt, bei der die Karte nicht an ihre Leistungsgrenze stößt. Im folgenden wurde auch das Eingangsinterface (Slot 6) am gsr81 beobachtet, um auszuschließen, dass CAR dort zusätzliche Last erzeugen könnte.

10 % Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 27/35-37/**8-10** | 26/42-44/**16-18** |
| Slot 6[%] | 5/5-6/**0-1** | 5/5/**0** |
| Delay[us] | 146-462/147-509 | 147-446/148-642 |

20% Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 27/34-36/**7-9** | 26/43-45/**17-19** |
| Slot6 [%] | 6/6/**0** | 5-6/5-6/**0** |
| delay[us] | 147-450/147-513 | 147-417/148-645 |

50% Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 28/37/**9** | 27/47/**20** |
| Slot6 [%] | 5-6/5-6/**0** | 5/5/**0** |
| delay[us] | 147-390/147-513 | 147-424/148-642 |

98% Überschreitung:
Eine 100%ige Überschreitungsrate ist hier ohne Paketverluste leider nicht möglich (s.o.), deshalb wird hier auf den „sicheren“ Wert von 9620 p/s (=98.3% Überschreitung) zurückgegriffen.

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 28/38/**10** | 28/50/**22** |
| Slot6 [%] | 6/6/**0** | 6/6/**0** |
| delay[us] | 147-470/149-491 | 147-470/149-543 |

### 3.1.2 Policing auf 4 Mbps

Tests mit 429 Byte Paketen (Senderate nicht gepolicter Verkehr: 9620 p/s (33.02 Mbps)).

10 % Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 26-28/35-36/**7-10** | 26-27/35-36/**8-10** |
| Slot 6[%] | 6/6/**0** | 5-6/5-6/**0** |
| Delay[us] | 147-465/147-524 | 147-413/149-643 |

50% Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 27/35-36/**8-9** | 26-27/35-38/**8-12** |
| Slot 6[%] | 6/6/**0** | 5-6/5-6/**0** |
| Delay[us] | 147-465/147-492 | 147-450/149-564 |

120% Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 26-27/35/**8-9** | 26-27/37-38/**10-12** |
| Slot 6[%] | 5-6/5-6/**0** | 5-6/5-6/**0** |
| Delay[us] | 147-465/147-526 | 147-422/149-642 |

### 3.1.3 Policing auf 8 Mbps

Tests mit 429 Byte Paketen (Senderate nicht gepolicter Verkehr: 9620 p/s (33.02 Mbps)).

10 % Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 26-27/34-35/**7-9** | 26-27/37/**10-11** |
| Slot 6[%] | 5/5/**0** | 5-6/5-6/**0** |
| Delay[us] | 147-429/147-518 | 147-444/149-642 |

50% Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 26/34-36/**8-10** | 26-27/37-40/**10-14** |
| Slot 6[%] | 5-6/5-6/**0** | 5-6/5-6/**0** |
| Delay[us] | 147-469/**147-474** | 147-390/**149-572** |

120% Überschreitung:

| | ref/1Link/**diff** | ref/6Link/**diff** |
|---|---|---|
| main CPU[%] | 1/1/**0** | 1/1/**0** |
| Slot 4[%] | 26/36/**10** | 26/41-42/**15-16** |
| Slot 6[%] | 5-6/5-6/**0** | 5-6/5-6/**0** |
| Delay[us] | 147-418/147-523 | 147-421/148-645 |

Die folgende Tabelle gibt einen Überblick der oben beschriebenen Tests über den CPU-Anstieg am Ausgangsinterface in Abhängigkeit von der Policingrate, der Überschreitungsrate und der Anzahl der Links an denen CAR aktiviert wurde. Alle Ergebnisse bewegen sich im Bereich zwischen 8 und 22 % CPU-Mehrlast.

| Überschreitungsrate [%] | Policing Rate [Mbps] | CPU-Erhöhung durch CAR an 1 Link [%] | CPU-Erhöhung durch CAR an 6 Links [%] |
|---|---|---|---|
| 10 | 4 | 7-10 | 8-10 |
| 50 | 4 | 8-9 | 8-12 |
| 120 | 4 | 8-9 | 10-12 |
| | | | |
| 10 | 8 | 7-9 | 10-11 |
| 50 | 8 | 8-10 | 10-14 |
| 120 | 8 | 10 | 15-16 |
| | | | |
| 10 | 17 | 8-10 | 16-18 |
| 20 | 17 | 7-9 | 17-19 |
| 50 | 17 | 9 | 20 |
| 98.3 | 17 | 10 | 22 |

Je höher die Überschreitungsrate, desto höher die CPU-Lasterhöhung.
CAR an 6 Links führt zu einer höheren CPU-Lasterhöhung als CAR an einem Link.

### 3.1.4 Einfluss von Betriebsfeatures beim Policing

Alle bisherigen Tests wurden unter reinen Laborbedingungen durchgeführt. Im folgenden werden einige Betriebsfeatures aktiviert, um das Verhalten von Policing unter betriebsnäheren Bedingungen zu untersuchen. *Multicast*, *Sampled Netflow*, *Accesslisten* und eine Routingtabelle von ca. 114000 Einträgen wurden aktiviert. Um noch näher an eine Betriebsumgebung zu kommen, müsste zusätzlich zu den genannten Features auch noch der Verkehr angepasst (echter Multicastverkehr, viele verschiedene Flows) und dynamisches Routing aktiviert werden, was nur mit viel Aufwand zu realisieren wäre und dann wohl dennoch nicht genau die Betriebsbedingungen wieder geben würde. Für diese Tests musste der *route memory* an allen beteiligten Interfaces auf 256 MB aufgerüstet werden. Ohne diesen Upgrade verklemmte sich der Router völlig nach dem Einspielen der Routingtabelle vom DFN-NOC.

Somit sieht die Konfiguration an einem E3 Link wie folgt aus:

```
interface Serial4/0
 ip address 192.129.3.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast reverse-path
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast ttl-threshold 32
 ip multicast boundary 18
 encapsulation ppp
 ip route-cache flow sampled
 ip mroute-cache distributed
 crc 32
 down-when-looped
 clock source internal
 no cdp enable
```

```
!

access-list 199 deny   ip host 193.61.196.132 host 141.38.43.44
access-list 199 deny   ip host 193.61.196.132 host 141.38.45.34
access-list 199 deny   ip host 193.61.196.132 host 141.38.46.45
access-list 199 permit ip any any
```

Zuerst wurde getestet, ob der Router bereits durch die aktivierten Betriebsfeatures, aber noch deaktiviertem Policing, die Senderate von 9620 p/s (33.02 Mbps) bei 429 B Paketen ohne Probleme durchbekommt. Dies konnte bestätigt werden. Die CPU an der LC liegt hier bei etwa 53%, die des Routeprozessors schwankend im 1-Minutenmittel bei ca. 3-6%, und die des Eingangsinterfaces *slot 6/0* bei ca. 5%. Um zu überprüfen, ob CAR zusammen mit eingeschalteten Betriebsfeatures Einfluss auf andere Interfaces hat, wurde zusätzlich auf dem STM1 Link *pos0/3* mit 145.41 Mbps gesendet und der Verkehr beobachtet und analysiert.

#### 3.1.4.1 Policing an 5 Links

Zunächst wurde an fünf der sechs Links Policing auf 17 Mbps aktiviert und auf allen 12 Strömen mit 9620 p/s (33.02 Mbps) gesendet. Es kam weder zu Paketverlusten auf dem sechsten Link, noch auf den Rückströmen noch auf der STM1 Leitung. Die CPU der 6 fach E3 LC lag bei 70%. Am Eingangsinterface oder am Routeprozessor konnte keine Veränderung festgestellt werden. Polict man auf 4 Mbps und wiederholt den soeben genannten Test, so kam es ebenfalls zu keinen unerwünschten Paketverlusten. Die CPU der E3-Karte lag bei ca. 70%.

#### 3.1.4.2 Policing an 6 Links

Polict man an allen sechs Links auf 17 Mbps und sendet auf allen 12 Strömen mit 9620 p/s (33.02 Mbps), so kam es weder zu Paketverlusten auf den Rückströmen noch auf der STM1 Leitung. Die CPU der 6 fach E3 LC lag bei 73-74%. Am Eingangsinterface oder am Routeprozessor konnte keine spürbare Veränderung festgestellt werden. Polict man auf 4 Mbps und wiederholt den soeben genannten Test, so kam es ebenfalls zu keinen unerwünschten Paketverlusten. Die CPU der E3-Karte lag bei ca. 72%.

### 3.1.5 Überprüfen der Statistik-Parameter

Mit Hilfe des Befehls "*sh int serial 4/0 rate-limit*" lassen sich einige Statistiken über Anzahl der durch CAR verworfenen Pakete (*exceeded packets*), der durchgelassenen Regel getreuen Pakete (*conformed packets*) und deren jeweiligen Raten anzeigen. Folgende Ausgabe gibt ein Beispiel wieder:

```
gsr81#sh int serial 4/0 rate-limit
Serial4/0
  Output
    matches: all traffic
     params: 8000000 bps, 1500000 limit, 3000000 extended limit
     conformed 1266339 packets, 548324787 bytes;action: transmit
     exceeded 106425 packets,46082025 bytes; action: drop
     last packet: 32212ms ago, current burst: 1626189 bytes
     last cleared 00:09:44 ago, conformed 7503537 bps, exceeded 630608 bps
```

Stichpunktartig wurde diese Ausgabe mit den gemessenen Werten vom Agilent überprüft:
Die *conformed packets* und die *exceeded packets* stimmen plus/minus ein Paket in ganz wenigen Fällen – exakt mit den Werten vom Agilent überein. Hingegen stimmen die

zugehörigen Raten *conformed bps* und *exceeded bps* erst nach längerer Zeit mit den Werten vom Agilent überein (mehrere Minuten).

### 3.1.6 Zusammenfassung

CAR auf der 6 fach E3 Linecard begrenzt den Verkehr gut jeweils knapp unter der eingestellten Policingrate. CAR führt nur zu einer CPU-Lasterhöhung am Ausgangsinterface, an dem es aktiviert wurde. Der Anstieg liegt bei den durchgeführten Tests im Bereich von 8 und 22% CPU-Mehrlast. Sowohl die Haupt-CPU des Routers als auch die des Eingangsinterfaces bleiben durch den Einfluss von CAR unberührt. Außerdem führt CAR unter der oben genannten Testumgebung zu keinen Paketverlusten an den anderen Links der E3-Karte, an denen CAR nicht aktiviert war. Die auf den Routern vorhandenen Statistiken über CAR stimmen in Bezug auf Anzahl durch CAR verworfener und zugelassener Pakete sehr genau überein. Die zugehörigen Raten sind jedoch träge. Werden Betriebsfeatures wie *Accesslisten, Sampled Netflow* und *Routingeinträge* konfiguriert, so führen diese unter der vom DFN-NOC vorgeschlagenen Konfiguration, zu keiner negativen Beeinflussung ungepolicter Links oder anderer Interfaces.

## 3.2 4 fach OC3 Linecard

Als Grundlage für die Flatrate Tests ist es auch hier wichtig, die maximale Paketrate der Karte zu kennen, die deshalb zu Beginn der Tests ermittelt wurde. Der theoretische maximale Paketdurchsatz berechnet sich nach folgender Formel (*Online-Hilfe auf dem Agilent Routertester*):

Theoretischer max. Durchsatz:
*max IP throughput = line rate * SONET/SDH overhead factor * PPP/HDLC overhead factor*
*line rate = 155.52 Mbps*
*SONET/SDH overhead factor = 1040/1080*
*PPP/HDLC overhead factor = IP packetlength/(IP packetlength+9)*

| IP packetlength [Byte] | max IP throughput [Mbps] | max IP rate [p/s] |
|---|---|---|
| 429 | 146.68 | 42739.73 |
| 4096 | 149.43 | 4560.29 |

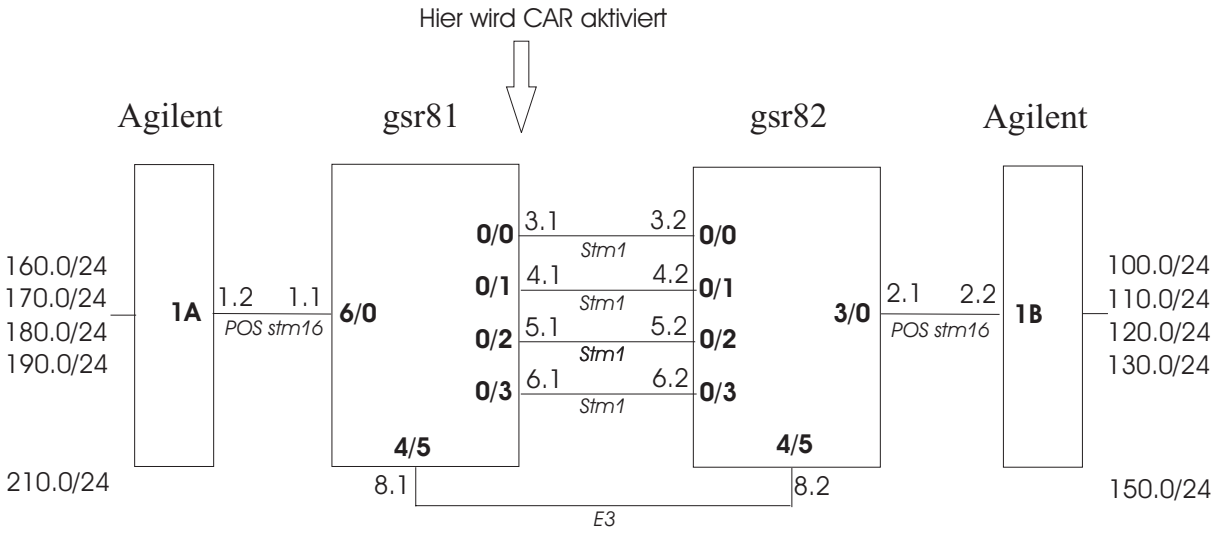
Die im Test ermittelte maximale Paketrate für 429 Byte Pakete beträgt 42385 p/s (145.47 Mbps) für einen Link und liegt somit etwas unter dem theoretischen maximalen Durchsatz. Hier stößt die 4fach OC3 LC an ihre Grenzen. Für die Flatrate Tests wurde deshalb die maximale Rate etwas reduziert (auf 42370 p/s = 145.41 Mbps), um verwirrende Paketverluste oder hohe Delayanstiege aufgrund der Leistungsgrenze der Linecard und Bytestuffing-Effekte[3] zu vermeiden. Wird mit dieser Rate auf allen 4 Links der Karte duplex gesendet, so ergibt sich dadurch eine Grund-CPU-Last von ca. 61 % auf der OC3-Karte.

### 3.2.1 Policing via CAR

Für die CAR-Tests an der 4 fach OC3 LC wurde stets der folgende Testaufbau verwendet:

IP-Adressen: 192.129.x.y

[3] Bestimmte Bitmuster müssen auf der SDH-Ebene durch Hinzufügen weiterer Bits maskiert werden, um Framegrenzen erkennen zu können. Da in unserem Fall die Pakete vom Routertester mit Zeitstempeln versehen werden, also dadurch beliebige Bitmuster in gehäufter Form auftreten, kann es passieren, dass diese zu Bytestuffing führen. Somit müssten auf SDH-Ebene, wie beschrieben, zusätzliche Bits hinzugefügt werden, was zu einer unerwünschten höheren Senderate und somit zu Paketverlusten führen kann, wenn das Interface am Router, wie in unserem Fall, keine volle Linerate verkraftet.




Die IP-Adressen wurden aus dem Bereich 192.129.x.y genommen. Aus Übersichtsgründen wurden jeweils nur die x- und y-Werte in der Zeichnung dargestellt. Als Verkehrsquelle und Senke verwendeten wir den Routertester von Agilent, der über jeweils eine STM16 Leitung mit den beiden GSRs verbunden ist. Die beiden GSR (gsr81 und gsr82) sind vom Typ 12008. In den folgenden Tests waren (es sei denn es wird explizit erwähnt) die beiden GSRs nur mit den nötigsten Eigenschaften in einer Laborumgebung konfiguriert. *Multicast, Netflow Accounting, Accesslisten, Routingprotokolle* die im Betrieb eingesetzt werden, waren also nicht eingeschaltet. Als IOS wurde die Version IOS 12.0.21(S) (gsr-p-mz.120-21.S1.bin) verwendet, die auch zur Zeit im G-WiN aktiv ist. Insgesamt erzeugt der Routertester 8 Flows, 4 in Richtung gsr82 (hin), 4 in Richtung gsr81 (rück). Dabei geht jeder der 4 Hin-Ströme (analog zu den Rückströmen) über einen anderen STM1-Link. Die Flows (UDP-Pakete, *Precedence Bit* 0) werden auf der anderen Seite vom Routertester wieder empfangen und analysiert. CAR wird an unterschiedlich vielen STM1-Links am gsr81 (Slot 0) aktiviert und zwar in ausgehender Richtung, also beim Verkehr vom gsr81 in Richtung gsr82. Somit entspricht der gsr81 einem Accessrouter AR im G-WiN und jeder STM1-Link am gsr82 je einem zugehörigen Kundenrouter KR.

Alle Pakete sind *instrumented*, d.h. der Agilent-Routertester schreibt in jedes Paket einen Zeitstempel anhand dessen die Laufzeit eines Paketes bestimmt werden kann.

Überschreitungsrate:

Bei 429 B Paketen:

| | | |
|---|---|---|
| Überschreitungsrate in Abhängigkeit des vereinbarten, erlaubten Verkehrsvolumens | 4% | 42% |
| 77.00 Mbps | 23277 p/s | 31741 p/s |

Erklärung einiger Tabellenwerte:

| | |
|---|---|
| *delay[us]* | Hier wird jeweils das Minimum und Maximum der Paketlaufzeiten aller 8 Ströme (4 hin, 4 rück) dargestellt |
| *ref* | Referenzmessung ohne CAR |
| *xLink/**diff*** | *xlink*: CAR ist an x Links aktiviert<br>*diff*: CPU-Erhöhung durch Einschalten von CAR (Differenz aus *xLink* und *ref*) |

#### 3.2.1.1 Policing auf 77 Mbps:

Aus Zeitgründen werden im folgenden alle Tests nur mit 429 Byte Paketen durchgeführt.
In einem ersten Test wurde die CPU Belastung an der 4fach OC3 LC gemessen, wenn an allen 4 Links gleichzeitig auf 77 Mbps gepolict wird, die Senderate der gepolicten Ströme 79.89 Mbps (knapp 4% Überschreitung) und die Senderate aller 4 Rückströme 42370 p/s (145.41 Mbps, knapp Linerate) betrug. Diese lag bereits bei 99%. Da dieses hohe Sendeverhalten der Rückströme wahrscheinlich nicht den realen Verkehrsflüssen entsprechen wird, wird im folgenden untersucht, in wie weit sich die CPU erhöht, wenn die Rate aller Hin- und Rückströme auf einen niedrigeren Level gleichgesetzt wird.

4 % Überschreitung
Senderate aller 8 Ströme (hin und rück): 23277p/s (79.89 Mbps)

| | ref | 1Link/**diff** | 2Link/**diff** | 4Link/**diff** |
|---|---|---|---|---|
| main CPU [%] | 1 | 1/**0** | 1/**0** | 1/**0** |
| Slot 0 [%] | 35 | 58/**23** | 65/**30** | 75/**40** |
| Slot 6 [%] | 5 | 5/**0** | 5/**0** | 5/**0** |
| Delay [us] | 72-373 | 72-586 | 73-685 | 73-754 |

Ergebnis:
Sendet man mit ca. 4 % mehr Verkehr als 77 Mbps und polict diesen auf 77 Mbps, so erhöht sich dadurch die CPU am STM1 Interface um ca. 23-40%. Die Haupt-CPU des Routers und die des Eingangsinterfaces erhöhen sich nicht. Zu Paketverlusten auf den ungepolicten Strömen kam es ebenfalls nicht. Bei den Tests mit CAR an 2 Links wurde exemplarisch gemessen, wie schnell der Router eine konstante zu hohe Senderate auf den gewünschten Wert begrenzt. Erst nach ca. 33 s traten die ersten Paketverluste auf, nach ca. 40 s erreichte der Durchsatz den Endwert von 76.3 Mbps.

42 % Überschreitung:
Senderate aller 8 Ströme: 31741 p/s (108.94 Mbps)

| | ref | 1Link/**diff** | 2Link/**diff** | 4Link/**diff** |
|---|---|---|---|---|
| main CPU [%] | 1 | 1/**0** | 1/**0** | 1/**0** |
| Slot 0 [%] | 46 | 78-80/**32-34** | 87/**41** | 97/**51** |
| Slot 6 [%] | 5 | 5/**0** | 6/**1** | 5/**0** |
| Delay [us] | 72-375 | 72-670 | 73-755 | 73-679 |

Hier kam es zu einer CPU-Erhöhung von ca. 32-51 %.

In einem zweiten Test wurde überprüft, ob diese hohe Last Einfluss auf andere Links hat, indem zusätzlich mit 9600 p/s (=32.95 Mbps) duplex über einen E3 Link (serial 4/5) gesendet wurden.

Ergebnis: dort gingen keine Pakete verloren

#### 3.2.1.2 Einfluss von Betriebsfeatures beim Policing

Alle bisherigen Tests wurden unter reinen Laborbedingungen durchgeführt. Im folgenden werden einige Betriebsfeatures aktiviert, um das Verhalten von Policing unter betriebsnäheren Bedingungen zu untersuchen. *Multicast*, *Sampled Netflow*, *Accesslisten* und eine Routingtabelle von ca. 114000 Einträgen wurden aktiviert. Um noch näher an eine Betriebsumgebung zu kommen, müsste zusätzlich zu den genannten Features auch noch der Verkehr angepasst (echter Multicastverkehr, viele verschiedene Flows) und dynamisches Routing aktiviert werden, was nur mit viel Aufwand zu realisieren wäre und dann wohl dennoch auch nicht genau die Betriebsbedingungen wieder geben würde. Für diese Tests musste der *route memory* an allen beteiligten Interfaces auf 256 MB aufgerüstet werden. Ohne diesen Upgrade verklemmte sich der Router völlig nach dem Einspielen der Routingtabelle vom DFN-NOC.

Somit sieht die Konfiguration an einem STM1 Link wie folgt aus:

```
interface POS0/0
 ip address 192.129.3.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast reverse-path
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast ttl-threshold 32
 ip multicast boundary 18
 encapsulation ppp
 ip route-cache flow sampled
 ip mroute-cache distributed
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable

access-list 199 deny   ip host 193.61.196.132 host 141.38.43.44
access-list 199 deny   ip host 193.61.196.132 host 141.38.45.34
access-list 199 deny   ip host 193.61.196.132 host 141.38.46.45
access-list 199 permit ip any any
```

Zuerst wurde getestet wie viele Pakete über die Karte geschickt werden können mit aktivierten Betriebsfeatures aber ohne CAR, ohne dass es zu Paketverlusten kommt. Der maximale Paketdurchsatz bei 429 B betrug etwas mehr als 100 Mbps pro Link. Die CPU-Last an der 4 fach OC3 LC liegt bei etwa 98-99%, die der Haupt- CPU bei etwa 3-6% und die des

Eingangsinterfaces *Slot 6/0* bei ca. 5%. Ohne Aktivieren der Betriebsfeatures konnten im Vergleich hierzu ca. 145 Mbps Durchsatz erreicht werden.
Um zu überprüfen, ob CAR zusammen mit eingeschalteten Betriebsfeatures Einfluss auf andere Interfaces hat, wurde zusätzlich auf dem E3 Link *serial4/5* mit 32.95 Mbps gesendet und der Verkehr beobachtet und analysiert.

#### 3.2.1.2.1 Policing an 3 Links

Zunächst wurde an drei der vier Links das Policing auf 77 Mbps aktiviert. Sobald mit mehr als 77 Mbps gesendet wurde (bei allen vier aktivierten Accesslisten), kam es zu Paketverlusten an allen 4 Links in Rückrichtung. Somit wird also auch der vierte Link, an dem kein Policing aktiviert ist, beeinflusst. Die Paketverluste lassen sich am Router als *ignored packets* am Interface *pos0/3* (und natürlich auch an allen anderen Links der 4fach OC3 LC) erkennen. Am ungepolicten, vierten STM1-Link kam es in Hinrichtung zu keinen Verlusten. Ebenso gingen keine Pakete an der E3 Karte verloren. Die CPU an der 4fach OC3 LC ist voll ausgelastet bei 99-100%, die Haupt-CPU und die des Eingangsinterfaces wurden durch CAR nicht sichtbar zusätzlich belastet.

Da die oben genannten 4 Accesslisten bereits bei geringfügiger Überlast zu Paketverlusten auch an dem ungepolicten Strom führen und Accesslisten im allgemeinen nicht dauerhaft aktiviert sind, wurden Tests durchgeführt, wie sich das Verhalten bei weniger oder keinen aktiven Accesslisten verhält:

Wurden alle Accesslisten deaktiviert, so konnte mit maximal 95 Mbps pro Link gesendet werden, ohne dass es zu Paketverlusten an allen 4 Rückrichtungen kam.

Wurden folgende zwei Accesslisten aktiviert
```
access-list 199 deny   ip host 193.61.196.132 host 141.38.43.44
access-list 199 permit ip any any
```
so konnte nur mit maximal 80 Mbps pro Link gesendet werden, ohne Paketverluste auf allen 4 Rückrichtungen zu bekommen. Bei einer Senderate von 95 Mbps war die Paketverlustrate so hoch, dass zum Teil nur ein Durchsatz auf den Rückrichtungen von 62 Mbps erreicht wurde.

#### 3.2.1.2.2 Policing an 4 Links

Wird auf allen 4 STM1 Links gepolict (bei vier aktivierten Accesslisten), so kam es ebenfalls zu Paketverlusten, sobald mit mehr als 77 Mbps gesendet wurde. Dabei wurden sowohl Pakete auf allen 4 Rückrichtungen als auch auf der E3 Hinrichtung verworfen! Lediglich die Rückrichtung der E3 Leitung blieb fehlerfrei.

Wurden alle Accesslisten deaktiviert, so konnte mit maximal 90 Mbps pro Link gesendet werden, ohne dass es zu Paketverlusten kam. Wurde mit 92 Mbps gesendet, so kam es zusätzlich zu Verlusten auf der E3 Hinrichtung, wurde mit 95 Mbps gesendet, so kam es zu Verlusten auf allen 4 STM1 Rückrichtungen und auf der E3 Hinrichtung, nicht jedoch auf der E3 Rückrichtung. Die Verluste auf der E3 Leitung waren als *ignored packets* bzw. als *no mem drops* am Eingangsinterface pos6/0 zu sehen (*Backpressure*).

Wurden folgende zwei Accesslisten aktiviert
```
access-list 199 deny   ip host 193.61.196.132 host 141.38.43.44
access-list 199 permit ip any any
```

so konnte nur mit maximal 80 Mbps pro Link gesendet werden, ohne Paketverluste zu bekommen. Schon bei einer Senderate von 82 Mbps kommt es zu Verlusten auf allen 4 STM1 Rückrichtungen und auf der E3 Hinrichtung.

Im Dokument „*Troubleshooting Ignored Errors and No Memory Drops on the Cisco 12000 Series Internet Router*“ von Cisco steht, wie man *no mem drops* umgehen kann, indem man entweder WRED aktiviert oder die Pufferlänge des ausgehenden Interfaces begrenzt. Beides wurde ausprobiert, jedoch ohne Erfolg. Die Konfiguration für WRED war dabei wie folgt:

```
int pos0/0
 tx-cos congestion

cos-queue-group congestion
 precedence all random-detect-label 0
 random-detect-label 0 388 1292 1
```

Die Pufferlänge wurde einmal auf 5000 und ein anderes mal auf 500 begrenzt mit:

```
tx-queue-limit 5000  oder
tx-queue-limit 500
```

Nach einem längeren Mailwechsel mit Cisco, bekamen wir von Cisco den Rat, WRED auch am Eingangsinterface *pos 6/0* in Richtung *Switch-Fabric* zu aktivieren. Dies soll verhindern, dass aufgrund der hohen Last an der 4 fach OC3 LC, sich durch *Backpressure* auch alle Puffer in Richtung *Switch-Fabric* am Eingangsinterface *pos 6/0* anstauen und überlaufen und somit andere Flows, die auch über das selbe Eingangsinterface (wie hier der Strom über den E3 Link) gehen, negativ beeinflusst. Die Konfiguration wurde dabei wie folgt vorgenommen:

```
rx-cos-slot 6 cos_stm16
!
slot-table-cos cos_stm16
 destination-slot 0 congestion_in
!
cos-queue-group congestion_in
 precedence all random-detect-label 0
 random-detect-label 0 6200 20668 1
```

Diese Konfiguration führte dazu, dass die Paketverluste auf der zusätzlichen E3 LC (die am Eingangsinterface *pos6/0* als *no mem drops* zu sehen waren) vermieden werden konnten. Paketverluste in Rückrichtung bei hoher Verkehrslast auf dem ungepolicten vierten Link werden dadurch allerdings nicht vermieden. Diese kommen dadurch zustande, dass die CPU der LC der Flaschenhals ist. Die Pakete werden somit als *ignored packets* bereits am Eingang der 4 port OC3 LC verworfen. Die hohe CPU-Last an dieser LC kann nur durch Verminderung der aktivierten Betriebsfeatures oder durch Reduzierung der Verkehrslast erzielt werden.

Ergebnis:
Sind bei der Routerkonfiguration im G-WiN Accesslisten nötig, so kommt es bereits bei vier Accesslisten zu Paketverlusten am vierten ungepolicten Link, wenn an dreien gepolict wird und knapp über 77 Mbps gesendet wird. Kann im G-WiN am AR auf Accesslisten verzichtet werden, so darf der Kunde höchstens 90 Mbps Verkehrslast, wenn an vier Links gepolict wird

und höchstens bis zu 95 Mbps Verkehrslast, wenn an drei Links gepolict wird, erzeugen, ohne dass es zu Paketverlusten kommen wird.

#### 3.2.1.3 Überprüfen der Statistik-Parameter

Mit Hilfe des Befehls "*sh int pos 0/0 rate-limit*" lassen sich einige Statistiken über Anzahl der durch CAR verworfenen Pakete (*exceeded packets*), der durchgelassenen Regel getreuen Pakete (*conformed packets*) und deren jeweiligen Raten anzeigen. Folgende Ausgabe gibt ein Beispiel wieder nach einer Messzeit von ca. 3 Minuten (gesendet wurde dabei mit einer kontinuierlichen Rate von 79.89 Mbps):

```
POS0/0
  Output
    matches: all traffic
      params:  77000000 bps, 14437500 limit, 28875000 extended limit
      conformed 4108618 packets, 1779M bytes; action: transmit
      exceeded 157328 packets, 68123024 bytes; action: drop
      last packet: 1ms ago, current burst: 15117289 bytes
      last cleared 00:03:09 ago, conformed 74976044 bps, exceeded
2870997 bps
gsr81#
```

Die *conformed packets* und die *exceeded packets* stimmen plus/minus ein Paket in ganz wenigen Fällen – exakt mit den Werten vom Agilent überein. Hingegen stimmen die zugehörigen Raten *conformed bps* und *exceeded bps* erst nach längerer Zeit mit den Werten vom Agilent überein (mehrere Minuten).

