Ein Meßsystem zur Bewertung von Multicast-Protokollen im Internet[1]

Markus Hofmann, Jörn Hartroth
Institut für Telematik, Universität Karlsruhe, Zirkel 2, 76128 Karlsruhe

1 Motivation

Das Internet als weltweite Kommunikationsplattform hat in den vergangenen Jahren einen enormen Aufschwung erhalten. Es ermöglicht die Kommunikation und Kooperation zwischen Partnern, die weltweit verteilt operieren. Zugleich bildet es die Basis für zahlreiche Dienste, wie beispielsweise das World Wide Web (WWW), News oder E-Mail. In zunehmendem Maße finden auch Systeme zur Durchführung von Videokonferenzen oder zur verteilten Teamarbeit das Interesse der Anwender. Banken und Zeitschriftenverlage gehen dazu über, Informationen über das Internet an eine prinzipiell beliebig große Anzahl von Empfängern zu verteilen. Das hierbei zugrundeliegende Kommunikationsprinzip läßt sich meist auf den Basisfall der sogenannten *Multicast*-Kommunikation abbilden. Hierbei übermittelt ein Sender die Daten an mehrere Empfänger. Dies kann durch mehrfache Nutzung der herkömmlichen Punkt-zu-Punkt Kommunikation erfolgen, indem der Sender mehrere Kopien ein und desselben Datenpaketes an die jeweiligen Empfänger adressiert. Es ist offensichtlich, daß eine solche Realisierung des Multicast-Dienstes äußerst ineffizient ist und in größeren, globalen Gruppen kaum eingesetzt werden kann. Dem Entwurf und der Entwicklung spezieller Multicast-Protokolle kommt demnach eine entscheidende Rolle zu.

Mit der breiten Verfügbarkeit von IP-Multicast [1], einer Protokollerweiterung zur Unterstützung von Gruppenkommunikation im Internet, wird erstmals das effiziente Arbeiten in verteilten Teams ermöglicht. Das Internet Protocol (IP) stellt lediglich einen unzuverlässigen Übertragungsdienst bereit. Zahlreiche Anwendungen benötigen jedoch einen gewissen Grad an Zuverlässigkeit. So muß beispielsweise die Verteilung von Börsenkursen, Zeitungen oder Off-Line-Videos gesichert erfolgen, da zahlende Empfänger alle Daten fehlerfrei erhalten müssen. Um trotz des stetig andauernden Anstiegs der Benutzerzahlen im Internet einen zuverlässigen und dennoch leistungsfähigen Datenaustausch zwischen mehreren Anwendern zu ermöglichen, müssen neuartige Kommunikationsprotokolle entwickelt und bewertet werden [6]. Aufgrund der stetig voranschreitenden Globalisierung der Kommunikationssysteme rückt dabei die Skalierbarkeit hinsichtlich der Teilnehmerzahlen zunehmend in den Mittelpunkt des Interesses. Diesem Umstand muß auch bei der Beurteilung und Bewertung der verschiedenen Multicast-Protokolle Rechnung getragen werden. Die Durchführung von Messungen in räumlich begrenzten Netzwerken mit einigen wenigen Rechensystemen ist nicht mehr ausreichend. Um eine realistische Bewertung der einzelnen Protokolle und ihrem Verhalten in globalen Netzen vornehmen zu können, muß eine große Anzahl weltweit verteilter Rechensysteme in die Messungen einbezogen werden. Dies erfordert bei manueller Konfiguration der beteiligten Systeme einen erheblichen Aufwand an Koordination. So muß beispielsweise zu einem zuvor definierten Zeitpunkt von den jeweiligen Systemadministratoren die Software zur Durch-

[1] 9. ITG/GI-Fachtagung „Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen (MMB'97)", Freiberg, 17.–19. September, 1997.

führung einer Messung konfiguriert und gestartet werden. Dies setzt die vorherige manuelle Übermittlung aller Konfigurationsparameter an die Teilnehmer voraus. Während der Durchführung der Messung muß deren Status überwacht werden, um mögliche Fehlerfälle frühzeitig zu erkennen. Anschließend sind die Meßergebnisse der jeweiligen Systeme zur Auswertung an eine zentrale Stelle zu übergeben. Erschwert wird die Koordination durch die Tatsache, daß die beteiligten Rechensysteme oftmals in unterschiedlichen Zeitzonen liegen. Aufgrund mehrstündiger Zeitverschiebungen ist es unmöglich, daß alle Systemadministratoren zum Zeitpunkt der Messung vor Ort anwesend sind. Häufig wird deshalb dem Initiator einer Messung ein Gastzugang auf den beteiligten Rechensystemen eingeräumt. Zwar kann in diesem Fall die Konfiguration der am Test beteiligten Systeme und die Durchführung der Messungen zentral von einem Rechensystem aus erfolgen, jedoch muß sich der Initiator der Messung auf jedem einzelnen der beteiligten Testsysteme manuell anmelden und die notwendigen Aktionen nacheinander ausführen. Dies stellt bei den betrachteten Szenarien mit mehreren hundert Empfängern einen enormen Aufwand dar. Zudem ist zur Einrichtung eines Gastzuganges ein gewisser Vertrauensvorschuß in die Person des Initiators einer Messung notwendig, was die Suche nach teilnahmewilligen Meßpartnern erschwert. Um trotz der genannten Probleme die Durchführung der Messungen unter Beteiligung weltweit verteilter Partner zu ermöglichen, wurde ein Meßsystem zur teilautomatisierten Durchführung von Meßreihen entworfen und implementiert.

