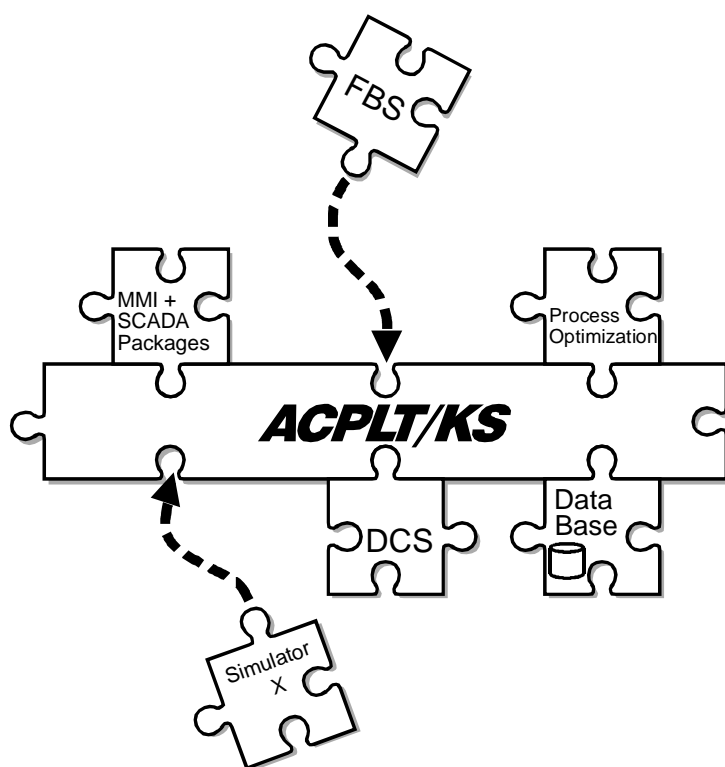


ACPLT/KS

Technologiepapier Nr. 2: Der Nachrichtentransport



Inhalt

1 Einleitung	3
2 Das Schichtenmodell	3
3 Remote Procedure Calls (RPC)	3
4 External Data Representation (XDR)	4
5 Das Transportprotokoll	5
5.1 UDP	5
5.2 TCP	6
6 AuA – Abkürzungen und Akronyme	7
7 Literaturverzeichnis	7

1 Einleitung

Das Kommunikationssystem ACPLT/KS (im folgenden auch „KS“ genannt) verbindet im Bereich von Anwendungen der Prozeß- und Betriebsführung rechnergestützte Werkzeuge untereinander und mit Prozeßleitsystemen. Dieses Technologiepapier beschreibt die Netzwerkschichten, auf denen ACPLT/KS für den Nachrichtentransport aufsetzt. Insbesondere die beiden Internetprotokolle TCP/IP und UDP/IP werden im folgenden hinsichtlich ihrer Eignung für die Übermittlung der KS-Nachrichtentelegramme untersucht.

2 Das Schichtenmodell

In einem Netzwerk kommunizieren die einzelnen Rechner miteinander mittels zuvor vereinbarter sogenannter „Protokolle“. Diese Protokolle definieren – wie im richtigen Leben – die Regeln und Konventionen der Kommunikationspartner. Aufgrund ihrer teilweise umfangreichen Komplexität werden Protokolle in „Schichten“ aufgeteilt, was insbesondere einer leichter zu handhabenden Implementierung zugutekommt.

Zwischen 1977 und 1984 [5] wurde von der ISO (der International Standards Organization) das OSI-Schichtenmodell entwickelt (OSI steht für „Open Systems Interconnect“). Es ist jedoch nicht als Spezifikation, sondern vielmehr als Strukturmodell anzusehen. Das OSI-Schichtenmodell besteht aus insgesamt sieben Schichten, die in Abbildung 2.1 dargestellt sind.

Schicht 7	Application	ACPLT/KS-Anwendung mit KS-Bibliothek sowie RPC-Bibliothek (Remote Procedure Calling)
6	Presentation	XDR (eXternal Data Representation)
5	Session	(nicht verwendet)
4	Transport	TCP (Transmission Control Protocol)
3	Network	IP (Internet Protocol)
2	Data Link	Beispielsweise Ethernet IEEE 802.2
Schicht 1	Physical	Ethernet IEEE 802.3

Abbildung 2.1: Das OSI-Schichtenmodell mit den bei ACPLT/KS für den Nachrichtentransport wichtigen Schichten.

