

Konzeption und Implementierung eines Systems zur Überprüfung der Durchlasswahrscheinlichkeiten und Verbindungs-QoS des ATM-Dienstes im G-WiN

**Kompetenzzentrum für ATM-Komponenten
- DFN-KomAK -**

Oktober 2000



**GMD - Forschungszentrum Informationstechnik GmbH
Forschungsinstitut für offene Kommunikationssysteme
GMD FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin**

Oktober 2000

Lutz Mark, GMD FOKUS



Inhaltsverzeichnis

1	Motivation	5
2	Aufgabenbeschreibung	6
3	Architektur	7
	3.1 Verteilung der Aufgaben.....	7
	3.2 Komponenten des Testsystems	8
4	Technik / Implementierung.....	9
	4.1 Synchronisation.....	9
	4.2 ATM-Testinterface.....	9
	4.3 Speicherung und Präsentation der Messergebnisse	10
5	Beschreibung des Prototyps	12
	5.1 Konfiguration	12
	5.2 Abfrage der Statistiken.....	12
6	Anhang.....	13
	Annex A: Testzellen	13
	Annex B: Konfiguration des Prototyps im KomAK-Labor	14
	Annex C: Beispielstatistik.....	15



1 Motivation

Der ATM-Dienst im G-WiN des DFN-Vereins wird auf Basis des T-Net-ATM der Deutschen Telekom realisiert. Die Telekom verspricht spezielle Werte für die Durchlasswahrscheinlichkeiten von ATM-SVCs sowie für die Dienstgüte von ATM SVC- und PVC-Verbindungen. Der DFN-Verein hat jedoch keine Möglichkeiten, die von der Telekom angegebenen Werte zu überprüfen. Die Telekom bietet keine Statistiken bzw. Werkzeuge zur Ermittlung der Werte an.

Der DFN-Verein hat ein Interesse daran, seinen Kunden einen zuverlässigen ATM-Dienst anzubieten. Hierfür ist es notwendig, Informationen über den Zustand der Netzinfrastruktur abrufen zu können. Sollten Einschränkungen in der Qualität der ATM-Dienste auffallen, kann der DFN-Verein seine Nutzer über den Grund der Schwierigkeiten informieren und rechtzeitig bei der Telekom auf Besserung drängen bzw. zusätzlich eine Minderung der Gebühren verlangen.

2 Aufgabenbeschreibung

Die Qualität des ATM-Dienstes erfährt der Anwender zum einen durch eine hohe Durchlasswahrscheinlichkeit sowie eine geringe Verzögerung von ATM-SVC-Verbindungsaufbauwünschen. Darüberhinaus ist es wichtig, dass auf den geschalteten Verbindungen eine dem Verkehrsvertrag entsprechende Dienstgüte zur Verfügung steht. Diese Werte können zu unterschiedlichen Zeiten und zwischen unterschiedlichen Standorten variieren. Daraus folgt zwangsläufig, dass die Messungen zwischen allen DFN-Knoten mit ATM-Zugang durchgeführt werden müssen und dass die Messungen in regelmäßigen über den Tag verteilten Intervallen erfolgen müssen.

Folgende Werte sollen gemessen werden:

1. Messung von Durchlasswahrscheinlichkeiten zwischen den Standorten
2. Messung der Verzögerung zum Aufbau von ATM-SVC-Verbindungen
3. Messung der QoS-Parameter auf den geschalteten ATM-Verbindungen

Die erhaltenen Messergebnisse müssen auf geeignete Weise gespeichert werden und jederzeit abrufbar sein.

Anforderungen an das Messsystem

Da die Messungen zwischen verschiedenen Standorten durchgeführt werden sollen, muss das Messsystem auf die jeweiligen Standorte verteilt werden. Die Steuerung und Überwachung der Messungen sowie die Sammlung und Speicherung der Ergebnisse kann hingegen an einer zentralen Stelle erfolgen. Der Abruf der Messergebnisse sollte nach Möglichkeit von beliebiger Stelle aus erfolgen können.