Gemäß der obigen Beschreibung wird im Rahmen dieses Beitrages folgende Notation verwendet. Unter einer *Messung* wird das Übertragen von Daten an eine Menge von Empfängern verstanden, wobei mehrere Werte zur Beurteilung der Leistungsfähigkeit des verwendeten Kommunikationsprotokolls gemessen und erfaßt werden. Das sendende und die empfangenden Rechensysteme werden hierbei als *Meßknoten* bezeichnet, die menschlichen Administratoren der Meßknoten als *Meßpartner*. Eine Folge unterschiedlicher Messungen ergibt eine *Meßreihe*. Ein *Meßszenario* beschreibt, welche realen Meßknoten an einer Messung beteiligt sind, welches Rechensystem die Rolle des Senders übernimmt und zu welchem Zeitpunkt die Datenübertragung gestartet werden soll. Die Software zur Koordination, Durchführung und Auswertung von Messungen wird als *Meßsystem* bezeichnet. Das in diesem Beitrag vorgestellte Meßsystem erlaubt die zentrale Definition von Meßszenarien und die automatisierte Kontrolle der an den Messungen beteiligten Rechensysteme. Es unterstützt den Anwender neben der Konfiguration und der Durchführung von Messungen auch bei der Auswertung der erhaltenen Meßergebnisse.

Im weiteren werden zunächst die Rahmenbedingungen und die Anforderungen an das entwikkelte Meßsystem beschrieben. Darauf aufbauend wird in Kapitel 3 die Architektur des Meßsystems und dessen Umsetzung in eine Implementierung vorgestellt. Kapitel 4 beendet den Artikel schließlich mit einer Zusammenfassung und einem Ausblick.

2 Anforderungen an das Meßsystem

Das Internet als globales Netzwerk ist gekennzeichnet durch die weltweite Verteilung seiner Teilnehmer. Diesem Umstand muß bei der Bewertung von Internet-Protokollen Rechnung getragen werden. So ist es zur realistischen Beurteilung unterschiedlicher Lösungsansätze nicht ausreichend, Messungen in einem lokalen Testnetz mit einer stark eingeschränkten Anzahl von Rechensystemen durchzuführen. Vielmehr sollten zur Bewertung von Multicast-Protokollen Szenarien betrachtet werden, bei denen eine große Anzahl weltweit verteilter Empfänger an

der Datenübertragung beteiligt ist. Dies setzt eine entsprechende Menge teilnahmewilliger Meßpartner voraus. Diese können nur dann gefunden werden, wenn für die Durchführung der Messungen kein allzu großer Verwaltungsaufwand auf Seiten der Meßpartner entsteht. Ebenso sollte die Bereitstellung eines Gastzuganges auf den jeweiligen Meßknoten keine unabdingbare Voraussetzung für die Teilnahme an den Messungen sein. Optimalerweise erhalten die Teilnehmer an einer Messung ein einfach zu installierendes Softwarepaket, welches einmalig gestartet wird und anschließend die notwendigen Aktionen gemäß eines zentral erstellten Testszenarios automatisch veranlaßt. Nur so ist eine entsprechend große Anzahl teilnahmewilliger Meßpartner zur Durchführung aussagekräftiger Messungen zu gewinnen. Es lassen sich demnach folgende Anforderungen an das Meßsystem identifizieren:

- *Autonome Durchführung von Messungen*: Das Meßsystem muß in der Lage sein, eine vorgegebene Meßreihe automatisch durchzuführen. Der Startzeitpunkt sollte frei wählbar sein und der Ablauf der Messungen soll durch eventuell auftretende Fehler möglichst nicht beeinträchtigt werden. Die bei den Messungen erhaltenen Resultate müssen für eine spätere Auswertung gespeichert und auftretende Fehler protokolliert werden. Zudem ist die Möglichkeit zur Statusabfrage einer laufenden Messung vorzusehen.
- *Fernbedienung des Meßsystems*: Da weltweit verteilte Messungen für praxisnahe Ergebnisse notwendig sind, müssen Teile des Meßsystems auf den entfernten Rechensystemen ablaufen. Dies sind beispielsweise Softwarekomponenten zum Senden bzw. zum Empfangen von Daten, als auch zur Erfassung von Meßwerten. Die Steuerung des verteilten Meßsystems sollte jedoch von einer zentralen Stelle aus erfolgen, um den Aufwand auf Seiten der Meßpartner möglichst gering zu halten.
- *Einfache Installation der verteilten Komponenten*: Die einzelnen Komponenten sollten sich möglichst einfach und ohne großen Aufwand bei den Meßpartnern installieren und aktivieren lassen. Idealerweise sollten danach auf Seiten der Meßpartner keine weiteren Wartungs- oder Konfigurationsaufgaben mehr anfallen. Dieser Anforderung kommt eine zentrale Rolle zu, da somit die Suche nach einer ausreichend großen Zahl von Meßpartnern erleichtert wird.
- *Einfache Bedienbarkeit*: Das Meßsystem soll die Festlegung und Konfiguration von Meßszenarien erleichtern. Hierzu eignet sich die Bereitstellung einer graphischen Benutzeroberfläche, in welche alle wesentlichen Aufgaben integriert werden. Insbesondere sollte der Benutzer bei der Auswertung der Meßergebnisse unterstützt werden. Wünschenswert ist auch die graphische Aufbereitung der Meßwerte und die Möglichkeit zur Ausgabe der Graphiken in einem gängigen Druckerformat (z.B. Encapsulated Postscript).

