Reduktion von Verzögerungsunterschieden bei der Gruppenkommunikation im Internet

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart zur Erlangung der
Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Jens-Uwe Klöcking

aus Erfurt

Hauptberichter:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Mitberichter:	Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
Tag der mündlichen Prüfung:	24. Juli 2006

Institut für Parallele und Verteilte Systeme (IPVS)
der Universität Stuttgart
2007

Danksagung

Mein Dank gilt all denen, die mich während der Anfertigung der Dissertation gefördert haben und ohne deren Mitwirken die Arbeit nicht zustande gekommen wäre.

Allen voran danke ich der Deutschen Forschungsgemeinschaft für die Möglichkeit, an der Universität Stuttgart im Rahmen des Graduiertenkollegs Parallele und Verteilte Systeme unter Leitung von Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn arbeiten zu dürfen.

Ganz besonderen Dank möchte ich meinem Doktorvater, Herrn Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel, für seine wertvollen Anregungen und die wohlwollend kritische Begleitung meiner Arbeit aussprechen, die den erfolgreichen Abschluss der Dissertation erst möglich machte.

Herr Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn war Zweitbetreuer dieser Arbeit und Leiter der jährlich stattfindenden Berichtskolloquien, die ein Podium für fachübergreifende Diskussionen waren und Impulse für die Weiterführung der Arbeit vermittelten. Hierfür und für die Übernahme des Zweitberichtes möchte ich ihm herzlich danken.

Meinen Kollegen in der Abteilung Verteilte Systeme, speziell Christian Maihöfer, danke ich für fachlichen Rat und für viele anregende Diskussionen.

Die Netzwerkadministration des Institutes für Parallele und Verteilte Systeme, und hier ist besonders Franz Fabian zu nennen, ermöglichte in großzügiger und dankenswerter Weise die teilweise mit einem hohen Datenvolumen verbundenen Multicastexperimente.

Nicht vergessen sei der von der DFG und der Universität Stuttgart geleistete finanzielle Support, dank dessen Teilergebnisse der Arbeit auf internationalen Fachtagungen vorgestellt werden konnten.

Ein herzliches Dankeschön gebührt auch meinen Eltern, die mich während meiner gesamten Ausbildung begleitet und in jeder Hinsicht unterstützt haben.

Surrey, den 08. Juni 2007 Jens-Uwe Klöcking

Kurzfassung

Eine Vielzahl neuer Anwendungsgebiete des Internets basiert auf der effizienten Übertragung einer Nachricht an eine Gruppe von Empfängern, der so genannten Gruppenkommunikation. Beispiele für derartige Anwendungen sind Nachrichten- und Softwareverteilung, verteilte Berechnungen, Videokonferenzen, Fernunterricht sowie Spiele. Die Netzwerkressourcen werden durch Gruppenkommunikation sehr effizient genutzt, denn das einmalige Senden einer Nachricht reicht aus, um von allen Teilnehmern einer Gruppe empfangen zu werden.

Einige Einschränkungen der Dienstqualität derzeitiger Gruppenkommunikationslösungen im Internet behindern jedoch ihre generelle Nutzung. Hiervon betroffen ist unter anderem die Fairness bezüglich der Verzögerung der Nachrichtenauslieferung. Dieser spezielle Parameter der Gruppenkommunikation bezeichnet die Zeitspanne zwischen dem ersten und letzten Eintreffen einer Nachricht bei einer Gruppe von Empfängern. Messungen mittels eines dafür entwickelten passiven Verfahrens zeigen für einige Anwendungen nicht tolerierbare Verzögerungsunterschiede auf. Das Ziel der Arbeit besteht darin, die Verzögerungsunterschiede zu minimieren, um einen fairen Dienst für nicht kooperative Anwendungen der Gruppenkommunikation im Internet, wie z. B. Informationsdienste und elektronische Märkte bereitzustellen.

Zur Lösung des Problems wurden drei Ansätze erarbeitet. Durch den Einsatz von Servern konnten Verzögerungsunterschiede der Nachrichten ausgeglichen werden. Die in der Anwendungsschicht angesiedelten Ansätze stellen keine besonderen Anforderungen an die Netzkomponenten und können daher schrittweise eingeführt werden. Darüber hinaus berücksichtigen sie die aktuelle Netzwerklast und sind so in der Lage, die Gesamtverzögerung der Nachrichtenauslieferung gering zu halten.

In einem der Ansätze überwacht sichere Hardware in Form von Smart Cards direkt bei den Empfängern den Auslieferungszeitpunkt der Nachrichten. Hierfür wurden drei Protokolle entwickelt, die die Synchronisation der Smart-Card-Uhren, die Auslieferung der Daten und eine Rückmeldung der tatsächlichen Verzögerung ermöglichen. Mittels Analyse und Simulationen wurde eine signifikante Reduktion der Verzögerungsunterschiede der Nachrichten zwischen den Empfängern nachgewiesen. Ein Prototyp wurde implementiert, um die mit gegenwärtiger Smart-Card-Hardware erreichbare Reduktion von Verzögerungsunterschieden zu ermitteln und die Tragfähigkeit des Ansatzes zu demonstrieren.

Reduction of Inter-Receiver Delay Jitter in Internet Group Communication

English Abstract of the Dissertation

The continuous growth of the Internet as well as the ongoing development of innovative services calls for efficient usage of transmission capacities. Multicast as a bandwidth-efficient transmission method can support many of these services in a resource-saving manner. It delivers a message from one source to a large number of receivers without creating copies at the sender. Rather, the routers in the network take care of replicating the package where necessary. Therefore, the message needs to be sent over each link of the network only once.

Among the applications benefiting from multicast are information distribution and group communication services, especially live audio and video services. In addition, games or caching of frequently used web pages can be efficiently realized with multicast. In the information society, however, the access to information does not only have a non-material value, but also gains influence on material values. The economic value of particular information equals the benefit from decisions possible to make on the basis of possessing this information. To give all users equal opportunities, it is therefore necessary to provide not only a fast but also a simultaneous access to the information required. This kind of service is called delay-fair delivery of messages.

Multicast is a prerequisite for a delay-fair delivery of messages as it allows the simultaneous transmission of messages. However, for many applications this is not sufficient. For example, considering a system that provides sensitive stock quote information from the user's point of view. The user expects to receive the information not later than any other participant of the service because he wants to be able to respond to the information with equal opportunities. Delay jitter among message copies caused for example by different path lengths to the receivers or varying load conditions in different Internet areas can thwart these expectations. This motivated the present work, which aims at the development of methods for reducing inter-receiver delay jitter.

In this work, applications are identified that benefit from a delay-fair service. A method to measure inter-receiver delay jitter is developed in order to allow service providers to estimate this parameter in the Internet. The method distinguishes itself by low administrative overhead as a way to passively monitor existing multicast groups. The main

contribution of this thesis is the development of novel methods for reducing inter-receiver delay jitter. In contrast to conventional synchronization methods, they are especially adapted to the Internet, consider security aspects and do not demand substantial changes of the network nodes.

The thesis is structured as follows. In Chapter 1, the motivation for this work is given and its structure is explained. Applications that require a small inter-receiver delay jitter are categorized in Chapter 2. Chapter 3 describes a method to measure inter-receiver delay jitter in the Internet. Methods to reduce inter-receiver delay jitter are presented and evaluated in Chapter 4 and 5, respectively. Finally, Chapter 6 summarizes the results and concludes with an outlook to specific optimization potential.

Non-Cooperative Real-Time Applications

Applications that distribute valuable information benefit from a delay-fair message delivery. After explaining the notion of fairness from a historical and a problem-oriented point of view, delay-fair message delivery is defined in Chapter 2. The delivery of a message is said to be delay-fair if the message is available to all participants within a time interval that does not significantly affect their competition. The definition implies (i) the participants are competing against each other for the earliest possible access to valuable information; (ii) the information must be available from only one source; (iii) the participants have to react to the information in order to individually profit from them. Applications that meet all three criteria are defined as non-cooperative real-time applications.

Subsequently, these applications are analyzed and categorized according to their fairness requirements. Differences among the applications derive from the kind of communication relationships between information source/sink and participants. Open, unidirectional relationships are distinguished from closed, bidirectional relationships. According to the direction of the information flow, the former category is further subdivided into two classes: Class 1 - *Information Delivery*, the information flows from the information source to the participants, and Class 2 - *Information Aggregation*, with reverse information flow from the participants to an information sink. This is called for because the system is open in the sense that participants can use the received information *outside* of the considered communication system as well as external information has influence to decisions participants make.

In contrast, communication in systems with closed, bidirectional relationships takes place only *within* the system. Externally gained information is useless within the system. Equally, information from the system (e.g. a game) is not applicable outside the system. Depending on the frequency of message exchange, closed bidirectional systems are further subdivided into two classes: Class 3 - *Discrete Information Exchange*, and Class 4 - *Continuous Information Exchange*.

Different fairness criteria apply to the four application classes. The fairness for Information Delivery (Class 1) is established if information is delivered simultaneously to all participants. For Information Aggregation (Class 2), the highest level of fairness is reached if the aggregator considers the actions of the participants in the sequence of their transmission time. Discrete Information Exchange (Class 3) allows an unambiguous correlation of information and reaction to it. The response time is therefore the relevant fairness criterion. By contrast, there is no way to measure the response time for Continuous Information Exchange (Class 4) because the system is unable to restore causal relations between information and reaction. The participants depend therefore on both a simultaneous information delivery and a fair assessment of the sequence of their actions. Based on this categorization as well as on the analysis of human response time, the requirements regarding inter-receiver delay jitter are defined.

Delay Jitter among Multicast Receivers

In Chapter 3, causes for delay differences of messages over unicast routes are discussed. It is established that variable delay is introduced in all layers of the protocol stack causing different delays for two consecutive messages. Jitter of unicast messages between a sender and a receiver is defined as the difference between maximum and minimum delay of messages within this communication. Multicast message delivery is even more susceptible to message jitter, as in general the message delay from the source to a specific receiver differs from that to any other receiver. This type of jitter is called inter-receiver delay jitter and is defined as the difference between maximum and minimum delay that message copies incur during group communication. It is mainly caused by different path lengths and different load at the network nodes.

Whether an application that requires small inter-receiver delay jitter can run on the infrastructure of the current Internet or not can be decided by measuring the inter-receiver delay jitter. The focus of this chapter is therefore on the introduction of a method to

measure inter-receiver delay jitter in the Internet. To keep the administrative overhead low, a passive measurement method is chosen. The data to calculate inter-receiver delay jitter are obtained by listening to multicast traffic of existing groups. This approach has the advantage that neither additional measurement software has to be installed at the participants of group communication nor changes to network nodes are necessary. Besides analyzing data of real groups, the interference with the measurements is kept at a minimum. The proposed method can be applied to sparse groups with a low join/leave frequency using the Real-Time Transport Protocol (RTP).

In order to gather delay jitter data, the timestamps of the RTP Control Protocol (RTCP) are used in a way that allows to calculate the one trip time of specific control packets. RTCP serves as a feedback protocol for multimedia applications. Using this protocol, senders transmit via multicast various statistical data in the form of so-called sender reports to other senders of a group communication. By determining the one-way transit times of all copies of such a multicast control packet to all of its destinations, the inter-receiver delay jitter can be calculated. Because the timestamps used are generated on different computers and are therefore not necessarily synchronized to a common time source, the accuracy of the clocks is observed. The offset of a remote clock is estimated using timestamp request packets with the lowest round trip time. By interpolating between clock offsets found at the lowest round trip time, the drift of the clocks is determined. This provides the opportunity to correct the RTCP time stamps and consequently calculate the accurate inter-receiver delay jitter.

The order of magnitude of inter-receiver delay jitter in the Internet is assessed by measuring inter-receiver delay jitter of control messages in Multicast Backbone (MBone) video conference sessions. Even for small groups, inter-receiver delay jitter in the order of seconds is observed, which is not tolerated by non-cooperative real-time applications like stock exchange information services, electronic markets or games. Therefore, chapter 4 is dedicated to the development of methods that use the advantages of multicast message delivery but simultaneously aim at a delay-fair service.

In Chapter 4, different methods are introduced to reduce inter-receiver delay jitter in group communication. The abstraction level of these methods is examined and their adaptability to overall delay analyzed. Security is proven to be the most important criterion for quality of service in case of non-cooperative applications and a precondition to realize the principle of fairness. The analysis of related work identifies fundamental disadvantages of present methods and justifies the necessity for a different approach. Currently

existing synchronization methods are not applicable to this problem, either they do not take security aspects into account or they require substantial changes to network nodes. Security aspects play an important role for a fair message delivery since a cooperative behavior of group communication receivers cannot be expected in the case of distributing valuable information.

Besides the minimization of inter-receiver delay jitter, other demands on a delay-fair service for non-cooperative real-time applications are defined. Among them are fairness over all participants, minimization of overall delay, tolerance regarding data rates and receiver composition, independence of network technologies as well as low message overhead.

A delay-fair service based on IP multicast offers the best prospects to fulfill these requirements. Essential characteristics are abstracted for problem solving in a system model consisting of a network, an IP multicast and a security model. The basic concept of the proposed methods to reduce inter-receiver delay jitter among participants of group communication assumes trusted regions near participants of group communication. Because of the necessary trust, the application layer is chosen for service provision in order to limit the trusted regions to individual, monitorable areas. Within these areas, the proposed methods adjust inter-receiver delay jitter of messages using dedicated servers or secure hardware close to the receivers by resending the information at globally synchronized time.

Based on the concept developed for the reduction of inter-receiver delay jitter, three approaches are presented, which differ mainly in function and location of the trusted region. In the *data-server approach*, data servers are placed into the trusted region. The data servers receive, decrypt and resynchronize encrypted messages from an information source and send them unencrypted at a specific revelation time to participants of this group communication. The data-server approach represents the most general case of this concept. Two further approaches are specializations, partly with higher demands, but also with a further decrease of inter-receiver delay jitter.

The *key-server approach* separates message delivery from provision of revelation information. The trusted regions provide keys for message encryption at revelation time, whereas the messages are sent directly to the participants.

The *time-server approach* reduces the function of the trusted region within the network to the provision of time information and establishes instead the trusted region directly at the participants. The information source sends encrypted messages via a multicast service.

The data sent is comprised of the encrypted message, the encrypted key for the message, and the revelation time. Only the secure hardware connected to the individual receiver can decrypt the message key. Hence, the receiver forwards these key data to the secure hardware. The secure hardware decrypts the key and supplies the receiver with the key at the given information revelation time. If the information revelation time is elapsed before the receiver obtains the data, he can request a deadline extension for subsequent messages by means of feedback. This results in an increased revelation time for all receivers.

The basic idea of the approach described here is to reduce the inter-receiver delay jitter to the delay differences among pairs of receivers and their - in terms of delay - nearest time server. The approach is based on the assumption that a multitude of certified time servers exist in the entire network and hence a small inter-receiver delay jitter is achievable. Compared to approaches that involve router changes, less network overhead is introduced and a partial deployment of the approach is possible. In contrast to the data server approach, no messages need to be stored within the network. The trusted region comprises only the time servers. The inter-receiver delay jitter is expected to be further reduced as it can be assumed that there will exist more trusted time servers than trusted general purpose servers or servers specialized to the inter-receiver delay jitter problem.

Three protocols are introduced: a clock synchronization protocol for providing the secure hardware clock with real time, a data delivery protocol assuring the encrypted delivery of the information and a feedback protocol for determining a suitable information revelation time.

- Clock Synchronization Protocol

  To prevent a premature message delivery, a clock synchronization protocol is presented that keeps the clocks of the secure hardware synchronized to a reference time and authenticates the source of the time information.

  In general, clock synchronization protocols are based on the assumption that the delay between the time server and the client and vice versa equals. Unfortunately, a participant can control both delays. Introducing an asymmetric artificial delay, he is able to deceive the secure hardware while estimating the delay to the time server. To rule out illegal advancement of the clocks, the time stamps received by the secure hardware have to be adopted as real time ignoring the delay between the time server and the secure hardware. This comes with the advantage that the time

server can work in broadcast mode. The clocks of the time servers are assumed to be synchronized.

A prerequisite for the clock synchronization protocol is to choose time servers that can be trusted and certify their public key by a certification authority. This is generally done by an off-line conversation. Then, the participant's host with attached secure hardware joins a multicast group that is used by time servers to distribute their time stamps. The receiver determines nearby servers by comparing the time stamps with his local clock. To be immune to time server downtime, the receiver chooses more than one server. The time servers publish their certified public keys by multicasting them to the group periodically or on request. The host delivers the certificates to the secure hardware. Periodically, signed time stamps are sent by the time servers and are forwarded by the host to the secure hardware for clock synchronization.

- Data Delivery Protocol

  Furthermore, a data delivery protocol is described that allows message delivery before message revelation time without compromising information secrecy. Three special characteristics of the problem are considered. First, the time aspect of the problem requires that each key issued to the receivers can only be used once, namely at the given information revelation time. Second, the computational speed of secure hardware is low. Therefore, the decryption should be accomplished with low computational requirements. Third, because of the potentially high number of receivers, a data distribution by multicast must be possible.

  To relieve the sender of key management, a group key manager (GKM) administers the group keys. Again, a prerequisite for the protocol is the issue of a certificate for the GKM and secure hardware public keys. In a challenge-response message exchange, the GKM authenticates itself to the sender and transmits a unique group key. Then, the hosts that want to receive data request a group key on behalf of their attached secure hardware from the GKM. By verifying the certificate of the secure hardware, the GKM checks whether a secure hardware is permitted to participate. The GKM encrypts the group key with the public key of the secure hardware attached to the receiver. The receiver forwards this encrypted key to the secure hardware.

  The group key is used to send a message key and the information revelation time

to the secure hardware. The symmetric encryption allows a rapid decryption at the secure hardware. The data are en-/decrypted with the message key, which is a symmetric key too. This allows secure multicast distribution of data.

- Feedback Protocol

  A dynamic mechanism to calculate information revelation time is presented in order to achieve a low overall message delay along with a low inter-receiver delay jitter. It uses quality of service feedback for adaptation of information revelation time to current network load. On behalf of the secure hardware, the feedback protocol allows the receiver to tell the sender that the message has not arrived before the information revelation time passed. Based on the feedback, the sender adjusts the information revelation time accordingly.

  The minimization of the overall delay is achieved by three components: a one trip time estimator at the group participants, a feedback mechanism and an information revelation time estimator at the sender. Observing the message arrival times, the one trip time estimator at the participant calculates the delay that an incoming message experienced on its way from the sender. Using an exponentially-weighted moving average algorithm, the estimator predicts future message delay. To improve the estimation, the variance of the message one trip time is taken into account as well. A feedback mechanism transmits this estimation to the sender. For small groups, this is achieved by a revelation time expiration notification from certain participants. For large groups, a hierarchical feedback mechanism is considered. An information revelation time estimator at the sender adapts the information revelation time for future messages according to the feedback received.

With all approaches, information secrecy is guaranteed by encryption of the messages up to the information revelation time. Before information revelation time, information and keys are only accessible by servers and secure hardware. Function and security of servers and secure hardware are guaranteed by a certification agency. Therefore, the approaches base on the trust in the certification agency as well as in the information source to deliver the information over just one service and not by other means.

The remainder of the chapter analyzes how the results obtained can be mapped to other application classes. The solution for Information Delivery (class 1) applications, can be applied to classes 2 and 3, which require beside a fair information access a fair interaction

with the system. A protocol for applications with open unidirectional communication (Information Aggregation, class 2), is proposed. For closed bidirectional communication (Discrete Information Exchange, class 3), a combination of mechanisms used for information delivery and aggregation is recommended. In this way, a participant's response time, which is defined as the fairness criterion for class 3, can be approximated.

Performance Analysis

In Chapter 5, the performance of methods proposed for reducing the inter-receiver delay jitter of group communication, i.e. the data server, key server and time server method, is analyzed and compared. First, important factors that influence the extent of inter-receiver delay jitter are discussed. Among them are the distance between server and participants, the link with the lowest transmission capacity as well as the length of the time-sensitive message. It is shown that the inter-receiver delay jitter increases linearly with increasing distance and can reach up to 100 ms. Thus, inter-receiver delay jitter can be reduced substantially by shortening the distance between server and participants.

In addition, it is demonstrated that the difference between maximum and minimum delay of a message over all distances between a receiver and its server is especially dependent on the transmission capacity of the particular links and on the length of the message. The examination of these factors shows that small transmission capacities and large messages lead to high delay-jitter. Key and time server approaches further reduce this kind of delay-jitter because the separate transmission of synchronization information makes both approaches independent of message length. An evaluation of message overhead shows for all approaches a linear dependency from the number of payload messages. The least message overhead is obtained for the time server approach provided that the message data rate is higher than the data rate of time messages.

Furthermore, message delay and resulting inter-receiver delay jitter are theoretically analyzed. The message delay between server and receiver is modeled using an M/G/1 queuing model of its links. Subsequently, inter-receiver delay jitter is estimated on the basis of a university scenario with high bandwidth and a home user scenario with ISDN access representing low bandwidth. The analysis shows that inter-receiver delay jitter is significantly more reduced with the key or time server approach than with the data server approach, which is mainly due to the separation of payload, revelation and time messages.

The algorithms have been implemented into a prototype in order to validate the designed protocols. Using a Gemplus 211PK Java Card, which is based on a Phillips 8-bit microcontroller, the performance of currently available smart card technology has been evaluated. The measurement of the authentication delay of time stamps with different methods resulted in values between 380 ms for DES-CBC with Message Authentication Code and 470 ms for SH1 DES. Expectedly, RSA based methods produced higher values of about 860 ms. The crucial delay-jitter for the different methods, however, was found to be extremely small: 11 ms for the RSA based method and 11 ms for 99 % of the measured data with a maximum delay-jitter of 20 ms for the DES-CBC method. Lower context switch times in operating systems should lead to even better results.

Interactive applications call for an even lower delay than presently achievable. The optimization potential of current smart cards is limited because the run time of the programs depends mainly on the performance of the cryptography co-processor. Opportunities for higher performance arise from a new generation of smart cards based on 32-bit-RISC technology with appropriate co-processors, improved encryption algorithms and integrated timers.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Das Internet ist ein weltweiter, dezentraler Verbund von Computer-Netzwerken zur Bereitstellung von Informationen und Dienstleistungen sowie zur Übermittlung von Nachrichten. Seine Entwicklung begann 1969 mit dem Zusammenschluss von 4 Knoten des ARPANET, eines von der Advanced Research Projects Agency (ARPA) des US-amerikanischen Verteidigungsministeriums entwickelten dezentralen Forschungsnetzes. Ein entscheidender Meilenstein war 1983 die Einführung des verbindungsorientierten Transfer Control Protocol (TCP) und des paketvermittelten Internet Protocol (IP) als einheitlichem Netzwerkprotokoll TCP/IP. Damit wurde das Internet zu einem Verbund von Netzwerken, die TCP/IP nutzen und als solches definiert [Cerf & Kahn 1974].

Entsprechend seiner Herkunft aus dem ARPANET war die Nutzung des Internets zunächst nur militärischen und wissenschaftlichen Einrichtungen erlaubt [Leiner et al. 1997]. Erst 1993 wurde das Internet für die Allgemeinheit zugängig. Das daraufhin einsetzende rasante Wachstum seiner Nutzergemeinde belegen eindrucksvoll vom *US Department of Commerce* veröffentlichte Zahlen. Während das Radio 38 Jahre und das Fernsehen 13 Jahre benötigten, bis sie 50 Millionen Teilnehmer erreichten, waren beim Internet hierfür nur 4 Jahre erforderlich [Margherio et al. 1998]. Bis zum September 2002 stieg die Zahl der Internetnutzer weltweit auf über 600 Millionen [ITU 2002].

Gleichzeitig dringt das Internet in immer neue Technik- und Dienstbereiche vor und gewinnt zunehmend ubiquitäre Verbreitung. Schon heute werden Mobilfunkgeräte mit Zu-

griffsmöglichkeiten auf das Internet versehen, Fahrzeuge werden über Internet mittels Ferndiagnose überwacht [OSGi 2000], eine Vielzahl von Sensoren wird zur Umweltbeobachtung kombiniert [Riebeek 2003], im Haushalt installierte Geräte sind über das Internet weltweit steuerbar [Werkmann & Schwarze 2002, Lee et al. 2003].

Andererseits wird über das Internet eine Vielzahl von Diensten bereitgestellt. Zu den populärsten zählen die seit 1971 verfügbare Elektronische Post und das 1991 gestartete World Wide Web (WWW), das seine Inbetriebnahme dem ersten, von Tim Berners-Lee am Counceil Européenne pour la Recherche Nucléaire (CERN) entwickelten World-Wide-Web-Server verdankt [Berners-Lee 1996]. Immer wieder reizen seitdem neue Dienste die vorhandenen Übertragungskapazitäten bis an ihre Grenzen aus. Das Wortspiel „World Wide Wait“ Mitte der 90er Jahre [Khare & Jacobs 1999, Guo & Matta 2001] und die Drosselung von Tauschbörsen-Diensten durch die Provider sind markante Beispiele hierfür [ZDNet UK News 2001].

Die Ausbreitung des Internets sowie die ständige Entwicklung innovativer Dienste zwingt Dienstanbieter zu einer effizienten Ausnutzung der Übertragungskapazitäten. Viele Dienste können dabei durch Multicast, ein sehr bandbreiteneffizientes Verfahren, unterstützt werden. Bei diesem Verfahren werden Nachrichten, die an eine Vielzahl von Empfängern gerichtet sind, nicht beim Sender vervielfältigt, sondern erst in den Vermittlungsknoten je nach Bedarf erzeugt. Somit wird auf jedem Übertragungsabschnitt nur eine Nachricht gesendet.

Zu den Anwendungen, die von Multicast profitieren, gehören Informationsverteildienste, Gruppenkommunikationsdienste und insbesondere Live-Audio- und Videoübertragungen. Aber auch Spiele oder das Vorhalten von häufig nachgefragten Web-Seiten [Linder et al. 2002] können mittels Multicast effizient realisiert werden.

Im Zeitalter der Informationsgesellschaft stellt der Zugang zu Informationen nicht allein einen ideellen Wert dar, sondern gewinnt zunehmend Einfluss auf materielle Werte. Dabei bemisst sich der Wert einer Information an dem wirtschaftlichen Nutzen, der sich daraus ableiten lässt. Um Chancengleichheit für die Nutzer zu wahren, ist sowohl ein schneller als auch ein fairer, d.h. zeitgleicher Zugang zu den Informationen erforderlich.

Multicast ist dafür eine notwendige Voraussetzung, da es zumindest das gleichzeitige Aussenden der Nachrichten gewährleistet. Für viele Anwendungen ist dies jedoch nicht ausreichend, da Verzögerungsunterschiede zwischen den Nachrichtenkopien auftreten. Betrachtet man zum Beispiel einen Börsentickerdienst aus der Sicht des Nutzers, so erwartet

dieser, dass er die Informationen nicht später erhält als andere Dienstteilnehmer, um mit der gleichen Chance auf die Nachricht reagieren zu können. Verzögerungsunterschiede zwischen den Nachrichtenkopien, hervorgerufen z. B. durch unterschiedliche Pfadlängen zu den Teilnehmern oder durch unterschiedliche Last in verschiedenen Bereichen des Internets, können diese Erwartung jedoch nicht erfüllen. Aus dieser Motivation heraus ist die vorliegende Arbeit entstanden. Ihr Ziel ist es, Verfahren zur Reduktion von Verzögerungsunterschieden zu entwickeln.

1.2 Überblick und Aufbau

In der vorliegenden Arbeit werden Anwendungen identifiziert, die von einem Verzögerungsunterschied-fairen Dienst profitieren können. Mit Hilfe eines hierfür entwickelten Messverfahrens können Dienstanbieter die bestehenden Verzögerungsunterschiede im Internet überwachen. Das Verfahren zeichnet sich dadurch aus, dass keine zusätzliche Mess-Software bei Multicast-Teilnehmern installiert werden muss und der hierfür erforderliche administrative Aufwand entfällt.

Der wesentliche Beitrag der Arbeit besteht in der Beschreibung neuartiger Verfahren zur Reduktion von Verzögerungsunterschieden zwischen Empfängern. Im Gegensatz zu bisherigen Verfahren sind diese speziell an die Eigenschaften des Internets angepasst. Die Betrachtung bestehender Synchronisationsverfahren zeigt, dass sie auf die Problemstellung nicht angewendet werden können, da sie keine Sicherheitsaspekte berücksichtigen oder umfangreiche Änderungen in den Netzknoten verlangen.

Die hier vorgestellten neuen Verfahren gleichen Verzögerungsunterschiede der Nachrichten zwischen Empfängern in abgesicherten Bereichen oder durch sichere Hardware aus. Mittels spezieller Server ist es möglich, diesen Ausgleich empfängernah zu realisieren. Aus dem Einsatz von Servern ergeben sich eine Reihe von Vorteilen für die entwickelten Verfahren. Da keine Annahmen über die Netzwerkknoten getroffen werden, passen sich die Verfahren gut an die gegenwärtige Struktur des Internets an, ohne dass Änderungen an den Netzwerkknoten notwendig werden. Dadurch können die Verfahren schnell und gezielt eingesetzt werden. Die Erweiterung auf ein dynamisches Verfahren erlaubt gleichzeitig die Reduktion der Gesamtverzögerung der Nachrichten. Bei einem der verwendeten Verfahren wird die Funktion der Überwachung des Zeitpunktes zur Veröffentlichung der Informationen auf sichere Hardware der Endgeräte übertragen. Die Serverfunktionalität kann sich

dadurch auf vertrauenswürdige Zeitinformation beschränken. Durch die Trennung von Synchronisationsinformation und Nachrichtenauslieferung ergibt sich die größtmögliche Reduktion des Verzögerungsunterschiedes. Zudem erfolgt die Nachrichtenauslieferung direkt über Multicast bis zum Empfänger.

Durch Analyse und Simulation wird die Tragfähigkeit der Ansätze gezeigt. Ein Prototyp wurde implementiert, um die mit gegenwärtig sicherer Hardware erreichbare Reduktion von Verzögerungsunterschieden zu ermitteln.

Im Einzelnen ist die Arbeit wie folgt gegliedert. In Kapitel 2 werden Anwendungen, die nur einen geringen Verzögerungsunterschied zwischen den Empfängern tolerieren, klassifiziert. Kapitel 3 beschreibt ein Verfahren zur Messung von Verzögerungsunterschieden im Internet. In Kapitel 4 werden die Verfahren zur Verringerung von Verzögerungsunterschieden zwischen Empfängern auf der Anwendungsschicht vorgestellt und in Kapitel 5 evaluiert. Die Ergebnisse werden in Kapitel 6 zusammengefasst und die Arbeit mit einem Ausblick auf spezielle Optimierungsmöglichkeiten abgeschlossen.

Kapitel 2

Nicht-kooperative Echtzeitanwendungen

Die in Abschnitt 1.1 erwähnte Verteilung von Börsenkursen ist nicht die einzige Anwendung, bei der es auf einen zeitgleichen Zugang zu Informationen ankommt. Viele weitere Anwendungen, die wertvolle Informationen verteilen, können von einer verzögerungsfairen Nachrichtenauslieferung profitieren oder werden erst durch diese ermöglicht. Sie werden in diesem Kapitel näher untersucht und als nicht-kooperative Echtzeitanwendungen charakterisiert. Insgesamt werden vier verschiedene Klassen von Anwendungen erfasst, für die auf der Basis unterschiedlicher Fairness-Ansprüche Anforderungen an einen verzögerungsfairen Dienst definiert werden. Vorangestellt wird in Abschnitt 2.1 eine Beschreibung des Problems am Beispiel der Verteilung von Ad-hoc-Informationen und in Abschnitt 2.2 eine Erläuterung zum Fairness-Begriff aus historischer und aktueller, problembezogener Sicht.

2.1 Problemstellung

Für einen vollkommenen Markt ist Markttransparenz eine unabdingbare Voraussetzung. Das bedeutet, die potenziellen Käufer eines Produktes, z. B. eines Wertpapiers, besitzen unmittelbar vor dem Kauf alle Informationen, die für ihre Kaufentscheidung relevant sind. Da jedoch in der Praxis aus verschiedenen, hier nicht näher zu erläuternden Gründen keine vollkommene Markttransparenz existiert, können die Teilnehmer am internationalen

Börsenhandel nur den Anspruch erheben, bei Kauf- und Verkaufsaktionen gegenüber ihren Mitbewerbern nicht benachteiligt zu werden. Im konkreten Fall heißt das, sie können erwarten, dass ihnen die erforderlichen Informationen einschließlich kurserheblicher Tatsachen, sogenannter Ad-hoc-Informationen, in gleichem Umfang und zu gleicher Zeit zur Verfügung stehen wie allen anderen Marktteilnehmern. Die praktische Umsetzung dieser Forderung ist jedoch mit den gegenwärtig zur Verfügung stehenden Verfahren nur bei Inkaufnahme erheblicher Nachteile in Bezug auf Reichweite, Übertragungsgeschwindigkeit und Kosten möglich. Die Gründe hierfür sind einmal in der Inhomogenität der Nachrichtendienste und somit fehlender Gleichzeitigkeit der Informationsübermittlung zu sehen, zum anderen in der begrenzten Reichweite verschiedener Dienste, z. B. von Satellitennetzwerken. Demgegenüber hat die Gruppenkommunikation mittels Multicast neben der globalen Reichweite den für die Übermittlung von Ad-hoc-Informationen entscheidenden Vorteil, dass Nachrichten mit hoher Geschwindigkeit gleichzeitig an viele Empfänger gesendet werden können. Einziger Nachteil dieses Internetdienstes gegenüber Satellitengebundenen Diensten ist die Tatsache, dass Multicast wegen der unterschiedlichen Weglängen, die Informationen im paketvermittelten Dienst bis zu ihrem Ziel durchlaufen, nicht für die gleichzeitige Auslieferung der Informationen garantiert. Es ist daher erforderlich, nach Lösungen zu suchen, die die Vorteile von Multicast nutzen und dennoch eine faire Auslieferung der Informationen garantieren.

2.2 Fairness

Der von dem altenglischen Wort *faegere* abgeleitete Begriff *fair* bedeutete ursprünglich passend, angenehm, schön. Er wurde im England früherer Jahrhunderte benutzt, um ein anständiges, d.h. standesgemäßes Verhalten von Mitgliedern gehobener Gesellschaftsschichten zu beschreiben. Erst im 19. Jahrhundert wurde er vom Sport aufgegriffen, wo er nach und nach einem Bedeutungswandel unterlag. Fair Play stand vor allem für das Einhalten der Spiel- und Wettkampfregeln [Weiss 2000]. Mit der Verwendung des Begriffes in Politik und Gesellschaft kam es zu einer weiteren Umwertung, so dass heute weltweit sehr unterschiedliche und teilweise widersprüchliche Vorstellungen von Fairness bestehen. Als anerkannte Grundprinzipien gelten:

- Größtmögliche Chancengleichheit
- Strikte Einhaltung von Regeln

- Achtung des Gegners und seiner Unantastbarkeit

Neben einer differenzierten Auslegung des Begriffs spielen auch unterschiedliche, den jeweiligen Anforderungen entsprechende Fairness-Levels eine Rolle. Für den Austausch von Gütern im Bereich des elektronischen Handels (E-Commerce) beispielsweise haben [Gartner et al. 1999] formale Fairness-Definitionen erarbeitet und drei eindeutig abgrenzbare Fairness-Levels definiert.

Bei der paketvermittelten Nachrichtenübertragung ist Fairness zur Sicherung des optimalen Datenflusses (*congestion control*) von entscheidender Bedeutung. Im speziellen Fall, der Gruppenkommunikation mittels Multicast, bezieht sich das sowohl auf das Verhalten innerhalb eines Multicast-Datenstromes, der Multicast-Datenströme untereinander sowie zwischen Multicast- und Unicast-Datenströmen [McCanne 1996, Vicisano et al. 1998, Rhee et al. 2000, Byers et al. 2000, Widmer 2003]. Während diese Fairness-Probleme bereits eine intensive Bearbeitung erfahren haben, ist über das infolge von Verzögerungsunterschieden bei der Informationsauslieferung auftretende Fairness-Problem relativ wenig bekannt. Seine Nichtbeachtung impliziert jedoch für den Einzelnen unter Umständen schwerwiegende wirtschaftliche Folgen, die bei einer verzögerungsfairen Nachrichtenauslieferung zu vermeiden sind. Der Begriff *verzögerungsfair* soll dabei wie folgt definiert werden:

> Definition 2.1 (Verzögerungsfaire Nachrichtenauslieferung)
> Eine Nachrichtenauslieferung an mehr als einen Teilnehmer ist verzögerungsfair, wenn die Nachricht allen Teilnehmern innerhalb eines den Wettbewerb nicht signifikant beeinflussenden Zeitintervalls zur Verfügung steht.

Im folgenden Abschnitt werden die von Verzögerungsunterschieden bei der Nachrichtenauslieferung betroffenen Anwendungen näher betrachtet und eine Klassifizierung bezüglich der erforderlichen Fairness vorgenommen.

2.3 Klassifizierung und Eigenschaften der Anwendungen

In diesem Abschnitt wird zunächst geklärt, bei welchen Anwendungen die verzögerungsfaire Nachrichtenauslieferung eine entscheidende Rolle für die Akzeptanz durch die jeweiligen Nutzer spielt und durch welche gemeinsamen Merkmale diese Anwendungen gekennzeichnet sind. Anschließend werden die einzelnen Anwendungen entsprechend ihrem spezifischen Fairnessanspruch klassifiziert.

Bereits die in Abschnitt 2.2 gegebene Definition der verzögerungsfairen Nachrichtenauslieferung impliziert eine zwischen den Teilnehmern bestehende Wettbewerbssituation, bei der die Teilnehmer um den frühestmöglichen Zugang zu werthaltigen Informationen konkurrieren. Bezeichnend ist dabei, dass die Teilnehmer die gewünschten Informationen nur von *einer* Informationsquelle beziehen können und, um daraus einen individuellen Nutzen zu ziehen, auf die erhaltenen Informationenen reagieren müssen. Anwendungen, für die diese drei Kriterien zutreffen, werden in dieser Arbeit als *nicht-kooperative Echtzeitanwendungen* bezeichnet.

> Definition 2.2 (Nicht-kooperative Echtzeit-Anwendungen)
> Nicht-kooperative Echtzeitanwendungen sind Anwendungen, bei denen zwei oder mehr Teilnehmer um den frühestmöglichen Zugang zu werthaltigen Informationen aus *einer* Informationsquelle konkurrieren und, indem sie auf erhaltene Informationen reagieren, aus diesen einen Nutzen ziehen.

Wichtigstes Kriterium für nicht-kooperative Echtzeitanwendungen ist das Bestehen einer Wettbewerbssituation, bei der die Teilnehmer um den Erhalt werthaltiger Informationen konkurrieren. Da sie aus einem frühen Informationszugang Vorteile ziehen, haben sie kein Interesse, sich kooperativ gegenüber anderen Teilnehmern zu verhalten. Entsprechend hoch ist das Manipulationsinteresse. Dabei ist ein Angreifer umso mehr an einer Manipulation interessiert, je wertvoller die Information ist und je früher sie dank Manipulation erhältlich ist. Infolgedessen ist das Manipulationspotenzial besonders groß bei Börsenkurstickern, Ad-hoc-Nachrichten und elektronischen Handelssystemen, dagegen gering bei Spielen.

Ein weiteres Kriterium ist die Anzahl unabhängiger Informationsanbieter, denn eine verzögerungsfaire Verteilung der Informationen ist nur sinnvoll, wenn die Informationen ausschließlich über *eine* Quelle zugängig sind. Gibt es für dieselben Informationen einen weiteren Anbieter, der keine verzögerungsfaire Verteilung anstrebt, ist ein verzögerungsfairer Dienst nutzlos. Beispielsweise ist eine verzögerungsfaire Verteilung von Wetterinformationen durch eine Agentur nicht sinnvoll, weil das Wetter von vielen voneinander unabhängigen Agenturen beobachtet wird. Dagegen stehen Unternehmen in der Pflicht, kursrelevante Informationen allen Interessenten gleichzeitig zugänglich zu machen. Die Bekanntmachung über *einen* Dienstanbieter erlaubt eine faire Verteilung.

Ob ein Teilnehmer aus einem frühen Informationszugang tatsächlich einen individuellen Nutzen zieht, hängt von seiner Reaktion auf die erhaltenen Informationen ab. So kann er

z. B. auf Ad-hoc-Meldungen mit einem Aktienkauf reagieren oder in einem Action-Spiel einen bestimmten Spielzug machen.

Während die definitionsgemäßen Merkmale auf alle nicht-kooperativen Anwendungen zutreffen, gibt es zwischen den einzelnen Anwendungen auch markante Unterschiede, die sich auf die erforderliche Fairness auswirken. In Abbildung 2.1 wird eine Klassifikation der nicht-kooperativen Echtzeitanwendungen im Hinblick auf die erforderliche Fairness vorgenommen.

Unterschiede zwischen den einzelnen Anwendungen lassen sich aus der Art der bestehenden Kommunikationsbeziehungen herleiten. Dabei sind offene, unidirektionale von geschlossenen, bidirektionalen Kommunikationsbeziehungen zu unterscheiden. Bei der erstgenannten Kategorie muss der unidirektionale Informationsfluss vom Informationssender zum Teilnehmer und vom Teilnehmer zum Informationsempfänger getrennt betrachtet werden. Die Offenheit des Systems in dem Sinne, dass von Teilnehmern empfangene Informationen auch außerhalb des Systems Verwendung finden können sowie externe Informationen auf die Entscheidungen der Teilnehmer Einfluss haben, erfordern unter dem Aspekt der Fairness die getrennte Betrachtung der Kommunikationswege. Im Gegensatz dazu erfolgt der Informationsfluss bei Systemen mit geschlossenen, bidirektionalen Kommunikationsbeziehungen ausschließlich innerhalb des Systems. Informationen, die außerhalb des Kommunikationssystems gewonnen werden, sind für das System wertlos. Ebenso können aus dem Kommunikationssystem stammende Informationen außerhalb des Systems nicht nutzbringend verwendet werden (z. B. bei einem Spiel). Eine weitere Unterteilung der Anwendungen ergibt sich daraus, dass bei den geschlossenen Kommunikationsbeziehungen entweder ein diskreter oder ein kontinuierlicher Informationsaustausch möglich ist.

Insgesamt resultieren vier Anwendungsklassen, für die unterschiedliche Fairnesskriterien gelten. Für Klasse 1 (Informationsverteilung) besteht die Fairness in der gleichzeitigen Auslieferung der Informationen an die Teilnehmer. Für Klasse 2 (Informationsempfang) wird größtmögliche Fairness erzielt, wenn der Informationsempfänger die Aktionen der Teilnehmer in der Reihenfolge des Sendens berücksichtigt. In Klasse 3 (Diskreter Informationsaustausch) kann eine eindeutige Zuordnung von Information und darauf folgender Reaktion getroffen werden. Die Reaktionsdauer der Teilnehmer spielt daher als Fairnesskriterium die entscheidende Rolle. In Klasse 4 (Kontinuierlicher Informationsaustausch) ist das Messen der Reaktionsdauer nicht möglich, weil das System infolge der Kontinuität des Informationsaustausches die kausalen Beziehungen zwischen einzelnen Informationen und Reaktionen nicht restaurieren kann. Die Teilnehmer sind infolgedessen sowohl auf die

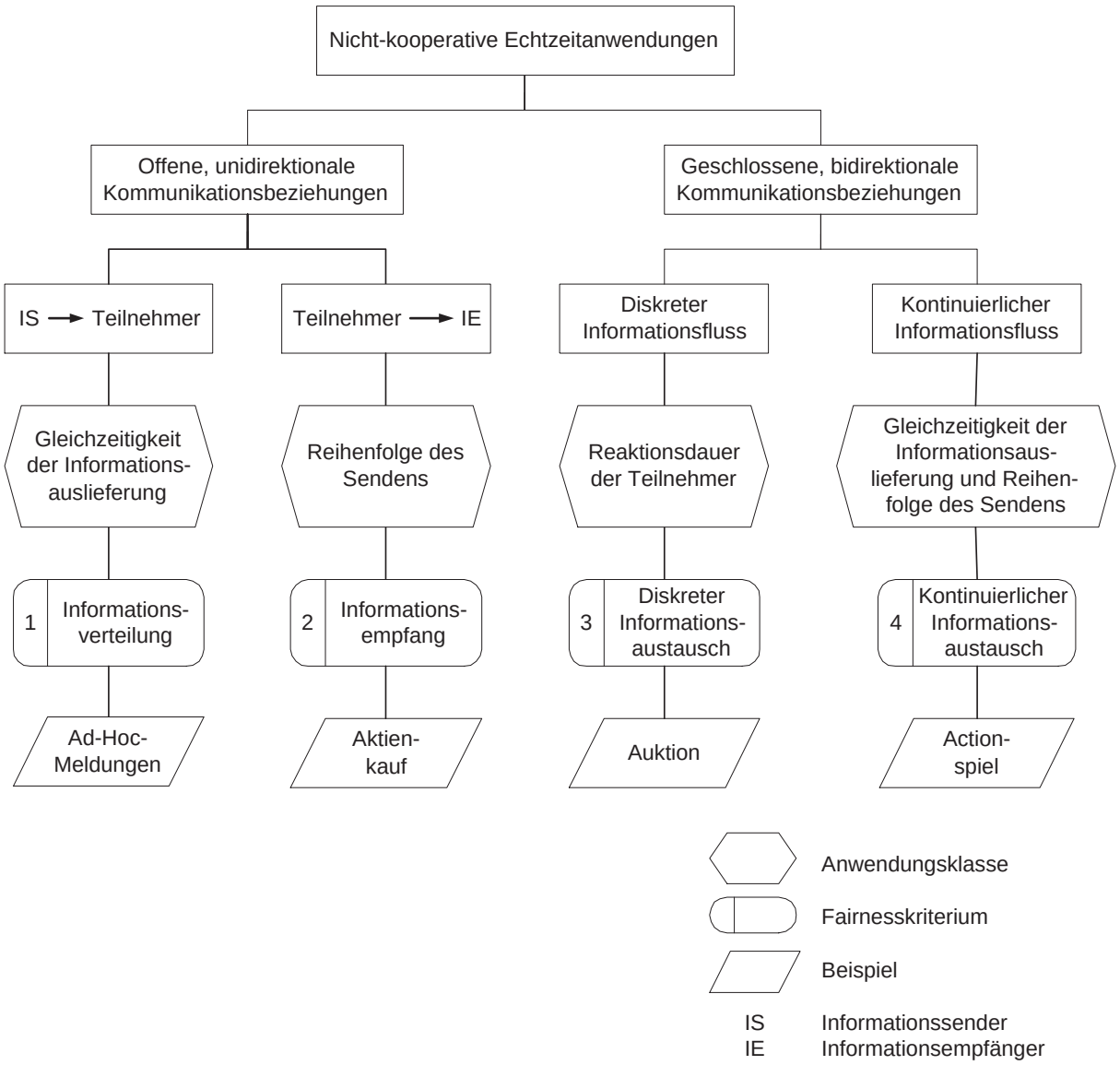


Abbildung 2.1: Klassifizierung von nicht-kooperativen Echtzeitanwendungen im Hinblick auf die erforderliche Fairness

Gleichzeitigkeit des Informationsempfangs als auch auf eine faire Auswertung der Sendereihenfolge angewiesen.

