Veröffentlicht in: U. Killat, W. Lamersdorf (Hrsg.): Kommunikation in Verteilten Systemen (KIVS) - 12. GI/ITG-Fachtagung,
Februar 2001, Hamburg,
Informatik Aktuell, Springer Verlag, ISBN 3-540-41645-5, S. 21-33

Policy-basiertes Metering für IP-Netze

Georg Carle, Sebastian Zander, Tanja Zseby

GMD FOKUS
Kaiserin-Augusta-Allee 31
D-10589 Berlin
[carle, zander, zseby]@fokus.gmd.de
http://www.fokus.gmd.de/glone/

Abstract. Die kommerzielle Erbringung von IP-Diensten macht neben der Kontrolle des Zugangs zu Diensten über Authentifizierung und Autorisierung eine leistungsfähige Messinfrastruktur zur Erfassung der Ressourcennutzung erforderlich. In Differentiated Services Netzen müssen dazu messtechnische Lösungen zur Erfassung von Accounting-Daten und zur Überprüfung der im Service Level Agreement zugesagten Grenzwerte der Dienstqualität bereitgestellt werden. Die häufig heterogene Messinfrastruktur und die effiziente Aufteilung von Messaufgaben im Netz erfordern eine flexible Architektur zur Ansteuerung und Verteilung. Es wird eine policy-basierte Architektur für verteiltes Metering zur Erbringung der geforderten Messaufgaben vorgestellt. Messaufgaben werden dazu in einer einheitlichen Meterkonfigurationssprache ausgedrückt, und eine automatische Verteilung sowie Umsetzung auf spezifische Meter-Konfigurationen für unterschiedliche Meter unterstützt. Hierdurch kann auf dienstspezifische Accounting-Anforderungen und dynamische Änderungen der Verkehrscharakteristik unter Berücksichtigung von Funktionalität, Auslastung und maximaler Leistungsfähigkeit einzelner Meter eingegangen werden.

Einleitung

Innerhalb der IETF-Arbeitsgruppe AAA (Authentication, Authorization, Accounting) und der IRTF-Forschungsgruppe AAAARCH (Authentication, Authorization, Accounting Architecture) wird an Lösungen zur Bereitstellung einer generischen AAA-Architektur [Laat00] für IP-Dienste und anwendungsorientierte Dienste gearbeitet. Nach Authentifizierung und Autorisierung eines Kunden erfolgt die Konfiguration des geforderten Dienstes. Handelt es sich bei dem Dienst z.B. um die bevorzugte Weiterleitung von IP-Paketen in einem Differentiated Services Netzwerk, so beinhaltet dies die Konfiguration der Marker und Scheduler. Die AAA-Architektur sieht sogenannte Application Specific Modules (ASMs) für die Konfiguration von Service Equipment [Voll00a] und Accounting Equipment [CaZZ00] vor. Bei Differentiated Services IP-Netzen ist neben der messtechnischen Erfassung des Ressourcenverbrauchs auch eine kontinuierliche oder auf Stichproben basierende Überprüfung der erbrachten Dienstqualität wichtig. In Netzen mit differenzierten Dienstgüten fallen also zwei unterschiedliche Messaufgaben an:

- Messung des Ressourcenverbrauchs für eine verursachergerechte Abrechnung
- Messung von Dienstqualitätsparametern zur Validierung der vereinbarten Qualitätsstufe

Um diese Messaufgaben zufriedenstellend zu bewältigen, wird eine Infrastruktur mit Messinstanzen an verschiedenen Punkten im Netz benötigt. Die Parameter zur Durchführung der Messung wie die jeweils benötigten Messwerte, Genauigkeit der Messung, Messintervall, etc. können dabei für verschiedene Tarife, Dienstklassen, Kundentypen, Tageszeiten oder andere Faktoren variieren. Viele der Parameter lassen sich aus dem zwischen Kunden und Provider vereinbarten Service Level Agreement (SLA) ableiten. Ein weiterer wichtiger Punkt stellt die meist heterogene Infrastruktur in vielen Provider-Netzen dar. Um das Zusammenspiel der vorhandenen Komponenten zu ermöglichen, muss eine konsistente Ansteuerung der unterschiedlichen Messwerkzeug-Typen unterstützt werden.

In diesem Beitrag stellen wir eine Architektur vor, in der Konfigurationsparameter in Form von Meter-Policies zur Ansteuerung einer verteilten heterogenen Messarchitektur eingesetzt werden. Die Meter-Policies werden aus den im SLA festgelegten Kenngrößen (z.B. Tarif, Messverfahren, Messhäufigkeit) abgeleitet. Damit wird die Erfassung der Ressourcennutzung über Accounting-Meter und die Überprüfung der im SLA vereinbarten QoS-Parameter über passive Messungen realisiert.

Das Dokument gliedert sich wie folgt: Kapitel 2 gibt einen Überblick über existierenden Lösungen zur messtechnischen Erfassung von Daten über den Ressourcenverbrauch und der bereitgestellten Qualität. In Kapitel 3 wird das Konzept des policy-basierten Metering vorgestellt. Dabei wird die Anwendung zur Erfassung der Ressourcennutzung und der Validierung von SLAs verdeutlicht. In Kapitel 4 wird die Meter-Architektur zur Ansteuerung verschiedener Messkomponenten vorgestellt. Kapitel 5 zeigt anhand von Messungen, wie die Leistungsfähigkeit eines Meter durch hierarchische policy-basierte Filterung verbessert werden kann. Kapitel 6 fasst die erzielten Ergebnisse zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

Metering für Accounting

Algorithmen zur Filterung und Klassifikation von IP-Paketen werden nicht nur in Accounting-Metern, sondern auch in verschiedenen anderen Anwendungen und Einsatzgebieten benötigt (z.B. Routing, Firewalling, Dienstdifferenzierung, etc.). Daher müssen bei der Beurteilung geeigneter Algorithmen die speziellen Anforderungen, die sich durch das Einsatzgebiet Accounting ergeben, berücksichtigt werden. Accounting stellt die Basis für die Rechnungserstellung dar. Daher sollten neu auftretende Datenströme von Beginn an erfasst werden. Spätestens nach erfolgreichem Abschluss der Autorisierung müssen daher die entsprechenden Filter-Regeln im Accounting-Meter zur Erfassung des Datenstromes existieren.