Weitere Anforderungen ergeben sich aus dem Systemumfeld. So sollte das Meßsystem auf den gängigen Betriebssystemen ablauffähig sein, weshalb beim Entwurf der Implementierungsarchitektur auf größtmögliche Portabilität zu achten ist. Dies impliziert, daß systemnahe Funktionen möglichst selten verwendet und in eigenständige Module gekapselt werden. Darüber hinaus müssen beim Entwurf die Übertragungseigenschaften des Internet berücksichtigt werden. Die Übertragungsqualität ist hier kaum vorhersagbar und unterliegt starken Schwankungen. Diese müssen vor allem bei der Koordination der Messungen berücksichtigt werden und dürfen keinen Einfluß auf den Ablauf einer Messung haben. Aus der weltweiten Streuung der Testpartner ergibt sich die Notwendigkeit zur zeitlichen Synchronisation der beteiligten Maschinen. Diese dient jedoch nicht der Ermittlung von Übertragungsverzögerungen, sondern vielmehr dem synchronisierten Start einer Messung. Aus diesem Grunde ist eine Genauigkeit im Sekundenbereich vollkommen ausreichend.

3 Die Architektur des Meßsystems

Gemäß den gestellten Anforderungen und den gegebenen Randbedingungen ergeben sich für das Meßsystem zwei unterschiedliche Aufgabenbereiche. Zum einen müssen die verschiedenen Meßszenarien an einem zentralen System definiert und an die beteiligten Meßknoten verteilt werden. Nach Abschluß der Messungen sind die auf den Meßknoten ermittelten Resultate zu erfassen und an einer zentralen Stelle statistisch auszuwerten. Zum anderen sind die eigentlichen Messungen unter Beteiligung der verteilten Meßknoten durchzuführen und die Ergebnisse zu speichern. Es lassen sich demnach eine zentrale Komponente des Meßsystems und mehrere verteilte Module zur Durchführung der eigentlichen Messung identifizieren. Diese Aufgabenteilung schlägt sich in einer zweigeteilten Architektur des Meßsystems nieder, die nach funktionalen Gesichtspunkten weiter unterteilt wurde (siehe **Abbildung 1**).

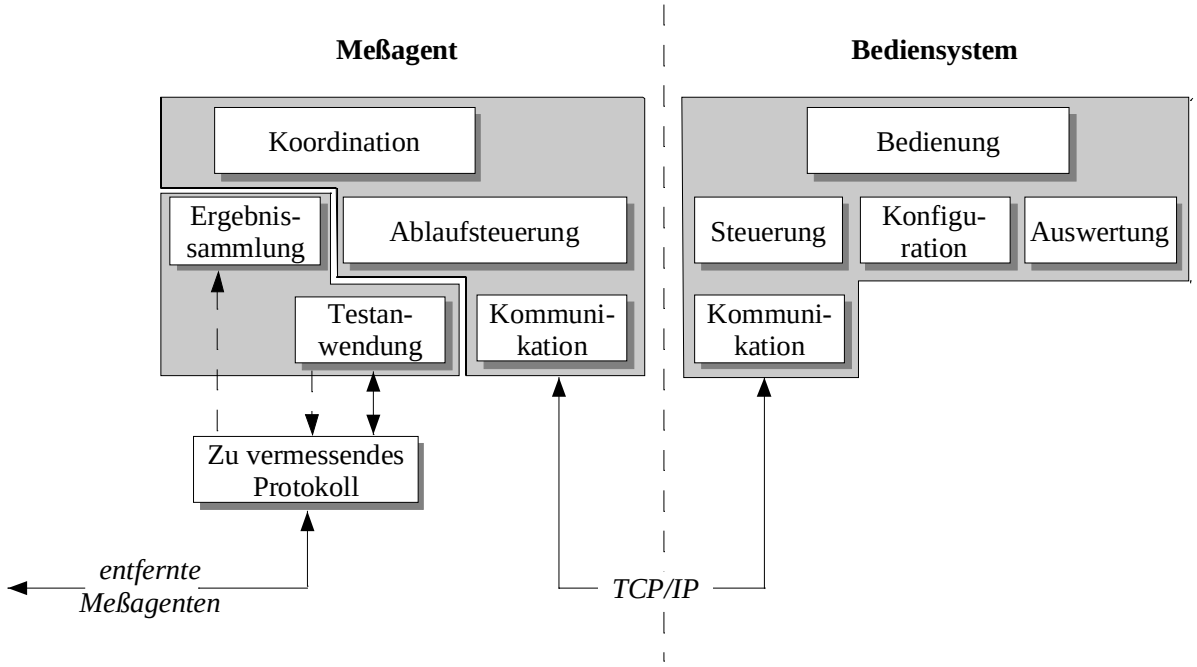

Abbildung 1 Architektur des Meßsystems

Das sogenannte Bediensystem stellt die zentrale Komponente des Meßsystems dar. Es erlaubt dem Benutzer das komfortable Erstellen von Meßszenarien. Dazu wird ein Konfigurationseditor mit graphischer Benutzeroberfläche zur Verfügung gestellt. Der Benutzer legt hiermit den Sender, die Empfängermenge und den Startzeitpunkt der Meßreihen fest. Darüber hinaus können weitere Parameter für den Datenaustausch, wie beispielsweise die Senderate oder die Paketgröße, definiert werden. Diese Angaben werden in einer Konfigurationsdatei abgelegt und vor Durchführung der Messungen automatisch an die beteiligten Meßknoten übermittelt. Das Bediensystem unterstützt den Anwender nach erfolgreicher Beendigung einer Messung bei der Auswertung und der Aufbereitung der Ergebnisse. Hierzu werden statistische Funktionen und Werkzeuge zur graphischen Darstellung der Meßwerte bereitgestellt. Eine weitere Aufgabe des Bediensystems umfaßt die Kontrolle aktiver Messungen. Diese müssen sowohl gestartet und gestoppt, als auch während der Ausführung überwacht werden. Ebenso kann der Benutzer zu jedem Zeitpunkt den Status einer aktiven Messung und den Zustand der daran beteiligten Meßknoten vom zentralen Bediensystem aus abfragen.

