FACHHOCHSCHULE FURTWANGEN
HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT

FACHBEREICH INFORMATIONSSYSTEME
STUDIENGANG COMPUTER NETWORKING

PROJEKTSTUDIUM 2/3 · SS 2002

ADRIAN WOIZIK · WOIZIK@FOO.FH-FURTWANGEN.DE
MARCUS BENEDIX · BENEDIX@FOO.FH-FURTWANGEN.DE

IMPLEMENTIERUNG VON QUALITY OF SERVICE (QOS) IM CAMPUS-NETZ DER FACHHOCHSCHULE FURTWANGEN

Abstract

Die vorliegende Arbeit ist die schriftliche Dokumentation des im Sommersemester 2002 durchgeführten Semesterprojekts im Studiengangs „Computer Networking“ an der Fachhochschule Furtwangen.

Beschrieben werden die ermittelten Strategien und Mechanismen für die Implementierung von Quality of Service im Campus der FH Furtwangen auf Cisco Catalyst 6506.

Dieses Dokument wurde unter Verwendung von LaTeX und BiBTeX erstellt.

Furtwangen, im Juli 2002

Marcus Benedix
Adrian Woizik

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1
Einleitung

Im Studiengang „Computer Networking“ der Fachhochschule Furtwangen ist sowohl im 5. als auch im 7. Semester ein „Projektstudium“ vorgesehen. Innerhalb dieses Projektstudiums soll im Laufe des Semesters ein grösseres Projekt in Teamarbeit geplant, durchgeführt und dokumentiert werden.

Bei der Durchführung des Projektes spielen neben dem rein technischen Aspekt auch Themen wie Teamarbeit und Projektmanagement eine Rolle.

1.1 Projektziel

Ziel dieses Projektes ist die Evaluierung des Einsatzes von Quality of Service im Campus-Netz der Fachhochschule Furtwangen und ihrer Aussenstelle in Villingen-Schwenningen auf der Catalyst-Architektur von Cisco. Sollte sich bei dieser Evaluierung herausstellen, dass eine erfolgreiche Implementierung möglich ist, so soll eine beispielhafte Konfiguration erstellt werden.

Für die Durchführung des Projektes ist die Einarbeitung in die Cisco-Betriebssysteme „IOS“ und „CatOS“ und in die IT-Infrastruktur der FH-Furtwangen notwendig, die Planung und Durchführung geeigneter Messmethoden, die Auswahl von für den Anwendungszweck optimaler QoS-Strategien und eventuell letzten Endes die Planung und Umsetzung auf dem Produktivsystem.

Da sämtliche praktischen Arbeiten am Produktivsystem erfolgen, ist eine besondere Vorsicht notwendig, um den regulären Betrieb des Netzes nicht zu stören.

Mit dieser Zielsetzung soll den Nutzern des Netzwerkes der Fachhochschule Furtwangen eine bessere Qualität hinsichtlich Datendurchsatz, Verzögerungen und Performance durch gezielte Priorisierungen und Bandbreitenbeschränkungen einzelner Dienste geboten werden.

Darüberhinaus soll mit den Ergebnisse des Projektes ein tieferes Verständnis des Einsatzes von verschiedenen QoS-Techniken in einem produktiven Netzwerk mittlerer Grösse erarbeitet werden.

1.2 Ausgangssituation

Das Rechenzentrum der Fachhochschule Furtwangen verfügt zu Projektbeginn über WAN-Anbindungen zum Internet über das BelWue-Netz mit Bandbreiten von 2·2 MBit/s in Furtwangen und 2 MBit/s in Villingen-Schwenningen. Diese Anbindungen werden von den einzelnen Fachbereichen der FH-Furtwangen und ihrer Zweigstelle in Villingen-Schwennigen, den Studentenwohnheimen in Furtwangen, den umliegenden Hochschulen (BA Villingen-Schwenningen, Musikhochschule Trossingen, Polizeihochschule Schwenningen) sowie einigen Schulen im Schwarzwald-Baar-Kreis genutzt (siehe Abbildung 1.1).

Insgesamt werden über diese Strecken allein durch die FH-Furtwangen ca. 2000 Hosts geroutet. Aufgrund dieser hohen Anzahl verschiedener Systeme und Nutzer gegenüber den relativ geringen Bandbreiten der Aussenanbindungen kommt es regelmässig zu Engpässen. Der Einsatz von QoS zur Beeinflussung der Verkehrscharakteristika ist daher wünschenswert, um die vorhandenen Ressourcen effektiver nutzen zu können.

Ausserdem wird durch die Nutzer des Campusnetzes ein nicht unerheblicher Anteil des von der FH verursachten Gesamttraffics durch Anwendungen erzeugt, deren Nutzen nicht unbedingt im Sinne wissenschaftlicher Arbeit liegt (z.B. Peer2Peer-Netze: Gnutella, eDonkey, Kaza). QoS soll auch eingesetzt werden, um legitimeren Traffic gegenüber diesen Anwendungen eine höhere Priorität zuzuordnen. Dieses Problem dürfte sich mit dem geplanten Ausbau der WAN-Anbindung auf 622 MBit/s noch weiter verschärfen.

Wie in Abbildung 1.2 ersichtlich ist am Standort Furtwangen der Layer3-Switch vom Typ Cisco Catalyst 6506 das zentrale Element der LAN-Topologie. Alle auf dem Campus verteilten Layer2-Switches sind über Glasfaser-Leitungen mit dem Core-Switch verbunden. Der Switch selbst ist darüberhinaus noch mit dem WAN-Router zum Belwue-Netz verbunden.

Die zusätzliche MSFC-Card des Switches verleiht dem Gerät neben Routing-Funktionalität[1], laut den Spezifikationen des Herstellers, auch QoS-Mechanismen zur Steuerung von Traffic-Flows. Aus diesem Grund bietet sich dieser Core-Router für die Implementierung von Campus-weitem QoS an.

[1] Aufgrund dieser Routing-Funktionalität kann der Switch auch als Core-Router bezeichnet werden. Im folgenden wird daher nur dieser Begriff verwendet.
Die zugrunde liegenden Methoden werden als „Multi Layer Protocol Switching“ bezeichnet und sind in [Web00] beschrieben.

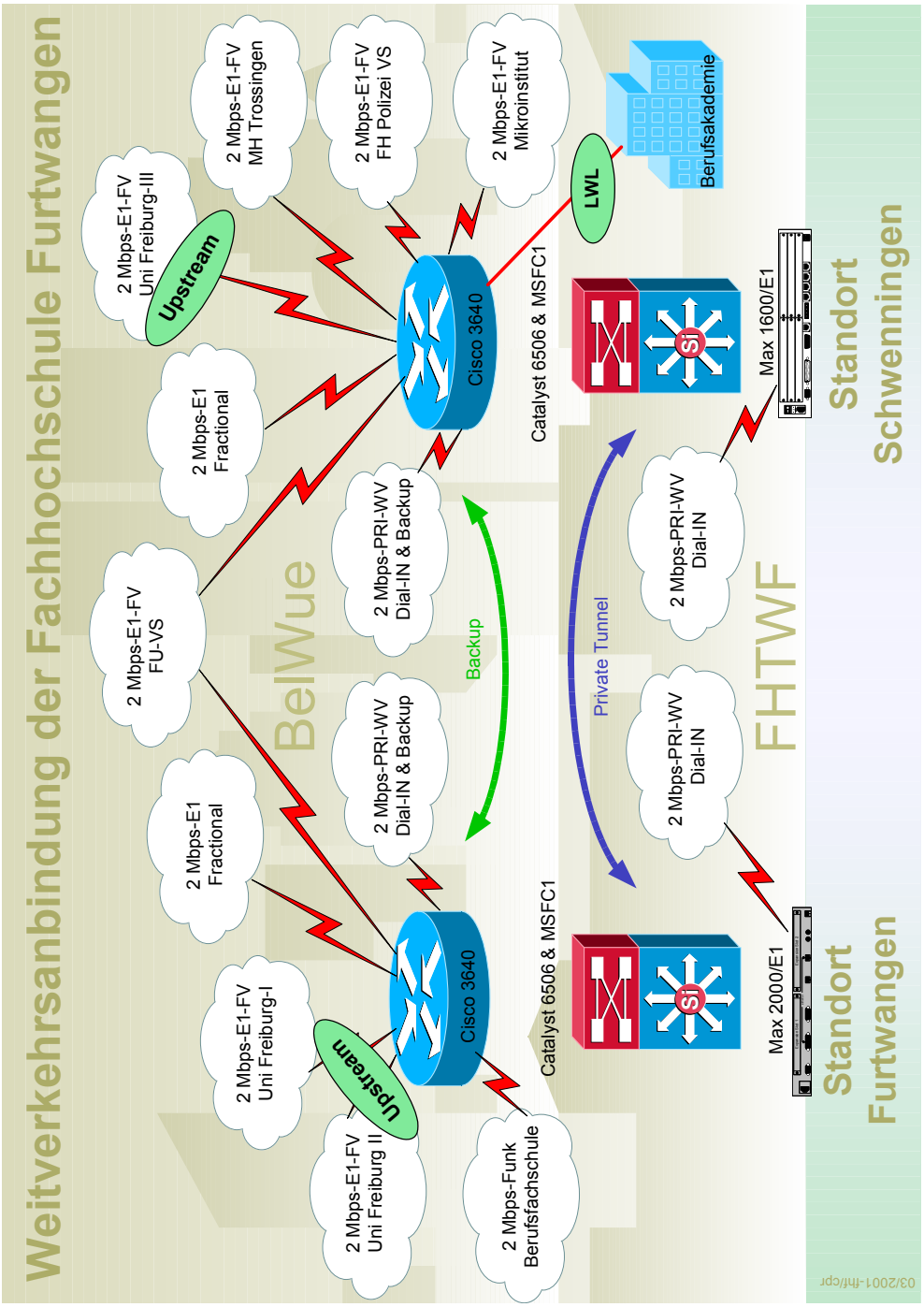

Abbildung 1.1: WAN-Anbindung der FH Furtwangen an das Belwue-Netz und andere Beteiligte, Stand: Juni 2002