#### 3.2.1.4 Zusammenfassung

CAR auf der 4 fach STM1 Linecard begrenzt den Verkehr bei einer Policing Rate von 77 Mbps auf ca. 76.3 Mbps. CAR führt zu einer CPU-Lasterhöhung am Ausgangsinterface, an dem es aktiviert wurde. Der Anstieg liegt bei den durchgeführten Tests im Bereich von 23-51% CPU-Mehrlast, was einer Gesamt-CPU-Last von bis zu 97% entspricht, was sehr hoch ist. Sowohl die Haupt-CPU des Routers als auch die des Eingangsinterfaces bleiben durch das Einschalten von CAR unberührt. Die Tests mit aktivierten Betriebsfeatures ergaben, dass Policing an 3 oder 4 Links an der 4 fach OC3 LC bei 4 eingeschalteten Accesslisten bereits zu unerwünschten Paketverlusten führen kann, sobald Pakete durch Policing verworfen werden. Sind im Betrieb keine Accesslisten nötig, kann mit maximal 90 Mbps (Policing an 4 Links) bzw. 95 Mbps (Policing an 3 Links) fehlerfrei gesendet werden. Um weitere Paketverluste auf anderen Interfaces zu vermeiden ist es unbedingt nötig, WRED am Eingangsinterface, also am Interface vom AR, der zum G-WiN Backbone führt, in Richtung *Switch-Fabric* zu aktivieren. Da in einer echten Betriebsumgebung jedoch auch noch dynamische Routingprotokolle, Multicastverkehr und viele wechselnde Flows hinzukommen, was die CPU noch stärker belasten könnte, kann es sogar im realen Betrieb noch zu einer Verschlechterung des Ergebnisses kommen.
Die auf den Routern vorhandenen Statistiken bzgl. der Raten konformer und nicht konformer Pakete sind träge.

### 3.2.2 Shaping via "traffic-shape rate"

Einige Vorbemerkungen zum Distributed Traffic Shaping (DTS):
Im Dokument
*http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120s/120s16/dts_e2.htm*
auf dem Cisco WWW-Server steht, dass bei Aktivierung von DTS mittels "*traffic-shape rate*" unbedingt auch WRED aktiviert werden sollte, um Paketverluste auf anderen Interfaces zu vermeiden. Bei Cisco haben wir Informationen angefordert, warum es zu solchen Paketverlusten kommen kann. Bisher trafen diese jedoch nicht bei uns ein. Prinzipiell ist es nicht ganz einfach, die Parameter (*min* und *max threshold*, *max probability value* und *exponential weight factor*), die bei der WRED Konfiguration benötigt werden, optimal zu bestimmen. Diese hängen vom aktuellen Verkehrsverhalten ab. Werden sie falsch bestimmt, kann es sein, dass WRED gar nicht zum Wirken kommt oder dass es zu stark auf temporäre Verkehrsbursts reagiert, und Pakete zu früh verworfen werden, was zu einer schlechten Linkauslastung führen würde. Zur Wahl der Parameter von WRED wurde folgendes Dokument von Cisco herangezogen:

*http://www.cisco.com/univercd/cc/td/doc/product/software/ios112/ios112p/gsr/wred_gs.htm*

Hier stehen Richtwerte für diese Parameter, die man als Startpunkt für den optimalen Auswahlprozess nehmen kann. Folgende Werte wurden da festgelegt:

für OC3: min threshold: 388
max threshold: 1292
max probability value: 1

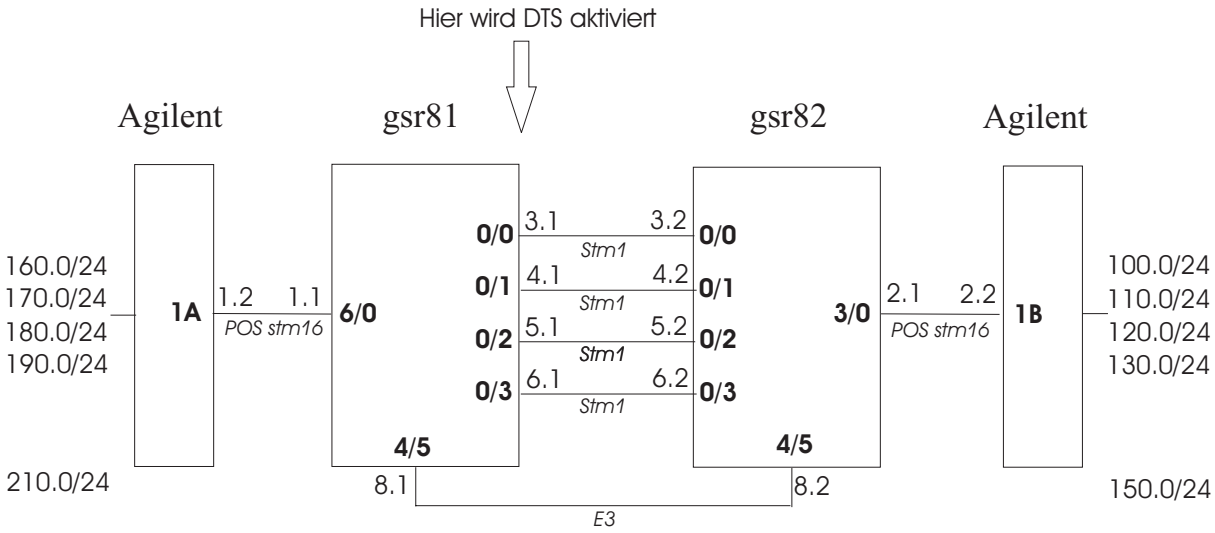
Für unsere Tests wurde WRED mit diesen Startwerte konfiguriert. Da es mit dieser Konfiguration zu keinen Paketverlusten auf anderen Interfaces kam, wurden diese nicht weiter optimiert. WRED wurde deshalb stets auf allen Links an denen DTS aktiviert wurde wie folgt konfiguriert:

```
int pos0/0
 tx-cos congestion


cos-queue-group congestion
 precedence all random-detect-label 0
 random-detect-label 0 388 1292 1
```

Für die DTS-Tests an der 4 fach OC3 LC wurde stets der folgende Testaufbau verwendet:

IP-Adressen: 192.129.x.y




Der E3 Links dient nur dazu, zu überprüfen, ob DTS Einfluss auf andere Interfaces hat. Hier wird duplex ein konst. Strom von 9600 p/s (32.95 Mbps, 429B) versendet. Die IP-Adressen wurden aus dem Bereich 192.129.x.y genommen. Aus Übersichtsgründen wurden jeweils nur die x- und y-Werte in der Zeichnung dargestellt. Als Verkehrsquelle und Senke verwendeten wir den Routertester von Agilent, der über jeweils eine STM16 Leitung mit den beiden GSRs verbunden ist. Die beiden GSR (gsr81 und gsr82) sind vom Typ 12008. In den folgenden Tests waren (es sei denn es wird explizit erwähnt) die beiden GSRs nur mit den nötigsten Eigenschaften in einer Laborumgebung konfiguriert. *Multicast, Netflow Accounting, Accesslisten, Routingprotokolle* die im Betrieb eingesetzt werden, waren also nicht eingeschaltet. Als IOS wurde die Version IOS 12.0.21(S) (gsr-p-mz.120-21.S1.bin) verwendet, die auch zur Zeit im G-WiN aktiv ist. Insgesamt erzeugt der Routertester 8 Flows, 4 in Richtung gsr82 (hin), 4 in Richtung gsr81 (rück). Dabei geht jeder der 4 Hin-Ströme (analog zu den Rückströmen) über einen anderen STM1-Link. Die Flows (UDP-Pakete, *Precedence Bit* 0) werden auf der anderen Seite vom Routertester wieder empfangen und analysiert.

DTS wird an unterschiedlich vielen STM1-Links am gsr81 (Slot 0) aktiviert und zwar in ausgehender Richtung, also beim Verkehr vom gsr81 in Richtung gsr82. Somit entspricht der gsr81 einem Accessrouter AR im G-WiN und alle vier STM1-Links am gsr82 je einem zugehörigen Kundenrouter KR. Gesendet wird auf allen 4 Links duplex mit je 42370 p/s (145.41 Mbps) bei 429 B Paketen außer auf den Links an denen geshapt wird. Da wird mit der jeweiligen Überschreitungsrate (s.u.) gesendet. Alle Pakete sind *instrumented*, d.h. der Agilent-Routertester schreibt in jedes Paket einen Zeitstempel anhand dessen die Laufzeit eines Paketes bestimmt werden kann.

Überschreitungsrate:

Bei 429 B Paketen:

| | | |
|---|---|---|
| Überschreitungsrate in Abhängigkeit des vereinbarten, erlaubten Verkehrsvolumens | 4% | 42% |

| | | |
|---|---|---|
| 77.00 Mbps | 23277 p/s | 31741 p/s |

Erklärung einiger Tabellenwerte:

| | |
|---|---|
| *delay[us]* | Hier wird jeweils das Minimum und Maximum der Paketlaufzeiten aller 8 Ströme (4 hin, 4 rück) dargestellt |
| *ref/xLink/**diff*** | *ref*: Referenzmessung ohne CAR<br>*xlink*: CAR ist an x Links aktiviert<br>*diff*: CPU-Erhöhung durch Einschalten von CAR (Differenz aus *xLink* und *ref*)<br>Da je nach Anzahl der Links, an denen CAR aktiviert ist, die Senderate variiert s.o., muss hier jeweils eine erneute Referenzmessung gemacht werden, um das Verhalten des Routers, mit exakt derselben Senderate, mit und ohne CAR vergleichen und daraus die CPU-Erhöhung berechnen zu können |

#### 3.2.2.1 Shaping auf 77 Mbps:

Im folgenden werden alle Tests mit 429 Byte Pakete durchgeführt.

4% Überschreitung:

| | ref/1Link/**diff** | ref/4Link/**diff** |
|---|---|---|
| main CPU [%] | 1/1/**0** | 1/1/**0** |
| Slot 0 [%] | 63/63/**0** | 63/62/**-1** |
| Slot 6 [%] | 5/5/**0** | 5/5/**0** |
| Delay [us] | 73-417/73-39639 | 75-438/73-41318 |

Es gingen keine Pakete auf dem E3 Interface, den ungeshapten STM1 Links oder auf den STM1 Rückrichtungen verloren. Der geshapte Strom wurde auf einen Wert zwischen 76.4 und 77.5 Mbps begrenzt.

42 % Überschreitung:

| | 4Link |
|---|---|
| main CPU [%] | 1 |
| Slot 0 [%] | 62-63 |
| Slot 6 [%] | 5-6 |
| Delay [us] | 74-57364 |

Es gingen keine Pakete auf dem E3 Interface oder auf den STM1 Rückrichtungen verloren.

Linerate (42370p/s = 145.41 Mbps ) auf allen 8 Flows,
Shaping auf allen 4 Links am gsr81:

| | 4Link |
|---|---|
| main CPU [%] | 1 |
| Slot 0 [%] | 62-63 |
| Slot 6 [%] | 5 |

| | |
|---|---|
| Delay [us] | 74-64783 |

Es gingen keine Pakete auf dem E3 Interface oder auf den STM1 Rückrichtungen verloren. Die STM1-Flows wurden jeweils auf Werte zwischen 76.4 Mbps und 77.5 Mbps geshapt.

Linerate (42370p/s = 145.41 Mbps ) auf allen 8 Flows,
Shaping auf 3 Links am gsr81:

| | 3Link |
|---|---|
| main CPU [%] | 1 |
| Slot 0 [%] | 62-63 |
| Slot 6 [%] | 5 |
| Delay [us] | 74-64596 |

Es gingen keine Pakete auf dem E3 Interface, dem ungeshapten vierten STM1-Link oder auf den STM1 Rückrichtungen verloren.

Ergebnis
Shaping via DTS führt an der 4fach OC3 LC zu keinem erkennbaren CPU-Anstieg. Es gingen ebenfalls keine Pakete auf den ungeshapten STM1-Links oder auf dem E3 Strom verloren. DTS führt aufgrund des Zwischenpufferns der Pakete zu einem Delayanstieg bei Paketen an geshapten Strömen von bis zu 65 ms.

#### 3.2.2.2 Einfluss von Betriebsfeatures beim Shaping

Auch hier wurde überprüft, welchen Einfluss Betriebsfeatures auf DTS haben. Die Interfaces wurden wie im *Kapitel 3.2.1.2 Einfluss von Betriebsfeatures beim Policing* konfiguriert, so dass die Tests unter derselben Testumgebung wie beim Policing durchgeführt wurden. Damit liegt auch hier die maximale verlustfreie Senderate ohne eingeschaltetem DTS aber mit aktivierten Betriebsfeatures bei etwas mehr als 100 Mbps pro Link. Die CPU-Last an der 4 fach OC3 LC liegt bei etwa 98-99%, die der Haupt- CPU bei etwa 3-6% und die des Eingangsinterfaces *Slot 6/0* bei ca. 5%.

Um zu überprüfen, ob DTS zusammen mit eingeschalteten Betriebsfeatures Einfluss auf andere Interfaces hat, wurde auch hier zusätzlich auf dem E3 Link *serial4/5* mit 32.95 Mbps gesendet und der Verkehr beobachtet und analysiert.

##### 3.2.2.2.1 Shaping an 3 Links

Wird mit einer Senderate von 100 Mbps über jeden STM1 Link gesendet und ist Shaping auf 77 Mbps auf dreien der vier Links aktiviert, so treten weder Verluste auf dem vierten ungeshapten Link, noch auf den Rückströmen noch auf dem E3 Link auf. DTS hatte keinen sichtbaren Einfluss auf die CPU des Routeprozessors, der 4fach OC3 LC oder auf die des Eingangsinterfaces *Slot 6/0*.

##### 3.2.2.2.2 Shaping an 4 Links

Wird mit einer Senderate von 100 Mbps über jeden STM1 Link gesendet und ist Shaping auf 77 Mbps auf allen vier Links aktiviert, so treten ebenfalls keine Verluste außer auf den geshapten Strömen auf. DTS hatte keinen sichtbaren Einfluss auf die CPU des Routeprozessors, der 4fach OC3 LC oder auf die des Eingangsinterfaces *Slot 6/0*.

Ergebnis:
Beim Einsatz von Betriebsfeatures kommt es durch zusätzliches Aktivieren von DTS zu keiner Verschlechterung des Durchsatzes oder zu Paketverlusten.

### 3.2.2.3 Überprüfen der Statistik-Parameter

Die verlorenen Pakete kann man als *output queue drops* an den STM1-Links sehen.
Diese stimmen mit den Zählern am Agilent (Tx-Rx) exakt überein

```
gsr81#sh int pos 0/3
 POS0/3 is up, line protocol is up
   Hardware is Packet over SONET
   Internet address is 192.129.6.1/24
   MTU 4470 bytes, BW 155000 Kbit, DLY 100 usec, rely 255/255, load 15/255
   Encapsulation PPP, crc 32, loopback not set
   Keepalive set (10 sec)
   Scramble enabled
   LCP Open
   Open: IPCP
   Last input 00:00:04, output 00:00:04, output hang never
   Last clearing of "show interface" counters 00:44:32
   Queueing strategy: random early detection (WRED)
   Output queue 0/40, 40865912 drops; input queue 0/75, 0 drops
   5 minute input rate 17995000 bits/sec, 5164 packets/sec
   5 minute output rate 9710000 bits/sec, 2761 packets/sec
      86863271 packets input, 37611573665 bytes, 0 no buffer
      Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
               0 parity
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      45997359 packets output, 19957499681 bytes, 0 underruns
      0 output errors, 0 applique, 0 interface resets
      0 output buffer failures, 0 output buffers swapped out
      0 carrier transitions
```

Die speziellen DTS-Statistiken "*sh traffic-shape pos0/0*" zeigen durch DTS verworfene Pakete jedoch nicht an. Dies wurde Cisco bereits gemeldet.

```
 gsr81#sh traffic-shape pos0/0
            Target    Sustain   Excess    Interval  Queue     Queue
I/F         Rate      bits/int  bits/int  (ms)      Length    Average
PO0/0       77000000  2464000   0         32        1150      1150
PO0/1       77000000  2464000   0         32        1185      1185
PO0/2       77000000  2464000   0         32        1166      1166
PO0/3       77000000  2464000   0         32        1055      1055
```

Der Excess Bits zählen nicht hoch, obwohl Pakete verloren gehen. Diese sind jedoch nötig, um bei Paketverlusten die Ursache (in unserem Fall höhere Senderate als erlaubte Shapingrate) bestimmen zu können. Anhand der *output queue drops* ist das nicht möglich. Diese können auch durch andere Ursachen entstehen.

Mit Hilfe des Befehls *sh traffic-shape statistics* lassen sich leider auch keine Werte über die verworfenen Pakete ermitteln. Hier wird sogar behauptet, dass DTS gar nicht konfiguriert sei.

```
gsr81(config)#int pos 0/0
```

```
gsr81(config-if)#traffic-shape rate 77000000
gsr81(config-if)#^Z
gsr81#sh traffic-shape stat
                 Acc. Queue Packets   Bytes     Packets   Bytes     Shaping
I/F              List Depth                     Delayed   Delayed   Active

gsr81#sh traffic-shape stat pos 0/0
Traffic shaping not configured on POS0/0
```

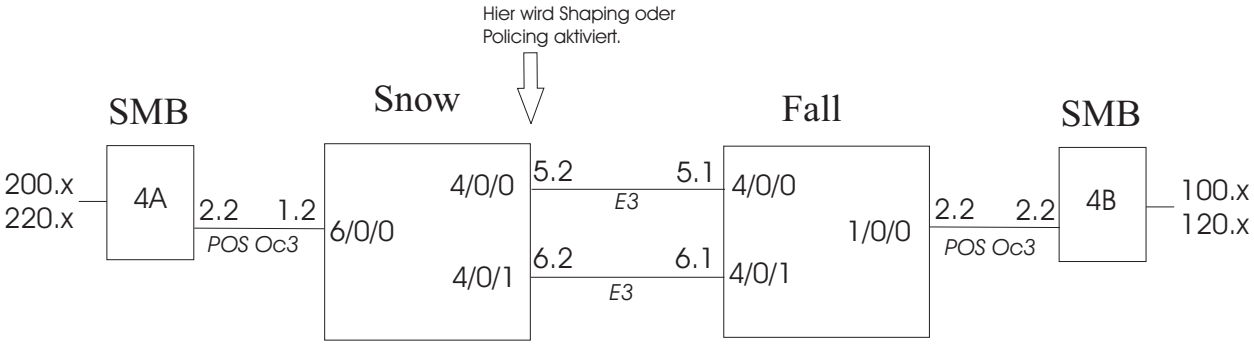
#### 3.2.2.4 Zusammenfassung

DTS auf der 4 fach STM1 Linecard begrenzt den Verkehr bei einer konfigurierten Shaping Rate von 77 Mbps zwischen 76.4 Mbps und 77.5 Mbps. Shaping via DTS führt an der 4fach OC3 LC zu keinem erkennbaren CPU-Anstieg. Es gingen ebenfalls keine Pakete auf den ungeshapten STM1-Links oder auf einem anderen Interface (hier E3-Interface) verloren. DTS führt aufgrund des Zwischenpufferns der Pakete zu einem Delayanstieg bei Paketen in geshapten Strömen von bis zu 65 ms. Auch unter Einsatz von Betriebsfeatures führt DTS zu keiner Verschlechterung des Durchsatzes oder zu Paketverlusten.
Da laut Empfehlung von Cisco bei DTS WRED aktiviert werden soll, und hier die für WRED zu konfigurierenden Parameter vom aktuellen Verkehrsverhalten abhängig sind, um optimal zu wirken, kann es sein, dass diese Parameter während des Betriebs nachoptimiert werden müssen.
Wir befragten Cisco, weshalb DTS zu keinem nennenswerten CPU-Anstieg, CAR hingegen zu einem Anstieg bis zu 51% führt. Nach längerer Diskussion bekamen wir zur Antwort, dass DTS effizienter implementiert sei als CAR.

# 4 Cisco 75xx und 2-fach E3 PA

E3 PAs befinden sich bereits im G-WiN im Einsatz. Hier soll nun untersucht werden, ob und wie sie im Flatrate-Konzept eingesetzt werden können.

Für die Tests der 2 fach E3 PAs wurde stets der folgende Testaufbau verwendet. Alle verwendeten Adressen/Subnetze lagen im Netzbereich 192.129.0./16.




Der zu testende Router (*Snow*) stellt einen AR im G-WiN dar. Die Verbindung zum SMB Port 4A entspricht der Verbindung zwischen AR und CR (Core-Router). Die beiden E3 Verbindungen zum *Fall* entsprechen jeweils einer Verbindung zu einem Anwender. Da im Labor keine Testgeräte mit E3 Schnittstellen zur Verfügung stehen, wurde ein zweiter Router (*Fall*) benötigt, um den Verkehr auf OC3 abbilden und mit dem SMB analysieren zu können.

Für jeden Test gab es 2 bzw. 4 Flows:

```
Flow1:  192.129.200.1 → 192.129.5.x → 192.129.100.1
Flow2:  192.129.100.1 → 192.129.5.x → 192.129.200.1
Flow3:  192.129.220.1 → 192.129.6.x → 192.129.120.1
Flow4:  192.129.120.1 → 192.129.6.x → 192.129.220.1
```

Shaping bzw. Policing wurde auf dem *Snow* auf 1 oder 2 seriellen Interfaces zum *Fall* für ausgehenden Verkehr wie folgt konfiguriert:

```
class-map match-all ANY
  match any
!
 policy-map shape34to4
  class ANY
    shape average 4000000
policy-map shape34to8
  class ANY
    shape average 8000000
policy-map shape34to17
  class ANY
    shape average 17000000
policy-map police34to4
  class ANY
     police 4000000 750000 1500000 conform-action transmit exceed-action
drop
policy-map police34to8
  class ANY
```

```
     police 8000000 1500000 3000000 conform-action transmit exceed-action
drop
policy-map police34to17
  class ANY
     police 17000000 3187500 6375000 conform-action transmit exceed-action
drop
```

Bei den Verkehrsströmen (`Flow1` und ggf. `Flow3`), die über geshapte bzw. gepolicte Verbindungen geschickt wurden, wurde die Paketrate variiert. Alle anderen Ströme wurden mit Linerate (1037 pps bei 4096Byte Paketen) bzw. nahezu Linerate (9620 pps (33,02 Mbps) bei 429 Byte Pakete) geschickt (im folgenden maximale Paketrate genannt).

In Absprache mit dem DFN NOC wurde IOS 12.0(21)S1 (rsp-pv-mz.120-21.S1.bin) verwendet, das zu diesem Zeitpunkt auch im G-WiN eingesetzt wurde. Für die Tests wurden zwei RSP4 mit 128 MB Route Memory und zwei VIP2-50 mit 128 MB DRAM und 8 MB SRAM vom DFN Verein ausgeliehen.

Auf dem Router war dCEF eingeschaltet und die Interfaces der E3-Verbindungen waren wie folgt konfiguriert:

```
!
interface Serial4/0/0
 ip address 192.129.5.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 encapsulation ppp
 ip route-cache distributed
 framing g751
 dsu bandwidth 34010
 crc 32
 down-when-looped
 no cdp enable
!
interface Serial4/0/1
 ip address 192.129.6.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 encapsulation ppp
 ip route-cache distributed
 framing g751
 dsu bandwidth 34010
 crc 32
 down-when-looped
 no cdp enable
!
```

## 4.1 Durchsatztest

Wir auch bei den E3 Tests am 12xxx wurde für die Bestimmung des theoretisch maximal möglichen Paketdurchsatzes folgende Formel von Cisco verwendet:

*HDLC_max_pps = 34010000 / ((IP_size + PPP_ovh_byte/pkt)*8)*
*PPP_ovh_byte/pkt für CRC32 = 9 bytes*

Daraus ergeben sich bei einer IP-Paketgröße von 429 Byte die theoretisch maximal möglichen 9706 pps, bei 4096 Byte 1037 pps .

Zunächst wurde geprüft, ob der theoretisch maximal mögliche Paketdurchsatz überhaupt erreicht werden kann. Hierfür wurden die `Flows 1` und `2` mit der maximalen Paketrate gesendet. Für die großen Pakete gab es hier keine Probleme. Bei 429 Byte Paketen hingegen gingen Pakete verloren. Aus diesem Grund wurde die Paketrate von 9706 pps auf 9620 pps reduziert, so dass keine Paketverluste mehr auftraten. Die so ermittelten Paketraten werden im folgenden als maximale Paketraten bezeichnet.

Die folgenden Tests wurden zunächst mit 4096 Byte Paketen begonnen, da hier nicht mit Performanceeinbußen wegen der Paketrate zu rechnen war. Geplant waren Tests mit 5 verschiedenen Paketgrößen. Anfängliche Tests zeigten aber, dass aus Zeitgründen nicht alle Paketraten getestet werden können. Deshalb wurden die Tests auf 429 Byte Pakete beschränkt, was auch der durchschnittlichen Paketgröße im G-WiN entspricht.

## 4.2 Shaping via MQC

### 4.2.1 Ein Strom mit Shaping

Für die Tests wurden `Flow1` (mit angepassten Paketraten) und `Flow2` (mit maximaler Paketrate) aktiviert. Es wurde untersucht, wie die CPU-Last steigt, wenn Shaping auf diesem E3-Link aktiviert wird. Die folgende Tabelle zeigt, dass - solange keine Pakete verworfen werden müssen (Überlast = 0%) - die CPU-Last durch die Aktivierung von Shaping bei großen Paketen nicht ansteigt und bei kleineren Paketen um 1 bis 6% . Wird jedoch mit 100% Überlast gesendet – d.h. statt 518 werden 1037 pps gesendet – steigt die CPU-Last auf dem VIP mit aktiviertem Shaping auf 99%. Die Last auf dem anderen VIP steigt dann ebenfalls an (5%). Die folgende Tabelle zeigt genauer, wie sich die CPU-Last „ohne Shaping/mit Shaping“ verhält.

| Paketgröße in Byte | Shapingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **4096** | -/17 | 518 | -/100 | 0/0 | 4/5 | 10/12 |
| | | 1037 | **-/200** | 0/0 | 6/7 | **13/99** |
| | -/8 | 244 | -/100 | 0/0 | 4/4 | 9/10 |
| | -/4 | 122 | -/100 | 0/0 | 3/3 | **9/52** |
| **429** | -/17 | 4851 | -/100 | 0/0 | 21/23 | 24/30 |
| | | 5336 | -/**110** | 0/0 | 21/23 | **25/99** |
| | | (9703) 9620 | -/**200** | 0/0 | 30/34 | **26/99** |
| | -/8 | 2283 | -/100 | 0/0 | 15/16 | 22/26 |
| | -/4 | 1141 | -/100 | 0/0 | 13/13 | 23/24 |
| | | 1255 | -/**110** | 0/0 | 13/13 | **20/99** |
| | | 2511 | -/**220** | 0/0 | 16/14 | **22/99** |

Ferner wurde untersucht, ob eine Verbindung zwischen der Anzahl der zu verwerfenden Pakete (Überlast) und der CPU-Belastung besteht. Wie die folgende Tabelle zeigt, steigt die CPU-Last auf dem geshapten Interface bereits bei einer Überlast von 10%.

| Paketgröße in Byte | Shapingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | 17 | 4851 | 100 | 0 | 23 | 30 |
| | 17 | 5336 | 110 | 0 | 23 | 99 |
| | 17 | 7277 | 150 | 0 | 29 | 99 |
| | 17 | (9706) 9620 | 200 | 0 | 34 | 99 |
| | 8 | 2283 | 100 | 0 | 16 | 26 |
| | 8 | 2511 | 110 | 0 | 16 | 99 |
| | 8 | 3424 | 150 | 0 | 19 | 99 |
| | 8 | 5022 | 220 | 0 | 22 | 99 |
| | 4 | 1141 | 100 | 0 | 13 | 24 |
| | 4 | 1255 | 110 | 0 | 13 | 99 |
| | 4 | 1712 | 150 | 0 | 13 | 99 |
| | 4 | 2511 | 220 | 0 | 14 | 99 |
| **4096** | 17 | 518 | 100 | 0 | 5 | 12 |
| | 17 | 570 | 110 | 0 | 5 | 99 |
| | 17 | 622 | 120 | 0 | 5 | 99 |
| | 17 | 778 | 150 | 0 | 6 | 99 |
| | 17 | 1037 | 200 | 0 | 6 | 99 |
| | 8 | 244 | 100 | 0 | 4 | 10 |
| | 8 | 268 | 110 | 0 | 5 | 99 |
| | 8 | 292 | 120 | 0 | 5 | 99 |
| | 8 | 366 | 150 | 0 | 5 | 99 |
| | 8 | 488 | 200 | 0 | 5 | 99 |
| | 8 | 537 | 220 | 0 | 5 | 99 |
| | 4 | 122 | 100 | 0 | 3 | 52 |
| | 4 | 134 | 110 | 0 | 3 | 99 |
| | 4 | 146 | 120 | 0 | 3 | 99 |
| | 4 | 183 | 150 | 0 | 4 | 99 |
| | 4 | 244 | 200 | 0 | 3 | 99 |
| | 4 | 268 | 220 | 0 | 3 | 99 |

Von Cisco bekamen wir die Aussage, dass bei einer weiteren Erhöhung der Paketrate nicht unbedingt ein weiterer Anstieg der CPU-Last zu erwarten sei und dass eine CPU-Last von 99% nicht zwangsläufig auf ein überlastetes Interface hindeutet. Dies wurde in späteren Tests mit einem zusätzlichen ungeshapten Flow und eingeschalteten Betriebsfeatures auch bestätigt.