Die 4 Anwendungsklassen sind in Abbildung 2.1 durch je ein Beispiel repräsentiert. Beispiele für nicht-kooperative Anwendungen zur Informationsverteilung stellen die Informationsverteilung über Börsenkursticker und die Übermittlung kursrelevanter Nachrichten mittels Ad-hoc-Meldungen von Unternehmen dar. In beiden Fällen werden Informationen von einer Informationsquelle an eine große Zahl von Teilnehmern übermittelt. Der Aktienkauf ist ein Beispiel für den Informationsempfang eines Handelssystems. Die Informationen, die zur Entscheidung über den Kauf geführt haben, können neben reinen Kursdaten dieses Handelssystems auch beliebige andere Informationen sein. Das System kann daher keine eindeutige Reaktion auf die zur Verfügung gestellten Kursdaten ableiten. Größtmögliche Fairness gegenüber den Teilnehmern bietet daher die Abwicklung der Aufträge in der korrekten Reihenfolge des Sendens. Ein Beispiel für nicht-kooperative Anwendungen mit geschlossenen Kommunikationsbeziehungen und diskretem Informationsfluss sind Auktionen. Neben der bekannten englischen Auktion, bei der die Käufer ihre Gebote solange erhöhen, bis nur noch ein Bieter übrig ist, der das Auktionsobjekt erhält, existieren noch andere Auktionsarten. Bei der holländischen Auktion senkt der Auktionator solange den Preis für das zur versteigernde Objekt, bis ein Bieter dem Preis zustimmt. Die Reaktionsdauer ab Bekanntgabe des neuen Preises ist hier das optimale Fairnesskriterium. Erfahren nicht alle Bieter gleichzeitig den aktuellen Preis, erwächst den zu spät informierten Bietern hieraus ein materieller Nachteil. Weitere Auktionstypen sind die Angebotsauktion, Gebotsauktion und Doppelte Auktion. Bei der Angebotsauktion machen die Verkäufer nacheinander Angebote an alle Bieter. Diese können ein Angebot jederzeit annehmen, aber selbst keine Gegenangebote abgeben. Die Umkehr dessen ist die Gebotsauktion, bei der die Bieter nacheinander Gebote abgeben, die die Verkäufer jederzeit annehmen können. Die Kombination aus beiden Modellen wird als Doppelte Auktion bezeichnet. Verkäufer und Käufer geben gleichzeitig Angebote ab. Die jeweils besten Gebote, d.h. die höchsten Gebote der Käufer und die niedrigsten Gebote der Verkäufer, werden öffentlich bekannt gegeben. Beide Seiten haben dann die Möglichkeit, eines der aktuell besten Gebote der Gegenseite anzunehmen. Auch bei diesen Auktionstypen bekommt der zuerst reagierende Teilnehmer den Zuschlag. Daher kommt es wiederum auf die Reaktionsdauer der Bieter an. Action-Spiele profitieren, bedingt durch ihren kontinuierlichen Informationsfluss, von einer schnellen, gleichzeitigen Offenlegung von Spielentscheidungen anderer Spieler, so dass alle Teilnehmer die gleichen Chancen für eine Reaktion erhalten.

Tabelle 2.1: Gruppenkommunikationseigenschaften potenzieller Anwendungen für einen verzögerungsfairen Dienst

Gruppenkommunikationseigenschaften	Informationsverteilung		Informationsempfang	Diskreter Informationsaustausch	Kontinuierlicher Informationsaustausch
	z.B. Börsenkursticker	z.B. Ad-hoc-Nachrichten	z.B. Aktienkauf	z.B. Auktion	z.B. Action-Spiel
Anzahl der Informationsquellen	1	1	hoch	1 oder mehrere	1 oder mehrere
Nachrichtenhäufigkeit	10/s	50/Tag	hoch	10/min	Hunderte/s
Nutzergruppe	unbekannt	unbekannt	-	unbekannt	bekannt oder unbekannt
Anzahl der Nutzer	hoch	hoch	hoch	gering bis mittel	gering bis mittel
Abonnierfrequenz	hoch	hoch	-	gering	gering

Zwar steht bei Spielen der materielle Gewinn meist nicht im Vordergrund, doch ist der ideelle Wert des Sieges in High-Score-Listen ausreichend, um den Vorteil des frühen Informationszugangs auszunutzen.

Demgegenüber fallen z. B. Videokonferenzen nicht unter die betrachteten Anwendungen. Obwohl auch Konferenzteilnehmer die Beiträge gleichzeitig empfangen sollten, um gleiche Chancen zur Wortmeldung zu haben, kann hier vom kooperativen Verhalten der Teilnehmer ausgegangen werden.

Für den Entwurf des Dienstes ist es erforderlich, die Eigenschaften der vier Anwendungsklassen in Bezug auf Gruppenkommunikation näher zu betrachten. Tabelle 2.1 fasst die Gruppenkommunikationseigenschaften von Anwendungen zusammen, die von einem frühen und zeitgleichen Informationszugang profitieren. Zugleich wird aufgezeigt, in welchen Punkten sich die einzelnen Anwendungen unterscheiden und welche Schlussfolgerungen

sich daraus für die Anforderungen an eine verzögerungsfaire Nachrichtenauslieferung ergeben. Charakteristisch für alle Anwendungen ist, dass sie nicht auf lokale Teilnehmermengen beschränkt sind und daher von einer globalen Verteilung der Informationen profitieren. Ein Kriterium ist die **Anzahl an Informationsquellen**. Bei der Informationsverteilung ist nur eine Quelle involviert. Bei einigen anderen Anwendungen, die ebenfalls von einem verzögerungsfairen Dienst profitieren, sind mehrere unabhängige Quellen an einer Sitzung beteiligt. Bei Spielen sind dies z. B. die Spielentscheidungen jedes Spielers, die sich fair auf das Spiel auswirken sollen. Ein weiterer Unterschied zwischen den Anwendungen besteht in der **Nachrichtenhäufigkeit**, die bei Börsenkurstickern mit etwa 10 Nachrichten pro Sekunde relativ hoch, im Fall von Ad-hoc-Nachrichten mit ca. 50 pro Tag eher gering ausfällt. Die **Art der Nutzergruppe** gibt Auskunft, ob dem Informationsanbieter die Empfänger bekannt sind oder nicht. Aus der Art und **Anzahl der Nutzer** leiten sich Anforderungen an die Skalierbarkeit des Dienstes ab. Abgesehen von sehr speziellen Auktionen und Ereignissen handelt es sich bei diesen Diensten um eine potenziell sehr große Zahl von Teilnehmern. Dies geht in der Regel mit einer hohen **Abonnierfrequenz** einher. Hierunter ist die Häufigkeit des Nutzerwechsels zu verstehen.

2.4 Anwendungsbedingte Anforderungen bezüglich des Verzögerungsunterschiedes

Einen wichtigen Anhaltspunkt für anwendungsbedingte Anforderungen im Hinblick auf den Verzögerungsunterschied geben die Experimente über die menschliche Wahrnehmung von Jitter und Mediensynchronisierung in [Steinmetz 1996]. Von Versuchspersonen wurden miteinander in Bezug stehende Audio- und Videoströme mit unterschiedlich präsentierten Verzögerungsunterschieden auf Lippensynchronität beurteilt. Ab einem Jitter von $\pm$ 80 ms wurde von den Personen Asynchronität festgestellt. Erwähnt wird außerdem, dass bei einem eng gekoppelten Audiosignal (z. B. Stereo) der Jitter schon bei $\pm$ 11 μs, bei einem eng gekoppelten Audio/Video-Signal (z. B. Musik mit Darstellung der gespielten Note) ab $\pm$ 5 ms bemerkt wird.

Um einen Nutzen zu erzielen, muss bei den in Abschnitt 2.3 betrachteten Anwendungen außer der Wahrnehmung der Information auch auf diese reagiert werden, beispielsweise mit einer Aktientransaktion eines Teilnehmers auf eine Ad-hoc-Mitteilung. Die Betrachtung der Reaktionsdauer ist daher wichtiger als die der Jitter-Wahrnehmung.

Tabelle 2.2: Komponenten der Reaktionsdauer des Menschen aus [Klebelsberg 1982, Quelle: Wargo 1967]

Vorgang	Zeitdauer [ms]
Reizaufnahme im Rezeptor	1-40
Übertragung zum Kortex	1-100
Zentrale Vorgänge	70-300
Efferente Übertragung	10-20
Latenzzeit des Muskels	30-70

Die Reaktionsdauer des Menschen wurde aus unterschiedlichen Blickwinkeln ermittelt. Verkehrspsychologen und Sportwissenschaftler haben hierzu umfangreiche Untersuchungen durchgeführt. Die Reaktionsdauer des Menschen beinhaltet die Wahrnehmung des Reizes, die Erkennung der Gegebenheit, die Entscheidung über die auszuführende Aktion und die Umsetzung in die motorische Handlung. Mit Handlungsbeginn, erkennbar durch die erste motorische Reaktion, endet die Reaktionsdauer. Für Einfachreaktionen unter Laborbedingungen sind die in Tabelle 2.2 dargestellten Werte ermittelt worden [Klebelsberg 1982, Quelle: Wargo 1967]. Bei Einfachreaktionen handelt es sich um einfache motorische Handlungen als Antwort auf von den Versuchspersonen erwartete Signale, z. B. das Drücken einer Taste nach Aufleuchten einer grünen Lampe. Reaktionen auf optische Reize dauern etwas länger als Reaktionen auf akustische Reize, weil nach [Weineck 1994] die Umwandlung von Lichtenergie in neuronale Impulse, die von der Netzhaut des Auges ins Gehirn weitergeleitet werden, mindestens 30 ms mehr Zeit beansprucht als die entsprechende Umwandlung von Schallenergie, die in Form neuronaler Impulse dem auditiven System (Gehörsinn) übermittelt wird.

Sportwissenschaftliche Studien haben sich vor allem mit der Reaktion auf den Startschuss als akustischem Signal auseinandergesetzt. Die Untersuchungsergebnisse der einzelnen Autoren weisen beachtliche Unterschiede auf. Sie sind vor allem auf die Art der Messung der Reaktion zurückzuführen. Eine Übersicht kann [Grosser & Starischka 1998] entnommen werden. Die eindeutige Definition des Fehlstarts erlaubt jedoch eine zusammenfassende Auswertung der Untersuchungsergebnisse. Als Fehlstart wird gewertet, wenn die Summe der Horizontalkräfte, die auf den Startblock einwirken, früher als 100 ms nach dem Schuss 250 N übersteigt. Nach dieser Definition ließen sich bei den meisten Spitzensportlern Reaktionszeiten auf den Startschuss von 150 bis 200 ms feststellen.

Die holländische Auktion ähnelt der Reaktion auf Einzelereignisse unter Laborbedingungen. Hier wird die Schrittweite der Betragsminderung in der Regel im Voraus bekannt sein, so dass sich der Dienstnutzer schon im Vorfeld auf seine Reaktion einstellen kann. Das optische Signal kann mit einem akustischen Signal verbunden werden, um eine kürzere Reaktionsdauer zu erreichen.

Schon Ende des 19. Jahrhunderts, lange vor der Begründung der Informationstheorie durch Shannon [Shannon 1976], wurde die Dauer von Auswahlreaktionen gemessen. Bei Auswahlreaktionen gilt es, auf eine Menge von Signalen mit verschiedenen, den Signalen in bestimmter Weise zugeordneten motorischen Handlungen zu reagieren. Die Versuchsanordnungen sind in der Regel so aufgebaut, dass die Versuchspersonen auf Lampen, die in bestimmten Zeitintervallen aufleuchten, durch Drücken einer zugehörigen Taste antworten müssen. Die Anzahl der Lampen k wurde von 1 bis 10 variiert. Die Untersuchungen ergaben, dass die Reaktionszeit linear mit $log(k)$ wächst. Die in [Klebelsberg 1982, Quelle: Liedemit 1977] gezeigten Ergebnisse weisen eine Reaktionszeit bei 8 Wahlmöglichkeiten ($k = 3$) von ca. 1 s und bei 32 Wahlmöglichkeiten ($k = 5$) von ca. 1.8 s aus. Durch Training kann die Reaktionszeit weiter verkürzt werden, im letztgenannten Bereich auf ca. 1 s.

Des Weiteren ist bekannt, dass ein Erwachsener 200 bis 400 Wörter pro Minute liest [Klein 2000]. Die Reaktionszeit auf Ad-hoc-Meldungen wird dementsprechend nicht unter einer Sekunde liegen können. Für den zu erbringenden Dienst ergibt sich daher, dass ein Verzögerungsunterschied von bis zu 20 ms bei Einfachreaktionen und bis zu 100 ms bei Wahlreaktionen keine entscheidende Benachteiligung für den einzelnen Dienstnutzer (Mensch) bedeutet.

In diesem Kapitel wurde gezeigt, welche Anwendungen von einem verzögerungsfairen Dienst profitieren, welche Kriterien für die Zuordnung zu den einzelnen Anwendungsklassen entscheidend sind und welche Anforderungen an einen verzögerungsfairen Dienst sich daraus ableiten lassen. Kapitel 3 wird sich speziell Anwendungen mit unterschiedlichen Anforderungen an den Verzögerungsunterschied zwischen Empfängern bei der Gruppenkommunikation im Internet zuwenden. Dabei soll ein Messverfahren vorgestellt werden, das es erlaubt, die Verzögerungsunterschiede von Gruppenkommunikationsteilnehmern über das Internet zu ermitteln und die Frage zu erörtern, für welche Anwendungen eine Reduktion des Verzögerungsunterschiedes erforderlich ist.

Kapitel 3

Verzögerungsunterschiede zwischen Empfängern bei Multicast

Kapitel 2 zeigte Anwendungen mit unterschiedlichen Anforderungen an den tolerierbaren Verzögerungsunterschied zwischen Empfängern auf. Eine Betrachtung der Ursachen für Verzögerungsunterschiede lässt vermuten, dass in einem paketvermittelten Netz wie dem Internet Verzögerungsunterschiede zwischen Empfängern bei der Gruppenkommunikation auftreten. Unbekannt ist jedoch ihre Größe, so dass die Frage, für welche Anwendungen die Verzögerungsunterschiede akzeptabel sind und für welche Anwendungen eine Reduktion erforderlich ist, nicht beantwortet werden kann. Hierüber können nur Messungen Aufschluss geben, die jedoch wegen des Fehlens eines geeigneten Messverfahrens bisher nicht möglich waren. Es ist daher erforderlich, ein Messverfahren zu entwickeln, mit dessen Hilfe die Verzögerungsunterschiede von Gruppenkommunikationsteilnehmern über das Internet ermittelt werden können. Eine wesentliche Anforderung an dieses Messverfahren besteht darin, Verzögerungsunterschiede im Internet auch ohne Softwareinstallation bei einer großen Empfängermenge zu beobachten. Dadurch würden Dienstanbieter in die Lage versetzt, zu entscheiden, für welche Anwendungen das gegenwärtige Internet eine geeignete Infrastruktur für die Datenübertragung bietet und für welche Anwendungen Maßnahmen zur Reduktion der Verzögerungsunterschiede zwischen den Teilnehmern getroffen werden müssen, um eine faire Verteilung der Informationen zu erreichen.

In diesem Kapitel werden zunächst die Ursachen für Verzögerungsunterschiede dargelegt. Anschließend wird ein Messverfahren vorgestellt, mit dessen Hilfe die aktuelle Situation im Internet analysiert werden kann. Messergebnisse schließen das Kapitel ab.

3.1 Ursachen von Verzögerungsunterschieden

Im folgenden Abschnitt werden zunächst die Ursachen für Schwankungen der Verzögerung von Nachrichten auf Unicast Übertragungswegen dargelegt. Darauf aufbauend wird in Abschnitt 3.1.2 der Verzögerungsunterschied zwischen Teilnehmern von Gruppenkommunikation definiert.

3.1.1 Allgemeine Ursachen

Nachrichten erfahren auf ihrem Weg vom Sender zum Empfänger eine Verzögerung. Hierzu tragen verschiedene Ursachen bei, von der Ausbreitungsgeschwindigkeit des Signals auf dem Übertragungsmedium über die in paketvermittelten Netzen typischen Warteschlangen in den Netzwerkknoten bis hin zur Verarbeitungsdauer der Nachrichten in höheren Protokollschichten bzw. auf den Endsystemen. Verzögerungen entstehen praktisch in allen Schichten einer Kommunikationsbeziehung und sie wirken sich unterschiedlich stark auf die einzelnen Übertragungseinheiten aus. Als Folge davon kommt es bei zwei aufeinander folgenden Nachrichten zu unterschiedlichen Verzögerungen, deren Differenz auch als Jitter bezeichnet wird. Für diesen Begriff gibt es in der Literatur unterschiedliche Definitionen. In [Kalmanek & Kanakia 1990] wird damit die Büschelartigkeit des Verkehrs bezeichnet. Im Gegensatz zu dem für die Bestimmung der Übertragungsrate eines Datenstromes herangezogenen Intervall wird für die Bestimmung des Jitters ein wesentlich kleineres Intervall betrachtet und der Jitter als die maximale Anzahl von Paketen in diesem Intervall definiert. In [Ferrari 1990] wird der Begriff verwendet, um damit die Größe der Verzerrung des Verkehrsmusters durch das Netzwerk zu erfassen. Der Jitter ist danach die maximale Differenz zwischen den Verzögerungen, die zwei beliebige Pakete zwischen zwei bestimmten Endpunkten erfahren. [Zhang 1995] klassifiziert die beiden Jittertypen als raten- bzw. verzögerungs-basierten Jitter.

Die hier gewählte Definition lehnt sich an den Begriff des verzögerungs-basierten Jitter an, da für die Fairnessbetrachtungen die maximalen Abweichungen der Verzögerung von Interesse sind.

Definition 3.1 (Jitter)
Der Jitter von Nachrichten zwischen einem Sender und einem Empfänger ist die Differenz aus der maximalen und minimalen Verzögerung von Nachrichten innerhalb dieser Kommunikationsbeziehung.

In der Bitübertragungsschicht drückt sich der Jitter durch die Abweichung des Signals vom erwarteten Moment der Signalflanke bei der Signalrückgewinnung aus. Dies wird unter anderem verursacht durch Nebensprechen, Phasenverschiebungen durch thermisches Rauschen oder unterschiedliche Frequenz der Oszillatoren. In der Sicherungsschicht treten bei Medien, die sich mehrere Teilnehmer teilen, durch das Medienzugriffsverfahren unterschiedliche Verzögerungen pro Rahmen auf. Ein charakteristisches Beispiel ist das Verfahren *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)* des Standards IEEE 802.3 [IEEE 2002]. In paketvermittelten Netzen treten in der Vermittlungsschicht durch den unterschiedlichen Füllstand von Warteschlangen für die Weitervermittlung in den Netzwerkknoten und bei dynamischem Routing unterschiedliche Verzögerungen für die einzelnen Datagramme auf. Die Transportschicht trägt durch Übertragungswiederholungen bei verlorengegangenen oder beschädigten Paketen sowie durch die Fluss- und Staukontrolle zum Jitter bei. Auf den Endsystemen teilen sich in der Regel mehrere Anwendungen Prozessorzeit und andere Ressourcen, so dass es auch hier zu Jitter in der Verarbeitung und Darstellung der übertragenen Informationen kommt. Die Größenordung des Jitter ist abhängig von seiner Ursache. Entsprechend differenziert sind die Auswirkungen des Jitter auf der Anwendungsebene.

3.1.2 Ursachen von Verzögerungsunterschieden zwischen Teilnehmern

Für die Fairness von Multicast-Anwendungen ist die Betrachtung des Verzögerungsunterschiedes zwischen den Teilnehmern von Bedeutung, denn, anders als bei Unicast, sind mehrere, an verschiedenen Orten stationierte Teilnehmer involviert. Daher wird die Größe des Verzögerungsunterschiedes noch von weiteren Ursachen beeinflusst. Die Kopien der Nachrichten werden zu den Teilnehmern über Pfade verschiedener Länge geleitet. Das hat Unterschiede in der Ausbreitungszeit des Signals, in der Knotenzahl, in den Warteschlangen und somit auch in den Wartezeiten der Datagramm-Kopien in den Knoten zur Folge. Zudem sind die Knoten einschließlich der Endgeräte heterogen, woraus unterschiedliche Verarbeitungsleistungen resultieren.

Es sei S ein Sender und $\mathcal{R} = \{T_1, ..., T_j, ..., T_n\}$ die Menge der Teilnehmer einer Multicast-Gruppe. Der Sender sendet dieser Multicast-Gruppe die Nachrichten $N_1, ..., N_i, ...$ Die Verzögerung, die die Nachricht i auf dem Weg von S zu T_j erfährt, sei mit δ_{ij} bezeichnet.

Definition 3.2 (Verzögerungsunterschied zwischen Teilnehmern)
Als Verzögerungsunterschied zwischen Teilnehmern J wird die Differenz aus maximaler und minimaler Verzögerung der Kopien einer Nachricht auf dem Weg zu den Teilnehmern bezeichnet.

$$J_i = \max_j \{T_j \in \mathcal{R} : \delta_{ij}\} - \min_j \{T_j \in \mathcal{R} : \delta_{ij}\}$$

Die maximal erfahrene Verzögerung wird als Gesamtverzögerung der Nachricht bezeichnet.

Definition 3.3 (Gesamtverzögerung einer Nachricht)
Die Gesamtverzögerung δ_i einer Nachricht i ist das Maximum der Einzelverzögerungen, die die Kopien der Nachricht zwischen Sender und Teilnehmer erfahren.

$$\delta_i = \max_j \{T_j \in \mathcal{R} : \delta_{ij}\}$$

Die Definitionen verdeutlicht Abbildung 3.1. Die Kopien einer Gruppennachricht N_i erfahren eine unterschiedliche Verzögerung δ_{ij} auf dem Weg vom Sender zu den Teilnehmern. Die maximale Verzögerung, die die Nachrichtenkopie zu Teilnehmer T_n erfährt, entspricht der Gesamtverzögerung δ_i. Das grau hinterlegte Zeitintervall, begrenzt durch die minimale und maximale Verzögerung, stellt den Verzögerungsunterschied zwischen den Teilnehmern dar.

Der Verzögerungsunterschied der Gruppenkommunikationsnachrichten zu einer Teilnehmermenge schwankt ebenfalls. Beispielsweise sind die Wartezeiten von Kopien zweier aufeinanderfolgender Nachrichten in der Warteschlange eines Routers voneinander verschieden. Darüber hinaus können Teilnehmerwechsel zu Schwankungen des Verzögerungsunterschiedes führen, wenn sich dadurch die maximale oder minimale Pfadlänge ändert oder Routing-Änderungen ausgelöst werden.

3.2 Messtechnik

In der Literatur gibt es bereits zahlreiche Untersuchungen über Verzögerung, Jitter und Verlustrate bei Unicast-Verkehr [Paxson 1997b]. Die Beobachtung der Unicast-Einwegzeit von einem Sender zu einer Empfängermenge ergibt jedoch nur eine grobe Schätzung

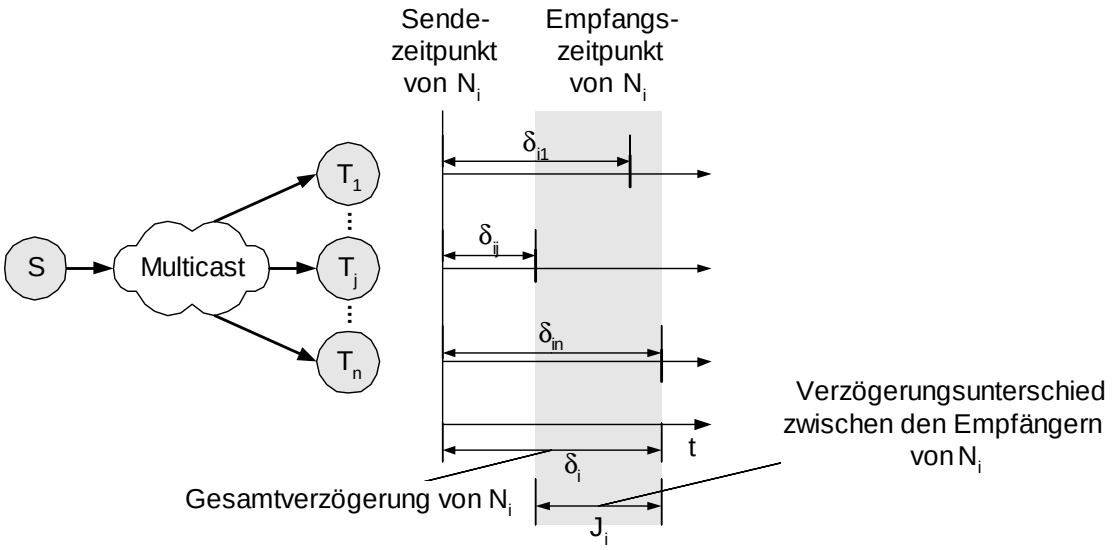


Abbildung 3.1: Empfangszeitpunkt einer Multicast-Nachricht und Verzögerungsunterschied zwischen Teilnehmern

des Verzögerungsunterschiedes zwischen den Teilnehmern. Die Ungenauigkeit der Methode ergibt sich zum Einen daraus, dass, im Gegensatz zu Multicast, die Pakete zu der Empfängermenge nur nacheinander versendet werden können und es sich daher nicht um denselben Beobachtungszeitpunkt handelt. Zum Anderen werden Multicast-Pakete von der Vermittlungsschicht im Allgemeinen mittels anderer Protokolle weitergeleitet als Unicast-Pakete. Das betrifft vor allem Routing-Protokolle für dünn besetzte Gruppen, d.h. räumlich verstreute Gruppen mit einer geringen Anzahl von Teilnehmern. Während das *Distance Vector Multicast Routing Protocol* [Waitzman et al. 1988] auf die Unicast-Routing-Information zurückgreift und dadurch ähnliche Pfade entstehen, hat das Routing anderer Protokolle, wie *Protocol Independent Multicast – Sparse Mode (PIM-SM)* [Estrin et al. 1997] und *Core Based Trees* [Ballardie et al. 1993] keine Gemeinsamkeiten mit Unicast-Routing. PIM kann sich sogar an die unterschiedlichen Anforderungen dicht und dünn besetzter Gruppen durch Wechsel zwischen quellenbasiertem Routing und einem gemeinsamen Routing-Baum anpassen, was zu unterschiedlichen Verzögerungen führt.

Die Anforderung an das Verfahren lautet deshalb, die Multicast-Protokolle mit einzubeziehen, auch wenn das Messen von Multicast-Verkehr einige Probleme in sich birgt. Zu den schon von Unicast her bekannten Messproblemen, wie z. B. nicht synchronisierte Uhren, kommen Multicast-typische Probleme hinzu. So muss die Messsoftware bei vielen Teilnehmern weltweit installiert werden, um eine große Zahl an potenziellen Teilnehmerorten

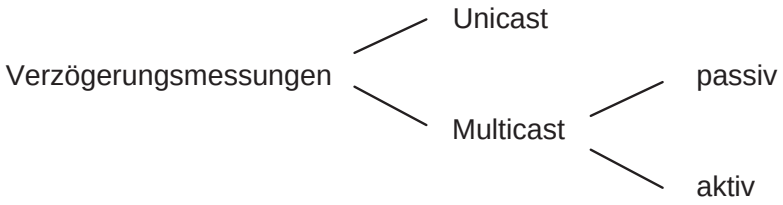


Abbildung 3.2: Einteilung der Messverfahren

einzubeziehen. Das verursacht bei einer großen Gruppe von Anwendern einen hohen administrativen Aufwand. Am besten können diese Probleme durch passives Messen umgangen werden. Bei einem passiven Verfahren werden Messdaten aus dem realen Datenverkehr extrahiert und mitprotokolliert, was die Untersuchung von existierenden Szenarien erlaubt. Bei den aktiven Verfahren werden hingegen periodisch Prüfpakete zur Bestimmung des Messwertes versandt, z. B. Testpakete für die Bestimmung der Paketverzögerung auf einem Pfad. Dadurch ist in der Regel eine spezifischere Datenerfassung als bei passiven Verfahren möglich. Andererseits beeinflussen Prüfpakete den realen Datenverkehr und somit auch die Messwerte.

3.2.1 Verwandte Arbeiten

Zur Überwachung des Multicast-Verkehrs wurden verschiedene Werkzeuge entwickelt, mit denen passive und aktive Verfahren realisiert werden (siehe Abbildung 3.2). Eine ausführliche Übersicht präsentieren [Sarac & Almeroth 2001]. Informationen über die Vermittlungsschicht von Multicast durch **passive Messungen** geben Werkzeuge wie *mtrace* oder *Monitor and Analysis of Traffic in Multicast Routers* (MANTRA) [Rajvaidya & Almeroth 2000]. *mtrace* ermittelt den umgekehrten Pfad einer Multicast-Gruppe vom Empfänger zum Sender und stellt Statistiken z. B. von Paketverlusten und Paketverzögerungen dar. MANTRA wertet Routing-Daten von einigen weltweit verteilten Multicast-Routern aus und kann daher Aussagen über den Status des Multicast-Netzes und der Multicast-Gruppen treffen. Weitere Werkzeuge geben Auskunft über die Teilnehmer einer Multicast-Sitzung (*mlisten* [Almeroth & Ammar 1996]), den Multicast-Baum (*mhealth* [Makofske & Almeroth 1999]) oder die Konfiguration eines Multicast-Routers (*mrinfo* [Fenner 1998]). Der *Network Node Manager Multicast* [Hewlett-Packard Company 2001] ist eine Erweiterung zu dem Produkt HP Open View und basiert auf dem *Simple Network Management Protocol (SNMP)*. Er erfragt die in den Routern verfügbaren Daten

der *Management Information Base (MIB)* und erzeugt Ansichten der Topologie und des Verkehrsaufkommens von Multicast. Das *Real-Time Transport Protocol (RTP)* bietet verschiedene statistische Informationen, die auf der Anwendungsschicht verfügbar sind (siehe Abschnitt 3.2.2). Auf ihnen basieren einige Werkzeuge, wie z. B. *RTPmon* [Bacher et al. 1996], das diese Informationen anzeigt. Ein weiteres Programm zum passiven Messen auf der Anwendungsschicht ist der *Session Directory Monitor* [Rajvaidya & Almeroth 2000]. Auf einer Anzahl von Multicast-Teilnehmern werden die Ankündigungen von Multicast-Sitzungen überwacht, woraus Schlüsse über die gegenwärtige Erreichbarkeit mittels Multicast gezogen werden.

Das *Multicast Beacon* [Chen et al. 2002] des US National Laboratory for Applied Network Research (NLANR) ist ein Beispiel für ein **aktives Messsystem** von Multicast-Netzparametern. Aktive Messprogramme senden periodisch Multicast-Pakete aus, die von allen an der Messung teilnehmenden Programmen über Multicast empfangen werden. Netzparameter wie Verzögerung, Jitter und Paketverlustrate werden von den Messprogrammen überwacht und periodisch einem Beacon-Server mitgeteilt, der die Ergebnisse zusammenfasst. Keines der Werkzeuge vermag jedoch Verzögerungsunterschiede zwischen Teilnehmern zu messen, weshalb hierfür ein neues Verfahren entwickelt werden musste.

Das im folgenden Abschnitt vorgestellte Verfahren erlaubt es, durch bloßes Mithören des Multicast-Verkehrs einer existierenden Gruppe die erforderlichen Daten zur Bestimmung des Verzögerungsunterschiedes zwischen den Teilnehmern zu ermitteln. Neben der Analyse von real existierenden Gruppen wird der Störeinfluss des Messverfahrens auf die Messung minimal gehalten. Das Messverfahren ist besonders für dünn besetzte Gruppen mit geringer Teilnehmerwechselrate geeignet.

3.2.2 Architektur

Die Architektur des Messsystems basiert auf dem Grundgedanken, die IP-Multicast-Protokolle so auszunutzen, dass durch passives Mitprotokollieren des Multicast-Verkehrs Daten zur Bestimmung des Verzögerungsunterschiedes zwischen Multicast-Teilnehmern ermittelt werden können (Abbildung 3.3). Dazu muss ein Protokoll gefunden werden, aus dessen Nachrichten sich die Einwegzeit der Kopien von Multicast-Paketen bestimmen lässt. Im Folgenden wird ein für diesen Zweck geeignetes Protokoll vorgestellt und aufgezeigt, wie aus den von diesem Protokoll versendeten Informationen der Verzögerungsunterschied zwischen Teilnehmern berechnet werden kann.

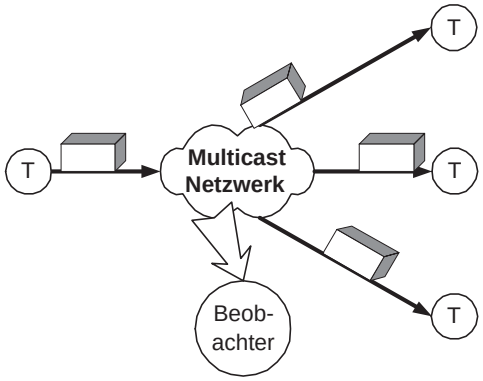


Abbildung 3.3: Grundgedanke des Messverfahrens: Passives Mitprotokollieren von Multicast-Verkehr

Zur Bestimmung der Verzögerungsunterschiede werden die Protokolle Real-Time Transport Protocol (RTP) und RTP Control Protocol (RTCP) [Schulzrinne et al. 1996] verwendet. RTP bietet Multimediaanwendungen nützliche Dienste an, wie die Identifikation der Nutzdaten, Kennzeichnung der Reihenfolge der Pakete, die Angabe von Zeitstempeln und die Kennzeichnung von logischen Einheiten (Frames) über Paketgrenzen hinweg. Darüber hinaus werden Konferenzanwendungen durch Transkodierung zwischen den Teilnehmern unterstützt, ein Grund für die weite Verbreitung des Protokolls für die Übertragung von Multimediadaten im Internet. Das zugehörige Steuerprotokoll RTCP dient als Feedback-Protokoll für die Anwendungen. Mittels RTCP übermitteln die Empfänger der multimedialen Datenströme statistische Informationen an die Sender, wie z. B. die Höhe des Paketverlustes, die höchste Sequenznummer der empfangenen Pakete und den Verzögerungsunterschied zwischen den eintreffenden Paketen. Der Sender verwendet diese Informationen, um die Qualität des Datenstromes an die Netzsituation anzupassen. Fünf Prozent der für eine Sitzung verwendeten Übertragungskapazität ist für solche Steuerdaten bestimmt.

Die Feedback-Informationen werden in Sender- und Empfänger-Reporten gesendet. Die Teilnehmer passen die Sendehäufigkeit der Reporte an die dafür bestimmte Übertragungskapazität und die Anzahl der Teilnehmer an. Wie in Abbildung 3.4 dargestellt, nehmen auf einen Sender-Report eines Senders (Sender 1) alle anderen Sender (Sender 2, 3) in ihrem nächsten Sender-Report Bezug. Die Reporte enthalten einen Header mit einem Bezeichner des Senders des Reports (*SSRC – Source ID*) und einen Report-Block *RB* für jeden Sender der Sitzung. Der Report-Block enthält einen Bezeichner des Senders, die statistischen Daten, den Zeitstempel des letzten von diesem Sender empfangenen Reports

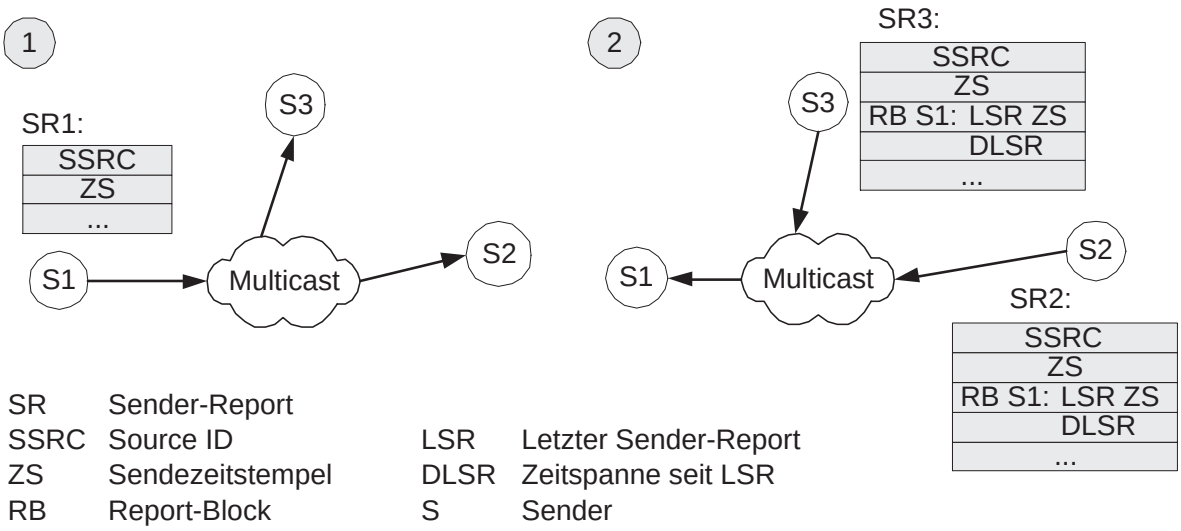


Abbildung 3.4: Berechnung des Verzögerungsunterschiedes von RTCP-Sender-Reporten. Sender 1 sendet seinen Sender-Report per Multicast an die Gruppe (1). Alle anderen Sender nehmen im Report-Block ihres Sender-Reports Bezug auf den Sender-Report von Sender 1 (2).

(*last sender report – LSR*) und die Zeitdauer seit dem Empfang dieses Reports (*delay since last sender report – DLSR*). Sender-Reporte enthalten zusätzlich eine Sender-Information, die einen 64-Bit-Zeitstempel im Format des *Network Time Protocol (NTP)* [Mills 1992] und eine Angabe über das von diesem Sender versendete Datenvolumen umfasst. Die Zeitstempel sind für Messungen der Umlaufzeit (*Round Trip Time – RTT*) der Pakete gedacht. Im Folgenden werden die Zeitstempel ausgenutzt, um die Einwegzeit (*One Trip Time – OTT*) der Pakete zu bestimmen und den Verzögerungsunterschied zwischen den Teilnehmern berechnen zu können.

(ZS^{SR_i}) bezeichne den Sende-Zeitstempel des Sender-Reports i. $LSR^{SR_j|RB_{S_x}}$ und $DLSR^{SR_j|RB_{S_x}}$ seien die im Report-Block für Sender S_x des Sender-Reports SR_j enthaltenen Angaben über den Sendezeitpunkt des letzten erhaltenen Sender-Reports von S_x sowie über die Zeitdauer seit dem letzten Empfang eines Sender-Reports von S_x.

Mit der Information aus allen Sender-Reporten, die auf denselben Sender-Report Bezug nehmen, d.h. für die der LSR Zeitstempel des Reportblocks für diesen Sender gleich der Sendezeit des betrachteten Sender-Reports ist ($LSR^{SR_j|RB_{S_x}} = ZS^{SR_i}$), kann die Einwegzeit der Kopien dieses Sender-Reports SR_i zu allen anderen Teilnehmern (Sendern) ermittelt werden.

Für die Berechnung der OTT eines zu betrachtenden Sender-Reports SR_i von Sender S_x zu S_y wird die Differenz aus dem Sendezeitpunkt (ZS^{SR_j}) des Bezug nehmenden Sender-Reports SR_j, der Zeitdauer ($DLSR^{SR_j|RB_{S_x}}$), die seit dem Empfang des betrachteten Sender-Reports vergangen ist, und dem Sendezeitpunkt (ZS^{SR_i}) des betrachteten Sender-Reports gebildet:

$$\forall j, LSR^{SR_j|RB_{S_x}} = ZS^{SR_i} \ : \ OTT^{SR_i}_{S_xS_y} = ZS^{SR_j} - DLSR^{SR_j|RB_{S_x}} - ZS^{SR_i}. \tag{3.1}$$

Nach Ausführung dieser Berechnung für Bezug nehmende Sender-Reporte kann der Verzögerungsunterschied zwischen den Teilnehmern von Sender-Report SR_i berechnet werden:

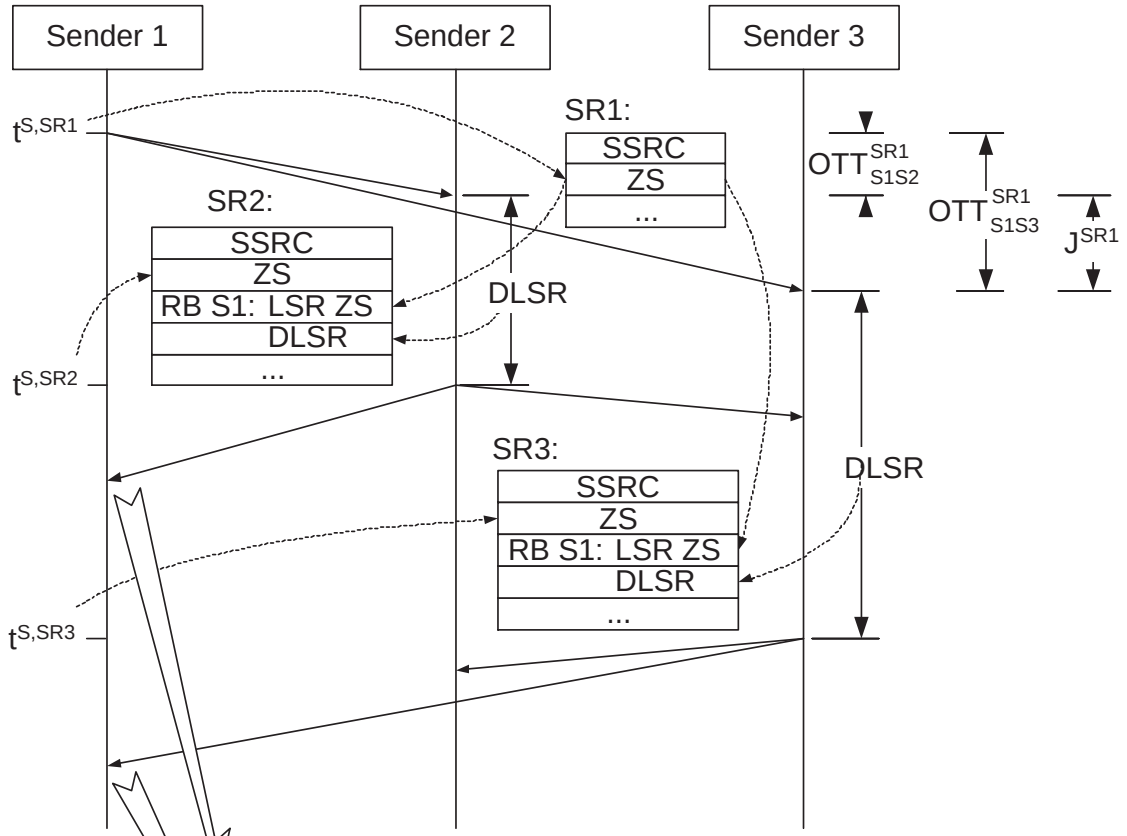
$$J^{SR_i} = \max_y \{OTT^{SR_i}_{S_xS_y}\} - \min_z \{OTT^{SR_i}_{S_xS_z}\} \tag{3.2}$$

In Abbildung 3.5 wird die Berechnung anhand eines Beispiels für 3 Sender im Zeitdiagramm gezeigt. Im Beispiel wird der Sender-Report $SR1$ betrachtet. Sender 2 und 3 nehmen in ihren Sender-Reporten $SR2$ und $SR3$ Bezug auf $SR1$, indem sie den Sende-Zeitstempel von $SR1$ im *Resource Block* für Sender 1 eintragen. Zum Sendezeitpunkt ihrer Bezug nehmenden Sender-Reporte werden die Zeitdauer seit Empfang von $SR1$ ($DLSR$) und ein Sende-Zeitstempel hinzugefügt. Aus diesen Angaben werden die Einwegzeiten der Kopien von $SR1$ sowie deren Verzögerungsunterschied zwischen Sender 2 und 3 berechnet.

RTCP-Empfänger-Reporte gleichen zwar den Sender-Reporten, enthalten jedoch nicht die Sender-Information, die den Zeitstempel des Sendezeitpunktes des Sender-Reports beinhaltet (ZS^{SR_j} in Gleichung 3.1). Deshalb können sie nicht zur Berechnung von OTTs herangezogen werden.

Folgende Einflüsse müssen bei der Evaluierung der Messergebnisse berücksichtigt werden:

- Sender-Reporte können, da sie ungesichert übertragen werden, verloren gehen. Liegen aus der definierten Empfängermenge nicht alle Sender-Reporte vor, die auf den Sender-Report, für den der Verzögerungsunterschied berechnet werden soll, Bezug nehmen, würde eine Einbeziehung des Verzögerungsunterschiedes dieses Sender-Reports das Ergebnis u. U. verfälschen. Das wäre der Fall, wenn die betrachtete Kopie eine maximale oder minimale Verzögerung erfahren hätte.
- Auch neu hinzukommende Sender oder die Sitzung verlassende Sender haben - messbedingt - einen Einfluss auf den Verzögerungsunterschied, da derartige Ereignisse zu einem längeren bzw. kürzeren Sender-Report führen. Das wirkt sich jedoch



$$OTT^{SR1}_{S1S2} = ZS^{SR2} - DLSR^{SR2/RB\ S1} - ZS^{SR1}$$

$$OTT^{SR1}_{S1S3} = ZS^{SR3} - DLSR^{SR3/RB\ S1} - ZS^{SR1}$$

$$J^{SR1} = \max_x \{OTT^{SR1}_{S1Sx}\} - \min_y \{OTT^{SR1}_{S1Sy}\}$$

S	Sender	RB	Report-Block
SR	Sender-Report	LSR	Letzter Sender-Report
SSRC	Source ID	DLSR	Zeitdauer seit LSR
ZS	Sendezeitstempel	OTT	Einwegzeit
		J	Verzögerungsunterschied

Abbildung 3.5: Berechnung des Verzögerungsunterschiedes von RTCP-Sender-Reporten. Sender 1 sendet seinen Sender-Report per Multicast an die Gruppe. Alle anderen Sender nehmen im Report-Block ihres Sender-Reports Bezug auf den Sender-Report von Sender 1.

nur bei Pfaden mit sehr kleiner Übertragungskapazität signifikant auf den Verzögerungsunterschied aus, so dass dies im Allgemeinen vernachlässigt werden kann. Einen signifikanten Einfluss auf den Verzögerungsunterschied haben Teilnehmerwechsel dagegen, wenn die OTTs zu diesen Teilnehmern im Bereich der maximalen oder minimalen OTT der gesamten Empfängermenge liegen.

- Die zur Berechnung verwendeten Zeitstempel werden von Uhren auf unterschiedlichen Computern generiert, die nicht notwendigerweise untereinander synchronisiert sind. Um exakte Messwerte zu erhalten, werden die Uhren daher auf ihre Genauigkeit untersucht und die Zeitstempel gegebenenfalls korrigiert, was im folgenden Abschnitt beschrieben wird.

3.2.3 Uhrenbeobachtung

Die Senderuhren, die die Zeitstempel für das RTCP-Protokoll erzeugen, sind im Allgemeinen nicht mit einer physikalischen Zeit t oder untereinander synchronisiert. Außerdem weisen sie unterschiedliche Ganggenauigkeiten auf. Beides äußert sich in differierenden Zeitangaben zum selben Ereignis.

Definition 3.4 (Zeitversatz)
Die Zeitdifferenz zwischen zwei Uhren zum Zeitpunkt m wird als Zeitversatz θ^m bezeichnet.