Ein Kennzeichen des Schichtenmodells sind die wohldefinierten Schnittstellen zwischen den Schichten. Dadurch soll erreicht werden, daß Änderungen innerhalb einer Schicht die angrenzenden Schichten überhaupt nicht oder zumindest möglichst wenig beeinflussen.

Nachfolgend werden nur diejenigen Schichten des OSI-Modells behandelt, die ACPLT/KS benutzt: Die Applikations- und die Präsentationsschicht, sowie die Transport- und die Netzwerkschicht. ACPLT/KS benötigt jedoch keine Session-Schicht. Die beiden untersten Schichten 1 und 2 hängen schließlich von der jeweils verwendeten Netzwerkarchitektur ab und sind hier nicht weiter von Interesse.

3 Remote Procedure Calls (RPC)

Ein „Remote Procedure Call“ – oder kurz RPC – ermöglicht es einem Programm (dem Klienten), eine Prozedur oder Funktion aufzurufen, die durch ein anderes Programm (dem Server) auf einem anderen Computer im Netzwerk bereitgestellt wird. Eine RPC-Bibliothek besorgt dabei das Ver- und Entpacken von Aufrufparametern und Antwort der Funktion oder Prozedur. Daneben kümmert sich die Bibliothek noch um den Versand dieser Informationen über das

Netzwerk (Abbildung 3.1). RPC-Bibliotheken sind in verschiedenen Implementationen beispielsweise für Unix, VMS und Windows 95/NT (zum Teil kostenlos) verfügbar. Diese Implementationen lassen sich zwei verschiedenen Stammbäumen zuordnen: auf der einen Seite ONC (Open Network Computing) RPC der Firma Sun und auf der anderen Seite das DCE (Distributed Computing Environment) RPC der Open Software Foundation (OSF). Die weitaus größte Verbreitung findet dabei die ONC/RPC-Bibliothek von Sun, da deren Quellcode frei verfügbar ist und beispielsweise das Network File System (NFS) auf ONC/RPC aufsetzt.