Die eigentliche Messung wird von den sogenannten Meßagenten durchgeführt. Diese werden auf den Meßknoten einmalig installiert. Ihre Aufgabe besteht darin, die am Bediensystem erstellten Meßszenarien entgegenzunehmen und die Messungen entsprechend der erhaltenen Konfiguration durchzuführen. Dazu veranlassen sie zu den angegebenen Zeitpunkten die entsprechenden Aktionen, wie beispielsweise das Senden oder das Empfangen von Daten unter Nutzung des zu vermessenden Protokolls. Während des Datentransfers werden periodisch die aktuellen Qualitätsparameter (z.B. Durchsatz, Fehlerrate, Verzögerung) gemessen und in einer Datei gespeichert. Nach Beendigung des Datentransfers werden die Ergebnisse an das Bediensystem übermittelt und stehen dort zur Auswertung bereit. Ebenso nehmen die Meßagenten während der Laufzeit einer Messung Anfragen und Kommandos des Bediensystems entgegen. So übermitteln sie beispielsweise auf Anfrage ihren Status oder brechen eine aktive Messung auf Befehl vorzeitig ab.

Im Überblick gestaltet sich der Ablauf einer Messung wie folgt. Zunächst wird die Software der Meßagenten an die Meßpartner verteilt und von diesen auf den von ihnen verwalteten Meßknoten installiert und aktiviert. Die Meßknoten sind nun in der Lage, Konfigurationsdateien vom Bediensystem zu empfangen. Am Bediensystem wird das Meßszenario erstellt und automatisch an die jeweiligen Meßknoten übermittelt. Diese werten die erhaltene Konfigurationsdatei aus und veranlassen zu den angegebenen Zeitpunkten die entsprechenden Aktionen. Nach erfolgreicher Beendigung der Messung werden die erhaltenen Ergebnisse an das Bediensystem übermittelt und dort mit Hilfe der bereitgestellten Werkzeuge ausgewertet. Nach der einmaligen Installation der Agenten-Software können die Messungen völlig unabhängig von den Meßpartnern von einer zentralen Stelle aus angestoßen und beliebig oft durchgeführt werden.

In den folgenden Abschnitten wird die überblickartig vorgestellte Architektur weiter verfeinert. Dazu werden die einzelnen Subkomponenten sowohl des Bediensystems als auch der Meßagenten näher betrachtet.

3.1 Das Bediensystem

Das Bediensystem besteht aus fünf Komponenten. Die *Konfigurationseinheit* dient der Erstellung von Meßszenarien. Diese werden mit Hilfe des Konfigurationseditors erstellt und in einer Konfigurationsdatei abgelegt. Der Konfigurationseditor stellt dem Anwender eine graphische Benutzeroberfläche zur Verfügung (siehe **Abbildung 2**) und führt zugleich eine Plausibilitätsüberprüfung der definierten Meßszenarien durch.

Die gespeicherte Konfigurationsdatei muß im Anschluß an die beteiligten Meßknoten übermittelt werden. Diese Aufgabe übernimmt die *Steuerungseinheit*. Sie stellt Funktionen zum Übermitteln von Meßszenarien an die beteiligten Meßknoten, zum Abfragen des Status einer Messung und zum Einsammeln der Ergebnisse bereit. Eine weitere Aufgabe der Steuerungseinheit ist die Synchronisation der beteiligten Systeme.

Zur Übertragung von Daten zwischen Bediensystem und Meßknoten bedient sich die Steuerungseinheit der Funktionen der *Kommunikationseinheit*. Diese ermöglicht einen zuverlässigen Datenaustausch zwischen den verschiedenen Systemen. Die Kommunikationseinheit ist sowohl im Bediensystem als auch in den Meßagenten vorhanden und nutzt zur gesicherten Übertragung der Meßszenarien und von Statusnachrichten das Transmission Control Protocol (TCP).

Die *Auswertungseinheit* übernimmt die Auswertung, Aufbereitung und die graphische Darstellung der erhaltenen Meßergebnisse. Diese können zur weiteren Verarbeitung in einer Textdatei zusammengefaßt und gespeichert werden.

Die *Bedieneinheit* realisiert die Schnittstelle zum Benutzer hin. Sie integriert die Konfigurations-, Auswertungs- und Steuerungseinheit in einer universellen Benutzeroberfläche (siehe Abbildung 2) und ermöglicht dem Benutzer eine einfache und komfortable Bedienung des gesamten Meßsystems.