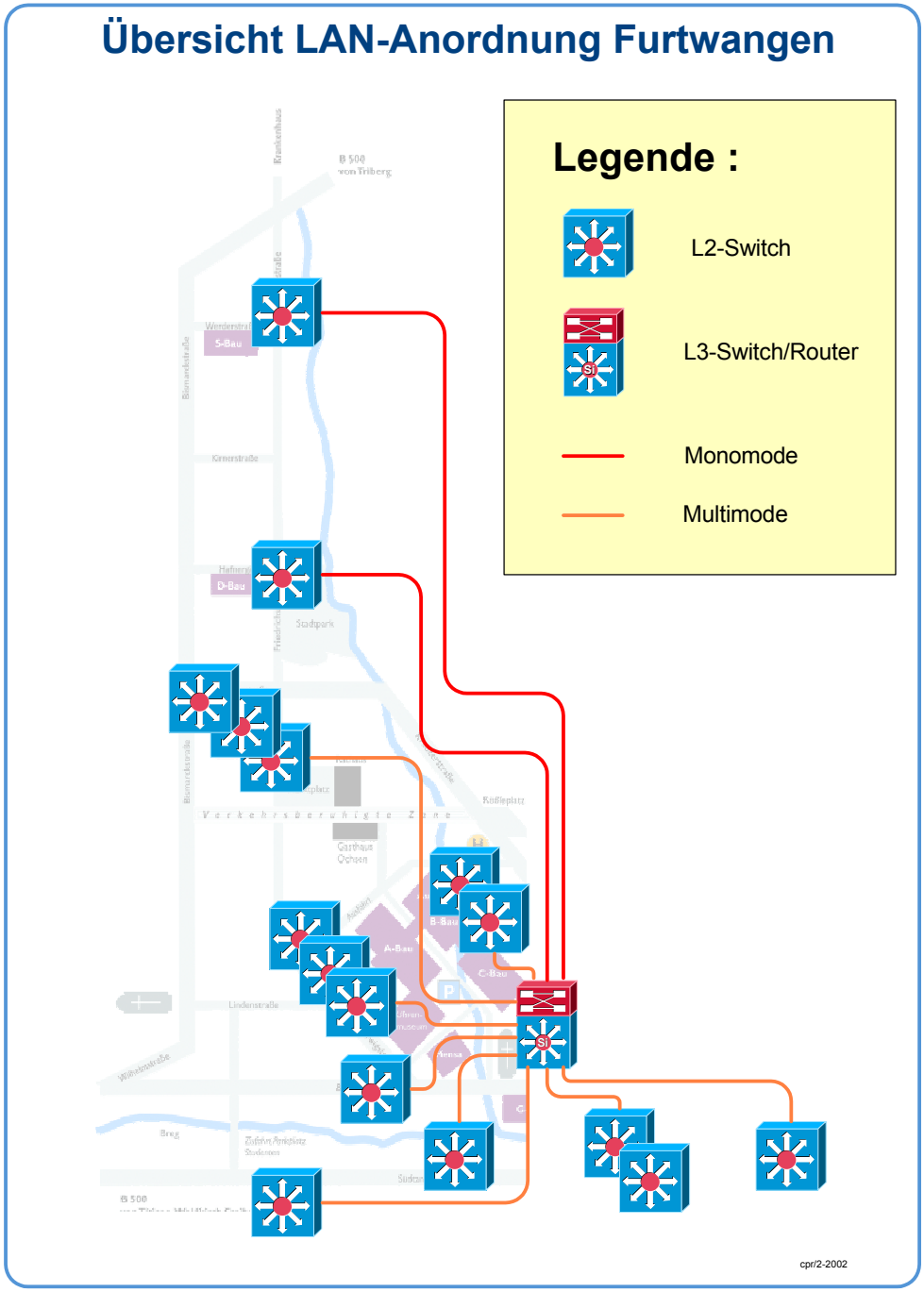

Abbildung 1.2: LAN-Infrastruktur am Standort Furtwangen, Stand: Juni 2002

1.3 Dokumentstruktur

Dieses Dokument ist in sechs Kapitel gegliedert:

Kapitel 1 dient der Einleitung zu dem Projekt an sich, der ursprünglichen Zielsetzung und der Beschreibung der dem Projekt zugrundeliegenden Problematik.

Kapitel 2 beschreibt die praktische Durchführung des Projektes inklusive Vorüberlegungen wie Projektplan und Notfallplan. Ausserdem wird die Begründung für Abweichungen vom Projektplan in der tatsächlichen Durchführung angegeben.

Kapitel 3 gibt einen theoretischen Überlick über QoS in IP-Netzen im Allgemeinen.
Erwähnung finden die Charakteristik von Best-effort-Netzen wie dem Campus-Netz der Fachhochschule Furtwangen, das Verhalten von Transportprotokollen in Überlastsituationen, die sich bietenden Möglichkeit durch den Einsatz von „Differentiated Services“ als QoS-Modell und benennt letztendlich die Kriterien, mit denen sich die „Quality of Service“ von Netzen definieren lassen.
Die erwähnten Punkte ergeben sich aus der Problemanalyse innerhalb des Campus-Netzes der FH-Furtwangen.

Kapitel 4 geht detailiert auf die QoS-Implementierung auf Ciscos Catalyst-Architektur ein und beschreibt die Konfiguration für die technische Realisierung innerhalb der Hardware.

Zur Implementierung von QoS auf der Catalyst-Architektur existieren, abgesehen von der technischen Dokumentation des Herstellers ([Cis00a]), derzeit nur wenige Quellen. Dieses Kapitel soll eine weitere Hilfestellung für eigene Konfigurationen geben.

Es schliesst mit Kapitel 5 ein Designvorschlag auf Basis unserer Erfahrungen für eine mögliche Implementierung von QoS auf dem Core-Router der FH-Furtwangen an.

Das letzte Kapitel 6 beendet die Dokumentation und zieht ein Fazit über den Projektverlauf und die geleistete Arbeit.

Kapitel 2
Projektmanagement

2.1 Projektplan

Für die Umsetzung der Zielsetzung steht der Zeitraum zwischen 04.04.2002 und 14.06.2002 zur Verfügung. Praktische Arbeiten am Core-Router kann bis zum 27.05.2002 durchgeführt werden. Auf Basis dieser Vorgaben lässt sich die Projektarbeit mit folgendem Zeitplan strukturieren:

2.1.1 Phase 1 (04.04. - 15.04.): Einarbeitung

Die Einarbeitungsphase dient zuerst zum Erwerb spezieller theoretischer Kenntnisse von Ciscos IOS im Bereich Quality of Service und der verfügbaren Hardware Cisco Catalyst 6506 mit ihren spezifischen Eigenheiten. Darüberhinaus wird ein detailierterer Überblick über die Infrastruktur des Campus-Netzes der Fachhochschule Furtwangen erarbeitet.

Hauptsächlich wird in dieser Phase mit schriftlicher Dokumentation von Cisco und der des Rechenzentrums der FH-Furtwangen, sowie mit Fachliteratur wie [Bon02] gearbeitet.

2.1.2 Phase 2 (15.04. - 29.04.): Bestandsaufnahme, Ausarbeitung Notfallplan, Aufbau Testumgebung, Messungen

Nach Abschluss der Einarbeitungsphase kann mit einer Bestandsaufnahme der aktiven Konfiguration des Core-Routers der FH Furtwangen begonnen werden. Mit der Bestandsaufnahme müssen weitere Besonderheiten der spezifischen Konfiguration, wie beispielsweise die VLAN-Konfiguration, geklärt werden.

Mit der Konfiguration werden Superuser-Rechte auf den Betriebssystemen des Gerätes übernommen. Die besondere Verantwortung, die mit diesen

Rechten übernommen wird, macht die Erstellung des in Abschnitt 2.2 erwähnten Notfallplanes erforderlich, um die Betriebssicherheit des Netzwerkes im Falle von konfigurationsbedingten Ausfällen sicherstellen zu können. Dieser Notfallplan wird aufgrund von den in Abschnitt 6 erwähnten Vorfällen später noch verschärft.

2.1.3 Phase 3 (29.04. - 06.05.): Möglichkeitsanalyse

Mit der Möglichkeitsanalyse sollen auf Basis der bis dahin erarbeiteten Kenntnisse, die Möglichkeiten für Einsatz von QoS auf der benannten Architektur ermittelt werden. Die in dieser Phase ausgearbeiteten Erkenntnisse sollen die genaue Planung und Anpassung an die Bedürfnisse der FH einer QoS-Policy in der nächsten Phase ermöglichen.

2.1.4 Phase 4 (06.05. - 20.05.): Policydefinition, Umsetzung, Messung

In der 4. Phase ist die Definition einer beispielhaften QoS-Policy in Zusammenarbeit mit dem Rechenzentrum der FH-Furtwangen geplant, sowie deren Umsetzung auf dem Produktivsystem und anschliessende Messungen der Auswirkungen. Während des Zeitraumes sollen mehrere Feinabstimmungen die definierte Policy an den vorliegenden Anwendungsfall anpassen. Die Wirksamkeit der Umsetzung solle mit Messungen bewiesen werden.

2.1.5 Phase 5 (20.05. - 27.05.): Auswertung, Rolloutplanung, Policy-Anpassung

Die 5. Phase sieht die Auswertung der bis zu diesem Zeitpunkt vorliegenden Ergebnisse mit dem Rechenzentrum und weitere Anpassungen vor. Des weiteren ist die Planung für den Rollout der Konfiguration auf dem System in Villingen-Schwenningen angedacht.

2.1.6 Phase 6 (27.05.): Rollout

Die vorletzte Phase sieht den Rollout der Beispielkonfiguration an beiden Standorten sowie die Übergabe und Einweisung der beteiligten Mitarbeiter des Rechenzentrums vor.

2.1.7 Phase 7 (27.05. - 14.06): Dokumentation, letzte Anpassungen

Die 7. und letzte Phase wird für die Dokumentation des Projektes und eventuell letzte Anpassungen eingeplant. Ausserdem beinhaltet diese Phase eine Zeitreserve von etwa 2 Wochen für möglicherweise in früheren Phasen aufgetretenen Zeitverzug.

2.2 Notfallplan

Aufgrund der hohen Kosten der eingesetzten Hardware lässt sich für dieses Projekt die Infrastruktur nicht im Versuchsaufbau simulieren. Dies hat zur Folge, dass sämtliche Konfigurationsänderungen, Tests und Messungen am Produktivsystem erfolgen müssen. Da das Campus-Netz rund um die Uhr von einer erheblichen Anzahl von Anwendern benutzt wird, ergeben sich auch keine unkritischen Zeitabschnitte, in denen zumindest ein kurzfristiger Ausfall des Core-Routers tragbar wäre. Ein solcher Zustand muss daher nach Möglichkeit vermieden werden. Sollten sich trotzdem Einschränkungen ergeben, so müssen diese so schnell wie möglich behoben werden. Ein Totalausfall, z.B. zu Nachtzeiten, und dessen Behebung durch physikalischen Zugang zum Gerät selbst nach mehreren Stunden darf unter keinen Umständen eintreten.

Um diesen Fall auszuschliessen, wird ein „Notfallplan“ ausgearbeitet. Dieser Plan soll die Betriebssicherheit des Campus-Netzes garantieren, indem auch im „worst-case“ die Benutzbarkeit des Netzes innerhalb kürzester Zeit wiederhergestellt und damit die Ausfallzeiten auf ein Minimum beschränkt werden können.

Im Detail sieht der Notfallplan für den Zeitraum von Evaluierung und Test eine Wiederherstellung der ursprünglichen Konfiguration nach Abschluss jedes Tests vor. Damit sollen Probleme durch Konfigurationen vermieden werden, die erst zu einem späteren Zeitpunkt zum Vorschein treten.

Für den Fall, dass während der Tests irrtümliche Konfigurationen vorgenommen werden, und eine Fehlerbehebung über Remote-Access auf Netzwerkebene nicht mehr möglich ist, besteht die Möglichkeit, einen Hard-Reset des Core-Routers vorzunehmen. Dafür steht eine Steckdosenleiste zur Verfügung, die via Einwahl über eine dedizierte ISDN-Leitung das Ein- und Ausschalten der Stromversorgung des Gerätes und damit ein Restart mit der ursprünglichen Konfiguration ermöglicht.

Ein Zugang zum Rechenzentrum der Fachhochschule und damit zu dem Gerät selbst besteht in der Regel nicht.

Kapitel 3
Quality of Service im Allgemeinen

3.1 Notwendigkeit

3.1.1 Charakteristik von Netzen auf Best-effort-Basis

Die heute in modernen Netzwerken eingesetzten Protokolle wurden in den 60er und 70er Jahren mit dem Ziel entwickelt, auch unter widrigen Umständen, wie schlechter Leitungsqualität und unzuverlässigen Verbindungen, ein gewisses Maß an Zuverlässigkeit bei der Datenübertragung zu gewährleisten. Das Ergebnis dieser Entwicklungen ist u.a. das IP-Protokoll mit „best-effort“-Verhalten.

Best-effort bedeutet, dass sich jedes Segment der beteiligten Netzwerke bemüht, die Weiterleitung der Datenpaketen optimal zu gestalteten. Dies ist aber wirklich nicht mehr als nur ein Bemühen, eine Garantie für die zeitnahe Zustellung der Daten (Delay) und die Varianz dieser Zeiten (Jitter) wird dabei genausowenig abgegeben wie eine Aussage über die Effektivität der Datenübertragung (Datendurchsatz) oder der Anteil an Paketverlusten (paketloss).

Die Qualität des Dienstes „Nachrichtenübertragung“ (Quality of Service) ist damit in IP-Netzen nicht definiert und schwankt für einzelne Anwendungen in Abhängigkeit der Gesamtbelastung der beteiligten Netzwerke.