Zur Bestimmung von Übertragungsverzögerungen zwischen verteilten Testservern ist es erforderlich, die lokalen Uhren der beteiligten Messsysteme zu synchronisieren.



3 Architektur

Das Messsystem soll einfach zu administrieren sein, daher sollen die Konfigurationen an zentraler Stelle vorgenommen werden können. Jeder DFN-Knoten mit Zugang zum T-Net-ATM wird mit einem ATM-Testserver ausgestattet. Die eigentlichen Tests werden zwischen den verteilten Testservern durchgeführt. Es wird davon ausgegangen, dass die Kommunikation zwischen der Kontrollinstanz und den Testservern über das IP-Backbone des G-WiNs erfolgen kann. So hat der Kontrollverkehr keinen Einfluss auf das zu testende ATM-Netz.

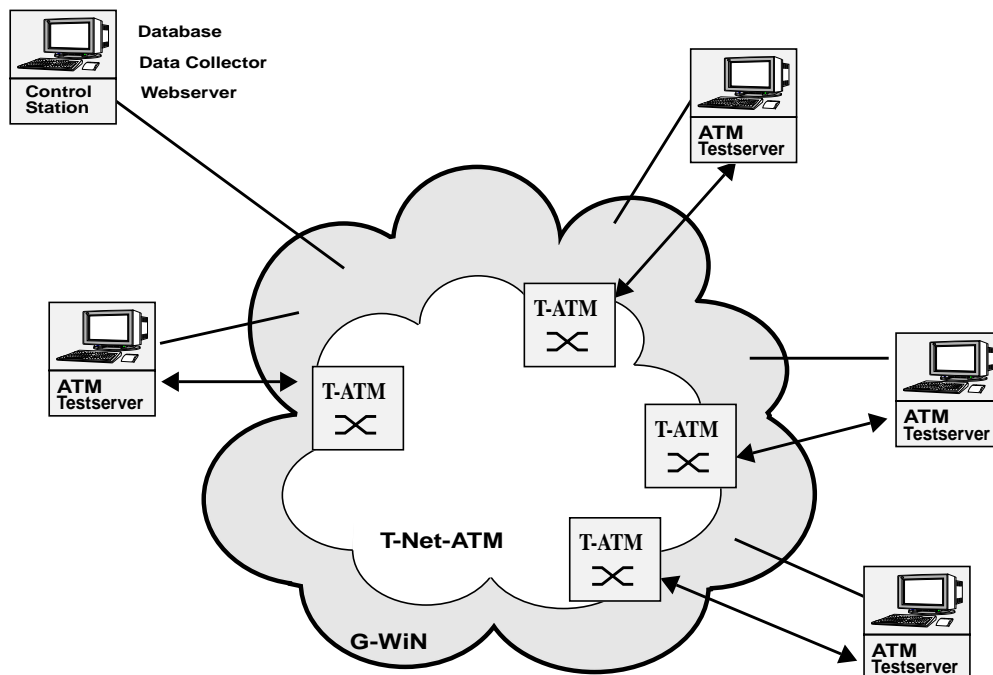


Abbildung 1: Aufbau des verteilten Testsystems

3.1 Verteilung der Aufgaben

Zum Erledigen der Messaufgaben eröffnen sich drei Alternativen:

1. Starten der Messapplikationen auf den verteilten Testkomponenten und Abholen der Ergebnisse durch einen zentralen Steuerungsprozess.

pro: Leicht zu implementieren und zu warten.

contra: Hohe Netzwerklast, da für jede durchzuführende Messaufgabe eine Verbindung zwischen der Kontrollinstanz und den entsprechenden Testkomponenten aufgebaut werden muss. Hohe CPU-Last auf dem Steuerungs-

rechner, Störungen/Unterbrechungen der Konnectivität zwischen Steuerungsprozess und den Testkomponenten beeinträchtigt die Messungen.