Eine Möglichkeit zur Erfüllung dieser Anforderung besteht in der permanenten Erfassung aller auftretenden Datenströme. Das von Cisco entwickelte Meter NetFlow [Cisc99] arbeitet nach diesem Prinzip. Auch das konfigurierbare Meter NeTraMet [RFC2063] stellt einen Modus zur Verfügung, in dem eine automatische Klassifikation aller Datenströme erfolgt. Die Erfassung aller Flows kann allerdings besonders bei feiner Granularität zu hohen Datenmengen führen. In Szenarien, in denen die Weiterleitung bestimmter Pakete (z.B. Best Effort Verkehr) umsonst ist oder durch eine Flat Rate abgedeckt wird, führt dies meist zu unnötigem Klassifikationsaufwand und zu überflüssigen Einträgen in der Flow-Tabelle des

Meters. Ein weiterer Nachteil dieser Variante ist die notwendige feste Definition der Attribute zur Eingruppierung der Pakete. Damit ist die Granularität der Erfassung fest vorgegeben und für alle Datenströme identisch. Die zweite Variante ist das schnelle dynamische Hinzufügen und Entfernen von Klassifikations-Regeln. Dies erfordert einen flexiblen Klassifizierer, der eine Veränderung der Regeln zur Laufzeit unterstützt.
Je nach eingesetztem Tarifmodell kann Accounting neben der Behandlung einer großen Anzahl an Filterregeln, auch die Klassifizierung anhand mehrerer Header-Felder erfordern. In [NARUS] wird ein System vorgestellt, in dem sogar eine semantische Verkehrs-Analyse zur Auswertung von Applikations-Informationen stattfindet, um Datenströme einzelnen Sessions auf Anwendungsebene zuzuordnen. Die Klassifikationsanforderungen für Accounting machen eine Aufteilung von Mess-Aufgaben auf verschiedene Messinstanzen zweckmäßig. So können zum Beispiel Filter-Regeln auf verschiedene Messinstanzen die der Datenstrom durchquert verteilt werden, (z.B. auf Ingress- und Egress-Router) um die Belastung der einzelnen Meter zu reduzieren.

Accounting Meter

Das Accounting-Meter NeTraMet [RFC2063] ist aus der Arbeit der IETF Arbeitsgruppe Real-Time Traffic Flow Measurement (RTFM) entstanden. Die RTFM Architektur setzt sich aus einem Meter, einem Meter Reader und einem Meter Manager zusammen. Das Meter lässt sich über SNMP konfigurieren und auslesen.

Cisco NetFlow [Cisc99] ist ein Router-internes Meter speziell für Cisco Router. Es setzt das Betriebssystem Cisco IOS voraus und bietet im Vergleich zu NeTraMet nur wenige Konfigurationsmöglichkeiten. Es werden alle IP-Flows in der feinsten Granularität gemessen und nur einige vorgegebene Aggregierungs-Schemata zur Verfügung gestellt.

Linux Netfilter [Netf00] ist ein Klassifizierer im Linux Kern, der zur Realisierung verschiedener Anwendungen wie Firewalling, NAT und Accounting verwendet werden kann. Pakete können an verschiedenen Stellen im Netzwerk-Kern erfasst und klassifiziert werden. Der Klassifizierer wird über die Kommandozeile konfiguriert. Tabelle 1 vergleicht die Hauptmerkmale der vorgestellten Meter.

	NeTraMet	**NetFlow**	**NetFilter**
Flow-Definition	Bidirektional (optional unidirektional)	Unidirektional	Variabel
Flow-Granularität	Variabel	Festes Set von Attributen, einige feste Aggregation-Schemata	Variabel
DiffServ Codepoint	JA	JA	JA
RSVP Flowspec	Erweiterungs-Konzept in [CaMM98]	NEIN	NEIN
Ipv6 Adressen	JA	NEIN	JA
Multicast Attribute	Geplant	Geplant für Cisco IOS 12.0(7)T	NEIN
Sampling	JA	NEIN	NEIN
QoS Messungen	JA (RTT)	NEIN	NEIN
Unterstützte Betriebssysteme	DOS, Linux, BSD, Solaris, IRIX	Cisco IOS	Linux
Kosten	Frei erhältlich	ca. $ 5000,- pro Lizenz	Frei erhältlich

Tabelle 1: **Vergleich der Accounting-Meter NeTraMet, NetFlow und Netfilter**

NeTraMet bietet die höchste Flexibilität der vorgestellten Meter. Die Definition von Verkehrsklassen kann über die Klassifikationsregel frei gewählt werden. Da die Klassifizierung im User Space stattfindet, kann NeTraMet auf verschiedenen Betriebssystemen eingesetzt werden. Cisco NetFlow ist ein vergleichsweise statisches Meter. Die Attribute, mit der die Klassen unterschieden werden, sind fest vorgegeben. Ausserdem ist das Meter ausschließlich zusammen mit Cisco IOS verfügbar. Dafür kann bei der Klassifikation eine höhere Leistungsfähigkeit erzielt werden. Bezüglich zusätzlicher Funktionalität schneidet NetFlow im Vergleich zu NeTraMet eher schlecht ab. Eine Erklärung dafür ist mit Sicherheit die freie Verfügbarkeit des NeTraMet-Quellcodes, die eine Weiterentwicklung des Tools durch interessierte Entwickler ermöglicht. Dies trifft auch auf den Linux-Klassifizierer Netfilter zu. Da Netfilter jedoch - anders als NeTraMet und NetFlow - primär für Anwendungen wie Firewalling konzipiert wurde, sind hier bisher nur wenige Erweiterungen für Accounting und Messzwecke verfügbar.