In Umgebungen wie LANs, in denen die Infrastruktur des Netzwerkes gegenüber den Anforderungen der Anwendungen ausreichend ist, also genügend Bandbreite und Übertragungskapazität für alle Anwender und Anwendungen zur Verfügung steht, reicht best-effort für zufriedenstellendes Arbeiten aus. Ein nachteiliges Verhalten wird erst bemerkbar, wenn an einer oder mehreren Stellen des Übertragungsweges Überlastungen auftreten und sogenannte „Bottlenecks“ entstehen.

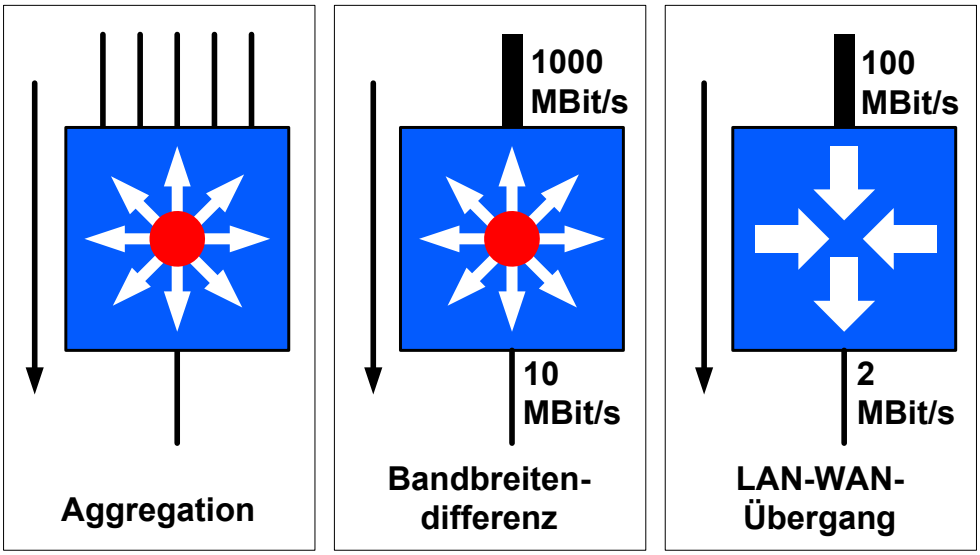

Abbildung 3.1: Ursachen von Paketstauungen (Bottlenecks) (Darstellung nach [Cis00b])

Diese Überlastungen treten aus den in Abbildung 3.1 skizzierten Gründen in den Ausgangs-Puffern der Interfaces von Switches und Routern auf:

Aggregation
: Mehrere Eingangsverbindungen werden mittels einen Switches auf einen Ausgangsverbindungen aggregiert. Übersteigt die Bandbreite des über die einzelnen Links eingehenden Traffics die Kapazität des Ausgangslinks, so kommt es zwangsläufig zu einem Überlauf des Ausgangspuffers (z.B. Abbildung 3.2).

Bandbreitendifferenzen
: Das vorher skizzierte Szenario tritt auch auf, wenn die Kapazität eines einzelnen Links die des Ausgangslinks übersteigt.

Übergang vom LAN ins WAN
: Der häufigste Fall tritt beim Übergang eines schnellen LANs in ein relativ langsameres WAN auf. Auch hier kommt es zum Überlauf des Ausgangspuffers des WAN-Routers.

Füllt sich der Ausgangspuffer eines Links, so macht sich dies durch Verzögerungen der Paketlaufzeit bemerkbar und schlägt sich in erhöhten Round-Trip-Times nieder. Füllt sich der Puffer weiter und kommt es wie in Abbildung 3.2 skizziert zu einem Überlauf des Output-Puffers; müssen Pakete verworfen werden. Dieser Fall macht sich durch Paketloss bemerkbar.

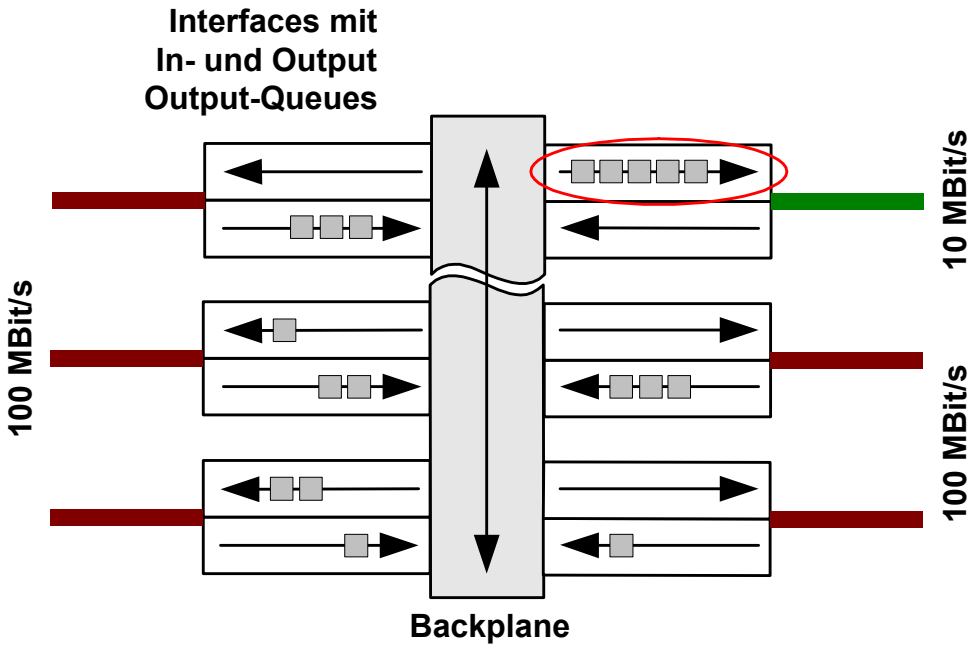

Abbildung 3.2: interne Struktur eines Switches, Überlauf eines Output-Puffers

Die Selektion der zu verwerfenden Pakete ist rein zufällig. Damit soll eine statistische Gleichverteilung der Paketverluste aller weitergeleiteten Pakete unabhängig von Protokoll oder Quelle- und Zieladresse erreicht werden. Der entstehende Effekt wird als „Best-effort“-Verhalten bezeichnet.

Die zufällige Selektion der zu verwerfenden Pakete bei überlaufenden Queues ist nicht immer erwünscht, da die Entscheidung nicht auf Basis der Bedürfnisse der beteiligten Anwendungen, sondern nur zufällig bestimmt wird. Diese unkontrollierten Congestions können zu nachteiligen Verhalten für die Anwendungen der Trafficflows führen.
Ziel, im Sinne von QoS, sind kontrollierbare Congestions, bei denen die zwangsläufig auftretenden Verluste bei konkurrierenden Flows durch vorher definierte Policies bestimmbar sind.

3.1.2 Charakteristika von Protokollen in Überlastsituationen

Transmission Control Protocol (TCP)

Das TCP-Protokoll wird von höheren Schichten verwendet, wenn es auf die gesicherte Zustellung von Paketen unbedingt ankommt (Übertragung von Dateien). TCP steuert die Übertragungsraten mittels zweier Algorithmen: Slow Start und Congestion Avoidance [Ste94], [Tan98].

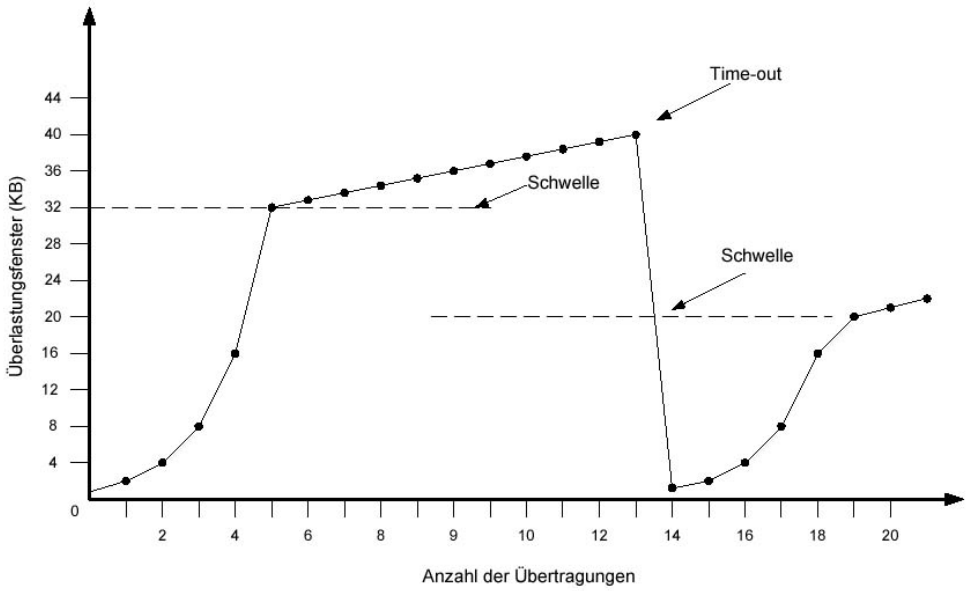

Abbildung 3.3: Überlastalgorithmen von TCP (Darstellung aus [Tan98])

Beide Algorithmen regeln dabei die Übertragungsrate der Verbindung durch allmähliche Vergrösserung des Congestion Window beim (Slow Start) und dessen Reduzierung bei Paketverlusten, die infolge von zu hohen Übertragungsraten und daraus folgenden Überlastungen des Übertragungsweges entstehen (Abbildung 3.3).

Damit regelt die Flusskontrolle des TCP-Protokolls die Übertragungsrate und passt sie vorhandenen Ressourcen dynamisch an. TCP versucht dabei generell, die maximal mögliche Bandbreite zu beanspruchen. Paketverluste sind somit sogar erwünscht, stellen sie doch die einzige Möglichkeit zur Signalisierung von Überlastsituationen dar.

In der Praxis zeigt sich allerdings, dass sich traffic-intensive Anwendungen, wie beispielsweise FTP, hinsichtlich der Beanspruchung der gemeinsamen Ressource, bei Konkurrenz mit anderen Anwendungen, sehr dominant verhalten können. Im Vergleich mit einer eher sparsamen Anwendung wie ssh, beansprucht FTP dabei einen Grossteil der zur Verfügung stehenden Bandbreite. Die stark interaktive ssh-Anwendung wird aufgrund der auftretenden Paketverluste, der notwendigen Neuübertragung und den daraus resultierenden hohen Verzögerungszeiten unbenutzbar.

Vermutlich liegt die Ursache darin, dass sich ein TCP-Flow mit hohem Paketdurchsatz bei Paketverlusten schneller erholt und die ursprüngliche Übertragungsrate eher wieder erreicht als ein konkurrierender Flow mit geringerem Durchsatz. Der zweite Flow ist von Paketverlusten durch Engpässe im Verhältnis zum Gesamtdurchsatz stärker betroffen.

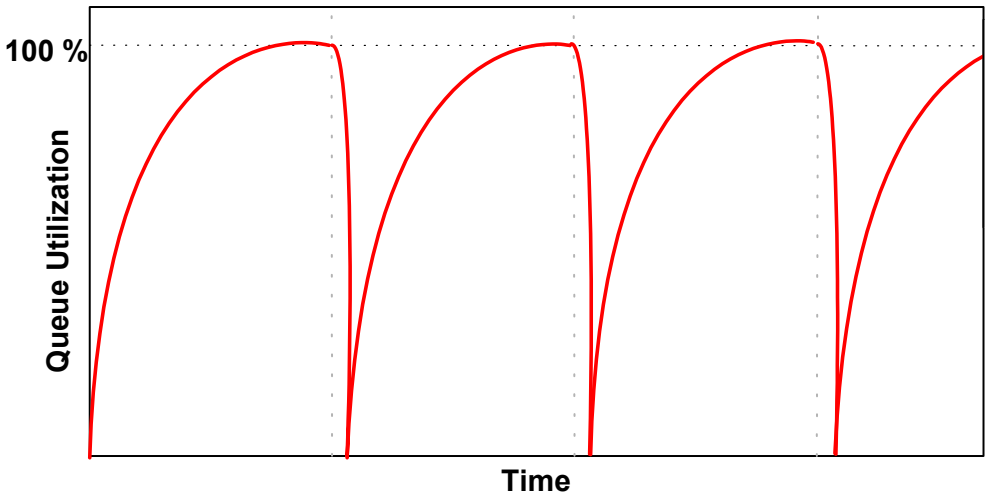

Abbildung 3.4: Queue-Auslastung beim Global-Synchronisation-Effekt

Die Eigenschaften des TCP-Protokolls können ausserdem unter bestimmten Bedingungen zu einem Effekt führen, der als „Global Synchronisation“-Effekt bekannt ist. Treten durch gesättigte Queues innerhalb eines Routers Paketverluste auf, so betrifft dies meist mehrere Flows. Deren Sender erkennen diese Verluste richtig als Überlastung und reduzieren ihre Übertragungsraten. Die Queue leert sich dadurch wieder. Slow Start sorgt anschliessend wieder für einen Anstieg der Übertragungsraten - aber nahezu zeitgleich bei allen Sendern. Damit tritt eine Synchronisation ein. Die Queue wird daraufhin wieder gesättigt und es kommt wenig später zu neuen Verlusten. Der „Global Synchronisation“-Effekt führt damit wie in Abbildung 3.4 skizziert zu Schwingungsverhalten im Verkehrsfluss und im Durchschnitt zu einer nicht optimalen Ausnutzung des Links (z.B. effektive Übertragungsraten von 3 MBit/s bei einer Kapazität von 4 MBit/s).