2. Starten der Messapplikationen auf den verteilten Testkomponenten durch einen zentralen Steuerungsprozess. Lieferung der Ergebnisse zur DB asynchron durch die Testkomponenten.

pro: Steuerungsrechner wird entlastet

contra: Hohe Netzwerklast

3. Die Testkomponenten arbeiten 'autonom'. Ihre Konfiguration kann durch eine zentrale Stelle erfolgen. Die Messergebnisse werden temporär auf den Testkomponenten gespeichert, bis sie von einem 'zentralen Datensammler' abgefragt werden.

pro: Steuerungsrechner wird entlastet, Netzwerk wird lediglich zur Konfiguration und bei der Abfrage der Messergebnisse belastet.

contra: Die Implementierung der Testkomponenten wird aufwendiger.

Die Idee eines zentralen Prozesses zur Steuerung der verteilten Messungen hätte eine unverhältnismäßige Netzlast über das IP-Backbone zur Folge. Außerdem skaliert diese Lösung nicht mit einer steigenden Zahl von Messaufgaben. Daher kam bei der Implementierung des Testsystems nur die dritte Alternative mit den autonom agierenden Testkomponenten in Frage.

Die Konfiguration wird an zentraler Stelle vorgenommen und auf die jeweiligen Testkomponenten 'geladen'. Die Testkomponenten führen die Messungen zu definierten Zeitpunkten durch und speichern die ermittelten Werte lokal. Ein zentraler Kollektor-Prozess erfragt die Messergebnisse von den Testkomponenten und speichert die Daten in der Messdatenbank.

Die Aufbereitung der Daten erfolgt jeweils mit den aktuellen Daten der DB durch den Webserver.

3.2 Komponenten des Testsystems

Die oben beschriebene Architektur besteht aus folgenden Komponenten:

1. **Testserver**: Sender- und Empfänger-Prozesse zum Aufbau der ATM-SVC-Verbindungen, Lastgeneratoren zum Generieren von Testverkehr zu definierten Zeitpunkten und mit fest definierten Verkehrsprofilen.
2. **Kontrollprozess** zur Konfiguration der Testserver. Außerdem kann dieser Prozess das System überwachen.
3. **Kollektorprozess** zum Einsammeln der Messergebnisse
4. **Datenbank** zur effizienten Speicherung der Messergebnisse
5. Ein **Webserver** zur Aufbereitung und Anzeige Messergebnisse



4 Technik / Implementierung

In diesem Kapitel wird auf Schlüsselfragen, die bei der Realisierung des Testsystems eine Rolle spielen eingegangen. Außerdem wird die Implementierung vorgestellt.

Alle Komponenten des Testsystems wurden auf Maschinen von SUN mit Solaris UNIX Betriebssystem implementiert. Diese Kombination bietet die Voraussetzung für eine stabile Laufzeitumgebung und ermöglicht darüberhinaus eine problemlose Fernwartung der Systeme. Eine hohe Zuverlässigkeit ist ein wichtiger Designaspekt des Testsystems.

4.1 Synchronisation

Wie oben bereits motiviert wurde, müssen die verteilt installierten Testkomponenten miteinander synchronisiert werden. Dies geschieht durch die Synchronisation der lokalen Uhren der Testkomponenten mit einem Timeserver. Alternativ lassen sich die Testkomponenten auch via spezieller Hardware direkt mit der Weltzeit synchronisieren.

Die Synchronisation der beteiligten Rechner geschieht über ihren G-WIN-IP-Anschluss und das Network-Time-Protokoll (ntp) bzw. über lokale GPS- oder DCF77-Empfänger.

4.2 ATM-Testinterface

Zur Durchführung der Tests wird ein Zugang zum ATM-Netz über ein ATM-Netzwerkinterface benötigt. Theoretisch lassen sich hierfür Standard-Netzwerkkarten einsetzen. Diese sind zwar relativ preiswert zu haben, jedoch sind mit diesen Karten keine genauen Messungen möglich.