Paket-Klassifikation

Accounting Meter benötigen im Kern einen Paket-Klassifizierer. Aufgrund der vielfältigen Einsatzgebiete für Paket-Klassifizierer stellt die Weiterentwicklung und Verbesserung von Algorithmen zur Paket-Klassifikation ein aktives Forschungsgebiet dar, bei dem in der Regel ein Kompromiss zwischen Klassifikationsgeschwindigkeit, Speicherbedarf und Flexibilität gefunden werden muss.
In [GuMc99] wird basierend auf der Analyse von derzeit verwendeten Filterregeln (für Routing, Firewalling und Bereitstellung von QoS-Diensten) ein einfacher heuristischer Algorithmus mit dem Namen Recursive Flow Classification (RFC) entworfen. Dieser nutzt die Struktur in den Filterregeln aus, um zusammenhängende Bereiche zu identifizieren und diese in sogenannte equivalent Class IDs zu gruppieren. Damit wird der Klassifikationsprozess auf Kosten eines etwas höheren Speicherbedarfs beschleunigt. Aufgrund der notwendigen Vorausberechnung bietet der Algorithmus jedoch nur eine sehr eingeschränkte Flexibilität bezüglich des Hinzufügen und Entfernens von Filterregeln. Außerdem basiert der Ansatz stark auf den in den untersuchten Beispiel-Filterregeln gefundenen Regel-Strukturen und ist daher nicht ohne weiteres auf beliebige Einsatzgebiete erweiterbar.
In [EnKa96] wird ein Dynamic Packet Filter (DPF) vorgestellt, der eine dynamische Code-Generation verwendet, um einen auf dem PathFinder-Filter [BaGP94] basierenden Paket-Filter zu optimieren. Der Filter-Code wird dabei direkt in Maschinensprache kompiliert. Dies ermöglicht zwar das schnelle Hinzufügen und Entfernen von Regeln, schafft aber auch eine Abhängigkeit vom verwendeten Betriebssystem. DPF ist ein Paket-Filter, eine Klassifizierung von Paketen ist daher nicht möglich.

Policy-basiertes Metering

Um eine verursachergerechte Tarifierung zu realisieren, müssen Informationen über die Ressourcenreservierung und –nutzung im Netz erfasst und gesammelt werden. Auch für die Überprüfung der im SLA vereinbarten Grenzwerte sollte eine messtechnische Erfassung der erbrachten Qualität erfolgen. Provider können sich durch unterschiedliche Dienstangebote und Tarifkonzepte sowie durch Bereitstellung spezieller Accounting-Dienste (z.B. detailierte Rechnungen, permanente Information über die momentane Ressourcennutzung, etc.) gegeneinander abgrenzen. Auf welche Weise die Ressourcennutzung erfasst werden soll, hängt unter anderem ab von der Größe und dem Zweck des Provider Netzes, den angebotenen Dienstklassen, den verwendeten Tarifierungsverfahren, den zu unterstützenden Accounting-Diensten sowie der Möglichkeit, vorhandene Infrastruktur (z.B. MIBs) nutzen zu können. Die Metering-Infrastruktur muss an die jeweilige Messaufgabe anpassbar sein und einen hohen Grad an Flexibilität bezüglich der Datenerfassung, -sammlung, -weiterleitung und -speicherung aufweisen. Ein Beispiel für die Ableitung der Messaufgabe aus dynamischen DiffServ-SLAs findet sich in [ShSt00].

Im folgenden stellen wir eine Architektur für policy-basiertes Metering vor. Die Architektur erlaubt eine Anpassung an unterschiedliche Messaufgaben für Accounting und SLA-Überprüfung. Die Verwendung von Meter-Policies ermöglicht zum einen den Austausch von Konfigurationsinformationen zwischen Providern. Zum anderen kann über Meter-Policies die Ansteuerung verschiedenartiger Meter-Typen in einer heterogenen Messumgebung realisiert werden.

Das SLA kann dabei als Basis für Informationen zu den benötigten Tarifvariablen und den vereinbarten Qualitätsparametern dienen. Diese bilden den Ausgangspunkt, um Policies für das Accounting und die QoS-Messungen zu definieren. Über AAA-Server können diese Informationen an andere Provider weitergeleitet werden. Über ein Application Specific Modul (ASM) wird dann das Mess-Equipment konfiguriert [CaZZ00].

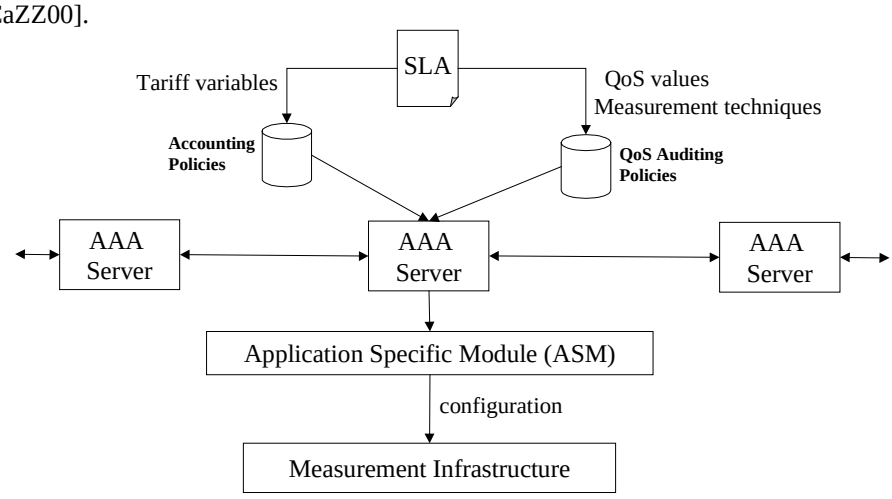

Abbildung 1: Konfiguration der Mess-Infrastruktur über Policies

Meter-Policies beschreiben die Regeln für die Erzeugung, den Transport und das Speichern der gemessenen Daten in einer standardisierten Form und können in die entsprechende Konfiguration der einzelnen Messelemente der Accounting-Infrastruktur umgewandelt werden. Meter-Policies erlauben die Konfiguration folgender Datenstrukturen und Parameter:

Meter-Records: Die im Meter-Record enthaltenen Messdaten (relevante Attribute) lassen sich aus dem verwendeten Tarifierungsverfahren bzw. der im SLA vereinbarten QoS-Überprüfung ableiten. Meter-Policies lassen sich einsetzen, um einem benachbarten Anbieter mitzuteilen, was dieser messen muss, z.B. im Fall von Roaming.