QoS kann mittels speziellen Algorithmen die drohende Sättigung der Queues frühzeitig erkennen und diesem Effekt durch gezielt dosierte Paketverluste entgegen wirken.

User Datagram Protocol (UDP)

UDP ist ein verbindungsloses Protokoll. Eine Flussteuerung und Sicherung der Datenübertragung ist nicht implementiert. Eine besondere Bedeutung bekommt UDP im Bereich Multicast. Hierfür eigenen sich lediglich UDP. TCP als verbindungsorientiertes Protokoll unterstützt lediglich die Kommunikation zwischen zwei Prozessen, aber nicht die bei Multicast-Anwendungen gewünschten 1:n-Beziehungen.

Die fehlende Flusskontrolle macht einen hohen Datendurchsatz und geringe Verzögerungen möglich, da auf zeitraubende Neuübertragung verlorener Pakete verzichtet wird. Aus diesem Grund kommt es bei Anwendungen zum Einsatz, bei denen Paketverlust toleriert werden kann. Die entstehenden Paketverluste werden entweder ignoriert (Audio- oder Videostreaming) oder auf Anwendungsebene erkannt und durch erneute Anforderung kompensiert (NFS).

Diese, bei nicht überlasteten Netzen erwünschte Eigenschaft, wird jedoch in Überlastungssituationen zum Nachteil. Durch die fehlende Flusskontrolle bemerkt UDP die Überlastung nicht und sendet weiter mit der ursprünglichen Datenrate. Im Falle von Neuanforderungen durch Paketverluste wird das ohnehin belastete Netz durch weitere Anforderungen noch mehr saturiert.

In der Praxis zeigt sich in Überlastsituationen die Aggressivität von UDP vor allem durch Dominanz gegenüber TCP.

3.2 Differentiated Service als QoS-Modell

Für den Einsatz von QoS haben sich zwei unterschiedliche Modelle entwickelt: Integrated und Differentiated Services.

Der Integrated-Services-Ansatz erfordert die Reservierung der notwendigen Ressourcen durch die Anwendung vor der eigentlichen Datenübertragung. Dazu bestimmt die beteiligte Anwendung die benötigten Ressourcen, z.B. eine garantierte Bandbreite von 128 kBit/s bei möglichst geringer Verzögerung, und führt dann eine exklusive Reservierung dieser Ressourcen mittels einem geeigneten Protokoll wie z.B. RSVP[1] durch. Jeder Router auf dem Pfad zwischen Sender und Empfänger muss diese Reservierung unterstützen.

Integrated Service ist damit ein anwendungsorientiertes QoS-Modell. Da sich mit diesem Modell die Probleme der FH Furtwangen nicht lösen lassen, wird es an dieser Stelle nicht weiter betrachtet und auch nur der Vollständigkeit wegen erwähnt. Nähere Informationen zu diesem Konzept finden sich in [Wan01].

Das „Differentiated Service“-Modell verfolgt einen gänzlich anderen Ansatz im Vergleich zu „Integrated Services“. Die benötigten Ressourcen werden im Vorfeld durch sogenannte „Service Level Agreements“ zwischen Netzbetreiber und -nutzer spezifiziert, garantiert und innerhalb der Netzinfrastruktur entsprechend umgesetzt. Anforderungen von Ressourcen zur Laufzeit durch Anwendungen finden nicht statt. Das DiffServ-Modell eignet sich damit besonders für die Zusicherung von Leistungsmerkmalen durch Netzbetreiber.

Abbildung 3.5 zeigt die Funktion dieses Modells:

[1] Ressource Reservation Setup Protocol

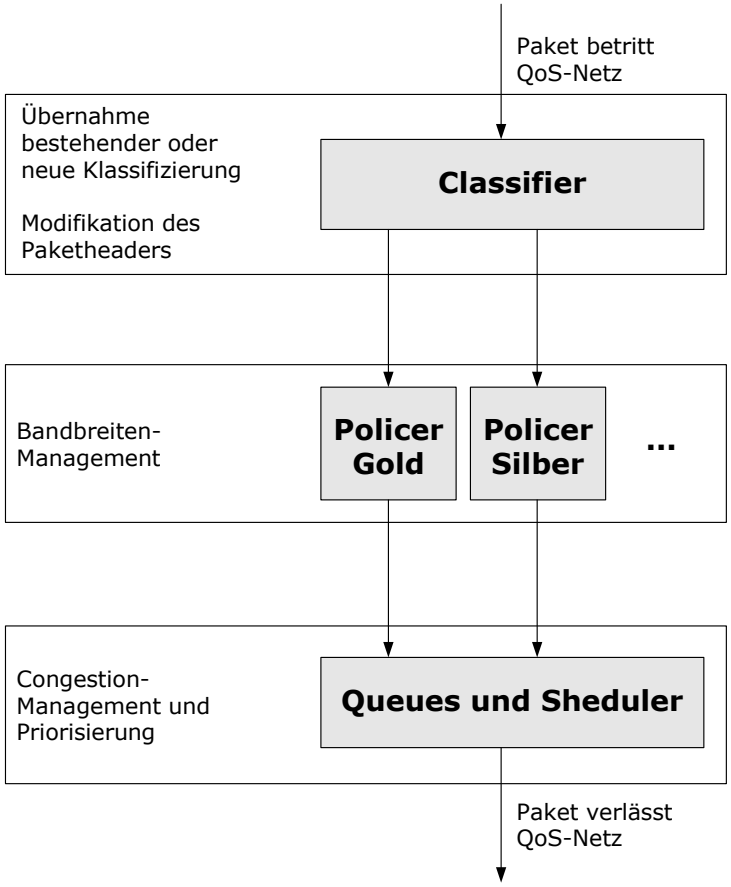

Abbildung 3.5: Ablaufschema des Differentiated-Service-Modells

3.2.1 Klassifizierung

Gelangt ein Paket in ein QoS-Netz, so wird es mindestens einmal auf Basis von Headerinformationen der Layer 2 bis 4 in unterschiedliche Klassen (Gold, Silber, Bronze, Blech) eingeteilt. Diese Merkmale könnten z.B. MAC-Adressen, IP-Adressen, Protokolle und Source- oder Destination-Ports der jeweiligen Anwendungen sein. Die Classifier speichern die Klassifizierungsinformation innerhalb der Header des jeweiligen Paketes (z.B. im ToS-Feld von IP-Paketen oder innerhalb des Headers von 802.1q-Paketen). Ob die jeweiligen Instanzen des Netzes (Router, Switches) einer bereits erfolgten Klassifikation vertrauen, mit Default-Werten überschreiben oder neu klassifizieren, ist für den jeweiligen Anwendungszweck beliebig konfigurierbar.

3.2.2 Bandbreiten-Managment (Traffic Conditioning)

Für jede Klasse werden Parameter wie Bandbreitenobergrenzen und Prioritäten definiert. Eingehender Traffic bzw. eingehende Flows werden daraufhin

von sogenannten „Traffic Conditionern“ oder „Policern“ in Abhängigkeit der zugewiesenen Klasse behandelt.

Diese Policer erzwingen die bestimmten Traffic-Obergrenzen durch gezielte Paketverluste („drop“), wenn das Trafficaufkommen dieser Klasse die vorher bestimmten Werte überschreitet, oder stufen diese sog. „out-of-profile“-Pakete in Ihrer Priorität herunter („markdown“, z.B.: Gold → Silber).

Mit diesen Mechanismen werden folgende Eigenschaften eines QoS-fähigen Netzes erzielt:

- Garantie von Mindest- und Festlegung von Maximalbandbreiten für einzelne Klassen (Mindestbandbreiten durch Aufteilung der möglichen Gesamtbandbreite auf alle Klassen; Maximalbandbreite durch Obergrenzen mit drop-Policern)
- Optimale Ressourcennutzung (Maximalbandbreite kann überstiegen werden, wenn weitere Ressourcen verfügbar sind und keine höher priorisierte Flows konkurrieren)

3.2.3 Congestion Management

Mittels Traffic Conditionern können Bandbreiten von Traffic-Klassen zwar geformt werden, aber nicht gegeneinander priorisiert werden. Eine Priorisierung, also die Entscheidung, welches Paket innerhalb einer Queue zuerst weitergeleitet werden soll und welche Pakete in Überlastsituationen zuerst verworfen werden können, kann nur innerhalb der In[2]- und Output-Queues der Interfaces erfolgen. Das bestimmende Kriterium sind dabei die Paketbewertungen der Classifier - Pakete mit niedrigster Priorität werden zuerst verworfen.

Ein Interface kann mehrere In- und Output-Queues besitzen (Abb. 4.1). Damit können einzelne Queues für bestimmte Traffic-Klassen reserviert werden und mittels Round-Robin-Verfahren abgearbeitet werden. Möglich sind auch sog. „strict-queues“, die immer erst komplett abgearbeitet werden müssen, bevor andere Queues bedient werden. Solche Queues garantieren höchste Prioritäten, können aber auch dazu führen, das kein anderer Traffic mehr weiter geroutet wird.

Dem Global-Synchronisation-Effekt können die Queues durch Algorithmen wie RED[3] entgegen wirken, indem sie bereits bei einem Füllstand von 60% unwichtige Pakete verwerfen und dadurch ein Überlauf der Queue verhindern.

[2] Congestions in Input-Queues treten in der Regel nicht auf, da die internen Busse von Routern und Switchen meist mehrfach schneller als die Interfaces selbst sind. Input- Congestions können damit nur bei Überlastungen der internen Backplanes entstehen. Das dürfte in der Regel nicht oder nur in besonderen Konfigurationen der Fall sein.

[3] Random Early Detection

3.2.4 Verteilung

Die einzelnen Teilschritte können auf mehrere Instanzen im Netz verteilt werden. So wäre es zumindest theoretisch denkbar, innerhalb der Infrastruktur der FH-Furtwangen nach Abbildung 1.2 die Klassifizierung den Layer-2-Switchen zu überlassen, das Bandbreiten-Management auf dem Core-Router vorzunehmen und das Congestion-Management am Bottleneck des WAN-Routers durchzuführen. Im Gegensatz zu einer einzigen QoS-Instanz auf z.B. dem WAN-Router kann damit die CPU-Belastung auf die einzelnen Geräte verteilt werden.