Eine Alternative wäre, spezielle ATM-Tester anzuschaffen. Diese bieten Hardwareunterstützung zum Generieren von genauen Zeitstempeln für empfangene ATM-Zellen und sind in der Lage definierte Verkehrsströme exakt auszusenden. Leider sind ATM-Tester sehr teuer und in der Programmierung recht unflexibel. Da der geplante Testaufbau vorsieht, jeden DFN-Knoten mit ATM-Zugang mit einem Testsystem auszustatten, würde der finanzielle Aufwand für diese Systeme unverhältnismäßig groß werden.

Einen Ausweg aus diesem Dilemma findet sich durch die Verwendung spezieller ATM-Testinterfaces (TANYA) mittels derer sich Standard-Workstations zu ATM-Testern aufrüsten lassen. Die TANYA-Karten wurden bei GMD FOKUS zum Durchführung von ATM-Konformitäts- und Leistungstests entwickelt.

Innerhalb dieses Projekts wurden die Funktionen dieser Karte erweitert, um den Ansprüchen des verteilten Testsystems zu entsprechen.

Die Testserver müssen in der Lage sein, ATM-SVC-Verbindungen aufzubauen bzw. eingehende ATM-SVC-Verbindungsaufbauwünsche entgegenzunehmen. Ferner müssen auf den geschalteten Verbindungen Testzellströme generiert werden.

ATM Testzellen

Das Testsystem misst Zellenverlustrate und Zellenübertragungsverzögerungen. Um diese Werte ermitteln zu können, müssen Zellströme spezieller ATM-Testzellen generiert werden. Die Testzellen beinhalten eine Sequenznummer, einen Zeitstempel und eine CRC-Prüfsumme. Das genaue Format der benutzten ATM-Testzellen wird im Anhang beschrieben.

Im Zeitstempel-Feld der Testzellen wird der Sende-Zeitpunkt eingetragen. Im Anschluss daran wird über die Testzellen eine Checksumme gebildet. Geschieht dieser Vorgang 'in Software', so kommt es zu einer Ungenauigkeit, da zwischen dem Eintragen des Zeitstempels und dem Aussenden der Zelle am Netzwerkinterface eine unbestimmte Zeit zum Bilden der Checksumme und zum Übertragen der Zelle zum Interface verstreicht. Eine Alternative wäre es das Einfügen des Zeitstempels und die CRC-Berechnung durch das Testinterface 'in Hardware' zu realisieren.

Das in diesem Projekt entwickelte Testsystem geht einen dritten Weg. Und zwar ist es mit TANYA möglich, die Testzellen präzise zu frei definierbaren Zeitpunkten auszusenden. So werden die Testzellen via Software generiert (mit einem Zeitstempel, der leicht in der Zukunft liegt) und dann an die TANYA-Hardware übermittelt. Diese verschickt die Testzellen dann zu genau den auch in der Payload der ATM-Testzellen definierten Zeitpunkten.

4.3 Speicherung und Präsentation der Messergebnisse

Zur effektiven Speicherung der Messergebnisse ist eine Datenbank erforderlich. Die Messergebnisse werden vom Kollektor-Prozess gesammelt und in die Datenbank eingetragen. Das Abfragen der Messergebnisse sowie die Aufbereitung erfolgt durch den Webserver durch Aufruf eines Skriptes oder CGI-Programms.

Der erste Ansatz war die Verwendung der frei verfügbaren Datenbank *MySQL* sowie der Skriptsprache PHP zur Präsentation der Ergebnisse auf dem Webserver. Die *MySQL*-Datenbank ist sehr mächtig und die Geschwindigkeit der Zugriffe äußerst schnell. Auch PHP erwies sich als eine ideale Sprache zum Generieren von dynamischen Webseiten. Dennoch wurde diese Variante schnell verworfen, da in kurzer Zeit deutlich wurde, dass die Anzahl der neu zu



entwickelnden Skripte den in diesem Projekt zur Verfügung stehenden Rahmen sprengen würde.