Die folgende Tabelle zeigt, wie sich das Delay bei den vorangegangenen Messungen verhalten hat. Für 4096 Byte Pakete wurde nur geprüft, wie sich das Delay bzgl. Unterschiedlicher Überlasten verhält. Bei 429 Byte Paketen wurde auch geschaut, wie sich das Delay durch Einschalten von Shaping verändert (für Paketraten mit 0% Überlast). Steht also in der Spalte gesendete Überlast ein „-„ , dann bedeutet dies, dass die Paketrate der Rate mit 0% Überlast entspricht, aber kein Shaping aktiviert war. Z.B. wurde bei 429 B Paketen mit einer Paketrate, die 17 Mbps entspricht, gesendet. Ohne Shaping betrug das Delay für `Flow1` 436.9 us, mit Shaping 606.7 us.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 |
|---|---|---|---|---|
| 4096 | shape34to17 | 0 | 2272.7 | 2183.3 |
| | | 10 | **1375199.4** | 2287 |
| | | 20 | 2094998.4 | 2285.6 |
| | | 50 | 2095539.1 | 2276.1 |
| | | 100 | 2095967.2 | 2214 |
| | shape34to8 | 0 | 2233.8 | 2187.8 |
| | | 10 | **2951901.1** | 2245 |
| | | 20 | 4449644 | 2297.6 |
| | | 50 | 4451479.9 | 2293.4 |
| | | 100 | 4451114 | 2248.4 |
| | | 120 | 4451825.5 | 2235.7 |
| | shape34to4 | 0 | 6114.3 | 2224.8 |
| | | 10 | 4849986.3 | 2241.2 |
| | | 20 | 4850012.2 | 2219.2 |
| | | 50 | 4850074.7 | 2237 |
| | | 100 | 4850118.2 | 2318.5 |
| | | 120 | 4850075.5 | 2259.7 |
| 429 | shape34to17 | - | 536.9 | 585 |
| | | 0 | 606.7 | 616.7 |
| | | 10 | **133917.1** | 710 |
| | | 50 | 515425.4 | 593.2 |
| | | 100 | 515793.4 | 668.7 |
| | shape34to8 | - | 526.1 | 652.9 |
| | | 0 | 711.8 | 591.6 |
| | | 10 | **289077.7** | 633.3 |
| | | 50 | 515637.2 | 652 |
| | | 120 | 515912.5 | 559.6 |
| | shape34to4 | - | 532.2 | 605.7 |
| | | 0 | 708.6 | 607.4 |
| | | 10 | 514835.1 | 605.8 |
| | | 50 | 515417.6 | 577.7 |
| | | 120 | 515845.8 | 583.2 |

Da Überlasten kontinuierlich gesendet wurden, stieg das Delay stetig bis zu einem Maximum an. Dann waren die Shaping-Queues voll und es wurden entsprechend Pakete verworfen. (Die fett dargestellten Werte sind geringer als das tatsächlichen Maximum, da der Smartbits nur eine begrenzet Anzahl von Messwerten speichern kann.) Bei Messungen mit 429 Byte Paketen ohne Überlast wurde festgestellt, dass durch Einschalten von Shaping das Delay von `Flow1` um ca. 0,2 ms anstieg.

### 4.2.2 Zwei Ströme mit Shaping

Für diese Tests wurden die `Flows 1` und `3` (mit jeweils angepassten Paketraten) und die `Flows 2` und `4` (mit maximaler Paketrate) verwendet. In der folgenden Tabelle sind die Werte für CPU-Belastung „ohne Shaping/Shaping an 2 IFs“ zusammengefasst. Neben der CPU-Erhöhung auf dem VIP mit 2 Shapingqueues, ist auch ein Anstieg der CPU auf dem eingehenden IF um bis ca. 8% feststellbar.

| Paketgröße in Byte | Shapingrate in Mbps | Sendelast in pps Flow1 und 3 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| 429 | -/17 | 4851 | 100 | 0/0 | 35/37 | 47/55 |
| | -/17 | 5336 | 110 | 0/0 | 37/43 | 48/99 |
| | -/17 | (9706) 9620 | 200 | 0/0 | 53/61 | 51/99 |
| | -/4 | 1141 | 100 | 0/0 | 21/21 | 40/43 |
| | -/4 | 1255 | 110 | 0/0 | 21/23 | 40/99 |
| | -/4 | 2511 | 220 | 0/0 | 27/29 | 43/99 |

Das Delay stieg wieder an, bis die Queue voll war und entsprechend dem eingestellten Shaping Pakete verworfen wurden.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| 429 | shape34to17 | 0 | 1046.6 | 974.6 | 1146.2 | 991.2 |
| | | 10 | 515466.4 | 1100.5 | 515437.5 | 1047.6 |
| | | 100 | 516321.7 | 1073.7 | 516382.1 | 1107 |
| | shape34to4 | 0 | 826.6 | 846.6 | 805.7 | 815.9 |
| | | 10 | 515258.2 | 920.5 | 515246.7 | 965 |
| | | 120 | 560760 | 857.1 | 560760 | 963.3 |

### 4.2.3 Zwei Ströme: Shaping auf einem Strom

Für diese Tests wurden wiederum alle 4 Flows verwendet, wobei nur auf dem `Flow1` die Paketrate angepasst wurde und auch Shaping nur für diesen einen Flow aktiviert wurde.

| Paketgröße in Byte | Shapingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| 429 | -/17 | 4851 | 100 | 0/0 | 47/50 | 50/52 |
| | -/17 | 5336 | 110 | 0/0 | 47/48 | 50/99 |
| | -/17 | (9706) 9620 | 200 | 0/0 | 53/55 | 51/99 |
| | -/4 | 1141 | 100 | 0/0 | 39/42 | 47/47 |
| | -/4 | 1255 | 110 | 0/0 | 40/39 | 47/99 |
| | -/4 | 2511 | 220 | 0/0 | 41/41 | 48/99 |

Hier war besonders interessant, ob der ungeshapte `Flow3` durch die hohe CPU-Last auf dem VIP beeinflusst wurde. Selbst in Langzeittests wurden nur Pakete für den geshapten `Flow1` verworfen. Alle anderen Flows hatten trotz der CPU-Last von 99% keine Paketverluste.

Auch die Delays verhielten sich normal. Auf dem geshapten Flow werden die Pakete verzögert, aller anderen Flows wurden nicht maßgeblich beeinflusst.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| 429 | shape34to17 | - | 528.2 | 560.1 | 658.6 | 659.8 |
| | | 0 | 689.2 | 699.4 | 783.4 | 810.8 |
| | | 10 | **42710.3** | 674.3 | 794.6 | 648.5 |
| | | 100 | **273608.3** | 721.7 | 887.6 | 726.2 |
| | shape34to4 | 0 | 628.1 | 551 | 701.5 | 664.3 |
| | | 10 | **59131.3** | 749.2 | 920.8 | 814.4 |
| | | 120 | **409649.1** | 733.6 | 884.2 | 789.4 |

### 4.2.4 Aktivierung von Betriebsfeatures

Hierfür wurden die Tests aus dem vorangegangenen Abschnitt mit einem geshapten und drei ungeshapten Flows wiederholt. Zusätzlich waren auf dem Router folgende Features und 114000 statische Routen konfiguriert.

```
access-list 199 deny ip host 193.61.196.132 host 141.38.43.44
access-list 199 deny ip host 193.61.196.132 host 141.38.45.34
access-list 199 deny ip host 193.61.196.132 host 141.38.46.45
access-list 199 permit ip any any

ip flow-export 192.129.7.2 1234
ip multicast-routing distributed
router ospf 1

int ser 4/0/0
serv out shape34to17
ip access-group 199 in
ip verify unicast reverse-path
ip pim sparse-mode
ip multicast ttl-threshold 32
ip multicast boundary 18
ip route-cache flow
ip mroute-cache distributed
ip pim bsr-border


int ser 4/0/1
ip access-group 199 in
ip verify unicast reverse-pat
ip pim sparse-mode
ip multicast ttl-threshold 32
ip multicast boundary 18
ip route-cache flow
ip mroute-cache distributed
ip pim bsr-border
```

Die Test wurden sowohl mit 128 MB als auch mit 256 MB Route-Memory durchgeführt. Dabei ergaben sich für die CPU-Belastung zwischen den beiden Ausbauzuständen nur Abweichungen von +/- 1% die im Bereich der Messungenauigkeit liegen dürften. Ohne Shaping erhöhte sich die Last auf der Haupt-CPU, während sie sich am eingehenden Interface verringerte gegenüber den Messungen ohne Betriebsfeatures. Wurde zusätzlich Shaping

aktiviert, stieg die CPU-Last auf dem geshapten VIP (ohne Überlast) um ca. 10% höher als ohne aktivierte Betriebsfeatures.

| Paketgröße in Byte | Shapingrate in Mbps | Sendelast Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| 429 | -/17 | 4851 | 100 | 12/14 | 37/37 | 50/61 |
| | -/17 | 5336 | 110 | 12/15 | 38/38 | 50/99 |
| | -/17 | (9706) 9620 | 200 | 12/17 | 43/43 | 50/99 |

Auch hier gingen bei Überlast trotz der hohen CPU-Last (99%) keine Pakete von ungeshapten Flows verloren. Dies legt allerdings nahe, dass – wie von Cisco erwähnt – die CPU-Last auf den VIP Boards keine Aussage über die tatsächliche Last macht. Dies kann auch in dem Cisco-Papier „Understanding VIP CPU Running at 99% and Rx-Side Buffering“ nachgelesen werden.

Die gemessenen Delays entsprechen denen ohne eingeschaltete Betriebsfeatures.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| 429 | shape34to17 | 0 | 711.6 | 598.9 | 822.1 | 676.1 |
| | | 10 | **42661.3** | 764.4 | 771 | 826.5 |
| | | 100 | **273622.8** | 708.4 | 1014.7 | 719.2 |

### 4.2.5 Zusammenfassung

Der Einsatz von Shaping via MQC ist möglich. Die Aktivierung von Shaping allein führt wenn überhaupt dann nur geringfügig zu einer Erhöhung der CPU-Belastung auf dem VIP Boards. Sobald jedoch aufgrund von Überlast Pakete verworfen/geshapt werden müssen, steigt die CPU-Last auf dem entsprechenden Board auf 99%. Da bei den Tests für zusätzliche Ströme – auch bei Aktivierung weiterer Features – keine Paketverluste festgestellt werden konnten, war das entsprechende Board aber nicht überlastet. Bleibt die Frage: Wie können Leistungsengpässe auf den VIP Boards frühzeitig erkannt werden?
Wird an mehreren Interfaces Shaping aktiviert, steigt die CPU-Last auf dem eingehenden und ausgehenden VIP Borad zusätzlich an.
Das Delay für geshapten Verkehr steigt erwartungsgemäß an. Hier stellt sich nur die Frage, wie kann das maximale Delay verkürzt werden?

## 4.3 Policing via MQC

### 4.3.1 Ein Strom mit Policing

Für die Tests wurden `Flow1` (mit angepassten Paketraten) und `Flow2` (mit maximaler Paketrate) aktiviert.
Zunächst wurde wieder geprüft, ob die Aktivierung von Policing am ausgehenden Interface zu einer Erhöhung der CPU-Last führt. Wie die folgende Tabelle zeigt, konnten hier für 4096 Byte Pakete keine maßgeblichen Änderungen festgestellt werden. Die Einträge haben analog zum Shaping die Bedeutung „ohne Policing/mit Policing“.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Policingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **4096** | -/17 | 1037 | **-/200** | 0/0 | 6/6 | 12/12 |
| | -/17 | 518 | -/100 | 0/0 | 5/5 | 11/12 |
| | -/8 | 244 | -/100 | 0/0 | 4/3 | 10/10 |
| | -/4 | 122 | -/100 | 0/0 | 3/3 | 10/9 |

Als nächstes wurde geprüft, ob die CPU-Last in Abhängigkeit von der Policingrate steigt. Auch hier waren, wie die nachfolgende Tabelle zeigt, für 4096 Byte Pakete keine gravierenden Änderungen festzustellen.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Policingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| 4096 | 17 | 518 | 100 | 0 | 5 | 12 |
| | 17 | 570 | 110 | 0 | 5 | 12 |
| | 17 | 622 | 120 | 0 | 6 | 12 |
| | 17 | 778 | 150 | 0 | 6 | 12 |
| | 17 | 1037 | 200 | 0 | 6 | 12 |
| | 8 | 244 | 100 | 0 | 3 | 10 |
| | 8 | 268 | 110 | 0 | 3 | 10 |
| | 8 | 292 | 120 | 0 | 4 | 10 |
| | 8 | 366 | 150 | 0 | 4 | 10 |
| | 8 | 488 | 200 | 0 | 4 | 10 |
| | 8 | 537 | 220 | 0 | 5 | 10 |
| | 4 | 122 | 100 | 0 | 3 | 9 |
| | 4 | 134 | 110 | 0 | 3 | 9 |
| | 4 | 146 | 120 | 0 | 3 | 9 |
| | 4 | 183 | 150 | 0 | 3 | 9 |
| | 4 | 244 | 200 | 0 | 3 | 10 |
| | 4 | 268 | 220 | 0 | 3 | 9 |

Wie die folgende Tabelle zeigt, erhöhte sich das Delay von `Flow1` durch die Aktivierung von Policing leicht. Für den Gegenstrom (`Flow2`) kann keine direkte Beeinflussung erkannt werden.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 |
|---|---|---|---|---|
| 4096 | police34to17 | - | 2194.3 | 2181.7 |
| | | 0 | 2261.6 | 2245 |
| | | 10 | 2310.8 | 2153 |
| | | 20 | 2366.9 | 2330.9 |
| | | 50 | 2357.5 | 2233.7 |
| | | 100 | 2309 | 2192.4 |
| | police34to8 | - | 2167.2 | 2189.2 |
| | | 0 | 2298.9 | 2114.1 |
| | | 10 | 2294.2 | 2196 |
| | | 20 | 2362.7 | 2328.4 |
| | | 50 | 2270.6 | 2168.5 |
| | | 100 | 2303.3 | 2200.4 |
| | | 120 | 2279.5 | 2173.4 |
| | police34to4 | - | 2184.3 | 2195.1 |
| | | 0 | 2204.6 | 2100.8 |
| | | 10 | 2244.5 | 2224.3 |
| | | 20 | 2296.3 | 2150.9 |
| | | 50 | 2274.3 | 2158 |
| | | 100 | 2194.2 | 2175.9 |
| | | 120 | 2296 | 2154.5 |

Da andere Tests zeigten, dass das Verhalten bei kleineren Paketgrößen durchaus andere, klarere Ergebnisse liefert, wurden die Tests noch einmal mit 429 Byte Paketen durchgeführt. Wie die folgende Tabelle zeigt, stieg die CPU am eingehenden IF nur geringfügig am ausgehenden, gepolicten IF um bis zu 11 % an.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | -/17 | 4851 | 100 | 0/0 | 21/23 | 25/35 |
| | -/17 | 5336 | 110 | 0/0 | 21/24 | 25/31 |
| | -/17 | (9703) 9620 | 200 | 0/0 | 30/34 | 26/37 |
| | -/4 | 1141 | 100 | 0/0 | 13/13 | 20/24 |
| | -/4 | 1255 | 110 | 0/0 | 13/14 | 20/24 |
| | -/4 | 2511 | 220 | 0/0 | 16/17 | 22/25 |

### 4.3.2 Zwei Ströme mit Policing

Für diese Tests wurden die `Flows 1` und `3` (mit jeweils angepassten Paketraten) und die `Flows 2` und `4` (mit maximaler Paketrate) verwendet. In der folgenden Tabelle sind die Werte für CPU-Belastung „ohne Policing/Policing an 2 IFs“ zusammengefasst. Je nach Paketrate kann die CPU-Last am ausgehenden IF um bis zu 17% (429 Byte Pakete) ansteigen. Beim eingehenden VIP lag dieser Wert bei 7%.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 und 3 | Senderate im Bezug auf Policingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | -/17 | 4851 | 100 | 0 | 35/38 | 47/64 |
| | -/17 | 5336 | 110 | 0 | 37/41 | 48/55 |
| | -/17 | (9706) 9620 | 200 | 0 | 53/60 | 51/67 |
| | -/4 | 1141 | 100 | 0 | 21/21 | 40/44 |
| | -/4 | 1255 | 110 | 0 | 21/21 | 40/44 |
| | -/4 | 2511 | 220 | 0 | 27/29 | 43/47 |
| **4096** | -/17 | 1037 | **-/200** | 0/0 | 12/12 | 23/23 |
| | -/17 | 518 | -/100 | 0/0 | 9/9 | 20/23 |
| | -/8 | 244 | -/100 | 0/0 | 7/7 | 18/21 |
| | -/4 | 122 | -/100 | 0/0 | 7/6 | 18/20 |

Die folgende Tabelle zeigt, dass es beim Policing mit 4096 Byte Paketen durch Erhöhung der Überlast und damit der Steigerung der Anzahl der zu verwerfenden Pakete (Droprate) zu keiner nennenswerten Erhöhung der CPU-Last kam.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 und 3 | Senderate im Bezug auf Policingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| 4096 | 17 | 518 | 100 | 0 | 9 | 23 |
| | 17 | 570 | 110 | 0 | 9 | 22 |
| | 17 | 622 | 120 | 0 | 9 | 22 |
| | 17 | 778 | 150 | 0 | 11 | 22 |
| | 17 | 1037 | 200 | 0 | 12 | 23 |
| | 8 | 244 | 100 | 0 | 7 | 21 |
| | 8 | 268 | 110 | 0 | 7 | 20 |
| | 8 | 292 | 120 | 0 | 8 | 20 |
| | 8 | 366 | 150 | 0 | 8 | 20 |
| | 8 | 488 | 200 | 0 | 9 | 20 |
| | 8 | 537 | 220 | 0 | 9 | 22 |
| | 4 | 122 | 100 | 0 | 6 | 20 |
| | 4 | 134 | 110 | 0 | 6 | 19 |
| | 4 | 146 | 120 | 0 | 6 | 19 |
| | 4 | 183 | 150 | 0 | 6 | 19 |
| | 4 | 244 | 200 | 0 | 7 | 20 |
| | 4 | 268 | 220 | 0 | 7 | 20 |

Das Delay der Flows lag in einem normalen Bereich. Die Erhöhung des Delays durch das Aktivieren von Policing – wie es in der Messung für einen Flow festgestellt wurde – konnte hier für `Flow3` nicht durchgehend festgestellt werden. Auch scheint eine höhere Droprate keinen weiteren Einfluss auf das Delay zu haben.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| 4096 | police34to17 | - | 2169.1 | 2220.4 | 2208.6 | 2218.6 |
| | | 0 | 2285.2 | 2215.6 | 2285.1 | 2207.2 |
| | | 10 | 2261.3 | 2274.2 | 2270.7 | 2271.5 |
| | | 20 | 2294.2 | 2426.5 | 2357.5 | 2209.6 |
| | | 50 | 2355.9 | 2310.5 | 2355.9 | 2313.2 |
| | | 100 | 2257.5 | 2228.9 | 2259.2 | 2229.5 |
| | police34to8 | - | 2177.6 | 2293 | 2325.7 | 2292.9 |
| | | 0 | 2395.6 | 2193.9 | 2395.7 | 2194.3 |
| | | 10 | 2297 | 2196.8 | 2294.4 | 2180.1 |
| | | 20 | 2527.5 | 2181 | 2383.2 | 2178.9 |
| | | 50 | 2272.5 | 2195.7 | 2271.5 | 2193.7 |
| | | 100 | 2263.7 | 2207.7 | 2263 | 2205.9 |
| | | 120 | 2285.3 | 2222.1 | 2285.2 | 2225.2 |
| | police34to4 | - | 2198.9 | 2273.5 | 2199.1 | 2271.5 |
| | | 0 | 2333.3 | 2320.2 | 2333.4 | 2229.7 |
| | | 10 | 2258.4 | 2323 | 2258.5 | 2416.6 |
| | | 20 | 2357.8 | 2301.7 | 2357.1 | 2224.4 |
| | | 50 | 2333.7 | 2310.3 | 2336.3 | 2308.3 |
| | | 100 | 2314.6 | 2360.5 | 2315.2 | 2362.4 |
| | | 120 | 2215.9 | 2243.7 | 2218.3 | 2241.6 |
| 429 | police34to17 | 0 | 1371.8 | 957.6 | 1080.9 | 950.8 |
| | | 10 | 1091.7 | 922.8 | 1114 | 937.4 |
| | | 100 | 1558.9 | 1131.4 | 1545.6 | 1101 |
| | police34to4 | 0 | 1016.6 | 1089.5 | 964.6 | 1031.8 |
| | | 10 | 915.8 | 877.2 | 915.5 | 828.9 |
| | | 120 | 1015.6 | 893.8 | 1004.7 | 856.5 |

### 4.3.3 Zwei Ströme: Policing auf einem Strom

Für diese Tests wurden wiederum alle 4 Flows verwendet, wobei nur auf dem `Flow1` die Paketrate angepasst wurde und auch Policing nur für diesen einen Flow aktiviert wurde. Auch hier war ein leichter CPU-Anstieg am eingehenden und ausgehenden IF zu beobachten. Bei nicht gepolicten Flows gingen keine Pakete verloren. Die Tabelle enthält die CPU-Werte in der Form „ohne Policing/mit Policing“.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Policingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | -/17 | 4851 | 100 | 0 | 47/49 | 50/59 |
| | -/17 | 5336 | 110 | 0 | 47/51 | 50/56 |
| | -/17 | (9706) 9620 | 200 | 0 | 53/58 | 51/63 |
| | -/4 | 1141 | 100 | 0 | 39/41 | 47/50 |
| | -/4 | 1255 | 110 | 0 | 40/41 | 47/50 |
| | -/4 | 2511 | 220 | 0 | 4143 | 48/50 |

Die folgende Tabelle zeigt die gemessenen Delays bei aktiviertem Policing.

| Paketgröße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| 429 | police34to17 | 0 | 1321.2 | 1032.6 | 1218.5 | 1047.1 |
| | | 10 | 1045.3 | 944.2 | 1056.5 | 1079.6 |
| | | 100 | 1208.5 | 1044.3 | 1071.1 | 1090.1 |
| | police34to4 | 0 | 865.4 | 896.1 | 831.6 | 808.4 |
| | | 10 | 892.7 | 929.1 | 1012.3 | 985.4 |
| | | 120 | 1011.1 | 908.5 | 894.5 | 932.5 |

### 4.3.4 Aktivierung von Betriebsfeatures

Bei diesen Tests wurden die gleichen zusätzlichen Features, wie beim Shaping aktiviert. Bei den Referenzmessungen ohne Policing sah man, dass die CPU-Last auf den eingehenden Interfaces geringer war, während auf der Main-CPU nun eine Last festzustellen war. Wurde nun noch Policing am ausgehenden IF für `Flow1` aktiviert, erhöhte sich die CPU-Last für dieses VIP Board um 13-22%. Die Tabelle zeigt die Werte bei aktivierten Betriebsfeatures für Messungen „ohne Policing/mit Policing“.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Policingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| 429 | -/17 | 4851 | 100 | 12/15 | 37/36 | 50/64 |
| | -/17 | 5336 | 110 | 12/15 | 38/36 | 50/63 |
| | -/17 | (9706) 9620 | 200 | 12/21 | 43/43 | 50/72 |

Die folgende Tabelle zeigt die gemessenen Delays bei aktiviertem Policing.

| Paketgröße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| 429 | police34to17 | 0 | 1010.7 | 1009.6 | 1041.6 | 1080.5 |
| | | 10 | 960.8 | 934.4 | 1022.5 | 1048.4 |
| | | 100 | 1153.2 | 1108 | 1092.3 | 1083.6 |

### 4.3.5 Zusammenfassung

Die Konfiguration von Policing via MQC ist möglich. Bei den Tests mit 4096 Byte Paketen konnte keine Beeinflussung des Verkehrs oder der CPU-Last durch die Aktivierung von Policing festgestellt werden. Die Tests mit 429 Byte Paketen zeigten jedoch, dass die CPU-Last des ausgehenden gepolicten IF/VIP um 13-22% angestiegen ist. Bei den Tests gingen keine Pakete von Flows verloren, die nicht gepolict wurden. Wird an weiteren Interfaces Policing aktiviert steigt die CPU-Last auf gepolictem und dem eingehenden VIP weiter an. Die Delays erhöhen sich nicht maßgeblich.

## 4.4 Shaping via „traffic-shape“

Neben der Konfiguration von Shaping via MQC gibt es auch noch die Möglichkeit, Shaping per „traffic-shape rate 17000000“ zu konfigurieren. Auf dem 75xx wird dann Generic Traffic Shaping (GTS) aktiviert. Die Verkehrsüberwachung und Glättung wird dabei vom zentralen Prozessor übernommen. Beim Distributed Traffic Shaping (Shaping via MQC) hingegen kann das Shaping verteilt auf den VIP Boards abgewickelt werden. GTS ist eine veraltete Methode und sollte auf den Cisco 75xx mit VIP Boards nicht mehr eingesetzt werden. Da auf den Cisco12xxx (GSR) aber kein Shaping via MQC möglich ist wurde „traffic-shape“ hier getestet.

Shaping wurde für diese Tests wie folgt konfiguriert:

```
!
int seriell 4/0/0
   traffic-shape-rate 17000000
!
```

Eine Beobachtung der Shapingrate ist hier nicht möglich. Lediglich per `show traffic-shape ser4/0/0` oder `show traffic-shape statistic ser4/0/0` kann man Informationen über den aktuellen Zustand erhalten.

### 4.4.1 Zwei Ströme: Shaping auf einem Strom

Für diese Tests wurden wieder alle 4 Flows verwendet, wobei Shaping nur für `Flow1` aktiviert wurde und die Paketrate auch nur auf diesem Flow variiert wurde. Der Route Prozessor war mit 256 MB ausgestattet.
Die folgende Tabelle zeigt, dass das Shaping auf der zentralen CPU durchgeführt wird und die VIP Boards nur noch das Weiterleiten übernehmen müssen. Die Darstellung erfolgt wieder als „CPU-Last ohne Shaping/ mit Shaping“.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | -/17 | 4851 | 100 | 0/4 | 47/47 | 50/47 |
| | -/17 | 5336 | 110 | 0/6 | 47/47 | 50/47 |
| | -/17 | (9703) 9620 | 200 | 0/21 | 53/53 | 51/47 |

Das Delay erhöhte sich nur für Flow1 auf den das Shaping wirkte.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| **429** | police34to17 | 0 | 1155.2 | 1003.3 | 1051.2 | 1070.5 |
| | | 10 | 28845.3 | 1102 | 1071.2 | 1279.8 |
| | | 100 | 127696.1 | 1023.4 | 1055.1 | 1136.6 |

### 4.4.2 Aktivierung von Betriebsfeatures

Zusätzlich zum eben verwendeten Testaufbau wurden hier die in Absatz 4.2.4 aufgeführten Betriebsfeatures aktiviert. Es wurde ebenfalls 256 MB Route Memory verwendet.
Die folgende Tabelle zeigt, dass durch Aktivierung von „traffic-shape“ nur die Last auf der zentralen CPU steigt. Im Vergleich zu Shaping via MQC hat das VIP 4 weniger CPU-Last während die zentrale CPU das Shaping übernehmen muss und damit auch höher belastet ist.

| Paketgröße in Byte | Shapingrate in Mbps | Sendelast In pps Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | -/17 | 4851 | 100 | 12/14 | 37/37 | 50/49 |
| | -/17 | 5336 | 110 | 12/23 | 38/38 | 50/51 |
| | -/17 | (9703) 9620 | 200 | 12/39 | 43/42 | 50/52 |

Das Delay stieg für den geshapten Flow. Mussten aufgrund des Shapings Pakete verworfen werden, stieg auch das Delay des ungeshapten Gegenstromes (Flow2) auf ca. das Doppelte. In Messungen ohne Betriebsfeatures wurde dieses Verhalten nicht festgestellt.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| **429** | police34to17 | 0 | 878.9 | 1018.9 | 1105 | 1079.5 |
| | | 10 | 37596.2 | 1008.4 | 2240.7 | 964.8 |
| | | 100 | 30376.7 | 1209.1 | 2387.3 | 1186.7 |

### 4.4.3 Zusammenfassung

Durch die Konfiguration von „traffic-shape“ konnte Generic Traffic Shaping (GTS) aktiviert werden. Statistiken über z.B. Drop- oder Durchsatzraten werden nicht unterstützt. Das Feature wird auf der zentralen CPU ausgeführt und erhöht dadurch deren Last. Die VIP Boards Übernehmen durch die Aktivierung von GTS keine weiteren Aufgaben. Ihre Last wird also dadurch auch nicht erhöht. **Achtung:** hier wurde nur Shaping für ein Interface getestet; werden mehrere Interfaces geshapt, ist eine Überlastung der CPU nicht auszuschließen. Die Konfiguration von Shaping via MQC (Kapitel 4.2) sollte der Konfiguration von GTS via „traffic-shape“ vorgezogen werden.

## 4.5 Policing via „rate-limit“

Auch dieses Feature wurde auf dem 75xx getestet, da Policing via MQC auf dem 12xxx nicht möglich war. „rate-limit“ oder CAR (Commited Access Rate) verwendet nur einen Bucket für Token von Bc (Normal Burst Size) und Be (Excess Burst Size). Class Based Policing (Policing via MQC) verwendet hingegen getrennte Buckets für Bc und Be.

Policing wurde für diese Test wie folgt konfiguriert:

```
!
int ser4/0/0
   rate-limit output 17000000 3187500 6375000 conform-action transmit
exceed-action drop
!
```

### 4.5.1 Zwei Ströme: Policing auf einem Strom

Für diese Tests wurden wieder alle 4 Flows verwendet, wobei Shaping nur für `Flow1` aktiviert wurde und die Paketrate auch nur auf diesem Flow variiert wurde. Der Route Prozessor war mit 256 MB ausgestattet.
Die folgende Tabelle zeigt, dass die CPU-Last auf dem gepolicten IF steigt, während sie auf dem anderen nahezu unbeeinflusst bleibt. Dir zentrale CPU wurde nicht belastet.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in us Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | -/17 | 4851 | 100 | 0/0 | 47/47 | 50/55 |
| | -/17 | 5336 | 110 | 0/0 | 47/48 | 47/51 |
| | -/17 | (9703) 9620 | 200 | 0/0 | 53/54 | 51/58 |

Das Delay verhielt sich wie folgt.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| **429** | police34to17 | 0 | 1016.9 | 1135.8 | 1063 | 1224.2 |
| | | 10 | 1022.7 | 1219.4 | 1118.5 | 1100.5 |
| | | 100 | 1302.4 | 1162.5 | 1153.1 | 1225.7 |

### 4.5.2 Aktivierung von Betriebsfeatures

Zusätzlich zum eben verwendeten Testaufbau wurden wieder die in Absatz 4.2.4 aufgeführten Betriebsfeatures aktiviert. Es wurde ebenfalls 256 MB Route Memory verwendet.
Wie die folgende Tabelle zeigt, erhöht sich die CPU-Last auf dem gepolicten IF um bis zu 23 %. Die Last auf der zentralen CPU steigt ebenfalls geringfügig an. Im Vergleich zu Policing via MQC ist die Last auf dem gepolicten VIP hier bis zu 10% höher.

| Paketgröße in Byte | Policingrate in Mbps | Sendelast in pps Flow1 | Senderate im Bezug auf Shapingrate in % | CPU-Last in % zentral | pos6 | ser4 |
|---|---|---|---|---|---|---|
| **429** | -/17 | 4851 | 100 | 12/14 | 37/37 | 50/65 |
| | -/17 | 5336 | 110 | 12/13 | 38/37 | 50/66 |
| | -/17 | (9703) 9620 | 200 | 12/15 | 43/43 | 50/73 |

Das Delay verhielt sich wie folgt.

| Paketgöße in Byte | Testreihe | gesendete Überlast in % | max. Delay in us Flow1 | max. Delay in us Flow2 | max. Delay in us Flow3 | max. Delay in us Flow4 |
|---|---|---|---|---|---|---|
| **429** | police34to17 | 0 | 1019 | 999.8 | 1121.8 | 1116.3 |
| | | 10 | 1141.4 | 1022.7 | 1164.3 | 1042.7 |
| | | 100 | 1072.8 | 1060.1 | 1120.5 | 1082.6 |

### 4.5.3 Zusammenfassung

CAR ist ein überholtes Feature für die Bandbreitenregulierung eines Flows, Ports oder Links. Auf der Plattform der Cisco75xx mit VIP Boards wurde es durch den Class-Based Policer (Policing via MQC) abgelöst. Die Messungen mit CAR ergaben eine CPU-Last-Erhöhung auf dem gepolicten IF. Mit Betriebsfeatures betrug diese bis zu 23 %; auch die zentrale CPU stieg unter diesen Bedingungen durch die Aktivierung von CAR geringfügig an. Beim Delay konnte keine Beeinflussung festgestellt werden.