3.2.5 Bewertungskriterien und Policies

Wenn die Qualität eines Netzes definiert werden soll, so werden dafür konfigurierbare und auch messbare Parameter benötigt, die Quality of Service definiert und einzelne Klassen voneinander differenziert. Diese Parameter werden in der Regel auch in den Service Level Agreement niedergeschrieben. In der Praxis haben sich für diese Definition vier Kriterien durchgesetzt:

Durchsatz
: garantierter Datendurchsatz pro Zeiteinheit (Bits/s)

Verzögerung
: Übertragungszeit von Sender bis Empfänger (Delay)

Jitter
: Varianz der Delay-Zeiten von aufeinanderfolgenden Datenpaketen eines Flows

Paketverlust
: Anteil der Pakete eines Flows in Prozent, der während der Übertragung verloren geht (Paketloss)

Unterschiedliche Anwendungen stellen unterschiedliche Anforderungen an die Qualität eines Netzes und damit an die Ausprägung dieser Parameter.
So ist Jitter beispielsweise für reine Datentransfers via FTP eher irrelevant, da es bei diesen Anwendungen auf möglichst hohe Transferraten und geringe Paketverluste ankommt.
Interaktive Anwendungen wie ssh benötigen vergleichsweise geringe Bandbreiten, stellen aber hohe Anforderungen an die Delay-Zeiten.
Audio-Streaming und VoIP-Anwendungen sind toleranter gegenüber Paketverlusten und benötigen relativ geringe Transferraten[4], sind aber empfindlich gegenüber grösseren Jitter-Werten[5] ([Sie00] beschäftigt sich ausführlich mit QoS für Audioübertragung in IP-Netzen).

[4] 64 KBit/s reichen aus für Sprachübertragung in ISDN-Qualität, 128 kBit/s für mp3-Qualität
[5] Verzerrungen als Folge

Klasse	Priorität	Bandbreite	Services
Gold	sehr hoch	512kBit/s garantiert	Netzwerkmanagement
Silber	hoch	bis 2 MBit/s garantiert	ssh, Audiostreaming, VoIP
Bronze	normal	verfügbarer Rest	default (http, ftp, mail, dns, ...)
Blech	gering	maximal 256 kBit/s	Peer2Peer (Napster), ICMP

Tabelle 3.1: Klassendefinition und -merkmale

Aus den verschiedenen Anforderungen der Anwendungen lässt sich nach einer Analyse des Verkehrsaufkommens eine Policy erstellen, die verschiedene Anwendungen in Klassen, wie in Tabelle 3.1 dargestellt, zusammenfasst. Auf Basis dieser Policy kann dann die Konfiguration für die Classifier und Policer erstellt werden.

3.2.6 Zusammenfassung

Durch Kombination der verschiedenen Manipulationsmöglichkeiten ermöglicht QoS intelligentes Flow-Management:

- Zusicherung von verfügbaren Mindestbandbreiten für definierte Klassen
- Verbesserung der Paket-loss-Charakteristia
- Vermeidung und Management von Stauungen in den Queues von Switchen und Routern
- Traffic-Shaping
- Setzen von Prioriäten einzelner Traffic-Flows während des Transports durch das gesamte Netzwerk

Kapitel 4
QoS auf Cisco Catalyst 6506

4.1 Implementierung

Die Implementierung von Quality of Service auf der vom Rechenzentrum der FH Furtwangen eingesetzten Cisco Catalyst 6506 mit Policy-Feature-Card (PFC) realisiert die Beeinflussung der verschiedenen Traffic-Flows im wesentlichen auf drei Ebenen (Darstellung 4.1):

- Klassifizierung (Layer 2, 3 und 4 innerhalb der Interfaces und der PFC)
- Bandbreiten-Management (innerhalb der PFC)
- Congestion Management (innerhalb der Interfaces)

4.1.1 Klassifizierung

Die Klassifizierung dient der Unterscheidung und Markierungen der unterschiedlichen Datenpakete auf Basis von Merkmalen auf Layer 2, 3 oder 4, den physikalischen Ports, über die ein Paket den Switch erreicht hat oder der VLAN-Zugehörigkeit. Ebenso kann eine bereits von anderen Routern oder Switches innerhalb des Netzwerkes vorgenommenen Klassifizierung übernommen werden.

Die Eingangsports können bereits vorgenommene Klassifizierungen übernehmen und die eingehenden Pakete in unterschiedliche Input-Queues einreihen. Massgeblich sind dafür Informationen in den Headern von nach 802.1Q-, 802.1p- oder ISL-gekapselten Paketen (trust-CoS).

Ebenfalls können die CoS-Werte der erhaltenen Pakete mit Default-Werten überschrieben werden, wenn den Angaben der eingehenden Pakete nicht unbedingt vertraut werden kann (untrusted).

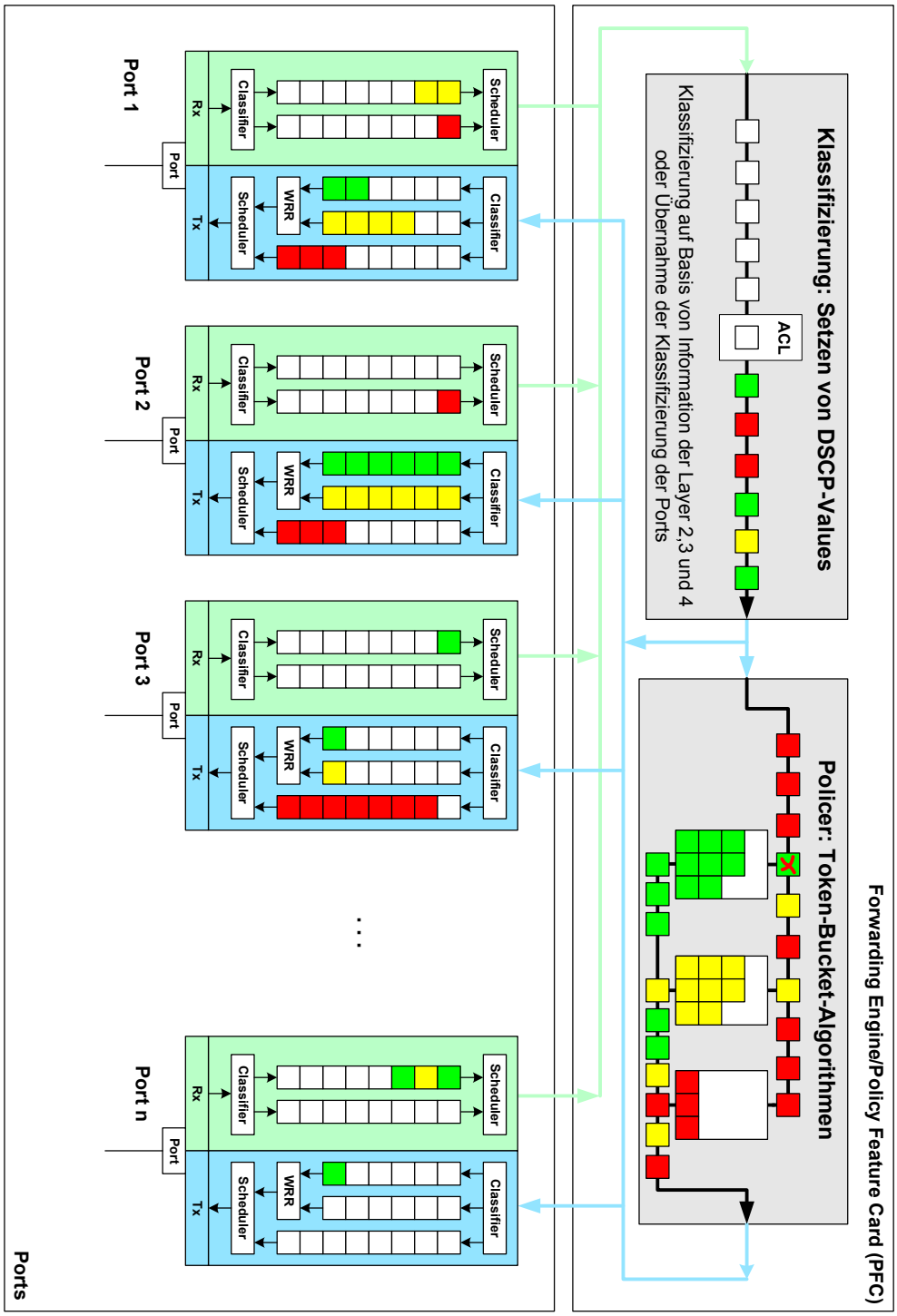

Abbildung 4.1: Schematische Darstellung der Komponenten der QoS-Implementierung auf Cisco Catalyst 6000

Eine weitere Klassifizierung nach Headerinformationen aus Layer 2, 3 und 4 kann mittels der Policy Feature Card innerhalb der Switch-Engine vorgenommen werden. Bereits erfolgte Klassifizierungen durch die Ports können übernommen werden.

Intern arbeitet die Policy-Feature-Card mit DSCP-Werten[1] als Prioritätsmerkmal. Dazu erfolgt ein CoS-DSCP-Mapping, wenn ein Paket die Switch-Engine erreicht. Beim Verlassen, also bei Übergabe eines Paketes an einen Output-Port, wird wieder ein Mapping von DSCP-CoS durchgeführt.

4.1.2 Bandbreiten-Management

Die Bandbreitensteuerung von verschiedenen Traffic-Flows wird innerhalb der PFC über sogenannte Policer durchgeführt. Massgeblich für das Policing eines Paketes (oder Traffic-Flows mit gleichen Merkmalen) sind über Classifier gesetzte ACLs.

Die Policer arbeiten nach dem Token-Bucket-Algorithmus. Bildlich kann dieser Algorithmus als ein undichter Eimer (Bucket) verstanden werden, in dem eingehenden Pakete (Token) gesammelt werden. Über das Leck des Eimers kann innerhalb eines Zeitintervalles eine bestimmte Anzahl (rate) von Paketen den Eimer verlassen (Abbildung 4.2).

Kommt es kurzzeitig zu einem höheren Zulauf an Paketen als Abfluss, so füllt sich der Eimer. Solange der Eimer aber sein maximales Fassungsvermögen (burst) noch nicht erreicht hat, hat dies auf die eintretenden Pakete noch keinen Einfluss. Kommt es aber zu einem Überlauf des Eimers, so können diese out-of-profile-Pakete entweder verworfen (drop) oder in ihrer Priorität heruntergestuft werden (markdown). Die weitere Behandlung dieser heruntergestuften Pakete obliegt dann der Entscheidung anderer Instanzen innerhalb des Switches (Drop bei überlaufenden Queues der Ausgangslinks) oder des Netzwerkes (Reaktion von nachfolgenden Routern auf diese niedrigere Priorität).

Mittels diesem Algorithmus wird eine Bandbreitenlimitierung oder auch Traffic-Shaping auf einen via rate konfigurierbaren Wert erreicht. Kurzzeitige Bandbreitenspitzen führen noch nicht zwingend zu Paketverlusten (drop) (und damit zur Korrektur der Flusskontrolle durch TCP) bzw. zu einer niedrigeren Priorisierung. Die Höhe der tolerierten Bandbreitenspitzen kann über den Burst-Parameter konfiguriert werden.