Zum Monitoren von Netzwerken und zur Darstellung der Messergebnisse wird bei GMD FOKUS das frei verfügbare *Cricket* (<http://cricket.sourceforge.net>) erfolgreich eingesetzt. Es wurde zwar ursprünglich zum Abfragen von Routern entwickelt, lässt sich jedoch auch für andere Aufgaben einsetzen. Cricket besteht aus zwei Komponenten: Einem *Collector* und einem *Grapher* und passt daher ideal in die im vorigen Kapitel entworfenen verteilten Testarchitektur. Zum Speichern der Messergebnisse verwendet Cricket die Round-Robin-Datenbank *RRD* (<http://www.rrdtool.org>).

Daher verwendet der implementierte Prototyp des Testsystems den Collector und Grapher von Cricket und RRD. Als Webserver dient ein Apache (<http://www.apache.org>) in der aktuellsten Version.

5 Beschreibung des Prototyps

Dieser Abschnitt beschreibt den Implementierten Prototyp. Und zwar welche Tests sich durchführen lassen, wie diese konfiguriert werden können und wie sich die Messergebnisse abfragen lassen.

5.1 Konfiguration

Die Konfiguration der Tests muss an zwei Stellen vorgenommen werden. Zum Einen müssen die entsprechenden Testserver konfiguriert werden, um die gewünschten Tests zu Starten. Zum Anderen muss dem Kollektor-Prozess mitgeteilt werden, dass neue Messergebnisse zu erfragen sind.

Die Konfiguration des Testserver erfolgt über ein Telnet-Interface. Über einfache Kommandos lässt sich die Tests konfigurieren, starten bzw. beenden. Außerdem sind Statusabfragen möglich.

Der Kollektor basiert auf Cricket und wird über einen sogenannten Config-Tree konfiguriert. Der Aufbau der Konfigurationsdatei ist relativ einfach: Jedem unterschiedlichem Test ist ein Verzeichnis zugeordnet. In diesem befindet sich eine Default-Datei, in welcher konfiguriert ist, mit welchem Kommando die Messergebnisse zu erfragen sind und wie diese graphisch dargestellt werden sollen. In einer zweiten Datei sind die betreffenden Testserver aufgeführt sowie eventuelle Parameter.

Derzeit nicht implementiert ist ein Web-Interface, welches es dem Benutzer ermöglicht beide Konfigurationen in einem Schritt durchzuführen.

5.2 Abfrage der Statistiken

Die Ergebnisse lassen sich mittels Internet-Browser über eine Web-Seite abrufen. Bei Bedarf lässt sich der Zugriff auf die Seite über ein Passwort schützen. Die Startseite bietet eine Liste der unterschiedlichen Statistiken (Konnektivität oder Dienstgüte) zur Auswahl. Hat der Benutzer eine Alternative ausgewählt, erscheint eine neue Seite mit den konfigurierten Tests. Also z.B. Konnektivität zwischen Berlin und Darmstadt, zwischen Erlangen und Hannover usw. Entscheidet sich der Benutzer nun für einen für einen der Tests, so wird die Seite mit den Statistiken generiert. Es werden standardmäßig eine Statistik über die letzten 36 Stunden bzw. über die letzten 10 Tage präsentiert. Darüberhinaus lassen sich Statistiken über den letzten Monat bzw. das letzte Jahr generieren und anzeigen.



6 Anhang

Annex A: Testzellen

Zur Bestimmung der Performance-Parameter ist es notwendig, ATM-Zellströme spezieller Testsequenzen auszusenden.