# 5 Gesamtergebnis

Im Vorfeld der Tests zeigte sich, dass nicht auf jeder der untersuchten Interfacekarten sowohl Policing als auch Shaping möglich ist. Die Verwendung des *Modular Quality of Service Command Line Interfaces* (*MQC*) ist bei den untersuchten Karten sogar nur auf dem 75xx zugängig.

Auf dem 75xx wurde der 2 fach E3 Port Adapter untersucht. Die Tests zeigten, dass sowohl Policing als auch Shaping via *MQC* eingesetzt werden können.
Ohne Überlast steigt die CPU-Last durch Aktivierung von Shaping für ein Interface (IF) sowohl auf dem eingehenden (2%) als auch auf dem ausgehenden geshapten (6%) VIP Board an. Wird kontinuierlich mit Überlast gesendet steigt die CPU-Last auf dem geshapten Board auf 99%. Nach Aussagen von Cisco und wie auch weitere Tests zeigten, bedeutet dies jedoch nicht, dass das Board überlastet ist. So wurde ein zusätzlicher ungeshapter Strom über das belastete Board nicht beeinflusst. Auch nach der Aktivierung verschiedener Betriebsfeatures gingen für ungeshapte Ströme keine Pakete verloren. In Überlastsituationen steigt das Delay der geshapten Ströme an.
Ähnlich wie beim Shaping führt die Aktivierung von Policing auf einem IF ohne Überlast zu einer geringfügigen Erhöhung der CPU-Last auf dem ausgehenden (7%) und eingehenden (3%) VIP Board. Bei Überlast erhöht sich die CPU-Last am ausgehenden (12%) Board noch stärker. Bei zusätzlich aktivierten Betriebsfeatures stieg sie um bis zu 22% gegenüber der Last ohne Policing. Bei den Tests gingen keine Pakete von ungepolicten Strömen verloren. Die Delays änderten sich durch Policing nicht maßgeblich.
Die Konfiguration von Shaping via „traffic-shape rate“ und Policing via „rate-limit“ ist möglich. Diese Techniken sind aber überholt, belasten die zentrale CPU oder bieten geringere Funktionalität und sollten besser via MQC realisiert werden.

Am GSR wurden die 6 fach E3 LC und die 4 fach OC3 LC (beide Engine 0) untersucht.
Auf der 6 fach E3 LC ist nur Policing via CAR möglich. Traffic Shaping wird nicht unterstützt. CAR begrenzt den Verkehr gut jeweils knapp unter der eingestellten Policingrate. CAR führt zu einer CPU-Erhöhung von 8 bis 22% auf der 6 fach E3 LC. Es kommt zu keinen Paketverlusten an ungepolicten Links bzw. auf anderen Interfaces (hier ein STM1 Link), auch dann nicht, wenn die vom DFN-NOC vorgeschlagenen Betriebsfeatures aktiviert werden.
Auf der 4 fach OC3 LC ist sowohl Policing via CAR als auch Shaping via „traffic-shape rate“ möglich. Ohne aktivierte Betriebsfeatures können bei 429 B Paketen bis zu 145 Mbps über jeden Link duplex gesendet werden. Werden die vom DFN-NOC vorgeschlagenen Betriebsfeatures aktiviert, so reduziert sich der Durchsatz je Link auf knapp über 100 Mbps, ohne dass Policing oder Traffic Shaping bereits aktiviert wurde. Hier stößt die LC an ihre Grenzen. Wird mit mehr Verkehr gesendet, so kommt es zu Paketverlusten in Form von *ignored packets* am Eingangsinterface der 4 fach OC3 LC. Wird zusätzlich CAR aktiviert, so kann es selbst bei einer niedrigen Senderate (knapp über 77 Mbps) zu Paketverlusten auf ungepolicten Links der 4 fach OC3 LC kommen, sobald der Router Pakete aufgrund des Policings verwerfen muss. CAR führte in unseren Tests zu einem CPU-Anstieg von bis zu 51%. Lediglich ohne Accesslisten ist ein verlustfreies Senden bis zu einer maximalen Senderate von 90 Mbps (wenn CAR an 4 Links aktiv ist) möglich. Da in einer echten Betriebsumgebung jedoch noch mehr Leistung vom Router abverlangt wird (dynamisches Routing, wechselnde Flows, Multicastverkehr) als es in unserer Laborumgebung mit

aktivierten Betriebsfeatures der Fall ist, kann es noch zu einer Verschlechterung des Ergebnisses kommen.
Traffic Shaping auf der 4 fach OC3 LC führt hingegen zu keinem sichtbaren CPU-Anstieg. Dort konnte, der durch die Betriebsfeatures reduzierte Durchsatz von knapp über 100 Mbps, trotz zusätzlichen Aktivieren von Shaping gesendet werden, ohne dass es zu Paketverlusten auf ungeshapten Links oder anderen Interfaces kam (hier ein E3 Link). Wir befragten Cisco, weshalb DTS zu keinem nennenswerten CPU-Anstieg, CAR hingegen zu einem Anstieg bis zu 51% führt. Nach längerer Diskussion bekamen wir zur Antwort, dass DTS effizienter implementiert sei als CAR.

# Testergebnisse 4OC3X/POS-LR-LC-B

**Cisco 12000 Series Four-Port OC-3c/STM-1c POS/SDH ISE Line Card, Long Reach**

(Version 1.0)

(G-WiN-Labor)

G-WiN-Labor
Regionales Rechenzentrum Erlangen (RRZE)
Martensstr. 1
91058 Erlangen
e-Mail: g-lab@rrze.uni-erlangen.de

18. Oktober 2004

# 1 Vorbemerkung

Die zu testenden Linecards vom Typ Cisco 4OC3X/POS-LR-LC-B (ISE), Engine 3 sollen im Egde-Bereich im G-WiN zum Einsatz kommen. An dieser Stelle im Netz ist es wünschenswert, differenzierte Vertragsmodelle mit dem Kunden aushandeln zu können. Um diese Vereinbarungen einzuhalten, muß das zu testende Equipment verschiedene Eigenschaften mitbringen, wie beispielsweise die Möglichkeit, Verkehr zu begrenzen oder Verkehrsklassen zu unterscheiden und zu behandeln.

Getestet wurde die Linecard 4OC3X/POS-LR-LC-B (ISE), Engine 3. Cisco empfiehlt für diese Karte mindestens das IOS Release 12.0(22)S oder höher. Eingesetzt wurde in diesen Tests die IOS-Version „IOS (tm) GS Software (C12KPRP-P-M), Version 12.0(24)S, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)“, wobei das Image c12kprp-p-mz.120-24.S.bin verwendet wurde.

Die getestete Linecards des Typs 4OC3X/POS-LR-LC-B hatten die Hardware Revisions Nummer 73-8090-01 rev. A0.

Tabelle 1 gibt einen Überblick über die verschiedenen Bauversionen der von Cisco angebotenen OC3-Karten. Werksmäßig werden die Karten mit 2*128 MB Routenspeicher ausgeliefert, ein Upgrade auf 2*256 MB DIMM ist möglich. Als Steckertyp wird bei den Karten:

- 4OC3X/POS-IR-LC-B
- 4OC3X/POS-LR-LC-B
- 8OC3X/POS-IR-LC-B
- 16OC3X/POS-I-LC-B

ein simplex oder duplex LC-Stecker benötigt.

*Tabelle 1: Varianten der OC3-Karten*

| Kartentyp | Produkt Code | Portanzahl | Reichweite[1] | Optik[2] | Stecker |
|---|---|---|---|---|---|
| 4OC3X/POS | 4OC3X/POS-IR-LC-B= | 4 | IR | SM | LC |
| | 4OC3X/POS-MM-MJ-B= | 4 | SR | MM | MTRJ |
| | 4OC3X/POS-LR-LC-B= | 4 | LR | SM | LC |

| | | | | | |
|---|---|---|---|---|---|
| 8OC3X/POS | 8OC3X/POS-IR-LC-B= | 8 | IR | SM | LC |
| | 8OC3X/POS-MM-MJ-B= | 8 | SR | MM | MTRJ |
| 16OC3X/POS | 16OC3X/POS-I-LC-B | 16 | IR | SM | LC |
| | 16OC3X/POS-M-MJ-B= | 16 | SR | MM | MTRJ |

1Intermediate-reach (IR), Short-reach (SR), Long-reach (LR)

2Single-Mode (SM), Multi-Mode (MM)

Die Ausgabe der Versionsnummern des getesteten Routers (12410) zeigt folgendes:

```
c4101#show version
Cisco Internetwork Operating System Software
IOS (tm) GS Software (C12KPRP-P-M), Version 12.0(24)S, EARLY DEPLOYMENT
RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Fri 17-Jan-03 18:12 by nmasa
Image text-base: 0x00010000, data-base: 0x031BD000

ROM: System Bootstrap, Version 12.0(20020627:181338) [rarcher-CSCdx94605 5],
DEVELOPMENT SOFTWARE
BOOTLDR: GS Software (C12KPRP-BOOT-M), Version 12.0(21.4)S3, EARLY
DEPLOYMENT MAINTENANCE INTERIM SOFTWARE

c4101 uptime is 1 hour, 57 minutes
System returned to ROM by reload
System image file is "tftp://131.188.81.54/c12kprp-p-mz.120-24.S.bin"

cisco 12410/PRP (MPC7450) processor (revision 0x00) with 524288K bytes of memory.
MPC7450 CPU at 665Mhz, Rev 2.1, 256KB L2, 2048KB L3 Cache
Last reset from power-on
```

# 2 Testergebnisse

## 2.1 Testaufbau

Der verwendete Testaufbau war, wenn im Text nicht anderweitig beschrieben, bei allen Tests identisch und ist in Abbildung 1 dargestellt. Zur Verfügung standen zwei Router der Marke Cisco der 1200er Baureihe. Hierbei handelt es sich um einen Cisco 12416 und um einen Cisco 12410.

Die für die Tests relevante Bestückung mit Linecards war auf beiden Routern identisch, jeweils eine OC48-Linecard, eine OC192-Linecard und jeweils eine Testkarte (4-fach OC3 Engine 3) waren eingebaut.

Die Ports der Testkarten wurden mit verschiedenen betriebsrelevanten Parametern konfiguriert, die während der gesamten Tests unverändert blieben.

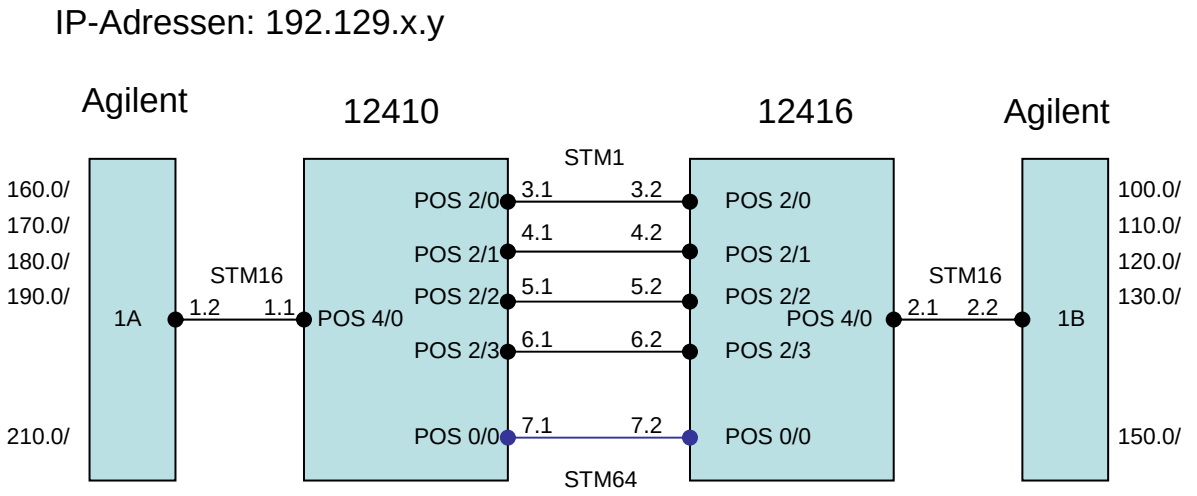



*Abbildung 1: Testaufbau*

Cisco stellte einen mit dem NOC Stuttgart und dem G-WiN-Labor abgesprochenen, detaillierten Testplan (siehe Anhang) zur Verfügung. Es stellte sich jedoch während der Laufzeit heraus, dass nicht alle Tests in der im Plan beschriebenen Art und Weise durchgeführt werden konnten. Beispielsweise ist der Befehl „set prec-continue“ nicht wie im Testplan beschrieben anwendbar (siehe Policing-Tests, Kap. 2.3.5). Dadurch traten einige erklärungsintensive Schwierigkeiten auf, die den Testablauf verzögerten. Auch der Umfang der Tests war für den Zeitraum, in dem alle notwendige Hardware zur Verfügung stand, zu groß.

## 2.2 Baseline Performance Test 4OC3X/POS-LR-LC-B

### 2.2.1 Durchsatztests, Testplan Nr. 1A

Zur Bestimmung des maximal möglichen Durchsatzes der ISE OC-3 Karte wurde zunächst die in Abbildung 2 dargestellte Konfiguration verwendet. Port 2/0 und Port 2/1 waren wie folgt identisch konfiguriert:

```
interface POS2/0
 ip address 192.129.3.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast source reachable-via any
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
```

```
 ip pim sparse-mode
 ip multicast boundary 18
 encapsulation ppp
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable
!
interface POS2/1
 ip address 192.129.4.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast source reachable-via any
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 18
 encapsulation ppp
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable
```

Zwei Agilent OC3-Module wurden verwendet, um bidirektionaler UDP-Verkehr mit verschiedenen Paketgrößen zu senden. Die Verwendung von TCP-Verkehr macht bei diesem Test und bei den anderen Tests keinen Unterschied, da der Agilent RouterTester über keinen vollständigen TCP-Stack verfügt und deshalb keine Flußkontrolle stattfindet.
Gemessen wurde die maximale Paketrate, bei der über einen Zeitraum von 10 Minuten kein Paketverlust feststellbar ist und kein Anstieg der Latencywerte zu verzeichnen ist.

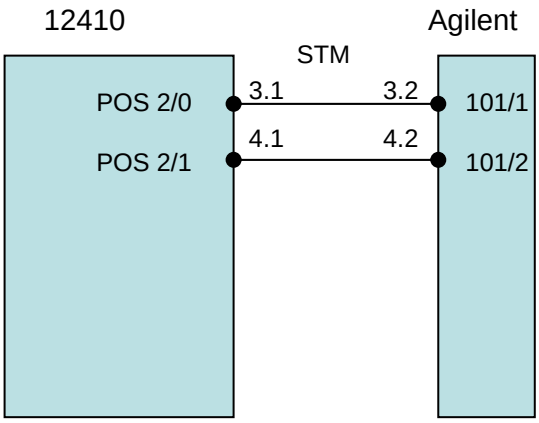



*Abbildung 2: vereinfachter Testaufbau für Durchsatztests*

In Tabelle 2 sind die ermittelten Durchsatz- und Latencywerte für verschiedene Paketgrößen dargestellt. Bei allen hier aufgezeigten Paketgrößen kann mit nahezu 100% der maximal möglichen Paketrate gesendet werden, ohne das dabei Paketverluste auftreten.

*Tabelle 2: Durchsatz und Latency bei bidirektionalem Verkehr*

| Paketlänge (Byte) | Senderate (Pakete/s) | Max. Empfangsrate (Pakete/s) | Max. Empfangsrate (Mbit/s) | Latency Min (µs) | Latency Avg (µs) | Latency Max (µs) |
|---|---|---|---|---|---|---|
| | | | | | | |
| 40 | 382041 | 382040 | 122.25 | 27 | 119 | 205 |
| 100 | 171743 | 171743 | 137.39 | 32 | 125 | 212 |
| 429 | 42740 | 42739 | 146.68 | 52 | 148 | 233 |
| 1500 | 12406 | 12405 | 148.86 | 115 | 212 | 299 |
| 4096 | 4560 | 4560 | 149.42 | 273 | 286 | 302 |

Der Verlauf der maximal möglichen Senderate in Abhängigkeit von der Paketgröße zeigt Abbildung 3. Aufgrund des höheren Overheadanteils ist die maximal mögliche Nutzdatenrate für kleine Paketgrößen geringer. Der Vergleich zwischen der theoretisch möglichen Senderate mit der tatsächlich paketverlustfrei gesendeten Rate ergibt jedoch, dass für alle Paketgrößen nahezu 100% Durchsatz erreicht wird.

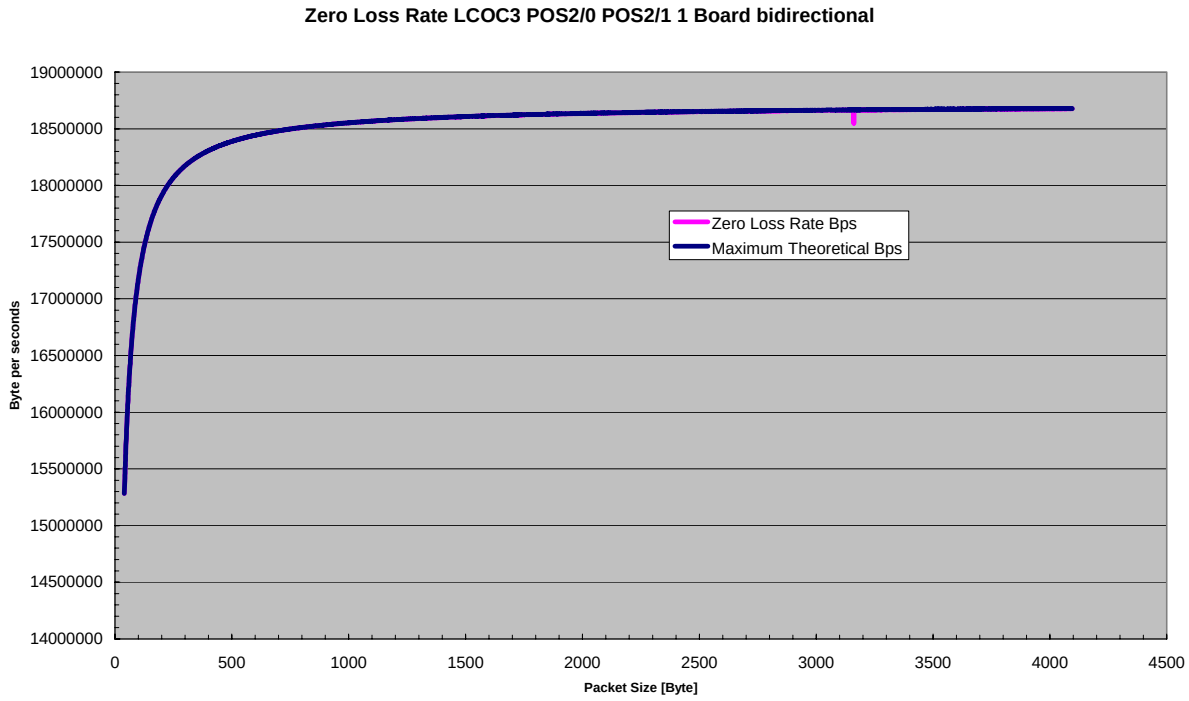



*Abbildung 3: maximale Senderate in Abhängigkeit der Paketgröße*

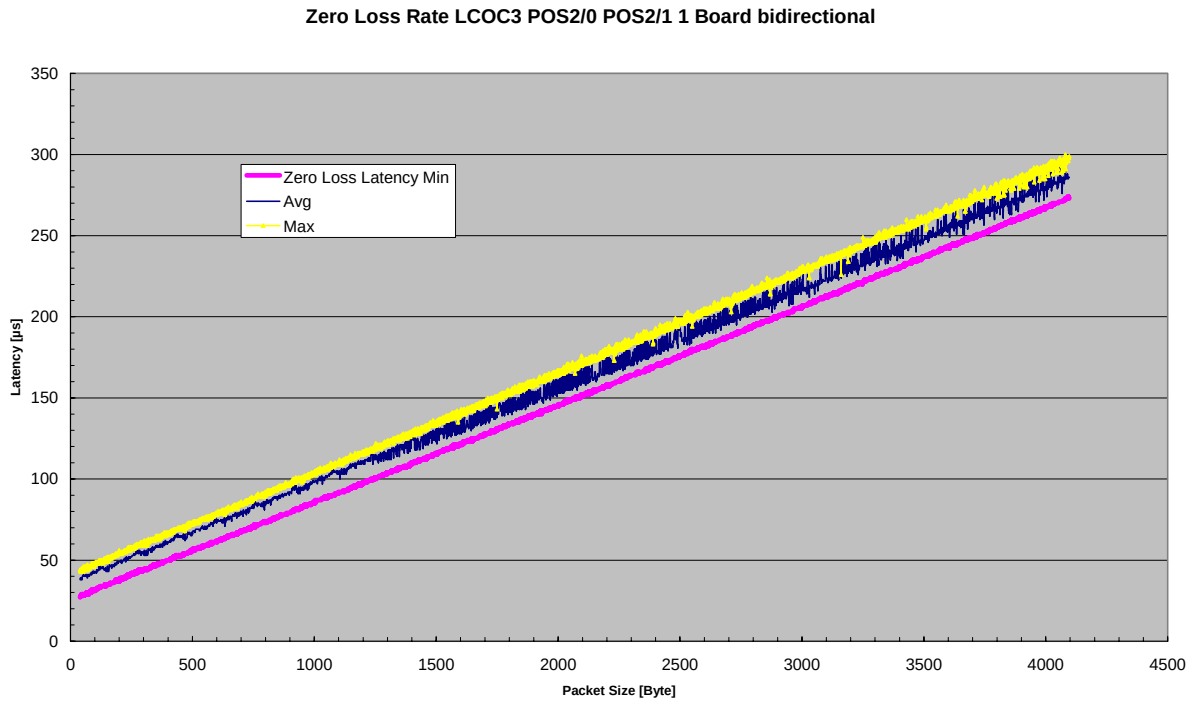



*Abbildung 4: Latency in Abhängigkeit der Paketgröße*

In Abbildung 4 ist die Abhängigkeit der maximalen, minimalen und durchschnittlichen Latencywerte von der Paketgröße dargestellt. Es ergibt sich ein linearer Zusammenhang mit geringer Streuung.

Zur Kontrolle wurden diese Tests stichpunktartig an dem in Abbildung 1 beschriebenen Testaufbau wiederholt. Dabei wurde TCP Verkehr über alle 4 Ports bidirektional gesendet. Es zeigten sich dabei keine signifikanten Unterschiede. Auch die Verwendung eines Verkehrsprofils bestehend aus verschiedenen Paketgrößen zeigte keine Unterschiede.

Die im Testplan vorgesehenen Performancetests bei gleichzeitig eingeschalteten Access-Listen bzw. Filterlisten oder gleichzeitig aktiviertem netflow-Accounting (Test 1B) wurden an dieser Stelle nicht durchgeführt, da z.B. bei den Policing-Tests (Kapitel 2.3.5.4) keine Performanceeinbußen festgestellt wurden, so dass davon ausgegangen werden kann, dass hier auch keine Beinflussung vorliegt. Wie im weiteren auch zu sehen sein wird, wurden in keinem Test Performanceeinbußen durch z.B. das Hinzufügen von Betriebsfeatures oder Accesslisten beobachtet.

### 2.2.2 Funktionalitätstest Shaping, Testplan Nr. 1C

Dieser Test soll zeigen, dass mehrere Klassen gleichzeitig auf unterschiedliche Werte mittels Shaping limitiert werden können. Die Klassen sollen dabei anhand ihrer Precedence Bits voneinander unterschieden werden. Hierfür wird auf dem c4101 eine *service-policy* mit drei Klassen konfiguriert: Die Klasse Expedited Forwarding entspricht allen Paketen mit Precedence Bit gleich fünf, die Klasse Gold entspricht allen Paketen mit Precedence Bit gleich zwei und zur Best Effort Klasse gehören alle Pakete mit Precedence gleich null. Da Shaping-Raten nur als ein Vielfaches von 64 kbps konfiguriert werden können, müssen die im Testplan angegeben Raten von 2, 20, und 7.9 Mbps auf 1.984, 19.968 und 7.872 Mbps angepasst werden. Um zu vermeiden, dass eine einzelne Klasse den gesamten Paketspeicher der LC in Anspruch nimmt, muss der Paketspeicher für jede Klasse limitiert werden. Der für diesen Zweck im Testplan angeführte Befehl *shape max-buffers* funktioniert nicht für die ISE LC. Hier ist das Kommando *queue-limit* zu verwenden. Hier kann man die maximale Anzahl von Paketen angeben, die eine Klasse im gesamten Paketbuffer zwischenspeichern darf. Da die Einheit Pakete und keine festen Bytes sind, ist diese Größe stark von der Paketgröße der ankommenden Pakete abhängig. Je kleiner die durchschnittliche Paketgröße ist, desto weniger Bytes lassen sich demzufolge zwischenspeichern. Die Einheit in Bytes für den Befehl *queue-limit* ist nach Aussage von Cisco derzeit nicht vorgesehen.
In einem ersten Test wurde die maximale Pufferlänge für jede Klasse auf 20000 Pakete festgelegt:

```
policy-map shape155to2M_20M_7.9M
  class prec_5
    shape average 1984000
    queue-limit 20000 packets
  class prec_0
    shape average 7872000
    queue-limit 20000 packets
  class prec_2
    shape average 19968000
    queue-limit 20000 packets
```

Gesendet wurden über alle 4 Ports jeweils 3 Ströme (UDP, 429 B) in Richtung 12410->12416

```
a) prec0, Senderate 14200 Pps (=48.73 Mbps)
b) prec2, Senderate 14200 Pps
c) prec5, Senderate 14200 Pps
```

und in Richtung 12416->12410 jeweils 1Strom (429 B) auf allen 4 Ports:

```
a) prec0, Senderate 42600 Pps
```

Jede Klasse bekam nur einen Bruchteil ihrer erlaubten Shapingrate durch. Cisco vermutete, dass dies am Byte-Stuffing liegen könnte. Da jedoch nicht mit voller Linerate gesendet wurde und das Problem sogar auch dann besteht, wenn die Senderate halbiert wird, kann es nicht daran liegen. Wird *queue-limit* von 20000 auf 200 Pakete reduziert, so erzielen alle Klassen auf allen 4 Links das gewünschte Ergebnis:

| | Durchsatz [Mbps] | Durchschn. Delay [ms] |
|---|---|---|
| Best-Effort | 7.87 | 87.2 |
| Gold | 19.97 | 34.4 |
| Expedited Forwarding | 1.98 | 346.0 |

Sowohl die Haupt-CPU als auch die CPU der LC (slot 2 und slot 4) bleibt konstant bei 0%. Scheinbar war das *queue-limit* zu hoch konfiguriert. Da der gesamte zur Verfügung stehende Paketspeicher bei 256 MB liegt, vermuteten wir, dass die ursprünglich konfigurierten 20000 Pakete pro Klasse auch funktionieren sollten, da diese zusammen nur einen Speicherplatz von 102.96 MB benötigen (4 Links * 3 Klassen * queue-limit Größe 20000 * Paketgröße 429 B). Kurz vor Ende der Tests, bekamen wir von Cisco die Information, dass der Paketpuffer in verschiedene Bereiche aufgeteilt wird, die nur von bestimmten Paketgrößen verwendet werden dürfen. Mit dem Befehl *exec slot 2 sh controller frfab|tofab queue* kann man sich die vorkonfigurierten Bereiche ansehen. In unserem Fall, würden alle Pakete der Größe 80 B bis 608 B nur 31.98 % vom gesamten Paketpuffer in Anspruch nehmen dürfen. Somit würde ein *queue-limit* von 20000 429 B großen Paketen pro Klasse mehr Speicherplatz einnehmen, als insgesamt zur Verfügung stünde. Dies könnte zu den deutlich zu geringen Durchsätzen geführt haben.

### 2.2.3 Zusammenfassung

Bei den Durchsatztest wurden keine signifikanten Probleme entdeckt. Der Durchsatz der Karte liegt im Bereich von 100 % des maximal möglichen Wertes, eine Erhöhung der CPU-Last auf der Karte und auf der Backplane konnte nicht festgestellt werden. Funktionalitätstests mit Shaping zeigen, dass die eingestellten Queue-Limits entscheidenden Einfluss auf das Verhalten haben. Verschiedene Paketgrößen werden verschiedenen (einstellbaren) Pufferbereichen zugeordnet, dadurch entsteht eine Abhängigkeit des Shaping Verhaltens von der tatsächlichen Paketgrößenverteilung. Beispielsweise kann das Queue-Limit nur in Anzahl von Paketen angegeben werden, was zu einer ungewünschten Paketgrößenabhängigkeit der zu limitierenden Bandbreite führt. Werden die Queue-Limits zu hoch gewählt, kann es passieren, dass manche Links keine Mindestgröße des Puffers mehr zugeteilt bekommen, so dass es zu unbestimmten Durchsatzeinbußen kommen kann. Dies gilt es bei der Einstellung der Shapingparameter zu berücksichtigen.

## 2.3 Rate-Limiting Using CAR

### 2.3.1 Test 1.2.2.1

Werden verschiedene *rate-limits* mit *extended Accesslisten* oder speziellen *rate-limit Accesslisten* konfiguriert und abgespeichert, so ist die Konfiguration auch nach einem *reboot* des Routers noch vorhanden.

### 2.3.2 Test 1.2.2.2

Folgende Accesslisten wurden getestet:

1. extended ACL auf der Ausgangsseite
2. rate-limit ACL mit Precedence Bits: am Ein- und Ausgang

Nur Verkehr, der den genannten ACL entspricht, wurde mittels Policing limitiert. Verkehr der nicht in eine ACL hineinfällt wurde verlustfrei weitergeleitet. Somit funktioniert Policing mit ACL korrekt.

Wird eine *CAR Policy* am Eingangsinterface konfiguriert, die die QoS Group für einen bestimmten Datenstrom (hier alle Pakete mit Precedence Bit gleich 2) auf einem bestimmten Wert setzt (hier 10), und wird eine CAR Policy basierend auf dieser QoS Group am Ausgangsinterface konfiguriert, so wird korrekt nur der Strom limitiert, für den am Eingangsinterface die QoS Group gesetzt wurde.