Das CatOS der Catalyst unterscheidet zwischen zwei Arten von Policern:

microflow Bandbreite für einen einzelnen Flow, mehrere Flows gleichen Typs bekommen werden einzeln betrachtet

aggregate Bandbreite für die Summe aller konkurrierenden Flows gleichen Typs

[1] Differentiated Service Class Priority

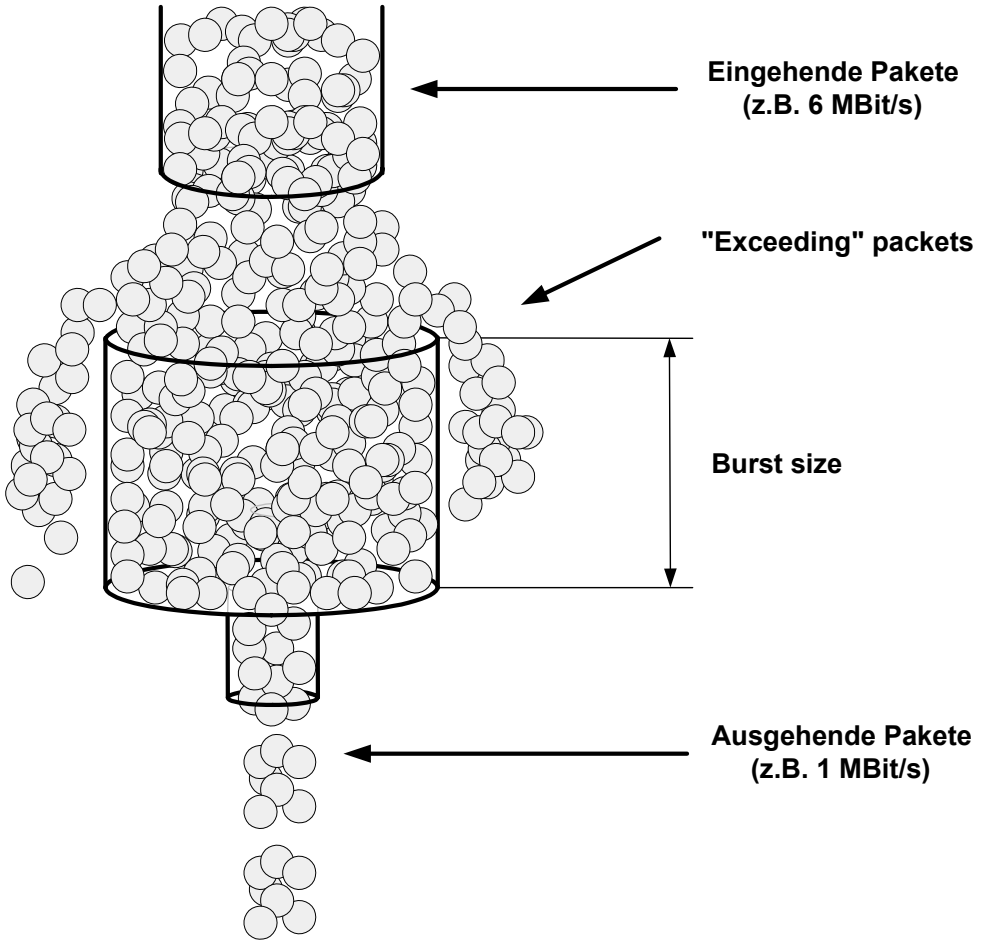

Abbildung 4.2: Bildliche Darstellung des Token-Bucket-Algorithmus zum Bandbreiten-Management

4.1.3 Congestion Management

Viel wichtiger als das Bandbreiten-Management durch Traffic-Shaping ist das Management von Congestions, also die Steuerung von Paketstaus bei überlasteten Ausgangslinks durch gezieltes Verwerfen von Paketen in Abhängigkeit der ihnen zugewiesenen Priorität bzw. Class-of Service (CoS). Erst an dieser Stelle werden Priorisierungen bestimmter Traffic-Flows und damit QoS im eigentlichen Sinn ermöglicht.

Ciscos Catalyst realisiert Congestion Management mittels verschiedener In- und Output-Queues innerhalb der Ports (siehe Abbildung 4.1). Ein- bzw. ausgehende Pakete eines Ports werden in je nach ihrer durch den CoS-Wert festgelegte Priorität in verschiedene Queues eingereiht. Sheduler bedienen die einzelnen Queues mit konfigurierbarer Gewichtung und steuern so die weitere Verteilung der Pakete aus den einzelnen Queues. Ausnahmen bilden

sogenannte „strict-priority-queues“, die exklusiv abgearbeitet werden. Die übrigenen Queues werden bei Verwendung von strict-priority-queues erst abgearbeitet, wenn die strict-priority-queue leer ist (Abbildung 4.1, Output-Port 2).

Congestions treten in der Regel nicht bei Input-Queues eines Ports auf, da die interne Verarbeitungsgeschwindigkeit von Switches und Routern hoch genug ist für die auftretenden Bandbreiten durch eingehenden Traffic.

Anders verhält es sich allerdings mit den Output-Queues der Ports. Treten die in 3.1.1 erwähnten Umstände (Aggregation mehrerer Links, Bandbreitendifferenzen, Übergang WAN-LAN) auf, so kommt es an dieser Stelle zwangsläufig zu Paketstaus. Diese Paketstaus können mit Ciscos Catalyst mit verschiedenen Queueingstrategien (RED, WRED, strict-queueing, ...) behandelt werden.

4.2 Installierte Versionen

Auf dem Core-Router sind zum Zeitpunkt des Projektes zwei Betriebssysteme installiert: IOS für die Realisierung von Funktionen auf Layer 3. Auf Layer 2 arbeitet das Betriebssystem CatOS. Die massgeblichen Funktionen für QoS sind noch innerhalb des CatOS implementiert. Aus Gründen der Vollständigeit sind im Folgenden die Versionsausgaben beider Betriebssysteme angegeben.

4.2.1 Cisco IOS

```
C6506:FU-Router>sh version
Cisco Internetwork Operating System Software
IOS (tm) MSFC Software (C6MSFC-IS-M), Version 12.0(7)XE,
EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)
TAC:Home:SW:IOS:Specials for info
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 30-Dec-99 02:55 by lstringr
Image text-base: 0x60008900, data-base: 0x60CAA000

ROM: System Bootstrap, Version 12.0(3)XE, RELEASE SOFTWARE
BOOTFLASH: MSFC Software (C6MSFC-BOOT-M), Version 12.0(3)XE1,
EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)

C6506:FU-Router uptime is 2 days, 8 hours, 44 minutes
System returned to ROM by power-on at 19:02:44 GMT Mon Apr 29 2002
System restarted at 10:34:31 GMT Fri Jul 5 2002
Running default software

cisco Cat6k-MSFC (R5000) processor with 57344K/8192K bytes of memory.
Processor board ID SAD04030BAB
R5000 CPU at 200Mhz, Implementation 35, Rev 2.1, 512KB L2 Cache
Last reset from power-on
Bridging software.
X.25 software, Version 3.0.0.
30 Virtual Ethernet/IEEE 802.3  interface(s)
```

```
123K bytes of non-volatile configuration memory.
4096K bytes of packet SRAM memory.

16384K bytes of Flash internal SIMM (Sector size 256K).
Configuration register is 0x2102

C6506:FU-Router>
```

4.2.2 CatOS

```
C6506:FU sh version
WS-C6506 Software, Version NmpSW: 5.3(2)CSX
Copyright (c) 1995-1999 by Cisco Systems
NMP S/W compiled on Oct 11 1999, 17:45:02

System Bootstrap Version: 5.3(1)

Hardware Version: 2.0  Model: WS-C6506  Serial #: TBA04070487

Mod Port Model               Serial #    Versions
--- ---- ------------------- ----------- ----------------------------------
1   2    WS-X6K-SUP1A-2GE    SAD04040AUD Hw : 3.1
                                         Fw : 5.3(1)
                                         Fw1: 5.1(1)CSX
                                         Sw : 5.3(2)CSX
                                         Sw1: 5.3(2)CSX
2   8    WS-X6408-GBIC       JAB040605RU Hw : 2.3
                                         Fw : 4.2(0.24)VAI78
                                         Sw : 5.3(2)CSX
3   8    WS-X6408-GBIC       JAB040405WK Hw : 2.3
                                         Fw : 4.2(0.24)VAI78
                                         Sw : 5.3(2)CSX
4   48   WS-X6248-RJ-45      SAD040806YF Hw : 1.1
                                         Fw : 4.2(0.24)VAI78
                                         Sw : 5.3(2)CSX
15  1    WS-F6K-MSFC         SAD04030BAB Hw : 1.3
                                         Fw : 12.0(7)XE,
                                         Sw : 12.0(7)XE,

       DRAM                    FLASH                   NVRAM
Module Total   Used    Free    Total   Used    Free    Total Used  Free
------ ------- ------- ------- ------- ------- ------- ----- ----- -----
1       65408K  30134K  35274K  16384K   5241K  11143K  512K  239K  273K

Uptime is 18 days, 21 hours, 49 minutes
```

4.3 QoS-Konfiguration CatOS

Auf den nächsten Seiten ist die von uns getestete Konfiguration kommentiert. Sie beschreibt die Klassifizierung und das Bandbreitenmanagement. Der Einsatz von Congestion Avoidance auf dem Core-Router macht keinen Sinn, da innerhalb des Gerätes keine Congestions auftreten. Aus diesem Grund wurden sie von uns auch nicht implementiert.

Die erste Option schaltet die erweiterten QoS-Funktionen der Hardware ein.

```
1 set qos enable
```

Die folgenden Direktiven setzen die DSCP-Map für die Markdown-Policer. Übersteigen die Raten eines Policers die definierten Werte, so können die überschüssigen Pakete entweder verworfen (`drop`) oder in ihrer Priorität heruntergesetzt (`markdown`) werden. Heruntergestufte Pakete werden vom Congestion Management überlasteter Queues eher verworfen als Pakete mit höherer Priorität.

Die mit diesen Direktiven aufgebaute DSCP-Map soll solchen heruntergestuften Paketen einen neuen DSCP-Wert von 10 geben, wenn der ursprüngliche DSCP-Wert 40 entsprach, einen Wert von 15 bei ursprünglichen 45 usw. Die gesetzten DSCP-Werte werden von den Output-Queues in das ToS-Feld der ausgehenden IP-Pakete geschrieben.

```
2  set qos policed-dscp-map 40,10:10
3  set qos policed-dscp-map 45,15:15
4  set qos policed-dscp-map 50,20:20
5  set qos policed-dscp-map 55,25:25
6  set qos policed-dscp-map 60,30:30
```

Die Policer-Definitionen müssen vor den ACL-Definitionen erfolgen. In unserer Konfiguration haben wir sechs verschiedene Policer gewählt. Die wichtigsten sind `policer_ssh` und `policer_web`, die den dahinterliegenden Diensten bestimmte Bandbreiten garantieren sollen: 64 kBit/s für einzelne ssh-Flows (`microflow`) und insgesamt 3 MBit/s für jeglichen http-Traffic (`aggregate`).

Die Policer `policer_p2p` und `policer_icmp` dienen der Beschränkung der Bandbreiten dieser Dienste auf sehr geringe Werte.

Für administrative Aufgaben und Email wurde ein eigener Policer `policer_admin` definiert, der Diensten des Netzwerkmanagements und Kommunikationsdiensten ingesamt 500 kBit/s garantiert. Der Policer `policer_default` beschränkt allen anderen Traffic auf maximal 1,5 MBit/s.

```
7  set qos policer microflow policer_ssh rate 64 burst 3 policed-dscp
8  set qos policer microflow policer_p2p rate 14 burst 1 drop
9  set qos policer aggregate policer_admin rate 500 burst 20 policed-dscp
10 set qos policer aggregate policer_web rate 3000 burst 20 policed-dscp
11 set qos policer aggregate policer_icmp rate 9 burst 1 policed-dscp
12 set qos policer aggregate policer_default rate 1500 burst 20 policed-dscp
```

Die folgenden Direktiven erzeugen ACLs zur Klassifizierung der zu routenden Pakete anhand von Protokollen und Diensten. Die Klassifikation wird in Form eines DSCP-Values in den Paket-Header geschrieben. Die Pakete werden nach der Klassifikation an die bereits definierten Policer übergeben.

Als erste Option setzt die default-ACL die DSCP-Werte für alle Pakete, auf die keine der nachfolgenden ACLs zutrifft. Dieser Wert ist das Kriterium für die Priorisierung von Traffic durch an anderer Stelle einsetzendes Congestion Managements in In- und Output-Queues.

Jeglicher interner Traffic (Quell- und Ziel-IP innerhalb des Netzes 141.28.0.0/16) wird nicht von QoS erfasst und behandelt.