Eine Uhr generiert Zeitstempel ZS_e zu Ereignissen e. Mittels einer Referenzuhr, die eine viel höhere Genauigkeit als die zu vermessende Uhr besitzen muss, kann die Gangabweichung ρ einer Uhr bestimmt werden.

$$\rho = \left| \frac{ZS_m - ZS_n}{m - n} - 1 \right|$$

Zu diesem Zweck beobachtet ein Messrechner die Uhren der Sender der Gruppe über die Messzeitspanne. Um die aktuelle Uhrzeit der Sender zu erfahren, sendet er periodisch Pakete an die Sender. Jedes Paket erhält vier Zeitstempel (Abbildung 3.6): den Sendezeitpunkt ZS_1 des Messrechners, den Empfangszeitpunkt ZS_2 des Multicast-Teilnehmers (Sender), den Sendezeitpunkt ZS_3 des Multicast-Teilnehmers und den Empfangszeitpunkt

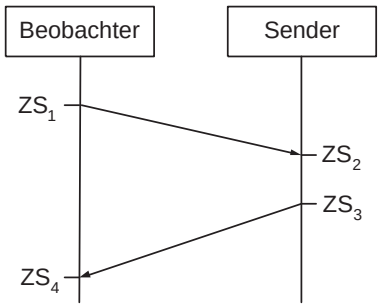


Abbildung 3.6: Beobachtung der Uhr eines Senders der Multicast-Gruppe

ZS_4 auf dem Messrechner. Aus den Zeitstempeln lässt sich die Umlaufzeit des Paketes (Round Trip Time, RTT) berechnen:

$$RTT = ZS_4 - ZS_1 - (ZS_3 - ZS_2) \tag{3.3}$$

Der Zeitversatz der Uhr zum Zeitpunkt ZS_4 des entfernten Rechners kann mittels der Zeitstempel wie folgt abgeschätzt werden:

$$\theta_{S_x}^{ZS_4} = ZS_4 - (ZS_3 + RTT/2) \tag{3.4}$$

Die Abschätzung des Zeitversatzes der Uhr des entfernten Senders gegenüber dem Messrechner kann über einen längeren Beobachtungszeitraum verbessert werden. Da die Pakete durch das Netzwerk unterschiedlich verzögert werden, kann aus den Zeitstempeln des Paketes mit der kleinsten Umlaufzeit während eines Beobachtungsabschnittes der Zeitversatz der Uhr des entfernten Rechners am genauesten bestimmt werden. Der Unterschied in der Verzögerung in Hin- und Rückrichtung des Paketes, verursacht durch Wartezeit in den Warteschlangen der Router, ist in diesem Fall am geringsten.

Zusätzlich zum Zeitversatz der Uhren wird die Gangabweichung der Uhren bestimmt, indem zwischen Messwerten mit kleinster Umlaufzeit interpoliert wird. Damit ist die Möglichkeit zur Korrektur der RTCP-Zeitstempel und somit auch der OTTs gegeben. Abweichungen höherer Ordnung, wie z. B. durch thermische Schwankungen des Quarzes verursachte Änderungen der Gangabweichung, werden für die Auswertung der Messergebnisse vernachlässigt.

Zur Berechnung der Gangabweichung der Uhren wird ein lineares Programm verwendet, das von Moon, Skelly and Towsley in [Moon et al. 1999] beschrieben ist. Dieser Algorithmus approximiert die Gangabweichung aus den minimalen Umlaufzeiten der Nachrichten

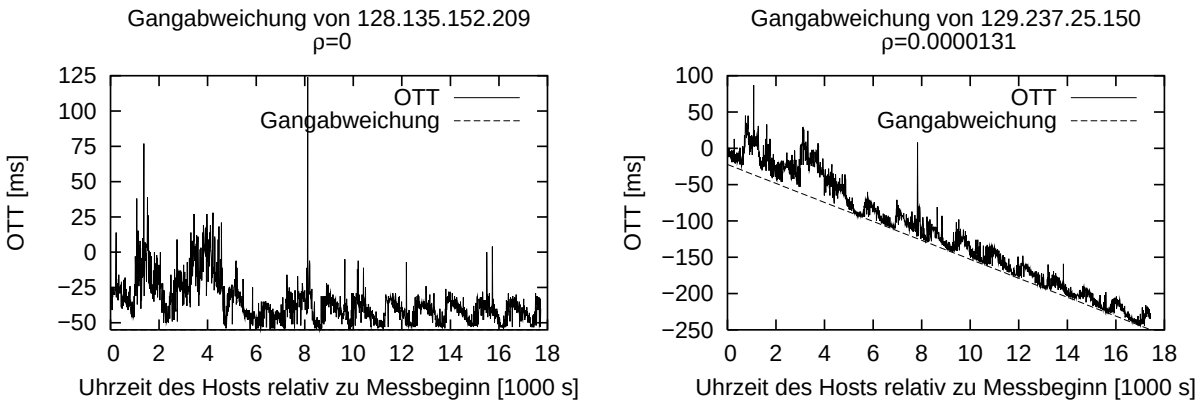


Abbildung 3.7: Beobachtungsergebnis bezüglich Gangabweichung bei Verwendung eines Uhrensynchronisationsprotokolls (links) und ohne dieses (rechts)

zur Uhrenbeobachtung. Die Auswertung der Gangabweichung gibt auch einen Hinweis darauf, ob der entsprechende entfernte Rechner ein Uhrensynchronisationsprotokoll verwendet oder nicht. Beträgt die approximierte Gangabweichung $\rho = 0$, wird ein Uhrensynchronisationsprotokoll verwendet. In Abbildung 3.7 wird das deutlich. Von Zeit zu Zeit wird dieselbe minimale Umlaufzeit für eine Zeitstempel-Nachricht erreicht.

Für Rechner, die ein Uhrensynchronisationsprotokoll verwenden, werden keine Korrekturen der Zeitstempel vorgenommen. In diesem Fall bestimmt die Güte der Uhrensynchronisation die Qualität der Messergebnisse. Nach mehrstündiger Verwendung des Uhrensynchronisationsprotokolls NTP geht Mills von einem Zeitversatz in der Größenordnung von einer Millisekunde aus [Mills 1992].

Die Daten von Hosts, die kein Uhrensynchronisationsprotokoll verwenden, können zur Berechnung des Verzögerungsunterschiedes herangezogen werden, wenn die Zeitstempel entsprechend des unter Einbeziehung der Gangabweichung berechneten Zeitversatzes korrigiert werden:

$$OTTcorr_{S_xS_y}^{SR_i} = ZS^{SR_j} + \theta_{S_y}^{ZS_{SR_j}} - DLSR^{SR_j|RB_{S_x}} - (ZS^{SR_i} + \theta_{S_x}^{ZS_{SR_i}}). \tag{3.5}$$

3.2.4 Konsistenzüberprüfung

Der Berechnung des Zeitversatzes der Uhren liegt die Annahme zu Grunde, dass das Paket in Richtung zum entfernten Rechner und in Gegenrichtung dieselbe Verzögerung erfährt. Dies ist im Internet jedoch nicht notwendigerweise gegeben. Zum Beispiel kann der Hin-

und Rückweg des Paketes teilweise über unterschiedliche Router führen. Paxson hat dies in [Paxson 1997a] untersucht. Von 11.339 mittels *traceroute* ermittelten Messpaaren lag bei 51 % keine Asymmetrie vor, bei 29 % war ein Hop und bei 20 % mehr als ein Hop unterschiedlich.

Eine ähnliche Untersuchung wurde in [Theilmann 2000] mittels von Web-Servern angebotenen Traceroute-Diensten durchgeführt. In einem 1999 ermittelten Datensatz mit 6960 in beiden Richtungen gemessenen Distanzen zwischen 119 Web-Servern wurde der mittlere Symmetriefehler über alle Paare von Mess-Servern bestimmt. Der Symmetriefehler betrug 2.0 Knoten mit einer Standardabweichung $\sigma = 2.3$ Knoten. Die Zeitabstände der Messungen für Hin- und Rückweg sind jedoch sehr groß, so dass hier auch dynamische Routing-Änderungen in die Messungen einfließen, die bei Uhrenbeobachtungen infolge des geringen Messabstandes eine eher untergeordnete Rolle spielen.

Mit einem passiven Messverfahren ist es nicht möglich, die Symmetrie des Hin- und Rückweges zu überprüfen, da kein standardmäßig installiertes Messprogramm für diesen Zweck vorliegt. Die Annahme gleicher Verzögerung auf Hin- und Rückweg kann daher zu Messungenauigkeiten führen. Folgende Tatsache wird jedoch zur Konsitenzüberprüfung der Abschätzung des Zeitversatzes der Uhren genutzt. Die Auswertung der Sender-Reporte ergibt ein Netz von Einwegzeiten zwischen den Sendern der Sitzung. Da die Verzögerung, die ein Paket erfährt, immer positiv ist, sollten nach erfolgreicher Korrektur der Zeitstempel keine negativen Einwegzeiten auftreten. Die Auswertung der Messergebnisse erfolgt nach vollständiger Erfassung der Messwerte, um die Genauigkeit der Verfahren zu erhöhen. Wird eine Online-Überwachung gewünscht, kann auch ein One-Pass-Algorithmus, wie z. B. die geregelte logische Uhr nach [Rab01], implementiert werden.

3.3 Ergebnisse

Zur Ermittlung konkreter Verzögerungsunterschiede wurde das Messverfahren auf das *Multicast Backbone (MBone)* angewandt. Während der Messungen protokollierte ein lokaler Host die RTCP-Sender-Reporte. Ein weiterer Host mit einer von einem auf dem Campus befindlichen Zeitserver synchronisierten Uhr beobachtete die Uhren der Sender der Gruppe.

Abbildung 3.8 zeigt Messergebnisse aus der Multicast-Gruppe *Places all over the world*, über die Videoströme geringer Datenrate (ca. 15-20 kbit/s) verteilt werden. Die Gruppe

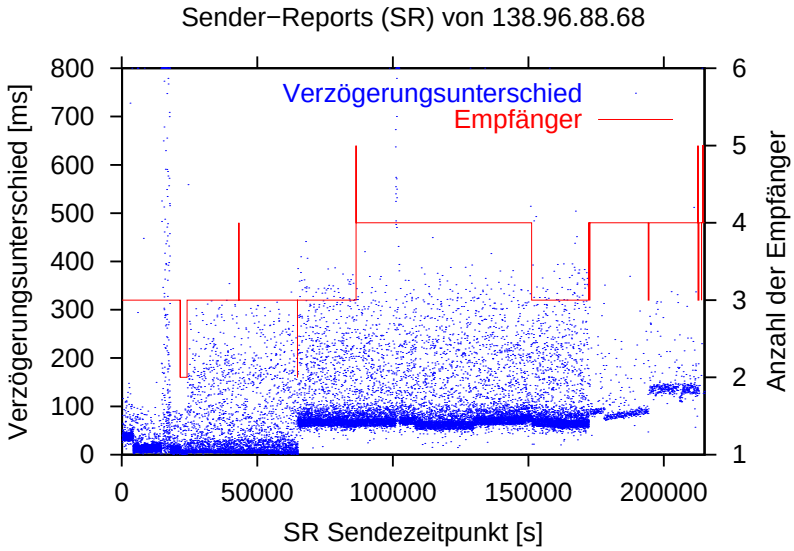


Abbildung 3.8: Verzögerungsunterschied zwischen Teilnehmern im MBone

wurde vom 20. bis 22. April 2002 60 Stunden beobachtet. In der Gruppe befanden sich Sender aus Deutschland, Frankreich, Südkorea, der Slowakei und den USA. In der Abbildung ist der Verzögerungsunterschied zwischen den Teilnehmern von Sender-Reporten des Senders 138.96.88.68 aus Frankreich dargestellt. Die Zeitstempel der Sender, die kein Uhrensynchronisationsprotokoll verwenden, wurden entsprechend den vorgestellten Algorithmen korrigiert. In der Darstellung im Punktediagramm ist anhand der Schwärzung gut zu erkennen, welche Messwerte besonders häufig auftreten. Außerdem ist die Anzahl der weiteren Sender während des Beobachtungsintervalls abgetragen. Die Abhängigkeit des Verzögerungsunterschiedes bei Wechsel der Teilnehmerzusammensetzung ist deutlich sichtbar. Zu Beginn empfangen nur Sender in Deutschland den Videostrom aus Frankreich, so dass der Verzögerungsunterschied zwischen ihnen gering ist. 64814 s nach Messbeginn kommt ein Host in den USA hinzu, der die RTCP-Sender-Reporte mit wesentlich höher Verzögerung erhält als die Sender in Deutschland. Dadurch erhöht sich der Verzögerungsunterschied signifikant. Ab 172197 s ist auch ein Host in Südkorea Gruppenmitglied und trägt zur maximalen OTT bei. Gleichzeitig tritt jedoch der Host aus den USA aus der Gruppe aus, so dass sich der Verzögerungsunterschied nur geringfügig ändert. Ab 194471 s kommt dieser Host wieder hinzu und sorgt für einen signifikanten Anstieg des Verzögerungsunterschiedes. Zwischenzeitlich ändert sich die Zusammensetzung der Teilnehmer, jedoch ohne den Verzögerungsunterschied signifikant zu beeinflussen.

Abbildung 3.9 stellt die Messergebnisse einer 48-Stunden-Messung vom 13. bis 14. Januar

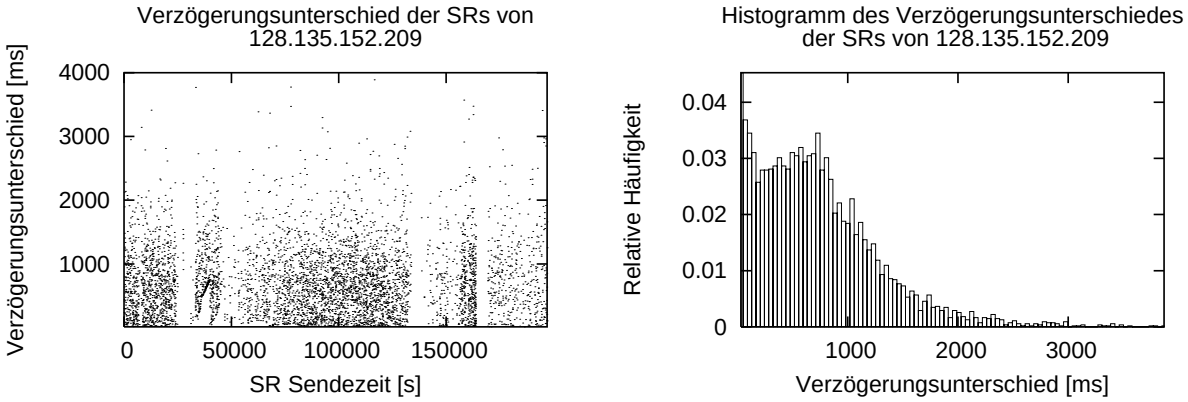


Abbildung 3.9: Verzögerungsunterschied zwischen den Teilnehmern der Multicast-Gruppe Access Grid Lobby (links) und dessen Histogramm (rechts)

2002 dar. Die beobachtete Gruppe *Access Grid Lobby* (224.2.177.155/55524) gehört zum Access-Grid-Videokonferenzsystem. Über diese Gruppe werden Videoströme von Empfangsräumen für Multimediakonferenzen verteilt. Für die Berechnung des Verzögerungsunterschiedes wurden nur Hosts berücksichtigt, die während des gesamten Messzeitraumes Teilnehmer der Sitzung waren. Dadurch ist der Verzögerungsunterschied über den gesamten Messzeitraum vergleichbar. Die Gruppengröße variierte während des Messzeitraumes zwischen 14 und 20 Sendern. Sechs Sender reagierten nicht auf die zur Uhrenbeobachtung notwendigen ICMP-Nachrichten und wurden, da keine Aussage über die Ganggenauigkeit der Uhren getroffen werden konnte, von der Berechnung ausgeschlossen. Acht Hosts waren während des gesamten Messzeitraumes aktiv und wurden in die Berechnung einbezogen. Außer einem Host in Großbritannien befanden sich alle Hosts in den USA. Drei Hosts (129.237.25.150, 193.62.114.7 und 204.121.50.17) sandten Sender-Reporte, enthielten aber keine Referenz auf die Sender-Reporte der anderen Sender. Für den in der Abbildung 3.9 dargestellten Verzögerungsunterschied der Sender-Reporte von Host 128.135.152.209 wurden daher 4 Hosts in die Berechnung des Verzögerungsunterschiedes einbezogen. Einflüsse durch Änderung der Paketgröße der Sender-Reporte bei Teilnehmerwechsel wurden ignoriert, da die Sitzung eine Übertragungskapazität von 4 bis 8 Mbit/s beanspruchte. Der Median dieses Verzögerungsunterschiedes betrug 641 ms und variierte zwischen 13 und 3890 ms. Das Histogramm des Verzögerungsunterschiedes ist in Abbildung 3.9 (rechts) dargestellt. Die Auswertung der Verzögerungsunterschiede ergab, dass die höchsten Einwegzeiten hauptsächlich zu Host 129.237.25.85 auftraten. Dies deutet auf spezifische Ursachen für Verzögerungen auf dem Übertragungsabschnitt zu diesem Host hin, wie z. B.

eine Firewall auf Anwendungsschicht.

Bei der Interpretation der Ergebnisse ist zu beachten, dass das vorgestellte Messverfahren nicht zwischen Verzögerungen im Netz und auf den Endgeräten unterscheiden kann. Da die im Rahmen dieser Arbeit entworfenen Protokolle auf die Anwendungsschicht abzielen, gibt das Messverfahren dennoch einen zweckmäßigen Einblick in die Größenordnung der anzutreffenden Verzögerungsunterschiede im Internet. Kontrollmessungen im *Local Area Network – LAN* und auf Endsystemen (vgl. [Klöcking & Rothermel 2002]) ergaben, dass der Einfluss der Endgeräte auf die Verzögerungen zwar gering, jedoch nicht vernachlässigbar ist.

Definitive Aussagen über einen ausschließlich vom Netzwerk verursachten Verzögerungsunterschied zwischen Teilnehmern können durch Betrachtung der Minima innerhalb eines größeren, z. B. durch fehlenden Teilnehmerwechsel gekennzeichneten Zeitintervalls gemacht werden.

Verzögerungsunterschiede zwischen Teilnehmern in der Größenordnung von 100 ms bis hin zum einstelligen Sekundenbereich konnten bereits bei kleineren Gruppen nachgewiesen werden. Dies ist für viele Anwendungen (siehe Kapitel 2) nicht akzeptabel. Daher werden im folgenden Kapitel Verfahren vorgestellt, die eine Reduktion der Verzögerungsunterschiede zwischen Teilnehmern ermöglichen. Die Verfahren zielen vor allem auf Ursachen von Verzögerungsunterschieden ab, die sich – wie die durch Teilnehmerwechsel verursachte unterschiedliche Pfadlänge – als signifikant erwiesen haben.

Kapitel 4

Verfahren zur Reduktion von Verzögerungsunterschieden zwischen Teilnehmern von Gruppenkommunikation

Die in Kapitel 3 durchgeführten Messungen machen deutlich, dass die Übertragung per Multicast bereits bei Nachrichtengrößen kleiner als 1500 Byte erhebliche Verzögerungsunterschiede aufweist. Für größere Nachrichten ist ein noch größerer Verzögerungsunterschied zu erwarten (vgl. hierzu auch die Berechnungen in Kapitel 5). Die Verzögerungsunterschiede liegen dabei deutlich über denen, die von den in Kapitel 2 beschriebenen Anwendungen toleriert werden. Ohne weitere Maßnahmen kann daher für solche Anwendungen keine geeignete Dienstqualität mittels Multicast-Übertragung bereitgestellt werden. Für kooperative Anwendungen existieren bereits mehrere Verfahren zur Synchronisation von multimedialen Datenströmen [Helbig 1996]. Diese Lösungen lassen sich jedoch nicht auf die hier betrachteten Anwendungen übertragen, da für sie Sicherheitsaspekte nicht relevant sind.

In diesem Kapitel wird zuerst der Lösungsraum für die Reduktion von Verzögerungsunterschieden zwischen Teilnehmern von Gruppenkommunikation aufgespannt. Anschließend werden verwandte Arbeiten dargestellt und in diesen Lösungsraum eingeordnet. Danach werden die Anforderungen an einen verzögerungsfairen Multicast-Dienst für nicht-

kooperative Echtzeitanwendungen zur Informationsverteilung erarbeitet und die Eigenschaften des Internets in einem Systemmodell abstrahiert. Es folgt die Vorstellung des verzögerungsfairen Multicast-Dienstes zur Informationsverteilung für nicht-kooperative Echtzeitanwendungen. Zur Optimierung der Gesamtverzögerung der Nachrichten wird ein Feedback-Mechanismus vorgestellt. Am Ende des Kapitels wird beschrieben, wie die Lösung auf andere nicht-kooperative Echtzeitanwendungen (Informationsempfang, diskreter und kontinuierlicher Informationsaustausch) übertragen werden kann.

4.1 Lösungsraum

Das Grundprinzip aller denkbaren Verfahren zur Erreichung von Fairness bezüglich des Verzögerungsunterschiedes besteht darin, die Kopien der zu verteilenden Nachrichten individuell so zu verzögern, dass sie gemeinsam mit der Kopie, die die größte Verzögerung erfährt, den Teilnehmern zugänglich werden. Dieses Ziel kann auf unterschiedlichen Wegen erreicht werden, die die Güte der Lösung beeinflussen. Die folgenden Kriterien dienen zur Klassifikation der Lösungswege:

1. Abstraktionsebene für die Realisierung der Mechanismen

   Ursachen von Verzögerungsunterschieden können, wie im Abschnitt 3.1 erläutert, in allen Schichten des Kommunikationsprotokoll-Stack auftreten. Dementsprechend können auch Lösungen zur Minimierung von Verzögerungsunterschieden in allen Schichten gefunden werden. Mit Verfahren der Vermittlungsschicht lasssen sich beispielsweise Änderungen der Pfadlänge durch Teilnehmerwechsel behandeln, mit Verfahren der Transportschicht durch Übertragungswiederholung hervorgerufene Verzögerungsunterschiede. Im Allgemeinen kann das Verfahren einer Schicht alle Ursachen dieser Schicht und der darunter liegenden Schichten ausgleichen. Da wesentliche Verzögerungsunterschiede erst auf der Vermittlungsschicht und höheren Schichten entstehen, ist es ausreichend, Verfahren der Vermittlungsschicht, der Transportschicht und der Anwendungsschicht zu betrachten.

2. Adaptivität an Gesamtverzögerung

   Parallel zur Minimierung des Verzögerungsunterschiedes zwischen den Empfängern streben die Anwendungen häufig eine minimale Gesamtverzögerung der Nachrichten an. Die Gesamtverzögerung der Nachrichten ändert sich zum Beispiel bei Wechsel von Teilnehmern, die zur maximalen Verzögerung von Nachrichten beitragen, oder durch Jitter der Nachrichten zu dem Teilnehmer, zu dem die Nachrichten eine maximale Verzögerung erfahren. Die Verfahren bieten hier einen unterschiedlich hohen Grad an Dynamik, auf die Änderung der Gesamtverzögerung zu reagieren.

3. Sicherheit

   Das wesentliche, aus dem Fairness-Aspekt sich ergebende Unterscheidungsmerkmal gegenüber Verfahren für kooperative Anwendungen, ist die Sicherheit. Sie ist notwendig, damit das Verfahren Fairness garantieren kann. Eine Lösung ist dabei umso sicherer, je weniger Elementen der Lösung Vertrauen bekundet werden muss. Es lassen sich Lösungen unterscheiden, bei denen allen Beteiligten Vertrauen entgegengebracht wird und Lösungen, bei denen das Vertrauen auf die Vermittlungsschicht des Kernnetzwerkes oder auf einzelne Systeme außerhalb des Kernnetzwerkes beschränkt ist.

   Äußerst geringe Verzögerungsunterschiede bei der Auslieferung der Informationen sind erreichbar, wenn die Nachrichtenverzögerung zu jedem Teilnehmer eindeutig bestimmt ist und die Informationen zeitlich gestaffelt so versendet werden, dass sie zum selben Zeitpunkt bei den Empfängern eintreffen. Eine direkte Bestimmung der Verzögerung ist jedoch nur möglich, wenn Vertrauen in die Teilnehmer besteht, da nur dann vertrauenswürdige Uhren beim Teilnehmer zur Verfügung stehen. Sind keine vertrauenswürdigen Uhren verfügbar, kann die Verzögerung zwischen zwei Knoten auch über die Bestimmung der Umlaufzeit geschätzt werden. Die Umlaufzeit ergibt sich aus der Differenz von Empfangszeit $T4$ und Sendezeit $T1$ einer Nachricht abzüglich der Verweilzeit $T3 - T2$ auf dem entfernten Knoten, die hier der Einfachheit halber mit $T3 - T2 = 0$ angenommen wird (siehe Abbildung 4.1). Unter der Annahme, dass die Verzögerung in beiden Richtungen zwischen den zwei Knoten gleich ist, ergibt die halbe Umlaufzeit $(T4 - T1)/2$ die Verzögerung zwischen Knoten 1 und 2. Ein Angreifer kann diese Annahme ausnutzen und die Verzögerung zu seinem Knoten größer erscheinen lassen. Führt er eine künstliche Verzögerung Δ ein, berechnet der Beobachter die Verzögerung als $(T4' - T1)/2$ mit $T4' = T4 + \Delta$ als

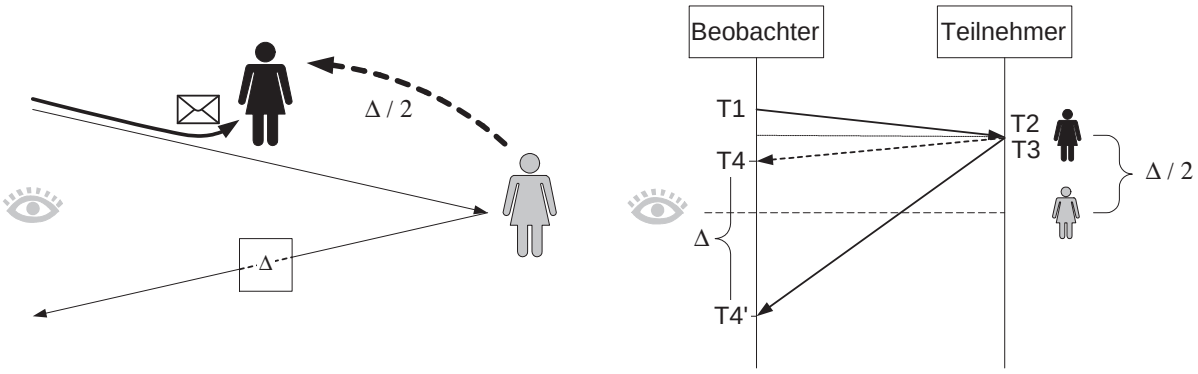


Abbildung 4.1: Ein Angreifer gelangt früher an die Nachricht, indem er den Eindruck erweckt, dass an ihn gerichtete Nachrichten eine hohe Verzögerung erfahren.

$(T4 + \Delta - T1)/2$. Es ergibt sich dementsprechend eine um $\Delta/2$ größere Verzögerung zwischen dem Knoten des Beobachters und dem Knoten des Angreifers aus der Sicht des Beobachters. Für die Anwendung der Nachrichtenauslieferung platziert der Angreifer die zusätzliche Verzögerung in Richtung vom Angreifer zum Beobachter, so dass er die Daten um $\Delta/2$ früher erhalten kann. Abbildung 4.1 illustriert diesen Sachverhalt. Während der Beobachter den Teilnehmer an der grau markierten Stelle wähnt, konnte der Teilnehmer die Nachricht schon an der schwarz markierten Stelle empfangen. Auch für eine Schätzung der Verzögerung im Zugangsnetz ist daher Vertrauen in die Teilnehmer notwendig.

Aus den Aspekten Abstraktionsebene, Adaptivität an Gesamtverzögerung und Sicherheit lässt sich der in Abbildung 4.2 dargestellte Lösungsraum aufspannen. Lösungen zur Behandlung von kooperativen Anwendungen befinden sich hierbei auf der durch die Achsen Abstraktionsebene und Adaptivität gebildeten Fläche, da Sicherheitsaspekte bezüglich des Verzögerungsunterschiedes nicht betrachtet werden. Eine ausführliche Klassifizierung von Verfahren zur Synchronisation von multimedialen Datenströmen befindet sich in [Helbig 1996].

4.2 Anforderungen an einen verzögerungsfairen Dienst

In diesem Abschnitt werden Anforderungen an einen verzögerungsfairen Dienst definiert, die als Kriterien für die Entwicklung geeigneter Methoden zur Bearbeitung des Problems

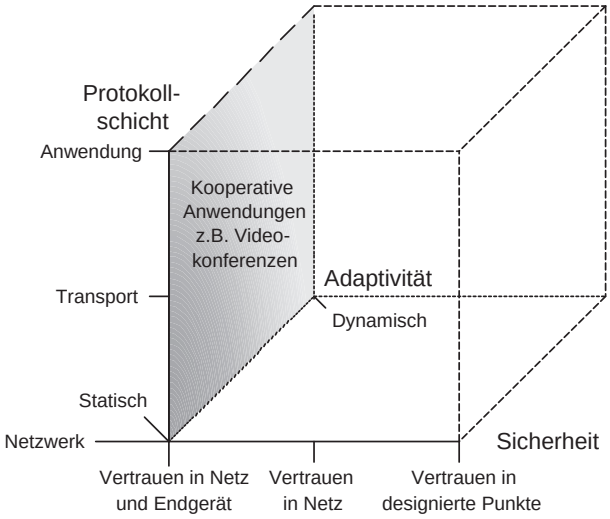


Abbildung 4.2: Lösungsraum für Verfahren zur Reduktion von Verzögerungsunterschieden

dienen. Dem Anwender können so optimal an sein Problem angepasste Verfahren offeriert werden. Ein verzögerungsfairer Dienst soll folgende Forderungen erfüllen:

1. Minimierung des Verzögerungsunterschiedes

   Der verzögerungsfaire Dienst soll möglichst alle durch verschiedene Ursachen hervorgerufenen Verzögerungsunterschiede ausgleichen.

2. Fairness über alle Teilnehmer

   Da das Optimierungskriterium des individuellen Teilnehmers – ein schneller Zugang zu den Informationen – dem globalen Optimierungskriterium der Fairness für alle Teilnehmer – gegenübersteht, soll der Dienst in der Lage sein, Fairness gegenüber unkooperativen Teilnehmern durchzusetzen.

3. Minimierung der Gesamtverzögerung

   Anwendungen, die eine Kombination der Klassen 1 und 2 darstellen, sowie Anwendungen der Klasse 4 profitieren von einer minimalen Gesamtverzögerung. Daher soll der Dienst optional auch in dieser Hinsicht optimiert werden.

4. Toleranz gegenüber Datenratenanforderung und Empfängerzusammensetzung

   Informationsverteilung stellt höchst unterschiedliche Anforderungen an die Übertragungskapazität. Die Lösung soll einen weiten Bereich von vereinzelten Nachrichten

bis zu isochronen Datenströmen unterstützen. Die Empfängerzusammensetzung ist beliebig und von dem Sender nicht beeinflussbar.

5. Unabhängigkeit von Netztechnologien

   Um einer Vielzahl von Teilnehmern die Möglichkeit des Informationsempfanges zu bieten, ist ein globaler, von heterogenen Netztechnologien unabhängiger Dienst erforderlich.

6. Geringer Nachrichtenaufwand

   Informationsverteildienste transportieren häufig ein hohes Datenvolumen zu einer großen Empfängermenge. Der Dienst soll daher eine effektive, ressourcenschonende Versendung von Informationen ermöglichen.

4.3 Verwandte Arbeiten

Verwandte Arbeiten zur Erzielung von Fairness in Bezug auf den Verzögerungsunterschied zwischen Teilnehmern der Gruppenkommunikation in nichtkooperativen Anwendungen werden anhand des in Abschnitt 4.1 aufgespannten Lösungsraumes eingeordnet und im Hinblick auf die Berücksichtigung von Sicherheitsaspekten erläutert. Es werden Verfahren unterschieden, bei denen Netz und Endgeräten, nur dem Netz oder nur designierten Punkten Vertrauen entgegengebracht wird.

4.3.1 Vertrauen in Netz und Endgeräte

Pulido und Lin stellen in [Pulido & Lin 1998] das Konzept *Simultaneous Multicast* vor. Unter Verwendung des empfängerorientierten Ressourcen-Reservierungsprotokolls RSVP wird in diesem Verfahren die Übertragungskapazität vom Empfänger so reserviert, das sich die Verzögerungsunterschiede zu den einzelnen Empfängern ausgleichen. Der Empfänger berechnet die zu reservierende Übertragungskapazität, indem er die zu übermittelnde Nachrichtengröße durch die Differenz aus maximaler Verzögerung der Nachricht und Verzögerung zu sich selbst dividiert. Das Verfahren baut damit im Prinzip auf dem Ausgleich der Serialisierungsverzögerung der Nachrichten auf den einzelnen Übertragungsabschnitten auf.

Der Sender erfasst die aktuelle Gesamtverzögerung über ein Feedback-Protokoll und stellt sie den Teilnehmern zur Berechnung der optimalen Reservierung zur Verfügung. Durch die Verwendung von Ressourcen-Reservierung in den Netzknoten ist das Verfahren der Vermittlungsschicht zuzuordnen.

4.3.2 Vertrauen in das Netz

In der Literatur finden sich zur Minimierung des Verzögerungsunterschiedes zwischen Empfängern hauptsächlich Verfahren, die einen speziellen Aufbau des Multicast-Routing-Baumes beschreiben. Diese Verfahren zielen auf die Reduktion von Verzögerungsunterschieden ab, die durch die verschiedenen Pfadlängen zu den Teilnehmern hervorgerufen werden. Die Verfahren können so verwendet werden, dass bei Teilnehmerwechsel der optimale Aufbau des Multicast-Baumes neu bestimmt wird.

Die verzögerungsfaire Optimierung des Multicast-Baumes stellt ein NP-vollständiges Problem dar [Rouskas & Baldine 1997]. Rouskas und Baldine geben hierfür einen heuristischen Algorithmus an, der in polynomieller Zeit zu einer suboptimalen, aber guten Lösung führt. Haberman und Rouskas präsentieren einen Algorithmus, der zusätzlich noch Kosten berücksichtigt [Haberman & Rouskas 1996]. Die Kosten werden zur gleichzeitigen Minimierung der Gesamtverzögerung verwendet. Hierfür werden weitere Optimierungsverfahren vorgeschlagen. [Wensheng & Zemin 1999] verwenden die Heuristik von [Rouskas & Baldine 1997] und minimieren die Kosten mittels eines neuronalen Netzes des Hopfield-Typs. [Sun & Langendoerfer 1999] benutzen einen genetischen Algorithmus zur Erzeugung eines kosten- und gesamtverzögerungsoptimierten sowie verzögerungsfairen Multicast-Baumes. Die Leistungsfähigkeit des Algorithmus ist um so größer, je kleiner die Zahl der Empfänger im Vergleich zur Gesamtzahl der Knoten ist.

Ein weiteres Verfahren der Vermittlungsschicht wird in [Ge et al. 1999] vorgestellt. Mit diesem Verfahren wird das Ziel verfolgt, die Pakete in den Routern so zu verzögern, dass sie zur selben Zeit die Endgeräte erreichen. Das Verfahren baut dabei auf einem verzögerungsfairen Multicast-Baum auf. Zusätzlich wird die Annahme getroffen, dass die maximalen Zeiten für das Scheduling der Pakete in den Routern bekannt sind. Die Router werden aus einem Verzögerungselement R und einem Scheduler S für jeden ausgehenden Übertragungsabschnitt modelliert. Das Verzögerungselement R_k des Knotens k verzögert die Pakete, bis sie für das Scheduling in Frage kommen. Die Verzögerung berechnet sich aus der Summe

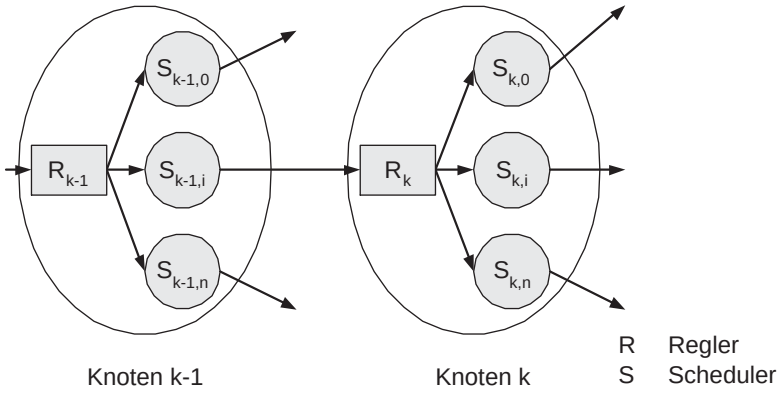


Abbildung 4.3: Routermodell des Packet-eligible-time-Algorithmus

- der Differenz aus der maximalen Verzögerung des i. Scheduler des Knotens $k-1$ bezeichnet mit $(S_{k-1,i})$ und der Verzögerung, die das Paket im Scheduler $S_{k-1,i}$ erfahren hat,

- der Differenz aus der maximalen Scheduler-Verzögerung über alle Scheduler von Knoten $k-1$ und der maximalen Scheduler-Verzögerung von Scheduler $S_{k-1,i}$ sowie

- der Differenz aus der maximalen Pfadlänge zu den Blattknoten über alle ausgehenden Übertragungsabschnitte des Routers $k-1$ und der maximalen Verzögerung des Pfades zu den Blattknoten beginnend mit dem Übertragungsabschnitt des Knotens k.

Die Verzögerung, die ein Paket durch den Scheduler erfährt, wird über im Paket-Header abgelegte Zeitstempel gemessen. Die maximale Verzögerung und die obere Schranke der Verzögerung des Schedulers $S_{k-1,i}$ werden vom Knoten $k-1$ zu Knoten k weitergegeben. Die Pfadverzögerung wird über ein separates Protokoll bestimmt und bei Änderungen der Gruppenmitgliedschaft aktualisiert.

Eine praktische Anwendung findet der Algorithmus in [Dongyan et al. 1999]. Hier wird das Problem der Echtzeitkommunikation mittels Multicast während der Verbindungsweiterreichung in zellularen Mobilfunknetzen adressiert. Die Verzögerungen und Verzögerungsunterschiede der Pakete werden als ortsabhängige Dienstqualitätsparameter betrachtet. Um zu verhindern, dass während einer Verbindungsweiterreichung ein mobiles Endgerät Paketverluste erfährt oder dass Pakete dupliziert werden, muss der Verzögerungsunterschied

zwischen den Zellen minimiert werden. Zum Einen minimiert ein erweiterter Paketscheduler den Verzögerungsunterschied innerhalb der Zelle, indem er die Paketberechtigungszeiten bei den zum Scheduling in Frage kommenden Paketen berücksichtigt. Zum Anderen wird der Verzögerungsunterschied zwischen den Zellen minimiert, indem den ausgehenden Übertragungsabschnitten, die mehr als einen Folgeknoten im Multicast-Baum besitzen, eine zusätzliche Wartezeit zugewiesen wird. Über ein Protokoll werden die Verzögerungen aller ausgehenden Übertragungsabschnitte einer Koppeleinrichtung ausgeglichen.

4.3.3 Vertrauen in designierte Punkte

In [Maxemchuk & Shur 2001] wird das Transportprotokoll *Timed Reliable Multicast Protocol (TRMP)* für einen verteilten Aktienmarkt vorgestellt. Das Protokoll kombiniert die Auslieferung der Nachrichten mit der Aufgabe der Transportschicht, die Auslieferung ohne Übertragungsfehler zu garantieren. Die Garantien werden durch Anordnung der Empfänger in drei Hierarchiestufen erreicht, wobei sich in den ersten zwei Hierarchiestufen vertrauenswürdige Empfänger befinden, die jeweils in Token-basierten Ringstrukturen zu Steuerungszwecken miteinander verbunden sind. Erhält ein Empfänger der ersten Hierarchiestufe das in dieser Hierarchiestufe kursierende Token, hat er die Möglichkeit, fehlende Nachrichten anzufordern und die Aufgabe, mittels einer Multicast-Nachricht alle bei ihm und bei den ihm zugeordneten Empfängern der zweiten Hierarchiestufe lückenlos eingegangenen Nachrichten zu bestätigen und das Token an seinen Ringnachbarn zu senden. Die Bestätigungsnachricht enthält einen Zeitstempel. In einer festgelegten Zeitspanne nach dem Sendezeitpunkt der Bestätigungsnachricht senden alle Knoten der zweiten Hierarchiestufe die Nachrichten per Multicast an die Teilnehmer. Die Zeitspanne beinhaltet die Übertragungszeit zu allen Empfängern einschließlich eventueller Anforderungen zur Neuübertragung verlorengegangener Informations- und Bestätigungsnachrichten.

4.3.4 Diskussion

Das Verfahren Simultaneous Multicast von [Pulido & Lin 1998] bringt für den hier vorgesehenen Anwendungsbereich einige Nachteile mit sich. Fairness wird mit diesem Verfahren nur für große Nachrichten erreicht. Das liegt darin begründet, dass die Verwaltung der Übertragungskapazität in der Vermittlungsschicht, die sogar unter Umständen Bursts zulässt, unabhängig vom Übertragungsmedium ist. Dadurch ist mit der Reservierung keine

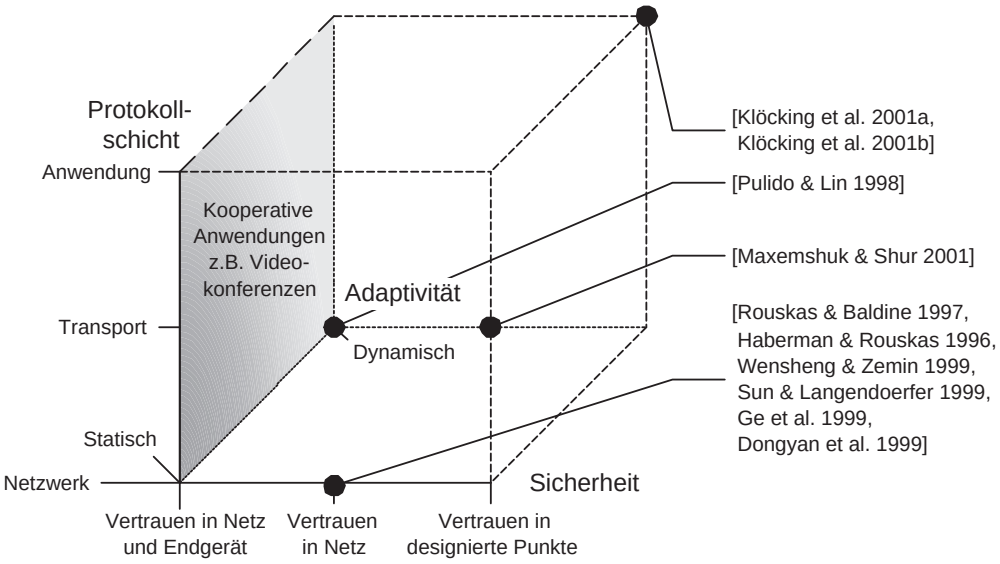


Abbildung 4.4: Klassifikation verwandter Arbeiten

echte Serialisierungsverzögerung verbunden, die dieses Verfahren auszunutzen versucht. Das Verfahren eignet sich nur für Nachrichten, die größer als das zulässige Burst-Volumen sind, weil nur solche Pakete entsprechend verzögert werden. Da RSVP empfängerorientiert ist, werden die Reservierungen vom Protokoll-Stack des Empfängers vorgenommen. Die Berücksichtigung des Zugangsnetzes zum Teilnehmer ist ein großer Vorteil des Verfahrens. Um bei Manipulation der Empfangsgeräte Fairness durchzusetzen, erfordern die hier betrachteten Anwendungen jedoch Sicherheitsvorkehrungen, die das Verfahren nicht bietet.

Die Verfahren zum Aufbau eines speziellen Multicast-Routing-Baumes sind attraktiv, da sie die Minimierung von Verzögerungsunterschieden im Netzwerk lösen. Dies erfolgt durch Ausgleich unterschiedlicher Pfadlängen zu den Empfängern. Ein Ausgleich unterschiedlicher Lastsituationen ist nicht möglich. Fairness kann durchgesetzt werden, solange Vertrauen in das gesamte Netzwerk besteht. Ein weiterer Nachteil dieser Verfahren ist, dass sie nur in Bereichen, in denen ein einheitliches Routing-Protokoll angewandt wird, Verwendung finden können und damit auf autonome Systeme beschränkt sind.

Auch das von [Ge et al. 1999] beschriebene Verfahren kann die Verzögerungsunterschiede für unterschiedliche Lastsituationen nicht minimieren, jedoch wird bei diesem Verfahren die maximale Scheduling-Zeit der Pakete berücksichtigt. Dadurch gleicht das Verfahren den Verzögerungsunterschied für den Volllastfall aus. Die Gesamtverzögerung wird damit auf den Extremfall festgesetzt. In wenig belasteten Netzen mit vielen Knoten und großer

Warteschlangenlänge wird daher keine minimal mögliche Gesamtverzögerung erreicht. Das Verfahren stellt hohe Anforderungen an die Knoten. Zum Einen fordert das Verfahren eine sofortige Umstellung aller Netzknoten. In begrenzten, unter einer Verwaltung stehenden Netzwerken ist dieses Verfahren vorteilhaft. Für große Netze ist die Annahme von Routern, die durchgängig dieses Verfahren beherrschen, jedoch unrealistisch. Zum Anderen wird die praktische Realisierung durch die Notwendigkeit der paketspezifischen Verarbeitung und Verzögerung erschwert. Die Verwaltung von Zuständen pro Paket und Datenfluss stellt einen hohen Zusatzaufwand für Router des Kernnetzes dar.

Bei TRMP wird die Minimierung von Verzögerungsunterschieden durch die Vorgabe einer Zeitspanne bis zum Wiederaussenden der Nachrichten nach Aussenden der Bestätigungsnachricht erreicht. Die Zeitspanne setzt sich zusammen aus der Zeitvorgabe für die Übertragungswiederholungen multipliziert mit deren Anzahl. Der Vorteil eines einheitlichen Zeitpunktes für das Wiederaussenden der Nachrichten wird jedoch dadurch relativiert, dass er alle bis zu diesem Zeitpunkt neu eingetroffenen Nachrichten bei einem Empfänger der ersten Hierarchie seit der letzten Bestätigung durch seinen Ringnachbarn betrifft. Bei hoher Informationsdatenrate können sich Empfänger mit einer Anbindung hoher Übertragungskapazität gegenüber Empfängern mit einer Anbindung niedriger Übertragungskapazität einen Vorteil durch schnelleren Zugriff auf die Nachrichten verschaffen. Die Fairness kann bei diesem Verfahren sehr gut durchgesetzt werden, da das Vertrauen auf einzelne vertrauenswürdige Knoten wie spezielle Sender, Empfänger und Server beschränkt ist. Die Festsetzung der Zeit für das Wiederaussenden wird nicht an die Anzahl der notwendigen Übertragungswiederholungen oder mögliche Veränderungen der Zeitvorgabe für Übertragungswiederholungen, z. B. durch eine veränderte Lastsituation, adaptiert. Dadurch ist die Gesamtverzögerung auf ein Vielfaches der Zeitvorgabe für Übertragungswiederholungen festgelegt.

Die grundsätzlichen Nachteile der bisherigen Verfahren liegen demzufolge darin, dass sie die Ursachen für die Verzögerungsunterschiede nur teilweise ausgleichen. Eine sichere Berücksichtigung des Zugangsnetzes erfolgt nicht. Die meisten Verfahren benötigen für die Durchsetzung der Fairness einen großen Vertrauensbereich. Bezogen auf die Gesamtverzögerung der Nachrichten arbeiten die Verfahren sehr statisch, da sie vom ungünstigsten Fall, beispielsweise von der höchsten Netzwerklast oder der maximalen Anzahl der Übertragungswiederholungen ausgehen. Dagegen wird die aktuelle Netzwerklast von keinem der bisher bekannten Verfahren berücksichtigt.