Meter-Record-Ziel: Das Ziel eines Meter-Records beschreibt, wohin ein Record gesendet werden soll. Das Ziel kann ein nachfolgender Billing-Prozeß, ein benachbarter Provider, ein Kunde (Hot Billing) oder eine Auditing-Datenbank sein.

Meter-Intervall: Das Meter-Intervall beschreibt, in welchen Abständen Meter Records generiert und an das Ziel gesendet werden.

Aufbewahrungszeit: Die Aufbewahrungszeit beschreibt, wie lange der Record (z.B. für Auditing-Zwecke) gespeichert werden muss.

Zugangskontroll-Liste: Diese Liste spezifiziert die Zugangsrechte zu den gemessen Daten, d.h. wer Lese- und/oder Schreibrechte auf die Daten hat.

Meter-Genauigkeit: Die Meter-Genauigkeit gibt an, mit welcher Genauigkeit die Daten erfasst bzw. gespeichert werden.

Meter-Granularität: Das Metering kann sich auf aggregierte Flows sowie auf einzelne Flows (spezifiziert durch Schicht-3-Informationen, sowie optional durch Informationen höherer Schichten) beziehen.

Überprüfung des SLAs durch QoS Messungen

Service Level Agreements (SLAs) beschreiben den Vertrag zwischen Nutzer und Provider. SLAs enthalten die vereinbarte Qualität, mit der ein Dienst vom Provider erbracht werden muss. Sie können darüber hinaus auch Vorschriften enthalten, wie die vereinbarte Qualitätsstufe kontrolliert wird. Auch genaue Angaben über die verwendeten Messverfahren und Reglungen über die Konsequenzen bei Nicht-Einhaltung der vorgegebenen Qualitätsgrenzen können Bestandteil des SLAs sein. [Verm99].

Zur Ermittlung der momentanen Qualität einer Verbindung sind permanente Messungen erforderlich. Bei der Verwendung von aktiven Messverfahren werden Test-Pakete auf die zu messende Strecke gesendet. Diese Testpakete führen zu unerwünschter Zusatzlast, welche die Messungen verfälschen kann. Bei Messungen über Providergrenzen hinweg ergibt sich eventuell zusätzlich das Problem, dass ein benachbarte Provider den Test-Verkehr nur ungern in das eigene Netz weiterleitet, insbesondere wenn besondere Paketformate verwendet werden. Passive Messverfahren besitzen den Vorteil, ohne zusätzlichen Verkehr die Qualität der Verbindung zu überprüfen.

Im Differentiated Services Model werden unidirektionale Datenströme betrachtet. Außerdem können sich Performance-Kenngrößen wie Paket-Verlustrate und Verzögerungszeit für die Hin- und Rückrichtung stark unterscheiden. Daher sollten bei der Ermittlung der Qualität einer Verbindung One-Way-Metriken gemessen

werden. Dies erfordert mindestens zwei Messpunkte im Netz. Es müssen also mindestens zwei Messpunkte entsprechend den im SLA spezifizierten Messanforderungen konfiguriert werden. Die im folgenden beschriebene Messarchitektur dient dazu, die Messanforderungen aus dem SLA abzuleiten und anhand einer Meter-Datenbank die Verteilung der Messaufgabe auf die unterschiedlichen Instanzen zu automatisieren.

Meter Architektur

Dieses Kapitel beschreibt die entwickelte Messarchitektur zur Erfassung der Accounting-Daten und zur Überprüfung der Dienstqualität. Abbildung 2 zeigt die einzelnen Komponenten der Architektur und deren Kommunikation untereinander.