Allen sonstigen Pakete, für die eine entsprechende Rule vorhanden ist, werden die in der Rule definierten DSCP-Werte zugewiesen. Diese Werte sind je nach Wichtigkeit der entsprechenden Dienste unterschiedlich. Dieselbe Rule weist ebenfalls den erfassten Paketen einen vorher definierten Policer zu.

```
13 # default-action
14 set qos acl default-action ip dscp 45  aggregate policer_default
15
16 # don't affect internal traffic
17 set qos acl ip acl_qos dscp 0 udp 141.28.0.0 0.0.255.255 141.28.0.0 0.0.255.255
18 set qos acl ip acl_qos dscp 0 tcp 141.28.0.0 0.0.255.255 141.28.0.0 0.0.255.255
19
20 #Web/FTP Rules
21 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any eq 80 any
22 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any any eq 80
23 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any eq 443 any
24 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any any eq 443
25 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any eq 8080 any
26 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any any eq 8080
27 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any eq 21 any
28 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any any eq 21
29 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any eq 20 any
30 set qos acl ip acl_qos dscp 50 aggregate policer_web tcp any any eq 20
31
32 #ssh qos ACL
33 set qos acl ip acl_qos dscp 60 microflow policer_ssh tcp any  eq 22 any
34 set qos acl ip acl_qos dscp 60 microflow policer_ssh tcp any  any  eq 22
35
36 #mail & administrative traffic
37 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 110 any
38 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 110
39 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 25 any
40 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 25
41 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 143 any
42 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 143
43 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 993 any
44 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 993
45 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 995 any
46 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 995
47 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 465 any
48 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 465
49 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 119 any
50 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 119
51 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 53
52 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 53 any
53 set qos acl ip acl_qos dscp 55 aggregate policer_admin udp any any eq 53
54 set qos acl ip acl_qos dscp 55 aggregate policer_admin udp any eq 53 any
55 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any any eq 161
56 set qos acl ip acl_qos dscp 55 aggregate policer_admin tcp any eq 161 any
57 set qos acl ip acl_qos dscp 55 aggregate policer_admin udp any any eq 161
58 set qos acl ip acl_qos dscp 55 aggregate policer_admin udp any eq 161 any
59
60 # icmp-traffic
61 set qos acl ip acl_qos dscp 40 aggregate policer_icmp icmp any any
62
63 # unwanted peer2peer
64 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any eq 1214 any
65 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any any eq 1214
66 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any eq 1234 any
```

```
67 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any any eq 1234
68 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any eq 4661 any
69 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any any eq 4661
70 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any eq 4662 any
71 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any any eq 4662
72 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any eq 6346 any
73 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any any eq 6346
74 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any eq 6347 any
75 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any any eq 6347
76 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any eq 6699 any
77 set qos acl ip acl_qos dscp 10 microflow policer_p2p tcp any any eq 6699
```

Die letzten Direktiven aktivieren die vorläufige Konfiguration (`commit`) und weisen die ACLs den entsprechenden Ports zu (`map`).

```
78 commit qos acl all
79 set qos acl map acl_qos 2/1-8,3/1-8,4/1-48
```

4.4 Ergebnisse und Messungen

Die im vorangegangen Abschnitt beschriebene Konfiguration wurde in dieser Form auf dem Gerät vorgenommen und aktiviert. Eine ungewöhnliche Belastung der Hardware durch diese Aktivierung im Vergleich zum Normalbetrieb konnte nicht festgestellt werden. Die Classifier arbeiteten gemäß ihrer Konfiguration. Der Core-Router modifizierte die ToS-Felder der IP-Header auf Basis der konfigurierten DSCP-Values, was sich mit mittels `crl_tos` aus der CoralReef-Suite ([Moo02]] auf einem an einem Monitorport des Routers angeschlossenen Testrechners bestätigen lies. Damit könnten diese Klassifizierungsinformationen von nachfolgenden Instanzen entlang des Übertragungsweges weiter verwendet werden.

Die Wirksamkeit der gesetzten Policer und die unterschiedliche Behandlung von einzelnen Flows verschiedener Klassen konnte nachgewiesen werden. Die Ergebnisse lagen innerhalb der konfigurierten Parameter. Für diese Messungen wurden hauptsächlich `trafshow`, `iperf`[2], `hping`[3] und `sting`[4] verwendet. Die in [AC01], [Sto99], [Sav99] und [Sie00] dokumentierten Erfahrungen sollten für eigene Messungen herangezogen werden.

Allerdings zeigt die Aktivierung von QoS auf dem Core-Router, mit seiner installierten OS-Version, unerwünschte Nebenwirkungen. So reichte die Aktivierung von QoS mittels `set qos enable` unter CatOS ohne weitere Konfiguration von ACLs oder Policern aus, um die Routingfunktionen des Gerätes zu stören. Die Ursache dieser Fehlfunktionen liegt vermutlich in einigen Software-Bugs des Betriebssystems, die in späteren Releases behoben wurden.

[2] Utility zur Durchsatzmessung von TCP und UDP
[3] Ermittlung von Paketverlusten unterschiedlicher Protokolle und Dienste
[4] Messung von Paketverlusten auf Hin- und Rückweg von TCP

Aufgrund dieser reproduzierbaren Fehler musste auf weitere Arbeiten am Gerät verzichtet werden. Die angesprochenen Messungen sind in dieser Arbeit nicht weiter dokumentiert, da es lediglich kurzfristige Prüfungen der Wirksamkeit der Konfiguration waren. Die für repräsentative Ergebnisse notwendigen Langzeitmessungen über mehrere Tage und Auswertungen der Ergebnisse konnten damit nicht durchgeführt werden.

Kapitel 5
Designvorschlag einer QoS-Implementierung in der Fachhochschule Furtwangen

5.1 Möglichkeiten mit QoS im vorliegenden Fall

Wie in Kapitel 3 erwähnt ist der Einsatz von QoS dann empfehlenswert, wenn es infolge von Paketstaus in den Ausgangsqueues von Routern zu Paketverlusten kommt. Mit QoS wird die Auswahl der zu verwerfenden Pakete nicht mehr rein zufällig vorgenommen, sondern anhand von definierten Prioritäten auf Basis von Layer 3- und Layer 4-Merkmalen.

Die Entscheidung, welche Pakete in einer solchen Stausituation verworfen werden müssen und welche nicht, kann nur innerhalb der überlasteten Queue gefällt werden. QoS muss also an genau der Stelle ansetzen, an der der Stau auftritt.

Im vorliegenden Fall ist das Problem die - für die hohe Anzahl der Nutzer zu gering dimensionierte - externe Anbindung. Die meistens überlastete Aussenanbindung führt zu Paketverlusten[1] innerhalb des Routers zum Belwue-WAN. Einzig und allein an dieser Stelle kann QoS sinnvoll zur Steuerung von Paketstaus mittels Congestion Managements eingesetzt werden.

Innerhalb des Campus-Netzes kommt es in der Regel nicht zu Kapazitätsengpässen, damit auch nicht zu überlasteten Queues des Core-Switches und folglich nicht zu Paketverlusten[2]. Priorisierungen im Sinne von QoS können

[1] Diese Behauptung lässt sich mit den internen Statistiken der Bandbreitennutzung des Rechenzentrums belegen.

[2] nachweisbar mit den Interface-Statistiken über gedropte Pakete des Core-Routers

damit auch nicht auf dem Core-Router vorgenommen werden, weil die QoS-Implementierung auf dem Switch keinerlei Informationen bekommt, wie sich Trafficaufkommen und Paketverluste auf dem WAN-Router zum Belwue zur jeweiligen Situation verhalten.

Eine Verlagerung des Paketstaus vom Router weg zum Switch durch Shaping des zum Router gerichteten Interfaces auf beispielsweise 4 MBit/s bzw. zu einem späterem Zeitpunkt auf 622 MBit/s wird von dem vorgefundenen Releases von IOS und CatOS nicht unterstützt.

Damit steht auf dem Switch bei dieser Infrastruktur keine realisierbare Möglichkeit zur Steuerung von Paketstaus im Sinne von QoS für ein- und ausgehenden Traffic zur Verfügung. Die Bandbreitennutzung einzelner oder ganzer Klassen von Flows lässt sich mittels Policer aber auf dem Core-Router beeinflussen.

5.2 Designvorschlag

Wie in Abschnitt 5.1 dargestellt, muss die Manipulation der Traffic-Flows zwingend am WAN-Router erfolgen. Die Klassifizierung und damit die Zuordnung von Prioritäten zu unterschiedlichen Flows kann allerdings auf dem Core-Router der FH-Furtwangen gemacht werden. Das Rechenzentrum hat damit die Möglichkeit, selbst zu definieren, wie der generierte ausgehende Traffic ausserhalb ihres Einflussbereiches behandelt werden soll. Der WAN-Router muss dieser Klassifikation des Core-Routers vertrauen (trust-dscp). Die Behandlung von eingehenden Traffic durch den WAN- oder Core-Router müsste am Endpunkt der WAN-Strecken in Freiburg erfolgen. Eine Manipulation eingehenden Traffics auf FH-Seite würde sich nur auf die Flusskontrolle von TCP auswirken. Bereits angekommener UDP-Traffic hat die Anbindung bereits saturiert, bevor auf FH-Seite Massnahmen dagegen unternommen werden können.

Die Verteilung der QoS-Funktionalitäten hat nebenbei noch einen nicht unwichtigen Vorteil: Die CPU-Belastung des WAN-Routers durch QoS kann so gering wie möglich gehalten werden. Die Klassifizierungsentscheidungen kann der Core-Router im Gegensatz zu softwarebasierenden Routern in Hardware durch sog. ASICs [3] erheblich schneller und ohne zusätzliche Belastung der CPU durchführen.

Die Klassifizierung belastete den Core-Router bei unseren Tests mit Bandbreiten von max. 4 MBit/s nur unwesentlich mehr. Aussagen über die Belastung bei höheren Bandbreiten bis 622MBit/s sowie über die Belastung des WAN-Routers durch zusätzliches Sheduling und Queueing können wir an dieser Stelle nicht abgeben, da praktische Tests in dieser Konfiguration nicht möglich waren.

[3] Application Specific Integrated Circuits

5.2.1 Klassifizierung

Die Klassifizierung der zu routenden Pakete wird vom Core-Router vorgenommen. Da den Prioritätsangaben der IP-Pakete aus dem Campus-Netz nicht vertraut werden kann[4], muss jedes nach extern zu routende Paket mindestens mit einem Defaultwert klassifiziert werden.

Eine Aufteilung in mindestens vier Trafficklassen wie in Tabelle 3.1 erscheint sinnvoll.

5.2.2 Traffic Conditioning

Das Bandbreiten-Management kann auch noch auf dem Core-Router vorgenommen werden. P2P-Traffic und anderer unerwünschter Traffic kann mittels Policer auf eine maximale Bandbreite von z.B. 128 kBit/s limitiert werden, anderen Diensten wie ssh können mittels microflow-Policern[5] und höheren Prioritäten Bandbreiten garantiert werden. Ähnliches gilt auch für andere denkbare Traffic-Klassen.

Wichtig für die Konfiguration ist ein genaues Verständnis der verschiedenen DSCP-Werte zur Prioritätssteuerung, da sie ein sehr leistungsfähiges Instrument zur Manipulation von Verkehrsströmen darstellen.

5.2.3 Congestion Management

Das Congestion Management muss wie bereits mehrfach erwähnt vom WAN-Router vorgenommen werden. Dafür werden die im ToS-Feld der IP-Header kodierten DSCP-Werte als Prioritätsmerkmal benutzt. Wie sich innerhalb des Routers das Queueing und Sheduling realisieren lässt, kann im Rahmen dieses Projekts nicht dargestellt werden, die Dokumentation zu diesen Geräten verspricht allerdings ausgefeilterere Möglichkeiten als die in Hardware implementierten Mechanismen der PFC der Catalyst.

5.2.4 Application Level Proxies

Eine Messung über den Zeitraum von 10 Tagen hat ergeben, dass der größte Anteil der Bandbreite für Traffic auf dem http-Port beansprucht wird. Dies ist durchaus so gewollt.

[4] Jedes Betriebssystem kann eigene Werte in das ToS-Feld der ausgesendeten Pakete schreiben.

[5] microflow-Policer in Verbindung mit mark-down bieten sich für ssh geradezu an. Eine Limitierung auf 64 kBit/s reicht für interaktives Arbeiten aus. Ein Filetransfer über das ssh-Protokoll mit scp lässt sich allerdings nicht auf Layer-4 von interaktiven Arbeiten unterscheiden. Mittels mark-down können Flows, die die 64 KBit/s überschreiten eine niedrigere Priorität zugewiesen werden. Sie erhalten trotzdem höhere Transferraten, wenn die Netzbelastung dies zulässt, werden aber auch eher verworfen, wenn Traffic mit höherer Priorität konkurriert.