4.4 Verzögerungsfairer Multicast-Dienst

Das *Internet Protocol* (IP) hat sich als ubiquitäres Protokoll zur Verbindung unterschiedlicher Kommunikationstechnologien wie LANs, Mobilfunk- und Satellitennetzen durchgesetzt. Für einen globalen Zugang zu Informationen ist das Internet daher das am besten geeignete Medium. Gruppenkommunikation über IP wird durch *Multicast* unterstützt. Die Datenübertragung durch Multicast bietet den Vorteil, dass das einmalige Senden einer Nachricht ausreicht, um eine beliebig große Empfängermenge zu erreichen. Das heißt, die Nachricht wird über jede einzelne Verbindung zwischen Knoten nur einmal übertragen. Der Nachrichtenaufwand für die Informationsverteilung ist dadurch minimal. Die in Abschnitt 4.2 aufgestellten Forderungen nach einem von heterogenen Netztechnologien unabhängigen Informationsverteildienst, der in der Lage ist, auch hohe Datenvolumina ressourcenschonend zu transportieren, werden damit optimal erfüllt. Mit dem verzögerungsfairen Multicast-Dienst wird nicht-kooperativen Echtzeitanwendungen der Klasse 1 ein Dienst zur Verfügung gestellt, der die verzögerungsfaire Verteilung von Informationen übernimmt.

4.4.1 Systemmodell

In diesem Abschnitt werden die zur Lösungsfindung wesentlichen Eigenschaften des Internets mittels eines Systemmodells von Netzwerk, IP-Multicast-Dienst und Sicherheitsmodell konkretisiert.

4.4.1.1 Netzwerkmodell

Ziel der Untersuchungen ist es, praktikable Lösungen zur Reduktion des Verzögerungsunterschiedes zwischen Empfängern zu erreichen. Das für die Untersuchungen verwendete Netzwerkmodell lehnt sich daher eng an das gegenwärtige Internet an, da größere Änderungen, wie z. B. der Austausch aller Router, mittelfristig unrealistisch sind.

Die Topologie des Kernnetzwerkes wird als ein aus einzelnen Dienstanbietern bestehendes Netzwerk angenommen. Kommunikationsteilnehmer sind über ein Zugangsnetzwerk mit dem Kernnetzwerk verbunden. Die Nachrichtenübertragung erfolgt mittels Paketvermittlung in Netzknoten. Die Netzknoten sind autonom und besitzen keinen gemeinsamen Speicher.

4.4.1.2 Dienstmodell

Für die Versendung von Gruppennachrichten wird das IP-Multicast-Modell zugrunde gelegt, das das Senden von jedem Gruppenmitglied an die Gruppe erlaubt. Gruppenmitglieder können jederzeit der Gruppe beitreten oder sie verlassen. Eine zentralisierte Gruppenverwaltung existiert nicht.

Die Anbieter stellen eine sogenannte Best-Effort-Qualität der Dienste zur Verfügung. Das bedeutet, dass sie im Rahmen ihrer Möglichkeiten alles für die Erbringung des Paketvermittlungsdienstes tun, jedoch keine Garantien für eine erfolgreiche oder zeitkritische Übermittlung geben. Die für die Erbringung von Dienstgarantien notwendigen Verfahren (z. B. Reservierung oder Zugangskontrolle) sind nicht global vorhanden und damit für die Minimierung von Verzögerungsunterschieden zwischen Empfängern nicht allgemein einsetzbar. Übertragungsfehler werden durch Fehlerkorrekturmaßnahmen der Transportschicht oder der Anwendungsschicht behandelt.

Des Weiteren wird angenommen, dass Dienste zur Synchronisation von Uhren verfügbar sind. Die Zeitinformation ist entweder über Funkempfänger oder Empfänger von Signalen des *Global Positioning System (GPS)* direkt erhältlich oder kann über Uhrensynchronisationsprotokolle mit ausreichender Genauigkeit empfangen werden. Das *Network Time Protocol (NTP)* zum Beispiel erreicht eine Synchronisationsgenauigkeit in der Größenordnung von einer Millisekunde [Mills 1995].

4.4.1.3 Vertrauens- und Sicherheitsmodell

Sicherheitsaspekte spielen eine große Rolle beim Entwurf von fairen Diensten. Werden sie nicht berücksichtigt, kann Fairness nicht garantiert werden. So geht z. B. die Staukontrolle von TCP, die die Kooperation von allen Endsysystemen fordert, derzeit noch vom Modell des sicheren Netzes und des nicht manipulierbaren Protokoll-Stack auf den Endsystemen aus. Die Verwendung aggressiverer Staukontroll-Algorithmen zeigt jedoch, dass die Erreichung eines fairen Durchsatzes zwischen Anwendungen dann nicht mehr möglich ist. Dieses Modell ist daher nicht auf den hier betrachteten Fall anwendbar.

Da es im Interesse des Empfängers ist, an die Informationen so früh wie möglich zu gelangen, geht das hier verwendete Modell nicht davon aus, den Empfängern Vertrauen entgegenzubringen. Hingegen kann dem Sender ein Interesse an der verzögerungsfairen Verteilung der Informationen unterstellt werden. Dem Kernnetz, bestehend aus einzelnen

konkurrierenden Dienstanbietern, kann nicht generell Vertrauen entgegengebracht werden. Es können jedoch einzelne vertrauenswürdige Bereiche geschaffen werden, wie z. B. Hochsicherheitsräume, die speziell abgesichert und überwacht werden. Über Vertrauen muss hier im Einzelfall entschieden werden. Das resultierende Modell ist in Abbildung 4.5 dargestellt.

Für die Verwirklichung einer Sicherheitsarchitektur werden Basisdienste zur Verschlüsselung und Entschlüsselung von Nachrichten vorausgesetzt. Es sollen sowohl symmetrische als auch asymmetrische Verschlüsselungsverfahren vorhanden sein. Bei den symmetrischen Verfahren wird mittels eines Schlüssels die Nachricht vom Sender verschlüsselt und kann dann über einen unsicheren Kanal übertragen werden. Unter Verwendung desselben Schlüssels kann der rechtmäßige Empfänger der Nachricht diese wieder entschlüsseln. Symmetrische Verfahren erfordern einen sicheren Kanal für den Austausch des Schlüssels.

Bei den asymmetrischen Verschlüsselungsverfahren haben Sender und Empfänger jeweils ein Schlüsselpaar. Das Schlüsselpaar hat die Eigenschaft, dass Nachrichten, die mit einem der beiden Schlüssel des Paares verschlüsselt wurden, sich nur mit dem anderen Schlüssel dieses Paares wieder entschlüsseln lassen und umgekehrt. Veröffentlicht der Erzeuger des Schlüsselpaares nun einen der Schlüssel und hält den anderen geheim, können Kommunikationspartner den öffentlichen Schlüssel zur Verschlüsselung der an ihn gerichteten Nachrichten verwenden. Die Nachrichten können nur von dem Besitzer des dazugehörigen privaten Schlüssels entschlüsselt werden. Bei den asymmetrischen Verschlüsselungsverfahren muss nur sichergestellt werden, dass der verwendete öffentliche Schlüssel tatsächlich vom gewünschten Kommunikationspartner stammt. Für die asymmetrische Verschlüsselung sind daher Verfahren für den Schlüsselaustausch und das Schlüsselmanagement erforderlich. Eine Einführung in symmetrische und asymmetrische Verschlüsselungsverfahren findet sich in [Schneier 1996]. [Perlman 1999] und [Borella 2000] geben eine Übersicht über Infrastrukturen für den Austausch öffentlicher Schlüssel und zugehörige Vertrauensmodelle. Für den sicheren Schlüsselaustausch soll eine von allen Parteien als vertrauenswürdig eingestufte Instanz existieren, auf die zurückgegriffen werden kann, um eine Partei zu authentifizieren.

Des Weiteren kann auf sichere Hardware zurückgegriffen werden, die die Eigenschaft hat, dass ihre Manipulation um Größenordnungen aufwändiger ist als der Nutzen, der aus einer Manipulation gezogen werden kann.

Hauptziel der in diesem Kapitel vorgestellten Sicherheitsprotokolle ist es, die Informationen vor verfrühtem Zugriff durch die Teilnehmer der Gruppenkommunikation zu schützen.

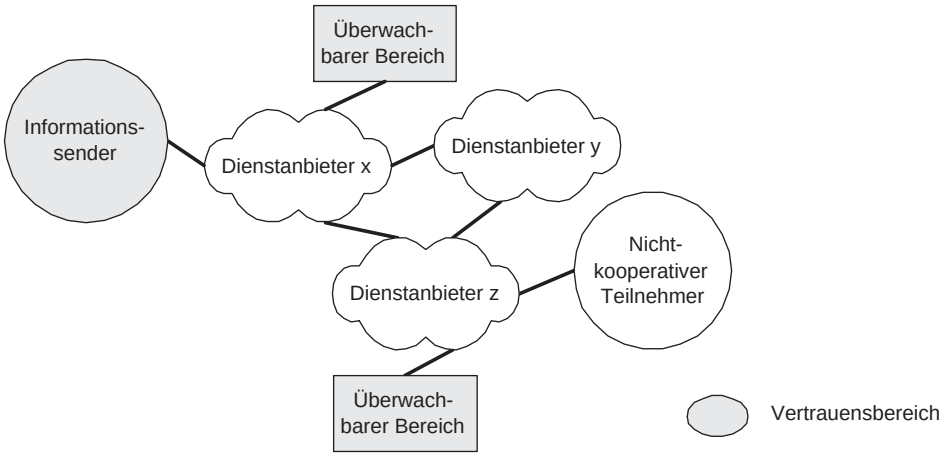


Abbildung 4.5: Netz- und Vertrauensmodell

Ein Angreifer soll alle im Dolev-Yao-Modell [Dolev & Yao 1983] enthaltenen Fähigkeiten besitzen. Er hat potenziell die volle Kontrolle über das Kommunikationsnetzwerk. Er kann Nachrichten abfangen, wiederholt senden, fälschen, verändern, blockieren und einfügen. Er kann Nachrichten zu/von jedem legitimen Knoten senden bzw. empfangen sowie als legitimer Teilnehmer auftreten. Insbesondere sei hier noch einmal herausgestellt, dass er auch in der Lage ist, Nachrichten beliebig zu verzögern.

4.4.2 Entwurfsentscheidung

Eines der bedeutendsten Entwurfsprinzipien, das die Entwicklung des Internets begleitet, ist das Prinzip der Ende-zu-Ende-Argumente von Saltzer, Reed und Clark [Saltzer et al. 1981]. Es besagt, dass Funktionen, die auf niedrigen Schichten zur Verfügung gestellt werden, ggf. redundant oder von geringem Nutzen sind, wenn die Folgekosten für deren Implementierung berücksichtigt werden. Der Grund hierfür ist, dass im Allgemeinen die Leistungsfähigkeit der Systeme herabgesetzt wird, wenn auf niedrigen Schichten Funktionen implementiert werden. Die Implementierung lohnt sich nur dann, wenn die Funktion in der niedrigen Schicht vollständig implementiert werden kann und alle Anwendungen einen Nutzen aus dieser Funktion ziehen. Folglich muss die Frage beantwortet werden, ob die gewünschte Funktion der Reduktion von Verzögerungsunterschieden vollständig in der Vermittlungsschicht implementiert werden kann und ob alle Anwendungen daraus einen Nutzen ziehen können.

Eine Betrachtung der Ursachen für Verzögerungsunterschiede ergibt, dass sich einige Ursachen auf der Vermittlungsschicht ausgleichen lassen. Das betrifft vor allem die Wandlung

von dynamischen Aspekten wie Scheduling-Zeit oder Pfadlängen in statische Aspekte wie maximale Scheduling-Zeit und maximale Pfadlänge. Einige Ursachen können jedoch nicht berücksichtigt werden, wie z. B. Übertragungswiederholungen aufgrund von Paketverlusten. Eine vollständige Implementierung in der Vermittlungsschicht ist daher nicht möglich. Darüber hinaus beschränkt sich der Nutzen der Funktion auf die Menge der sich nicht kooperativ verhaltenden Echtzeitanwendungen. Außerdem würden Implementierungen in der Vermittlungsschicht mittels spezieller Scheduler den Nutzen noch weiter einschränken und bestimmte QoS-Verbesserungen der Vermittlungsschicht (z. B. *Integrated Services*, *Differentiated Services*) nicht erlauben. Daher ist der Aufwand für die Implementierung in der Vermittlungsschicht nicht gerechtfertigt und die Betrachtung höherer Schichten erforderlich.

Hier bietet sich zunächst die Transportschicht an. Die Forderung der Anwendungen nach geringem Verzögerungsunterschied zwischen den Empfängern ist eng mit der Forderung nach verlustloser Übertragung der Nachrichten verbunden, denn die gleichzeitige Auslieferung der Nachrichten nur an einen Teil der Empfänger ist ebenfalls unfair. Hierzu sind Fehlererkennung, Fehlerkorrektur, Fluss- und Staukontrolle erforderlich, von denen sich insbesondere die Fehlerkorrektur durch Übertragungswiederholung signifikant auf den Verzögerungsunterschied auswirkt. Für verschiedene Anwendungen werden verschiedene Fehlerkorrekturmethoden genutzt. Es wäre daher von Nachteil, den Dienst an eine spezielle Methode der Transportschicht zu koppeln. Ein allgemeiner Entwurf für alle Anwendungen ist daher nur oberhalb der Transportschicht, d.h. in der Anwendungsschicht, aussichtsreich.

Als Grund für die Wahl der Anwendungsschicht zur Dienstbereitstellung ist vor allem das notwendige Vertrauen zu nennen. Je kleiner der Bereich ist, dem Vertrauen entgegengebracht werden muss, desto sicherer ist es, dass der Dienst die an ihn gestellten Anforderungen erfüllt. Da Vertrauen in das gesamte Netz aus den in Abschnitt 4.4.1.3 erläuterten Gründen nicht gerechtfertigt ist, muss sich die Lösung auf einzelne gut zu überwachende Punkte beschränken.

Im Folgenden werden daher Verfahren der Anwendungsschicht präsentiert. Das Verfahren der Datenserver repräsentiert den allgemeinsten Fall. Die weiteren Ansätze sind Spezialisierungen mit teilweise erhöhten Anforderungen, aber auch höherer Reduktion des Verzögerungsunterschiedes. Die für die Verfahren benötigten Protokolle werden schrittweise eingeführt und in Kapitel 5 evaluiert.

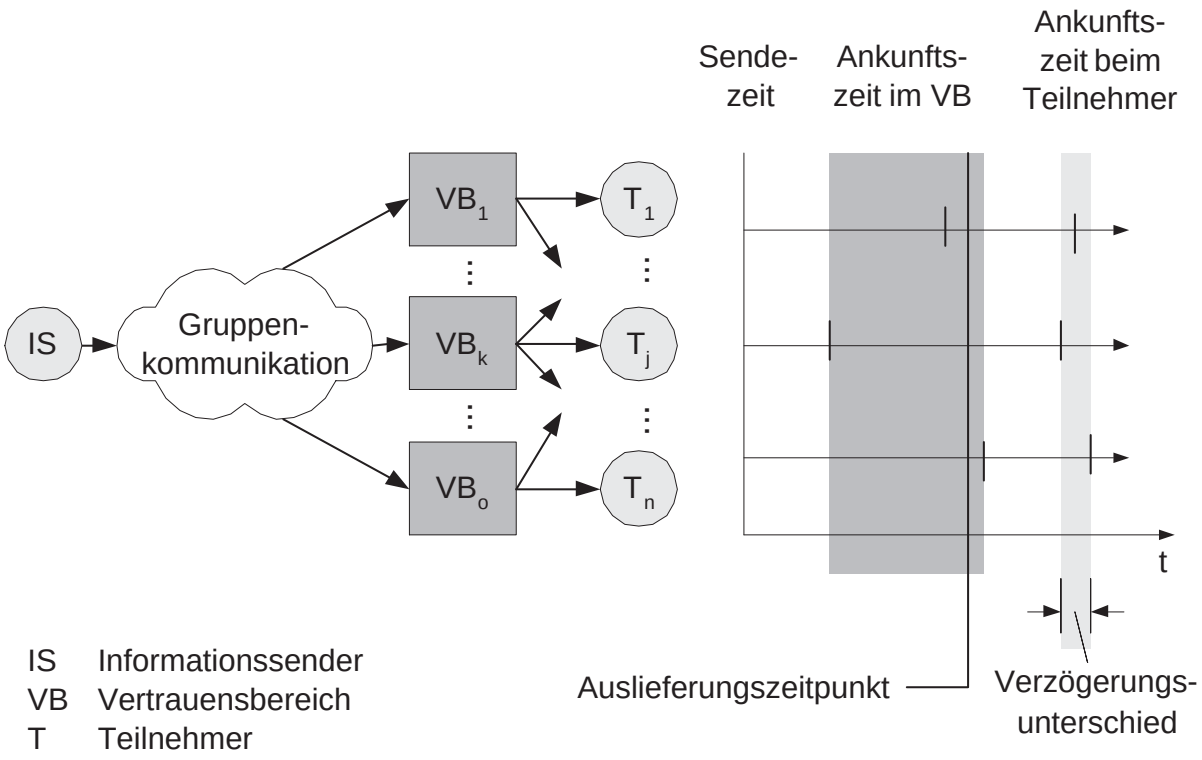


Abbildung 4.6: Grundkonzept des verzögerungsfairen Dienstes

4.4.3 Konzept und Schnittstellen

Das Grundkonzept des hier vorgestellten Verfahrens zur Reduktion der Verzögerungsunterschiede zwischen Teilnehmern von Gruppenkommunikation geht von Vertrauensbereichen nahe den Teilnehmern aus (siehe Abbildung 4.6). Um die bis dahin entstandenen Verzögerungsunterschiede auszugleichen, werden die Nachrichten in diesen Vertrauensbereichen resynchronisiert. Das wird durch das zeitgleiche Wiederaussenden der Nachrichten erreicht. Dieser Zeitpunkt soll als Auslieferungszeitpunkt bezeichnet werden:

> Definition 4.1 (Auslieferungszeitpunkt)
> Der Auslieferungszeitpunkt t^A einer Nachricht ist der Zeitpunkt, zu dem die Nachricht den Vertrauensbereich verlassen kann.

In Abbildung 4.7 sind die Funktionsbausteine zur Erbringung des verzögerungsfairen Dienstes dargestellt. Die Funktion **Auslieferung** sendet die Nachrichten zum jeweiligen Auslieferungszeitpunkt an die Teilnehmer und stellt sicher, dass die Nachrichten den Vertrauensbereich nicht vor dem Auslieferungszeitpunkt verlassen.

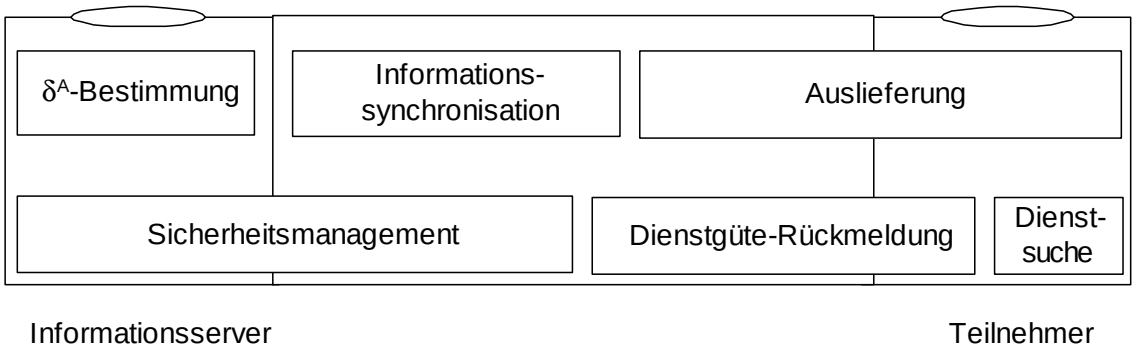


Abbildung 4.7: Funktionsbausteine des verzögerungsfairen Dienstes

Damit jeder Vertrauensbereich denselben Auslieferungszeitpunkt für jede Nachrichtenkopie wählt, muss dieser Zeitpunkt allen Vertrauensbereichen bekannt sein. Hierfür wird auf externe vertrauenswürdige Zeitinformation und Information vom Sender zurückgegriffen. Die **Informationssynchronisation** übernimmt die Aufbereitung dieser Information. Da von einem Netzwerk ohne synchrone Nachrichtenauslieferung ausgegangen wird, existiert keine Möglichkeit, allen Servern zur selben Zeit mitzuteilen, dass die Nachrichten ausgeliefert werden können. Deshalb setzt der Sender vor dem Versand der Nachricht den Auslieferungszeitpunkt fest, zu dem die Teilnehmer die Nachricht frühestens erhalten dürfen. Der Sender teilt diese Information den Vertrauensbereichen in Form des Sendezeitpunktes der Nachricht und der Auslieferungsverzögerung mit.

Definition 4.2 (Auslieferungsverzögerung)
Die Auslieferungsverzögerung δ^A gibt die Zeitdauer zwischen Sendezeitpunkt t^S einer Nachricht von Sender S bis zum Auslieferungszeitpunkt t^A dieser Nachricht an.

Der Auslieferungszeitpunkt ist folglich die Summe aus Sendezeitpunkt plus Auslieferungsverzögerung.

$$t^A = t^S + \delta^A$$

δ^A-Bestimmung. Der Sender bestimmt die Auslieferungsverzögerung durch Schätzung, Konfiguration oder Testnachrichten ohne Anwendungsdaten. Im Abschnitt 4.7 wird ein dynamischer Algorithmus zur Bestimmung des Auslieferungszeitpunktes vorgestellt. Dieser nutzt die **Dienstgüterückmeldung** zur Anpassung der Auslieferungsverzögerung an den aktuellen Netzwerkzustand.

Die Vertrauensbereiche stellen die Nachrichten wiederum über Multicast zur Verfügung. Aufgabe der **Dienstsuche** auf Teilnehmerseite ist es, einen Vertrauensbereich zu wählen, von dem die Nachrichten mit der geringsten Verzögerung eintreffen. Dies erfolgt mit der Technik des *Expanding Ring Search* [Boggs 1983] oder mittels *Token Repository Service* [Rothermel & Maihöfer 1999].

Weil Multicast empfängerinitiiert ist, gilt es zu verhindern, dass die Teilnehmer die Daten direkt vom Sender beziehen. Das **Sicherheitsmanagement** stellt durch Verschlüsselung sowie durch administrative Massnahmen sicher, dass die Nachrichten vor dem Auslieferungszeitpunkt geschützt sind. Die synchrone Weiterleitung der Nachrichten beruht auf einem gemeinsamen Verständnis der Vertrauensbereiche über die aktuelle Uhrzeit. Da die Zeitinformation aus externen Quellen bezogen wird, verifiziert das Sicherheitsmanagement außerdem durch Überprüfung von Signaturen, ob diese Zeitinformation vertrauenswürdig ist. Vertrauensbereiche sind die in Abschnitt 4.4.1.3 erwähnten Bereiche Sender und Sichere Hardware sowie die durch Verschlüsselung der Nachricht geschaffenen Bereiche. Der Auslieferungszeitpunkt fällt daher in der Regel mit dem Zeitpunkt zusammen, zu dem die sichere Hardware die Nachricht weiterleitet. Wie im Abschnitt 4.6 beschrieben, können sich vertrauenswürdige Bereiche über die sichere Hardware hinaus erstrecken. Den Teilnehmern werden keine sicherheitskritischen Funktionen übertragen. Sie leiten lediglich die erhaltenen Informationen an die Anwendung weiter.

Aus den in Abschnitt 4.4.2 erläuterten Gründen wird der Dienst oberhalb der Transportschicht erbracht. Er setzt auf ein beliebiges Transportprotokoll für Gruppenkommunikation auf. Abbildung 4.8 verdeutlicht die Einordung in den Protokollstack.

Die Schnittstelle, über die die Anwendung auf den verzögerungsfairen Dienst zugreift, offeriert die verzögerungsfaire Versendung von Nachrichten, die Abfrage der Auslieferungsverzögerung und die Angabe eines Auslieferungszeitpunktes.

Mittels folgenden Primitivs sendet die Anwendung des Senders Informationen I verzögerungsfair an eine Gruppe G:

```
sendeVerzoegerungsfair(G,I)
```

Optional ist die Angabe eines Auslieferungszeitpunktes t^A möglich:

```
sendeVerzoegerungsfair(G,I,t^A)
```

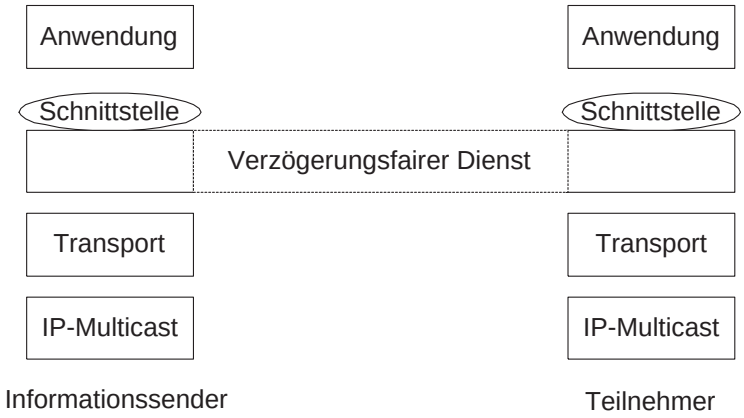


Abbildung 4.8: Schnittstellen des verzögerungsfairen Dienstes

Wird von der Anwendung ein Auslieferungszeitpunkt angegeben, berechnet der Dienst unter Berücksichtigung der aktuellen Auslieferungsverzögerung den Sendezeitpunkt und überprüft, ob der berechnete Sendezeitpunkt in der Zukunft liegt. Anderenfalls wird die Information sofort versendet.

$$t^S = \max\{t, t^A - \delta^A\}$$

Die Anwendung kann mittels folgender Funktion die aktuelle Auslieferungsverzögerung erfahren, so dass sie in die Lage versetzt wird, den Zeitpunkt für die Bereitstellung von zeitkritischen Daten zu ermitteln:

`liesAuslieferungsverzoegerung(G):` δ^A

Die Schnittstelle der Teilnehmer-Anwendung hat keine über den normalen Datenempfang hinausgehende Funktionalität:

```
empfange(G): I
```

Bestehende Anwendungen werden volltransparent unterstützt, indem die gesamten Nachrichten durch den Dienst verzögerunsfair über eine alternative Gruppe versendet und für den Teilnehmer wieder auf die ursprüngliche Gruppe umgesetzt werden.

4.5 Realisierungsvarianten

Im folgenden werden drei Realisierungsvarianten des Konzepts zur Reduktion von Verzögerungsunterschieden zwischen Empfängern vorgestellt. Sie unterscheiden sich in Funktion und Ort der Vertrauensbereiche. Im Datenserver-Verfahren werden Datenserver in den

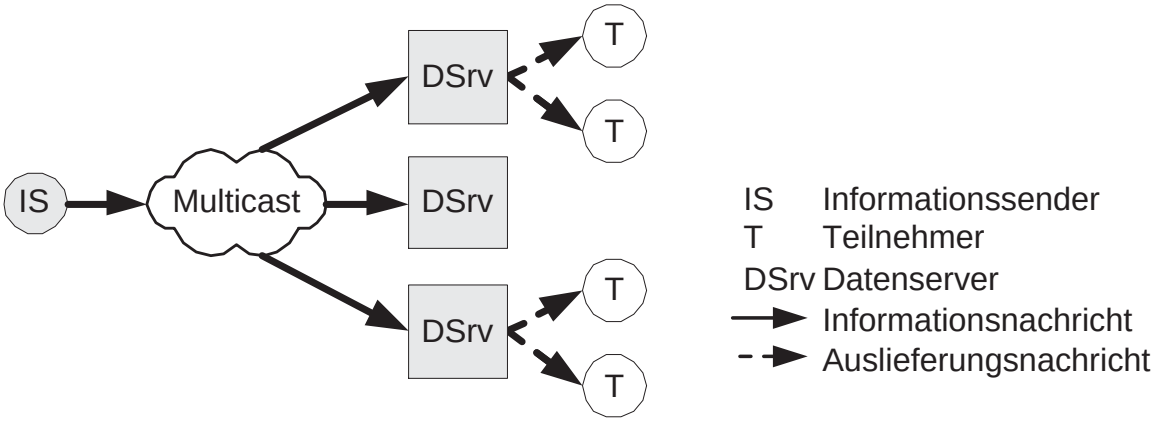


Abbildung 4.9: Grundlegendes Prinzip des Datenserver-Verfahrens

Vertrauensbereichen platziert, die die Informationen resynchronisieren. Die Vertrauensbereiche im Schlüsselserver-Verfahren stellen den Teilnehmern zum Auslieferungszeitpunkt Schlüssel zur Entschlüsselung der Informationen zur Verfügung. Das Zeitserver-Verfahren reduziert den Vertrauensbereich im Netzwerk auf die Bereitstellung von Zeitinformation und baut stattdessen mittels sicherer Hardware einen Vertrauensbereich direkt beim Teilnehmer auf.

4.5.1 Das Datenserver-Verfahren

Der Grundgedanke dieser Methode besteht darin, die Daten nicht direkt an die Empfänger zu leiten, sondern durch Zwischenschaltung von Datenservern, die sich in der Nähe der Empfänger befinden, Verzögerungsunterschiede auszugleichen [Klöcking et al. 2001a]. Die in Abbildung 4.9 dargestellte Topologie dieses Verfahrens zeigt Datenserver in den Vertrauensbereichen. Diese Server haben die Aufgabe, die Daten vom Sender zu empfangen, zu speichern und zu einer vom Sender vorgegebenen Zeit weiterzuleiten. Die Server erreichen durch diese Resynchronisation der Nachrichten die in Abbildung 4.6 veranschaulichte Glättung des Verzögerungsunterschiedes zwischen den Empfängern, da ein Großteil der in Abschnitt 3.1 beschriebenen Ursachen für Verzögerungsunterschiede zwischen den Empfängern ausgeschaltet wird. Das Verfahren reduziert die Verzögerungsunterschiede umso wirkungsvoller, je näher sich die Server bei den Teilnehmern befinden.

```
method sendeVerzoegerungsfair(G, I, t^A)
if t < (t^A − δ^A(G)) then
   schedule(t^A − δ^A(G), sendeVerzoegerungsfair(G, I));
else
   sendeVerzoegerungsfair(G, I);
end if

method sendeVerzoegerungsfair(G, I)
t^S = t;
IN := InformationsNachricht.neu(I, t^S, δ^A(G));
IN.hashAuslieferungsZeit;
IN.verschluesseleSym(Ring.S_{IS/DSrv}(G_IN(G)));
sende(G_IN(G), IN);
```

Algorithmus 4.1: Algorithmus des Informationssenders im Datenserver-Verfahren

Aufgabe des Informationssenders ist es, einen Teil des Sicherheitsmanagements und die δ^A-Bestimmung zu übernehmen. Der Sender setzt die Auslieferungsverzögerung nach dem in Abschnitt 4.7 näher beschriebenen Verfahren fest. Falls die Anwendung einen Auslieferungszeitpunkt vorgegeben hat, werden die Nachrichten einem Scheduler übergeben. Zum Sendezeitpunkt wird ein Hash-Wert über Sendezeit und Auslieferungsverzögerung gebildet, die zu übertragende Information und der gebildete Hash-Wert mit dem Sitzungsschlüssel verschlüsselt und mit Sendezeitpunkt und Auslieferungsverzögerung versandt. Der Sender arbeitet Algorithmus 4.1 ab.

Die Datenserver übernehmen Teile des Sicherheitsmanagements, die Aufbereitung der Informationssynchronisation und die Funktion der Auslieferung. Sie empfangen die verschlüsselten Informationsnachrichten des Informationssenders, entschlüsseln diese, verifizieren die Authentizität des Sendezeitpunktes und der Auslieferungsverzögerung, ermitteln den Auslieferungszeitpunkt der Nachricht und senden diese unverschlüsselt an die Teilnehmer. Die Server arbeiten demgemäß Algorithmus 4.2 ab.

Für die Übermittlung der Information vom Sender zu den Teilnehmern werden mittels Algorithmus 4.2 zwei Nachrichten, die Informationsnachricht und die Auslieferungsnachricht, gesendet:

Informationsnachricht		: IS ⇒ DSrv	:	$\{I, h(t^S, \delta^A)\}^{S_{IS/DSrv}}, t^S, \delta^A$
Auslieferungsnachricht	$t \geq t^A$	: DSrv ⇒ T	:	I

```
    method verarbeiteInformationsNachricht(G_IN)
    IN := InformationsNachricht.empfange(G_IN);
    IN.entschluessele(S_IS/DSrv(G_IN));
    if IN.digest ≠ h(IN.t^S, IN.δ^A) then
 5:   behandleAusnahme('Auslieferungszeit gefälscht');
    end if
    if t < (IN.t^S + IN.δ^A) then
      schedule(IN.t^S + IN.δ^A, sende(G_AN(G_IN), IN.I));
    else
10:   sende(G_AN(G_IN), IN.I);
    end if
```

Algorithmus 4.2: Algorithmus des Datenservers im Datenserver-Verfahren

In der verwendeten Notation für die Sicherheitsprotokolle bezeichnet IS den Informationssender, Srv einen Server und T den Teilnehmer. ZS stellt einen Zeitstempel dar, S den Schlüssel eines symmetrischen Verschlüsselungsverfahrens, PS den privaten und $\ddot{O}S$ den öffentlichen Schlüssel eines asymmetrischen Verschlüsselungsverfahrens. $\{N\}^S$ bedeutet, dass die Nachricht N mit dem Schlüssel S verschlüsselt ist. Die Einwegfunktion h liefert einen Hashwert. Ferner bezeichnet die Funktion $(N)sig_X$, dass die Nachricht N von X signiert wurde $(N, \{h(N)\}^{GS_X})$. Die Nachrichtenübermittlung durch Unicast wird durch einen Pfeil $\rightarrow$, die durch Multicast durch einen Doppelpfeil $\Rightarrow$ gekennzeichnet.

Voraussetzung für den Austausch dieser Nachrichten ist, dass durch eine Kombination aus Zertifizierung und asymmetrischer Verschlüsselung nur vertrauenswürdige Server und der Sender den Sitzungsschlüssel $S_{IS/DSrv}$ erhalten. Ziel des verschlüsselten Nachrichtenaustausches ist es, die Information nicht vor dem Auslieferungszeitpunkt den Teilnehmern zugänglich zu machen.

Die von den Datenservern empfangene *Informationsnachricht* des Informationssenders beinhaltet den Sendezeitpunkt der Nachricht, die Auslieferungsverzögerung sowie die mittels eines symmetrischen, zwischen Informationssender und Datenservern vereinbarten Schlüssels verschlüsselte Information. Der verschlüsselte Teil der Nachricht besteht aus der zu übertragenden Information I sowie einem Hashwert über den Sendezeitpunkt und die Auslieferungsverzögerung. Sendezeitpunkt und Auslieferungsverzögerung sind nicht geheim, jedoch müssen sie aus einer vertrauenswürdigen Quelle stammen und einer Nachricht zuordenbar sein. Über den Hashwert und die Verschlüsselung mit dem Sitzungsschlüssel werden Sendezeitpunkt und Auslieferungsverzögerung für die Sitzung an die

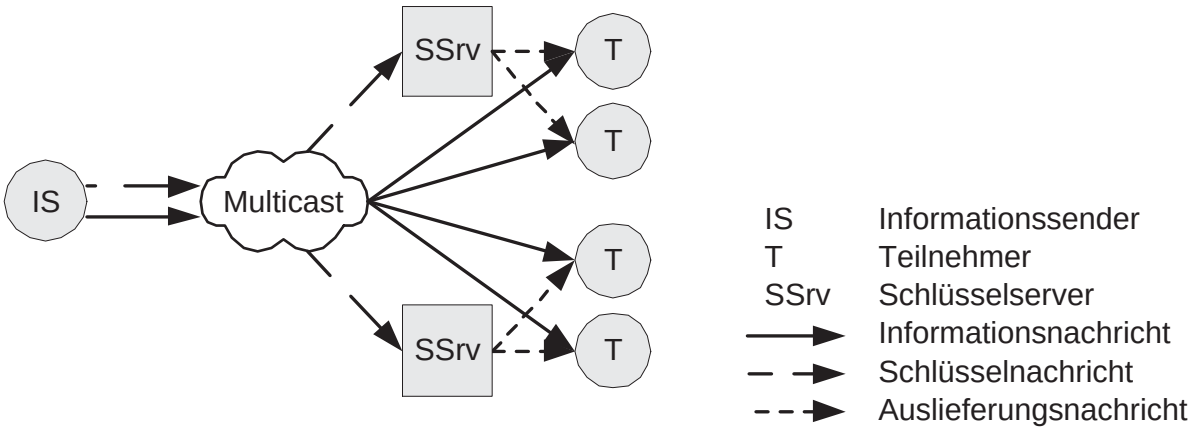


Abbildung 4.10: Grundlegendes Prinzip des Schlüsselserver-Verfahrens

Nachricht gebunden. Die *Auslieferungsnachricht* mit der unverschlüsselten Information I wird von den Datenservern an die Teilnehmer unter der Bedingung versandt, dass die aktuelle Uhrzeit mindestens den Auslieferungszeitpunkt erreicht hat.

4.5.2 Das Schlüsselserver-Verfahren

Das Schlüsselserver-Verfahren ist ein weiteres Verfahren, mit dem Verzögerungsunterschiede zwischen Empfängern von Gruppenkommunikation reduziert werden können. Im Gegensatz zum Datenserver-Verfahren werden hier die Angaben zur Bestimmung des Auslieferungszeitpunktes von der Information getrennt übertragen. Dies ist vorteilhaft für die Übertragung langer Nachrichten, denn die Nachrichtengröße hat auf Übertragungsabschnitten geringer Übertragungskapazität starken Einfluss auf die Verzögerung und somit auf den Verzögerungsunterschied zwischen Empfängern (siehe auch Abschnitt 5.1). Ein einfaches Beispiel soll dies verdeutlichen. Wird eine 64 KByte große Nachricht über einen Übertragungsabschnitt mit einer Kapazität von 56 kbit/s übertragen, so entsteht allein durch die Serialisierung eine Verzögerung von 9.4 s. Dieselbe Nachricht wird auf einem Übertragungsabschnitt der Kapazität von 1 Gbit/s nur um 0.52 ms verzögert. Handelt es sich bei den beiden Nachrichten um Kopien derselben Nachricht an unterschiedliche Teilnehmer, so ist ein Verzögerungsunterschied zwischen den Teilnehmern von 9.4 s entstanden. Im Vergleich dazu ruft eine Nachricht von 60 Byte nur einen Verzögerungsunterschied von 8.6 ms hervor.

```
    method sendeVerzoegerungsfair(G, I)
    t^S := t;
    IN := InformationsNachricht.neu(SequenzNr.folgendeNummer(G), I);
    S_I := Informationsschluessel.zufaellig;
5:  IN.verschluesseleSymInformation(S_I);
    sende(G_IN(G), IN);
    SN := SchluesselNachricht.neu(SequenzNr.Nummer(G), S_I, t^S, δ^A(G));
    SN.hashAuslieferungszeit;
    SN.verschluesseleSymInformationsschluessel(Ring.S_IS/S_SSrv(G));
10: sende(G_SN(G), SN);
```

Algorithmus 4.3: Algorithmus des Senders im Schlüsselserver-Verfahren

Für Anwendungen mit langen Nachrichten ist es daher sinnvoll, die Informationsübertragung von der Auslieferungsnachricht zu trennen, da der Auslieferungszeitpunkt durch kürzere Nachrichten bekannt gegeben werden kann. Dazu stellt der Sender die Informationsnachrichten mit einem Einmal-Schlüssel verschlüsselt per Gruppenkommunikation zur Verfügung, so dass sie von den Teilnehmern direkt empfangen, jedoch nicht sofort entschlüsselt werden können (Abbildung 4.10):

Informationsnachricht		: IS ⇒ T	:	$i, \{I\}^{S_I}$
Schlüsselnachricht		: IS ⇒ SSrv	:	$i, \{S_I, h(t^S, \delta^A)\}^{S_{IS/SSrv}}, t^S, \delta^A$
Auslieferungsnachricht	$t \geq t^A$	: SSrv ⇒ T	:	i, S_I

```
    method verarbeiteSchluesselnachricht(G_SN)
    SN := SchluesselNachricht.empfange(G_SN);
    SN.entschluesseleSym(Ring.S_IS/S_SSrv(G_SN));
    digest := h(SN.t^s, SN.δ^A);
5:  if (SN.digest ≠ digest) then
        behandleAusnahme('Auslieferungszeit gefälscht');
    end if
    AN := AuslieferungsNachricht.neu(SN.i, SN.S_I);
    schedule(SN.t^S + SN.δ^A, sende(G_AN(G_SN), AN);
```

Algorithmus 4.4: Algorithmus des Schlüsselservers im Schlüsselserver-Verfahren

Den Einmal-Schlüssel S_I zur Entschlüsselung der Information sowie den zugehörigen Auslieferungszeitpunkt übermittelt der Sender in der Schlüsselnachricht an die Schlüsselserver. Diese geben die Schlüssel zum Auslieferungszeitpunkt an die Empfänger in der Auslieferungsnachricht weiter. Die fortlaufende Nachrichtennummer i versetzt den Teilnehmer in die Lage, den Informationsschlüssel und die verschlüsselte Informationsnachricht zuzuordnen. Die Algorithmen 4.3, 4.4 und 4.5 für Informationssender, Schlüsselserver und Teilnehmer beschreiben sicherheitsrelevante Aspekte der Nachrichtenverarbeitung in Pseudocode.

```
    method beitritt(G)
    beitritt(G_IN(G));
    beitritt(G_AN(G));

    method I empfange(G)
 5: IN := InformationsNachricht.empfange(G_IN(G));
    Puffer.speichereInformationsNachricht(G, IN.i, IN);
    if Puffer.AuslieferungsNachricht(G, IN.i) then
      Puffer.entschluesseleSymInformation(G, IN.i);
    end if
10: AN := AuslieferungsNachricht.empfange(G_AN(G));
    Puffer.speichereAuslieferungsNachricht(G, AN.i, AN);
    if Puffer.InformationsNachricht(G, AN.i) then
      Puffer.entschluesseleSymInformation(G, AN.i);
    end if
15: return Puffer.naechsteEntschluesselteInformation(G);
```

Algorithmus 4.5: Algorithmus des Teilnehmers im Schlüsselserver-Verfahren

4.5.3 Das Zeitserver-Verfahren

Im Schlüsselserver-Verfahren wird durch minimale Nachrichtenlänge der durch unterschiedliche Kapazität der Übertragungsabschnitte hervorgerufene Verzögerungsunterschied zwischen den Empfängern reduziert. Jedoch tragen unterschiedliche Warteschlangenlängen in den Routern zur Entstehung von zusätzlichen Verzögerungsunterschieden bei. Um dieser Ursache ebenfalls entgegenzuwirken, wird in diesem Abschnitt die Idee verfolgt, sichere Hardware direkt bei den Teilnehmern zu platzieren, um dort einen Vertrauensbereich zu etablieren [Klöcking et al. 2001b]. Dieses Vorgehen erlaubt, auch die

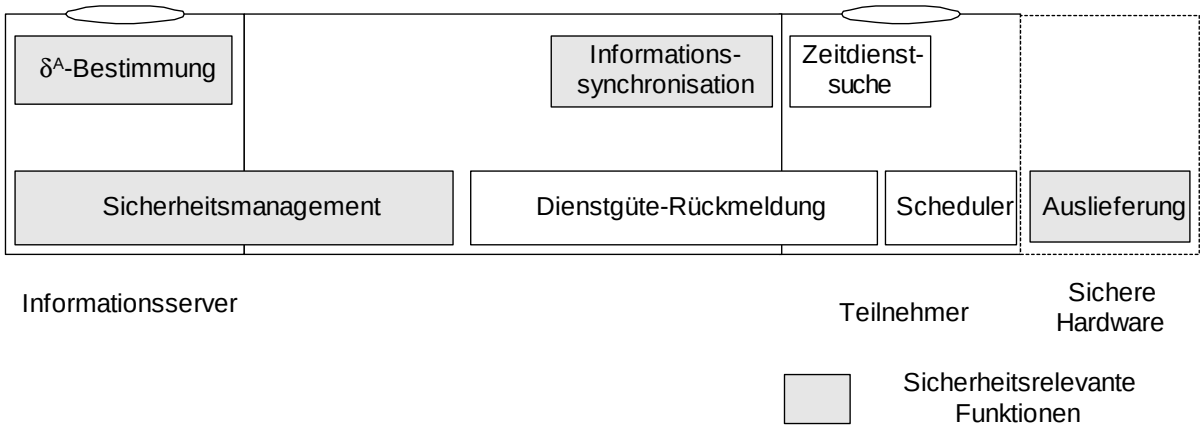


Abbildung 4.11: Funktionsbausteine des Zeitserver-Verfahrens

Schlüssel vor dem Auslieferungszeitpunkt zum Teilnehmer zu übertragen, um sie dort zum Auslieferungszeitpunkt sofort verfügbar zu haben. Die Funktion der Auslieferung wird von der sicheren Hardware übernommen (Abbildung 4.11).

Auch im Zeitserver-Verfahren ist es das Ziel der Algorithmen und Protokolle, ein Bekanntwerden der Information vor dem Auslieferungszeitpunkt zu verhindern. Beim Entwurf der Protokolle für dieses Verfahren ist besonders zu beachten, dass der Teilnehmer nun die gesamte Kommunikation zwischen dem Informationssender und dem Vertrauensbereich überwacht. Es muss sichergestellt werden, dass der Teilnehmer keine Nachrichten in die Kommunikation mit der sicheren Hardware einschleusen kann, z. B. um den Auslieferungszeitpunkt zu manipulieren oder sich durch Verzögerung der Nachrichten Vorteile zu verschaffen. Außerdem ist zu berücksichtigen, dass es sich bei sicherer Hardware um hochintegrierte Systeme handelt, die auf Kosten der Verarbeitungsleistung nur eine minimale Angriffsfläche für Manipulationen bieten. Die sichere Hardware kann daher nicht wie die Server im Datenserver-Verfahren die gesamte Information entschlüsseln, sondern sie wird wie im Schlüsselserver-Verfahren zur Entschlüsselung eines Informationsschlüssels verwendet.