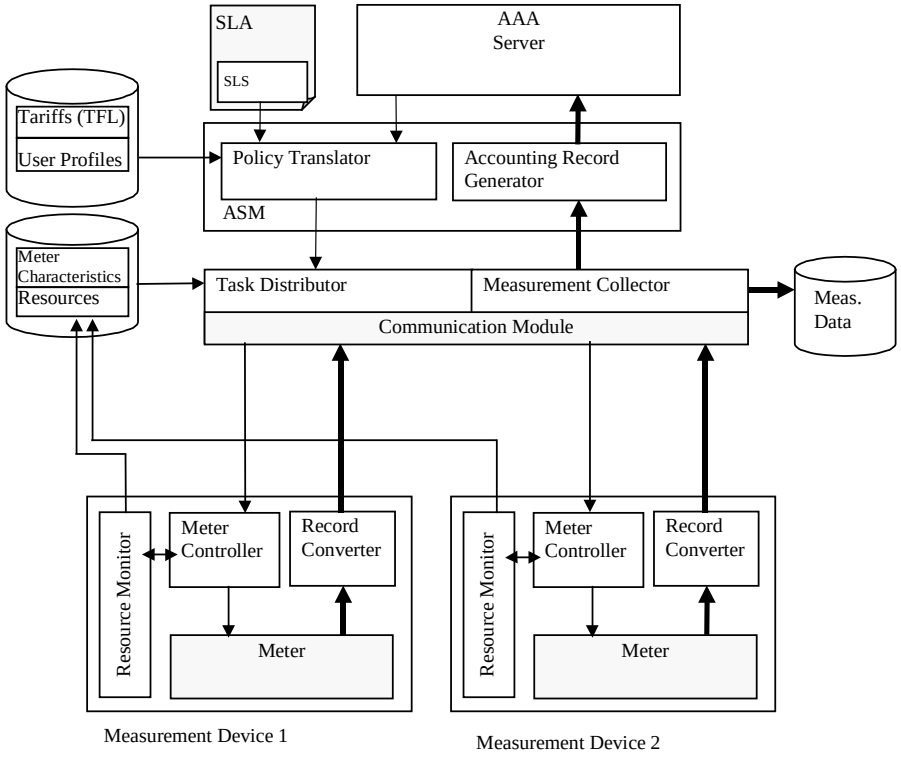

Abbildung 2: Meter-Architektur

Der Policy Translator generiert anhand der Service Level Specification (SLS) [BeSB00], des Tarifs und des Kunden Profils die entsprechende Meter Policy. Zur Beschreibung des Tarifs wird die Tarifbeschreibungssprache TFL (Tariff Formula Language, [CaHZ99]) eingesetzt. Soll das Metering für einen Kunden eines anderen Anbieters konfiguriert werden, erhält der Policy Translator Anweisungen über den AAA-Server. Der Task Distributor verteilt die Meter-Policies an die entsprechenden

Messgeräte. Er entscheidet über die Verteilung anhand der vorhandenen Meter und deren Position in bezug auf den Pfad des zu messenden Verkehrs, der Ressourcenauslastung der einzelnen Meter, sowie der Funktionalität der Meter. Auch die Messergebnisse vorangegangener Messungen können eine Änderung der Konfiguration veranlassen. So ist es z.B. möglich QoS-Werte zunächst mit einer relativ einfachen Messmethode nur grob zu erfassen (z.B. Erfassung der Round-Trip-Time statt des geforderten One-way-delays) und nur falls die ungenaueren Messwerte bestimmte Grenzwerte überschreiten, eine präziserer Messung (z.B. zur Bestimmung des One-way-delays) zu starten. Das Communication Module übernimmt die Verteilung der Meter-Policies an die einzelnen Meter und den Transport der gemessenen Werte von den Metern zum Collector. Der Meter Controller wertet die empfangenen Policies aus und wandelt sie in entsprechende Konfigurations-anweisungen für das jeweilige Meter sowie den Resource Monitor um. Der Resource Monitor misst die verfügbare Ressourcen auf dem Meter System (CPU-Auslastung, Speicher) und sendet diese in regelmäßigen Abständen an ein zentrales Repository und an den Meter Controller. Der Meter Controller ist in der Lage, lokal Maßnahmen bei einer kritischen Ressourcenauslastung zu ergreifen, z.B. lediglich priorisierte IP-Flows zu vermessen. Ermittelte Messwerte werden vom Meter Controller über das Record Converter Module an den Measurement Collector gesendet. Der Record Converter konvertiert die gemessenen Meter-spezifischen Records in ein standardisiertes Format. Der Measurement Collector speichert die empfangenen Werte in einem Repository. Wurden Daten für einen Kunden eines anderen Anbieters gemessen, werden im Accounting Record Generator Accounting Records generiert. Der AAA-Server sendet die Accounting Records dann an den AAA-Server des Kunden.

Umsetzung von SLA-Angaben in Meter Policies

Das folgende Beispiel verdeutlicht, wie SLA-Angaben in Meter-Policies umgesetzt werden. Die Meter-Informationen enthalten Ort, Typ und andere Kenngrößen der im Netz verfügbaren Meter. Die Nutzer-Informationen enthalten Informationen über den verwendeten Dienst und den anzuwendenden Tarif.

Diese Informationen werden vom Policy-Translator ausgewertet und daraus Meter-Policies generiert. Anhand der Flow-Angaben aus dem Kundenprofil und den benötigten Tarifvariablen aus der Tarif-Datei wird eine Klassifikationsregel für den Datenstrom des Nutzers generiert. Diese wird anhand der in den Meter Characteristics spezifizierten Adressen für Meter und Reader an die entsprechende Komponenten des Meter Systems übertragen. Auf diese Weise wird die neue Klassifikationsregel beim entsprechenden Meter eingefügt. Hierfür wurde eine einfache Meter-Policy-Sprache definiert, die das Setzen von Parametern sowie das Hinzufügen und Entfernen von Meter-Rules erlaubt. Da die Überprüfung der erbrachten Dienstleistung in der Regel Teil der Dienstleistungsvereinbarung ist, sollten SLAs präzise Angaben zur messtechnischen Ressourcenerfassung und Überprüfung der erbrachten Qualität enthalten. Mess-Parameter, die nicht durch das SLA vorgegeben sind, werden durch Voreinstellungen des Providers festgelegt.

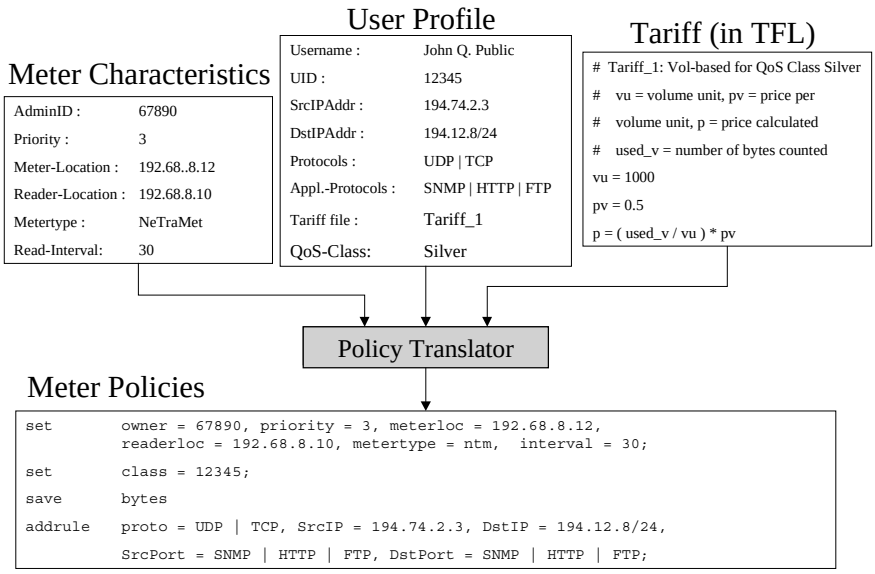

Abbildung 3: Umsetzung von SLA-Angaben (Kundenprofil und Tarif) in Meter-Policies