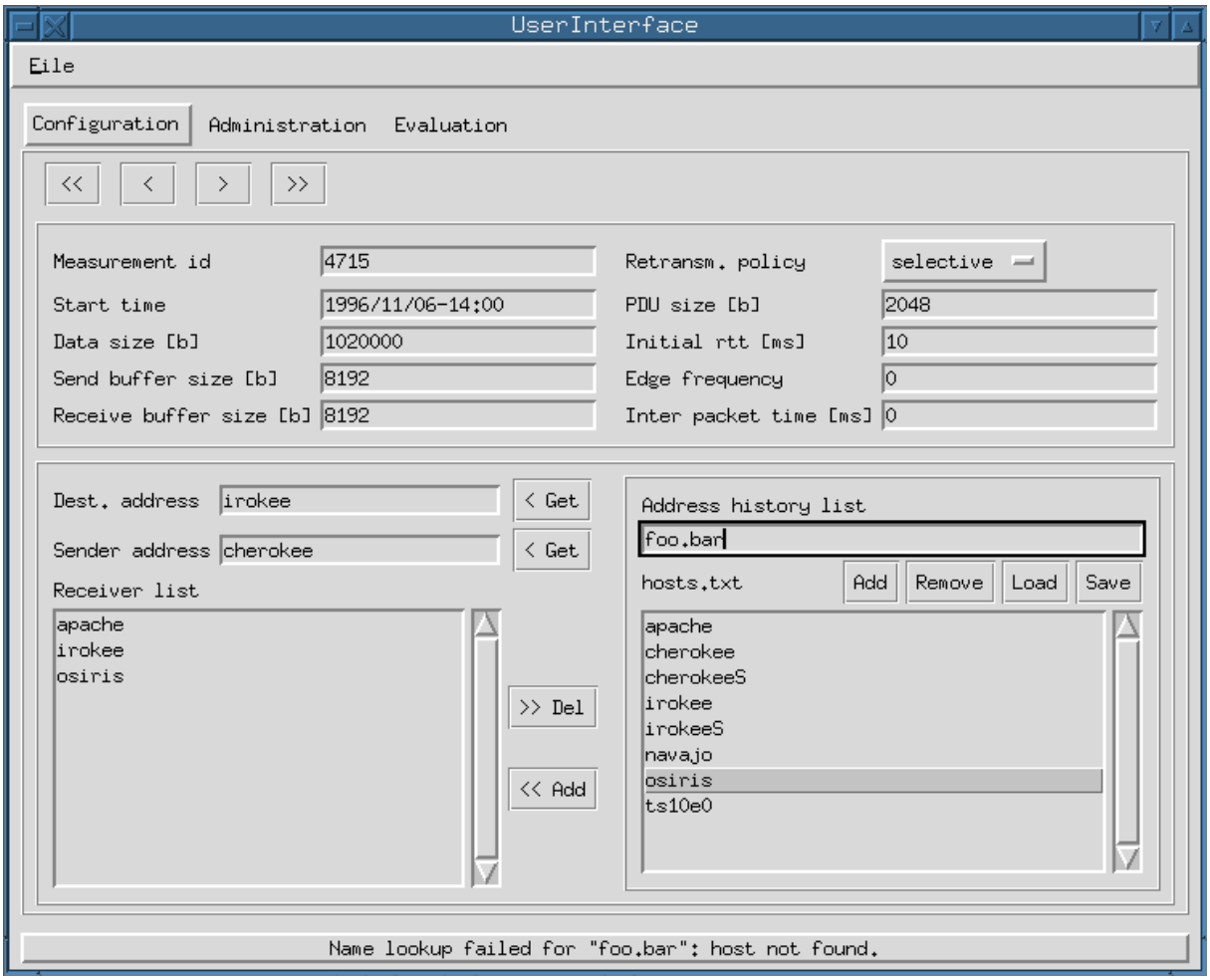

Abbildung 2: Graphische Benutzeroberfläche des Konfigurationseditors

Das Bediensystem als zentrale Komponente des Meßsystems ist lediglich auf einem einzelnen Rechner zu installieren. Aus diesem Grunde entfällt für das Bediensystem die Forderung nach einer einfachen Installation und nach eingeschränkten Anforderungen an die Systemumgebung. Jedoch sollte eine hohe Portabilität gewahrt bleiben, um das Bediensystem auf beliebigen Plattformen einsetzen zu können. Aus diesem Grunde wurde das gesamte Bediensystem in *Python* [3] erstellt. In Python erstellte Programme sind auf allen Systemen unverändert ablauffähig, für die eine Portierung des Python-Interpreters existiert. Dies ist u.a. sowohl für die gängigen UNIX-Systeme als auch für Windows 95, Windows NT und OS/2 der Fall. Python bietet standardmäßig mächtige Module (z.B. für den Zugriff auf Netzwerk-Sockets oder zur Handhabung von Strings) und komplexe Datentypen (z.B. Tupel und Listen). Eine besondere Stärke von Python stellt die Unterstützung für den Netzwerkzugriff dar. Zur Programmierung

graphischer Benutzeroberflächen wird eine objektorientierte Schnittstelle zu dem weit verbreiteten Tk [12] bereitgestellt. Der Vorteil von Python/TK gegenüber Tcl/Tk [12] ist die Unterstützung objektorientierter Programmiermethoden. Dies stellt insbesondere bei der Erstellung graphischer Benutzerschnittstellen einen großen Vorteil dar, was ursprünglich auch die Wahl von Python gegenüber Java begründete. Derzeit wird jedoch an einer Realisierung des Bediensystems in Java gearbeitet.

3.2 Der Meßagent

Im Gegensatz zum Bediensystem, welches nur auf einem einzelnen, zentralen Rechensystem abläuft, agieren die Meßagenten auf mehreren Rechnern im Internet. Sie müssen auf allen Meßknoten installiert und konfiguriert werden. Der Meßagent besteht aus fünf Teilkomponenten. Wie in Abbildung 1 dargestellt, werden diese bei der Implementierung auf zwei Prozesse verteilt.

Der Prozeß `measure` umfaßt die Testanwendung und die Ergebnissammlung. In ihm sind die funktional zusammengehörenden Aufgaben des Einlesens eines Meßszenarios, dem Starten der dazugehörigen Messung und der Erfassung von Meßwerten zusammengefaßt. `measure` wird entsprechend dem Inhalt einer Konfigurationsdatei automatisch zum angegebenen Zeitpunkt durch die Steuerungseinheit gestartet. Im Einzelnen gliedert sich der Programmablauf dieses Prozesses wie folgt:

1. *Auswerten der Kommandozeile*: Die Kommandozeile wird eingelesen und ausgewertet. Optionale Aufrufparameter sind beispielsweise die Namen der Konfigurations- und der Ergebnisdatei.
2. *Einlesen des Meßszenarios*: Die Konfigurationsdatei wird eingelesen und auf syntaktische Korrektheit überprüft. Dazu wurde unter Nutzung der GNU-Programme `flex` [10] und `bison` [2] ein Parser erstellt.
3. *Initialisieren der Messung*: Die Messung wird initialisiert, d.h. die internen Datenstrukturen werden angelegt und eine gegebenenfalls notwendige Registrierung wird beim Daemon des zu vermessenden Protokolls vorgenommen.
4. *Durchführen der Messung*: Es wird auf den angegebenen Startzeitpunkt gewartet und danach die Messung aktiviert.
5. *Übermitteln der gespeicherten Ergebnisse*: Nach erfolgreicher Beendigung einer Messung werden die erhaltenen Ergebnisse in einer Datei abgelegt und diese an das Bediensystem übermittelt.