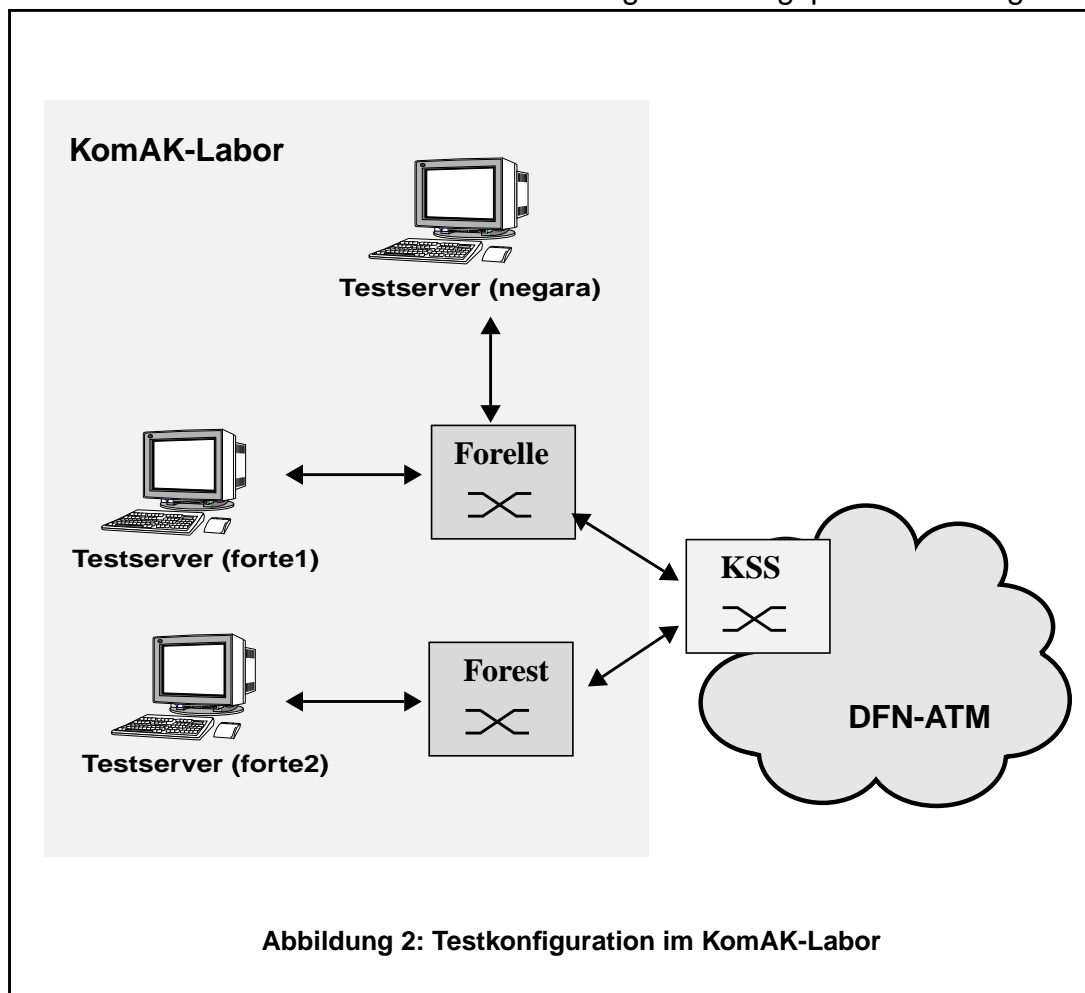
Die ATM-Testzellen haben den in der folgenden Tabelle definierten Aufbau. Der im Projekt KomAK implementierte Prototyp des Testsystems benutzt die Elemente Sequenznummer und Sequenznummer zur Bestimmung der Zellverlustrate sowie den Zeitstempel zur Messung der Übertragungsverzögerung.