Konfiguration unterschiedlicher Meter-Typen

Die Benutzung von Meter-Policies ermöglicht die einheitliche Konfiguration unterschiedlicher Meter-Typen. Die Meter-Policies werden vom Meter Controller in die für den entsprechenden Meter passende Konfiguration umgewandelt. Abbildung 4 zeigt die policy-basierte Konfiguration von zwei unterschiedlichen Meter-Typen (NeTraMet und Cisco NetFlow) dar.

Das NeTraMet Meter wurde zusätzlich erweitert, um den Klassifizierer mit einem Filter im Kernel (BPF) kombinieren zu können [CATZ00]. Policies können hierbei dazu benutzt werden, das Kernel-Filter so zu konfigurieren, dass Pakete von IP-Flows, die nicht gemessen werden sollen, gar nicht erst an den Klassifizierer im User Space weitergereicht werden. Hierdurch kann eine Überlastung des Meters vermieden werden. Die Meter-Policies für die relevanten Flows und die zu messenden Attribute werden vom Meter Controller in die Meter-spezifische Konfiguration (NeTraMet ruleset) übersetzt. Außerdem wird der Meter Reader, der die gemessenen Daten per SNMP ausliest, entsprechend konfiguriert. Während beim NeTraMet Meter konfiguriert werden kann, welche Flows gemessen werden, liefert das NetFlow Meter grundsätzlich Messergebnisse für alle Flows. Nicht benötigte Messergebnisse müssen verworfen werden. Meter Policies können im Meter Controller in Filter-Anweisungen für den Flow Collector umgewandelt. Der Meter selbst ist hier statisch und deshalb nicht von der Konfiguration betroffen.

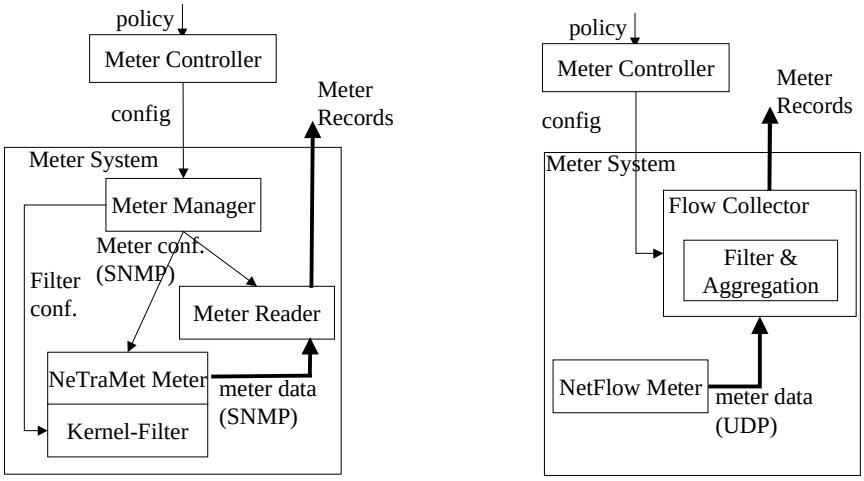

Abbildung 4: Konfiguration eines erweiterten, hierarchischen RTFM-Meters und eines erweiterten NetFlow-Meters

Verteilung der Messaufgaben und Sammeln der Messergebnisse

Zum Verteilen der Messaufgaben in Form von Meter-Policies und zum Sammeln der Messergebnisse in Form von Meter-Records wird eine Erweiterung des National Internet Measurement Infrastructure (NIMI) Systems verwendet [PaAM00]. NIMI stellt ein System zur Ansteuerung verschiedener Messtools dar. NIMI Probes stellen die Mess-Funktionen zur Verfügung, die über Kontroll-Instanzen konfiguriert und ausgelesen werden können. Die Ansteuerung der Messtools erfolgt zur Zeit über Skripte, die als NIMI-Wrapper dienen. NIMI unterstützt die Sammlung von Meter-Daten über das TCP-Protokoll und verfügt über Zugangskontrollmechanismen.

Durch Verteilung von Messaufgaben auf mehrere Meter lässt sich eine Überlastung einzelner Meter vermeiden. Wenn durch Aggregierung von IP-Flows sehr hohe Daten- und Paketraten anfallen, bei der die Leistung eines einzelnen Meters nicht ausreicht, so kann die gleiche Messaufgabe auf mehrere hierarchische Meter verteilt werden. Jedes beteiligte Meter muss hierbei nur noch einen Teil der IP-Flows vermessen, während die restlichen Flows jeweils durch ein Kernel-Filter entfernt werden können.

Messungen

Nachfolgend präsentieren wir Ergebnisse unserer Leistungsuntersuchungen unter Verwendung des NeTraMet-Meters. Wir demonstrieren den Einfluss unterschiedlicher Verkehrscharakteristiken auf die benötigten Meter-Ressourcen. Anschließend zeigen wir, wie bei unserer Erweiterung von NeTraMet die policy-basierte Konfiguration eines Kernel-Filters dazu genutzt werden kann, eine Überlastung des Meters zu vermeiden.