Die Koordinations-, Steuerungs- und Kommunikationseinheit sind im Prozeß `measured` zusammengefaßt, welcher auf jedem Meßknoten im Hintergrund abläuft. Die Kommunikation zwischen Bediensystem und Meßknoten wird über die *Kommunikationseinheit* abgewickelt. Das Bediensystem übermittelt Kommandos an die Meßknoten, diese führen das Kommando aus und senden das Resultat zurück an das Bediensystem. Die Nachrichten werden hierbei über eine zuverlässige TCP-Verbindung gesendet. Aufgabe der *Ablaufsteuerung* ist die Koordination und Überwachung einer Messung. Sie stellt hierzu Funktionen zum Starten und zum Stoppen sowie zur Abfrage des Status einer Messung bereit. Als Bindeglied zwischen Kommunikationssystem und Ablaufsteuerung agiert die *Koordinationseinheit*. Sie bildet den Programmrahmen das Daemons `measured`, wertet eintreffende Kommandos aus und veran-

laßt die entsprechenden Aktionen. Zudem ist sie für die Initialisierung des Daemons und seiner Teilkomponenten verantwortlich.

Einer zentralen Bedeutung beim Zusammenspiel der einzelnen Komponenten des Meßsystems kommt der Konfigurations- und der Ergebnisdatei zu. Diese werden sowohl vom Bediensystem als auch von den Meßagenten ausgewertet und interpretiert. Ihre Syntax muß demnach global eindeutig festgeschrieben sein. Zugleich gibt deren Umfang die Möglichkeiten zur Durchführung von Messungen vor. So können beispielsweise nur solche Meßszenarien definiert werden, die allein mit der Syntax der Konfigurationsdatei zu beschreiben sind. Auch können nur diejenigen Parameter durch eine Messung evaluiert werden, die in der Syntax der Ergebnisdatei berücksichtigt wurden. Da die zu bewertenden Kommunikationsprotokolle häufig erweitert und zusätzliche Funktionalität oder Qualitätsparameter aufgenommen werden, müssen beide Dateiformate möglichst flexibel und leicht erweiterbar gehalten werden. Die beiden folgenden Abschnitte befassen sich aus diesem Grunde detailliert mit beiden Dateiformaten.

3.3 Die Konfigurationsdatei

Die vollständige Beschreibung eines Meßszenarios wird in der *Konfigurationsdatei* abgelegt. Sie enthält neben einer global eindeutigen Kennung des beschriebenen Meßszenarios alle Parameter, die zur Durchführung einer Messung notwendig sind. Insbesondere umfaßt dies auch die Definition des Senders und der Empfängermenge, also der Zusammensetzung der Multicast-Gruppe und die Anordnung ihrer Mitglieder. Neuartige Ansätze zur Realisierung skalierbarer Multicast-Dienste unterscheiden jedoch nicht mehr ausschließlich zwischen Sender und Empfängern. Vielmehr definieren diese neuartige Verwaltungssysteme zur hierarchischen Anordnung der Gruppenmitglieder in einer baumartigen Struktur [4], [5], [7], [13]. Um auch solche Multicast-Algorithmen mit Hilfe des Meßsystems bewerten zu können, wurde bei der Festlegung des Formats der Konfigurationsdatei bereits die Möglichkeit zur Angabe solcher Verwaltungssysteme (Gruppenverwalter) vorgesehen. Ebenso kann in der Konfigurationsdatei angegeben werden, ob eine Gruppenhierarchie statisch durch entsprechende Angaben definiert oder dynamisch zur Laufzeit durch das zu vermessende Protokoll etabliert wird.

Die Konfigurationsdatei ist zeilenweise aufgebaut und wird im Textformat gespeichert, was eine leichte Erweiterbarkeit garantiert. Jede Zeile wird mit einem Schlüsselwort eingeleitet, auf welches ein oder mehrere Parameter folgen. Kommentare werden mit einem '#' eingeleitet. Folgende Schlüsselwörter wurden bisher definiert:

- ID - Global eindeutige Kennung einer Messung, die vom Benutzer frei wählbar ist
- START - Startzeit der Messung im Format 'Jahr/Monat/Tag-Stunden:Minuten'
- DATASIZE - Gesamtmenge der zu übertragenden Daten
- PDUSIZE - Größe der zu sendenden Dienstdateneinheiten
- SNDBUFSZ - Größe des von der Testanwendung zu verwendenden Sendepuffers in Byte
- RCVBUFSZ - Größe des zu verwendenden Empfangspuffers in Byte
- INTERVAL - Zeitlicher Abstand zwischen dem Senden zweier aufeinanderfolgender Datenpakete in Millisekunden
- RETRANS - Das vom Protokoll zu verwendende Fehlerbehebungsverfahren; definiert sind bisher Go-Back-N (GOBACKN) und selektive Übertragungswiederholung (SELECTIVE)

- RTT - Die vom Protokoll zu verwendende initiale Paketumlaufzeit in Millisekunden
- DESTINATION - Die Zieladresse für die Datenübertragung (Multicast-Adresse), optional kann zusätzlich eine durch Doppelpunkt getrennte Port-Nummer angegeben werden
- SENDER bzw. RECEIVER - Der Sender bzw. die an der Messung beteiligten Empfänger; diese Schlüsselwörter besitzen einen oder mehrere durch Komma getrennte Parameter, welche nach dem Schema <*Schlüssel*> = <*Wert*> aufgebaut sind; die definierten Parameter sind in **Tabelle 1** aufgeführt