### 2.3.3 Test 1.2.2.3

Auf dem Link POS2/0 ist Policing auf 76.992 Mbps aktiviert.

```
interface POS2/0
rate-limit output 76992000 14437500 16777215 conform-action transmit exceed-action drop
```

Zunächst soll überprüft werden, ob Ströme ihren Verkehr verlustfrei durchbekommen, solange sie unter ihrem konfigurierten Policing Wert senden. Hierzu werden 3 Flows mit je 25 Mbps über POS2/0 gesendet. Die gesamte Senderate liegt somit knapp unter dem konfigurierten Policing Wert. Es kommt zu keinen Verlusten. Somit arbeitet Policing wie erwartet und lässt konformen Verkehr fehlerfrei durch.
Senden die 3 Flows (a, b, c) mit 20, 20 und 40 Mbps, so wird der Gesamtverkehr aufgrund des Policings korrekt auf 76.98 reduziert. Die Anzahl der *conformed packets* stimmen mit den am Agilent RouterTester ermittelten Werten bis auf ca. 20 Pakete überein. Die Anzahl der *exceeded packets* sind identisch mit den Werten des Agilent routerTesters (*sh int pos2/0 rate-limit*):

| Flows | Durchsatz [Mbps] | Delay [us] |
|---|---|---|
| a | 19.146458 | ca. 92.16 |
| b | 19.129072 | ca. 110.5 |
| c | 38.713313 | ca. 71.7 |

Der Verkehr der einzelnen Flows wird dabei nicht ganz gleichmässig aufgeteilt. Je mehr Pakete verworfen werden müssen, desto ungleichmäßiger wird der Verkehr auf mehrere Flows aufgeteilt (siehe auch Abschnitt 2.3.5.2).

### 2.3.4 Test 1.2.2.4

#### 2.3.4.1 Test A

Mit diesem Test soll der Einfluss konfigurierter Burst-Parameter einer CAR Policy auf den Verkehr gezeigt werden. Hierzu wird burstartiger Verkehr mit dem Routertester erzeugt. Der Agilent RouterTester stellt dabei die Parameter *Intended Load*, *Burst Load* und *Burst Length* zur Verfügung. *Intended Load* ist die durchschnittliche Senderate des Stroms, *Burst Load* gibt die Senderate an, mit der der Burst geschickt werden soll. Die *Burst Length* bestimmt die Anzahl der Pakete, die während eines Bursts mit der Geschwindigkeit *Burst Load* verschickt werden. Auf 3 Links der QOC3 LC werden in ausgehender Richtung jeweils eine CAR Policy mit gleicher *Average rate*, aber unterschiedlichen Burst Parametern (T1 und T2) konfiguriert.

```
interface POS2/0
 rate-limit output 76992000 38496 50000 conform-action transmit
exceed-action
drop
!
interface POS2/1
 rate-limit output 76992000 38496 200000 conform-action transmit
exceed-action
drop
!
interface POS2/2
rate-limit output 76992000 50000 200000 conform-action transmit
exceed-action
drop
```

Über jeden dieser drei Links wird ein aus 429 B großen UDP-Paketen bestehender burstartiger Strom gesendet. Die Konfiguration für die drei Ströme ist identisch:

*Intended Load = 76.98871*
*Burst Load = 146.2 (nahezu Linerate)*
*Burst Length = 200 Pakete*

Ergebnis:

| | Verluste [Anzahl] | Durchsatz [Mbps] |
|---|---|---|
| POS2/0 | 31181 | |
| POS2/1 | 3061 | |
| POS2/2 | 34 | |

Der gemessene Durchsatz bestätigt den gewünschten Effekt der Burst Parameter. Bei gleicher Senderate (burstartig), wird der Durchsatz um so höher (die Anzahl der Paketverluste um so geringer), je höher die Burstwert T1 und T2 am Cisco konfiguriert sind.

Der folgende Test soll die Genauigkeit der beiden Parameter T1 und T2 zeigen. In ausgehender Richtung wird am Interface POS2/0 und POS2/1 Policing mit unterschiedlichen Burstwerten konfiguriert:

```
interface POS2/0
 rate-limit output 76992000 38496 50000 conform-action transmit exceed-action drop
```

```
interface POS2/1
 rate-limit output 76992000 50000 200000 conform-action transmit exceed-action drop
```

Aus früheren Tests ist bekannt, dass die durchschnittliche Empfangsrate nicht exakt mit der am Cisco konfigurierten *Average Rate* beim Policing übereinstimmt: mit den oben genannten CAR Policies wird z.B. ein konstanter Strom von 80 Mbps auf 76.988714 Mbps limitiert. Dieser Wert entspricht also der tatsächlichen durchschnittlich erlaubten Senderate. Um die Burst Parameter T1 und T2 nun auf ihre Exaktheit hin zu überprüfen, muss man einen Strom erzeugen, der mit der *Intended Load* von 76.988714 Mbps sendet, und die Bursts unterschiedlich wählt. Sind die Bursts kleiner als T1, so darf der Strom zu keinen Paketverlusten führen, sind die Burst größer als T2, so müssen die Bursts verkleinert werden und es muss zu Paketverlusten kommen. Sind die Burst zwischen T1 und T2, wird RED-Like verworfen.

T1 von POS2/0 entspricht 89.7 429 B großen Paketen (T1:Paketgröße)
T2 von POS2/0 entspricht 116.55 429 B großen Paketen
T1 von POS2/1 entspricht 116.55 429 B großen Paketen
T2 von POS2/1 entspricht 466.2 429 B großen Paketen

Für das Interface POS2/0 bedeutet das, dass Bursts mit einer durchschnittlichen Senderate von 76.988714 Mbps und einer Burstlänge kleiner als 89.7 Paketen ohne Paketverluste durchkommen sollten. Bursts mit einer durchschnittlichen Senderate von 76.988714 Mbps und einer Burstlänge von mehr als 116.55 Paketen, sollten hingegen zu Paketverlusten führen.

Jeder der folgenden Tests hatte eine Messdauer von 5 Minuten.

| | Intended Load [Mbps] | Burst Load [Mbps] | Burst Length [packets] | Drops [Anzahl] | Receive [Mbps] | |
|---|---|---|---|---|---|---|
| a1) POS2/0 | 76.98871 | 146.2 | 88 | 0 | 76-77 | Burst < T1 |
| a2) POS2/1 | 76.98871 | 146.2 | 115 | **26** | 76-77 | |
| b1) POS2/0 | 76.98871 | 146.2 | 120 | 64 | 76-77 | Burst > T2 |
| b2) POS2/1 | 76.98871 | 146.2 | 470 | 374522 | ca 73 | |
| c1) POS2/0 | 76.980 | 146.2 | 88 | 0 | 76-77 | Burst < T1 |
| c2) POS2/1 | 76.980 | 146.2 | 115 | 0 | 76-77 | |
| d1) POS2/0 | 76.980 | 146.2 | 120 | **0** | 76-77 | Burst > T1 |
| d2) POS2/1 | 76.980 | 146.2 | 470 | 374304 | ca 73 | |

Test a1 und a2 sollten eigentlich zu keinen Verlusten führen (da die Burstwerte des Sendestroms kleiner als die konfigurierten T1-Werte sind), jedoch treten bei a2 leichte Verluste auf.
Test b1 und b2 sollten auf jeden Fall zu Verlusten führen, was auch bestätigt werden konnte.
Da vielleicht die durchschnittliche Senderate geringfügig zu hoch gewählt wurde, werden die gleichen Tests nochmals mit einer niedrigeren Senderate durchgeführt.
Test c1 und c2 senden nun mit einer etwas geringeren Senderate und einem kleineren Burstwert als ihre konfigurierten T1-Werte. Es kommt wie gewünscht zu keinen Verlusten.
Sendet man hingegen mit größeren Burst-Werten als die konfigurierten T2s, so kommt es im Test d1 jedoch zu keinen Verlusten. Entweder arbeiten die Burst-Werte T1 und T2 nicht

völlig exakt, oder die ermittelte durchschnittliche Senderate für die konfigurierten CAR Policies wurde nicht genau genug bestimmt.

Ergebnis:
Ob die konfigurierten Burstsizes T1 und T2 völlig exakt arbeiten, lässt sich nur schwer nachprüfen, da die Sendeparameter sehr exakt gewählt werden müssen. Ungefähr scheinen die Werte jedoch zu stimmen.

#### 2.3.4.2 Test B

Dieser Test soll zeigen, dass sowohl konformer als auch nicht konformer Verkehr markiert werden kann. Auf POS2/0 ist folgende Policing Policy aktiviert:

```
rate-limit output 76992000 38496 50000 conform-action set-prec-transmit 5
exceed-action set-prec-transmit 2
```

Ein konstanter Strom von 80 Mbps (429 B Pakete, Precedence Bit = 0) wird über POS2/0 gesendet. Limitiert wird auf 76.992 Mbps, somit müssten ca. 96.25 % des Sendeverkehrs konform und ca. 3.75 % nicht konform sein. Beim konformen Verkehr soll das Precedence Bit auf 5, beim nicht konformen Verkehr auf 2 gesetzt werden. Mit Hilfe der Capture-Funktion des Agilent lassen sich die Precedence Bits überprüfen. Von insgesamt 118881 empfangenen und analysierten Paketen, waren 4475 (entspricht 3.76 %) mit prec2 und 114406 (entspricht 96.24 %) mit prec5 markiert. Somit markiert der Router den Verkehr richtig. Die Anzahl der ermittelten exceeded Pakete beim Router (*sh int pos2/0 rate-limit*) stimmt wieder exakt mit dem Wert vom Agilent überein, der Zähler für die konformen Paketen differiert um 1.

### 2.3.5 1.2.2.5 Multiple CAR rule tests

#### 2.3.5.1 multiple independent CAR rules (remark)

Vier verschiede, unabhängige Policing Regeln sind am POS2/0 in ausgehender Richtung konfiguriert. Vier verschiedene Verkehrsklassen/-ströme (prec2, prec3, prec4 und alle Pakete mit Zieladresse gleich 192.129.100.2) sollen jeweils auf 10 Mbps limitiert werden. Dabei soll beim konformen Verkehr das Precedence Bit auf 5 beim nicht konformen Verkehr auf 1 gesetzt werden.

```
interface POS2/0
rate-limit output access-group rate-limit 2 10000000 512000 1000000
conform-action set-prec-transmit 5 exceed-action set-prec-transmit 1
 rate-limit output access-group rate-limit 3 10000000 512000 1000000
conform-action set-prec-transmit 5 exceed-action set-prec-transmit 1
 rate-limit output access-group rate-limit 4 10000000 512000 1000000
conform-action set-prec-transmit 5 exceed-action set-prec-transmit 1
 rate-limit output access-group 150 10000000 512000 1000000 conform-action
set-prec-transmit 5 exceed-action set-prec-transmit 1


access-list 150 permit ip any host 192.129.100.2
access-list 150 deny   ip any any
access-list rate-limit 2 2
```

```
access-list rate-limit 3 3
access-list rate-limit 4 4
```

Folgende 5 Verkehrsströme werden dabei über das Interface gesendet:

```
a)  an 192.129.100.1: Senderate 20 Mbps, prec0
b1) an 192.129.100.2: Senderate 20 Mbps, prec0
b2) an 192.129.100.2: Senderate 20 Mbps, prec0
c)  an 192.129.100.1: Senderate 20 Mbps, prec2
d)  an 192.129.100.1: Senderate 20 Mbps, prec3
e)  an 192.129.100.1: Senderate 20 Mbps, prec4
```

Welches Ergebnis sollte erzielt werden:

| Strom | Gesendet [Mbps] | Was sollte damit passieren | | |
|---|---|---|---|---|
| A | 20 prec 0 | 20 prec0 | | |
| b1 | 20 prec 0 | 5 Mbps prec5, 15 Mbps prec1 | | |
| b2 | 20 prec 0 | 5 Mbps prec5, 15 Mbps prec1 | | |
| C | 20 prec 2 | 10 Mbps prec5, 10 Mbps prec1 | | |
| D | 20 prec 3 | 10 Mbps prec5, 10 Mbps prec1 | | |
| E | 20 prec 4 | 10 Mbps prec5, 10 Mbps prec1 | | |
| Summe | | 20 Mbps prec0, 40 Mbps prec5, 60 Mbps prec1 = Verhältnis von 1:2:3 | | |

Testergebnis:
Alle Ströme bekommen ihre Senderate voll durch (keine Verluste). Mit der Capture-Funktion vom Agilent konnte überprüft werden, dass der Verkehr ordnungsgemäß markiert wurde. Von insgesamt 83917 aufgezeichneten Paketen wurden 13986 mit Precedence gleich 0 , 27961 mit Precedence gleich 5 und 41970 mit Precedence gleich 1 markiert. Dies entspricht einem Verhältnis von 1: 1.99 : 3.00, was sehr genau mit dem zu erwartenden Ergebnis von 1:2:3 übereinstimmt.

### 2.3.5.2 multiple independent CAR rules: (drop exceeded packets)

Der vorherige Test wird nun wiederholt (selbe Sendeströme, 4 Policing-Regeln) mit der Änderung, dass die einzelnen Verkehrsströme, ohne Ummarkieren, nun auf 10 Mbps limitiert werden, d.h. nicht konformer Verkehr soll verworfen werden.

```
interface POS2/0
rate-limit output access-group rate-limit 2 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 3 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 4 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group 150 10000000 512000 1000000 conform-action
transmit exceed-action drop


access-list 150 permit ip any host 192.129.100.2
access-list 150 deny   ip any any
access-list rate-limit 2 2
```

```
access-list rate-limit 3 3
access-list rate-limit 4 4
```

Es werden wieder die 5 Verkehrsströme wie beim letzten Test über POS2/0 gesendet:

```
a)  an 192.129.100.1: Senderate 20Mbps, prec0
b1) an 192.129.100.2: Senderate 20Mbps, prec0
b2) an 192.129.100.2: Senderate 20Mbps, prec0
c)  an 192.129.100.1: Senderate 20Mbps, prec2
d)  an 192.129.100.1: Senderate 20Mbps, prec3
e)  an 192.129.100.1: Senderate 20Mbps, prec4
```

Zu erwarten wäre:

1. Die Ströme b) bis e) sollten auf 10 Mbps limitiert werden.
2. Der Verkehr von b1 und b2 soll in Summe auf 10 Mbps limitiert werden und gleichmäßig auf b1 und b2 aufgeteilt werden.
3. Strom a) sollte unverändert bleiben.

Ergebnis:

| Strom | Durchsatz [Mbps] | Verluste |
|---|---|---|
| a) | 20 | Nein |
| b1)<br>b2) | 0.26<br>9.73 | ja<br>ja |
| c) | 10 | Ja |
| d) | 10 | Ja |
| e) | 10 | Ja |

Das Ergebnis ist nicht wie erwartet. Zwar begrenzen alle 4 Policingregeln den Verkehr jeweils auf die angegebenen 10 Mbps, aber die beiden Ströme b1 und b2 werden nicht gleich behandelt. Strom b2 wird stark bevorzugt.

Dieses Problem besteht auch beim Shaping. Es ist außerdem nicht nur auf diese LC beschränkt. Auf einer OC48 Engine3 LC tritt dieses Phänomen ebenfalls auf. Wir informierten Cisco darüber, haben aber bisher noch keine Antwort bekommen. Mit Hilfe der Capture-Funktion des Agilent RouterTesters konnten wir nachweisen, dass die beiden Ströme b1 und b2 gleichmässig gesendet wurden.

### 2.3.5.3 multiple dependent CAR rules (continue)

Dieser Test soll zeigen, dass kaskadierte Policing Regeln möglich sind. Folgende drei Tests wurden durchgeführt:

a)
Folgende Regel ist auf POS2/0 aktiviert:

```
!
interface POS2/0
rate-limit output access-group rate-limit 10 10000000 512000 1000000
conform-action continue exceed-action drop
```

```
 rate-limit output access-group 151 5000000 512000 1000000 conform-action
transmit exceed-action drop

access-list 151 permit tcp any any
access-list 151 deny   ip any any
access-list rate-limit 10 0
```

Ein UDP-Strom (prec = 0) mit einer Senderate von 11 Mbps wird korrekt auf 10 Mbps begrenzt, da er nur in die erste Policingregel fällt. Ein TCP-Strom mit einer Senderate von 11 Mbps durchläuft hingegen beide Policingregeln und wird somit korrekt auf 5 Mbps limitiert.

b)

```
!
interface POS2/0
 rate-limit output access-group rate-limit 10 10000000 512000 1000000
conform-action set-prec-continue 1 exceed-action drop
 rate-limit output access-group 151 5000000 512000 1000000 conform-action
transmit exceed-action drop

access-list 151 permit tcp any any
access-list 151 deny   ip any any
access-list rate-limit 10 0
```

Ein versendeter UDP-Stroms mit 11 Mbps (prec 0) wird auf 10 Mbps korrekt begrenzt, und das Precedence Bit wird auf 1 gesetzt. Ein gesendeter TCP-Stroms mit 11 Mbps (prec 0) wird auf 5 Mbps limitiert, und das Precedence Bit wird auf 1 gesetzt.

c)

```
!
interface POS2/0
rate-limit output access-group rate-limit 10 10000000 512000 1000000
conform-action set-prec-continue 1 exceed-action drop
 rate-limit output access-group rate-limit 1 5000000 512000 1000000
conform-action transmit exceed-action drop
 encapsulation ppp

access-list rate-limit 1 1
access-list rate-limit 10 0
```

Ein versendeter UDP-Stroms mit 11Mbps (prec 0) wird auf 10Mbps begrenzt, und das Precedence Bit wird auf 1 gesetzt. Eigentlich sollte dieser Strom im weiteren auf 5 Mbps eingeschränkt werden, da der ummarkierte Strom von der zweiten Regel weiter behandelt werden müsste. Sendet man direkt einen UDP-Strom bei dem das Precedence Bit bereits auf 1 gesetzt wird, so wird dieser Strom korrekt auf 5Mbps limitiert. Prinzipiell würde Regel 2 somit funktionieren.

Der im Testplan angegebene Vorschlag, konformen Verkehr zu markieren und diesen neue markierten Verkehr nun nochmals durch weitere Policing Regeln weiterzubehandeln ist somit nicht möglich.

Cisco bestätigt uns, dass das Markieren von konformen Paketen und das anschließende weiterbehandeln dieser Pakete aufgrund des neuen, gesetzten Precedence Bits nicht möglich ist. Wird beim Policing das Precedence Bit ummarkiert, so bleibt die Ummarkierung in

folgenden Regeln unbeachtet. Dort wird stets vom ursprünglichen Precedence Bit ausgegangen.
Findet die Weiterbehandlung der Ströme jedoch nach anderen Kriterien statt (z.B. extended ACL), funktionieren *continue* und *set-prec-continue* korrekt.

#### 2.3.5.4 Performance-Tests

Der folgende Test soll zeigen, dass es zu keinen Performance-Einbußen kommt, wenn mehrere Policing-Regeln auf mehreren Links gleichzeitig aktiviert sind. Folgende Regeln sind konfiguriert.

```
interface POS2/0
rate-limit output access-group rate-limit 2 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 3 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 4 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 1 10000000 512000 1000000
conform-action transmit exceed-action drop

interface POS2/1
 rate-limit output access-group rate-limit 2 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 2 9000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 3 9000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 3 8000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 1 11000000 512000 1000000
conform-action transmit exceed-action drop

!
interface POS2/2
rate-limit output access-group rate-limit 2 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 3 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 4 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 1 10000000 512000 1000000
conform-action transmit exceed-action drop

interface POS2/3

 rate-limit output access-group rate-limit 2 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 3 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 4 10000000 512000 1000000
conform-action transmit exceed-action drop
 rate-limit output access-group rate-limit 1 10000000 512000 1000000
conform-action transmit exceed-action drop
 encapsulation ppp
```

```
access-list rate-limit 1 1
access-list rate-limit 2 2
access-list rate-limit 3 3
access-list rate-limit 4 4
```

#### 2.3.5.4.1 Policing an 4 Links

Da auf dem Agilent pro Port nur 15 Flows erlaubt sind, werden pro QOC3 Link jeweils nur 3 unterschiedliche Flows (prec0, prec1, prec2) versendet: mit konstanten und burstartigen Sendeverhalten, mit einer Paketgröße und mit einem Paketmix. Somit fließt nicht über jede der oben konfigurierten Policingregel auch wirklich Verkehr.

Versendete Verkehrsströme:

*über POS2/0*
*a0) Senderate 105 Mbps, prec0, const, 429 B*
*b0) Senderate 20 Mbps, prec1, const, 429 B*
*c0) Senderate 20 Mbps, prec2, const, 429 B*

*über POS2/1*
*a1) Senderate: Burst: 105 Mbps, Burst Load: 146 Mbps, Length: 30, 429 B, prec0*
*b1) Senderate : Burst: 20 Mbps, Burst Load: 146 Mbps, Length: 30, 429 B, prec1*
*c1) Senderate : Burst: 20 Mbps, Burst Load: 146 Mbps, Length: 30, 429 B, prec2*

*über POS2/2*
*a2) Senderate: Burst: 105 Mbps, Burst Load: 146 Mbps, Length: 30, prec0, Packet Range: 48-1500 B*
*b2) Senderate : Burst: 20 Mbps, Burst Load: 146 Mbps, Length: 30, prec1, Packet Range: 48-1500 B*
*c2) Senderate : Burst: 20 Mbps, Burst Load: 146 Mbps, Length: 30, prec2, Packet Range: 48-1500 B*

*über POS2/3*
*a3) Senderate 105 Mbps, prec0, const, Packet Range: 48-1500 B*
*b3) Senderate 20 Mbps, prec1, const, Packet Range: 48-1500 B*
*c3) Senderate 20 Mbps, prec2, const, Packet Range: 48-1500 B*

Zusätzlich wird noch über jeden der 4 Links der QOC3 LC ein konstanter Rückstrom von 146.2 Mbps (429 B) und über die OC192 LC ein Strom (hin und zurück) mit einer konstanten Senderate von 1500 Mbps geschickt. Diese Strömen dienen nur dazu, um festzustellen, ob Policing im Lasttest negativen Einfluss auf andere LC oder auf die Gegenströme hat.

Ergebnis:

Während der folgenden Tests bleibt die Last der Haupt-CPU und der CPU der LCs (slot 2 und slot 4) konstant bei 0-1 %.

| Ströme | Durchsatz POS2/0 [Mbps] | Durchsatz POS2/1 [Mbps] | Durchsatz POS2/2 [Mbps] | Durchsatz POS2/3 [Mbps] |
|---|---|---|---|---|
| Prec0 | 105 | 105 | 105 | 105 |
| Prec1 | 9.98 | 10.93-10.98 | 9.98-10.01 | 9.82-10.14 |
| Prec2 | 9.98 | 9.97-10.02 | 9.93-10.07 | 9.87-10.01 |

Es treten keine Verluste auf den Rückströmen oder auf den Strömen der OC192 Karte auf. Alle nicht limitierten Ströme mit Precedence Bit gleich 0 kommen ebenfalls verlustfrei durch. Somit führt Policing zu keinen unerwünschten Paketverlusten an anderen Strömen. Der Durchsatz bei burstartigen Strömen oder bei Strömen mit verschiedenen Paketgrößen schwankt etwas.

#### 2.3.5.4.2 Policing an 3 Links

Nun wird Policing an POS2/0 deaktiviert, so dass nur der Verkehr an 3 Links begrenzt wird. Gesendet wird wie im Test 2.3.5.4.1.

Ergebnis:

| Ströme | Durchsatz POS2/0 [Mbps] | Durchsatz POS2/1 [Mbps] | Durchsatz POS2/2 [Mbps] | Durchsatz POS2/3 [Mbps] |
|---|---|---|---|---|
| Prec0 | 105 | 105 | 105 | 105 |
| Prec1 | 20 | 10.93-10.97 | 9.92-10.03 | 9.97-10.03 |
| Prec2 | 20 | 9.97-10.01 | 9.91-10.02 | 9.94-10.02 |

Es treten keine Verluste auf den Rückströmen oder auf den Strömen der OC192 Karte oder auf dem vierten nicht limitierten Link auf. Somit führt Policing zu keinen unerwünschten Paketverlusten an anderen Strömen.

### 2.3.6 Zusammenfassung

Policing auf der QOC3LC Engine3 führt zu keinen Performanceeinbußen. Sowohl auf der Haupt-CPU als auch auf der CPU der LC konnte keine Lasterhöhung festgestellt werden. Policing hat keinen negativen Einfluss auf andere Ströme derselben Karte. Solange ein Strom unterhalb der Policinggrenze sendet, kommt er verlustfrei durch. Die Zähler auf dem Router (conformed/exceeded packets) sind korrekt. Mittels Policing kann Verkehr sowohl verworfen als auch (z.B. in eine schlechtere Klasse) ummarkiert werden. Policing funktioniert auch mit ACL, speziellen rate-limit-ACL und auf QoS-Gruppen. Auch kaskadiertes Policing funktioniert, solange Pakete nicht ummarkiert werden und diese in einer weiteren Regel weiter behandelt werden sollen. Hier würde sich der Router nicht auf die ummarkierten Werte beziehen, sondern stets auf die Ausgangswerte. Werden mehrere, gleiche Ströme (z.B. mehrere gleichartige Anwendungen, die dieselbe Bandbreite und selbe Paketgrößen verwenden gemeinsam limitiert, so wird der gemeinsame Durchsatz nicht gleichmäßig auf die einzelnen Ströme verteilt. Dieses Phönomen ist jedoch nicht auf diese LC beschränkt, sondern tritt beispielsweise auch auf einer OC48 LC (Engine2 oder 3) auf. Eine Erklärung hierfür ist unbekannt, eine Antwort von Cisco steht noch aus.

## 2.4 Rate-Limiting Using Traffic Shaping

Traffic Shaping als weitere Möglichkeit, Verkehrsströme einzuschränken, ist auf der ISE OC3 Linecard sowohl in ingress-, als auch in egress-Richtung möglich. Die Shaping Parameter werden über MQC-definierte Klassen gesetzt. Beispielsweise wird mit der folgenden Konfiguration der Verkehr mit einem Precedencebitwert von 3 auf 16.4 Mbps beschränkt:

```
class-map match-all prec3
  match ip precedence 3
class prec3
    shape average 16384000
    queue-limit 1000 packets
```

Wie in Kap. 2.2.2 schon beschrieben, ist der Befehl `shape max-buffers` auf diesen Linecards nicht vorhanden, so dass mit `queue-limit` gearbeitet werden muss. Dies führt zu einer Paketgrößenabhängigkeit der Queue. Die Angabe eines Queue-Limits ist zwar nicht zwingend erforderlich, sollte aber geschehen, um die Größe der Output- bzw. Inputqueue zu begrenzen. Andernfalls könnte ein überlasteter Strom/Link den gesammten Paketpuffer in Anspruch nehmen, was dazu führen könnte, dass nicht überlastete Ströme/Links ihren Verkehr nicht mehr verlustfrei durchbekommen. Ohne Begrenzung des Paketpuffers würden die Laufzeiten der Pakete in inakzeptable Bereiche ansteigen.

Diese Verkehrslimitierung wird am Interface in Ausgangs- oder Eingangsrichtung gesetzt. Im folgenden Beispiel wird am Interface POS2/0 Eingangsseitig Shaping aktiviert:

```
interface POS2/0
 ip address 192.129.3.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast source reachable-via any
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 18
 encapsulation ppp
 service-policy input prec3
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable
```

### 2.4.1 Traffic-Shaping Test 1.2.2.6

Die Funktionalität und Genauigkeit bzw. Granularität wurde zunächst mit dem in Abbildung 2 beschriebenen Testaufbau ausgangsseitig am Interface POS2/0 überprüft.

Gesendeter Verkehrsstrom: Agilent 101/2 (4.2) -> 12410 POS 2/1 (4.1) - 12410 POS 2/0 (3.1, hier shaping aktiviert) -> Agilent 101/1 (3.2), TCP unidirektional, Paketgröße: 429 B, Tx = 99% OC3 = 145.22 Mbps.

Variiert wurde hier die Shaping Rate zwischen 4 kbps und 64 Mbps. Beispielsweise wird der ankommende Strom mit der folgenden Policy-Map auf 32 Mbps auf Interface POS2/0 ausgehend begrenzt.

```
policy-map shape155to32M
  class ANY
    shape average 32768000
    queue-limit 1000 packets

interface POS2/0
 ip address 192.129.3.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 encapsulation ppp
service-policy output shape155to32M
 no ip mroute-cache
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable
```

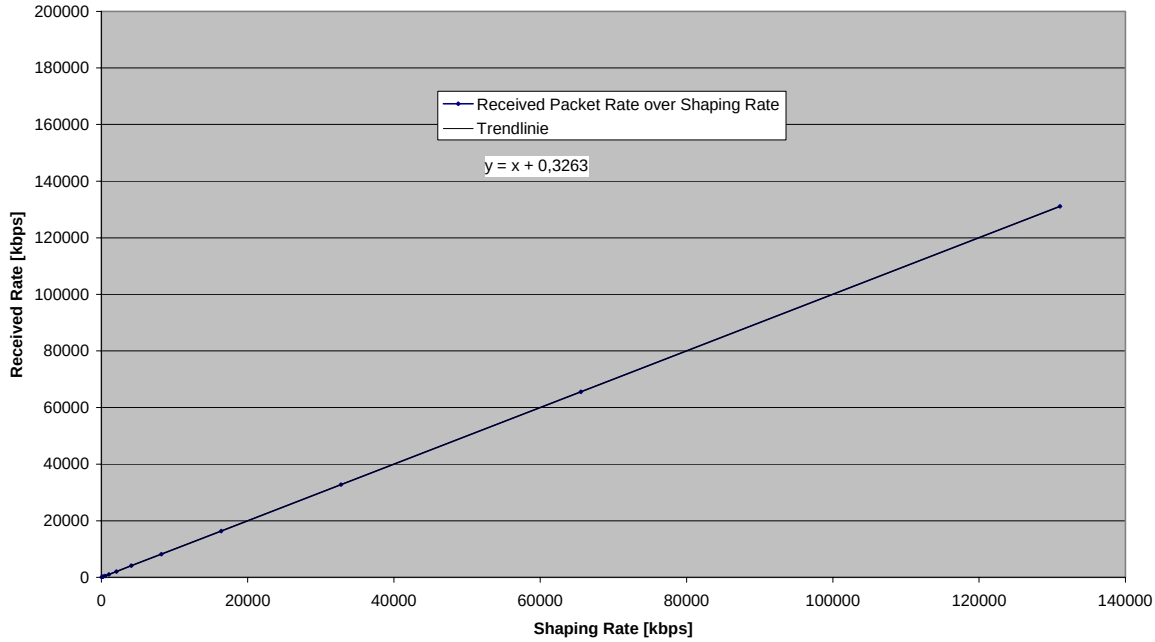



*Abbildung 5: Genauigkeit der Shaping-Einstellung*

Die Genauigkeit, mit der die eingestellten Shaping Parameter von der Linecard eingehalten wird, ist in Abbildung 5 dargestellt. Man kann erkennen, das die erwarteten Werte mit den gemessenen sehr genau übereinstimmen. In der tabellarischen Darstellung der Ergebnisse (Tabelle 3) sieht man sehr geringe Abweichungen zwischen der eingestellten Shaping Rate und den erhaltenen Werten. Bei diesem Test wurden keine Queue Limits eingestellt, was bei Shaping Raten unter 4096 kbps zu sehr hohen und periodisch schwankenden Latency Werten führt. Bei höheren Shaping Raten tritt dieses Verhalten nicht auf. Nach Rücksprache mit Cisco muß unbedingt ein Queue Limit eingestellt werden, um konstante Werte zu erreichen. Informationen darüber, was die Linecard ohne explizite Konfiguration des Queue Limits per default einstellt, konnte Cisco bisher nicht liefern.