Einen Überblick über das Verfahren gibt Abbildung 4.12. Der Informationssender verteilt die Informationen in verschlüsselten Informationsnachrichten an die Teilnehmer. Die Informationsnachricht wird zusammen mit der Schlüsselnachricht, die für die sichere Hardware bestimmt ist, versendet. Darin enthalten ist der Schlüssel zur Entschlüsselung der versandten Information. Er wird dem Teilnehmer zum Auslieferungszeitpunkt von der sicheren Hardware bekannt gegeben. Mit dem Ziel, die Uhren der sicheren Hardware mit

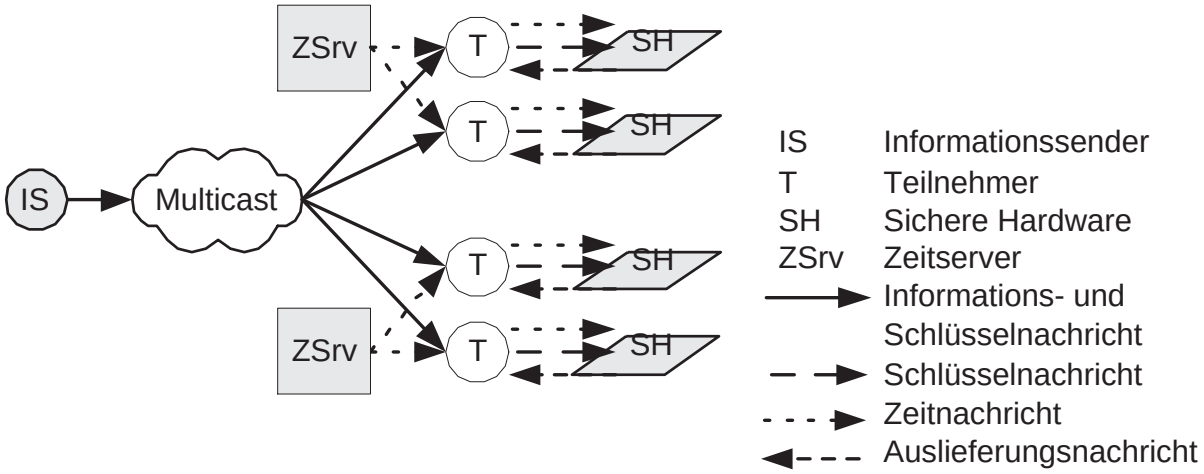


Abbildung 4.12: Grundlegendes Prinzip des Zeitserver-Verfahrens

einer Referenzzeit synchron zu halten, stellen die in der Abbildung dargestellten Zeitserver vertrauenswürdige Zeitinformation zur Verfügung.

Der Verzögerungsunterschied hängt davon ab, inwieweit die an die Teilnehmer angeschlossene sichere Hardware einen korrekten Auslieferungszeitpunkt bestimmt. Dies wiederum ist direkt von der Korrektheit der Uhren der sicheren Hardware abhängig. Mit dem im folgenden Abschnitt vorgestellten Protokoll sollen die Uhren synchronisiert werden.

4.5.3.1 Uhrensynchronisationsprotokoll

Uhren haben nur eine begrenzte Ganggenauigkeit und weichen von der Referenzzeit ab. Da Abweichungen der Uhren der sicheren Hardware zu unterschiedlichen Auslieferungszeitpunkten zwischen den Teilnehmern führen würden, sollen die Uhren der sicheren Hardware mit einem Uhrensynchronisationsprotokoll periodisch mit einer Referenzzeit synchronisiert werden. Um eine verfrühte Auslieferung zu verhindern, soll die Uhr jedoch auf keinen Fall vorgestellt werden können. Die Uhrzeit der sicheren Hardware soll ferner nicht mittels vertrauensunwürdiger Zeitinformation, die der Teilnehmer einschleusen könnte, manipulierbar sein.

Uhrensynchronisationsprotokolle in Netzwerken basieren auf der Annahme, dass die Verzögerung einer Nachricht vom Zeitserver zum Klienten und in Gegenrichtung gleich ist. Wegen des in Kapitel 4.1 beschriebenen Problems, die Verzögerung im Zugangsnetz zu

bestimmen, ist diese Annahme im vorliegenden Fall nicht erfüllt. Der Teilnehmer kann die Verzögerung so manipulieren, dass die Uhr auf der sicheren Hardware vorgeht. Baut der Teilnehmer ein Verzögerungselement in den von ihm zu einem Zeitserver führenden Abschnitt ein, stellt ein Uhrensynchronisationsprotokoll, das von symmetrischen Übertragungsabschnitten ausgeht, die Uhr der sicheren Hardware um die Hälfte des Betrages der Verzögerung des Elementes vor. Damit erzwingt der Teilnehmer eine verfrühte Auslieferung der Daten.

Die einzig sichere Möglichkeit, ein Vorstellen der Uhr zu verhindern, besteht darin, dass die sichere Hardware die empfangenen Zeitstempel jeweils als aktuellen Zeitpunkt übernimmt und die Verzögerung, die der Zeitstempel auf dem Weg vom Zeitserver erfahren hat, nicht durch Schätzungen ausgleicht. Damit ist garantiert, dass die Uhr der sicheren Hardware nicht vorgeht. Der Verzögerungsunterschied wird somit im Wesentlichen durch das Nachgehen der Uhren, d.h. die Verzögerung der Zeitnachrichten zu den Teilnehmern, bestimmt.

Die völlige Unabhängigkeit der Zeitnachrichten von den Informationsnachrichten wird ausgenutzt, um im Synchronisationsintervall mehrere Zeitstempel zu empfangen. Dadurch erhöht sich die Wahrscheinlichkeit, Zeitstempel zu erhalten, die nur eine geringe Verzögerung in den Warteschlangen der Router erfahren haben. Die Verzögerung der Zeitnachrichten wird auf diese Weise weiter reduziert, so dass der Verzögerungsunterschied auf der Übertragungsstrecke vom Server zum Teilnehmer geringer ausfällt als beim Datenserver- und beim Schlüsselserver-Verfahren. Hinzu kommt ein vorteilhafter Nebeneffekt: Der Zeitserver kann in einem Multicast- oder Broadcast-Modus arbeiten und wird somit nicht durch eine Vielzahl einzelner Synchronisationsanfragen überlastet.

Damit keine vertrauensunwürdige Zeitinformation von der sicheren Hardware akzeptiert wird, muss die Quelle der Zeitnachricht authentifizierbar sein. Für das in Abbildung 4.13 dargestellte Uhrensynchronisationsprotokoll wurden daher vertrauenswürdige Zeitserver gewählt. Eine Agentur überwacht die Zeitserver und zertifiziert die öffentlichen Schlüssel der Zeitserver (Abbildung 4.13(a)) über einen sicheren Kommunikationsweg. Der mit der sicheren Hardware ausgerüstete Teilnehmer tritt der Gruppenkommunikation bei, über die Zeitserver ihre Zeitstempel verteilen (Abbildung 4.13(b)). Der Teilnehmer bestimmt einen Zeitserver, von dem die Zeitnachrichten die geringste Verzögerung erfahren. Um unabhängig von Zeitserverausfällen zu sein, kann auch mehr als ein Zeitserver ausgewählt werden. Die Zeitserver geben ihre zertifizierten öffentlichen Schlüssel periodisch bzw. auf

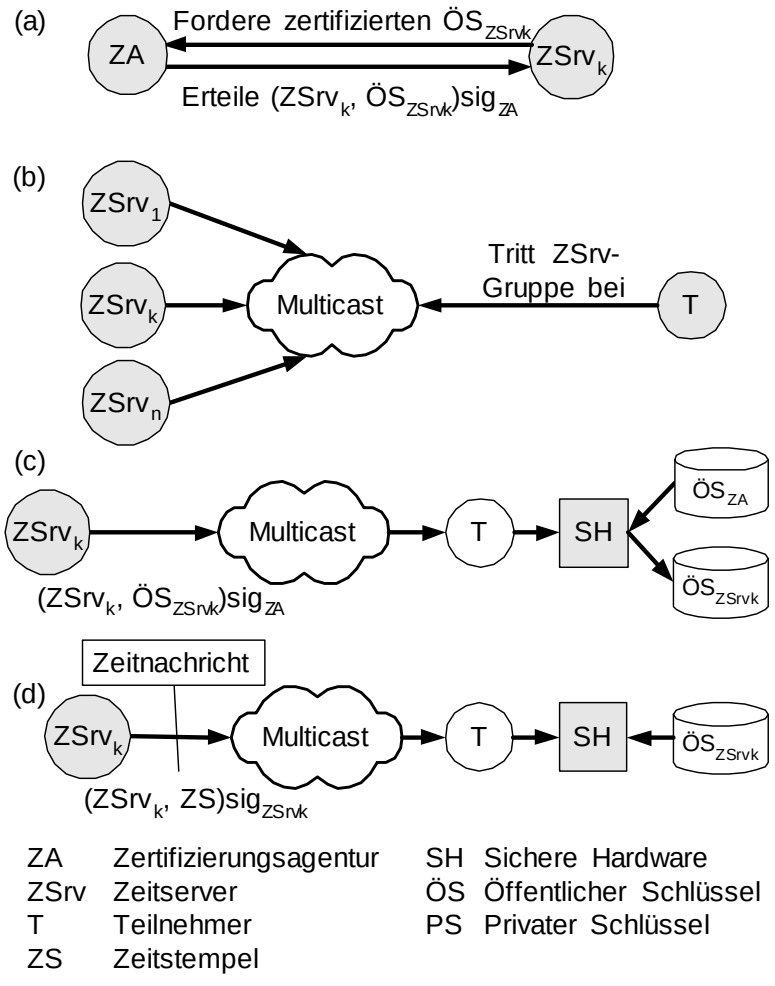


Abbildung 4.13: Uhrensynchronisationsprotokoll

Anforderung bekannt. Diese werden vom Teilnehmer an die sichere Hardware weitergeleitet (Abbildung 4.13(c)). Der Teilnehmer übergibt ebenfalls die von den Zeitservern mittels ihrer privaten Schlüssel authentifizierten Zeitstempel an die sichere Hardware (Abbildung 4.13(d)). Die sichere Hardware überprüft das Zertifikat mit dem öffentlichen Schlüssel des Zeitservers und stellt ihre Uhr entsprechend dem Zeitstempel. Der Zeitserver versendet periodisch die in Algorithmus 4.6 erstellten Nachrichten. Das Zeitdiagramm des Protokolls ist in Abbildung 4.15 dargestellt.

```
   method sendeZertifikat(G_ZN)
   ZertN := ZertifikatsNachricht.neu(Ring.Zertifikat);
   sende(G_ZN, ZertN);

   method sendeZeitnachricht(G_ZN)
5: ZN := ZeitNachricht.neu(ZSrv, t);
   ZN.signiere(Ring.PrivaterSchluessel);
   sende(G_ZN, ZN);
```

Algorithmus 4.6: Algorithmus des Zeitservers im Zeitserver-Verfahren

Eine positive Gangabweichung lässt die Uhr nach einer gewissen Zeitspanne so weit vorgehen, dass der Teilnehmer Vorteile gegenüber anderen Teilnehmern durch verfrühte Auslieferung der Informationen erhält. Die sichere Hardware beendet deshalb ihre Aktivität, wenn keine Synchronisationsnachrichten innerhalb des maximalen Synchronisationsintervalls τ^S (siehe Gleichung 4.1) eingetroffen sind.

Das Mitlesen der Zeitstempel ermöglicht dem Teilnehmer, die Zeitstempel, die eine besonders geringe Verzögerung erfahren haben, an die sichere Hardware weiterzuleiten, um eine geringe Abweichung von der Referenzzeit zu erzielen.

4.5.3.2 Synchronisationsintervall der Uhren der sicheren Hardware

Da die Ganggenauigkeit der Uhren auf der sicheren Hardware begrenzt ist, müssen sie von Zeit zu Zeit neu gestellt werden. Dadurch wird verhindert, dass der Zeitunterschied zwischen den Uhren so groß wird, dass einige Teilnehmer signifikante Vorteile durch verfrühte Bereitstellung des Nachrichtenschlüssels erlangen können. Als Sicherheitsmaßnahme hört die sichere Hardware daher mit der Auslieferung von Nachrichtenschlüsseln auf, wenn das maximale Synchronisationsintervall τ^S überschritten wird. Dieses Intervall lässt sich aus

der maximal tolerierten Gangabweichung a_{max} und der Ganggenauigkeit der Uhr ρ_{max} wie folgt berechnen:

$$\tau^S = \frac{a_{max}}{\rho_{max}} = \frac{1}{2} \cdot \frac{J_{Uhr}}{\rho_{max}} \tag{4.1}$$

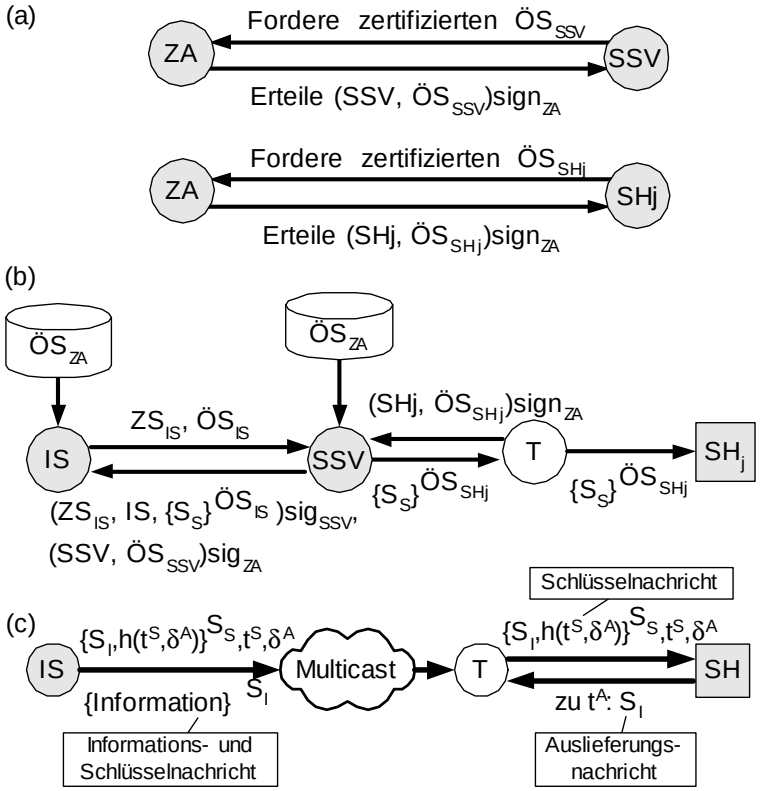
Die maximal tolerierte Gangabweichung der Uhr einer sicheren Hardware gegenüber der Referenzuhr darf nur halb so groß sein wie der tolerierte Verzögerungsunterschied J_{Uhr}, der von den Uhren verursacht wird.

Für Quarzuhren beträgt die Ganggenauigkeit typischerweise $\rho_{max} = 10^{-6}$. Wenn ein durch die Uhren verursachter Verzögerungsunterschied von 5 ms gestattet werden soll, darf der maximale Abstand zwischen zwei Zeitstempeln nicht größer sein als 40 Minuten. Bei einem maximalen Verzögerungsunterschied von 0.5 ms und gleicher Uhrengenauigkeit beträgt das Synchronisationsintervall dementsprechend 4 Minuten. Der Zeitserver sollte jedoch mehr als einen Zeitstempel pro Synchronisationsintervall versenden, zum Einen, um Paketverluste auszugleichen, zum Anderen, um dem Teilnehmer zu ermöglichen, einen Zeitstempel nahe der minimalen Einwegzeit zu verwenden.

4.5.3.3 Datenauslieferungsprotokoll

Das Datenauslieferungsprotokoll soll sicherstellen, dass die Informationen vor dem Auslieferungszeitpunkt zum Teilnehmer übermittelt werden können, jedoch erst zum Auslieferungszeitpunkt in verwertbarer Form vorliegen. Dies wird durch die verschlüsselte Versendung der Informationen sowie die Verteilung eines Informationsschlüssels über eine sichere Kommunikation zwischen Informationssender und sicherer Hardware erreicht. Die sichere Kommunikation wird durch den Dienstanbieter über einen Sitzungsschlüsselverwalter (SSV) aufgebaut. Damit der Sender von einem vertrauenswürdigen Dienst ausgehen kann, authentifiziert sich der SSV beim Sender. In einem vorbereitenden Schritt (a) lässt der Dienstanbieter daher von den von ihm erstellten asymmetrischen Schlüsselpaaren den öffentlichen Schlüssel des SSV und der sicheren Hardware von einer Zertifizierungsagentur mittels eines Zertifikats bestätigen (Abbildung 4.14(a)). Dies wird auf einem sicheren Kommunikationsweg außerhalb des Kommunikationsnetzwerkes durchgeführt.

Um dem Sender die Verteilung über Gruppenkommunikation zu erlauben und der sicheren Hardware eine ressourcenarme Entschlüsselung zu ermöglichen, wird für die sichere Kommunikation vom Sender eine symmetrische Verschlüsselung verwendet. Hierfür vergibt der SSV im Schritt (b) einen symmetrischen Sitzungsschlüssel (Abbildung 4.14(b)).



IS	Informationssender	ZA	Zertifizierungsagentur
T	Teilnehmer	SSV	Sitzungsschlüsselverwalter
SH	Sichere Hardware	ÖS	Öffentlicher Schlüssel
t^S	Sendezeitpunkt	PS	Privater Schlüssel
δ^A	Auslieferungsverzögerung	S_S	Sitzungsschlüssel
t^A	Auslieferungszeitpunkt	S_I	Informationsschlüssel
ZS	Zeitstempel		

Abbildung 4.14: Datenauslieferungsprotokoll beim Zeitserver-Verfahren

S_S-Anforderungsnachricht	:	IS $\rightarrow$ SSV	:	$ZS_{IS}, \ddot{O}S_{IS}$
Sitzungsschlüsselnachricht	:	SSV $\rightarrow$ IS	:	$(ZS_{IS}, IS, \{S_S\}^{\ddot{O}S_{IS}})sig_{SSV}$, $(SSV, \ddot{O}S_{SSV})sig_{ZA}$
S_S-Anforderungsnachricht	:	T $\rightarrow$ SSV	:	$(SH_j, \ddot{O}S_{SH_j})sig_{ZA}$
Sitzungsschlüsselnachricht	:	SSV $\rightarrow$ T	:	$\{S_S\}^{\ddot{O}S_{SH_j}}$
Sitzungsschlüsselnachricht	:	T $\rightarrow$ SH	:	$\{S_S\}^{\ddot{O}S_{SH_j}}$

Jeder beliebige Sender kann einen Sitzungsschlüssel für sich anfordern. Der SSV authentifiziert sich beim Informationssender in der Sitzungsschlüsselnachricht durch Signierung des Zeitstempels vom Informationssender. Der Sitzungsschlüssel wird vom SSV signiert und – mit dem öffentlichen Schlüssel des Senders verschlüsselt – zusammen mit dem Zertifikat des SSV an den Sender gesandt. Die Teilnehmer fordern für die sichere Hardware ebenfalls den Sitzungsschlüssel an. Dazu ist die Authentifizierung der sicheren Hardware gegenüber dem Dienstanbieter notwendig. Da dem Dienstanbieter der öffentliche Schlüssel der Zertifizierungsagentur bekannt ist, kann er das Zertifikat des öffentlichen Schlüssels der sicheren Hardware überprüfen. Nach Überprüfung des Zertifikats versendet der Sitzungsschlüsselverwalter den Sitzungsschlüssel verschlüsselt mit dem entsprechenden öffentlichen Schlüssel der sicheren Hardware an den Teilnehmer. Der Teilnehmer gibt den Schlüssel an die sichere Hardware weiter.

Im Schritt (c) wird nun die sichere Kommunikation zwischen Sender und sicherer Hardware zur Übermittlung eines Informationsschlüssels sowie von Sendezeit und Auslieferungsverzögerung für die an die Teilnehmer verschlüsselt gesendete Informationsnachricht genutzt 4.14(c)).

Informations- und Schlüsselnachricht		:	IS $\Rightarrow$ T	:	$\{I\}^{S_I}\|\|\{S_I, h(t^S, \delta^A)\}^{S_S}, t^S, \delta^A$
Schlüsselnachricht		:	T $\rightarrow$ SH	:	$\{S_I, h(t^S, \delta^A)\}^{S_S}, t^S, \delta^A$
Auslieferungsnachricht	$t \geq t^A$	:	SH $\rightarrow$ T	:	S_I

```
    method S_SN fordereSitzungsSchluessel(G_IN, SSV)
    ZS := t;
    S_SFN := SitzungsschluesselAnforderungsNachricht.neu(G_IN, ZS, Ring.OeS(IS));
    sende(SSV, S_SFN);
 5: S_SN := SitzungsschluesselNachricht.empfange(SSV);
    if (ZS ≠ S_SN.ZS) or (not S_SN.ueberpruefeIdentitaet(IS)) then
      behandleAusnahme('Unangeforderter Sitzungsschlüssel');
    end if
    if not S_SN.ueberpruefeSignaturZA(SSV, Ring.OeS(ZA)) then
10:   behandleAusnahme('Zertifikat ungültig');
    end if
    if not S_SN.ueberpruefeSignaturSSV(S_SN.OeS(SSV)) then
      behandleAusnahme('Signatur ungültig');
    end if
15: S_SN.entschluesseleSym(Ring.PS);
    return S_SN.S_S;

    method sendeVerzoegerungsfair(G, I)
    SSV := Sitzungsschluesselverwalter(G_IN(G));
    if not (Ring.S_S(G_IN(G), SSV)) then
20:   S_S := fordereSitzungsschluessel(G_IN(G), SSV);
      Ring.speichereSitzungsschluessel(G_IN(G), SSV, S_S);
    end if
    t^S := t;
    S_I := Informationsschluessel.zufaellig;
25: SN := SchluesselNachricht.neu(S_I, t_s, δ^A(G_IN(G)));
    SN.hashAuslieferungszeit;
    SN.verschluesseleSym(Ring.S_S(G_IN(G), SSV));
    IN := InformationsNachricht.neu(I);
    IN.verschluesseleSym(S_I);
30: sende(G_IN(G), IN||SN);
```

Algorithmus 4.7: Algorithmus des Informationssenders im Zeitserver-Verfahren

```
    method method SitzungsschluesselVergabe()
    SSFN := SitzungsschluesselAnforderungsNachricht.empfang;
    if SSFN.typ = IS then
      SSN := SitzungsschluesselNachricht.neu(SSFN.Zeitstempel, SSFN.Sender);
 5:   SS := SymmetrischerSchluessel.zufaellig();
      Ring.Sitzungsschluessel(GIN(SSFN.Sender), SS);
      SSN.verschluesseleASym(SSFN.OeS, SS);
      SSN.signiereNachricht;
      SSN.speichereZertifikat(Ring.Zertifikat);
10:   sende(SSFN.Sender, SSN);
    else if SSFN.typ = SH then
      if not SSFN.ueberpruefeZertifikat(Ring.Oes(ZA)) then
        behandleAusnahme('Zertifikat ungültig');
      end if
15:   if not SSFN.ueberpruefeIdentitaetstyp(SH) then
        behandleAusnahme('Falsche Identität');
      end if
      SSN := SchluesselNachricht.neu(Ring.Sitzungsschluessel(SSFN.G));
      SSN.verschluesseleAsym(SSFN.OeS);
20:   sende(SSFN.Sender, SSN);
    end if
```

Algorithmus 4.8: Algorithmus des Sitzungsschlüsselverwalters im Zeitserver-Verfahren

Die Schlüsselnachricht, die den mit dem Sitzungsschlüssel verschlüsselten Informationsschlüssel sowie die authentifizierte Sendezeit und Auslieferungsverzögerung enthält, reicht der Teilnehmer an die sichere Hardware weiter. Zur Auslieferungszeit stellt die sichere Hardware den Informationsschlüssel dem Teilnehmer in der Auslieferungsnachricht zur Verfügung, damit dieser an die Information gelangen kann. Der Informationsschlüssel kann nach Bekanntgabe nicht wieder verwendet werden und wird vom Sender für jede Information neu vergeben. Die Algorithmen 4.7, 4.8, 4.9, 4.10 verdeutlichen die sicherheitsrelevanten Aspekte der Nachrichtenverarbeitung aus Sicht des Informationssenders, des Schlüsselverwalters, des Teilnehmers sowie der sicheren Hardware.

```
    method beitritt(G)
    beitritt(G_ZN(G));
    beitritt(G_IN(G));

    method entschluesseleInformationsschluessel(index, SN)
 5: S_I := SH.entschluesseleSymInformationsschluessel(Puffer.Gruppe(index), SN);
    Puffer.entschluesseleSymInformation(index, S_I);

    method I empfange(G)
    if not SH.Sitzungsschluessel(G_IN(G)) then
      S_SFN := SitzungsschluesselAnfNachricht.neu(SSV(G_IN(G)),SH.OeS);
10:   sende(SSV(G_IN(G)), S_SFN);
      S_SN := SitzungsschluesselNachricht.empfange(SSV(G_IN(G)));
      SH.speichereSitzungsschluessel(G_IN(G), S_SN);
    end if
    if not SH.ZSrvZertifikat(G_ZN(G)) then
15:   ZertN := ZertifikatsNachricht.empfange(G_ZN(G));
      SH.speichereZSrvZertifikat(G_ZN(G), ZertN);
      ZN := ZeitNachricht.empfange(G_ZN(G));
      SH.stelleUhr(ZN);
    else if ZN := ZeitNachricht.empfangen(G_ZN(G)) then
20:   SH.stelleUhr(ZN);
    end if
    N := InformationsUndSchluesselNachricht.empfange(G_IN(G));
    index := Puffer.speichereInformationsNachricht(G_IN(G), N.IN);
    schedule (N.t^S + N.δ^A, entschluesseleInformationsschluessel(index, N.SN));
25: return Puffer.naechsteEntschluesselteInformation(G_IN(G));
```

Algorithmus 4.9: Algorithmus des Teilnehmers im Zeitserver-Verfahren

Das Zeitsynchronisationsprotokoll sowie das Datenauslieferungsprotokoll sind als Zeitdiagramm in Abbildung 4.15 zusammen dargestellt. Das Diagramm beinhaltet die Aktion des Senders, einen Sitzungsschlüssel anzufordern und die Aktionen des Teilnehmers, einer Zeitserver-Gruppe beizutreten, vom Sitzungsschlüsselverwalter den Sitzungsschlüssel anzufordern sowie der Informations-Gruppe beizutreten. Der Teilnehmer empfängt danach vom Zeitserver periodisch Zeitnachrichten sowie vom Informationssender die verschlüsselten Informations- und Schlüsselnachrichten. Der Teilnehmer leitet die Zeitnachrichten

und Schlüsselnachrichten an die sichere Hardware weiter. Zum Auslieferungszeitpunkt stellt die sichere Hardware den Informationsschlüssel bereit.

4.5.3.4 Sicherheitsbetrachtung der Algorithmen

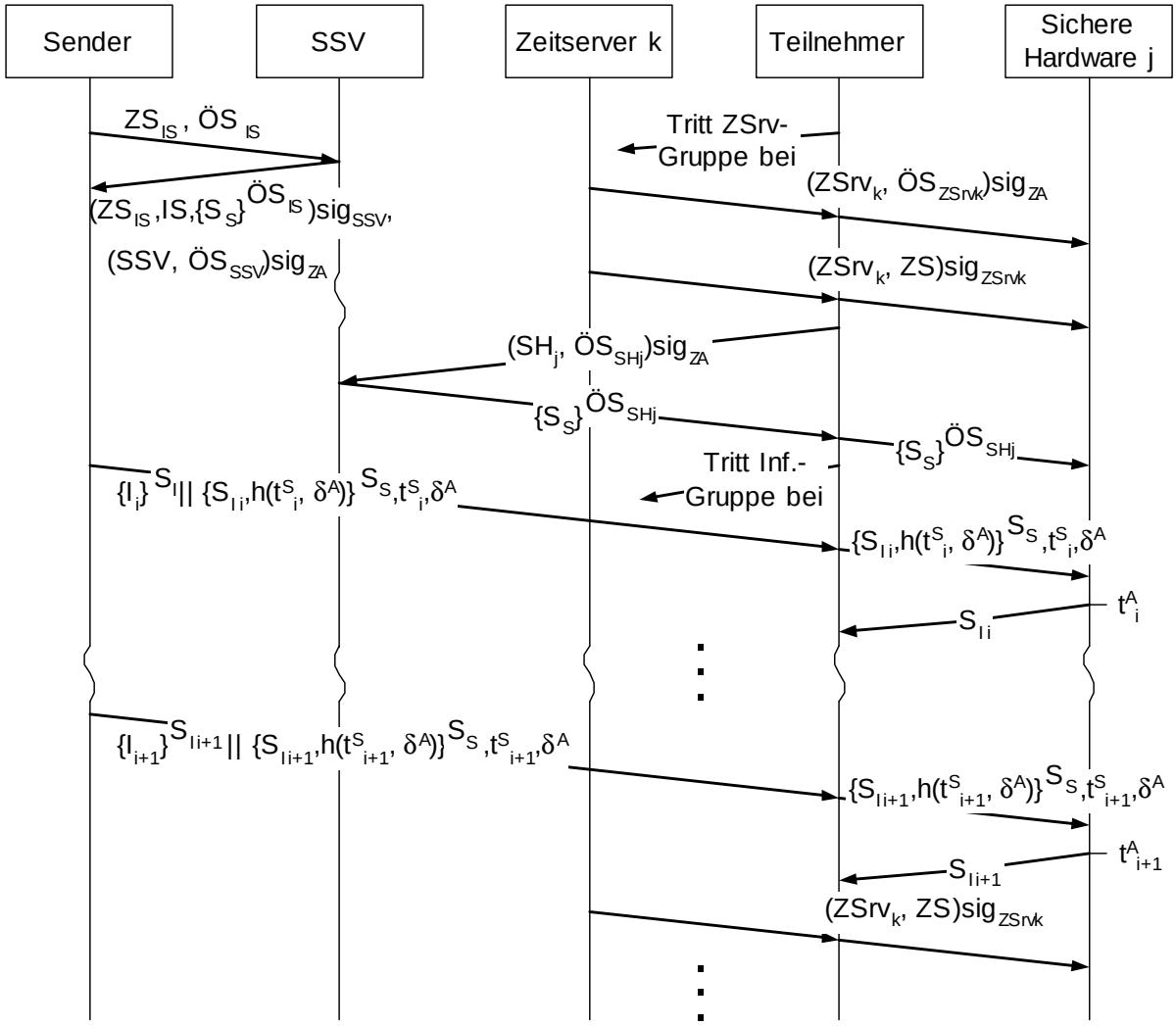
Das im Abschnitt 4.4.1.3 angenommene allgemeine Angriffsmodell erlaubt einem Angreifer, Nachrichten abzufangen, wiederholt zu senden, zu fälschen, zu verändern, zu blockieren, einzufügen sowie als legitimer Teilnehmer aufzutreten. Daraus lassen sich insbesondere die im Folgenden beschriebenen Angriffszenarien für die Protokolle ableiten.

Abgesehen von Angriffen auf die Verfügbarkeit des Dienstes, besteht das grundlegende Ziel eines Angreifers darin, an die Informationen früher als andere Teilnehmer zu gelangen. Um dieses Ziel zu erreichen, kann er prinzipiell zwei Wege verfolgen: entweder er versucht, die Daten zu entschlüsseln oder er bemüht sich, die Uhrzeit zu beeinflussen.

Um die Daten zu entschlüsseln, muss ein Angreifer an den Informationsschlüssel, den Sitzungsschlüssel oder an die asymmetrischen Schlüssel von Schlüsselverwalter oder sicherer Hardware gelangen. Da öffentliche Schlüssel bekannt sein dürfen und private Schlüssel nicht ausgetauscht werden, gelangt ein Angreifer nicht an relevante asymmetrische Schlüssel.

Ein Angreifer kann für sich selbst einen symmetrischen Sitzungsschlüssel anfordern und versuchen, ihn einem Informationssender weiterzugeben. Er kann jedoch nicht den für sich angeforderten Sitzungsschlüssel für den Informationssender neu verschlüsseln, ohne dass gleichzeitig die Signatur der Nachricht ungültig wird. Eine vom Angreifer erzeugte Signatur wäre ebenfalls ungültig. Der Angreifer kann auch einen Sitzungsschlüssel unter Vorspiegelung der Identität eines Informationssenders anfordern, ihn dann aber nicht entschlüsseln. Wiederholtes Senden eines alten, ggf. kompromittierten Sitzungsschlüssels wird vom Informationssender durch Vergleich des Zeitstempels erkannt.

Ein Angreifer kann außerdem der sicheren Hardware einen beliebigen Sitzungsschlüssel senden, daraus jedoch keinen Nutzen ziehen, da die sichere Hardware damit die Informationsschlüssel nicht entschlüsseln kann. Dasselbe trifft auch für Wiederholungsangriffe des Sitzungsschlüssels auf die sichere Hardware zu. Da die sichere Hardware selbst keine Informationen versendet, gilt analog, dass sie die Aktualität des Sitzungsschlüssels nicht überprüfen muss.



ZA	Zertifizierungsagentur	T	Teilnehmer
SSV	Sitzungsschlüsselverwalter	SH	Sichere Hardware
ÖS	Öffentlicher Schlüssel	t^S	Sendezeitpunkt
PS	Privater Schlüssel	δ^A	Auslieferungsverzögerung
S_S	Sitzungsschlüssel	t^A	Auslieferungszeitpunkt
S_I	Informationsschlüssel	ZS	Zeitstempel

Abbildung 4.15: Zeitdiagramm der Protokolle für das Zeitserver-Verfahren

```
    method Schluessel OeS() return Ring.OeS(SH);

    method Zeit Uhrzeit() return t;

    method LogischerWert SitzungsschluesselGespeichert(G)
    return Ring.SitzungsschluesselGespeichert(G);

    method LogischerWert ZSrvZertifikatGespeichert(G)
    return Ring.ZSrvZertifikatGespeichert(G);

    method speichereSitzungsschluessel(G, S_SN);
    S_SN.entschluesseleAsymSitzungsschluessel(Ring.PS);
    Ring.speichereSitzungsschluessel(G, S_SN.S_S);

    method schluessel entschluesseleSymInformationsschluessel(SN)
    if t - t_Synch > τ^S then
      behandleAusnahme('Uhrensynchronisationsintervall überschritten');
    end if
    SN.entschluesseleSym(Ring.Sitzungsschluessel);
    if not SN.ueberpruefeGueltigkeitAuslieferungszeit then
      behandleAusnahme('Auslieferungszeit gefälscht');
    end if
    if t < SN.t^S + SN.δ^A then
      behandleAusnahme('Auslieferungszeit nicht erreicht');
    end if
    return SN.S_I;

    method speichereZSrvZertifikat(G, ZertN)
    if not (ZertN.ueberpruefeZertifikat(Ring.OeS(ZA))) then
      behandleAusnahme('Zertifikat ungültig');
    end if
    Ring.speichereZSrvZertifikat(G, ZertN.ZSrvZertifikat);

    method stelleUhr(ZN)
    if not ZN.ueberpruefeSignatur(Ring.ZSrvZertifikat(ZN.Gruppe) then
      behandleAusnahme('Zeitstempel gefälscht');
    end if
    t := ZN.ZS;
    t_Synch := t;
```

Algorithmus 4.10: Algorithmus der sicheren Hardware im Zeitserver-Verfahren

Der Informationsschlüssel selbst ist über Sitzungsschlüssel gesichert. Er wird dem Angreifer daher erst zur Zeit $t \geq t_a$ der Uhr der sicheren Hardware zugängig.

Sendezeitpunkt und Auslieferungsverzögerung der Nachricht können ebenfalls durch einen Angreifer verändert werden. Da jedoch diese Zeitangaben über einen Hash-Wert an die Information gebunden sind, verliert dieser seine Gültigkeit, so dass der Angriff von der sicheren Hardware detektiert wird.

Die Alternative für den Angreifer besteht darin, die Uhr der sicheren Hardware zu verstellen, um den Informationsschlüssel früher zu erlangen. Dazu kann er entweder versuchen, die Uhr einer sicheren Hardware vorzustellen oder die Uhr der sicheren Hardware aller anderen Teilnehmer zurückzustellen. Das Zurückstellen einer Uhr der sicheren Hardware lässt sich durch Verzögern der Zeitstempel-Nachrichten einfach realisieren. Das Zurückstellen der Uhren aller anderen Teilnehmer ist aufgrund der Vielzahl der Teilnehmer, der vertrauenswürdigen Zeitserver sowie der Zugangsmöglichkeiten der Teilnehmer zu diesen Zeitservern mit hohen Kosten verbunden und daher nicht praktikabel. Ein Zurückstellen einzelner Uhren stellt wiederum ein Angriff auf die Verfügbarkeit des Dienstes dar.

Zum Vorstellen der Uhren der sicheren Hardware bieten sich entsprechend dem Angriffsmodell die folgenden Möglichkeiten. Ein Abfangen der Nachrichten durch den Teilnehmer als potenziellen Angreifer ist erlaubt. Dies eröffnet ihm jedoch nur die erwünschte Möglichkeit, den aktuellsten Zeitstempel auszuwählen. Wiederholungen alter Zeitstempel werden von der sicheren Hardware zwar akzeptiert, jedoch stellen diese die Uhren der sicheren Hardware nur zurück. Gefälschte, veränderte Zeitstempel werden von der sicheren Hardware durch Verifizierung der Signatur erkannt und abgewiesen. Das Verzögern der Zeitstempel führt nur zu einem Zurückstellen der Uhr, da die Umlaufzeit der Zeitnachrichten nicht berücksichtigt wird. Ein vollständiges Blockieren der Zeitstempel durch den Angreifer über das maximale Synchronisationsintervall hinaus in der Hoffnung, dass die Uhr der sicheren Hardware vorgeht, hat zur Folge, dass die sichere Hardware die Entschlüsselung der Informationsschlüssel einstellt. Zusätzlich eingefügte Zeitstempel sind entweder eine Wiederholung oder gefälscht. Ein Angreifer kann nicht als Zeitserver auftreten, da er keinen zertifizierten Schlüssel mit Zeitserveridentität erlangen kann bzw. die Zeitserver gemäß den Voraussetzungen vertrauenswürdig sind. Eine Manipulation zugunsten eines Teilnehmers ist daher nicht möglich, da das Vorstellen der Uhr verhindert und die Authentizität der Zeitinformation überprüft wird.

4.5.3.5 Smart Cards als sichere Hardware beim Teilnehmer

Wegen des hohen Sicherheitsstandards und der niedrigen Kosten werden Smart Cards für die Implementierung der sicheren Hardware vorgeschlagen. Smart Cards sind scheckkartengroße Plastikkarten, auf denen sich ein Prozessor, Speicher und ein Betriebssystem befinden. Smart Cards sind sehr gut gegen Manipulationen geschützt. Prozessor und Speicher befinden sich innerhalb desselben *Die*, was das Mitprotokollieren des Signalaustauschs erschwert. Insbesondere sind Smart Cards gegen wichtige mögliche Angriffe gefeit, wie das Abtragen von einzelnen Schichten der Smart Card, um optisch Daten auszulesen, die Manipulation der Spannung oder Taktrate, um den Prozessorfehler auszulösen, Angriffe mittels hoher Temperaturen oder Röntgenstrahlen und die Überwachung der Stromversorgung der Karte [Hansmann et al. 2000]. Sie eignen sich daher besonders als sichere Hardware zur Speicherung von vertraulichen Informationen und zur Ausführung sicherheitsrelevanter Programme. Für eine Manipulation ist ein so extrem hoher Aufwand nötig, dass er praktisch ausgeschlossen werden kann. In einem asymmetrischen Verschlüsselungsschema ist die Smart Card ideal zur Speicherung des privaten Schlüssels und zur Authentifizierung von Daten geeignet. Der Systemtakt der Karte wird bei manchen Karten von einem externen Takt, den der Kartenleser zur Verfügung stellt, abgeleitet. Für die hier diskutierte Anwendung besteht in diesem Fall die Gefahr eines Angriffes durch Höhertaktung der Karte, um früher an die gewünschten Informationen zu kommen. Dies kann durch Verwendung von Smart Cards mit internen Taktquellen verhindert werden [Moore et al. 2002].

Für das in Abschnitt 4.5.3.1 vorgestellte Protokoll muss die sichere Hardware folgende Anforderungen erfüllen:

- Nachrichtenentschlüsselung mittels symmetrischer und asymetrischer Verschlüsselungsverfahren
- Aufbewahrung von Schlüsseln
- Ausführung von Programmen (z. B. für Zeitvergleich von Auslieferungszeit und interner Uhr)
- Uhr mit ausreichender Ganggenauigkeit

Gegenwärtige Smart Cards offerieren symmetrische und asymmetrische Verschlüsselungsverfahren, DES und RSA, mit Schlüssellängen von 512 bis 1024 Bit. Die Verschlüsselungsverfahren werden durch einen speziellen Koprozessor ausgeführt. Die Kommunikation mit

der Smart Card erfolgt über Smart-Card-Lesegeräte, die an die serielle Schnittstelle angeschlossen werden. Programme können je nach Smart Card in Basic, C oder Java erstellt werden. Mit der Verwendung von so genannten Java Cards sind die Vorteile von Java, wie Objektorientierung, Ausnahmebehandlung und Typenschutz auch für die Softwareentwicklung für Smart Cards gegeben. Von Sun Microsystems wurde eine Untermenge der Java API und Java Virtual Machine für Smart Cards spezifiziert, um deren knappen Ressourcen Rechnung zu tragen.

Smart Cards können somit die Anforderungen bezüglich Entschlüsselung, Schlüsselaufbewahrung und Programmausführung erfüllen. Derzeit unterstützen Smart-Card-Betriebssysteme noch keine Uhren oder die Verwendung von Timern. Andererseits sind schon flache Batterien [Varta AG 2004] für Smart Cards entwickelt worden und die Integrationsmöglichkeiten von Uhren wird untersucht [GEMPLUS S.A. 2002].

4.6 Administrative Maßnahmen zur Reduktion des Verzögerungsunterschiedes

Wie in Abschnitt 4.1 gezeigt, kann die im Zugangsnetz auftretende Verzögerung nicht sicher bestimmt werden. In einigen Fällen erstreckt sich jedoch der Vertrauensbereich über die Grenzen des Servers hinaus und kann voll zur Reduktion der Verzögerungsunterschiede zwischen den Teilnehmern ausgeschöpft werden. Zwei Beispiele sollen dies verdeutlichen. Eine bekannte Verzögerung zum Teilnehmer auf einer Übertragungsstrecke kann durch vorzeitige Versendung der Nachrichten ausgeglichen werden. Dies ist beispielsweise der Fall, wenn auch Satellitenübertragungsstrecken zur Auslieferung verwendet werden. Falls die Sendestation zum Vertrauensbereich gehört, kann die Verzögerung der Übertragung bis zur Erdoberfläche (bei geostationären Satelliten 240 ms) berücksichtigt werden. Für Nachrichten, die nur nach einem vollständigen Empfang von Nutzen sind, kann auch die Serialisierungsverzögerung auf den aus dem Vertrauensbereich abgehenden Übertragungsabschnitten ausgeglichen werden. Zum Beispiel können signierte Zeitstempel um die durch Serialisierung der Nachricht auf einem Übertragungsabschnitt geringer Übertragungskapazität entstehende Verzögerung früher versendet werden. Der Zeitstempel ist frühestens zu der im Zeitstempel bezeichneten Zeit beim Teilnehmer. Da die sichere Hardware die vollständige Signatur zur Authentifizierung des Zeitstempels benötigt, ist die Nachricht vor ihrem vollständigen Empfang wertlos. Die Auslieferungsverzögerung wird vom Sender so vergrößert, dass genügend Zeit für eine vorgezogene Weitergabe der Nachrichten bleibt.

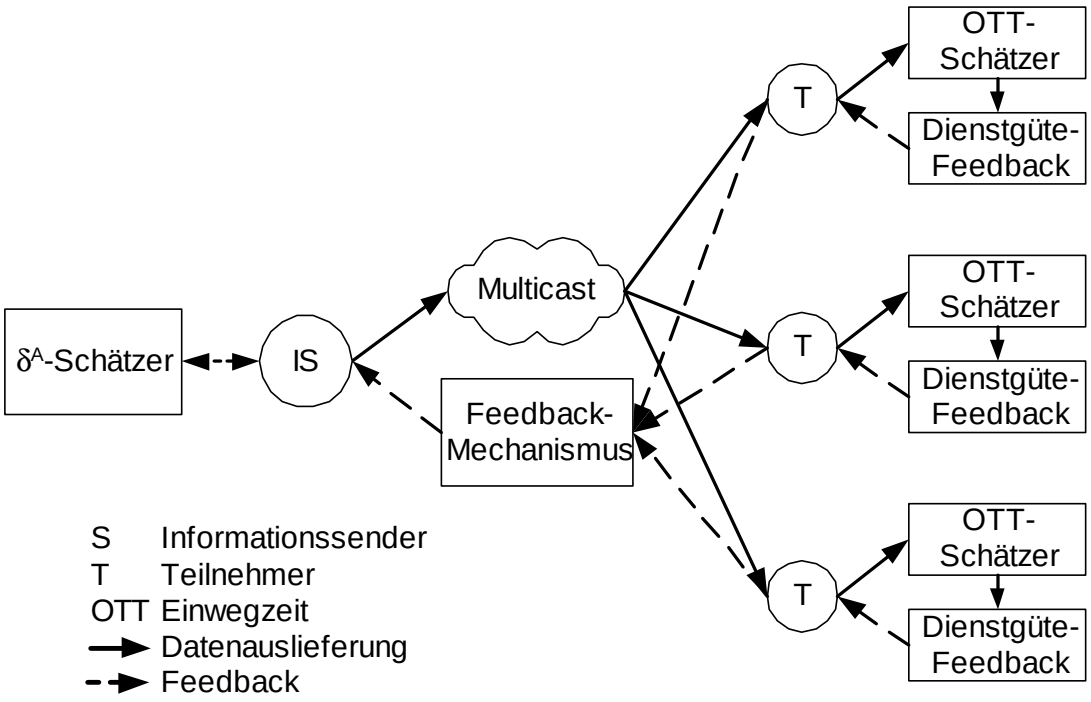


Abbildung 4.16: Modell zur Reduktion der Gesamtverzögerung der Nachrichten

4.7 Minimierung der Gesamtverzögerung der Nachrichten

Die bestmögliche Reduktion des Verzögerungsunterschiedes zwischen den Empfängern der Nachrichten wird erreicht, wenn die Auslieferungsverzögerung auf einen so hohen Wert festgelegt wird, dass ihn die Übertragungszeit der Nachrichten vom Sender zu den Vertrauensbereichen garantiert nicht überschreitet. Einige Anwendungen mit kontinuierlichem Informationsfluss fordern jedoch sowohl einen geringen Verzögerungsunterschied als auch eine geringe Gesamtverzögerung der Übertragung. Durch dynamische Anpassung der Auslieferungsverzögerung an die aktuelle Transferzeit der Nachrichten wird auch dieses Ziel erreicht.

Geben die Empfänger der Nachricht dem Sender eine Rückmeldung über die Verzögerung der Nachricht, kann der Sender dies als Grundlage für die Festsetzung der zukünftigen Auslieferungsverzögerung verwenden. Abbildung 4.16 gibt einen Überblick über die im Folgenden beschriebenen Komponenten.

OTT-Schätzer. Da in den Informations- bzw. Schlüsselnachrichten die Sendezeit enthalten ist, sind die Empfänger in der Lage, die Einwegzeit (*One-Trip Time – OTT*) der Nachricht zu bestimmen. Durch die in Abschnitt 3.1 beschriebenen Ursachen weisen die Nachrichten sehr unterschiedliche Einwegzeiten auf, so dass über die Angabe der aktuellen Einwegzeit hinaus eine Schätzung der OTT für die folgenden Nachrichten zweckmäßig ist. Hierfür wird ein exponentiell gewichteter gleitender Durchschnitt (*Smoothed One Trip Time – SOTT*) aus der aktuellen Einwegzeit OTT_{i+1} und vorangegangenen Einwegzeiten $SOTT_i$ berechnet:

$$SOTT_{i+1} = (1 - \alpha_1) \cdot SOTT_i + \alpha_1 \cdot OTT_{i+1}$$

Zur Verbesserung der Schätzung wird darüber hinaus noch die Varianz der Einwegzeit $OTTVAR$ berücksichtigt:

$$OTTVAR_{i+1} = (1 - \alpha_2) \cdot OTTVAR_i + \alpha_2 \cdot |SOTT_{i+1} - OTT_{i+1}|$$

Die Parameter α_1 und α_2 beeinflussen das Ausmaß, in dem vorangegangene Einwegzeiten bei der Berechnung des Durchschnittes berücksichtigt werden. Die gewünschte Auslieferungsverzögerung ergibt sich dann als

$$\delta^A = SOTT + k \cdot OTTVAR$$

Der Parameter k bestimmt die Defensivität der Schätzung. Je höher k gewählt wird, desto stärker wird die Varianz berücksichtigt und desto unwahrscheinlicher ist es folglich, Nachrichten nach Ablauf der Auslieferungsverzögerung zu erhalten.