Schlüsselwort	***Optional***	***Default***	***Beschreibung***
ADDRESS	Nein	-	Die IP-Adresse des Endsystems
STATIC	Ja	0	STATIC = 1 ⇒ Statische Festlegung der Gruppenhierarchie; wird dieser Parameter auf Null gesetzt und somit ein dynamischer Aufbau der Gruppenhierarchie zur Laufzeit gewählt, so werden die folgenden Schlüsselwerte ignoriert
MASTER	Ja	Sender	Der übergeordnete Gruppenverwalter des Endsystems
LGC	Nein	0	LGC = 1 ⇒ Endsystem agiert als Gruppenverwalter

Tabelle 1 Schlüsselwörter der Konfigurationsdatei

Die Menge der definierten Schlüsselwörter und Parameter kann leicht ergänzt werden, um protokollspezifische Besonderheiten zu erfassen. **Abbildung 3** enthält eine beispielhafte Konfigurationsdatei.

```
#
# Test setup
#
ID              42                              # unique id
START           1997/12/01-14:17                # start time

DATASIZE        10240000                        # total number of bytes
PDUSIZE         1024                            # size of a single packet
SNDBUFSIZE      8196                            # send buffer
RCVBUFSIZE      32768                           # size of receive buffer
INTERVAL        0                               # time between two packets

SENDER          ADDRESS = 129.13.42.125, LGC = 1
DESTINATION     233.0.0.42
RECEIVER        ADDRESS = 129.13.35.77, MASTER = 198.43.5.134
RECEIVER        ADDRESS = 198.43.5.134, LGC = 1

RETRANS         SELECTIVE                       # retransmission policy
RTT             10                              # initial round trip time
# END
```

Abbildung 3 Beispiel für den Aufbau einer Konfigurationsdatei

3.4 Die Ergebnisdatei

Jeder Meßknoten erfaßt während der Durchführung einer Messung leistungsspezifische Werte und legt diese in der sogenannten *Ergebnisdatei* ab. Diese enthält darüber hinaus eine Kennung des jeweiligen Meßknotens und der jeweiligen Messung. Die Werte werden im Textformat abgelegt. Dies erleichtert die Implementierung der Lese- und Schreibroutinen. Auf den Einsatz eines Binärformates wurde bewußt verzichtet, um eine schnelle Kontrolle der Ergebnisse zu ermöglichen und eventuelle Erweiterungen am Dateiformat zu erleichtern.

Die Zeilen einer Ergebnisdatei sind zu drei logischen Bereichen gruppiert, die unmittelbar aufeinander folgen:

Datei $\rightarrow$ *Teil-1 Teil-2 Teil-3*

Der erste Teil enthält Ergebniswerte, die sowohl von einem sendenden als auch von einem empfangenden Meßknoten erfaßt werden. Dies sind beispielsweise die Kennung der Messung, die Multicast-Adresse oder auch die Anzahl der übertragenen Bytes und der gesendeten bzw. empfangenen Pakete. Der zweite Teil enthält verbindungsinterne Informationen, wie etwa die durchschnittliche Paketumlaufzeit und deren Varianz und die zur Datenübertragung benötigte Prozeßrechenzeit. Im dritten Teil finden sich schließlich sender- und empfängerspezifische Ergebnisse. Dies sind beispielsweise die Verbindungsaufbau- oder die Übertragungszeit.

3.5 Synchronisation von Bediensystem und Meßagenten

Um eine Messung durchführen zu können, müssen die beteiligten Meßknoten zu einem definierten Zeitpunkt bestimmte Aktionen ausführen. Die Empfänger müssen beispielsweise der angegebenen Multicast-Gruppe beitreten, der Sender hingegen muß mit der Übertragung von Daten beginnen. Die Angabe eines absoluten Startzeitpunktes ist hierbei nicht ausreichend. Zum einen können die Systemuhren der Meßknoten und des Bediensystems auf Grund von Ungenauigkeiten voneinander abweichen (Gangunterschied), zum anderen können die beteiligten Rechensysteme in unterschiedlichen Zeitzonen liegen. Zum gleichzeitigen Start einer Messung auf den Meßknoten muß demnach ein Abgleich der lokalen Systemzeiten erfolgen.

Die Synchronisation von Systemuhren in verteilten Systemen kann unter Verwendung des Network Time Protocol (NTP) [9] erfolgen. Hierzu muß jedoch auf jedem der beteiligten Systeme ein NTP Daemon installiert und konfiguriert werden. Dies widerspricht jedoch der Forderung nach einfacher Installation und Handhabung des Meßsystems. Die Notwendigkeit zur Installation der NTP Software würde den Kreis potentieller Meßpartner weiter einschränken. Zudem ist die von NTP erbrachte Genauigkeit nicht notwendig. Eine Synchronisation im Sekundenbereich ist für die annähernd zeitgleiche Aktivierung der Meßknoten ausreichend.

Der vorgesehene Mechanismus zur Uhrensynchronisation wird im folgenden an Hand eines Beispiels erläutert: Ein Bediensystem möchte die Meßuhr eines Meßknotens derart synchronisieren, so daß anschließend die Uhr des Meßknotens um maximal *n* Sekunden von der Uhr des Bediensystems abweicht.

Um dies zu erreichen, sendet das Bediensystem zum Zeitpunkt T_1 ein Paket an das Meßsystem, in welches die lokale Systemzeit $S_{1,B}$ des Bediensystems zum Zeitpunkt T_1 eingetragen wird. Die lokale Systemzeit des Meßknotens zum Zeitpunkt T_1 sei $S_{1,M}$. Der Gangunterschied beider Uhren beträgt also $\Delta G = S_{1,B} - S_{1,M}$.