*Tabelle 3: Shaping - Genauigkeit*

| Shaping to [kbps] | Shaping to [Pps] | Received Rate [kbps] | Received Packets [Pps] | average latency [µs] |
|---|---|---|---|---|
| 64 | 19 | 65 | 19 | 42000000 |
| 128 | 37 | 130 | 38 | 42000000 |
| 256 | 75 | 257 | 75 | 42000000 |
| 512 | 149 | 511 | 149 | 42000000 |
| 1024 | 298 | 1023 | 298 | 42000000 |
| 2048 | 597 | 2049 | 597 | 42000000 |
| 4096 | 1193 | 4094 | 1193 | 42000000 |
| 8192 | 2387 | 8192 | 2387 | 31616273 |
| 16384 | 4774 | 16381 | 4773 | 15808156 |
| 32768 | 9548 | 32769 | 9548 | 7904097 |
| 65536 | 19096 | 65537 | 19096 | 3952068 |
| 131072 | 38191 | 131068 | 38190 | 1976052 |

Die Shaping Rate kann nur in Schritten von 64 kbps eingestellt werden. Zur Überprüfung, wie exakt diese Werte eingehalten werden, wurde im Bereich hoher Shaping Raten (ca. 131 Mbps) die Rate schrittweise um 64 kbps erniedrigt. Tabelle 4 zeigt, dass auch bei hohen Senderaten die Shaping Raten präzise eingehalten werden. In diesem Test wurde die Queue-Länge auf 1000 Pakete eingestellt, dies bedingt die Latency-Werte, die linear bei sinkender Shaping Rate ansteigen.

*Tabelle 4: Shaping - Granularität*

| Shaping to [kbps] | Shaping to [Pps] | Received Rate [kbps] | Received Packets [Pps] | average latency [µs] |
|---|---|---|---|---|
| | | | | |
| 131072 | 38191 | 131068 | 38190 | 26243.92 |
| 131008 | 38172 | 131006 | 38172 | 26253.21 |
| 130944 | 38154 | 130941 | 38153 | 26266.04 |
| 130880 | 38135 | 130876 | 38134 | 26278.6 |
| 130816 | 38117 | 130814 | 38116 | 26291.26 |
| 130752 | 38098 | 130749 | 38097 | 26304.26 |
| 130688 | 38079 | 130684 | 38078 | 26317.18 |
| 130624 | 38061 | 130618 | 38059 | 26330.19 |
| 130560 | 38042 | 130557 | 38041 | 26343.3 |
| 130496 | 38023 | 130492 | 38022 | 26356.12 |
| 130432 | 38005 | 130426 | 38003 | 26368.64 |
| 130368 | 37986 | 130365 | 37985 | 26381.78 |
| 130304 | 37967 | 130299 | 37966 | 26394.67 |
| 130240 | 37949 | 130234 | 37947 | 26407.65 |

Den Einfluss der Queue-Länge auf die Latency der Pakete zeigt Tabelle 5. Die Shaping Rate wurde fest auf 131072 kbps eingestellt, die Verkehrsparamter waren die gleichen wie in den vorhergehenden Tests. Erwartungsgemäß zeigt sich ein linearer Zusammenhang zwischen eingestellter Queuelänge und der mittleren Latency der Pakete.

*Tabelle 5: Shaping - Queue Length*

| Queue Length [Packets] | Received Rate [%] | Received Rate [kbps] | average latency [µs] |
|---|---|---|---|
| 10 | 90.3% | 131068 | 316.36 |
| 50 | 90.3% | 131068 | 1364.19 |
| 100 | 90.3% | 131068 | 2672.3 |
| 500 | 90.3% | 131068 | 13151.54 |
| 1000 | 90.3% | 131068 | 26243.92 |
| 5000 | 90.3% | 131068 | 130983.62 |
| 100000 | 90.3% | 131068 | 261908.58 |

Die Paketgrößenabhängigkeit der erreichten Shaping Rate ist ebenfalls gering. In Tabelle 6 sieht man, dass, bei einer eingestellten Rate von 32768 kbps auf der Empfangsseite auch diese nahezu erreicht wird. Die Abweichung beträgt unter 1%.

*Tabelle 6: Shaping - Paketgröße*

| Packet-size [Bpp] | Sending Rate [Mbps] | Sending Rate [Pps] | Shaping to [Pps] | Received Packets [Pps] | Received Rate [%] | Received Rate [kbps] |
|---|---|---|---|---|---|---|
| 64 | 129.98 | 253874 | 64000 | 64000 | 25.2% | 32768 |
| 100 | 136.02 | 170026 | 40960 | 40956 | 24.1% | 32765 |
| 429 | 145.22 | 42312 | 9548 | 9548 | 22.6% | 32769 |
| 1500 | 147.38 | 12282 | 2731 | 2731 | 22.2% | 32772 |
| 4096 | 147.94 | 4515 | 1000 | 1000 | 22.1% | 32768 |

### 2.4.2 Traffic-Shaping, weitere Tests (1.2.2.7 ff.)

Um Traffic-Shaping unter hohen Lasten zu simulieren, wurde mit dem in Abbildung 1 beschriebenen Testaufbau mehrere Ströme auf mehreren Interfaces der Linecard gesendet.
Die gesendeten Ströme wurden unterschiedlich markiert, entweder durch das Setzen des Precedence Bits oder durch Setzen des Source bzw. Destination Ports (z.B. www, ftp) im UDP Header.
Abhängig vom markierten Strom wurden unterschiedliche Shaping-Raten gesetzt und das Verhalten der Karte überprüft.

Die definierten Klassen waren:

```
class-map match-all ANY
  match any
class-map match-all ftp
  match access-group 102
class-map match-all prec_3
  match ip precedence 3
class-map match-all prec_2
  match ip precedence 2
class-map match-all prec_1
  match ip precedence 1
class-map match-all prec_0
  match ip precedence 0
class-map match-all prec_7
  match ip precedence 7
class-map match-all prec_6
  match ip precedence 6
class-map match-all prec_5
  match ip precedence 5
class-map match-all prec_4
  match ip precedence 4
class-map match-all www
  match access-group 101

policy-map shape10queuelimits
```

```
  class prec_1
    shape average 64000
    queue-limit 200 packets
  class prec_2
    shape average 128000
    queue-limit 200 packets
  class prec_3
    shape average 256000
    queue-limit 200 packets
  class prec_4
    shape average 512000
    queue-limit 200 packets
  class prec_5
    shape average 1024000
    queue-limit 200 packets
  class prec_6
    shape average 2048000
    queue-limit 200 packets
  class prec_7
    shape average 4096000
    queue-limit 200 packets
  class www
    shape average 8192000
    queue-limit 200 packets
  class ftp
    shape average 16384000
    queue-limit 200 packets
```

mit folgenden Access-Listen

```
access-list 101 permit tcp any any eq www
access-list 102 permit tcp any any eq ftp
```

Diese Parameter wurden auf alle vier Ports der Lincard angewendet:

```
interface POS2/0
 ip address 192.129.3.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast source reachable-via any
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 18
 encapsulation ppp
 service-policy output shape10queuelimits
 crc 32
 down-when-looped
 clock source internal
```

```
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable
!
interface POS2/1
 ip address 192.129.4.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast source reachable-via any
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 18
 encapsulation ppp
 service-policy output shape10queuelimits
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable
!
interface POS2/2
 ip address 192.129.5.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast source reachable-via any
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 18
 encapsulation ppp
 service-policy output shape10queuelimits
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
 no cdp enable
!
interface POS2/3
 ip address 192.129.6.1 255.255.255.0
 ip access-group 199 in
```

```
ip verify unicast source reachable-via any
no ip redirects
no ip directed-broadcast
ip pim bsr-border
ip pim sparse-mode
ip multicast boundary 18
encapsulation ppp
service-policy output shape10queuelimits
crc 32
down-when-looped
clock source internal
pos ais-shut
pos framing sdh
pos scramble-atm
pos flag s1s0 2
no cdp enable
```

Im Einzelnen wurden diese Ströme gesendet:

TCP 429 Byte 6 % OC48

1A-POS4/0-POS2/0-POS2/0-POS4/0-1B : prec 1
1A-POS4/0-POS2/1-POS2/1-POS4/0-1B : prec 2
1A-POS4/0-POS2/2-POS2/2-POS4/0-1B : www
1A-POS4/0-POS2/3-POS2/3-POS4/0-1B : ftp
1A-POS4/0-POS0/0-POS0/0-POS4/0-1B : bidirectional 75 % OC48 (control)

Es zeigte sich, dass bei Shaping auf der Ausgangsseite alle Ströme gemäß der zugewiesenen Parameter von der Linecard behandelt werden. Eine Erhöhung der CPU-Last wurde dabei weder auf der Linecard noch auf der Backplane festgestellt.

```
c4101#exec slot 2 sh proc cpu | include CPU
========= Line Card (Slot 2) =========
CPU utilization for five seconds: 0%/0%; one minute: 0%; five
minutes: 0%

c4101#sh proc cpu | include CPU
CPU utilization for five seconds: 0%/0%; one minute: 0%; five
minutes: 0%
```

### 2.4.3 Zusammenfassung Traffic Shaping

Die Traffic Shaping Tests zeigen, dass sowohl die eingestelleten Werte hinsichtlich Granularität und Exaktheit für Betriebszwecke geeignet eingehalten werden. Die Limitierung auf 64 kbps-Schritte wirkt sich nur in Bereichen aus, in denen der Verkehr auf sehr kleine Raten limitiert wird, was in der Praxis selten vorkommen sollte. Die Karte schreibt keine Konfiguration eines Queue-Limits vor. Dieses ist aber unbedingt einzustellen, da bei kleinen Shapingraten das Latencyverhalten sonst unvorhersehbar ist. Die Karte zeigt keine

Performanceeinbußen bei hoher Belastung durch mehrere zu limitierende Verkersströme. Auch die CPU-Last der Backplane wird durch Shaping nicht beeinflusst.

## 2.5 WRED

RED (Random Early Detection) und WRED (Weighted RED) als Mechanismus zur Überlastverhinderung wurde getestet, um die Möglichkeit festzustellen, ob verschieden priorisierte Verkehrsströme entsprechend unterschiedlicher WRED-Parameter behandelt werden können. Dazu wurde die folgende Klasse definiert:

```
class-map match-any prec1-4
  match ip precedence 1
  match ip precedence 2
  match ip precedence 3
  match ip precedence 4
```

Die eingestellten (W)RED-Parameter sind in der folgenden Policy-Map gelistet:

```
policy-map wred
  class prec1-4
    bandwidth percent 99
   random-detect
 random-detect precedence 1 1000 packets 2024 packets 1
 random-detect precedence 2 1400 packets 3448 packets 1
 random-detect precedence 3 1800 packets 5896 packets 1
```

Die eingestellten Parameter wurde so gewählt, dass sich die Bereiche, in denen die verschiedenen Verkehrsströme durch RED beeinflusst werden, überlappen. Dies hat den Sinn, dass festgestellt werden kann, ob Verkehrsströme mit einer höheren Priorität auch bevorzugt behandelt werden, bzw. niedrig priore Ströme zuerst verworfen werden. Dieses Profil wurde am Interface POS2/0 aktiviert:

```
interface POS2/0
 ip address 192.129.3.1 255.255.255.0
 ip access-group 199 in
 ip verify unicast source reachable-via any
 no ip redirects
 no ip directed-broadcast
 ip pim bsr-border
 ip pim sparse-mode
 ip multicast boundary 18
 encapsulation ppp
 service-policy output wred
 crc 32
 down-when-looped
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
```

```
pos flag s1s0 2
no cdp enable
```

Gesendet wurden über das Interface POS2/0 drei Ströme, wobei jeweils das Precedencebit auf 1, 2 und 3 gesetzt war. Die Ströme mit prec = 1 bzw. prec = 2 waren konstant (20000 Pps, 429 Bpp, UDP), der Verkehrstrom mit prec = 3 wurde von 0 Pps bis ca. 45000 Pps gesteigert. In Abbildung 6 ist die Paketrate der einzelnen Ströme über der Paketrate des prec = 3 Stroms aufgetragen. Es ist zu sehen, dass zunächst der Strom mit prec = 1 allein verworfen wird, bis der RED-Startwert des Stroms prec = 2 erreicht wird. Mit weiter steigendem prec = 3 Verkehr sinken die beiden anderen Ströme bis vom niedrigst-priorem Strom alles verworfen wird (oberer Wert der RED-Parameter). Danach sinkt der Strom mit prec = 2 auf 0, so dass zuletzt nur noch der höchstpriore Strom durchgelassen wird. Dieses Verhalten entspricht dem Erwarteten. Die Beobachtung der CPU-Last der Backplane und der Karte zeigte keine Erhöhung der Auslastung.

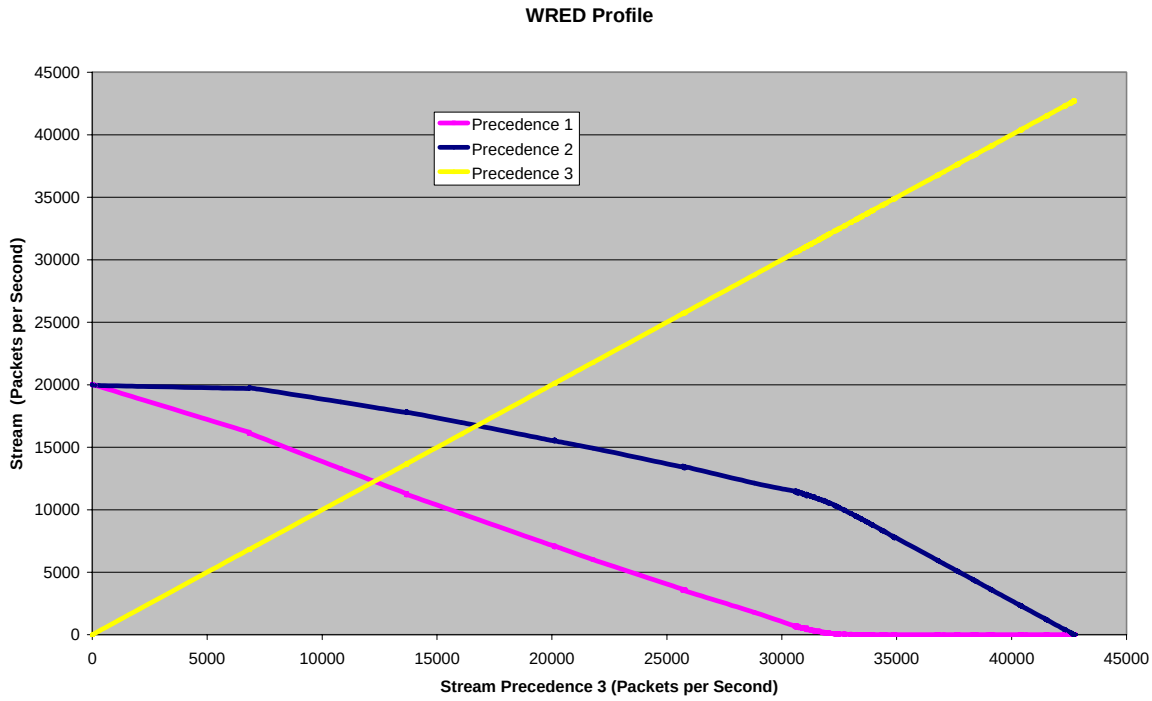



*Abbildung 6: WRED-Profil*

## 2.6 MDRR

Im folgenden wird das Scheduling-Verfahren MDRR getestet. Hierzu werden 6 verschiedene Klassen definiert, die anschließend mittels MDRR unterschiedlich am Ausgangsinterface POS2/0 behandelt werden sollen. Eine Klasse – alle Pakete mit Precedence Bit gleich 5 – wird als *low latency* Klasse deklariert. Pakete aus dieser Klasse sollen vor allen anderen bevorzugt behandelt werden. Die folgende Tabelle zeigt welche Klassen definiert wurden und mit welcher Priorität diese vom Scheduler behandelt werden.

| Klasse | Welcher Verkehr gehört zu dieser Klasse | Welche MDRR-Priorität bzw. wie viel Mindestbandbreite steht dieser Klasse zu |
|---|---|---|
| *prec1* | Alle Pakete mit Precedence Bit = 1 | 25 % ~ 36.55 Mbps |
| *prec2* | Alle Pakete mit Precedence Bit = 2 | 20 % ~ 29.24 Mbps |
| *prec3_4* | Alle Pakete mit Precedence Bit = 3 oder 4 | 20 % ~ 29.24 Mbps |
| *Prec6* | Alle Pakete mit Precedence Bit = 6 | 10 % ~ 14.62 Mbps |
| *low-latency* | Alle Pakete mit Precedence Bit = 5 | *strict prior* |
| *class-default* | Alle anderen Pakete, hier mit Precedence Bit = 0 | Rest: ca. 25% ~ 36.55 Mbps |

Die entsprechende Routerkonfiguration sieht wie folgt aus:

```
policy-map MDRR
  class prec3_4
    bandwidth percent 20
   random-detect
 random-detect precedence 3 255 packets 511 packets 1
 random-detect precedence 4 255 packets 511 packets 1
  class prec_1
    bandwidth percent 25
   random-detect
 random-detect precedence 1 255 packets 511 packets 1
  class prec_2
    bandwidth percent 20
   random-detect
 random-detect precedence 2 255 packets 511 packets 1
  class prec_6
    bandwidth percent 10
   random-detect
 random-detect precedence 6 255 packets 511 packets 1
  class prec_5
    priority
    police 640000 4470 4470 conform-action transmit exceed-action
drop
  class class-default
   random-detect
 random-detect precedence 0 255 packets 511 packets 1
 random-detect precedence 7 255 packets 511 packets 1
```

Im folgenden werden 4 Tests durchgeführt, anhand denen die Funktionalität von MDRR überprüft wird.

### 2.6.1 Alle Klassen sind überlastet

Folgende Ströme werden mit dem Routertester erzeugt und über die QOC3LC geschickt. Somit entsteht eine Überlast an POS2/0 in ausgehender Richtung.

über POS2/0 in ausgehender Richtung:
a) prec0, 50 Mbps, 429 B
b) prec1, 50 Mbps, 429 B

c) prec2, 50 Mbps, 429 B
d) prec3, 50 Mbps, 429 B
e) prec5, 0.6 Mbps, 429 B
f) prec6, 50 Mbps, 429 B

Ergebnis:

| Klasse | Durchsatz [Mbps] | Delay [ms] |
|---|---|---|
| *class-default* | 35.65 | 39.25 |
| *prec1* | 35.65 | 39.14 |
| *prec2* | 29.89 | 49.29 |
| *prec3* | 29.89 | 49.29 |
| *low-latencvy* | 0.6 | 0.588 |
| *prec6* | 14.98 | 109.5 |

Alle Klassen senden mehr als ihre konfigurierte Mindestbandbreite. Der Durchsatz jeder Klasse entspricht in etwa dem zu erwartenden Duchsatz gemäß der jeweiligen konfigurierten Scheduling Bandbreite. Die low-latency Klasse hat - wie gewünscht - das geringste Delay und keine Verluste. Das Delay der anderen Klassen richtet sich schön nach den konfigurierten MDRR-Parametern. Je höher die Priorität, desto geringer ist das Delay.

### 2.6.2 Die Klasse class-default sendet nichts

Nun wird der Test 2.6.1 wiederholt mit der Änderung, dass die Klasse *class-default* nichts mehr sendet. Die anderen senden wie gehabt, die Router-Konfiguration bleibt wie oben beschrieben. Da die Klasse *class-default* nicht ihre zur Verfügung stehende Bandbreite von 35.65Mbps in Anspruch nimmt, soll diese nun von den anderen Strömen mitgenutzt werden können. Die zusätzliche Bandbreite soll dabei gemäß der konfigurierten MDRR-Parameter aufgeteilt werden, d.h. Klassen mit einem hohen MDRR-Wert dürfen mehr von der zusätzlich zur Verfügung stehenden Bandbreite in Anspruch nehmen, als Klassen mit einem niedrigeren MDRR-Wert. Zu erwarten wäre für jede Klasse folgende Bandbreitenerhöhung.

| Klasse | MDRR-Priorität | Zu erwartende Durchsatzerhöhung in % von der zur Verfügung stehenden Bandbreite von 35.65Mbps |
|---|---|---|
| *class-default* | 25% | -> 0% |
| *prec1* | 25% | -> 33.3% ~ 11.81 Mbps |
| *prec2* | 20% | -> 26.7% ~ 9.52 Mbps |
| *prec3* | 20% | -> 26.7% ~ 9.52 Mbps |
| *low-latencvy* | Strict prior | -- |
| *prec6* | 10% | 13.3% ~ 4.74 Mbps |
| *Sum* | 100% | 100% ~ 35.65 Mbps |

Testergebnis

| Klasse | Durchsatz [Mbps] | Delay [ms] | Durchsatzerhöhung [Mbps] | Abweichung vom erwarteten Wert |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| *class-default* | 0 | -- | 0 | -- |
| *prec1* | 48.68 | 22.6 | 13.05 | +9.9 % |
| *prec2* | 38.93 | 35.11 | 9.11 | -5.0% |
| *prec3* | 38.93 | 35.11 | 9.11 | -5.0% |
| *low-latency* | 0.6 | 0.588 | -- | --- |
| *prec6* | 19.5 | 82.8 | 4.52 | -4.64% |
| *Summe* | 146.64 | | 35.79 | |

Die freie Bandbreite wird komplett auf die anderen Ströme aufgeteilt, allerdings weicht die Aufteilung auf die einzelnen Ströme vom zu erwartenden Wert um bis zu 9.9% ab.

### 2.6.3 Die Klasse prec1 sendet nichts

Nun wird der Test 2.6.1 wiederholt mit der Änderung, dass die Klasse *prec1* nichts mehr sendet. Die anderen senden wie gehabt, die Router-Konfiguration bleibt wie oben beschrieben. Die zur Verfügung stehende Bandbreite von 35.65Mbps der Klasse prec1 soll nun auch von den anderen Klassen mitgenutzt werden können. Zu erwarten wäre folgende Bandbreitenerhöhung.

| Klasse | MDRR-Priorität | Zu erwartende Durchsatzerhöhung in % von der zur Verfügung stehenden Bandbreite von 35.65 Mbps |
|---|---|---|
| *class-default* | 25 % | -> 33.3 % ~ 11.81 Mbps |
| *prec1* | 25 % | -> 0 % |
| *prec2* | 20 % | -> 26.7 % ~ 9.52 Mbps |
| *prec3* | 20 % | -> 26.7 % ~ 9.52 Mbps |
| *low-latency* | Strict prior | -- |
| *prec6* | 10 % | 13.3 % ~ 4.74 Mbps |
| *Sum* | 100 % | 100 % ~ 35.65 Mbps |

Ergebnis

| Klasse | Durchsatz [Mbps] | Delay [ms] | Durchsatzerhöhung [Mbps] | Abweichung vom erwarteten Wert |
|---|---|---|---|---|
| *class-default* | 39.65 | 33.49 | 4 | -66.3 % |
| *prec1* | 0 | - | 0 | -- |
| *prec2* | 42.55 | 30.28 | 12.68 | +33.0 % |
| *prec3* | 42.55 | 30.28 | 12.68 | +33.0 % |
| *low-latencvy* | 0.6 | 0.588 | -- | --- |
| *prec6* | 21.31 | 74.9 | 6.34 | +33.7 % |
| *Sum* | 146.66 | | 35.70 | |

Auch hier wird die frei gewordene Bandbreite komplett von den anderen Strömen mitgenutzt. Der tatsächliche Durchsatz weicht nun jedoch um bis zu 66.3% vom erwarteten Durchsatz ab! Die *default-class* hat einen niedrigeren Durchsatz als die Klassen *prec2* oder *prec3*, obwohl ihre zur Verfügung stehende MDRR-Priorität größer ist als die der beiden anderen Klassen.

Scheinbar wird die *class-default* Klasse im Vergleich zu den anderen Klassen benachteiligt. Im letzten Jahr wurden MDRR-Tests auch auf der QOC48LC Engine 4+ durchgeführt. Auf diesen Karten konnte das Phänomen, dass die *default-class* schlechter behandelt wird, nicht festgestellt werden. Wir befragten Cisco diesbezüglich, eine Antwort steht jedoch noch aus.

### 2.6.4 Klasse prec1 sendet etwas weniger als Mindestbandbreite

Dieser letzte MDRR-Test soll zeigen, ob eine Klasse ihren Verkehr bis zu ihrer konfigurierten Mindestbandbreite mittels MDRR verlustfrei senden kann, auch dann, wenn Überlast am Interface vorliegt. Die erlaubte Mindestbandbreite für die Klasse *prec1* beträgt ca. 36.55 Mbps. Demzufolge sollte ein Strom mit einer Senderate von 35Mbps verlustfrei durchgehen. Der Test zeigt, dass zwar ein paar Pakete (301 Pakete) in den ersten 2 Sendesekunden verloren gingen, anschließend traten jedoch keine weiteren Verluste mehr auf. Mit der *default-class* verhält es sich genauso. Einen Strom mit 35Mbps bekommt diese Klasse (bis auf die ersten 2 Sekunden) verlustfrei durch, auch dann, wenn Überlast vorliegt.

# 3 Zusammenfassung

In den hier vorliegenden Tests wurde das Performance- und CoS-Verhalten der Linecards vom Typ Cisco 12000 Series Four-Port OC-3c/STM-1c POS/SDH ISE Line Card, Long Reach untersucht. Die Karten wurden als Leihstellung durch Cisco zur Verfügung gestellt.

Die durchgeführten Tests richteten sich nach Möglichkeit nach einem in Zusammenarbeit mit Cisco, dem NOC Stuttgart und dem G-WiN-Labor erstellten Testplan. Auf Grund der nur kurzen Zeit, in der das vollständige Testequipment zur Verfügung stand, und wegen Verzögerungen im Testablauf durch Schwierigkeiten bei der Testdurchführung konnten nicht alle Punkte des Testplans in der gewünschten Weise durchgeführt werden.

Im Einzelnen waren Tests in folgenden Bereichen vorgesehen:

- Performance Tests IPv4
- Policing mit CAR
- Traffic Shaping
- WRED, MDRR
- Performance Tests IPv6

Gemäß des vorgegebenen Plans wurden die IPv4 Performance Tests und die Policing Tests vollständig durchgeführt. Die Shaping Tests konnten teilweise zusammengefasst werden, da sich die Ergebnisse als zufriedenstellend erwiesen haben. MDRR und WRED konnten lediglich auf Funktionalität überprüft werden, einzelne Parameter wurden nicht näher betrachtet. Die Überprüfung der IPv6-Performance wurde im Testzeitraum nicht mehr durchgeführt.

Alle Tests fanden mit Unicast Verkehr statt, das Verhalten mit Multicast Verkehr wurde nicht überprüft.

Vorherige Tests mit Engine 0 - OC3-Karten zeigten, dass Policing via CAR zu erheblichen CPU-Lasterhöhungen auf der Linecard führt. In der hier getesteten Engine 3-Version kann dies nicht mehr festgestellt werden, die untersuchten Eigenschaften werden auf der Karte durch entsprechende Asics abgedeckt. Die Funktionalität von Shaping, Policing und WRED kann als zufriedenstellend bewertet werden, alle Tests verliefen im wesentlichen erwartungsgemäß. Einige Fragen blieben jedoch von Cisco bisher unbeantwortet, wie z.B. die Benachteiligung der class-default bei MDRR, die relativ hohe Abweichung vom erwarteten Durchsatz bei der Aufteilung ungenutzter Bandbreite gemäß der eingestellten MDRR-Prioritäten auf andere Klassen und die nicht gleiche Behandlung identischer Ströme bei Policing/Shaping. Die nach den Tests getroffenen Aussagen seitens Cisco können nicht mehr verifiziert werden, da die Karten wieder zurückgesendet werden mussten. Für die derzeitigen Anforderungen des DFN-Vereins im G-WiN erscheint die Karte im Edge-Bereich als einsetzbar.

# Testergebnisse 4GE-SFP-LC

**Cisco 12000 Series Four-Port GE ISE Line Card, with Multimode GBics (GLC-SX-MM)**

(Version 1.0)

(G-WiN-Labor)

G-WiN-Labor
Regionales Rechenzentrum Erlangen (RRZE)
Martensstr. 1
91058 Erlangen
e-Mail: g-lab@rrze.uni-erlangen.de

18. Oktober 2004

# 1 Vorbemerkung

Die zu testenden Linecards vom Typ Cisco 4GE-SFP LC (ISE), Engine 3 sollen im Egde-Bereich im G-WiN zum Einsatz kommen. An dieser Stelle im Netz ist es wünschenswert, differenzierte Vertragsmodelle mit dem Kunden aushandeln zu können. Um diese Vereinbarungen einzuhalten, muß das zu testende Equipment verschiedene Eigenschaften mitbringen, wie beispielsweise die Möglichkeit, Verkehr zu begrenzen oder Verkehrsklassen zu unterscheiden und zu behandeln.

Getestet wurde die Linecard 4GE-SFP-LC (ISE), Engine 3. Eingesetzt wurde bei diesen Tests die IOS-Version 12.0(26)S. Für die Performanztests wurde kurzfristig auch die Version 12.0 (25)S2 verwendet, da Cisco hier eine Verbesserung der Ergebnisse vermutete. Da dies jedoch nicht zutraf und IPv6 Multicast (ein zwingend benötigtes Feature) in dieser Version nicht möglich ist, wurden die Tests schließlich wieder mit der neueren Version 12.0(26)S fortgesetzt.

Die getestete Linecards des Typs 4GE-SFP-LC (ISE) hatten die Hardware Revision Nummern 800-22811-02revB0 und 800-22811-03revA0.