Feedback-Mechanismus. Die Empfänger teilen über einen Feedback-Mechanismus dem Sender ihre gewünschte Auslieferungszeit mit. Prinzipiell können Empfänger einer Multicast-Nachricht Feedback mittels Unicast an den Sender, mittels Multicast an die Gruppe oder über einen hierarchischen Mechanismus an den Sender schicken. Nur ein hierarchisches Verfahren garantiert jedoch Skalierbarkeit für eine sehr hohe Zahl von Empfängern. Bietet das Netzwerk einen hierarchischen Dienst, wie z. B. Concast [Calvert & Griffioen 2000], wird dieser verwendet. Anderenfalls wird auf den Endsystemen eine Hierarchie nachgebildet. Dazu wird ein auf minimale Verzögerung optimierter Baum mittels des *Token Repository Service* [Maihöfer & Rothermel 1999, Maihöfer 2001] erstellt. Die Knoten berechnen die jeweils maximal gewünschte Auslieferungsverzögerung und leiten

diese Information zur übergeordneten Hierarchieebene und letztendlich zum Sender weiter.

Verzögerungsunterschied-Schätzer. Der Sender passt entsprechend dem erhaltenen Feedback δ^A_{FB} die Auslieferungsverzögerung an die aktuell gewünschte Auslieferungszeit an. Damit einzelne Empfänger die Angabe des Feedback nicht zur absichtlichen Beeinträchtigung der Dienstqualität missbrauchen können, wird ein Maximalwert für die Auslieferungsverzögerung δ^A_{max} berücksichtigt. Diesen legt der Sender entsprechend den Anforderungen an die maximal tolerierbare Verzögerung für den Dienst bzw. entsprechend den aus der Vergangenheit bekannten maximalen Verzögerungen fest.

$$\delta^A = \min\{\delta^A_{max}, \delta^A_{FB}\}$$

4.8 Fairness des Informationsempfanges

Neben dem fairen Informationszugang ist bei den Anwendungsklassen 2 bis 4 auch eine faire Interaktion mit dem System notwendig. Die Anwendungsklassen 2 und 4 verlangen, die Reihenfolge der Aktionen der einzelnen Teilnehmer fair zuzuordnen. Die Teilnehmer von Anwendungen mit geschlossenem, diskretem Informationsfluss (Klasse 3) unterscheiden sich untereinander durch ihre Reaktionszeit. Während der Teilnehmer bei der Informationsverteilung bestrebt ist, möglichst früh an die Information zu gelangen, besteht sein Ziel bei der Interaktion mit Informationssystemen darin, eine möglichst große Bedenkzeit zu haben und die Nachricht – für das System unbemerkt – später als andere Empfänger zu senden. In diesem Abschnitt wird erörtert, welche Verfahren auf diese Anwendungsklassen anwendbar sind.

4.8.1 Anwendungen mit offenen, unidirektionalen Kommunikationsbeziehungen zum Informationsempfang

Besteht die Fairness bei Klasse-1-Anwendungen im *gleichzeitigen* Empfang der von einer Informationsquelle an die Teilnehmer von Gruppenkommunikation gesendeten Nachrichten, so geht es bei Anwendungen der Klasse 2 um das Gegenteil, die Übermittlung der Nachrichten an das Informationssystem in der *differenzierten* Reihenfolge entsprechend

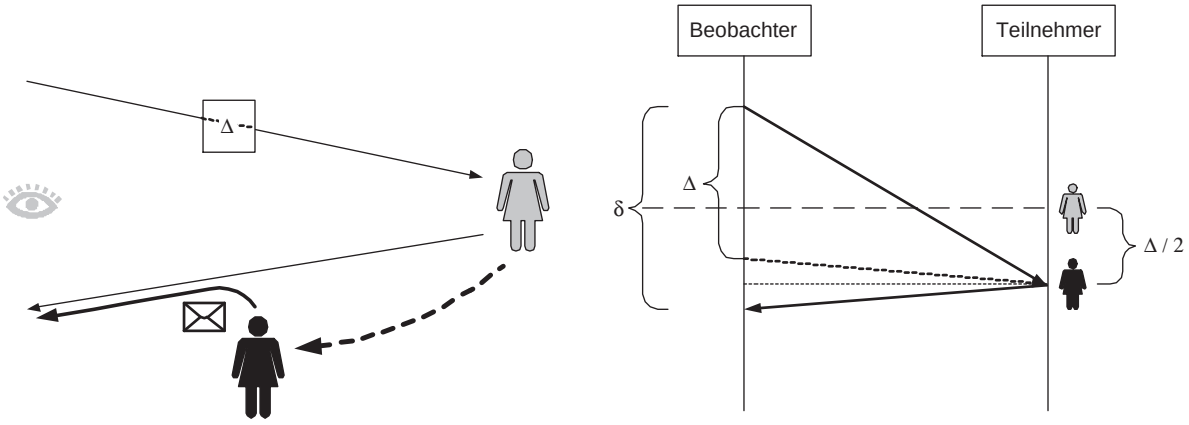


Abbildung 4.17: Vorspiegelung einer höheren Verzögerung beim Senden von Nachrichten

den von den Teilnehmern gewählten unterschiedlichen Sendezeitpunkten. Eine faire Interaktion mit dem System wäre somit dann gegeben, wenn die Nachrichten der Teilnehmer in der Reihenfolge ihres Sendezeitpunktes Berücksichtigung fänden.

Auch bei dieser Anwendung sind bestimmte Maßnahmen zu treffen, damit das System die erforderliche Fairness erbringen kann. Das ist darauf zurückzuführen, dass das System die Uhrzeit beim Teilnehmer und damit die Reihenfolge des Sendens nicht sicher bestimmen kann. Angenommen, die Zeit würde wieder durch ein Uhrensynchronisationsprotokoll bestimmt. Der Angreifer aus dem Beispiel in Abschnitt 4.1 wird in diesem Fall sein Verzögerungselement Δ auf dem Weg vom Server zum Teilnehmer platzieren (Abbildung 4.17). Dadurch wird er – bei einer für den Beobachter gleichen Sendezeit – in die Lage versetzt, eine Gebotsnachricht an das Informationssystem $\Delta/2$ Zeiteinheiten später als andere Teilnehmer zu versenden, um seine Bedenkzeit zu vergrößern. Alternativ ist es dem Angreifer möglich, sich bei unveränderter Bedenkzeit einen Vorsprung von $\Delta/2$ Zeiteinheiten in der beobachteten Sendereihenfolge zu sichern. Dies führt beispielsweise bei der holländischen Auktion, bei der der Auktionspreis von einem Höchstpreis bis zum ersten Kaufsignal eines Auktionsteilnehmers herabgesetzt wird, zu einer unfairen Vergabe des Auktionsgegenstandes.

Da dem Teilnehmer nicht vertraut werden kann, sind auch hier Vertrauensbereiche zur Lösung des Problems notwendig. Analog zum Datenserver-Verfahren der Informationsverteilung kann der Sendezeitpunkt unter Vernachlässigung des Verzögerungsunterschiedes zwischen Teilnehmern und Servern auf die Ankunftszeit der Nachrichten bei Datenservern abgebildet werden. Den Teilnehmern wird die Verantwortung für die Wahl eines Netzzu-

ganges mit geringer Verzögerung übertragen. Folgendes Protokoll kann dafür verwendet werden:

$$\begin{array}{lcll} \text{N1} & : & \text{T} \rightarrow \text{DSrv} & : \quad G \\ \text{N2} & : & \text{DSrv} \rightarrow \text{IE} & : \quad (i, t_S, G) sig_{DSrv} \end{array}$$

Ein Teilnehmer T sendet sein Gebot G an einen Datenserver $DSrv$. Der Datenserver bestimmt die Ankunftszeit der Nachricht ($N1$) und weist diese als Sendezeit t_S des Gebotes aus. Durch Signierung assoziiert er die Sendezeit mit dem Gebot und einer laufenden Nachrichtennummer i und sendet dies als Nachricht ($N2$) zum Informationsempfänger IE. Die Nachrichtennummer, die der Server vergibt, dient zur Entdeckung von Angriffen auf die Kommunikation zwischen Datenserver und Informationsempfänger. Ohne die Nachrichtennummer könnte ein Angreifer eine Nachricht vorab auf Verdacht signieren lassen, auf dem Weg vom Datenserver zum Informationsempfänger abfangen und erst dann versenden, wenn sie benötigt wird (dann jedoch mit eindeutigem Zeitvorteil gegenüber den anderen Teilnehmern).

Neben der Verwendung von Datenservern ist auch hier wieder der Einsatz sicherer Hardware beim Teilnehmer denkbar, die die Signierung der zu versendenden Nachrichten übernimmt. Um eine größere Bedenkzeit zu verhindern, muss in diesem Fall gefordert werden, dass die Uhr der sicheren Hardware vom Teilnehmer nicht zurückgestellt werden kann. Ferner müssen einmal signierte Nachrichten dem Informationsempfänger zugestellt werden; denn wäre ein Nachrichtenverlust von signierten Nachrichten möglich, könnte ein Teilnehmer Nachrichten vorab signieren lassen und, um die Bedenkzeit zu erhöhen, nur im Bedarfsfall versenden.

Um ein Zurückstellen der Uhr zu verhindern, wird das Uhrensynchronisationsprotokoll für diesen Anwendungsfall so entworfen, dass die Uhr des Servers nach der Uhr der sicheren Hardware gestellt wird. Dazu generiert die sichere Hardware periodisch signierte Zeitstempel. Der Teilnehmer leitet diese an einen Zeitserver weiter. Der Zeitserver bestimmt daraufhin den Zeitversatz Θ_{SH} der sicheren Hardware.

$$\begin{array}{lcll} \text{N1} & : & \text{SH} \rightarrow \text{ZSrv} & : \quad (SH, t_S) sig_{SH} \qquad : \quad \Theta_{SH} = t - t_S \\ \text{N2} & : & \text{ZSrv} \rightarrow \text{SH} & : \quad (SH, t_S, \Theta_{SH}) sig_{ZSrv} \end{array}$$

Die Aktualität der Zeitstempel verhindert Angriffe durch Wiederholung der Nachrichten. Hierfür muss die sichere Hardware eine fortlaufende Zeit ab Aufbringen der Anwendung auf die sichere Hardware erzeugen. Das ist möglich, wenn sie einen nicht-flüchtigen, sicher beschreibbaren Speicher besitzt. Alternativ kann dies auch über eine Initialisierungsnachricht des Zeitservers mit einem aktuellen Zeitstempel erfolgen.

Um Informationen zu versenden, bildet der Teilnehmer einen Hash-Wert über sein Gebot und übergibt dieses der sicheren Hardware in $N1$. Die sichere Hardware bestimmt die Sendezeit durch Addition des Zeitversatzes Θ_{SH} zur Uhrzeit der sicheren Hardware. Danach signiert sie die Sendezeit, eine laufende Nachrichtennummer und den Hash-Wert des Gebotes und übermittelt dies in $N2$ an den Teilnehmer. Der Teilnehmer ergänzt das Gebot und sendet $N3$ an den Informationsempfänger.

$$\begin{array}{lllll} \text{N1} & : \text{T} \rightarrow \text{SH} & : h(G) & : & t_S = t + \Theta_{SH} \\ \text{N2} & : \text{SH} \rightarrow \text{T} & : (i, t_S, h(G))sig_{SH} & & \\ \text{N3} & : \text{T} \rightarrow \text{IE} & : G, (i, t_S, h(G))sig_{SH} & & \end{array}$$

Zur Endeckung von Angriffen durch absichtlichen Nachrichtenverlust ist das Protokoll in eine Eröffnungsphase, in der die Teilnehmer registriert werden, und in eine Schlussphase, in der die fortlaufende Nachrichtennummer abschließend überprüft wird, einzubetten. Die Überprüfung kann durch periodische Statusmeldungen unterstützt werden.

Analog zu den Anwendungen zur Informationsverteilung wird mit der sicheren Hardware eine Verringerung der Verzögerung zwischen den Teilnehmern und Servern erreicht, da die Ermittlung des Zeitunterschiedes für die Zeitnachrichten nicht mehr von der Größe der Informationsnachrichten abhängig ist und aus mehreren Zeitnachrichten die ausgewählt werden kann, die die geringste Verzögerung erfahren hat.

4.8.2 Anwendungen mit geschlossenen Kommunikationsbeziehungen

Bei der Klasse der geschlossenen, bidirektionalen Kommunikationsbeziehungen mit diskretem Informationsfluss (Anwendungsklasse 3) ist ein Gebot des Teilnehmers direkt an eine einzelne Information gebunden. Beispielsweise folgt bei der holländischen Auktion dem fallenden Angebot des Auktionators die Entscheidung des Auktionsteilnehmers über

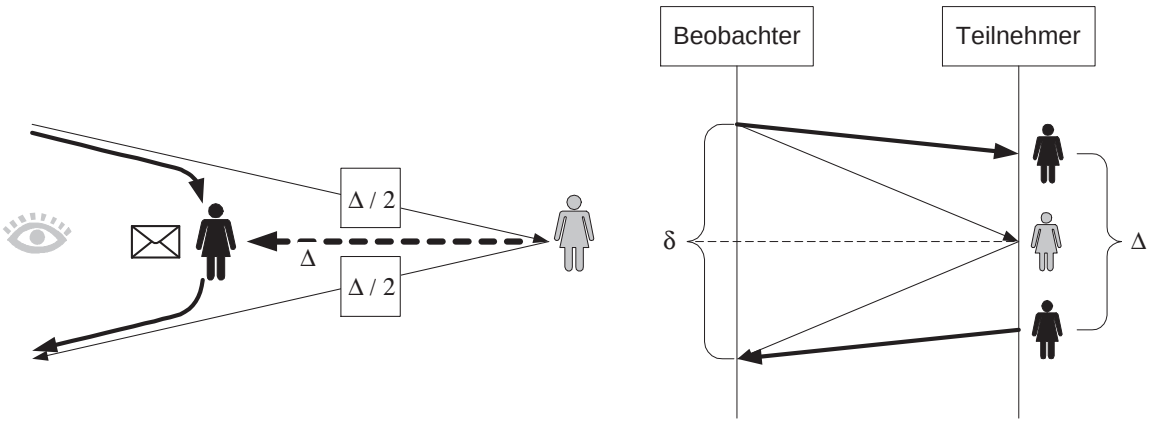


Abbildung 4.18: Angriff des Teilnehmers auf geschlossene, diskrete Anwendungen

den Kauf des Objektes. Der zuerst reagierende Teilnehmer erhält den Zuschlag. Fairnesskriterium ist daher die Reaktionsdauer des Teilnehmers.

Wird die Zeitdauer der Reaktion des Teilnehmers über eine Uhr beim Teilnehmer bestimmt, kann ein Angreifer in beiden Richtungen zwischen Server und Teilnehmer Verzögerungselemente für die Nachrichten zur Verzögerungsbestimmung platzieren und die Informationsnachrichten ungehindert passieren lassen. Er gewinnt dabei eine um die Summe der Verzögerungselemente größere Bedenkzeit (Abbildung 4.18).

Mittels sicherer Hardware lassen sich beide Arten von Nachrichten so verbinden, dass es dem Angreifer unmöglich wird, Information und Verzögerungsbestimmung getrennt zu verzögern. Die sichere Hardware kann dann die Dauer des unverschlüsselten Vorliegens der Nachricht messen. Das unverschlüsselte Vorliegen der Nachricht korreliert jedoch nicht unbedingt mit der Reaktionsdauer des Anwenders, weil der Zeitpunkt, von dem an die Nachricht unverschlüsselt vorliegt, für den Beobachter wegen des Uhrensynchronisationsproblems nicht ersichtlich ist. Treffen zwei Teilnehmer eine geheime Absprache, können sie durch asymmetrisches Verzögern der Nachrichten eine beliebig hohe Bedenkzeit erreichen, indem einer der Teilnehmer seinen Zeitpunkt, von dem an die Nachricht unverschlüsselt vorliegt, früh wählt und der andere Teilnehmer spät. Einem Beobachter bleibt dies verborgen (siehe Abbildung 4.19).

Da sich geheime Absprachen zwischen den Teilnehmern nicht verhindern lassen, ist ein Messen der Reaktionsdauer im Allgemeinen nicht möglich. Um dennoch Fairness auch für die Anwendungsklasse 3 sicherzustellen, bleibt die Möglichkeit, eine Kombination der Verfahren der Informationsverteilung und des Informationsempfanges einzusetzen. Damit

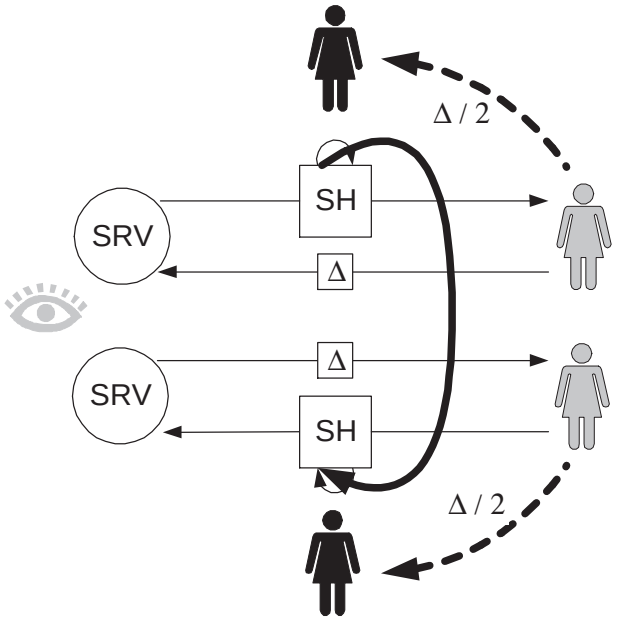


Abbildung 4.19: Geheime Absprache zwischen Teilnehmern bei geschlossenen, diskreten Anwendungen

ist ein gemeinsamer Auslieferungszeitpunkt für alle Nachrichtenkopien festgelegt, so dass geheime Absprachen der Teilnehmer unmöglich sind. Die Reaktionsdauer der Teilnehmer ergibt sich dann aus der Sendereihenfolge der Gebote.

4.9 Zusammenfassung

In Kapitel 4 werden unterschiedliche Wege zur Reduktion von Verzögerungsunterschieden zwischen Teilnehmern von Gruppenkommunikation vorgestellt und die für die Realisierung der Maßnahmen gewählte Abstraktionsebene begründet. Außerdem wird die Anpassungsfähigkeit an die Gesamtverzögerung betrachtet. Als wesentliches Kriterium für die erforderliche Dienstqualität unter den Bedingungen nicht-kooperativer Anwendungen wurde die Sicherheit herausgestellt, da sie die entscheidende Voraussetzung für die Garantie von Fairness ist. Die Betrachtung verwandter Arbeiten auf diesem Gebiet zeigt die grundsätzlichen Nachteile bisheriger Verfahren auf und begründet die Notwendigkeit weiterführender Maßnahmen.

Neben der Minimierung des Verzögerungsunterschiedes wurden weitere Anforderungen

an einen verzögerungsfairen Dienst für nicht-kooperative Echtzeitanwendungen definiert. Hierzu gehören: Fairness über alle Teilnehmer, Minimierung der Gesamtverzögerung, Toleranz gegenüber der Datenratenanforderung und Empfängerzusammensetzung, Unabhängigkeit von Netztechnologien sowie ein geringer Nachrichtenaufwand. Diese Anforderungen lassen sich durch einen verzögerungsfairen Multicast-Dienst über IP am besten erfüllen.

Die zur Lösungsfindung wesentlichen Eigenschaften des Internets werden in einem Systemmodell aus Netzwerk-, IP-Multicast- und Sicherheitsmodell abstrahiert. Das Grundkonzept des vorgestellten Verfahrens zur Reduktion der Verzögerungsunterschiede zwischen Teilnehmern von Gruppenkommunikation geht von Vertrauensbereichen nahe den Teilnehmern aus. Wegen des notwendigen Vertrauens sieht es die Anwendungsschicht zur Dienstbereitstellung vor und beschränkt die Vertrauensbereiche auf einzelne, gut überwachbare Punkte. Zum Ausgleich der Verzögerungsunterschiede werden die Nachrichten durch Wiederaussenden zu einem gemeinsamen Auslieferungszeitpunkt resynchronisiert.

Basierend auf dem erarbeiteten Konzept zur Reduktion von Verzögerungsunterschieden zwischen Teilnehmern werden drei Realisierungsvarianten vorgestellt, die sich vor allem durch Funktion und Ort der Vertrauensbereiche unterscheiden. Im Datenserver-Verfahren werden Datenserver in den Vertrauensbereichen platziert, die die verschlüsselten Nachrichten vom Informationssender empfangen, entschlüsseln, resynchronisieren und zu einem bestimmten Auslieferungszeitpunkt unverschlüsselt an die Teilnehmer senden. Das Datenserver-Verfahren repräsentiert den allgemeinsten Fall dieses Konzeptes; alle weiteren Ansätze sind Spezialisierungen, zum Teil mit erhöhten Anforderungen, aber auch mit höherer Reduzierung der Verzögerungsunterschiede. Beim Schlüsselserver-Verfahren werden Informationsübertragung und Auslieferungsnachricht voneinander getrennt. Die Vertrauensbereiche stellen den Teilnehmern zum Auslieferungszeitpunkt Schlüssel zur Entschlüsselung der Informationsnachrichten zur Verfügung, während die Informationsnachrichten direkt zum Teilnehmer gesendet werden. Das Zeitserver-Verfahren reduziert die Funktion des Vertrauensbereiches im Netzwerk auf die Bereitstellung von Zeitinformation und baut stattdessen mittels sicherer Hardware einen Vertrauensbereich direkt beim Teilnehmer auf. Um eine zu frühe Auslieferung der Information zu verhindern, wird ein Uhrensynchronisationsprotokoll erarbeitet, das die Uhren mit einer Referenzzeit synchron hält und die Authentizität der Zeitinformation überprüft. Das Vorstellen der Uhr durch den Teilnehmer ist damit ausgeschlossen. Außerdem wird ein Datenauslieferungsprotokoll präsentiert, das die Übermittlung der Informationen vor dem Auslieferungszeitpunkt sicherstellt, jedoch erst zum Auslieferungszeitpunkt in verwertbare Form überführt.

Bei allen Verfahren ist die Kommunikation des Dienstes bis zum Auslieferungszeitpunkt durch Verschlüsselung der Informationen gesichert. Vor dem Auslieferungszeitpunkt liegen die Informationen bzw. Schlüssel nur dem Informationssender und den Servern bzw. der sicheren Hardware vor. Die Funktion und Sicherheit der Server und die Nichtkompromittierbarkeit der sicheren Hardware wird durch eine Zertifizierungsagentur garantiert. Die Verfahren basieren daher auf dem Vertrauen in den Informationssender, die Informationen nur über den Dienst und nicht auf anderen Wegen zu verteilen, sowie in die Zertifizierungsagentur.

Um bei kontinuierlichem Informationsfluss neben der Minimierung des Verzögerungsunterschiedes auch eine möglichst geringe Gesamtverzögerung der Übertragung zu erreichen, wird ein dynamischer Algorithmus zur Bestimmung des Auslieferungszeitpunktes vorgestellt. Er nutzt die Dienstgüterückmeldung zur Anpassung der Auslieferungsverzögerung an den aktuellen Netzwerkzustand.

Die Lösung lässt sich prinzipiell auch auf die Anwendungsklassen 2 und 3 übertragen, die neben einem fairen Informationszugang eine faire Interaktion mit dem System erfordern. Für die Anwendungen mit offenen, unidirektionalen Kommunikationsbeziehungen zum Informationsempfang (Klasse 2) wird ein Protokollvorschlag unterbreitet. Für geschlossene, bidirektionalen Kommunikationsbeziehungen (Klasse 3) wird empfohlen, Verfahren der Informationsverteilung und des Informationsempfanges zu kombinieren. Auf diese Weise kann aus der Sendereihenfolge der Antworten auf die Reaktionsdauer der Teilnehmer, die bei Klasse 3 als Fairnesskriterium definiert wird, geschlossen werden.

Kapitel 5

Leistungsbewertung

In Kapitel 4 wurden drei Realisierungsvarianten des Konzeptes zur Reduktion von Verzögerungsunterschieden zwischen Teilnehmern von Gruppenkommunikation vorgestellt, die im Folgenden verglichen werden. Bei den Verfahren handelt es sich um das Datenserver-Verfahren, das Schlüsselserver-Verfahren und das Zeitserver-Verfahren. Sie unterscheiden sich in Funktion und Platzierung der Vertrauensbereiche, in denen eine Resynchronisation der Gruppenkommunikationsnachrichten vorgenommen wird. Allen Verfahren ist gemeinsam, dass sie in der Anwendungsschicht realisiert werden und dass sie geeignet sind, bestimmte Sicherheitsmängel bisheriger Verfahren zu überwinden. Unter Zugrundelegung der in Kapitel 4 formulierten Anforderungen an einen verzögerungsfairen Dienst wird die Leistungsfähigkeit der Verfahren eingeschätzt. Für die Verfahren wird außerdem eine analytische Leistungsbewertung vorgenommen. Weiterhin werden Messergebnisse der Implementierung eines Prototyps präsentiert.

5.1 Allgemeine Leistungsuntersuchungen

Aufbauend auf den unterschiedlichen Prinzipien der einzelnen Reduktionsverfahren werden in diesem Kapitel die wesentlichen Faktoren besprochen, die auf die Größe des Verzögerungsunterschiedes und auf die Gesamtverzögerung Einfluss nehmen. Im Anschluss daran werden Nachrichtenaufwand und Speicherbedarf behandelt. In Tabelle 5.1 sind die drei Realisierungsvarianten für den verzögerungsfairen Dienst gegenübergestellt.

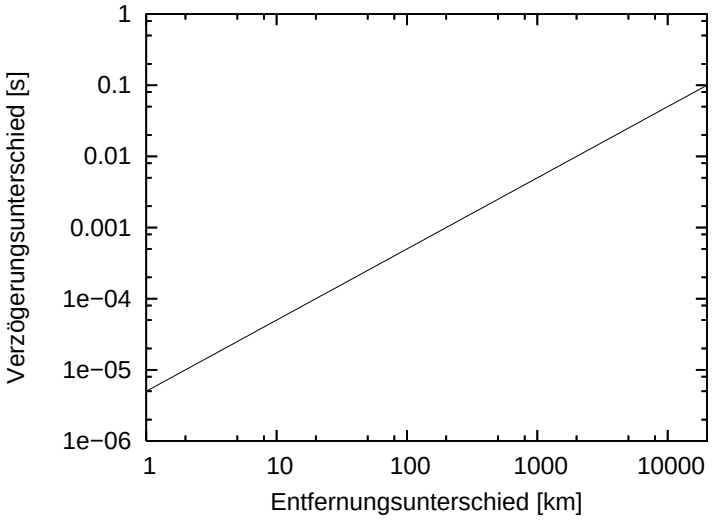


Abbildung 5.1: Abhängigkeit des Verzögerungsunterschiedes von dem maximalen Unterschied der Entfernung der Server von den Teilnehmern. Der Kurvenverlauf verdeutlicht die Wirkung der Server-Verfahren: Durch Platzierung der Server in der Nähe der Teilnehmer kann der Verzögerungsunterschied wesentlich gesenkt werden.

Bei allen untersuchten Verfahren werden die Nachrichtenkopien der über Gruppenkommunikation verteilten Information in Vertrauensbereichen resynchronisiert, wodurch eine Reduktion der Verzögerungsunterschiede auf Anwendungsebene erzielt wird. Im Datenserver-Verfahren wird die Resynchronisation durch Platzierung von Servern nahe den Teilnehmern der Gruppenkommunikation erreicht. Die Schlüsselserver ermöglichen eine Trennung von Daten und Synchronisationsinformation. Mittels Zeitserver-Verfahren kann darüber hinaus die Synchronisationsinformation redundant ausgelegt werden. Die vollständige Trennung von Daten und Synchronisationsinformation erlaubt die Unterstützung von unterschiedlich priorisierten virtuellen Kanälen oder gänzlich unterschiedlichen Zugangsnetzen für den Zugriff auf diese Informationen. Die Anforderungen an die Vertrauensbereiche sind minimal, da diese nur vertrauenswürdige Zeitinformation zur Verfügung stellen müssen.

Verzögerungsunterschied. Der Verzögerungsunterschied aller vorgestellten Verfahren ist abhängig von der Genauigkeit der Serveruhren, der Festlegung der Auslieferungsverzögerung, Unterschieden in der Verarbeitungszeit der Nachrichten auf den Servern und bei den Teilnehmern. Hinzu kommt die Differenz zwischen maximaler und minimaler Verzögerung der Auslieferungsnachrichten (beim Datenserver- und Schlüsselserver-Verfahren)

bzw. der Zeitnachrichten (beim Zeitserver-Verfahren) über alle Server-Teilnehmer-Paare.

Im Folgenden werden drei Komponenten betrachtet, die einen signifikanten Anteil am Verzögerungsunterschied haben: die maximale Entfernung der Server von den Teilnehmern, der in der Regel im Zugangsnetz anzutreffende Übertragungsabschnitt mit der kleinsten Übertragungskapazität sowie die Länge der zeitkritischen Nachricht.

Die räumliche Entfernung zwischen Servern und Teilnehmern ist die Ursache für die Signalausbreitungsverzögerung der Nachrichten. Soll beispielsweise eine Nachricht von einem Punkt der Erde aus alle anderen Punkte erreichen, darunter auch den am weitesten entfernten, müsste das Signal mindestens einen halben Erdumfang zurücklegen. Die Verzögerung, mit der die Nachricht bei dem am weitesten entfernten Teilnehmer eintrifft, beträgt mindestens 100 ms (Lichtgeschwindigkeit im Glasfaserkabel ca. 200 000 km/s). Gegenüber einer Nachricht, die ein in unmittelbarer Nähe des Servers befindlicher Teilnehmer erhält, folgt daraus ein Verzögerungsunterschied von 100 ms. In Abbildung 5.1 ist die Abhängigkeit des Verzögerungsunterschiedes von der maximalen Entfernung der Server von den Teilnehmern dargestellt. Darin wird gezeigt, dass der Verzögerungsunterschied mit dem Entfernungsunterschied der Server von den Teilnehmern linear zunimmt. Gleichzeitig geht daraus hervor, dass sich aus der Verkürzung des Teilnehmer-Server-Abstandes ein hohes Potenzial für die Reduktion des Verzögerungsunterschiedes ergibt.

Zwei weitere Faktoren haben signifikanten Einfluss auf die Differenz zwischen maximaler und minimaler Verzögerung der Nachrichten über alle Server-Teilnehmer-Paare. Das ist zum Einen die Übertragungsrate der Übertragungsabschnitte zwischen den Routern und zum Anderen die Nachrichtenlänge. Eine geringere Übertragungsrate oder eine längere Nachricht vergrößert die Zeit, die zum Senden der Nachricht benötigt wird und durch die sich der vollständige Empfang der Nachricht beim Teilnehmer verzögert. Während im Backbone des Internets von hohen Übertragungsraten (1 Gbit/s und höher) ausgegangen werden kann, sind die Übertragungsraten im Zugangsnetz teilweise sehr gering. Typische Übertragungsraten für xDSL sind 768 kbit/s, für ISDN 64 kbit/s und für analoge Modems 56 kbit/s. Für mobile Teilnehmer liegen die Übertragungsraten teilweise noch darunter. Zum Beispiel beträgt die Sendezeit für ein Paket der Größe von 1000 Byte auf einer ISDN-Verbindung mit einer Übertragungsrate von 64 kbit/s 125 ms. In Abbildung 5.2 ist der durch die Serialisierung der Nachrichten auf dem Übertragungsabschnitt der kleinsten Übertragungskapazität auftretende Verzögerungsunterschied gegenüber einem Teilnehmer mit einer Übertragungsrate von 1 Gbit/s auf den Übertragungsabschnitten zwischen Server und Teilnehmer in Abhängigkeit von Nachrichtenlänge und Übertragungskapazität

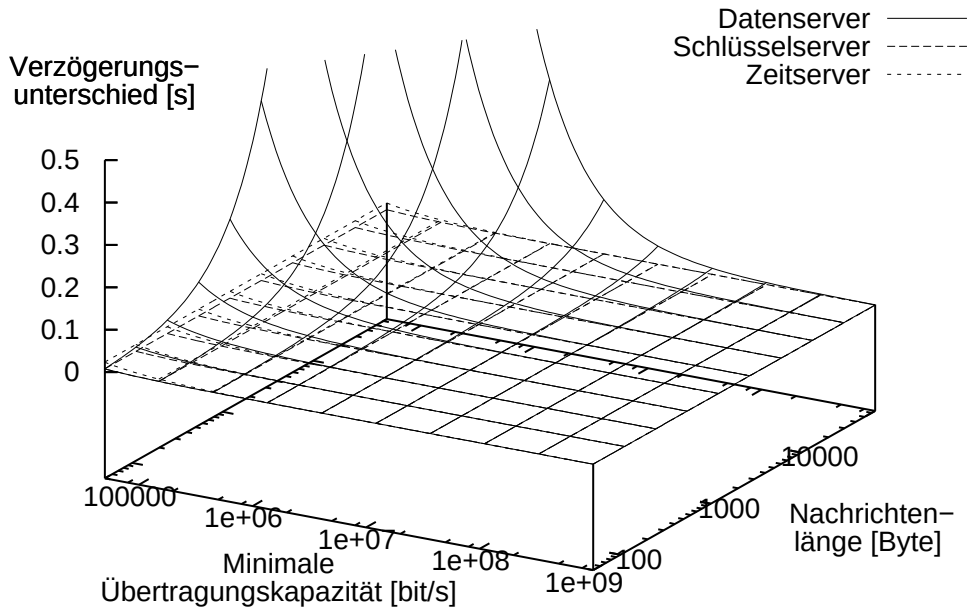


Abbildung 5.2: Abhängigkeit des durch Serialisierung einer Nachricht bedingten Verzögerungsunterschiedes von der Nachrichtenlänge und der minimalen Übertragungskapazität auf den Übertragungsabschnitten zwischen Server und Teilnehmer bezogen auf einen Teilnehmer mit einer Übertragungsrate von 1 Gbit/s

dargestellt. Für das Datenserver-Verfahren führen geringe Übertragungskapazität und hohe Nachrichtenlängen zu hohen Verzögerungsunterschieden. Die getrennte Versendung der Synchronisationsinformation im Schlüsselserver- und Zeitserver-Verfahren machen diese Realisierungsvarianten unabhängig von der Länge der Informationsnachrichten. Die maximal mögliche Reduktion des Verzögerungsunterschiedes sollte daher von einem dieser beiden Verfahren erwartet werden.

Gesamtverzögerung. Gegenüber der Gesamtverzögerung einer einfachen Auslieferung der Daten mittels einer Multicast-Nachricht werden die Informationen beim Datenserver-Verfahren vom Sender zunächst an die Datenserver geleitet und von dort nach Erreichen des Auslieferungszeitpunktes an die Empfänger weitergegeben. Im Vergleich zu einer direkten Auslieferung führt das im Allgemeinen zu einer höheren Verzögerung der Nachricht durch den Umweg und die Verarbeitungzeit des Servers. Das Gleiche gilt für die Schlüssel beim Schlüsselserver-Verfahren. Im Zeitserver-Verfahren trägt die maximale Verzögerung der Zeitnachrichten und die Kommunikation mit der sicheren Hardware zur Gesamtverzögerung bei. Außerdem vergrößert sich die Gesamtverzögerung bei allen drei Verfahren durch die Ver-und Entschlüsselung der Nachrichten. Schließlich wird die Gesamtverzögerung auch in Abhängigkeit von der Defensivität der Bestimmung der Auslieferungsverzögerung erhöht.

Nachrichtenaufwand. Zur Ermittlung des Nachrichtenaufwandes wird die Anzahl der Nachrichten A bestimmt, die für eine Nutznachricht im Mittel gesendet werden. Beim Datenserver-Verfahren wird für eine Nutznachricht die Informationsnachricht per Multicast versendet. Jeder Datenserver versendet für jede Informationsnachricht eine Auslieferungsnachricht, ebenfalls per Multicast. Es sei d die Anzahl der Datenserver, dann werden für eine Nutznachricht $A = 1+d$ Nachrichten benötigt. Mit n als Anzahl der Nutznachrichten ergibt sich $A(n) = n + n \cdot d = n(1 + d)$. Analog gilt für das Schlüsselserver-Verfahren, bei dem die Informationsnachricht direkt an die Teilnehmer und die Schlüsselnachricht an die Schlüsselserver versendet werden, $A(n) = n + n + n \cdot s = n(2 + s)$ mit $s =$ Anzahl der Schlüsselserver. Beim Zeitserver-Verfahren werden die Informationsnachrichten zusammen mit den Schlüsselnachrichten versandt. Die Auslieferungsnachrichten werden nur von der sicheren Hardware zum Teilnehmer transferiert und erzeugen keinen zusätzlichen Nachrichtenaufwand. Die Zeitnachrichten werden unabhängig von der Anzahl der Nutznachrichten zur Synchronisation der Uhren der sicheren Hardware versandt. Es sei z die

Tabelle 5.1: Vergleich der Verfahren zur Reduktion von Verzögerungsunterschieden

	Datenserver	Schlüsselserver	Zeitserver
Prinzip	Synchronisation der Daten nahe den Teilnehmern	Trennung von Daten und Synchronisationsinformation	Redundante Synchronisationsinformation
Den Verzögerungsunterschied beeinflussende Faktoren	Genauigkeit der Serveruhren Korrektheit der Auslieferungsverzögerung Verarbeitungszeitunterschiede der Nachrichten Server-Teilnehmer-Anbindung Größe der Auslieferungsnachricht (Information)	Genauigkeit der Serveruhren Korrektheit der Auslieferungsverzögerung Verarbeitungszeitunterschiede der Nachrichten Server-Teilnehmer-Anbindung Größe der Auslieferungsnachricht (Schlüssel)	Genauigkeit der Serveruhren Korrektheit der Auslieferungsverzögerung Verarbeitungszeitunterschiede der Nachrichten Server-Teilnehmer-Anbindung Größe der Zeitnachricht
Die Gesamtverzögerung beeinflussende Faktoren	Informationsnachrichten über Server Ver- und Entschlüsselung der Nachrichten	Schlüsselnachrichten über Server Ver- und Entschlüsselung der Nachrichten	Zeitnachrichten von Server zu Teilnehmer Ver- und Entschlüsselung der Nachrichten Kommunikation mit sicherer Hardware
Nachrichtenaufwand	O(n) $A(n) = n(1 + d)$	O(n) $A(n) = n(2 + s)$	O(n) $A(n) = n(1 + z \cdot \frac{\lambda_{ZN}}{\lambda_{IN}})$
Speicherbedarf	Informationen auf Server	Schlüssel auf Server	Schlüssel bei Teilnehmer

Anzahl der Zeitserver, λ_{IN} die mittlere Ankunftsrate der Informationsnachrichten und λ_{ZN} die mittlere Ankunftsrate der Zeitnachrichten eines Zeitservers beim Teilnehmer, dann ist die Anzahl der Nachrichten pro Nutznachricht:

$$A = 1 + z \cdot \frac{\lambda_{ZN}}{\lambda_{IN}}.$$

Der Term $z \cdot \frac{\lambda_{ZN}}{\lambda_{IN}}$ spiegelt den Mittelwert der Anzahl der Zeitnachrichten wider, die jeweils pro Informationsnachricht versandt werden. Die Anzahl der Nachrichten in Abhängigkeit von den Nutznachrichten ist demzufolge:

$$A(n) = n + n \cdot z \cdot \frac{\lambda_{ZN}}{\lambda_{IN}} = n(1 + z \cdot \frac{\lambda_{ZN}}{\lambda_{IN}}).$$

Werden beispielsweise im Mittel 10 Informationsnachrichten pro Sekunde versendet und 100 Zeitserver versenden alle 100 s eine Zeitnachricht, beträgt der Nachrichtenaufwand bezogen auf die Anzahl der Nutznachrichten $A(n) = n(1 + 100 \cdot \frac{0.01}{10}) = 1.1n$, d.h. es entsteht ein Mehraufwand von 10 % durch die Zeitserver.

Bei allen Verfahren ist die Anzahl der Nachrichten linear von der Anzahl der Nutznachrichten abhängig. Somit ergibt sich die Ordnung $O(n)$ für alle Verfahren.

Speicherbedarf. Die Server des Datenserver-Verfahrens speichern die gesendeten Nachrichten bis zum Auslieferungszeitpunkt zwischen. Sie müssen in der Lage sein, die Informationsmenge aus dem Produkt von Informationsübertragungsrate und Auslieferungsverzögerung zu speichern. Schlüsselserver speichern dagegen lediglich Schlüssel der Informationen. Der Speicherbedarf ergibt sich aus dem Produkt von Schlüsselübertragungsrate und Auslieferungsverzögerung. Im Zeitserver-Verfahren besteht kein Speicherbedarf auf den Servern, da die Teilnehmer die Schlüssel selbst zwischenspeichern. Der zusätzliche Speicherbedarf ergibt sich ebenfalls aus dem Produkt von Schlüsselübertragungsrate und Auslieferungsverzögerung.

5.2 Spezielle Leistungsuntersuchungen

In diesem Abschnitt wird für die im Abschnitt 4.4 vorgestellten Verfahren zur Reduktion von Verzögerungsunterschieden eine Leistungsbewertung vorgenommen. Dazu wird

zuerst eine Analyse der Nachrichtenverzögerung und der Verzögerungsunterschiede der Protokolle durchgeführt. Anschließend wird der Verzögerungsunterschied zwischen den Teilnehmern anhand zweier extremer Szenarien abgeschätzt. Schließlich wird mittels eines auf Smart Cards implementierten Prototyps die Leistungsfähigkeit gegenwärtiger sicherer Hardware beurteilt.

Basierend auf der Analyse der Protokolle der Verfahren werden in diesem Abschnitt folgende Fragestellungen erörtert:

- Auf welches Niveau können die Verzögerungsunterschiede mit den Ansätzen gesenkt werden?
- Wie hoch ist die Verzögerung, die durch diesen Ansatz verursacht wird?
- Mit welcher Rate können neue Informationen bereitgestellt werden?

Die Beantwortung der ersten beiden Fragen wird aus der Analyse der Nachrichtenverzögerung zu den Teilnehmern abgeleitet. Die Beantwortung der dritten Frage ist insbesondere für das Zeitserver-Verfahren von Bedeutung. Die Rate, mit der neue Informationen bereitgestellt werden können, ist gleich dem Intervall zwischen zwei Auslieferungszeitpunkten und hängt davon ab, wie häufig neue Nachrichtenschlüssel von der sicheren Hardware zur Verfügung gestellt werden können. Tabelle 5.2 gibt einen Überblick über die in der Leistungsbewertung des Zeitserver-Verfahrens verwendeten mathematischen Symbole.