Momentan ist allerdings nicht bekannt, ob dieser Traffic wirklich zu 100% durch http erzeugt wird, oder ob anderer Traffic über diesen Port getunnelt wird. Dieses Tunneln wird dann zum Problem, wenn bekannt wird, dass sich benachteiligende QoS-Policer durch geschickte Wahl anderer Ports umgehen lassen (z.B. Tunneln von P2P-Traffic über Port 80 oder 21).

Da es nicht Sinn und Zweck von Switches oder Routern ist, Traffic auf ihren Inhalt zu prüfen, muss die Tunnel-Problematik anderweitig gelöst werden. Das Problem lässt sich nur mit (transparenten) Application Level Proxies auf den entsprechenden Ports angehen und auch nur teilweise lösen. Es kann aber potenziell alle QoS-Anstrengungen torpedieren. Daher sollte nach Implementierung von QoS wenigstens geprüft werden, ob sich das durch QoS zu manipulierende Verkehrsverhalten auch wirklich ändert oder ob diese Änderungen erst durch Tunneling entstehen.

Kapitel 6
Fazit

Der geplante Verlauf des Projektes konnte mit dem tatsächlichen Ablauf bis zur Möglichkeitsanalyse (Phase 3, 2.1.3) zeitlich und inhaltlich eingehalten werden. Während der Analyse zeigte sich allerdings allmählich, dass von der ursprünglichen Planung abgewichen werden musste und die Realisierung des Projektzieles in ihrem anfänglichen Umfang unwahrscheinlicher wurde.

Die Ursachen dafür waren im Einzelnen:

1. Ein massgebliches Feature, die Priorisierung definierbarer Trafficflows gegenüber anderen konkurrierenden Flows, lässt sich aus technischen Gründen auf dem Core-Router allein nicht umsetzen, sondern erst auf dem nachfolgenden WAN-Router zum Belwue. Die technischen Gründe dafür sind in Abschnitt 5.1 beschrieben.
   Damit war die gewünschte Lösung durch geeignete Konfiguration des Core-Router allein nicht möglich. Um das Projektziel zu erreichen, hätte der Arbeitsbereich auch auf Belwue-Systeme ausgeweitet werden müssen, was allerdings im zeitlichen Rahmen des Projektes nicht möglich war. Dazu wären Eingriffe auf Systeme ausserhalb des Zuständigkeitsbereiches der FH Furtwangen notwendig gewesen, die sich zumindest organisatorisch nicht kurzfristig umsetzen liessen.

   Aus diesem Grunde wurde beschlossen, die Konfiguration der Infrastruktur der Fachhochschule insoweit vorzubereiten, dass für eine ganzheitliche Lösung auf den externen Systemen des BelWü so wenig Änderungen wie möglich zusätzlich gemacht werden müssen.

2. Im Vorfeld nicht bekannte Softwarefehler im Betriebssystem des Routers führten bei aktiviertem QoS zu reproduzierbaren Störungen in Teilen des Campus-Netzes, die jedoch im Zussamenhang mit dem QoS aus technischer Sicht nicht erklärbar sind. Da die Auswirkungen dieses Effektes in dem Produktivsystem nicht akzeptabel waren, musste bei Verwendung der installierten Betriebssysteme auf die Aktivierung von QoS zwangsläufig verzichtet werden. Kapitel 4.4 liefert eine detailierte technische Beschreibung des Fehlverhaltens.

Dieser Fehler trat während der Arbeiten an dem Produktivsystem zwei Mal mit empfindlichen Folgen für die angeschlossenen Netze auf. Eine Fehlerbehebung hätte ein Update der Firmware auf den Geräten notwendig gemacht. Da dieses Update innerhalb des zur Verfügung stehenden Zeitrahmens nicht durchführbar war, und dafür nötige Serviceverträge mit dem Hersteller erst noch abgeschlossen werden, konnte eine Implementierung von QoS mit den vorhandenen Firmware-Versionen nicht durchgeführt werden.

Persönlich konnten wir sehr wertvolle und interessante Erfahrungen im Cisco-Umfeld, insbesondere mit den Betriebssysteme IOS und CatOs, dem Betrieb in einem grösseren Campus-Netz und nicht zuletzt der damit verbundenen Verantwortung sammeln. Desweiteren konnten wir unsere Kenntnisse im Bereich QoS, insbesondere dem Differentiated-Service-Modell, der Charakteristik von Verkehrsflüssen und besonderen Eigenschaften von Transport-Protokollen weiter vertiefen. Bedauerlicherweise liessen sich aus vorher genannten technischen Gründen keine Messungen durchführen.

Insgesamt stellte dieses Projekt für uns eines der bislang interessantesten Betätigungsfelder unseres Studiums dar.

Wir möchten uns bei Herrn Claus-Peter Rohner, dem Leiter des Rechenzentrums der FH Furtwangen, für den Projektauftrag, die Unterstützung in technischen und organisatorischen Angelegenheiten und nicht zuletzt für das in uns gesetzte Vertrauen bedanken. Herrn Prof. Dr. Friedbert Kasper gilt unser Dank für die fachliche Betreuung.

Konfiguration CatOS

```
1
2  Configuration cutted. This is the public version of the document.
3
4  If you have a Question about the config contact
5  morrow@unfug.org or
6  vertigo@unfug.org
```

Glossar

B

Bandbreite
Übertragungskapazität eines Links in Bits/s.

Burst
Parameter für den Token-Buket-Algorithmus bei der Definition eines Policers. Spezifiziert die Puffergröße eines Token-Buckets.

C

CatOS
Catalyst Operation System, ursprüngliches Betriebssystem auf Catalyst-Switches der Firma Catalyst (wurde später von Cisco aquiriert)

Congestion
Paketstauung innerhalb einer Queue aufgrund Überlastung eines Links.

Congestion Avoidance
Algorithmus der Flusskontrolle von TCP. Reduziert die Übertragungsrate eines TCP-Flows durch Verkleinerung des Congestion Windows bei Paketverlusten aufgrund Überlastung von Queues.

CoS
Class of Service.

D

Differentiated Services
QoS-Modell. Steuerung im Sinne von QoS durch das Netzwerk.

DSCP Value
Differentiated Service Class Priority Value
Neue Definition des ToS-Feldes innerhalb des IP-Headers zur Nutzung von Differentiated Services. Definition in [KN98].

F

Flow
Zusammenfassung aller Datenpakete, die zu einer gemeinsamen Datenübertragung gehören. Ein Flow wird bezeichnet mit Protokoll Quell- und Ziel-IP, und Quell- und Ziel-Ports.

Frame
Datenpaket auf OSI-Layer 2.

I

Integrated Services
Qos-Modell. Reservierung benötigter Bandbreiten durch die Anwendung mittels geeignetem Protokoll.

IOS
Internet Operation System, einheitliches Betriebssystem auf Cisco-Geräten

J

Jitter
Varianz der Verzögerungszeiten der Pakete eines Flows.

L

LAN
Local Area Network, lokal begrenztes Netz mit hohen Übertragungsraten

M

MSFC
Multi Layer Feature Card. Erweiterungsmodul für die Catalyst-Architektur. Erweitert Catalyst-Switches u.a. um Layer3-Funktionalität.

Multi Layer Protocol Switching
Oberbegriff für neuartige Routing-Technologie von Cisco. Ermöglicht Routing-Funktionalität in Hardware auf Switchen und dadurch gegenüber konventionellen Routern sehr performant.

P

Paket
Datenpaket auf OSI-Layer 3.

PFC
Policy Feature Card. Bestandteil der MSFC für Cisco-Catalyst
Ermöglicht u.a. Klassifizierung, Markierung und Behandlung (Classification, Marking, Policing) von Traffic auf Basis von Policy Maps.

Policing
Massnahme zur Bandbreiten-Limitierung von einzelnen oder ganzen Klassen von Trafficflows durch gezielte Paketverluste.

Q

Queueing
Zuweisung von Frames zu Queues und Zwischenspeicherung.

R

Rate
Parameter der Definition eines Policers. Spezifiziert die nutzbare Bandbreite einer Klasse von Flows in kBit/s. Übersteigt der tatsächliche Wert dieses Mass, so gilt der Flow als „out-profile “und kann entweder verworfen oder niedriger priorisiert werden.

Round-Trip Time (RTT)
Paketlaufzeit von Sender zum Empfänger und zurück. Massgeblicher Parameter für die Qualität interaktiver Services. Ausserdem Parameter für die Flusssteuerung von TCP.

S

Scheduling
Abarbeitungsfolge mehrerer Queues. Mehrere Scheduling-Algorithmen möglich: Round Robin, Weighted Round Robin, strict-queueing ...

Service Level Agreement (SLA)
Definition des Umfangs von Service-Dienstleistungen zwischen Netzbetreiber und -kunden.

Shaping
Massnahme zur Bandbreiten-Limitierung von einzelnen oder ganzen Klassen von Trafficflows durch gezielte Verzögerung von Paketen ohne Paketverluste.

Slow Start
Algorithmus der Flusskontrolle von TCP. Steuert durch allmähliche Vergrösserung des Congestion Window die Übertragungsrate eines TCP-Flows

T

ToS
„Type of Service“, Feld innerhalb des IP-Headers zur Prioritätsangabe. Die weitere Behandlung des IP-Paketes erfolgt mit den von Cisco implementierten QoS-Algorithmen auf Basis des Feldes.

W

WAN
Wide Area Network

Literaturverzeichnis

[AC01] Alan Clarkson, Paul Kummer, Robin Tasker. *A Project to Investigate the Effectiveness of QoS Techniques*. http://icfamon.dl.ac.uk/I2QoS/final-report.pdf, 2001. 29

[Bon02] Boney, James. *Cisco IOS in a Nutshell*. O'Reilly & Associates Inc., 1. Auflage, 2002. ISBN 1-56592-942-x. 7

[Cis00a] Cisco. *Catalyst 6000 and 6500 Series Software Configuration Guide*. Technischer Bericht, Cisco, 2000. 5

[Cis00b] Cisco. *Troubleshooting Catalyst Switches (Part 2)*, 2000. 12

[KN98] K. Nicols, etal. *RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. ftp://ftp.isi.edu/in-notes/rfc2474.txt, 1998. 39

[Moo02] Moore, David, Keys, Ken, Koga, Ryan, Lagache, Edouard, und k. claffy. *The CoralReef software suite as a tool for system and network andministrators*. www.caida.org/outreach/papers/2001/CoralApps/CoralApps.pdf, 2002. 29

[Sav99] Savage, Stefan. *Sting: a TCP-based Network Measurement Tool*. http://www.cs.washington.edu/homes/savage/papers/Usits99.pdf, 1999. 29

[Sie00] Siemens, Edgar. *Realisierung und Bewertung eines QoS-Dienstes im Campus-Netz*. Diplomarbeit, Universität Hannover, Lehrgebiet Rechnernetze und Verteilte Systeme, 2000. 19, 29

[Ste94] Stevens, Richard. *TCP/IP Illustrated, Vol. 1*. Addison-Wesley, 17. Auflage, 1994. ISBN 0-201-63346-9. 13

[Sto99] Stoy, Robert und Jähnert, Jürgen. *Test of CISCO's IP QoS Implementation considering Differntiated Services*. http://www.cnaf.infn.it/~ferrari/tfng/doc/ds/dstest-unist-v0.9.doc, 1999. 29

[Tan98] Tanenbaum, Andrew S. *Computernetzwerke*. Prentice Hall, 1998. ISBN 3-8272-9568-8. 13, 14

[Wan01] Wang, Zheng. *Internet QoS, Archtitectures and Mechanisms for Quality of Service*. Morgan Kaufmann Publisher, 2001. ISBN 1-55860-608-4. 16

[Web00] Webb, Karen. *Multi Layer Switched Netzwerke*. Markt+Technik Verlag, München, 2000. ISBN 3-8272-5854-5. 2

Index