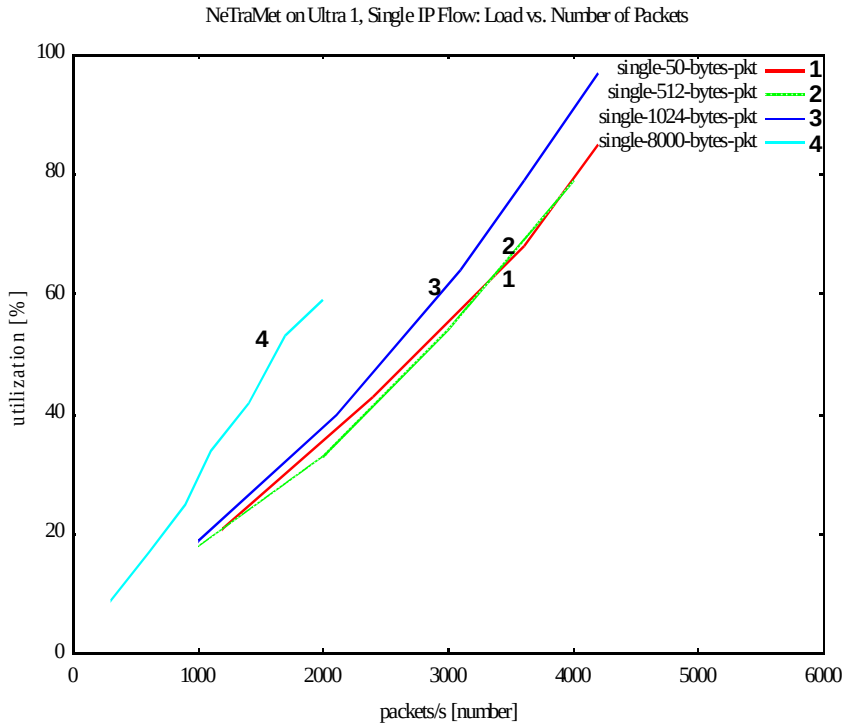

Abbildung 5: CPU-Last in Abhängigkeit der Paketrate für unterschiedliche Paketgrößen (NeTraMet)

Die erste Messung wurde auf einer SUN Ultra 1 unter Solaris 2.6 durchgeführt. Zur Messung der CPU-Last wurde das Programm `vmstat` eingesetzt. Abbildung 5 zeigt das Anwachsen der CPU-Last eines Meters (NeTraMet) bei wachsender Paketrate für vier unterschiedliche Paketgrößen. Es wird deutlich, dass beim betrachteten Szenario die Leistungsfähigkeit des Meters in erster Linie von der Paketrate abhängt. Die Paketgröße und damit auch die Datenrate hat demgegenüber einen deutlich geringeren Einfluss. Dieses Ergebnis entspricht den Erwartungen, da die Verarbeitungsschritte beim Metering sich überwiegend auf den Paketkopf beziehen, und weil die Anzahl der Kontextwechsel des Betriebssystems mit wachsender Paketrate zunimmt.

Die zweite Messung wurde auf einem PC unter FreeBSD (Filtering mit BPF im Kernel) durchgeführt. Abbildung 6 zeigt das Experiment, bei dem Accounting durch ein hierarchisches Meter (NeTraMet mit aktiviertem BPF-Kernel-Filter) durchgeführt wird. Variiert wird hierbei die Anzahl der Flows, die durch das Kernel-Filter herausgefiltert werden. Die Anzahl der vorhandenen IP-Flows ist konstant. Der NeTraMet-Prozess im User-Bereich muss hierbei nur die Flows verarbeiten, die durch den Kernel nicht herausgefiltert werden. Bei sinkender Anzahl herausgefilterter IP-Flows nimmt die Systemlast des Benutzerprozesses, und damit auch die gesamte CPU-Last zu. Das Experiment zeigt, dass bei geeigneter Konfiguration des Kernel-Filters sowie der Filter-Regeln im User-Bereich eine Überlast einzelner Meter abgewendet werden kann.

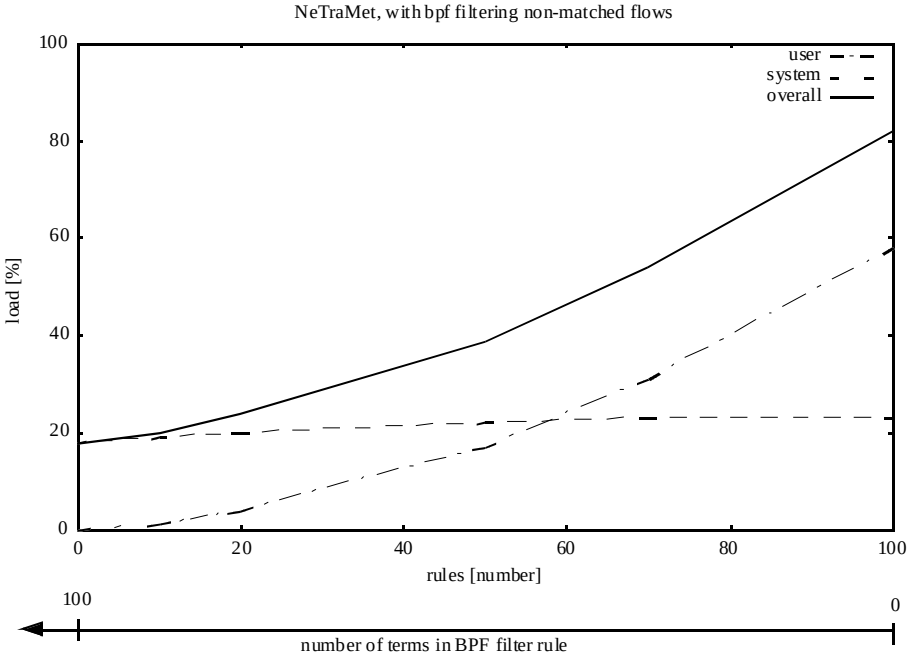

Abbildung 6: CPU-Last in Abhängigkeit der Filterregeln in Userspace und Kernel (NeTraMet mit BPF-Erweiterung)

Zusammenfassung und Ausblick

In IP-Netzen mit differenzierter Dienstqualität fallen vielfältige Messaufgaben an, um verursachergerechte Tarifierung, sowie eine Überprüfung von Dienstqualitätsparametern zu ermöglichen. Wir stellen für die hierzu erforderliche Messinfrastruktur eine Architektur für policy-basiertes Metering vor, die geeignet ist, einen Accounting-Dienst konform mit der generischen AAA-Architektur [Laat00] der IRTF-AAAARCH-Gruppe zu realisieren. Ein Policy-Translator setzt SLA-Angaben mit Kundenprofil und Tarifformel in Meter-Policies um, die von einem Task-Distributor an geeignete Meter weitergeleitet werden. Aus Ausdrücken der Meter-Policy-Sprache werden Meter-spezifische Konfigurationsinformationen generiert. In zukünftigen Arbeiten wird untersucht, inwieweit sich SLA Angaben in Abhängigkeit von den Eigenschaften der verfügbaren Meter in Meter Policies umsetzen lassen.