Die Ausgabe  der Versionsnummern des getesteten Routers (12410) zeigt folgendes:

```
Cisco Internetwork Operating System Software
IOS (tm) GS Software (C12KPRP-P-M), Version 12.0(26)S, EARLY DEPLOYMENT
RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Mon 25-Aug-03 12:11 by nmasa
Image text-base: 0x00010000, data-base: 0x032DB000

ROM: System Bootstrap, Version 12.0(20020627:181338) [rarcher-CSCdx94605
5], DEVELOPMENT SOFTWARE
BOOTLDR: GS Software (C12KPRP-BOOT-M), Version 12.0(21.4)S3, EARLY
DEPLOYMENT MAINTENANCE INTERIM SOFTWARE

c4101 uptime is 3 weeks, 6 days, 1 hour, 7 minutes
System returned to ROM by reload
System image file is "tftp://131.188.81.54/c12kprp-p-mz.120-26.S.bin"

cisco 12410/PRP (MPC7450) processor (revision 0x00) with 524288K bytes of
memory.
MPC7450 CPU at 665Mhz, Rev 2.1, 256KB L2, 2048KB L3 Cache
Last reset from sw reset

1 Route Processor Card
2 Clock Scheduler Cards
5 Switch Fabric Cards
2 OC48 POS controllers (2 POS).
1 OC192 POS controller (1 POS).
1 Single Port Gigabit Ethernet/IEEE 802.3z controller (1 GigabitEthernet).
1 Four Port Gigabit Ethernet/IEEE 802.3z controller (4 GigabitEthernet).
2 Ethernet/IEEE 802.3 interface(s)
5 GigabitEthernet/IEEE 802.3 interface(s)
```

```
3 Packet over SONET network interface(s)
2043K bytes of non-volatile configuration memory.

125440K bytes of ATA PCMCIA card at slot 0 (Sector size 512 bytes).
65536K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x2102
```

# 2 Testergebnisse

## 2.1 Testaufbau

Der verwendete Testaufbau war, wenn im Text nicht anderweitig beschrieben, bei allen Tests identisch und ist in Abbildung 1 dargestellt. Zur Verfügung standen zwei Router der Marke Cisco der 1200er Baureihe. Hierbei handelt es sich um einen Cisco 12416 und um einen Cisco 12410.

Die für die Tests relevante Bestückung mit Linecards war auf beiden Routern identisch, jeweils zwei OC48-Linecards und jeweils eine Testkarte (4-fach GE Engine 3) waren eingebaut. Der Verkehr über die oberen beiden GE-Strecken (GE1/0 und GE1/1) wird dabei von den Agilent-Ports 102/2 und 101/2 erzeugt und empfangen, der Verkehr über die anderen beiden GE-Strecken verläuft über die Ports 102/1 und 101/1 des Agilents.

Die Ports der Testkarten wurden mit verschiedenen betriebsrelevanten Parametern konfiguriert, die während der gesamten Tests unverändert blieben.

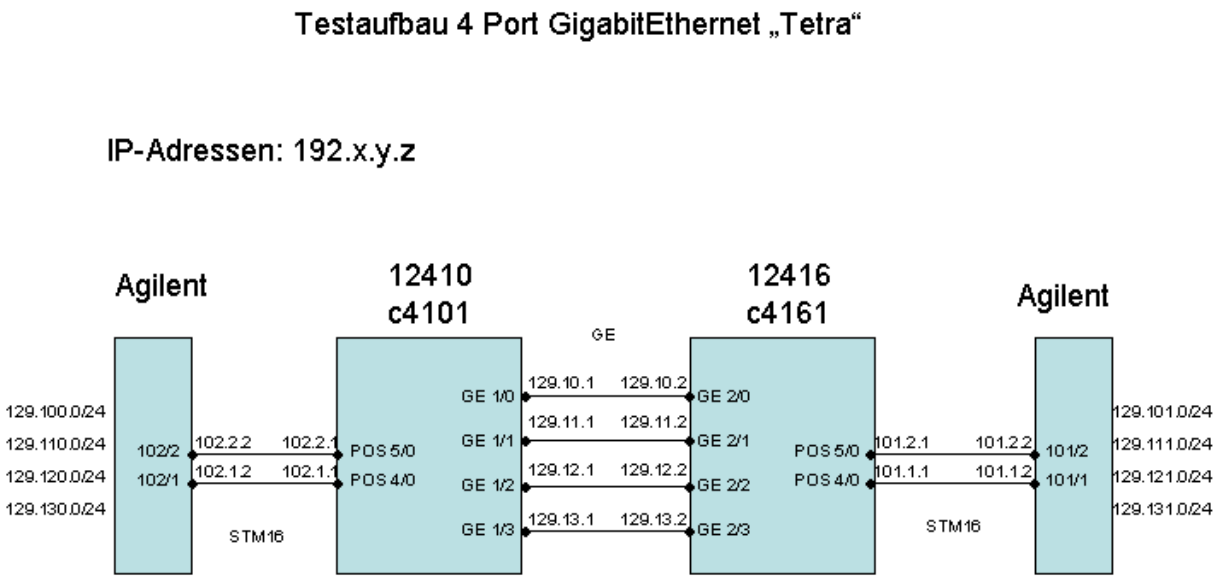



*Abbildung 1: Testaufbau*

Die GE-Interfaces sind bei den Tests wie folgt konfiguriert:

```
ip multicast-routing distributed
!
```

```
interface GigabitEthernet1/0
 mtu 4470
 ip address 192.129.10.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 no cdp enable
!

ip flow-sampling-mode packet-interval 100
```

## 2.2 Test-Verkehr

Um für die Verkehrsgenerierung bei den Tests eine möglichst betriebsnahe IP-Paketgrößenverteilung zu verwenden, wurde zunächst eine Analyse über 18 Stunden des G-WiN Verkehrs zwischen dem CR-Erlangen1 und dem AR-Erlangen1 durchgeführt.
Dabei wurde folgende IP-Paketverteilung ermittelt:

| Bucket | Paketlänge [Byte] | Prozentsatz |
|---|---|---|
| 1 | 40 | 18.84 |
| 2 | 1500 | 13.46 |
| 3 | 52 | 11.77 |
| 4 | 1420 | 7.28 |
| 5 | 1492 | 6.30 |
| 6 | 48 | 4.94 |
| Rest | Durchschnitt: 348 | 37.42 |

Die Werte für die Paketlänge 1-6 entsprechen jeweils exakt einer Paketgröße, beim Rest ist es der Mittelwert der Paketgröße. D.h. 63% des Verkehrs besteht lediglich aus sechs verschiedenen Paketgrößen, der restl. Teil hat eine durchschnittliche Paketgröße von 348 Bytes.

Bei der Durchführung der Tests wurde ein Routertester der Firma Agilent (SW 6.1.2.3D 6.2 Ipv6 beta) als Verkehrsgenerator eingesetzt. Leider kann bei diesem Gerät nicht einfach ein Verkehrsstrom erzeugt werden, der der obigen Paketgrößenverteilung entsprechen würde. Vielmehr ist für jede Paketgröße ein separater Strom mit der entsprechenden Paketrate zu konfigurieren. Da eine große Menge einzelner Ströme den Aufwand der Tests stark vergrößert (jeder einzelne Strom muss stets an die spezielle Konfiguration angepasst werden), wurden für die Tests folgende zwei Paketverteilungen verwendet:

a) G-WiN IMIX

| Bucket | Paketlänge [Byte] | Prozentsatz |
|---|---|---|
| 1 | 40 | 19 |
| 2 | 1500 | 13 |
| 3 | 52 | 12 |

| | | |
|---|---|---|
| 4 | 1420 | 7 |
| 5 | 1492 | 6 |
| 6 | 48 | 5 |
| 7 | 41-400 | 30 |
| 8 | 401-1499 | 7 |

Wie beim eigentlichen G-WiN Verkehr werden die sechs häufigsten Paketgrößen mit ihrem entsprechenden Verkehrsvolumen direkt verwendet. Der restliche Teil wurde in zwei Bereiche aufgeteilt, um ungefähr auf die durchschnittliche gemessene Paketgröße des G-WiN Verkehrs zu gelangen. Der Bereich 41-400 Bytes bedeutet hierbei, dass alle Paketgrößen zwischen 41 und 400 Bytes gleich häufig auftreten.

b) Gleichverteilter Mix: 40 B-1500 B

Als vereinfachter Paketmix wurden für die QoS-Tests ein Mix an Paketen zwischen 40 und 1500 B Paketen verwendet, die jeweils alle gleichverteilt vorkommen.
Der G-WiN IMIX wurde lediglich für die Durchsatztests verwendet. Für alle anderen Tests wurde auf den für die Durchführbarkeit freundlicheren gleichverteilten Mix 40 B-1500 B zurückgegriffen.

Für alle Tests (außer es wird explizit erwähnt) werden UDP-Pakete vom Agilent Routertester (RT) erzeugt und über die zwei Testrouter geschickt. Die Verwendung von TCP-Verkehr macht bei diesen Tests keinen Unterschied, da der RT über keinen vollständigen TCP-Stack verfügt und deshalb keine Flusskontrolle stattfindet.

## 2.3 Baseline Performance Test 4GE-SFP-LC (ISE)

### 2.3.1 Durchsatztests

Bei den Durchsatztests stellte sich heraus, dass ein Port der einen GE-Karte defekt war. Aus diesem Grund konnten zunächst die Tests nur über die restlichen drei Ports durchgeführt werden. Nachdem eine Ersatzkarte von Cisco eintraf, wurde der maximal mögliche, verlustfreie Durchsatz für die beiden oben genannten Paketmixe auch über vier Ports wiederholt. Das Ergebnis ist in den folgenden zwei Tabellen zu sehen.

a) Verkehr über 3 Ports simultan (duplex):

| IP-Paketgröße [Byte] | *Theoretisch.* max. mögl. Durchsatz (100% auf einem GE-Interface am Agilent-RT) pro Port [pps] | *Gemessener* max. mögl. verlustfreier Paketdurchsatz Pro Port [pps] | In Prozent [%] |
|---|---|---|---|
| 48 | 1453488,30 | 1172991,00 | 80,70 |
| 64 | 1225490,10 | 1131076,30 | 92,30 |
| 88 | 992063,40 | 867187,50 | 87,41 |
| 128 | 753012,00 | 752757,30 | 99,97 |
| 256 | 425170,00 | 424715,90 | 99,89 |
| 512 | 227272,70 | 227163,40 | 99,95 |

| | | | |
|---|---|---|---|
| 768 | 155086,80 | 149967,70 | 96,70 |
| 1024 | 117702,40 | 114341,00 | 97,14 |
| 1500 | 81274,30 | 78249,30 | 96,28 |
| | | | |
| Mix: | | | |
| 40 B-1500 B | 154700,20 | 149421,50 | 96,59 |
| G-WiN IMIX | 216504,10 | 208717,80 | 96,40 |

b) Verkehr über 4 Ports simultan (duplex):

| IP-Paketgröße<br><br>[Byte] | *Theoretisch.* max. mögl. Durchsatz (max. nur 3 mal GE möglich!)<br><br>pro Port [pps] | *Gemessener* max. mögl. Verlustfreier Paketdurchsatz<br><br>Pro Port [pps] | In Prozent<br><br>[%] |
|---|---|---|---|
| Mix: | | | |
| 40 B-1500 B | 116025,15 | 114877,80 | 99,01 |
| G-WiN IMIX | 162378,075 | 159742,30 | 98,38 |

Da die GE-Karten laut Cisco-Dokumentation nur Verkehr von max. 3 mal GE-Linerate schaffen, hier aber über 4 Ports gleichzeitig Verkehr geschickt wird, entsprechen hier 100% auf einem der 4 Ports ¾ des theoretisch maximal möglichen Durchsatzes bei den Tests über 3 Ports.

Das Ergebnis zeigt, dass in keinem der durchgeführten Tests ein Durchsatz von 100% erzielt werden konnte. Ein großer Performanzeinschnitt ist vor allem bei den 88 B großen Paketen zu sehen, obwohl Cisco hier 100% verspricht.

## 2.4 Überlast

### 2.4.1 Überlastete Ports/überlastete LC: Wie verhalten sich die Ports?

Im folgenden wurde getestet, ob im Falle einer Überlast die Ports der GE-Karte gleich behandelt werden.

#### 2.4.1.1 Über 4 Ports Duplex-Verkehr senden

Über jeden der 4 Ports der GE-Karte wurde mit 975 Mbps (156651.6 frames/s) in beide Richtungen gesendet. Der erzielte Durchsatz schwankte dabei um bis zu 74 Mbps[1], was folgende Tabelle zeigt:

| | gemessener Durchsatz [Mbps] |
|---|---|
| Gig1/0 | 752 |

[1] Die in diesem Testbericht angegebenen Durchsätze in Mbps entsprechen, wenn nicht anders vermerkt, jeweils IP plus PPP-Overhead.

| | |
|---|---|
| Gig1/1 | 678 |
| Gig1/2 | 752 |
| Gig1/3 | 678 |

D.h. der Durchsatz ist nicht gleichverteilt bei Überlast, was sich auch in den folgenden Tests zeigt.

#### 2.4.1.2 Über 3 Ports Duplex-Verkehr senden

Über 3 Ports der GE-Karte wurde mit 1000 Mbps (160668.3 frames/s) in beide Richtungen gesendet. Der erzielte Durchsatz schwankte dabei um bis zu 100 Mbps, was folgende Tabelle zeigt:

| | gemessener Durchsatz [Mbps] |
|---|---|
| Gig1/0 | -- |
| Gig1/1 | 962 |
| Gig1/2 | 880 |
| Gig1/3 | 859 |

#### 2.4.1.3 Über 2 Ports Duplex-Verkehr senden

Über 2 Ports der GE-Karte wurde mit 1500 Mbps (241002.5 frames/s) in beide Richtungen gesendet. Manchmal verteilte sich der Gesamtdurchsatz auf die beiden Ports gleichmäßig, in anderen Fällen schwankte er aber auch um bis zu 50 Mbps:

| | gemessener Durchsatz [Mbps] |
|---|---|
| Gig1/0 | -- |
| Gig1/1 | 958 |
| Gig1/2 | 937 |
| Gig1/3 | -- |

Ergebnis:
In einer Überlastsituation kann der Durchsatz der einzelnen Ports um bis zu 100 Mbps schwanken! Wir befragten Cisco dies bezüglich und bekamen als Antwort:

*[rschoen, 14Jan2004]*
*> In that scenario you are overloading the PLIM (meaning over about 3.2Mpps)*
*> and you are not doing PLIM QoS you can get some strange variation in packet*
*> rates through the ports. This is usually because you exhaust all of the*
*> buffers in the plim queue manager and you refill them in a time-cycle that*
*> makes one port run hotter than others. For reference remember the architecture*
*> of the line card (attached)*

PLIM-QoS kann eingesetzt werden, um bei mehreren Klassen eine Klasse bevorzugt zu behandeln. Dabei wird die bevorzugte Klasse einer *high-priority Queue* zugewiesen. Bei einem Treffen mit Cisco hier in Erlangen befragten wir Cisco, wie man in unserem Fall, bei

dem alle Pakete gleichbehandelt werden sollen, also keine unterschiedlichen Klassen vorhanden sind, PLIM-QoS eingesetzt werden kann. Hier meinte Cisco, dass PLIM-QoS wohl dann nicht anwendbar sei. Weitere Informationen bekamen wir nicht.

### 2.4.2 Überlastete LC, werden auch Ports beeinträchtigt, die nur eine geringe Verkehrslast verursachen?

Bei diesem Test soll untersucht werden, wie sich der Verkehr über einen relativ gering belasteten Port verhält, wenn die GE-Karte insgesamt (durch die anderen Ports) überlastet wird. Da die GE-Karte nur 3 mal GE an Bandbreite schafft, kann die Überlastsituation der GE-Karte relativ leicht hervorgerufen werden.

Beispiel:
Vier Kunden sind jeweils über einen der 4 GE-Ports an derselben LC angeschlossen. Anhand der Durchsatztests (siehe Kapitel 2.3.1) gehen über 4 Ports maximal 4 mal 114877,80 frames/s (=715 Mbps) an Bandbreite verlustfrei durch (mit dem Paketmix 40 B-1500 B). Kann man die Benutzer nun so beschränken, dass keiner mehr als die erlaubten 715 Mbps Verkehr erzeugt oder empfängt? Zum Beispiel, könnte man alle Nutzer auf 710 Mbps z.B. mittels Shaping oder Policing begrenzen, so dass ein Nutzer, der mehr als 715 Mbps sendet und dadurch potentiell die LC überlastet, keinen negativen Einfluss auf die anderen Nutzer ausübt?

Test:
Auf allen 4 Ports der GE-LC am c4101 wird Shaping **ausgangsseitig** auf 710 Mbps aktiviert.

a) **LC ist nicht überlastet**: Zieht 1 Kunde *K* (ge1/0) beispielsweise 850 Mbps über die GE-Karte aus dem G-WiN, die anderen 3 jeweils 650 Mbps (c4101 -> c4161), so ist die Karte noch nicht überlastet (850 Mbps + 3*650 Mbps = 2800 Mbps < 4*715 Mbps = 2860 Mbps). Der Kunde *K* wird korrekt auf die 710 Mbps begrenzt, die anderen Kunden haben keine Paketverluste.
b) **LC ist überlastet**: Zieht 1 Kunde *K* nach wie vor 850 Mbps über die GE-Karte, die anderen 3 hingegen jeweils 700 Mbps (c4101 -> c4161), so ist die Karte überlastet (850 Mbps + 3*700 Mbps = 2950 Mbps > 4*715 Mbps = 2860 Mbps). Der Kunde *K* wird zwar auf die 710 Mbps begrenzt, aber zusätzlich treten Paketverluste[2] an Port ge1/1 auf. Der Durchsatz dort wird auf 666 Mbps reduziert. Somit kann durch die Überlast der LC ein Kunde negativ beeinflusst werden, obwohl dieser unter seinem maximal erlaubten Durchsatz bleibt!

Wir fragten bei Cisco nach, ob dieses Problem umgangen werden kann. Zuerst meinte Cisco, dass die Paketverluste auf dem Port ge1/1 deshalb auftreten, da der überlastete Port ge1/0 den Paketpuffer von ge1/1 mit ausschöpft, da Shaping die Pakete nicht (unmittelbar) verwirft, sondern erst mal versucht zwischenzuspeichern. Da jedoch die Puffer der Ports jeder für sich mittels dem Befehl *queue-limit* begrenzt waren, konnte sich der überlastete Port keinen Puffer von den anderen Ports nehmen. Auch Policing (anstelle von Shaping), bei dem alle Pakete, die der konfigurierten Bandbreiten-Policy nicht entsprechen sofort verworfen werden, führte zu keiner Verbesserung. Nach mehreren Diskussionen gab Cisco schließlich zur Antwort, dass der negative Einfluss auf Ports mit geringem Verkehr in dieser Überlastsituation aufgrund der Architektur des Routers **nicht** vermieden werden kann.

[2] output queue drops an ge1/0, input errors/ignored errors an pos5/0

Der andere Fall, Karte ist überlastet, 1 Kunde **sendet** zu viel Verkehr zur GE-Karte, Shaping ist **eingangsseitig** auf der GE-Karte aktiviert, führt hingegen zu keinen Problemen. Hier wird der Verkehr, der über den 710 Mbps liegt am Eingang der LC verworfen, so dass die Karte erst gar nicht überlastet wird. Andere Ports werden nicht beeinflusst.

## 2.5 Rate-Limiting using CAR

Zuerst konnte in einem einfachen Test gezeigt werden, dass Policing via CAR prinzipiell sowohl am Ein- als auch am Ausgang eines GE-Ports möglich ist. Als nächstes wurde dann die Exaktheit bei der Einhaltung der konfigurierten Policing-Bandbreiten überprüft und die Granularität ermittelt. Dabei wurde stets über alle 4 Ports duplex gesendet mit dem Paketmix 40 B-1500 B und einer Senderate von 710 Mbps. Am Port 0 der GE-LC auf dem c4101 wurden verschiedene Policingwerte für den gesamten Ausgangsverkehr konfiguriert und der erzielte Durchsatz ermittelt. Das Ergebnis ist in folgender Tabelle dargestellt.

| Konf. Wert [kbps] | Gemessener Durchsatz [packets/s] | Gemessener Durchsatz (IP+PPP) [Mbps] | Min Delay [us] | Max Delay [us] | Avg Delay [us] |
|---|---|---|---|---|---|
| 64 | 312-314 | 0,0746 | 25,7 | 340,6 | 108,3 |
| 128 | 311-318 | 0,148 | 25,7 | 340,6 | 108,3 |
| 256 | 560-570 | 0,292 | 25,7 | 340,6 | 108,3 |
| 512 | 1000-1017 | 0,5767 | 25,5 | 384,2 | 108,3 |
| 1024 | 1736-1759 | 1,135[3] | 25,3 | 384,3 | 107,8 |
| 2048 | 2898-2964 | 2,2357 | 25,5 | 375,9 | 106,6 |
| 4096 | 4638-4711 | 4,3947 | 25,5 | 383,6 | 107,4 |
| 8192 | 7054-7122 | 8,6457 | 25,4 | 386,5 | 107,4 |
| 16384 | 9947-10062 | 17,024 | 25,7 | 392,9 | 109,3 |
| 32768 | 13156-13321 | 33,614 | 25,6 | 394,9 | 111,9 |
| 65536 | 17127-17371 | 66,636 | 25,6 | 403,9 | 115,5 |
| 131072 | 27809-28072 | 132,855 | 25,9 | 411,3 | 120,0 |
| 262144 | 48153-48438 | 265,995 | 25,3 | 575,7 | 123,9 |
| 524288 | 88307-88541 | 529,929 | 25,9 | 584,0 | 128,3 |

*Tabelle 1: Wie exakt werden die konf. Policingwerte eingehalten*

| Konf. Wert [kbps] | Gemessener Durchsatz [packets/s] | Gemessener Durchsatz (IP+PPP) [Mbps] | Min Delay [us] | Max Delay [us] | Avg Delay [us] |
|---|---|---|---|---|---|
| 524352 | 88267-88544 | 529,994 | 26,05 | 576,6 | 128,0 |
| 524416 | 88294-88640 | 530,058 | 25,9 | 582,7 | 127,8 |
| 524480 | 88335-88599 | 530,123 | 26,2 | 583,6 | 127,7 |
| 524544 | 88339-88611 | 530,188 | 26,1 | 577,7 | 127,8 |
| 524608 | 88342-88704 | 531,506 | 26,0 | 583,8 | 127,7 |

*Tabelle 2: Granularität in 64kbps Schritten (kleinster Abstand)*

[3] Bei diesem Test ging 1 Paket über das Interface ge 1/1 verloren.

Bis auf einen Einzelfall kam es zu keinen Paketverlusten auf den ungepolicten Strömen. Die Granularität ist im gemessenen Bandbreitenbereich sehr fein.

Beim Konfigurieren der Policingwerte wurde festgestellt, dass die maximale konfigurierbare Burstsize bei 16777215 Bytes liegt, was folgendes Beispiel zeigt:

```
c4101(config-if)#rate-limit output 65536000 12288000 24576000 conform-action
                 transmit exceed-action drop
                 Adjusting exceed burst to match hardware limit of 16777215 bytes

c4101(config-if)#
```

## 2.6 Rate-Limiting using Shaping

Ein erster Test zeigte auch hier, dass prinzipiell Shaping sowohl am Ein- als auch am Ausgangsinterface möglich ist. Als nächstes wurde dann die Exaktheit bei der Einhaltung der konfigurierten Shaping-Bandbreiten überprüft und die Granularität ermittelt. Dabei wurde stets über 3 Ports duplex gesendet mit dem Paketmix 40 B-1500 B und einer Senderate von 920 Mbps. Der vierte Port der GE-LC war zu diesem Zeitpunkt defekt. Einige Zeit später bekamen wir von Cisco schließlich ein Ersatzboard fuer die GE-LC. Am Port 1 der GE-LC auf dem c4101 wurden verschiedene Shapingwerte für den gesamten Ausgangsverkehr konfiguriert und der erzielte Durchsatz ermittelt. Wichtig ist, dass beim Shaping auch stets ein *queue-limit* konfiguriert wird, da andernfalls der Paketspeicher der gesamten Karte von einem Interface in Anspruch genommen werden kann, sobald dort Pakete durch das Shaping zwischengespeichert werden müssen.

```
!
policy-map shapeGEtox
  class ANY
    shape average 128000
    queue-limit 200 packets

!
interface GigabitEthernet1/1
service-policy output shapeGEtox
```

Das Ergebnis der Shapingtests ist in folgender Tabelle dargestellt.

| Konfig. Shapingrate [kbps] | Gemessener Durchsatz [packets/s] | | Gemessener Durchsatz (IP+PPP) [Mbps] | | Delay [ms] | |
|---|---|---|---|---|---|---|
| | **min** | **max** | **min** | **max** | **min** | **max** |
| 64 | 7 | 10 | 0,06 | 0,07 | 23098 | 24422 |
| 128 | 14 | 19 | 0,13 | 0,13 | 12400 | |
| 256 | 32 | 38 | 0,25 | 0,26 | 5743 | 6247 |
| 512 | 66 | 72 | 0,51 | 0,53 | 2950 | 3007 |
| 1024 | 127 | 143 | 1,03 | 1,04 | 1432 | 1511 |
| 2048 | 262 | 280 | 2,06 | 2,07 | 721 | 736 |
| 4096 | 532 | 555 | 4,13 | 4,14 | 370 | 379 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8192 | 1070 | 1102 | 8,25 | 8,27 | 180 | 188 |
| 16384 | 2166 | 2226 | 16,52 | 16,53 | 91 | 93 |
| 32768 | 4432 | 4552 | 33,05 | 33,06 | 44 | 45 |
| 65536 | 9689 | 9785 | 66,15 | 66,17 | 20 | 21 |
| 131072 | 20215 | 20469 | 132,37 | 132,39 | 9 | 10 |
| 262144 | 41480 | 41716 | 267,8 | 267,82 | 4,8 | 4,9 |
| 524288 | 84131 | 84265 | 529,67 | 529,69 | 2,4 | 2,5 |
| 786432 | 126709 | 126906 | 794,54 | 794,55 | 1,6 | 1,7 |
| 851968 | 137508 | 137776 | 860,78 | 860,8 | 1,5 | 1,6 |

*Tabelle 3: Wie exakt werden die konf. Shapingwerte eingehalten*

| Konfig. Shapingrate [kbps] | Gemessener Durchsatz [packets/s] | | Gemessener Durchsatz (IP+PPP) [Mbps] | | Delay [ms] | |
|---|---|---|---|---|---|---|
| | **min** | **max** | **min** | **max** | **min** | **max** |
| 524352 | 84108 | 84255 | 529,74 | 529,76 | 2,4 | 2,5 |
| 524416 | 84110 | 84316 | 529,8 | 529,81 | 2,4 | 2,5 |
| 524480 | 84142 | 84296 | 529,86 | 529,88 | 2,4 | 2,5 |
| 524544 | 84098 | 84342 | 529,93 | 529,96 | 2,4 | 2,5 |
| 524608 | 84191 | 84364 | 529,99 | 530,01 | 2,4 | 2,5 |
| 524672 | 84144 | 84334 | 530,06 | 530,08 | 2,4 | 2,5 |

*Tabelle 4: Granularität in 64kbps Schritten (kleinster Abstand)*

Auf den nicht-geshapten Strömen gingen keine Pakete verloren. Die Granularität ist im gemessenen Bandbreitenbereich sehr gut.

## 2.7 Lasttest mit Policing und Shaping

Nun werden mehrere Ströme gleichzeitig auf gig1/0 und gig1/1 unterschiedlich gepolict und auf gig1/2 und gig1/3 unterschiedlich geshapt. Kommt es zu Paketverlusten bei ungepolicten bzw. ungeshapten Strömen?

Die Konfiguration sah dabei wie folgt aus:

```
policy-map shapeGEto10M
  class ANY
    shape average 9600000
    queue-limit 200 packets

policy-map shapeGEto2M_20M7.9M
  class prec_5
    shape average 1984000
    queue-limit 200 packets
  class prec_0
    shape average 7872000
    queue-limit 200 packets
  class prec_2
```

```
    shape average 19968000
    queue-limit 200 packets


!
interface GigabitEthernet1/0
rate-limit  output  access-group  rate-limit  1  10000000  512000  1000000
conform-action transmit exceed-action drop
 rate-limit  output  access-group  rate-limit  2  10000000  512000  1000000
conform-action transmit exceed-action drop
rate-limit  output  access-group  rate-limit  3  10000000  512000  1000000
conform-action transmit exceed-action drop
rate-limit  output  access-group  rate-limit  4  10000000  512000  1000000
conform-action transmit exceed-action drop

!
interface GigabitEthernet1/1
rate-limit  output  access-group  rate-limit  1  10000000  512000  1000000
conform-action continue exceed-action drop
 rate-limit output access-group 151 1000000 512000 1000000 conform-action
transmit exceed-action drop

!
interface GigabitEthernet1/2
service-policy output shapeGEto10M
!
interface GigabitEthernet1/3
service-policy output shapeGEto2M_20M7.9M

!
access-list 151 permit tcp any any
access-list 151 deny   ip any any
access-list rate-limit 1 1
access-list rate-limit 2 2
access-list rate-limit 3 3
access-list rate-limit 4 4
```

Über jeden Port werden in Summe 710 Mbps (Mix: 40 B-1500 B) in beide Richtungen verschickt, d.h. die Karte ist bzgl. der maximalen Paketrate nicht überlastet, aber nahezu an ihrer Kapazitätsgrenze von 4 mal 715 Mbps ( siehe Durchsatztests, Kapitel 2.3.1).

| | Gesendet [Mbps] | Zu erwartender Durchsatz [Mbps] | Gemessener Durchsatz [Mbps] |
|---|---|---|---|
| **gig1/0 “policing”** | | | |
| Prec1 | 100 | 10 | 10,2 |
| Prec2 | 100 | 10 | 10,2 |
| Prec3 | 100 | 10 | 10,2 |
| Prec4 | 100 | 10 | 10,2 |
| Prec0 | 310 | 310 | 310 |
| **gig1/1 “policing”** | | | |
| Prec0 | 510 | 510 | 510 |
| Prec1 UDP<br>TCP | 100<br>100 | 5<br>1 | 5,17<br>1,01 |
| **gig1/2 “shaping”** | | | |

| | | | |
|---|---|---|---|
| Prec0 | 710 | 10 | 9,7 |
| **gig1/3 “shaping”** | | | |
| Prec0 | 100 | 7,8 | 7,9 |
| Prec2 | 100 | 19,9 | 20,17 |
| Prec5 | 100 | 1,9 | 2,0 |
| Prec1 | 410 | 410 | 410 |

Die Last der Haupt-CPU, die am Slot 4 und 5 lag dabei  bei ca. 0-1%, am Slot 1 mit dem GE-Interface bei 9-14%.

Es kam zu keine Verlusten auf den ungepolicten oder ungeshapten Strömen.
Es wird korrekt gepolict und geshapt.