Bytes	Beschreibung
1 (1)	Version
2-4 (3)	Sequence number (24 Bit)
5-12 (8)	Timestamp (64 Bit)
13 (1)	Original CLP-Bit
14-16 (3)	Sequence length
17-19 (3)	No. of cells with CLP=0
20-22 (3)	No. of previous cells in the sequence with CLP=0
23-44 (22)	Unused
45-48 (4)	CRC-32 Checksum

Annex B: Konfiguration des Prototyps im KomAK-Labor

Dieser Abschnitt beschreibt den Aufbau des Testnetzwerks im KomAK-Labor sowie die Konfiguration des innerhalb von KomAK implementierten Prototyps des Testsystems zur Überprüfung von Durchlasswahrscheinlichkeiten und Dienstgüte.

Das KomAK-Labor hat zwei Anschlüsse an den ATM-SVC-Piloten des B-WiNs. Diese sind mit zwei ATM-Switches (Forelle und Forest) im KomAK-Labor verbunden. Ferner wurden drei Testserver (Negara, Forte1 und Forte2) installiert. Die Testserver sind mit TANYA-Testinterfaces ausgestattet. Negara und Forte1 sind jeweils über Multimode-Fibre mit einem OC3-Port der Forelle verbunden (Abbildung 2). Die Forte2 ist an den ATM-Switch Forest angeschlossen. Auf den Ports der ATM-Switche ist UNI3.1 als Signalisierungsprotokoll konfiguriert.



Außerdem sind am Tests zwei weitere Workstations beteiligt, auf denen der Kollektorprozess bzw. der Webserver laufen.

Die Konnektivitäts- und Dienstgütetests werden durch die Testserver auf der Forte1 und Forte2 durchgeführt. Als Ziel der Tests dient in beiden Fällen die Negara.



Annex C: Beispielstatistik

Dieser Abschnitt zeigt am Beispiel der Messung des Call-Setup-Delays, also der Zeit die zum Aufbau einer SVC-Verbindung zwischen zwei Testservern über das ATM-Netz benötigt wird, wie sich die Messwerte auf dem Webserver als Statistiken darstellen lassen.

In der Graphik aus Abbildung 3 wird das gemessene Call Setup Delay in einer Tagesstatistik präsentiert.

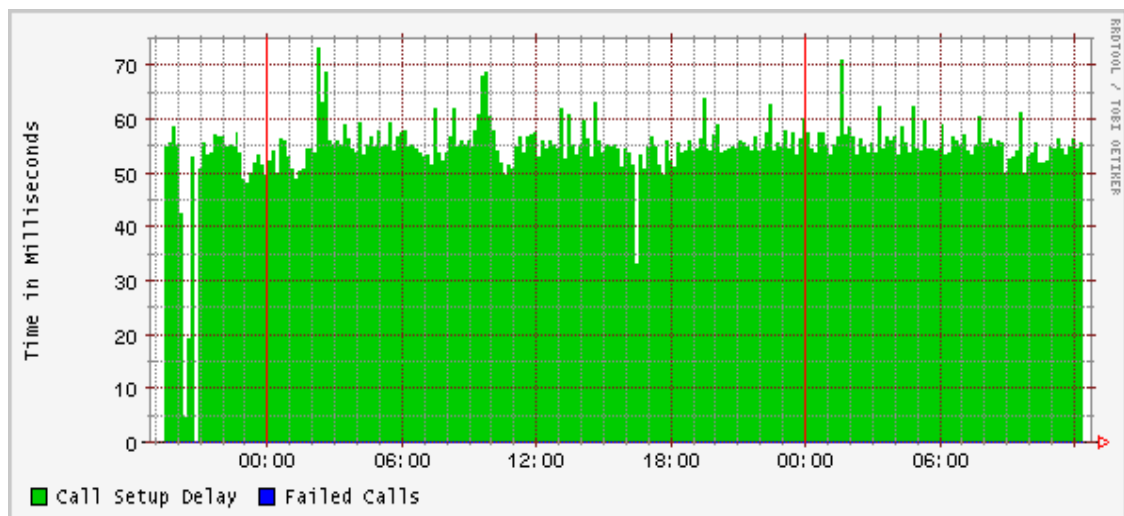


Abbildung 3: Call Setup Delay Tagesstatistik