Tabelle 5.2: Für die Leistungsbewertung verwendete mathematische Symbole

Notation	Erklärung
i	Nachricht i
k	Server k
j	Teilnehmer j
IN	Informationsnachricht
SN	Schlüsselnachricht
AN	Auslieferungsnachricht

ZN	Zeitnachricht
a_{max}	Maximale Gangabweichung der Uhr einer sicheren Hardware
ρ_{max}	Ganggenauigkeit der Uhr der sicheren Hardware
θ_{SH}	Zeitversatz der Uhr der sicheren Hardware relativ zum Sender
τ^S	Maximales Synchronisationsintervall der Uhren der sicheren Hardware
δ_i	Gesamtverzögerung von Nachricht i
δ_{ij}	Verzögerung der Nachricht i vom Sender zum Teilnehmer j
δ_i^A	Auslieferungsverzögerung
t_i^S	Sendezeitpunkt von Nachricht i
δ^{Sig}	Ausbreitungsverzögerung des Signals
δ^P	Verzögerung durch Verarbeitung in den Routern
δ^W	Verzögerung durch Warteschlange
$\delta^{\ddot{U}}$	Verzögerung durch Übertragung der Nachricht auf das Medium
λ	Mittlere Ankunftsrate
μ	Bedienrate
$E(B)$	Erwartungswert der Bediendauer
$E(W)$	Erwartungswert der Wartezeit
$Var(W)$	Varianz der Wartezeit
ρ	Auslastungsfaktor
$\delta_{SH}^{\ddot{U}\,SN}$	Übertragungsverzögerung der Schlüsselnachricht vom Teilnehmer zur sicheren Hardware
$\delta_{ij}^{E\,SN}$	Verzögerung durch Entschlüsselung der Schlüsselnachricht
J	Verzögerungsunterschied zwischen den Teilnehmern
J_{Uhr}	Durch die Gangabweichung der Uhren der Sicheren Hardware verursachter, maximal erlaubter Verzögerungsunterschied

J_{SH}^{ZN}	Verzögerungsunterschied durch Authentifizierung der Zeitnachricht auf der sicheren Hardware
$\tau_{min}^{t^A}$	Minimales Intervall zwischen zwei Auslieferungszeitpunkten
l_N	Länge der Nachricht N
C	Übertragungskapazität eines Übertragungsabschnittes
A	Nachrichtenaufwand, Anzahl der Nachrichten pro Nutznachricht
d	Anzahl der Datenserver
s	Anzahl der Schlüsselserver
z	Anzahl der Zeitserver

5.2.1 Nachrichtenverzögerung und Verzögerungsunterschied zwischen den Teilnehmern

Die Verzögerung δ_{ij} einer Nachricht i zu einem Teilnehmer j im **Datenserver-Verfahren** setzt sich zusammen aus der Verzögerung δ_{ik}^{IN} der Informationsnachricht IN vom Informationssender zu Datenservern k und der Verzögerung δ_{kj}^{AN} der Auslieferungsnachricht AN von den Datenservern zu den Teilnehmern.

$$\delta_{ij} = \delta_{ik}^{IN} + \delta_{kj}^{AN}$$

Die Verzögerung der Informationsnachricht wird entweder aus der Differenz von Auslieferungszeitpunkt t_i^A und Sendezeitpunkt t_i^S bestimmt oder – im Falle eines verspäteten Eintreffens der Informationsnachricht beim Server k – durch die Verzögerung $\delta_{ik}^{\ddot{U}S}$, die die Informationsnachricht vom Informationssender bis zum Datenserver erfährt. Hinzu kommt die Verzögerung δ_{ik}^{EIN}, die durch die Entschlüsselung der Informationsnachricht in Vorbereitung des Weitersendens als Auslieferungsnachricht eintritt.

$$\delta_{ik}^{IN} = \max\{t_i^A - t_i^S, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{EIN}\} = \max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{EIN}\}$$

Die Gesamtverzögerung δ_i – das Maximum der Verzögerungen über alle Teilnehmer – ergibt sich als:

$$\delta_i = \max_{\forall T_j \in R}\{\delta_{ij}\} = \max_{\forall T_j \in R, T_j \in DSrv_k}\{\max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{EIN}\}, \delta_{kj}^{AN}\}.$$

Der Verzögerungsunterschied zwischen den Teilnehmern im Datenserver-Verfahren wird aus der Differenz von maximaler und minimaler Verzögerung der Informationsnachricht und der Auslieferungsnachricht über alle Teilnehmer gebildet:

$$\begin{aligned} J_i = & \max_{\forall T_j \in R, T_j \in DSrv_k}\{\max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{EIN}\} + \delta_{kj}^{AN}\} \\ & - \min_{\forall T_j \in R, T_j \in DSrv_k}\{\max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{EIN}\} + \delta_{kj}^{AN}\}. \end{aligned} \tag{5.1}$$

Im **Schlüsselserver-Verfahren** ist die Verzögerung der Nachricht i in der Regel abhängig von der Summe der Verzögerungen aus Schlüsselnachricht SN und Auslieferungsnachricht, die über die Schlüsselserver verteilt wird. Sollte die direkt zu den Teilnehmern gesendete Informationsnachricht nach der Auslieferungsnachricht bei den Teilnehmern eintreffen, bildet die Verzögerung der Informationsnachricht die Verzögerung der Nachricht. Nach dem Eintreffen von Informationsnachricht und Auslieferungsnachricht beim Teilnehmer wird die Zeitspanne δ_{ij}^{EIN}zur Entschlüsselung der Informationsnachricht benötigt. Zur Vereinfachung der Darstellung sei ohne Beschränkung der Allgemeinheit angenommen, dass der Sendezeitpunkt von Informations- und Schlüsselnachricht gleich ist.

$$\delta_{ij} = \max\{\delta_{ij}^{IN}, \delta_{jk}^{SN} + \delta_{kj}^{AN}\} + \delta_{ij}^{EIN}, \quad \text{für} \quad t_{ij}^{SIN} = t_{ik}^{SSN}$$

Die Verzögerung der Schlüsselnachricht wird entweder durch die Auslieferungsverzögerung oder durch die Verzögerung, die die Schlüsselnachricht auf ihrem Weg zum Schlüsselserver erfährt, zuzüglich deren Entschlüsselung, bestimmt.

$$\delta_{ik}^{SN} = \max\{t_i^A - t_i^S, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{ESN}\} = \max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{ESN}\}$$

Daher ergibt sich für die Gesamtverzögerung einer Nachricht im Schlüsselserver-Verfahren:

$$\delta_i = \max_{\forall T_j \in R}\{\delta_{ij}\} = \max_{\forall T_j \in R, T_j \in SSrv_k}\{max\{\delta_{ij}^{IN}, \max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{ESN}\} + \delta_{kj}^{AN}\} + \delta_{ij}^{EIN}\}$$

und für den Verzögerungsunterschied zwischen den Teilnehmern:

$$J_i = \max_{\forall T_j \in R, T_j \in SSrv_k} \{max\{\delta_{ij}^{IN}, \max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{ESN}\} + \delta_{kj}^{AN}\} + \delta_{ij}^{EIN}\}$$
$$- \min_{\forall T_j \in R, T_j \in SSrv_k} \{\max\{\delta_{ij}^{IN}, \max\{\delta_i^A, \delta_{ik}^{\ddot{U}S} + \delta_{ik}^{ESN}\} + \delta_{kj}^{AN}\} + \delta_{ij}^{EIN}\}.$$

Die Verzögerung δ_{ij} einer Nachricht i zu einem Teilnehmer j im **Zeitserver-Verfahren** setzt sich aus den Verzögerungen dreier Nachrichten zusammen. Einen Bestandteil bildet die Verzögerung der gemeinsam versendeten Informations-und Schlüsselnachricht zum Teilnehmer einschließlich der Weiterleitung der Schlüsselnachricht zur sicheren Hardware. Einen weiteren Teil trägt die von der sicheren Hardware generierte Auslieferungsnachricht zum Teilnehmer sowie die dadurch ermöglichte Entschlüsselung der Informationsnachricht bei. Da es sich bei der Freigabe des Informationsschlüssels um eine lokale Entscheidung der jeweiligen sicheren Hardware handelt, fließt der Zeitversatz der Uhren der sicheren Hardware gegenüber einer Referenzzeit bei der Berechnung der Höhe der Nachrichtenverzögerung mit ein. Der Zeitversatz durch die Übertragung der Zeitnachrichten ZN vom Zeitserver k zum Teilnehmer des parallel dazu ablaufenden Uhrensynchronisationsprotokolls bildet den dritten Verzögerungsanteil δ_{kj}^{ZN}. Weiterhin ist die Gangabweichung der Uhren der sicheren Hardware gegenüber einer gemeinsamen Referenzzeit zu berücksichtigen. Die Gangabweichung ist in Gleichung 5.2 vereinfachend mit der maximal erlaubten Gangabweichung a_{max} angegeben.

$$\delta_{ij} = \delta_{ij}^{IN||SN} + \delta_{ij}^{AN} + \delta_{kj}^{ZN} + a_{max} \tag{5.2}$$

In der Regel wird wiederum die Schlüsselnachricht bis zum vom Sender vorgegebenen Auslieferungszeitpunkt t_i^A von der sicheren Hardware verzögert. Sollte die Informations- und Schlüsselnachricht jedoch verspätet eintreffen und die Entschlüsselung des Informationsschlüssels nicht vor dem Auslieferungszeitpunkt beendet sein, ist die auf der Übertragungsstrecke tatsächlich erfahrene Verzögerung $\delta_{ij}^{\ddot{U}S\,IN||SN}$, die Übertragung der Schlüsselnachricht zur sicheren Hardware $\delta_{iSHj}^{\ddot{U}SN}$ sowie die Verzögerung durch Entschlüsselung der Schlüsselnachricht δ_{ij}^{ESN} zu berechnen:

$$\begin{aligned}\delta_{ij}^{IN||SN} &= \max\{t_i^A - t_i^S, \delta_{ij}^{\ddot{U}S\,IN||SN} + \delta_{iSHj}^{\ddot{U}SN} + \delta_{ij}^{ESN}\} \\ &= \max\{\delta_i^A, \delta_{ij}^{\ddot{U}S\,IN||SN} + \delta_{iSHj}^{\ddot{U}SN} + \delta_{ij}^{ESN}\}.\end{aligned}$$

Die im Zeitserver-Verfahren durch die sichere Hardware erstellte Auslieferungsnachricht wird zum Teilnehmer übertragen $\delta_{iSHj}^{\ddot{U}\,AN}$ und dient zur Entschlüsselung der Informationsnachricht $\delta_{ij}^{E\,IN}$.

$$\delta_{ij}^{AN} = \delta_{iSHj}^{\ddot{U}\,AN} + \delta_{ij}^{E\,IN}$$

Der Zeitversatz der Uhren der sicheren Hardware wird durch den Versand der Zeitnachrichten des Uhrensynchronisationsprotokolls hervorgerufen. Für die Berechnung des Zeitversatzes der Uhren wird angenommen, dass die Uhren der Server mit der Referenzzeit synchron sind. Der Zeitversatz der Uhr der sicheren Hardware j gegenüber dem Zeitserver ist durch die Dauer zwischen Ausstellung des Zeitstempels auf dem Zeitserver bis zum Stellen der Uhr auf der sicheren Hardware gegeben und ist durch die von der Zeitnachricht verursachte Verzögerung δ_{kj}^{ZN} gekennzeichnet. Sie umfasst die Verzögerung $\delta_{kj}^{\ddot{U}S\,ZN}$, die die Zeitnachricht auf der Übertragungsstrecke vom Zeitserver zum Teilnehmer erfährt, die Verarbeitungszeit beim Teilnehmer δ_j^P, die Übertragungszeit zur sicheren Hardware $\delta_{SHj}^{\ddot{U}\,ZN}$ sowie die Verarbeitungszeit auf der sicheren Hardware zur Authentifizerung der Zeitnachricht $\delta_{SHj}^{E\,ZN}$. Der Zeitversatz einer bestimmten Uhr nach dem Eintreffen eines Zeitstempels auf der sicheren Hardware ist demnach:

$$\delta_{kj}^{ZN} = \delta_{kj}^{\ddot{U}S\,ZN} + \delta_j^P + \delta_{SHj}^{\ddot{U}\,ZN} + \delta_{SHj}^{E\,ZN}.$$

Die Gangabweichung der Uhren der sicheren Hardware wird durch die Konstante a der maximal erlaubten Gangabweichung berücksichtigt (Gleichung 5.2). Nach der Gangabweichung der Uhren und dem maximal erlaubten Verzögerungsunterschied der Nachrichten richtet sich die Häufigkeit, mit der Zeitstempel die sichere Hardware erreichen müssen (siehe Abschnitt 4.5.3.2). Für die Nachrichtenverzögerung zu einem Teilnehmer ergibt sich:

$$\delta_{ij} = \max\{\delta_i^A, \delta_{ij}^{\ddot{U}S\,IN||SN} + \delta_{iSHj}^{\ddot{U}\,SN} + \delta_{ij}^{E\,SN}\} + \delta_{iSHj}^{\ddot{U}\,AN} + \delta_{ij}^{E\,IN} + \delta_{kj}^{ZN} + a_{max}$$

und für die Gesamtverzögerung über alle Teilnehmer:

$$\begin{aligned}\delta_i = \max_{\forall T_j \in R, T_j \in ZSrv_k}\{\delta_{ij}\} = \max_{\forall T_j \in R, T_j \in ZSrv_k}\{&\max\{\delta_i^A, \delta_{ij}^{\ddot{U}S\,IN||SN} + \delta_{iSHj}^{\ddot{U}\,SN} + \delta_{ij}^{E\,SN}\} \\ &+ \delta_{iSHj}^{\ddot{U}\,AN} + \delta_{ij}^{E\,IN} + \delta_{kj}^{ZN}\} + a_{max}.\end{aligned}$$

Der Verzögerungsunterschied J_i zwischen den Teilnehmern einer Nachricht i wird im Zeitserver-Verfahren durch den frühesten und spätesten Zeitpunkt der Entschlüsselung

der Informationsnachricht durch die Teilnehmer der Teilnehmermenge R bestimmt:

$$J_i = \max_{\forall T_j \in R, T_j \in ZSrv_k} \{\max\{\delta_i^A, \delta_{ij}^{\ddot{U}\,SIN||SN} + \delta_{iSHj}^{\ddot{U}\,SN} + \delta_{ij}^{E\,SN}\} + \delta_{iSHj}^{\ddot{U}\,AN} + \delta_{ij}^{E\,IN} + \delta_{kj}^{ZN}\} + a_{max}$$
$$- \min_{\forall T_j \in R, T_j \in ZSrv_k} \{\max\{\delta_i^A, \delta_{ij}^{\ddot{U}\,SIN||SN} + \delta_{iSHj}^{\ddot{U}\,SN} + \delta_{ij}^{E\,SN}\} + \delta_{iSHj}^{\ddot{U}\,AN} + \delta_{ij}^{E\,IN} + \delta_{kj}^{ZN}\} - a_{max}. \tag{5.3}$$

5.2.2 Kleinstes Intervall zwischen zwei Auslieferungszeitpunkten

Das kleinste Intervall zwischen zwei Auslieferungszeitpunkten $\tau_{min}^{t^A}$ bestimmt die Rate, mit der neue Informationen zur Verfügung gestellt werden können. Es wird durch die Zugriffsdauer auf die sichere Hardware zur Entschlüsselung des Nachrichtenschlüssels bestimmt. Basierend auf den vorangegangenen Betrachtungen ergibt sich dieses Intervall als Summe der Verzögerung $\delta_{iSHj}^{\ddot{U}\,SN}$ bei der Übertragung der Schlüsselnachricht zur sicheren Hardware, der Verzögerung $\delta_{ij}^{E\,SN}$ bei der Entschlüsselung der Schlüsselnachricht sowie der Verzögerung $\delta_{iSHj}^{\ddot{U}\,AN}$ bei der Rückgabe der Auslieferungsnachricht:

$$\tau_{min}^{t^A} = \delta_{iSHj}^{\ddot{U}\,SN} + \delta_{ij}^{E\,SN} + \delta_{iSHj}^{\ddot{U}\,AN}.$$

Ist die Gangabweichung der Uhren der sicheren Hardware nicht vernachlässigbar, muss die Dauer und Häufigkeit des Zugriffs des Zeitprotokolls auf die sichere Hardware berücksichtigt werden. Darüber hinaus vergrößert sich das Intervall, sobald gleichzeitig von mehreren voneinander unabhängigen Informationsquellen Daten bezogen werden.

Da die Nachrichtenlänge zu einem Nachrichtenschlüssel beliebig gewählt werden kann, hat das Intervall zwischen zwei Auslieferungszeiten vernachlässigbare Auswirkungen auf die Verzögerungsunterschiede der Informationsverteilung. Dies gilt, sofern die Entschlüsselung der Informationen bei den Teilnehmern keine signifikanten Verzögerungsunterschiede hervorruft.

5.2.3 Modellierung der Übertragungsstrecke Server-Teilnehmer

Während sich der Verzögerungsunterschied bis zum Verlassen des Vertrauensbereiches (Server bzw. sichere Hardware) durch geeignete Wahl des Auslieferungszeitpunktes auf eine vernachlässigbare Größe reduzieren lässt, kann der Verzögerungsunterschied zwischen Server und Teilnehmern durch die beschriebenen Verfahren nicht vollständig ausgeglichen

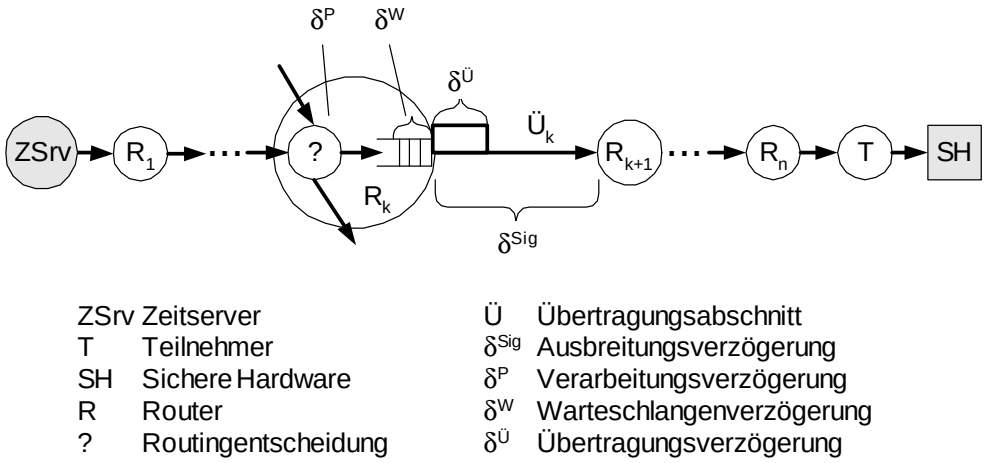


Abbildung 5.3: Modell der Übertragungsstrecke zwischen Server und Teilnehmer für die Berechnung der Nachrichtenverzögerung

werden. Dies betrifft die Auslieferungsnachrichten im Datenserver- und Schlüsselserver-Verfahren sowie die Zeitnachrichten im Zeitserver-Verfahren. Zur Bestimmung dieses verbleibenden Verzögerungsunterschiedes soll das nachfolgend erläuterte Modell der Übertragungsstrecke $\ddot{U}S$ zwischen Server und Teilnehmer verwendet werden.

Das Netzwerk zwischen einem Server und einem Teilnehmer wird, wie im Bild 5.3 dargestellt, aus einer Anzahl von Routern, die durch Übertragungsabschnitte miteinander verbunden sind, modelliert. Die Dauer der Übertragung der Auslieferungsnachricht (bzw. der Zeitnachricht) setzt sich zusammen aus der Dauer des Versendens der Nachricht auf dem Server $\delta_k^{\ddot{U}}$, der Signallaufzeit δ_k^{Sig} sowie der Summe der Dauer von Verarbeitungszeit δ_r^P, Wartezeit δ_r^W, Senden $\delta_r^{\ddot{U}}$ und der Signallaufzeit δ_r^{Sig} auf dem Übertragungsabschnitt des Routers r zu Router $r+1$.

$$\delta_{kj}^{\ddot{U}SN} = \delta_k^{\ddot{U}} + \delta_k^{Sig} + \sum_{r=1}^{n} (\delta_r^P + \delta_r^W + \delta_r^{\ddot{U}} + \delta_r^{Sig})$$

Nachfolgend wird eine Formel zur Berechnung von Erwartungswerten für die Verzögerung der Nachricht auf der Übertragungsstrecke vom Server zum Teilnehmer abgeleitet. Da es sich bei der Verteilung der Nachrichten um einen zeitkritischen Vorgang handelt, wird angenommen, dass ein Transportprotokoll ohne Übertragungswiederholungen, ggf. mit Vorwärts-Fehlerkorrektur, verwendet wird. Paketverluste haben daher keine Auswirkungen auf das Protokoll, was die Verwendung eines Warteschlangenmodells ermöglicht.

Für die Abschätzung der Verzögerungen auf der Übertragungsstrecke wird die durchschnittliche Wartezeit $E[\delta_r^W]$ eines Paketes in der Warteschlange des Routers r berechnet. Die Paketankunftszeiten werden als ein Poisson-Prozess mit exponentiell verteilten Zwischenankunftszeiten modelliert. Bei einer beliebig verteilten Bediendauer ergibt sich daraus ein M/G/1-Warteschlangensystem. Die durchschnittliche Wartezeit $E[W]$ in einem M/G/1-Warteschlangensystem kann mit Hilfe der Pollaczek-Khinchin-Formel [Kleinrock 1976] berechnet werden:

$$E[W] = \frac{\lambda \cdot E[(B)^2]}{2(1-\rho)}.$$

Dabei entspricht λ der Ankunftsrate der Pakete bei dieser Warteschlange, B der Bediendauer und ρ der Systemauslastung. Die Varianz der Wartezeit $Var[W]$ wird folgendermaßen bestimmt:

$$Var[W] = (E[W])^2 + \frac{\lambda \cdot E[(B)^3]}{3(1-\rho)}.$$

Der Erwartungswert der Verzögerung der Nachrichten zwischen Server und Teilnehmer ergibt sich unter Annahme der statistischen Unabhängigkeit der Wartezeiten in den Warteschlangen der Router somit zu:

$$E[\delta_{kj}^{\ddot{U}SN}] = \delta_k^{\ddot{U}} + \delta_k^{Sig} + \sum_{r=1}^{n}(\delta_r^P + \frac{\lambda_r \cdot E[(B_r)^2]}{2(1-\rho_r)} + \delta_r^{\ddot{U}} + \delta_r^{Sig}). \tag{5.4}$$

Die Varianz beträgt:

$$Var[\delta_{kj}^{\ddot{U}SN}] = \sum_{r=1}^{n}\left((E[W_r])^2 + \frac{\lambda_r \cdot E[(B_r)^3]}{3(1-\rho_r)}\right). \tag{5.5}$$

5.2.4 Leistungsvergleich der Verfahren

Die in den vorangegangenen Abschnitten vorgestellten theoretischen Analysen werden in diesem Abschnitt anhand beispielhafter praktischer Ergebnisse belegt. Zur Bestimmung des Verzögerungsunterschiedes zwischen den Teilnehmern interessieren insbesondere Extremfälle des Netzzuganges. Daher wurden ein Universitätsszenario und ein Heimanwenderszenario ausgewählt. Das Universitätsszenario ist durch hohe Übertragungsraten im Zugangsnetz geprägt. Für das Heimanwenderszenario wurden ein ISDN-Anschluss im Zugangsnetz und geringere Übertragungsraten im Provider-Backbone gewählt.

Das erste Szenario spiegelt den Arbeitsplatz an einer Universität wider und ist der Anbindung des Institutes für Parallele und Verteilte Systeme an den nächstliegenden vertrauenswürdigen Server außerhalb des Campus entnommen. Dieser befindet sich sieben Hops entfernt in *Baden-Württembergs extended LAN* (BelWü). Die Übertragung verläuft über einen Abschnitt mit einer Übertragungskapazität von 100 Mbit/s, ein Campus-Netzwerk mit 622 Mbit/s Übertragungskapazität, drei 1-Gbit/s-MAN-Übertragungsabschnitte und ein 100-Mbit/s-LAN beim Dienstanbieter.

Das zweite Szenario ist ein ISDN-Einwahlszenario zu einem Provider, der einen vertrauenswürdigen Zeitserver anbietet. Hierfür wurde angenommen, dass der 64-kbit/s-ISDN-Kanal über einen 2-Mbit/s-Übertragungsabschnitt in ein Backbone mit 34 Mbit/s Übertragungskapazität mündet. Der Server ist an dieses Backbone über einen 100-Mbit/s-Übertragungsabschnitt angeschlossen.

Für das Zugangsnetz im Einwahlszenario werden 2 Situationen unterschieden. Als erste Situation wird ein belastetes Netzwerk betrachtet, in der der Teilnehmer z. B. ein großes Volumen an Informationen oder andere Datenströme (etwa Dateitransfer oder HTTP-Daten) gleichzeitig empfängt. Diese Situation wird mit einem Netzauslastungsfaktor von $\rho = 0.9$ beschrieben. In der zweiten Situation wird im Falle der Schlüsselserver- und Zeitserver-Verfahren die Möglichkeit der Trennung der Echtzeitdaten von anderen Datenströmen im Zugangsnetz ausgenutzt. Dabei handelt es sich beispielsweise um in RFC2689 [Bormann 1999] vorgeschlagene Mechanismen, die Echtzeitdaten auf spezielle Weise im Zugangsnetz zum Teilnehmer berücksichtigen können. Falls diese nicht verfügbar sind, wird dies durch ausschließliche Verwendung eines ISDN-Kanals für die Schlüssel- und Zeitnachrichten erreicht. Die Informationsnachrichten können dann parallel über den zweiten ISDN-Kanal empfangen werden. Die Auslastung des separaten Netzzuganges für die Schlüsselnachrichten soll $\rho = 0.5$ betragen. Das spiegelt die Tatsache wider, dass der separate Kanal nicht belastet ist, jedoch für jede Informationsnachricht eine Schlüsselnachricht überträgt. Die Netzwerkauslastung des separaten Kanals im Zeitserver-Verfahren richtet sich nach der maximal tolerierten Gangabweichung der Uhren der sicheren Hardware. Sollen die Uhren mit nicht mehr als $J_{Uhr} = 2\,\text{ms}$ zum Verzögerungsunterschied im Zeitserver-Verfahren beitragen, darf das maximale Synchronisationsintervall nicht größer sein als

$$\tau^S = \frac{J_{Uhr}}{2\rho_{max}} = \frac{2\,\text{ms}}{2 \cdot 10^{-6}} = 1000\,\text{s}.$$

Um Toleranz gegenüber Nachrichtenverlusten zu haben, sollen 10 Zeitnachrichten im Intervall empfangen werden. Die Auslastung des separaten Netzzuganges ist daher mit

$$\rho = \frac{\lambda}{\mu} = \lambda \frac{l_{ZN}}{C} = \frac{10}{1000\,\mathrm{s}} \cdot \frac{56\,\mathrm{Byte}}{64\,\mathrm{kbit/s}} = 7 \cdot 10^{-5}$$

gegeben.

Für die Berechnung werden folgende Annahmen getroffen:

- Die Auslieferungsverzögerung ist vom Sender so bestimmt worden, dass die Informationen vor dem Auslieferungszeitpunkt auf den Servern bzw. beim Teilnehmer vorliegen.
- Um eine bessere Vergleichbarkeit der Ergebnisse zu erzielen, wird weiterhin angenommen, dass eine Vielzahl von Servern zur Verfügung steht und die räumliche Entfernung des Teilnehmers zum nächstgelegenen Server gering ist. Daher werden für die Berechnung der Verzögerung der Nachrichten zwischen Server und Teilnehmer die Signallaufzeiten vernachlässigt.
- Die Router können die Pakete mit der Geschwindigkeit des nächsten Übertragungsabschnittes verarbeiten. Die Verarbeitungszeit der Knoten wird deshalb vernachlässigt.
- Für den durch die sichere Hardware verursachten Verzögerungsunterschied werden die im Abschnitt 5.2.5 noch zu besprechenden Ergebnisse des Zeitserver-Verfahrens verwendet.
- Der durch Entschlüsselung der Informationen bei den Teilnehmern entstehende Verzögerungsunterschied ist vernachlässigbar gering. Die Verwendung baugleicher Smart Cards stellt sicher, dass keine Verzögerungsunterschiede durch unterschiedliche Verarbeitungsleistung der sicheren Hardware hervorgerufen werden.
- Um eine bessere Anwendbarkeit der Ergebnisse zu erreichen, erfolgen die Untersuchungen getrennt für Netzwerk und sichere Hardware. Damit wird den für die sichere Hardware zu erwartenden größeren Technologiesprüngen Rechnung getragen.
- Weiterhin werden folgende Annahmen getroffen: Die Paketlänge des Hintergrundverkehrs soll mit 500 Byte konstant sein. Der Hintergrundverkehr erzeugt eine

Netzauslastung von $\rho = 0.9$. Datennachrichten haben ebenfalls eine Länge von 500 Byte. Schlüsselnachrichten haben eine Länge von 50 Byte (8 Byte Sicherungsschicht-Protokoll-Kopf z. B. PPP, 20 Byte UDP-Kopf, 8 Byte allgemeine Informationen und Server-ID, 6 Byte Nachrichten-ID, 8 Byte DES-Informationsschlüssel). Zeitnachrichten haben eine Länge von 56 Byte (8 Byte PPP-Kopf, 20 Byte UDP-Kopf, 8 Byte allgemeine Informationen, 8 Byte Zeitstempel, 8 Byte Schlüssel-ID, 4 Byte Authentifizierung mittels DES-CBC).

Entsprechend den getroffenen Annahmen ergibt sich aus Gleichung 5.1 der Verzögerungsunterschied einer Nachricht als:

$$J_i = \max_{\forall T_j \in R, T_j \in DSrv_k} \{\delta_{kj}^{AN}\} - \min_{\forall T_j \in R, T_j \in DSrv_k} \{\delta_{kj}^{AN}\} = \delta_{k\,ISDN}^{AN} - \delta_{k\,UNI}^{AN}.$$

Unter Zuhilfenahme des Modells der Übertragungstrecke zwischen Server und Teilnehmer wird der Erwartungswert $E[J_{DSrv}] = E[\delta_{k\,ISDN}^{AN}] - E[\delta_{k\,UNI}^{AN}]$ für den Verzögerungsunterschied der Nachrichten im Datenserver-Verfahren berechnet. Der Erwartungswert und die Varianz der Verzögerung der Auslieferungsnachricht des ISDN-Teilnehmers beträgt:

$$E[\delta_{k\,ISDN}^{AN}] = \delta_k^{\ddot{U}} + \sum_{r=1}^{n} \left(\frac{\lambda_r \cdot E[(B_r)^2]}{2(1-\rho_r)} + \delta_r^{\ddot{U}} \right)$$

$$Var[\delta_{k\,ISDN}^{AN}] = \sum_{r=1}^{n} \left((E[W_r])^2 + \frac{\lambda_r \cdot E[(B_r)^3]}{3(1-\rho_r)} \right).$$

Da die Paketsendedauer für einen bestimmten Übertragungsabschnitt konstant ist, wird das zweite und dritte Moment der Sendezeit als $E[(B)^2] = (E[B])^2$ und $E[(B)^3] = (E[B])^3$ berechnet.

Mit $\lambda = \rho / E[B]$ ergibt sich:

$$E[\delta_{k\,ISDN}^{AN}] = \delta_k^{\ddot{U}} + \sum_{r=1}^{n} \left(\frac{\rho_r \cdot (E[B_r])^2}{E[B_r] \cdot 2(1-\rho_r)} + \delta_r^{\ddot{U}} \right) = \delta_k^{\ddot{U}} + \sum_{r=1}^{n} \left(\frac{\rho_r \cdot E[B_r]}{2(1-\rho_r)} + \delta_r^{\ddot{U}} \right)$$

$$Var[\delta_{k\,ISDN}^{AN}] = \sum_{r=1}^{n} \left((E[W_r])^2 + \frac{\rho_r \cdot (E[B_r])^3}{E[B_r] \cdot 3(1-\rho_r)} \right) = \sum_{r=1}^{n} \left((E[W_r])^2 + \frac{\rho_r \cdot (E[B_r])^2}{3(1-\rho_r)} \right).$$

Die Bediendauer entspricht der Paketsendedauer l_L/C_r der Hintergrundlast.

$$E[\delta^{AN}_{k\,ISDN}] = \delta^{\ddot{U}}_k + \sum_{r=1}^{n}\left(\frac{\rho_r \cdot \frac{l_L}{C_r}}{2(1-\rho_r)} + \delta^{\ddot{U}}_r\right)$$

$$Var[\delta^{AN}_{k\,ISDN}] = \sum_{r=1}^{n}\left((E[W_r])^2 + \frac{\rho_r \cdot \left(\frac{l_L}{C_r}\right)^2}{3(1-\rho_r)}\right)$$

$$= \sum_{r=1}^{n}\left(\left(\frac{\rho_r \cdot \frac{l_L}{C_r}}{2(1-\rho_r)}\right)^2 + \frac{\rho_r \cdot \left(\frac{l_L}{C_r}\right)^2}{3(1-\rho_r)}\right) = \sum_{r=1}^{n}\left(\frac{(4\rho_r - {\rho_r}^2) \cdot \left(\frac{l_L}{C_r}\right)^2}{12(1-\rho_r)^2}\right)$$

Für das ISDN-Einwahlszenario folgt eine Verzögerung von 355 ms mit einer Varianz von 91 ms, da:

$$\begin{aligned} E[\delta^{AN}_{k\,ISDN}] &= \frac{500\,\text{Byte}}{100\,\text{Mbit/s}} + \frac{0.9 \cdot \frac{500\,\text{Byte}}{34\,\text{Mbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{34\,\text{Mbit/s}} + \frac{0.9 \cdot \frac{500\,\text{Byte}}{2\,\text{Mbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{2\,\text{Mbit/s}} \\ &\quad + \frac{0.9 \cdot \frac{500\,\text{Byte}}{64\,\text{kbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{64\,\text{kbit/s}} \\ &\approx 355\,\text{ms} \\ Var[\delta^{AN}_{k\,ISDN}] &= \frac{4 \cdot 0.9 - 0.9^2}{12(1-0.9)^2}\left(\left(\frac{500\,\text{Byte}}{34\,\text{Mbit/s}}\right)^2 + \left(\frac{500\,\text{Byte}}{2\,\text{Mbit/s}}\right)^2 + \left(\frac{500\,\text{Byte}}{64\,\text{kbit/s}}\right)^2\right) \\ &\approx 91\,\text{ms} \end{aligned}$$

und für das Universitätsszenario eine Verzögerung von 0.58 ms mit einer Varianz von $7.6 \cdot 10^{-5}$ ms:

$$E[\delta^{AN}_{k\,UNI}] = \delta^{\ddot{U}}_k + \sum_{r=1}^{n}\left(\frac{\rho_r \cdot \frac{l_L}{C_r}}{2(1-\rho_r)} + \delta^{\ddot{U}}_r\right)$$

$$
\begin{aligned}
&= \frac{500\,\text{Byte}}{100\,\text{Mbit/s}} + \frac{0.9 \cdot \frac{500\,\text{Byte}}{100\,\text{Mbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{100\,\text{Mbit/s}} \\
&\quad + \frac{0.9 \cdot \frac{500\,\text{Byte}}{622\,\text{Mbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{622\,\text{Mbit/s}} \\
&\quad + \frac{0.9 \cdot \frac{500\,\text{Byte}}{1\,\text{Gbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{1\,\text{Gbit/s}} + \frac{0.9 \cdot \frac{500\,\text{Byte}}{1\,\text{Gbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{1\,\text{Gbit/s}} \\
&\quad + \frac{0.9 \cdot \frac{500\,\text{Byte}}{1\,\text{Gbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{1\,\text{Gbit/s}} + \frac{0.9 \cdot \frac{500\,\text{Byte}}{100\,\text{Mbit/s}}}{2(1-0.9)} + \frac{500\,\text{Byte}}{100\,\text{Mbit/s}} \\
&\approx 0.58\,\text{ms}
\end{aligned}
$$

$$
\begin{aligned}
Var[\delta^{AN}_{kUNI}] = \frac{4 \cdot 0.9 - 0.9^2}{12(1-0.9)^2} \Bigg(&\left(\frac{500\,\text{Byte}}{100\,\text{Mbit/s}}\right)^2 + \left(\frac{500\,\text{Byte}}{622\,\text{Mbit/s}}\right)^2 + \left(\frac{500\,\text{Byte}}{1\,\text{Gbit/s}}\right)^2 \\
&+ \left(\frac{500\,\text{Byte}}{1\,\text{Gbit/s}}\right)^2 + \left(\frac{500\,\text{Byte}}{1\,\text{Gbit/s}}\right)^2 + \left(\frac{500\,\text{Byte}}{100\,\text{Mbit/s}}\right)^2 \Bigg) \\
\approx 7.6 \cdot 10^{-5}\,&\text{ms}.
\end{aligned}
$$

Der Verzögerungsunterschied für das Datenserver-Verfahren beträgt daher:

$$E[J_{DSrv}] = E[\delta^{AN}_{kISDN}] - E[\delta^{AN}_{kUNI}] \approx 355\,\text{ms} - 0.58\,\text{ms} \approx 354\,\text{ms}$$

mit einer Varianz von:

$$Var[J_{DSrv}] = Var[\delta^{AN}_{kISDN}] + Var[\delta^{AN}_{kUNI}] \approx 91\,\text{ms} + 7.6 \cdot 10^{-5}\,\text{ms} \approx 91\,\text{ms}.$$

Analog erfährt die Auslieferungsnachricht beim Schlüsselserver-Verfahren einen Verzögerungsunterschied im belasteten Zugangsnetz von:

$$
\begin{aligned}
E[J_{SSrv}] &= E[\delta^{AN}_{kISDN}] - E[\delta^{AN}_{kUNI}] \approx 297\,\text{ms} - 0.46\,\text{ms} \approx 297\,\text{ms} \\
Var[J_{SSrv}] &= Var[\delta^{AN}_{kISDN}] + Var[\delta^{AN}_{kUNI}] \approx 91\,\text{ms} + 7.6 \cdot 10^{-5}\,\text{ms} \approx 91\,\text{ms}.
\end{aligned}
$$

Unter Ausnutzung der Trennung von Informations- und Schlüsselnachrichten mit ($\rho_3 =$

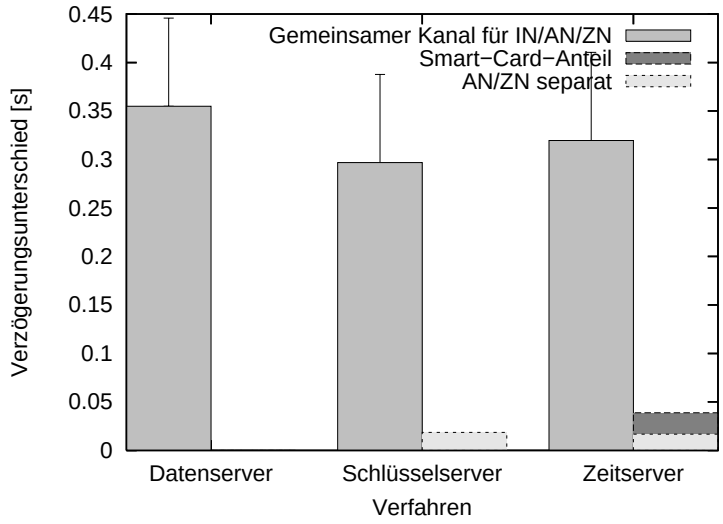


Abbildung 5.4: Leistungsvergleich der Verfahren zur Reduktion des Verzögerungsunterschiedes zwischen Teilnehmern, die über einen ISDN-Anschluss oder über ein Hochgeschwindigkeitsnetz mit dem Server verbunden sind. Für das Schlüssel- und Zeitserver-Verfahren ist außerdem der Verzögerungsunterschied bei Übertragung der Auslieferungs- bzw. Zeitnachrichten im einem separaten ISDN-Kanal dargestellt. IN=Informationsnachricht, AN=Auslieferungsnachricht, ZN=Zeitnachricht.

0.5) für die Auslastung des Übertragungskanals der Auslieferungsnachrichten ergibt sich:

$$\begin{aligned} E[J_{SSrv}] =& E[\delta^{AN}_{k\,ISDN}] - E[\delta^{AN}_{k\,UNI}] \approx 19\,\text{ms} - 0.46\,\text{ms} \approx 19\,\text{ms} \\ Var[J_{SSrv}] =& E[\delta^{AN}_{k\,ISDN}] + E[\delta^{AN}_{k\,UNI}] \approx 0.12\,\text{ms} + 7.6 \cdot 10^{-5}\,\text{ms} \approx 0.12\,\text{ms}. \end{aligned}$$

Aus Gleichung 5.3 ergibt sich für die Szenarien im Zeitserver-Verfahren der Verzögerungsunterschied als:

$$\begin{aligned} J_{ZSrv} &= \delta^{ZN}_{k\,ISDN} - \delta^{ZN}_{k\,UNI} + J_{Uhr} \\ &= \delta^{\ddot{U}\,S\,ZN}_{k\,ISDN} + \delta^{\ddot{U}\,ZN}_{SH\,ISDN} + \delta^{E\,ZN}_{SH\,ISDN} - \delta^{\ddot{U}\,S\,ZN}_{k\,UNI} + \delta^{\ddot{U}\,ZN}_{SH\,UNI} + \delta^{E\,ZN}_{SH\,UNI} + J_{Uhr}. \end{aligned}$$

Der durch die sichere Hardware hervorgerufene Anteil

$$\delta^{\ddot{U}\,ZN}_{SH\,ISDN} + \delta^{E\,ZN}_{SH\,ISDN} - \delta^{\ddot{U}\,ZN}_{SH\,UNI} + \delta^{E\,ZN}_{SH\,UNI}$$

hängt stark von der verwendeten Technologie der sicheren Hardware ab. Für eine marktübliche Smart Card wird dieser Verzögerungsunterschied in Abschnitt 5.2.5 zur Authenti-

fizierung mit DES-CBC als $J_{SH}^{ZN} = 20\,\text{ms}$ bestimmt. Der Verzögerungsunterschied der Zeitnachricht unter Verwendung von Gleichung 5.4 beträgt

$$E[J_{ZSrv}] = E[\delta_{kISDN}^{ZN}] - E[\delta_{kUNI}^{ZN}] + J_{SH}^{ZN} + J_{Uhr} \approx 298\,\text{ms} - 0.46\,\text{ms} + 20\,\text{ms} + 2\,\text{ms} \approx 320\,\text{ms}$$

und die durch die Übertragungsstrecke hervorgerufene Varianz des Jitters

$$Var[J_{ZSrv}] = Var[\delta_{kISDN}^{ZN}] + Var[\delta_{kUNI}^{ZN}] \approx 91\,\text{ms} + 7.6 \cdot 10^{-5}\,\text{ms} \approx 91\,\text{ms}.$$

Für die Situation unter Ausnutzung der Trennung von Informations- und Zeitnachricht ist ein Verzögerungsunterschied von

$$\begin{aligned} E[J_{ZSrv}] &= E[\delta_{kISDN}^{ZN}] - E[\delta_{kUNI}^{ZN}] + J_{SH}^{ZN} + J_{Uhr} \\ &= 17\,\text{ms} - 0.46\,\text{ms} + 20\,\text{ms} + 2\,\text{ms} \approx 39\,\text{ms} \\ Var[J_{ZSrv}] &= Var[\delta_{kISDN}^{ZN}] + Var[\delta_{kUNI}^{ZN}] \approx 0.093\,\text{ms} + 7.6 \cdot 10^{-5}\,\text{ms} \approx 0.093\,\text{ms} \end{aligned}$$

zu erwarten.

Abbildung 5.4 stellt den Verzögerungsunterschied der Verfahren in den einzelnen Situationen dar. Es ist deutlich erkennbar, dass mit dem Schlüssel- und Zeitserver-Verfahren der Verzögerungsunterschied am effektivsten reduziert wird. Hierzu trägt vor allem die Trennung von Informations- und Auslieferungsnachricht bzw. von Informations- und Zeitnachricht bei. Ohne die Trennung der Nachrichten ist die Reduktion des Verzögerungsunterschiedes im Vergleich zum Datenserver-Verfahren weniger signifikant, da die Paketlänge der Informationsnachrichten moderat mit nur 500 Byte angenommen wurde. Zu beachten ist, dass die Abschätzungen keine Signallaufzeiten enthalten, die Verfahren hingegen unterschiedliche Anforderungen an die Server stellen und daher eine unterschiedliche Anzahl und Platzierung der Server zu vermuten ist. Da an die Zeitserver die geringsten Anforderungen gestellt werden, kann ihnen die geringste maximale Entfernung von einem Teilnehmer unterstellt werden. In den betrachteten Szenarien dürfte beispielsweise der vom Teilnehmer am weitesten entfernte Server des Schlüsselserver-Verfahrens maximal 4000 km weiter vom Teilnehmer entfernt sein, als der am weitesten entfernte Zeitserver vom Teilnehmer des Zeitserver-Verfahrens. Anderenfalls würde das Schlüsselserver-Verfahren den Vorteil des geringeren Verzögerungsunterschiedes verlieren.

5.2.5 Leistungsbewertung einer Smart-Card-Implementierung

Um die in Abschnitt 4.5.3.1 entworfenen Protokolle zu validieren und die mit gegenwärtiger sicherer Hardware erreichbare Leistung zu bestimmen, wurde ein Prototyp implementiert [Feufel 2001]. Wegen der derzeit noch fehlenden Uhrenunterstützung wurden die Protokolle wie folgt abgeändert und eine provisorische Lösung implementiert. Der Zeitserver sendet die zertifizierten Zeitstempel mit einer hohen Rate, so dass für jede Nachricht ein aktueller Zeitstempel zur Verfügung steht. Die Uhr auf der Smart Card wird demzufolge emuliert, wobei die authentifizierten Zeitstempel zum Weiterstellen verwendet werden.

Durch die fehlende Uhrenunterstützung ist es der Smart Card nicht möglich, die Verzögerung der Authentifizierung der Zeitstempel beim Abgleich der Uhr mit dem Zeitstempel zu berücksichtigen. Da die Verzögerung der Bereitstellung des Nachrichtenschlüssels dadurch wesentlich erhöht wird, wurden außer dem asymmetrischen Authentifizierungsverfahren noch symmetrische Verfahren untersucht. Dazu gehören DES und Triple DES in Verbindung mit den Hash-Algorithmen MD5 und SHA-1 sowie DES-CBC mit *Message Authentication Code*.

Für die Messungen wurde ein HP Vectra VL, PII-300 MHz unter Windows NT 4.0 mit Service Pack 6 verwendet. Als Java-Umgebung kam JRE 1.3.0-C mit Java Hotspot Client VM (mixed mode) zum Einsatz. Bei der für die Messungen zur Verfügung stehenden Smart Card handelt es sich um eine Gemplus GemXpressso 211 PK Full Version. Dies ist eine Java Card nach Spezifikation 2.1.1. Sie basiert auf einem 8-Bit-Microcontroller von Phillips. Der Kryptographie-Koprozessor der Karte unterstützt den Secure Hash Algorithm (SHA-1), MD5, DES und RSA. Die Kommunikation mit dem Lesegerät erfolgte über das T0-Protokoll. Das Lesegerät Gemplus GemPC 410 ist über die serielle Schnittstelle an das Open Card Framework 1.2, den Gemplus CardTerminal Treiber Version 4.1 und die JavaComm API angebunden.

In Abbildung 5.5 wird die Verifizierungsdauer für Zeitstempel für die verschiedenen Authentifizierungsverfahren gezeigt. Dafür wurde in der Anwendung des PC die Zeitspanne vom Absenden der Information zur Smart Card bis zur Rückmeldung der erfolgten Authentifizierung gemessen. Die Verzögerung des RSA-basierten Verfahrens ist mit einem Mittelwert von 865 ms ungefähr doppelt so hoch wie die der symmetrischen Verfahren SHA-1 DES (471 ms), MD5 DES (453 ms) und DES CBC (384 ms). Da die Smart Card keine Uhrenunterstützung bietet, hat die Verzögerung im vorliegenden Fall Einfluss auf das kleinste Intervall zwischen zwei Auslieferungszeiten. Mit Uhrenunterstützung ist die

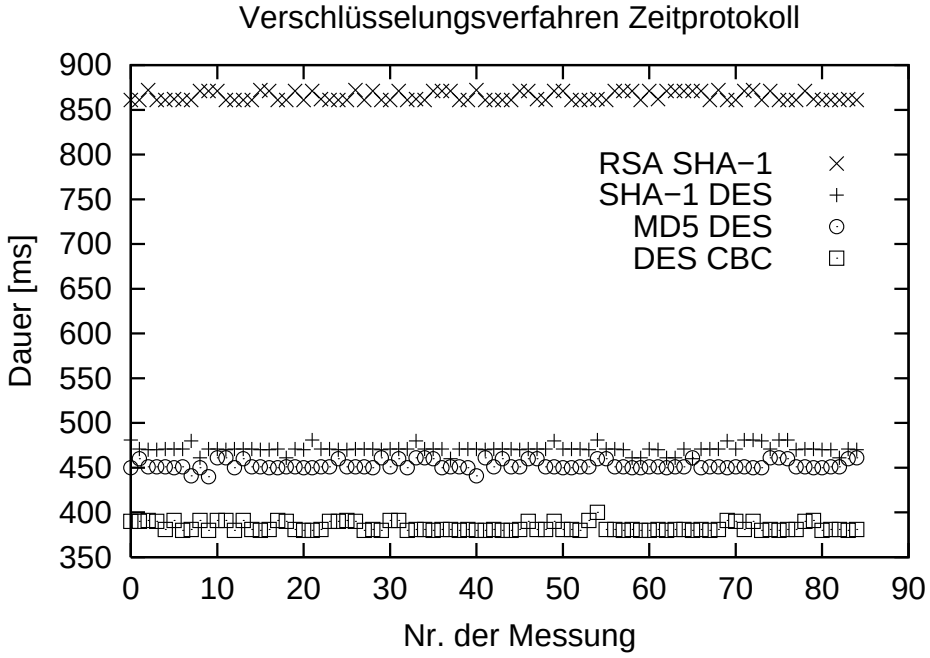


Abbildung 5.5: Verifizierungsdauer für Zeitstempel mit verschiedenen Verifizierungsverfahren

Authentifizierung der Zeitstempel dagegen unabhängig vom Entschlüsseln des Nachrichtenschlüssels und hat vernachlässigbare Auswirkungen auf das Intervall zwischen zwei Auslieferungszeiten.

Der Verzögerungsunterschied zwischen den Teilnehmern wird durch die Unterschiede in der Verifizierungsdauer J_{SH}^{ZN} verursacht. In Abbildung 5.6 ist die kumulative Verteilung der Verzögerungsunterschiede der Verifizierungsdauer dargestellt. Der Anstieg $m = 0$ der Kurven in den Intervallen $[1\,\text{ms}, 9\,\text{ms}]$ und $[11\,\text{ms}, 19\,\text{ms}]$ deutet auf eine Schedulergranularität des Betriebssystems des Client-PC von 10 ms hin, was in weiteren Experimenten bestätigt werden konnte. Bei RSA liegen 60 %, bei MD5 DES und DES CBC über 70 % der Messwerte nahe dem Minimum und 99 % der Messwerte weichen nicht mehr als 11 ms vom Minimum ab. Bei SHA-1 DES weichen 85 % der Messwerte nicht mehr als 11 ms vom Minimum ab. Das Messverfahren wie auch die Anwendungsleistung sind von der Schedulergranularität abhängig. Eine Reduktion der Zeitscheibengröße würde das Potenzial der Smart Card noch besser ausschöpfen und die schon jetzt sehr geringen Verzögerungsunterschiede der Authentifizierung weiter minimieren.