## 2.8 MDRR

Zunächst wurden die MDRR-Tests analog zu den früheren Tests auf der 4fach OC3 Engine3 LC durchgeführt. Dabei wurde folgende Konfiguration verwendet:

```
policy-map MDRR
  class prec3_4
    bandwidth percent 20
    random-detect
    random-detect precedence 3 255 packets 511 packets 1
    random-detect precedence 4 255 packets 511 packets 1
  class prec_1
    bandwidth percent 25
    random-detect
    random-detect precedence 1 255 packets 511 packets 1
  class prec_2
    bandwidth percent 20
    random-detect
    random-detect precedence 2 255 packets 511 packets 1
  class prec_6
    bandwidth percent 10
    random-detect
    random-detect precedence 6 255 packets 511 packets 1
  class prec_5
    priority
     police cir 640000 bc 4470 be 4470 conform-action transmit  exceed-action drop
  class class-default
    random-detect
    random-detect precedence 0 255 packets 511 packets 1
    random-detect precedence 7 255 packets 511 packets 1
```

Das Ergebnis entsprach in etwa den Tests auf der 4fach OC3 LC. Nutzt eine Klasse ihre ihr zugesicherte Bandbreite nicht voll aus, so kann diese zwar von den anderen Klassen mit benutzt werden, jedoch entsprach die Aufteilung nicht den konfigurierten bandwidth-Werten. Damals als auch bei diesen Tests wurde uns von Cisco mitgeteilt, dass die Aufteilung nicht nur nach den bandwidth-Werten passiere, sondern auch nach sog. internen „Weight Bytes“. Bei diesen Tests bemerkte Cisco zudem, dass es auch daran liegen könnte, dass der default-Klasse keine Bandbreite zugewiesen sei. Wir sollten daher für die default-Klasse „bandwidth-remaining“ mit einem gewünschten minBW konfigurieren. Alternativ könne man auch den "bandwidth" Befehl verwenden und einen BW-Wert in % angeben.
Zunächst wurde daher der Befehl bandwidth-remaining ausprobiert:

```
class class-default
    random-detect
    random-detect precedence 0 255 packets 511 packets 1
    random-detect precedence 7 255 packets 511 packets 1
    bandwidth remaining percent 100
```

Bei den folgenden Tests wurde MDRR nun am Ausgang von gig1/1 aktiviert. An diesem Interface wurde eine Überlastsituation durch das gewählten Sendeprofil erzeugt. Die Klasse default-class wird durch Pakete mit Precedence Bit gleich 0 gekennzeichnet.

### 2.8.1 Alle Klassen senden mehr

Sendeprofil:
Rückrichtung "<-": über alle 4 Ports je 700 Mbps, Mix:40 B-1500 B, prec0
Hinrichtungen: gig1/0, gig1/2, gig1/3: je 500 Mbps, Mix:40 B-1500 B, prec0
gig1/1: prec0=1=2=3=6=260 Mbps (Summe: 1300 Mbps)
prec5=0.5 Mbps
Ergebnis:

| gig1/1 | Erzielter Durchsatz [Mbps] | Erzieltes Durchsatzverhältnis | Erwartetes Durchsatzverhältnis gemäß der konf. Bandwidth-Werte |
|---|---|---|---|
| Prec0 | 49,22 | 0,33 | 2,4 |
| Prec1 | 260,0 | 1,73 | 2,5 |
| Prec2 | 251,36 | 1,67 | 2 |
| Prec3 | 251,36 | 1,67 | 2 |
| Prec6 | 150,39 | 1 | 1 |

Keine Verluste auf anderen Strömen!
Starke Abweichung vom erwarteten Durchsatzverhältnis!

### 2.8.2 Prec0 sendet nichts

Sendeprofil:
Rückrichtung "<-": über alle 4 Ports je 700 Mbps, Mix:40 B-1500 B, prec0
Hinrichtungen: gig1/0, gig1/2, gig1/3: je 500 Mbps, Mix:40 B-1500 B, prec0
gig1/1: prec1=2=3=6=325 Mbps (Summe: 1300 Mbps)
prec5=0.5 Mbps
Ergebnis:

| gig1/1 | Erzielter Durchsatz [Mbps] | Erzieltes Durchsatzverhältnis | Erwartetes Durchsatzverhältnis gemäß der konf. Bandwidth-Werte |
|---|---|---|---|
| Prec0 | -- | -- | -- |
| Prec1 | 303,76 | 2,02 | 2,5 |
| Prec2 | 253,2 | 1,68 | 2 |
| Prec3 | 253,2 | 1,68 | 2 |
| Prec6 | 152,19 | 1 | 1 |

Keine Verluste auf anderen Strömen!

### 2.8.3 Prec1 sendet nichts

Sendeprofil:
Rückrichtung "<-": über alle 4 Ports je 700 Mbps, Mix:40 B-1500 B, prec0
Hinrichtungen: gig1/0, gig1/2, gig1/3: je 500 Mbps, Mix:40 B-1500 B, prec0
gig1/1: prec0=2=3=6=325 Mbps (Summe: 1300 Mbps)
prec5=0.5 Mbps

Ergebnis:

| gig1/1 | Erzielter Durchsatz [Mbps] | Erzieltes Durchsatzverhältnis | Erwartetes Durchsatzverhältnis gemäß der konf. Bandwidth-Werte |
|---|---|---|---|
| Prec0 | 114,25 | 0,53 | 2,4 |
| Prec1 | -- | -- | -- |
| Prec2 | 316,35 | 1,47 | 2 |
| Prec3 | 316,35 | 1,47 | 2 |
| Prec6 | 215,38 | 1 | 1 |

Keine Verluste auf anderen Strömen!
Starke Abweichung vom erwarteten Durchsatzverhältnis !

Der Befehl remaining-bandwidth löst das Problem also nicht!
Im folgenden wurde der bandwidth Befehl mit einem Wert in % der default-Klasse zugewiesen und die oben stehenden Tests wiederholt.

```
class class-default
   random-detect
   random-detect precedence 0 255 packets 511 packets 1
   random-detect precedence 7 255 packets 511 packets 1
   bandwidth percent 24
```

### 2.8.4 Alle Klassen senden mehr

Sendeprofil:
Rückrichtung "<-": über alle 4 Ports je 700 Mbps, Mix:40 B-1500 B, prec0
Hinrichtungen: gig1/0, gig1/2, gig1/3: je 500 Mbps, Mix:40 B-1500 B, prec0
gig1/1: prec0=1=2=3=6=260 Mbps (Summe: 1300 Mbps)
prec5=0.5 Mbps

Ergebnis:

| gig1/1 | Erzielter Durchsatz [Mbps] | Erzieltes Durchsatzverhältnis | Erwartetes Durchsatzverhältnis gemäß der konf. Bandwidth-Werte |
|---|---|---|---|
| Prec0 | 228,49 | 2,26 | 2,4 |
| Prec1 | 228,54 | 2,26 | 2,5 |
| Prec2 | 202,11 | 2,00 | 2 |
| Prec3 | 202,11 | 2,00 | 2 |
| Prec6 | 101,11 | 1 | 1 |

Keine Verluste auf anderen Strömen!

### 2.8.5 Prec0 sendet nichts

Sendeprofil:
Rückrichtung "<-": über alle 4 Ports je 700 Mbps, Mix:40 B-1500 B, prec0
Hinrichtungen: gig1/0, gig1/2, gig1/3: je 500 Mbps, Mix:40 B-1500 B, prec0
gig1/1: prec1=2=3=6=325 Mbps (Summe: 1300 Mbps)
prec5=0.5 Mbps

Ergebnis:

| gig1/1 | Erzielter Durchsatz [Mbps] | Erzieltes Durchsatzverhältnis | Erwartetes Durchsatzverhältnis gemäß der konf. Bandwidth-Werte |
|---|---|---|---|
| Prec0 | -- | -- | -- |
| Prec1 | 320,8 | 2,50 | 2,5 |
| Prec2 | 256,61 | 2,00 | 2 |
| Prec3 | 256,61 | 2,00 | 2 |
| Prec6 | 128,34 | 1 | 1 |

Keine Verluste auf anderen Strömen!

### 2.8.6 Prec1 sendet nichts

Sendeprofil:
Rückrichtung "<-": über alle 4 Ports je 700 Mbps, Mix:40 B-1500 B, prec0
Hinrichtungen: gig1/0, gig1/2, gig1/3: je 500 Mbps, Mix:40 B-1500 B, prec0
gig1/1: prec0=2=3=6=325 Mbps (Summe: 1300 Mbps)
prec5=0.5 Mbps

Ergebnis:

| gig1/1 | Erzielter Durchsatz [Mbps] | Erzieltes Durchsatzverhältnis | Erwartetes Durchsatzverhältnis gemäß der konf. Bandwidth-Werte |
|---|---|---|---|
| Prec0 | 312,1 | 2,40 | 2,4 |
| Prec1 | -- | -- | -- |
| Prec2 | 260,09 | 2,00 | 2 |
| Prec3 | 260,09 | 2,00 | 2 |
| Prec6 | 130,08 | 1 | 1 |

Keine Verluste auf anderen Strömen!

### 2.8.7 Prec 1 sendet weniger als erlaubt, bringt prec1 seinen Verkehr verlustfrei durch?

Sendeprofil:
Rückrichtung "<-": über alle 4 Ports je 700 Mbps, Mix:40 B-1500 B, prec0
Hinrichtungen: gig1/0, gig1/2, gig1/3: je 500 Mbps, Mix:40 B-1500 B, prec0
gig1/1: prec0=2=3=6=269 Mbps (Summe: 1075 Mbps)
prec1=225 Mbps
prec5=0.5 Mbps

Ergebnis:

| gig1/1 | Erzielter Durchsatz [Mbps] | Erzieltes Durchsatzverhältnis | Erwartetes Durchsatzverhältnis gemäß der konf. Bandwidth-Werte |
|---|---|---|---|
| Prec0 | 232,1 | 2,30 | 2,4 |
| Prec1 | 225 keine Verluste! | keine Verluste | keine Verluste |
| Prec2 | 202,09 | 2,00 | 2 |
| Prec3 | 202,09 | 2,00 | 2 |
| Prec6 | 101,08 | 1 | 1 |

Keine Verluste auf anderen Strömen!

Mit diesem Befehl erhält man nun in ausreichender Genauigkeit den gewünschten Durchsatz für jede Klasse.

Wie auch schon bei den 4 fach OC3 Karten festgestellt wurde, kann mit MDRR nur 99% der Bandbreite explizit auf Klassen verteilt werden.

## 2.9 WRED

Um WRED zu testen wurde folgende WRED-Policy am 12410 auf gig1/0 am Ausgang aktiviert:

```
class-map match-any prec1-3
  match ip precedence 1
  match ip precedence 2
  match ip precedence 3

policy-map wred
  class prec1-3
    bandwidth percent 99
    random-detect
    random-detect precedence 1 1000 packets 2024 packets 1
    random-detect precedence 2 1400 packets 3448 packets 1
    random-detect precedence 3 1800 packets 5896 packets 1
```

Diese Policy sieht vor, dass im Falle einer Überlast am Port gig1/0 zuerst der prec1, dann der prec2 und zuletzt der prec3 Verkehr verworfen werden soll. Verwendet wurde wieder unser Standard Test-Setup aus Abbildung 1.

Gesendet wurden folgende Ströme:
auf alle 4 Rückrichtungen "<-": je 1 Flow (40 B-1500 B Paketmix) mit je 700 Mbps, prec0

über die Hinrichtungen "->" gig1/1, gig1/2 und gig1/3:
je 1 Flow (40 B-1500 B Paketmix) mit je 400 Mbps, prec0

über Hinrichtung gig1/0:
3 flows:
a) prec1: 50000 f/s = 311.2 Mbps
b) prec2 50000 f/s = 311.2 Mbps

c) prec3 variierend von 50000 f/s bis 190000 f/s

Der prec3 Verkehr wurde kontinuierlich gesteigert und somit die Überlast auf gig1/0 erhöht. Die Durchsätze für prec1 und prec2 wurden ermittelt:

Ergebnis:

CPU: main, slot 4, slot 5: 0-1%
slot1: 9-16%, im 1min durchschnitt: ca. 9-11%

| Senderate von prec3 | | Erzielter Durchsatz prec1 | prec2 | prec3 | Besonderheiten |
|---|---|---|---|---|---|
| [f/s] | [Mbps] | [Mbps] | [Mbps] | [Mbps] | |
| 50000 | 311,28 | 311,28 | 311,28 | 311,2 | keine Verluste |
| 60000 | 372,97 | 278,06 | 311,91 | 372,91 | Verluste auf prec1 |
| 62500 | 388,79 | 263,00 | 310,86 | 388,97 | kont. Verluste auf prec1,<br>6 Verluste am Anfang bei prec2 |
| 65000 | 404,71 | 246,72 | 311,39 | 404,75 | kont. Verluste auf prec1,<br>26 Verluste am Anfang bei prec2 |
| 67500 | 420,06 | 231,90 | 310,98 | 419,95 | kont. Verluste auf prec1 und prec2 |
| 70000 | 435,72 | 217,68 | 309,46 | 435,74 | kont. Verluste auf prec1 und prec2 |
| 80000 | 497,73 | 160,75 | 304,46 | 497,62 | kont. Verluste auf prec1 und prec2 |
| 100000 | 622,11 | 50,03 | 290,39 | 622,46 | kont. Verluste auf prec1 und prec2 und<br>112 Verluste am Anfang bei prec3 |
| 110000 | 684,48 | 0,21 | 278,38 | 684,26 | kont. Verluste auf prec1, prec2 und prec3 |
| 120000 | 747,15 | 0 | 232,08 | 730,80 | Durchsatz auf prec1=0 |
| 130000 | 809,06 | 0 | 189,05 | 773,81 | |
| 150000 | 933,87 | 0 | 112,97 | 849,89 | |
| 160000 | 996,72 | 0 | 91,37 | 871,34 | **Verluste an ge1/1** !<br>ignored an pos5/0,<br>Senderate “->”: 2820,27 Mbps<br>Ist max. Kapazität der LC erreicht? |
| 160000 | 995,53 | 0 | 77,81 | 885,05 | Senderate über ge1/3 reduziert auf 100 Mbps -> keine Verluste mehr an ge1/1 |
| 180000 | 1120 | 0 | 12,78 | 950,09 | Senderate über ge1/3: 100 Mbps |
| 188000 | 1170,01 | 0 | 0 | 962,90 | Senderate über ge1/3: 100 Mbps<br>Durchsatz von prec2=0 |
| 190000 | 1182,55 | 0 | 0 | 962,84 | Verluste an ge1/1 (40-100p/s), ignored an pos5/0<br>Senderate "->": 2703,4 Mbps<br>Ist max. Kapazität der LC erreicht? |
| 190000 | 1182,40 | 0 | 0 | 962,87 | Senderate über ge1/3 reduziert auf 50 Mbps !<br>-> keine Verluste an ge1/1mehr<br>Solange Senderate über ge1/3 < 96 Mbps |

| | | | | |
|---|---|---|---|---|
| | | | | kommt es zu keinen Verlusten bei ge1/1 |

Ergebnis:
WRED an sich funktioniert. Es wird wie erwartet zuerst prec1, dann prec2 und zuletzt prec3 Verkehr verworfen.
Ist die Überlast sehr hoch am gig1/0 (WRED muss viel verwerfen), so scheint die LC eine geringeren Gesamtdurchsatz zu vertragen. Dies erkennt man daran, dass auf einem anderen Port (hier gig1/1) ebenfalls Pakete verloren gehen, obwohl dort die Verkehrslast gering ist und die gesamte Durchsatzlast der LC unter dem eigentlichen max. möglichen, verlustfreien Durchsatz von ca. 715 Mbps je Port (also 715 Mbps*4=2860 Mbps) liegt.
Verringert man die Gesamtlast der LC in Hinrichtung (hier Verkehr über gig1/3 von 400 auf 100 bzw. 50 Mbps), lässt aber die Senderate und die WRED-Konfig über Port gig1/0 gleich, so verwirft WRED wie erwartet und keine anderen Ports der LC werden beeinflusst.

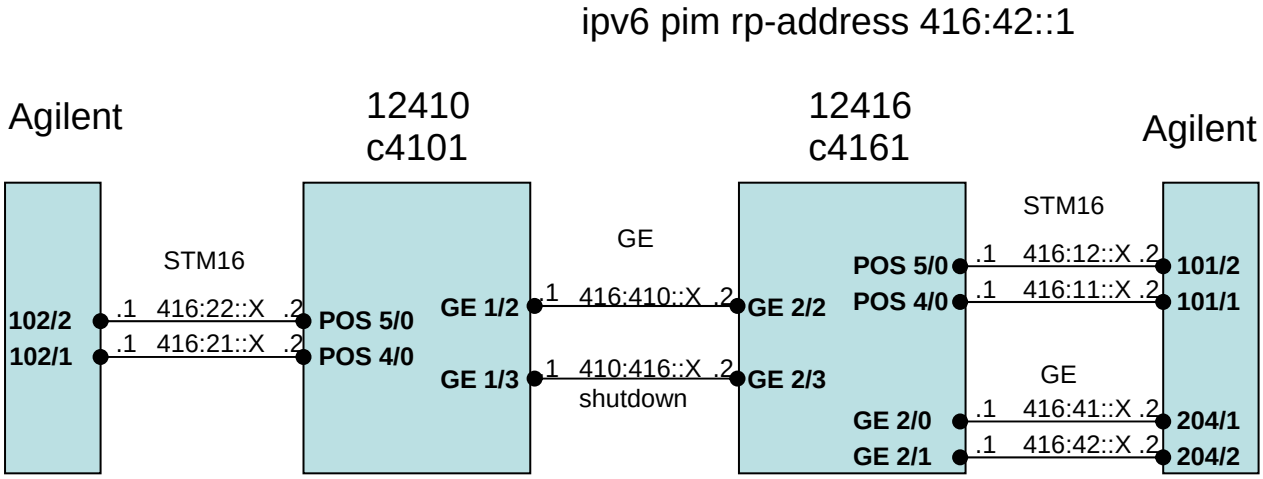
Cisco vermutete dass die durch WRED verringerte Gesamtkapazität der LC daran liegen könnte, dass nur an einem Port WRED aktiviert war. Aus diesem Grund wurde der obige Test wiederholt, jedoch ohne Verbesserung. Auch Änderungen an der WRED-Konfiguration führten zu keinem besseren Ergebnis. Nach weiteren Tests wurde festgestellt, dass je niedriger die Überlast am Interface gig1/0 ist, desto näher kommt man an die maximal mögliche Gesamtlast von 2860 Mbps. Folgende Tabelle zeigt die Abhängigkeit der max. möglichen Gesamtlast der LC (also ohne dass Verluste auf den Interfaces gig1/1, gig1/2 oder gig1/3 entstehen) von der Überlast an gig1/0. WRED ist dabei auf allen 4 Ports aktiviert, jedoch kommt WRED nur auf Port 0 zum Tragen, da nur dort auch eine Überlastsituation vorliegt.

| Gesendet über gig1/0 **zunehmende Überlast !** [Mbps] | Gesendet über gig1/1, gig1/2, gig1/3, je [Mbps] | Gesamtlast auf LC [Mbps] | Traten Verluste auf gig1/1 auf, d.h. war die LC überlastet? |
|---|---|---|---|
| **1040** | 600 | 2840 | nein |
| **1050** | 600 | 2850 | ja |
| **1100** | 578 | 2834 | ja |
| **1100** | 577 | 2831 | nein |
| **1200** | 540 | 2820 | ja |
| **1200** | 539 | 2817 | nein |
| **1300** | 502 | 2806 | ja |
| **1300** | 500 | 2800 | nein |
| **1400** | 463 | 2789 | ja |
| **1400** | 462 | 2786 | nein |

Schlussfolgerung:
Anhand der Tests sieht man, dass der max. mögliche Gesamtdurchsatz der LC von ca. 2860 Mbps (mit unserem verwendeten Paketmix) um so stärker bei aktiviertem WRED sinkt, je höher die Überlast an einem Port ist.

## 2.10 IPv6 Multicast

### 2.10.1 Testaufbau




### 2.10.2 Testbeschreibung:

Auf dem Agilent Routertester lief eine frühe Beta Software. Dadurch war es nicht möglich, ausführliche Tests durchzuführen. Auch war es notwendig, allen Multicastverkehr vom Agilent an die "Link-Local-Adresse“ zu schicken, damit Cisco sie weiterleitet. Agilent ist dieses Problem bekannt.

Auf dem 12416 wurde ein RP eingerichtet, der statisch für alle IPv6 Multicastgruppen zuständig ist. Im Laufe des Tests wurden von verschiedenen Interfaces Multicastpakete geschickt und von den anderen empfangen. Allerdings kam es bei einer Senderaten von 100% MC-Traffic von einem GE-Port am 12416 zu Verlusten am 12410. Dies könnte an den Engine 2 STM16 Karten liegen. Wenn es ein Performance Problem der Karten ist, dann hätten die Verluste auch an den POS-Interfaces auf dem 12416 auftreten müssen.

Da im Labor, außer den alten STM4 und STM16 Engine 2 Karten, keinen neueren Karten zur Verfügung stehen, konnte diese Verluste nicht genauer untersucht werden.

Ein weiteres Problem während der Tests war, dass bei bestimmten Konfigurationen die CPU-Last auf 100% ging. Dieses konnte aber aus Zeitgründen nicht näher untersucht werden.

### 2.10.3 Fazit:

Prinzipiell ist IPv6 Multicast mit dem GE-Tetra Karten möglich. Allerdings sollte vor dem Betreiben der G-WiN-Router im Dual-Stack Betrieb unbedingt weitere Tests durchgeführt werden. Im Moment ist es nicht sicher, ob der IPv6-Traffic nicht den IPv4 beeinflussen würde. Hierzu müssten aber die im G-WiN eingesetzten Kartentypen auch im Labor verfügbar sein.

# 3 Zusammenfassung

In den hier vorliegenden Tests wurde das Performance-, CoS- und das Ipv6 Multicast Verhalten der Linecards vom Typ Cisco 12000 4GE-SFP LC (ISE), Engine 3 Line Card untersucht. Die Karten wurden als Leihstellung durch Cisco zur Verfügung gestellt.

Im Einzelnen wurden Tests in folgenden Bereichen durchgeführt:

- Performance Tests IPv4
- Verhalten bei Überlast
- Policing mit CAR
- Traffic Shaping
- WRED, MDRR
- IPv6 Multicast

Laut Datenblatt von Cisco soll die 4GE-SFP LC eine Performance von 2,5 Gbps für 40-46 Byte große Pakete und 3 Gbps für Pakete über 80 Bytes haben. Da die Karte jedoch 4 GE-Ports besitzt kann selbst bei großen Paketen relativ leicht eine Überlastsituation entstehen. Aus diesem Grund wurde neben den Performance-Tests insbesondere auch das Verhalten bei einer Überlastsituation untersucht:
Die Performance-Tests wurden mit einzelnen Paketgrößen, sowie mit 2 versch. Paketmixen über 3 bzw. 4 Ports duplex durchgeführt. Das Ergebnis zeigt, dass in keinem der durchgeführten Tests ein Durchsatz von 100% erzielt werden konnte. Ein großer Performanzeinschnitt ist vor allem bei den 88 B großen Paketen zu sehen, obwohl Cisco hier 100% bei der Verwendung von 3 der 4 Ports verspricht.
Das Verhalten der LC bei Überlast wurde im folgenden untersucht. Zunächst wurde festgestellt, dass über die Ports um bis zu 100 Mbps untersch. Durchsätze erzielt werden, d.h. ist die LC überlastet, scheinen manche Ports gegenüber anderen Ports bevorzugt behandelt zu werden. In einem weiteren Test wurde festgestellt, dass bei überlasteter LC ein gering verkehrsmäßig belasteter Port, an dem z.B. ein Kunde angeschlossen ist, negativ beeinflusst werden kann (Paketverluste). Wird die Überlast der LC durch zuviel Verkehr am Eingang der Tetra-LC hervorgerufen, so kann dieser neg. Einfluss auf einen gering belasteten Kundenport z.B. mittels Shaping verhindert werden. Im umgekehrten Fall, gibt es jedoch auch laut Cisco aufgrund der Archtiektur des Routers keine Lösung hierfür. Aus diesem Grund sollte die Tetra-LC nur mit 3 der 4 Ports belegt werden!
Die Tests mit Policing und Shaping verliefen erwartungsgemäß. Sowohl Policing als auch Shaping funktioniert prinzipiell sowohl am Ein- als auch am Ausgangsinterface der LC. Bis auf einen Einzelfall kam es zu keinen Paketverlusten auf den ungepolicten bzw. ungeshapten Strömen. Die Granularität ist im gemessenen Bandbreitenbereich sehr fein.
Die MDRR-Tests führten zunächst zu einem ähnlichen Ergebnis wie auch bei den früheren Tests auf der 4 fach OC3 Engine 3 LC, d.h. zugesicherte Bandbreite für Klassen, die nicht ausgenutzt wird, kann zwar von anderen Klassen mitbenutzt werden, jedoch geschieht diese Aufteilung nicht gemäß der konfigurierten bandwidth-Werte. Ein Lösungsvorschlag von Cisco – die Verwendung von remaining-bandwidth für die class-default Klasse – führte zu keiner Verbesserung. Ciscos Alternativvorschlag – für die class-default Klasse ebenso einen bandwidth-Wert zu konfigurieren – brachte schließlich ein befriedigendes Ergebnis.
WRED an sich funktioniert. Wie erwartet werden gemäß der konfigurierten WRED-Policy die Klassen mit den niedrigeren WRED-Parametern zuerst verworfen. Jedoch stellte sich heraus,

dass die maximale Gesamtkapazität der LC bei aktiviertem WRED abnimmt, je höher ein einzelner Port überlastet ist.
Auf dem Agilent Routertester lief eine frühe Beta Software. Dadurch war es nicht möglich, ausführliche Ipv6 Multicast-Tests durchzuführen. Prinzipiell ist IPv6 Multicast mit dem GE-Tetra Karten möglich. Allerdings sollte vor dem Betreiben der G-WiN-Router im Dual-Stack Betrieb unbedingt weitere Tests durchgeführt werden. Im Moment ist es nicht sicher, ob der IPv6-Traffic nicht den IPv4 beeinflussen würde. Hierzu müssten aber die im G-WiN eingesetzten Kartentypen auch im Labor verfügbar sein.

# 4 Anhang

Konfiguration der Router für die Ipv6 Multicast-Tests:

**Config 12416 - c4161**

```
Current configuration : 3197 bytes
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname *****
!
boot-start-marker
boot system tftp gsr-p-mz.120-26.S.bin 131.188.81.54
boot system tftp gsr-p-mz.120-24.S.bin 131.188.81.54
boot-end-marker
!
redundancy
 mode rpr
enable password *******
!
username ****** privilege 15 password *******
!
ip subnet-zero
ip rcmd rsh-enable
no ip domain-lookup
no mpls traffic-eng auto-bw timers frequency 0
ipv6 unicast-routing
ipv6 multicast-routing
!
interface Loopback0
 ip address 192.188.81.69 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface GigabitEthernet2/0
 description to the other router
 mtu 4470
 ip address 192.129.10.1 255.255.255.0
```

```
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 ipv6 address 416:41::1/64
 ipv6 enable
 ipv6 rip toto enable
 no cdp enable
!
interface GigabitEthernet2/1
 description to the source
 mtu 4470
 ip address 192.129.11.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 ipv6 address 416:42::1/64
 ipv6 enable
 ipv6 rip toto enable
 no cdp enable
!
interface GigabitEthernet2/2
 description to the other Cisco router
 mtu 4470
 ip address 192.129.12.2 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 ipv6 address 3FFE:FFFF:10::1/64
 ipv6 enable
 ipv6 rip toto enable
 no cdp enable
!
interface GigabitEthernet2/3
 mtu 4470
 ip address 192.129.13.2 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
```

```
 ipv6 address 410:416::2/64
 ipv6 enable
 no cdp enable
 shutdown!
interface POS4/0
 description Agilent Port 101/1
 ip address 192.101.1.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
 encapsulation ppp
 no ip mroute-cache
 ipv6 address 416:11::1/64
 ipv6 enable
 ipv6 rip toto enable
 ipv6 pim dr-priority 4294967295
 crc 32
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
!
interface POS5/0
 description Agilent Port 101/1
 ip address 192.101.2.1 255.255.255.0
 no ip directed-broadcast
 ip router isis
 encapsulation ppp
 no ip mroute-cache
 ipv6 address 416:12::1/64
 ipv6 enable
 ipv6 rip toto enable
 ipv6 pim dr-priority 4294967295
 crc 32
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
!
interface Ethernet0
 mac-address 0000.0011.2d00
 ip address 131.188.81.69 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
!
router isis
!
router rip
 version 2
 redistribute connected
 network 172.18.0.0
 no auto-summary
!
no ip classless
!
snmp-server community public RO
```

```
!
ipv6 router rip toto
 redistribute connected
!
ipv6 pim rp-address 416:42::1
!
line con 0
line aux 0
line vty 0 4
 exec-timeout 35791 0
 password 7 070C
 login local
 transport input telnet
 transport output telnet
!
end
```

**Config 12410 - c4101**

```
Current configuration : 3121 bytes
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ****
!
boot-start-marker
boot system tftp c12kprp-p-mz.120-26.S.bin 131.188.81.54
boot system tftp c12kprp-p-mz.120-25.S2.bin 131.188.81.54
boot system tftp c12kprp-p-mz.120-24.S.bin 131.188.81.54
boot-end-marker
!
redundancy
 mode rpr
enable password *****
!
username ****** password 0 *******
!
!no ip subnet-zero
no ip domain-lookup
ipv6 unicast-routing
ipv6 multicast-routing
!
!interface Loopback0
 ip address 192.188.81.71 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
!

interface GigabitEthernet1/0
 mtu 4470
 ip address 192.129.10.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
```

```
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 ipv6 enable
 no cdp enable
!
interface GigabitEthernet1/1
 mtu 4470
 ip address 192.129.11.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 ipv6 enable
 no cdp enable
!
interface GigabitEthernet1/2
 description to the other Cisco router
 mtu 4470
 ip address 192.129.12.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 ipv6 address 416:410::1/64
 ipv6 enable
 ipv6 rip toto enable
 no cdp enable
!

interface GigabitEthernet1/3
 mtu 4470
 ip address 192.129.13.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip pim sparse-dense-mode
 ip route-cache flow sampled input
 ip sdr listen
 load-interval 30
 negotiation auto
 ipv6 address 416:410::1/64
 ipv6 enable
 no cdp enable

 shutdown!
interface POS4/0
 description Agilent Port 102/1
```

```
 ip address 192.102.1.1 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 ipv6 address 410:21::1/64
 ipv6 enable
 ipv6 rip toto enable
 crc 32
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
!
interface POS5/0
 description Agilent Port 102/1
 ip address 192.102.2.1 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 ipv6 address 410:22::1/64
 ipv6 enable
 ipv6 rip toto enable
 crc 32
 clock source internal
 pos ais-shut
 pos framing sdh
 pos scramble-atm
 pos flag s1s0 2
!
interface Ethernet0
 mac-address 0000.0010.7500
 ip address 131.188.81.71 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
!
interface Ethernet1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
router rip
 version 2
 redistribute connected route-map distri
 network 172.18.0.0
 no auto-summary
!
no ip classless
!!
ipv6 router rip toto
 redistribute connected
!
ipv6 pim rp-address 416:42::1
!!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 exec-timeout 10000 0
```

```
 password c
 login local
 transport input telnet
 transport output telnet
line vty 5 99
 exec-timeout 10000 0
 password c
 login local
 transport input telnet
 transport output telnet
!
no exception warmstart
end
```