Das Meßsystem erhält dieses Paket zum Zeitpunkt T_2 und sendet es sofort an das Bediensystem zurück. Zudem berechnet das Meßsystem die Differenz zwischen seiner lokalen Uhrzeit $S_{2,M}$ bei Empfang des Paketes und dem Zeitstempel $S_{1,B}$ aus dem Paket. Diese Differenz, zusammengesetzt aus der Übertragungsverzögerung (T_2 - T_1) und dem Gangunterschied der Uhren vom Bediensystem und dem Meßknoten ($\Delta G = S_{1,B} - S_{1,M}$), wird im Meßknoten als Offset gespeichert. Der vom Meßsystem gespeicherte Offset weicht demnach um genau $T_2 - T_1$ vom eigentlichen Gangunterschied der Uhren ab.

Das Bediensystem erhält das zurückgesendete Paket zum Zeitpunkt T_3, was der lokalen Systemzeit $S_{3,B}$ entspricht. Die gesamte Umlaufzeit des Paketes berechnet sich also zu $S_{3,B} - S_{1,B.}$ Da $T_1 < T_2 < T_3$ ist, kann das Bediensystem davon ausgehen, daß der vom Meßknoten berechnete Offset um höchstens $T_3 - T_1 = S_{3,B} - S_{1,B}$ ($\geq T_2 - T_1$) vom realen Gangunterschied der Uhren abweicht. In der Praxis wird diese obere Schranke noch deutlich unterboten, da die echte Abweichung aufgrund annähernd symmetrischer Verzögerungszeiten etwa bei $(T_3 - T_1)/2$ liegt.

Um die eingangs geforderte Genauigkeit von maximal n Sekunden Abweichung zu gewährleisten, wiederholt das Bediensystem diesen Vorgang so lange, bis die gemessene Umlaufzeit $T_3 - T_1$ (obere Abschätzung der Ungenauigkeit) kleiner als die vorgegebene Schranke von n Sekunden ist. Dabei muß die maximale Anzahl von Wiederholungen begrenzt werden, um eine Terminierung des Algorithmus zu gewährleisten. Da globale Paketumlaufzeiten im Internet nur äußerst selten wenige Sekunden übersteigen, kann mit diesem Verfahren eine Synchronisation der Meßknoten im Sekundenbereich erreicht werden. Dies ist für den benötigten Zweck vollkommen ausreichend.

4 Zusammenfassung und Ausblick

Das in diesem Artikel vorgestellte Meßsystem erlaubt eine weitgehend automatisierte und zentral gesteuerte Durchführung von Messungen zur Bewertung von Multicast-Protokollen im Internet. Durch seine einfache Handhabung und die hohe Portabilität eignet es sich für den Einsatz in großen, globalen Kommunikationsgruppen. Das Meßsystem wurde bisher zur Bewertung des Xpress Transport Protocol [11] und zur Messung von Paketverlustraten im Multicast Backbone (MBone) [8] eingesetzt. Weitere Arbeiten an dem Projekt umfassen die Realisierung der Bedieneinheit in Java und die Anpassung der Testanwendung an zusätzliche Multicast-Protokolle.

5 Danksagung

Der Dank gilt vor allem Herrn Markus Schöpflin, der maßgeblich am Entwurf des Meßsystems beteiligt war und dieses implementiert hat. Ebenso sei Herrn Manfred Rohrmüller gedankt, mit dessen Hilfe ein Großteil des Meßsystems nach Java portiert wird. Außerordentlicher Dank gebührt Herrn Prof. Dr. Dr. Gerhard Krüger, der das gesamte Projekt zeitlich und organisatorisch ermöglicht hat.

6 Literatur

[1] S. Deering, D. Cheriton: Multicast Routing in Datagram Internetworks And Extended LANs, ACM Transactions on Computer Systems, 8(2):85-110, Mai 1990.

[2] C. Donnelly, R. Stallmann: Bison, The YACC-compatible Parser Generator, for Bison version 1.25, Free Software Foundation, Inc., November 1995.

[3] T. Himstedt: Objektbeschwörung, iX, Seite 144 - 153, Oktober 1996.

[4] M. Hofmann: A Generic Concept for Large-Scale Multicast, In: B. Plattner (Hrsg.), Broadband Communications, Proceedings of International Zurich Seminar on Digital Communications (IZS'96), LNCS, No. 1044, Springer Verlag, Februar 1996.

[5] M. Hofmann: Enabling Group Communication in Global Networks, Proceedings of Global Networking'97, Calgary, Alberta, Kanada, Juni 1997.

[6] M. Hofmann: Scalable Multicast Communication in the Internet, ConneXions, Vol. 10, No. 10, Oktober 1996.

[7] H. Holbrook, S. Singhal, D. Cheriton: Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation, Computer Communication Review, Vol. 25, No. 4, Proceedings of ACM SIGCOMM'95, August 1995.

[8] V. Kumar: MBone - Interactive Multimedia on the Internet, New Riders Publishing, 1995.

[9] D. L. Mills: Network Time Protocol (Version 3), Specification, Implementation and Analysis, RFC 1305, März 1992.

[10] V. Paxson: Flex, A fast scanner generator, Edition 2.5, for flex version 2.5, Free Software Foundation, Inc., März 1995.

[11] T. Strayer: Xpress Transport Protocol, Revision 4.0, XTP Forum, Santa Barbara, März 1995.

[12] B. Welch: Practical Programming in Tcl and Tk, Prentice Hall, 1995.

[13] R. Yavatkar, J. Griffioen, M. Sudan: A Reliable Dissemination Protocol for Interactive Collaborative Applications, Proceedings of ACM Multimedia'96, 1996.