In Tabelle 5.3 sind die Messergebnisse der Einzelverfahren (a-c) und einer Kombination des mit einer geringeren Verzögerung behafteten Verfahrens zur Uhrenaktualisierung

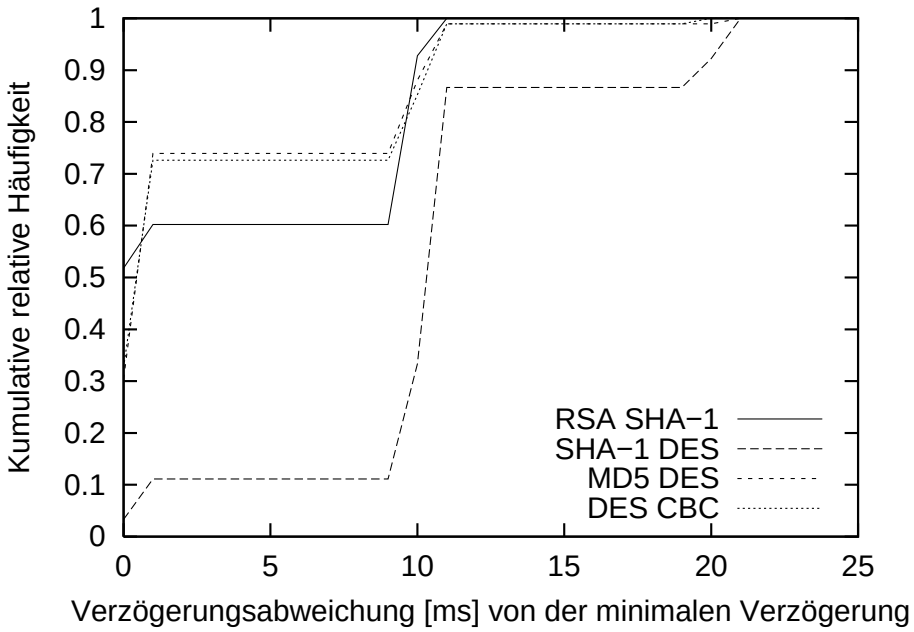


Abbildung 5.6: Kumulative Verteilung der Verzögerungsunterschiede der Verifizierungsdauer für Zeitstempel mit verschiedenen Verifizierungsverfahren

(DES CBC) (b) mit dem Verfahren zur Bereitstellung des Nachrichtenschlüssels mittels DES (c) zusammengefasst. Die mit dem kombinierten Verfahren (d) erhaltene Verzögerung von 1030 ms entspricht zugleich dem kleinsten Intervall, in dem die Anwendung aufeinanderfolgende Datenaktualisierungen vornehmen kann.

Zukünftige Entwicklungen im Bereich sicherer Hardware werden zu einer weiteren Leistungssteigerung führen. Interaktive Anwendungen erfordern ein Absenken der Verzögerungen unter die hier gemessenen Werte. Uhrenunterstützung wird die Gesamtverzögerung auf das Niveau der Verzögerung zur Bereitstellung des Nachrichtenschlüssels absenken. Die durch die Datenübertragung zur Smart Card verursachte Verzögerung kann durch verbesserte Lesegeräte mit höherer Datenübertragungsrate, z. B. über die USB-Schnittstelle des Clients, reduziert werden. Implementierungen in Assembler sind gegenüber Java-Implementierungen im Allgemeinen zwei- bis dreimal schneller [Johannes 2000]. Bei der vorliegenden Anwendung ist das Optimierungspotenzial jedoch begrenzt, da die Laufzeit der Programme vorwiegend durch die Leistung des Kryptographie-Koprozessors bedingt ist. Eine Steigerung der Verarbeitungsleistung verspricht die nächste Generation der Smart Cards mit auf 32-Bit-RISC-Technologie basierenden Prozessoren und dementsprechend verbesserten Koprozessoren. Die Bereitstellung anderer Verschlüsselungsverfahren wird ebenfalls zu einer Leistungssteigerung beitragen. So benötigen auf elliptischen

Tabelle 5.3: Messergebnisse von Verzögerung und Verzögerungsunterschied des Uhrensynchronisations- und Datenauslieferungsprotokolls durch die Smart Card

Algorithmus	Mittlere Verzögerung (ms)	Verzögerungsunterschied (ms)
(a) Authentifizierung Zeitstempel mittels RSA SHA-1	865	11
(b) Authentifizierung Zeitstempel mittels DES CBC	384	20
(c) Bereitstellung Nachrichtenschlüssel mittels DES	572	31
(d) Kombination der Verfahren von (b) und (c)	1030	100

Kurven basierende Verschlüsselungsverfahren bei gleicher Sicherheit im Vergleich zu RSA eine geringere Verarbeitungsleistung.

Darüber hinaus sind auch andere Formen sicherer Hardware denkbar, wie z. B. USB-Stecker mit darin integrierten Verarbeitungseinheiten. Sie bieten jedoch allein schon wegen ihrer Größe ein unerwünscht hohes Angriffspotenzial, was die Sicherheit der Lösung beeinträchtigen würde.

5.3 Zusammenfassung

In Kapitel 5 wird die Leistungsfähigkeit der vorgestellten Verfahren zur Reduktion von Verzögerungsunterschieden zwischen Teilnehmern von Gruppenkommunikation – Datenserver-, Schlüsselserver- und Zeitserver-Verfahren – bewertet und miteinander verglichen. Im Abschnitt 5.1 werden einige wesentliche Faktoren besprochen, die die Größe des Verzögerungsunterschiedes beeinflussen. Hierzu gehören die Entfernung der Server von den Teilnehmern, der Übertragungsabschnitt mit der kleinsten Übertragungskapazität sowie die Länge der zeitkritischen Nachricht. Es wird gezeigt, dass der Verzögerungsunterschied mit dem maximalen Entfernungsunterschied über alle Server-Teilnehmer-Paare linear zunimmt und im Extremfall bis zu 100 ms betragen kann. Aus der Verkürzung des Teilnehmer-Server-Abstandes ergibt sich somit ein hohes Potenzial für die Reduktion des Verzögerungsunterschiedes. Weiterhin wird dargelegt, dass die Differenz zwischen maximaler und minimaler Verzögerung der Nachrichten über alle Server-Teilnehmer-Paare von

der unterschiedlichen Übertragungsrate der einzelnen Übertragungsabschnitte sowie von der Länge der Nachrichten abhängig ist. Wie die Prüfung dieser Faktoren zeigt, führen beim Datenserver-Verfahren geringe Übertragungskapazität und große Nachrichtenlängen zu besonders hohen Verzögerungsunterschieden. Schlüssel- und Zeitserver-Verfahren verursachen dagegen geringere Verzögerungsunterschiede, da sie – infolge der getrennten Versendung der Synchronisationsinformation – von der Länge der Informationsnachrichten unabhängig sind.

Die Ermittlung des Nachrichtenaufwandes ergab für alle Verfahren eine lineare Abhängigkeit der Anzahl der Nachrichten von der Anzahl der Nutznachrichten. Für Nutzdatenraten größer als die Datenrate der Zeitnachrichten ist der Nachrichtenaufwand beim Zeitserver-Verfahren am geringsten.

Im Abschnitt 5.2 werden Nachrichtenverzögerung und der durch die Protokolle bedingte Verzögerungsunterschied einer theoretischen Analyse unterzogen. Für die Verzögerung zwischen Servern und Teilnehmern wird ein M/G/1-Warteschlangenmodell der Übertragungsstrecke erstellt. Anschließend wird der Verzögerungsunterschied zwischen den Teilnehmern anhand eines Universitätsszenarios mit hohen Übertragungsraten und eines Heimanwenderszenarios mit ISDN-Anschluss und geringen Übertragungsraten im Zugangsnetz abgeschätzt. Die Analysen zeigen, dass der Verzögerungsunterschied mit dem Schlüssel- und mit dem Zeitserver-Verfahren um eine Größenordnung effektiver reduziert wird als mit dem Datenserver-Verfahren. Hauptanteil hieran hat die Trennung von Informations- und Auslieferungsnachricht bzw. von Informations- und Zeitnachricht.

Zur Validierung der entworfenen Protokolle und zur Bestimmung der mit gegenwärtiger sicherer Hardware erreichbaren Leistung wurde ein Prototyp mit Smart Cards implementiert. Die Messung der Verifizierungsdauer für verschiedene Authentifizierungsverfahren der Zeitstempel ergab Werte zwischen 380 ms für DES-CBC mit *Message Authentication* Code und 470 ms für SH1 DES. Für das RSA-basierte Verfahren ergaben sich erwartungsgemäß signifikant höhere Werte von ca. 860 ms. Der für die Verfahren entscheidende Verzögerungsunterschied ist äußerst gering. Für das RSA-basierte Verfahren beträgt er 11 ms und für DES-CBC ist er für 99 % der Messwerte ≤ 11 ms mit einem maximalen Verzögerungsunterschied von 20 ms. Geringere Prozesswechselzeiten des Teilnehmerbetriebssystems sollten hier zu noch besseren Ergebnissen führen.

Für interaktive Anwendungen ist ein Absenken der Verzögerungen unter die hier gemessenen Werte erforderlich. Bei den gegenwärtig verfügbaren Smart Cards ist das Optimierungspotenzial jedoch begrenzt, da die Laufzeit der Programme vorwiegend durch die

Leistung des Kryptographie-Koprozessors bestimmt ist. Möglichkeiten für eine weitere Leistungssteigerung im Bereich der sicheren Hardware ergeben sich durch die Entwicklung einer neuen Generation von Smart Cards, die mit Prozessoren auf der Basis der 32-Bit-RISC-Technologie und entsprechend verbesserten Koprozessoren ausgestattet sind sowie durch andere, auf elliptischen Kurven basierende Verschlüsselungsverfahren.

Kapitel 6

Zusammenfassung und Ausblick

Gruppenkommunikation mittels IP-Multicast ist eine hocheffiziente Übertragungsweise einer Nachricht an eine Gruppe von Empfängern. Einschränkungen in der Dienstqualität gegenwärtiger Gruppenkommunikationslösungen im Internet stehen jedoch einer generellen Nutzung dieser Technik noch im Wege. So ist die verzögerungsfaire Nachrichtenauslieferung bisher nicht garantiert. Dieser spezielle Parameter der Gruppenkommunikation ist die Zeitspanne zwischen dem ersten und letzten Eintreffen einer Nachricht bei einer Gruppe von Empfängern. Mit dem Ziel, die Verzögerungsunterschiede so klein wie möglich zu halten, werden in der vorliegenden Arbeit drei innovative Ansätze zur Reduktion von Verzögerungsunterschieden bei der Gruppenkommunikation im Internet vorgestellt und ihre Leistungsfähigkeit bewertet. Von einem verzögerungsfairen Dienst profitieren insbesondere nicht-kooperative Echtzeitanwendungen, darunter bestimmte Informationsdienste, elektronische Märkte und Spiele.

Die Entwicklung eines Verfahrens zur Messung von Verzögerungsunterschieden im Internet bildet den *ersten Schwerpunkt* der Arbeit. Um den administrativen Aufwand für die Messungen gering zu halten, wurde einem passiven Verfahren gegenüber einem aktiven der Vorzug gegeben. Die erforderlichen Daten zur Bestimmung des Verzögerungsunterschiedes wurden durch Mithören von Multicast-Verkehr real existierender Gruppen ermittelt, ohne dass zusätzliche Mess-Software bei Multicast-Teilnehmern installiert werden musste oder Routeränderungen erforderlich waren.

Zur Erfassung der Verzögerungsunterschiede wurden die Zeitstempel des RTP Control Protocol (RTCP) so ausgenutzt, dass die Einwegzeit spezieller Pakete bestimmt und da-

mit der Verzögerungsunterschied dieser Pakete zwischen den Empfängern berechnet werden konnte. Da die verwendeten Zeitstempel von Uhren auf unterschiedlichen Computern generiert werden und somit nicht notwendigerweise synchronisiert sind, wurden die Uhren mittels einer Referenzuhr auf Ganggenauigkeit untersucht und erforderlichenfalls korrigiert. Der Zeitversatz der Uhr des entfernten Rechners ergab sich aus den Zeitstempeln des Paketes mit der kleinsten Umlaufzeit. Indem zwischen Messwerten mit kleinster Umlaufzeit interpoliert wird, konnte zusätzlich zum Zeitversatz auch die Gangabweichung der Uhren bestimmt werden. Damit war die Möglichkeit zur Korrektur der RTCP-Zeitstempel und somit auch der Einwegzeit gegeben.

Die bei Videokonferenzsitzungen im Multicast Backbone (MBone) gemessenen konkreten Verzögerungsunterschiede von Steuerungsnachrichten vermitteln einen Einblick in die Größenordnung der im Internet anzutreffenden Verzögerungsunterschiede zwischen Gruppenkommunikationsteilnehmern. Die bereits bei kleineren Gruppen nachgewiesenen Verzögerungsunterschiede bis zum einstelligen Sekundenbereich sind für nicht-kooperative Echtzeitanwendungen, wie z. B. Börseninformationsdienste, elektronische Märkte und Spiele, nicht tolerierbar. Daher widmete sich der *zweite Schwerpunkt* der Arbeit der Entwicklung von Verfahren, die die Vorteile der Gruppenkommunikation nutzen, zugleich aber eine faire Auslieferung von Informationen bei den Teilnehmern sicherstellen.

Bestehende Synchronisationsverfahren konnten auf die Problemstellung nicht angewendet werden, weil sie Sicherheitsaspekte nicht berücksichtigen oder umfangreiche Änderungen in den Netzknoten verlangen. Sicherheitsaspekte spielen aber für die faire Nachrichtenauslieferung eine entscheidende Rolle, da bei der Verteilung werthaltiger Informationen durch Gruppenkommunikation ein kooperatives Verhalten der Empfänger nicht vorausgesetzt werden kann. Annahmen über Vertrauen in die involvierten Entitäten wurden daher auf wenige punktuelle Vertrauensbereiche begrenzt. Innerhalb dieser gleichen die hier vorgestellten Verfahren Verzögerungsunterschiede der Nachrichten zwischen Empfängern mittels spezieller Server oder durch sichere Hardware empfängernah in der Anwendungsschicht aus. Da keine Annahmen über die Netzwerkknoten getroffen werden, passen sich die Verfahren in die gegenwärtige Internetstruktur ohne Routeränderungen ein. Dadurch können die Verfahren schnell und gezielt eingesetzt werden.

Die erste Realisierungsvariante stützt sich auf Datenserver, die die von einem Sender erhaltenen Nachrichten nicht unmittelbar, sondern erst zu einem vom Sender vorgegebenen Auslieferungszeitpunkt an die Empfänger weiterleiten. Dadurch wird eine Glättung des Verzögerungsunterschiedes zwischen den Empfängern erreicht.

Bei der zweiten Variante erfolgt die Übertragung der über Multicast verschlüsselt zur Verfügung gestellten Informationsnachrichten getrennt von der Übertragung von Schlüsselnachrichten, die bei einem Schlüsselserver hinterlegt werden. Die Empfänger können die erhaltenen Nachrichten erst dann entschlüsseln, wenn sie den hierzu erforderlichen Schlüssel - zu einem vom Sender vorgegebenen Auslieferungszeitpunkt - vom Schlüsselserver erhalten haben. Beide Varianten zeichnen sich durch ein hohes Maß an Sicherheit aus, da sich die sicherheitsrelevanten Informationen bis zum Auslieferungszeitpunkt beim Dienstanbieter, nicht beim Teilnehmer, befinden.

In einem weiteren Ansatz, dem Zeitserver-Verfahren, wird die Überwachung des Zeitpunktes zur Veröffentlichung der Informationen auf sichere Hardware der Endgeräte übertragen. Die Serverfunktionalität kann sich dadurch auf vertrauenswürdige Zeitinformation beschränken. Aus der vollständigen Trennung von Informations- und Zeitnachrichten ergibt sich die größtmögliche Reduktion des Verzögerungsunterschiedes auf der Server-Teilnehmer-Strecke. Sowohl die Informations- als auch die Schlüsselauslieferung erfolgen direkt über Multicast bis zum Empfänger, was den Nachrichtenaufwand minimiert. Hierfür wurden zwei Protokolle entwickelt, die die Synchronisation der Smart-Card-Uhren und die Auslieferung der Daten ermöglichen. Die Erweiterung dieser sowie der vorgenannten Realisierungsvarianten auf dynamische Verfahren durch Rückmeldung der tatsächlichen Verzögerung berücksichtigt die aktuelle Netzwerklast und erlaubt gleichzeitig, die Gesamtverzögerung der Nachrichten zu reduzieren.

Die Analyse der Übertragung im Zugangsnetzwerk zeigt, dass der Verzögerungsunterschied umso stärker ausfällt, je länger die Nachrichten sind und je geringer die Übertragungskapazität ist. Daher kann das Verfahren mit sicherer Hardware beim Empfänger durch die kurzen Zeitnachrichten den Verzögerungsunterschied am besten ausgleichen. Außerdem erweist es sich auf Grund der Wiederholbarkeit der Zeitnachrichten als tolerant gegenüber Übertragungsfehlern. Die Nachrichtenverzögerung und der durch die Protokolle bedingte Verzögerungsunterschied werden einer theoretischen Analyse unterzogen. Hierfür wird ein Modell der Übertragungsstrecke zwischen Servern und Teilnehmern erstellt. Die Implementierung eines Prototyps beim Teilnehmer mittels Smart Cards erlaubt es, die entworfenen Protokolle zu validieren und die mit gegenwärtiger sicherer Hardware erreichbare Leistung zu bestimmen. Die auf zwei Szenarien mit unterschiedlicher Übertragungskapazität im Zugangsnetzwerk basierenden Analysen ergeben, dass das Schlüsselserver- und das Zeitserver-Verfahren den Verzögerungsunterschied um eine Größenordnung effektiver reduzieren als das Datenserver-Verfahren. Hauptanteil hieran hat die Trennung von

Informations- und Auslieferungsnachricht bzw. von Informations- und Zeitnachricht.

Die Verbreitung der Gruppenkommunikation mittels IP-Multicast hängt in hohem Maße davon ab, ob es gelingt, Fairness-Ansprüche an dieses hocheffiziente Übertragungsverfahren zu befriedigen. Einen Beitrag hierzu sollen die in der Arbeit aufgezeigten Wege zur Reduktion von Verzögerungsunterschieden zwischen Teilnehmern von Multicast-Gruppenkommunikation leisten.

Forschungsbedarf besteht vor allem im Hinblick auf die weitere Reduzierung der Gesamtverzögerung der Nachrichten. Darüber hinaus könnten technologische Verbesserungen zu einer Minimierung des Intervalls zwischen zwei Auslieferungszeiten führen, was z. B. für kombinierte Anwendungen aus Klassen 1 und 2 sowie für die Anwendungsklasse 4 vorteilhaft wäre. Für die Reduzierung der Gesamtverzögerung werden zwei Strategien verfolgt, erstens die orthogonale Kombination der hier vorgestellten Verfahren mit Multilayer-Verfahren und zweitens die Optimierung von Schätzverfahren zur Bestimmung der Auslieferungszeit.

Die Leistungsfähigkeit von Smart Cards, Informationsschlüssel zu entschlüsseln, begrenzt die zur Zeit mögliche Minimierung des Intervalls zwischen zwei Auslieferungszeiten auf etwa 500 ms. Im Zuge der Verbesserung der Leistungsfähigkeit sicherer Hardware und damit beschleunigter Datenübertragungs- und Verschlüsselungsverfahren, ist eine weitere Verringerung der Zeitdauer zwischen zwei aufeinander folgenden Informationsfreigaben zu erwarten.

Die Multilayer-Strategie ist vor allem bei unterschiedlichen Übertragungskapazitäten der Teilnehmer von Bedeutung, da der Ausgleich der Verzögerungsunterschiede zu einer neuen Unfairness führt. Diese besteht darin, dass Teilnehmer mit optimalem Netzzugang nicht die volle Übertragungskapazität ausschöpfen können. Sie sind vielmehr gezwungen, auf Teilnehmer, zu denen die Übertragungsabschnitte eine geringere Übertragungskapazität haben, Rücksicht zu nehmen. Zur Lösung dieses Problems könnten die hier vorgestellten Verfahren mit einem Multilayer-Verfahren kombiniert werden. Multilayer-Verfahren sind bereits für *Congestion Control* von Multicast [Byers et al. 2000] sowie für *Graceful Degradation* multimedialer Datenströme [McCanne 1996] untersucht worden.

Multilayering ermöglicht differenzierte Übertragungsraten und verhindert, dass der Teilnehmer mit der geringsten Übertragungskapazität im Netzzugang die Verzögerung bestimmt. Im vorliegenden Fall könnte z. B. einem Teilnehmer, dessen Netzzugang nur über

geringe Übertragungskapazität verfügt, der für ihn interessante Ausschnitt aus dem Informationsangebot (beispielsweise Kurse des DAX gegenüber Kursen des DAX und des MDAX sowie Kursen von DAX, MDAX und NYSE in zusätzlichen Layers) verzögerungsfair bereitgestellt werden, ohne dass er selbst zur Erhöhung der Gesamtverzögerung der Informationsübertragung beiträgt. Durch die Gruppierung der Teilnehmer anhand der Übertragungskapazität und Nachrichtenlänge werden Verzögerungsunterschiede im Datenserver-Verfahren weiter reduziert. Daraus ergibt sich jedoch auch eine Einschränkung für Multilayer-Verfahren: Teilnehmer mit geringer Übertragungskapazität im Netzzugang erhalten weniger Informationen als solche mit hoher Übertragungskapazität. Soll zu allen Teilnehmern das gleiche Informationsvolumen gelangen, sind Multilayer-Verfahren nicht anwendbar.

Feedback-Steuerung ist eine weitere Möglichkeit, die Gesamtverzögerung zu minimieren. Sie richtet sich auf die Optimierung von Schätzverfahren zur Bestimmung der Auslieferungszeit. Mit einer hohen Auslieferungsverzögerung lassen sich Verzögerungsunterschiede zwar minimieren, jedoch nur unter Inkaufnahme einer hohen Gesamtlaufzeit. Je genauer die Schätzung der Auslieferungszeit ist, desto eher ist eine Optimierung der Auslieferungsverzögerung möglich, ohne neue Verzögerungsunterschiede zu riskieren. Die Qualität der Schätzung der Auslieferungszeit könnte vom Prinzip der Selbstähnlichkeit [Li & Mills 2001, Hagiwara et al. 2001] oder von der Vergabe von Garantien [Rao 2002] profitieren. Voraussetzung hierfür sind Langzeitbeobachtungen von Multicast-Gruppen, die die Dynamik von Verzögerungsunterschieden zwischen Teilnehmern offenbaren. Mit ihrer Hilfe sollte es gelingen, statistische Modelle der Dynamik des Verzögerungsunterschiedes für Feedbackprotokolle zu erhalten und daraus das Kurzzeitverhalten des Verzögerungsunterschiedes vorherzusagen.

Anhang A

Glossar und Abkürzungsverzeichnis

Ad-hoc-Mitteilungen
Durch das Wertpapierhandelsgesetz vorgeschriebene Publikation von außergewöhnlichen, den Aktienkurs potenziell beeinflussenden Unternehmensnachrichten, wie z. B. Gewinnerwartungen, Quartalszahlen und Beteiligungen, mit dem Ziel der gleichzeitigen Information aller Aktionäre einer Gesellschaft.

API – Application Programming Interface
Dokumentierte Software-Schnittstelle, mit deren Hilfe ein Programm die Funktionen eines anderen Programms nutzen kann.

ATM – Asynchronous Transfer Mode
Von der ITU-T für das Breitband-ISDN spezifizierte verbindungsorientierte Vermittlungs- und Multiplextechnik basierend auf Protokoll-Dateneinheiten konstanter Länge von 53 Byte.

Auslieferungsverzögerung
Vom Sender festgelegte Zeitdauer zwischen Sendezeitpunkt und →Auslieferungszeitpunkt der Informationen.

Auslieferungszeitpunkt
Zeitpunkt, zu dem die Informationen frühestens den Vertrauensbereich verlassen dürfen. Summe aus Sendezeitpunkt und →Auslieferungsverzögerung.

CBC – Cipher Block Chaining
CBC ist ein Verschlüsselungsmodus von Blockverschlüsselungsverfahren. Beginnend mit

einem Initialisierungsvektor wird der im Durchlauf $x-1$ verschlüsselte Block über XOR mit dem im Durchlauf x zu verschlüsselnden Block verknüpft und darauf der Schlüssel zur Verschlüsselung des Blockes angewandt. Dadurch hängen alle verschlüsselten Blöcke von den vorangegangenen Blöcken ab.

Core Based Trees
Ein Multicast-Routing-Protokoll, das für das Senden an weit verteilte Gruppen optimiert ist und einen gemeinsamen Spannbaum für alle Sender verwendet.

CSMA/CD – Carrier Sense Multiple Access with Collision Detection
Nichtdeterministisches Medienzugriffsprotokoll in lokalen Netzen. Sendewillige Stationen überwachen das Übertragungsmedium, senden, sofern es nicht belegt ist, stoppen die Übertragung und senden ein Stausignal, falls ein weiterer Sender detektiert wird. Danach warten sie eine variable Zeitspanne bis zur Übertragungswiederholung.

DES – Data Encryption Standard
Blockverschlüsselungsalgorithmus des 1974 von IBM entwickelten Verschlüsselungssystems, das von 1977 bis 2000 von der US-Regierung als offizielles Datenchiffriersystem eingesetzt wurde. Mit diesem symmetrischen Verfahren werden Blöcke zu je 64 Bits mit einem gemeinsamen 56-Bit-Schlüssel verschlüsselt.

DLSR – delay since last sender report
Feld in jedem Block eines RTCP-Reports, in das ein Teilnehmer die Zeit einträgt, die seit dem Empfang des letzten, vom selben Sender erhaltenen Sender-Reports vergangen ist.

Differentiated Services
In der Internet Engineering Task Force (IETF) standardisiertes Verfahren der Vermittlungsschicht zur Erweiterung der Dienstqualität von IP-Netzen. Durch die Einteilung der Pakete in Dienstklassen kann für jede Klasse eine differenzierte Dienstqualität zur Verfügung gestellt werden.

GPS – Global Positioning System
Satellitenbasiertes System zur Bestimmung geographischer Positionen.

PS – Privater Schlüssel
In asymmetrischen Verschlüsselungsverfahren verwendeter, geheimzuhaltender Schlüssel.

ICMP – Internet Control Message Protocol
Protokoll zur Übertragung von Fehler- und Steuernachrichten der Vermittlungsschicht im Internet.

IEEE – Institute of Electrical and Electronic Engineers
Vereinigung von Ingenieuren der Elektrotechnik und Elektronik, die vom American National Standards Institute (ANSI) zur Herausgabe von technischen Standards anerkannt ist.

Integrated Services
Framework zur Erweiterung der Dienstqualität für Datenströme in IP-Netzen.

IS – Informationssender
Sender der verzögerungsfair zu verteilenden Informationen.

ISDN – Integrated Services Digital Network
Digitales Telekommunikationsnetzwerk, das verschiedene Dienste wie Sprach- und Datenübertragung integriert.

Jitter
Dienstqualitätsparameter, der die Unterschiede in der Übertragungszeit von Nachrichten beschreibt.

LAN – Local Area Network
Computernetzwerk von geringer geographischer Ausdehnung.

LSR – Last Sender Report
In jedem Block eines RTCP-Reports vorgesehenes Feld, in das ein Teilnehmer den Zeitpunkt einträgt, zu dem er vom selben Sender den letzten Sender-Report empfangen hat.

MAN – Metropolitan Area Network
Auf ein Stadtgebiet oder einen Ballungsraum beschränktes Netz, das hohe Übertragungsgeschwindigkeiten ermöglicht.

MANTRA – Monitor and Analysis of Traffic in Multicast Routers
Passives Monitor- und Analysewerkzeug für Multicast-Routing-Daten.

MBone – Multicast Backbone
Virtuelles Netz im Internet bestehend aus Routern und Teilnetzen, die Multicast-Funktionalität bieten.

MD5 – Message Digest Algorithm
Hashfunktion für kryptographische Verfahren, die einen beliebig langen Text auf einen 128-Bit-Wert abbilden.

MIB – Management Information Base
Definition der Verwaltungsobjekte, die ein Netzwerkgerät zur Verfügung stellt und auf die über →SNMP zugegriffen werden kann.

Markttransparenz
Bezeichnet die Verfügbarkeit aller Informationen über Güter, Preise und Konditionen, um die am Markt angebotenen Produkte untereinander vergleichen zu können. Markttransparenz stellt eine Voraussetzung für einen vollkommenen Markt dar.

Multicast
Die Übertragung einer Nachricht an eine Gruppe von Empfängern.

Multicast-Teilnehmer
Empfänger von Multicast-Nachrichten einer Multicast-Gruppe

NS-2 – Network Simulator
Ereignisgesteuerter Simulator für Netzwerkforschung mit umfangreicher Unterstützung von TCP/IP, Routing und Multicast-Protokollen über drahtgebundene und drahtlose Medien.

NTP – Network Time Protocol
Protokoll zur Uhrensynchronisation in Computernetzwerken.

ÖS – Öffentlicher Schlüssel
Einer der beiden Schlüssel eines asymmetrischen Verschlüsselungsverfahrens. Er wird Kommunikationspartnern zur Verschlüsselung von Nachrichten oder zur Überprüfung von Signaturen bekanntgegeben.

OTT – One Trip Time
Einwegzeit einer Nachricht.

PIM – Protocol Independent Multicast
Multicast-Routing-Protokoll, das sowohl für kleine, weit verteilte als auch für große, dicht besetzte Gruppen optimiert ist.

QoS – Quality-of-Service
Parameter zur Beschreibung der Dienstqualität der Informationsübertragung, wie z. B. Durchsatz, Verzögerung und Bitfehlerrate.

RISC – Reduced Instruction Set Computer
Prozessorarchitektur, deren Komplexität durch Reduktion des Befehlssatzes und Verzicht auf Mikrocode zugunsten einer schnelleren Befehlsausführung vermindert ist.

RSA – Rivest, Shamir und Adleman
Asymmetrisches Verschlüsselungsverfahren, benannt nach den Erfindern Ron Rivest, Adi Shamir, and Leonard Adleman.

RSVP – Resource Reservation Protocol
Empfängerorientiertes Verfahren zur Aushandlung von Ressourcenreservierungen für Datenströme.

RTCP – RTP Control Protocol
Steuer- und Feedbackprotokoll für →RTP.

RTO – Retransmission Timeout
Zeitvorgabe bis zur Übertragungswiederholung unbestätigter Segmente des Transportprotokolls TCP.

RTP – Real-Time Transport Protocol
Protokoll zur Übertragung multimedialer Datenströme.

RTT – Round Trip Time
Umlaufzeit einer Nachricht.

Routing
Durch die Netzwerkknoten vorgenommene Verkehrslenkung der Dateneinheiten auf Schicht-3 des ISO/OSI-Referenzmodells

SH – Sichere Hardware
Durch besondere Vorkehrungen vor Manipulationen geschützte Recheneinheit.

SHA-1 – Secure Hash Algorithm
Hashfunktion für kryptographische Verfahren, bildet einen Eingangstext auf einen 160-Bit-Wert ab.

SNMP – Simple Network Management Protocol
Protokoll zur Überwachung und Verwaltung von Rechnernetzen.

SOTT – Smoothed One Trip Time
Gleitender Durchschnitt der Paket-Einwegzeit.

SSV – Sitzungsschlüsselverwalter
Verwalter von Schlüsseln des symmetrischen Verschlüsselungsverfahrens bei der Schlüsselserver-Realisierung.

TCP/IP – Transfer Control Protocol/Internet Protocol
Verbindungsorientiertes Transportprotokoll und verbindungsloses Netzwerkprotokoll. TCP/IP wird häufig auch zur Bezeichnung des Protokoll-Stack des Internets verwendet.

TRMP – Timed Reliable Multicast Protocol
Token-basiertes Transportprotokoll für den Datenaustausch auf virtuellen Börsenplätzen.

User Datagram Protocol (UDP)
Verbindungsloses Transportprotokoll.

USB – Universal Serial Bus
Serielle Rechnerschnittstelle mit Datenübertragungsraten bis zu 450 Mbit/s.

Verzögerungsunterschied zwischen Multicast Teilnehmern
Laufzeitunterschied der Nachrichtenkopien einer Multicast-Nachricht.

Literaturverzeichnis

Almeroth, K.; Ammar, M. (1996). MBone Collection Tool, `http://www.cc.gatech.edu/computing/Networking/projects/mbone/`.

Bacher, D.; Swan, A.; Rowe, L. A. (1996). rtpmon: A Third-Party RTCP Monitor, *ACM Multimedia 96, Boston, MA, USA*.

Ballardie, T.; Francis, P.; Crowcroft, J. (1993). Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing, *Proceedings of SIGCOMM '93, San Francisco, CA*, ACM, pp. 85–95.

Berners-Lee, T. (1996). WWW: Past, Present, and Future, *IEEE Computer* **29**(10): 69–77.

Boggs, D. (1983). Internet broadcasting, *Technical Report CSL-83-3*, XEROX Palo Alto Research Center.

Borella, M. S. (2000). Methods and Protocols for Secure Key Negotiation Using IKE, *IEEE Network* **14**(4): 18–29.

Bormann, C. (1999). Providing Integrated Services over Low-bitrate Links, RFC 2689.

Byers, J.; Frumin, M.; Horn, G.; Luby, M.; Mitzenmacher, M.; Roetter, A.; Shaver, W. (2000). FLID-DL: Congestion control for layered multicast, *Proc. Second International Workshop on Networked Group Communication (NGC 2000)*, Palo Alto, CA, USA, pp. 71–81.

Calvert, K.; Griffioen, J. (2000). Internet Concast Service, Internet Draft, draft-calvert-concast-svc-00.txt.

Cerf, V. G.; Kahn, R. E. (1974). A Protocol for Packet Network Intercommunication, *IEEE Transactions on Communications* **22**(5): 637.

Chen, K.; Kutzko, M.; Rimovsky, T. (2002). Multicast Beacon Server v0.8.X (Perl), `http://dast.nlanr.net/Projects/Beacon/`.

Dolev, D.; Yao, A. (1983). On the security of public key protocols, *IEEE Transactions on Information Theory* **29**(2): 198 – 208.

Dongyan, X.; Baochun, L.; Nahrstedt, K.; Liu, J. W.-S. (1999). Providing Seamless QoS for Multimedia Multicast in Wireless Packet Networks, *SPIE-Int. Soc. Opt. Eng. Proceedings of Spie - the International Society for Optical Engineering*, Vol. 3528, pp. 352–61.

Estrin, D.; Farinacci, D.; Helmy, A.; Thaler, D.; Deering, S.; Handley, M.; Jacobson, V.; Liu, C.; Sharma, P.; Wei, L. (1997). Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, RFC 2117, `http://www.ietf.org/rfc/rfc2117.txt`.

Fenner, B. (1998). *mrouted 3.9-beta, mrinfo and other tools. Documentation.*

Ferrari, D. (1990). Client Requirements for Real-Time Communciation Services, *IEEE Communications Magazine* **28**(11): 65–72.

Feufel, D. (2001). *Entwurf eines skalierbaren, sicheren Uhrensynchronisationsmechanismus und dessen Implementierung für Java Cards*, Master's thesis, Universität Stuttgart, Fakultät Informatik.

Gartner, F. C.; Pagnia, H.; Vogt, H. (1999). Approaching a Formal Definition of Fairness in Electronic Commerce, *18th IEEE Symposium on Reliable Distributed Systems, Lausanne, Switzerland*, pp. 354–359.

Ge, Y.; Hou, J. C.; Tyan, H.-Y. (1999). A packet eligible time calculation mechanism for providing temporal QoS, *IEEE International Conference on Communications, Vancouver, BC, Canada*, Vol. 2, pp. 721–6.

GEMPLUS S.A. (2002). Cartel Project, `http://www.gemplus.com/smart/r_d/projects/cartel/index.htm`.

Grosser, M.; Starischka, S. (1998). *Das neue Konditionstraining für alle Sportarten für Kinder, Jugendliche und Aktive*, BLV Sportwissen, BLV Verlagsgesellschaft mbH, München, Wien, Zürich.

Guo, L.; Matta, I. (2001). The War between Mice and Elephants, *Proc. 9th IEEE International Conference on Network Protocols (ICNP'01) , Riverside, CA, USA*.

Haberman, B. K.; Rouskas, G. N. (1996). Cost, Delay, And Delay Variation Conscious Multicast Routing, *Technical Report TR-97-03*, Department of Computer Science, North Carolina State University, Raleigh, NC, USA.

Hagiwara, T.; Majima, H.; Matsuda, T.; Yamamoto, M. (2001). Impact of Round Trip Delay Self-Similarity on TCP Performance, *Proceedings - Tenth International Conference on Computer Communications and Networks (ICCCN 2001), Scottsdale, AZ, USA*, The Institute of Electrical and Electronics Engineers, Inc.

Hansmann, U.; Nicklous, M. S.; Schäck, T.; Seliger, F. (2000). *Smart Card Application Development Using Java*, Springer, Berlin, Heidelberg, New York, London, Paris, Tokyo.

Helbig, T. (1996). *Kommunikation und Synchronisation multimedialer Datenströme in verteilten Systemen*, PhD thesis, Fakultät Informatik der Universität Stuttgart.

Hewlett-Packard Company (2001). Network Node Manager Multicast 2.0 - Administrator's Guide, `http://h20229.www2.hp.com/products/mcast/index.html`.

IEEE (2002). 802.3 IEEE Standard Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, `http://standards.ieee.org/getieee802/download/802.3-2002.pdf`.

ITU (2002). Internet indicators: Hosts, Users and Number of PCs, `http://www.itu.int/ITU-D/ict/statistics/at_glance/Internet02.pdf`.

Johannes, P. (2000). MAOS Platforms - Technical Status Report, *Technical Report maos-report-10-04-2000*, Europay International.

Kalmanek, C. R.; Kanakia, H. (1990). Rate Controlled Servers for Very High-Speed Networks, *IEEE Global Telecomm. Conf., San Diego, CA, USA*, pp. 300.3.1–300.3.9.

Khare, R.; Jacobs, I. (1999). W3C Recommendations Reduce 'World Wide Wait', `http://www.w3.org/Protocols/NL-PerfNote.html`.

Klebelsberg, D. (1982). *Verkehrspsychologie*, Springer, Berlin, Heidelberg, New York.

Klein, B. (2000). incops - Einführung in die Kognitive Psychologie, `http://art2.ph-freiburg.de/incops/`.

Kleinrock, L. (1976). *Queuing Systems: Computer Applications*, Vol. II, John Wiley and Sons, New York, Chichester, Brisbane, Toronto.

Klöcking, J.-U.; Maihöfer, C.; Rothermel, K. (2001a). Reducing Multicast Inter-receiver Delay Jitter - A Server Based Approach, *in* P. Lorenz (ed.), *Proceedings of the International Conference on Networking (ICN 2001), Colmar, France*, Vol. 1 of *Lecture Notes in Computer Science 2093*, Springer, Berlin, Heidelberg, New York, London, Paris, Tokyo, pp. 498–507.

Klöcking, J.-U.; Maihöfer, C.; Rothermel, K. (2001b). A Smart Card Based Solution to Minimize Inter-receiver Delay Jitter, *in* J. Li; R. Luijten; E. K. Park (eds), *Proceedings - Tenth International Conference on Computer Communications and Networks (ICCCN 2001), Scottsdale, AZ, USA*, The Institute of Electrical and Electronics Engineers, Inc., pp. 96–101.

Klöcking, J.-U.; Rothermel, K. (2002). A Simple Method for Inter-receiver Delay Jitter Measurements in the MBone, *in* M. S. Obaidat; F. Davoli; I. Onyuksel; R. Bolla (eds), *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), San Diego (CA), USA*, The Society for Modeling and Simulation International, pp. 83–90.

Lee, C.; Nordstedt, D.; Helal, S. (2003). Enabling Smart Spaces with OSGi, *IEEE Pervasive Computing* **2**(3): 89–94.

Leiner, B. M.; Cerf, V. G.; Clark, D. D.; Kahn, R. E.; Kleinrock, L.; Lynch, D. C.; Postel, J. (1997). The Past and Future History of the Internet, *Communications of the ACM* **40**(2): 102–108.

Li, Q.; Mills, D. (2001). Jitter-Based Delay-Boundary Prediction of Wide-Area Networks, *IEEE/ACM Transactions on Networking* **9**(5): 578–590.

Linder, H.; Stering, W.; Norbury, J. (2002). IP Multicast Push and Broadcast on Demand in FRA Networks, IST-1999-11571 EMBRACE D14.

Maihöfer, C. (2001). *The Token Repository Service: A Universal and Scalable Mechanism for Constructing Multicast Acknowledgement Trees*, Springer, chapter 11.

Maihöfer, C.; Rothermel, K. (1999). Constructing Height-Balanced Multicast Acknowledgment Trees with the Token Repository Service, *Technical Report TR-1999-15*, Universität Stuttgart, Fakultät Informatik.

Makofske, D.; Almeroth, K. (1999). MHealth: A Real-Time Multicast Tree Visualization and Monitoring Tool, *Network and Operating System Support for Digital Audio and Video (NOSSDAV '99) , Basking Ridge New Jersey, USA*.

Margherio, L.; Henry, D.; Cooke, S.; Montes, S. (1998). The Emerging Digital Economy, `https://www.esa.doc.gov/Reports/EmergingDig.pdf`.

Maxemchuk, N. F.; Shur, D. H. (2001). An Internet Multicast System for the Stock Market, *ACM Transactions on Computer Systems* **19**(3): 384–412.

McCanne, S. (1996). *Scalable Compression and Transmission of Internet Multicast Video*, PhD thesis, University of California Berkeley.

Mills, D. (1995). Improved Algorithms for synchronizing Computer Network Clocks, *IEEE Transactions Networks* pp. 245–54.

Mills, D. L. (1992). Network Time Protocol (Version3) Specification, Implementation and Analysis, RFC 1305.

Moon, S. B.; Skelly, P.; Towsley, D. (1999). Estimation and Removal of Clock Skew from Network Delay Measurements, *Proceedings of IEEE INFOCOM'99, New York, NY, USA*.

Moore, S.; Anderson, R.; Cunningham, P.; Mullins, R.; Taylor, G. (2002). Improving Smart Card Security using Self-timed Circuits, *The Eighth IEEE International Symposium on Asynchronous Circuits and Systems (async 2002), Manchester, UK*, pp. 193–200.

OSGi (2000). OSGi Provides Open Platform for the Internet-Enabled Car, Press release, `http://www.osgi.org/documents/news_events/press_releases/pressrel1016900.pdf`.

Paxson, V. (1997a). End-to-End Routing Behavior in the Internet, *IEEE/ACM Transactions on Networking* **5**(5): 601–615.

Paxson, V. (1997b). *Measurements and Analysis of End-to-End Internet Dynamics*, PhD thesis, University of California, Berkeley.

Perlman, R. (1999). An Overview of PKI Trust Models, *IEEE Network* **13**(6): 38–43.

Pulido, J.-M.; Lin, K.-J. (1998). SM: Real-Time Multicast Protocols for Simultaneous Message Delivery, *5th International Conference on Real-Time Computing Systems and Applications, Hiroshima, Japan*.

Rajvaidya, P.; Almeroth, K. (2000). A scalable architecture for monitoring and visualizing multicast statistics, *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM), Austin, Texas, USA*.

Rao, N. S. (2002). Probabilistic Guarantees on Message Delays Over Wide-Area Networks Using In-Situ Instruments, *Proceedings of the Eleventh International Conference on Computer Communications and Networks (ICCCN 2002), Miami, FL, USA*, IEEE.

Rhee, I.; Ozdemir, V.; Yi, Y. (2000). TEAR: TCP emulation at receivers - flow control for multimedia streaming, *Technical report*, Department of Computer Science North Carolina State University.

Riebeek, H. (2003). Brazil Tests World's Largest Environmental monitoring System, *IEEE Spectrum* .

Rothermel, K.; Maihöfer, C. (1999). A robust and efficient mechanism for constructing multicast acknowledgment trees, *Proceedings of the Eight International Conference on Computer Communications and Networks (IEEE ICCCN)*, pp. 139–45.

Rouskas, G.; Baldine, I. (1997). Multicast Routing with End-To-End Delay and Delay Variation Constraints, *IEEE JSAC* **15**: 346–56.

Saltzer, J.; Reed, D.; Clark, D. (1981). End-to-End Arguments in System Design, *Second International Conference on Distributed Computing Systems, Paris, France*, IEEE, pp. 509–512.

Sarac, K.; Almeroth, K. (2001). Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring, *Journal of High Speed Networking, Special Issue on Management of Multimedia Networking* .

Schneier, B. (1996). *Applied Cryptography*, John Wiley and Sons, New York, Chichester, Brisbane, Toronto.

Schulzrinne, H.; Casner, B.; Frederick, R.; Jacobson, V. (1996). RTP: A Transport Protocol for Real-Time Applications, RFC 1889.

Shannon, C. E. (1976). Die mathematische Theorie der Kommunikation, *in* C. E. Shannon; W. Weaver (eds), *Mathematische Grundlagen der Informationstheorie*, Oldenbourg, München, pp. 41–130.

Steinmetz, R. (1996). Human Perception of Jitter and Media Synchronization, *IEEE Journal on Selected Areas in Communications* **14**(1): 61–72.

Sun, Q.; Langendoerfer, H. (1999). Computation of constrained multicast trees using a genetic algorithm, *European Transactions on Telecommunications & Related Technologies* **10**(5): 513–16.

Theilmann, W. (2000). *Themenspezifische Informationssuche im Internet mit Hilfe mobiler Programme*, PhD thesis, Universität Stuttgart.

Varta AG (2004). VARTA CardPower for next Generation Smart Cards, `http://www.varta-microbattery.com/en/mb_data/documents/printing_material_oem/LEAFLET_CardPower_en.pdf`.

Vicisano, L.; Crowcroft, J.; Rizzo, L. (1998). TCP-like congestion control for layered multicast data transfer, *Proceedings of IEEE INFOCOM*, Vol. 3, pp. 996–1003.

Waitzman, D.; Partridge, C.; Deering, S. (1988). Distance Vector Multicast Routing Protocol, RFC 1075.

Weineck, J. (1994). *Optimales Training - leistungsphysiologische Trainingslehre unter besonderer Berücksichtigung des Kinder- und Jugendtrainings*, PERIMED-spitta, Med. Verl.-Ges.

Weiss, H. (2000). Olympischer Sport in der Informationsgesellschaft: Der Fairnessgedanke, `http://www.nok.de/projekt/faecher/religion/fairness/`.

Wensheng, S.; Zemin, L. (1999). Routing multipoint connections in packet-switched computer networks, *High Technology Letters, China* **5**(1): 21–4.

Werkmann, H.; Schwarze, S. (2002). BSH startet Feldtest mit vernetzten smartHome Hausgeräten, Pressemitteilung, `http://www.osgi.org/documents/news_events/member_press/BSH_ProSyst_NetAtHome_engl.131102.pdf`.

Widmer, J. C. (2003). *Equation-Based Congestion Control for Unicast and Multicast Data Streams*, PhD thesis, Fakultät für Mathematik und Informatik, Universität Mannheim.

ZDNet UK News (2001). BTopenworld throttles P2P applications, `http://www.zdnet.co.uk/search/index.htm?c=news&q=BTopenworld+throttles+P2P+applications`.

Zhang, H. (1995). Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines, *Computer Communications: Special Issue on System Support for Multimedia Computing* **10**(18).