Die Verwendung von Meter-Policies erlaubt eine effiziente Verwaltung von Netzen mit heterogenen Messkomponenten.

Bei dem erweiterten, auf NeTraMet basierenden hierarchischen RTFM-Meter kann ein Kernel-Filter so konfiguriert werden, dass auch bei begrenzten Systemressourcen eine Überlastung des Meters vermieden werden kann. Messergebnisse für das NeTraMet-Meter zeigen, dass hohe Paket-Raten zu einer Überlastung des Meters führen können und wie sich durch geeignete Konfiguration eines Kernel-Filters die CPU-Last des Meters verringern lässt.

Ein flexibles Meters für Accounting und QoS-Messungen basierend auf dem Linux Klassifizierer Netfilter, das sich über Policies ansteuern läßt, ist zur Zeit in Entwicklung. Eine erste Vergleichs-Messung mit NeTraMet zeigte bereits eine deutlich bessere Leistungsfähigkeit des Linux-Klassifizierers.

Die vorgestellte Architektur für policy-basiertes Metering erweist sich somit als geeignet, um eine heterogene, flexible und leistungsfähige Messinfrastruktur in IP-Netzen bereitzustellen. Im Rahmen des von DFN-Verein und deutscher Telekom geförderten Projektes QUASAR wird die Eignung der policy-basierten IP-Meter-Architektur für das G-WIN untersucht werden.

Literaturangaben

[AbAH00] Bernard Aboba, Jari Arkko, David Harrington, “Introduction to Accounting Management”, <draft-ietf-aaa-acct-03.txt>, Work in Progress, Mai 2000

[BaGP94] Mary L. Bailey, Burra Gopal, Michael A. Pagels, Larry L. Peterson, Prasenjit Sarkar: “PATHFINDER. A Pattern-Based Packet Classifier”, In Proceedings of the 1rst Symposium on Operating System Design and Implementation, Monterey, California, November 1994.

[BeSB00] Y. Bernet, A. Smith, S. Blake, D. Grossman, “A Conceptual Model for Diffserv Routers”, draft-ietf-diffserv-model-03.txt, Work in Progress, Mai 2000

[BrBl00] Nevil Brownlee, Alan Blount, “Accounting Attributes and Record Formats”, Internet-Draft draft-ietf-aaa-accounting-attributes-03.txt, Work in Progress, April 2000

[CaHZ99] Georg Carle, Felix Hartanto, Michael Smirnow, Tanja Zseby, “Charging and Accounting for QoS-enhanced IP Multicast”, IFIP Sixth International Workshop on Protocols For High-Speed Networks (PfHSN '99), August 1999, Salem, MA, U.S.A.

[CaMM98] Massimiliano Canosa, Martino De Marco, Alessandro Maiocchi, "Traffic accounting mechanism for Internet Integrated Services", Technical Report, CEFRIEL, Politecnico di Milano, 1998

[CATZ00] Georg Carle, Jens Tiemann and Tanja Zseby, “Assessment of accounting Meters with Dynamic Traffic Generation based on Classification Rules”, The First Passive and Active Measurement Workshop (PAM2000), Hamilton, New Zealand, April 2000, S. 127-133.

[CaZZ00] G. Carle, S. Zander, T.Zseby, “Policy-based Accounting”, draft-irtf-aaaarch-pol-acct-00.txt, Work in Progress, Juni 2000

[Cisc99] NetFlow Services and Applications, White Paper, Cisco Systems, 1999

[EnKa96] D. Engler, M. F. Kaashoek, DPF: Dynamic Code Generation Packet Filter, SIGCOMM 96, S. 53-59.

[GuMc99] Pankaj Gupta and Nick McKeown, “Packet Classification on Multiple Fields”, Proc. ACM Special Interest Group on Data Communication - SIGCOMM 1999, Cambridge, Massachusetts, USA, September 1999

[Laat00] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, "Generic AAA Architecture", draft-irtf-aaaarch-generic-01.txt, Work in Progress, März 2000

[NARUS] NARUS Building the business of the Internet: Technical Note: “NARUS System”, http://www.narus.com

[Netf00] The Netfilter Project HomePage: http://www.samba.org/netfilter/

[PaAM00] Vern Paxson, Andrew K. Adams and Matt Mathis, "Experiences with NIMI", Passive & Active Measurement Workshop PAM 2000, Hamilton, New Zealand, April 2000

[RFC2063] N. Brownlee, C. Mills, G. Ruth, "Traffic Flow Measurement: Architecture", RFC2063, IETF, Januar 1997

[ShSt00] Syed Shaah and P. Steenkiste, "Cooperative Metering for Receiver Initiated Service Level Agreements", Proceedings of NOSSDAV 2000, Chapel Hill, North Carolina, U.S.A., Juni 2000

[Voll00a] John Vollbrecht, et al, "AAA Authorization Framework", draft-irtf-aaaarch-authorization-framework-00.txt, Work in Progress, Januar 2000

[Voll00b] John Vollbrecht et al., "AAA Authorization Application Examples", draft-irtf-aaaarch-authorization-apps-00.txt, Work in Progress, Januar 2000.