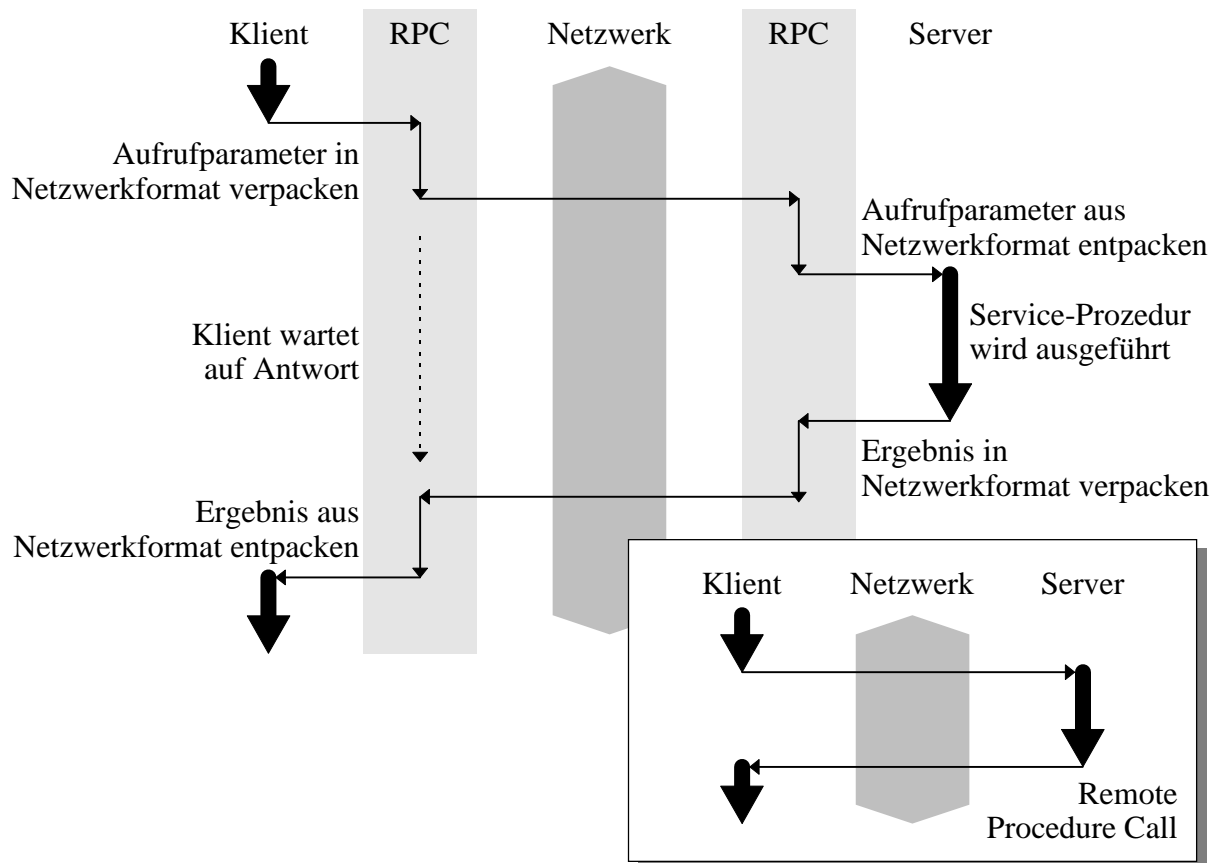


Abbildung 3.1: Mittels eines Remote Procedure Call kann ein Klient eine Funktion oder Prozedur im Server ausführen.

ACPLT/KS benutzt ONC/RPC, um damit den Nachrichtenaustausch auf Hochsprachenebene zu realisieren. Im Rahmen des EMS-Projektes [1] wurde am Lehrstuhl für Prozeßleittechnik ein solches Kommunikationskonzept untersucht und die prinzipielle Realisierbarkeit gezeigt.

4 External Data Representation (XDR)

Damit Klient und Server trotz einer möglicherweise unterschiedlichen internen Repräsentation insbesondere von Zahlen trotzdem Daten miteinander austauschen können, arbeitet ONC/RPC mit der sogenannten „External Data Representation“ XDR auf Schicht 6 (Presentation Layer). XDR geht für seine standardisierte Datendarstellung von einer kanonischen (gemeinsamen einheitlichen) Repräsentation der zu übertragenden Informationen aus. Es existiert hierbei nur eine einzige einheitliche Byteanordnung (bei XDR ist dieses die „Big Endian“-Anordnung), eine einheitliche Darstellung von Gleitkommazahlen gemäß IEEE, und so weiter.

Der Vorteil dieses kanonischen Ansatzes zeigt sich dann, wenn RPC und XDR auf eine neue Plattform portiert werden sollen. In diesem Fall sind keine Änderungen an den bisherigen XDR-Implementationen auf den bereits vorhandenen Plattformen erforderlich. Nur auf der neu hinzugekommenen Plattform muß die XDR-Bibliothek an die lokale Datendarstellung angepaßt werden. Dieser kanonische Ansatz hat sich bereits bei den Netzwerkprotokollen TCP/IP,

UDP/IP, ICMP und dem Ethernet bestens bewährt. Der Zeitbedarf für die Umwandlung in die oder aus der XDR-Netzwerkdarstellung ist zudem fast immer gegenüber dem Zeitbedarf zu vernachlässigen, den die Daten benötigen, um die unteren Schichten 5 bis 1 zu passieren, über das Netzwerk zu wandern und beim Empfänger schließlich wieder bis zur XDR-Schicht 6 vorzudringen.

5 Das Transportprotokoll

Verfügbare RPC-Implementationen verwenden als Transportschicht in der Regel entweder das verbindungslose UDP (User Datagram Protocol) oder aber das verbindungsorientierte TCP (Transmission Control Protocol). Daneben existiert noch ein „Transport Independent RPC“ (TIRPC), das auf dem „Transport Layer Interface“ von UNIX System V, Release 3 und 4 aufbaut und damit nur dort verfügbar ist.

5.1 UDP

Das „User Datagram Protocol“ (oder kurz UDP) ist ein verbindungsloser Dienst. Sobald die UDP-Transportschicht von einer der höheren Schichten 5–7 einen Datenblock erhält, wird dieser mit einer Prüfsumme versehen an die darunterliegende IP-Schicht weitergeleitet. Die IP-Schicht portioniert den Datenblock in passende IP-Häppchen, die sie zwecks Versand zum Empfänger an die Schicht 2 (Data Link) weiterverfüttet. Für die senderseitige UDP-Schicht ist der Vorgang damit abgeschlossen – sie interessiert nicht, ob das Datenpaket jemals den Empfänger erreicht.

Netzwerke sind für die IP-Datenpakete ein gefährliches Pflaster. Die IP-Datenpakete können auf ihrem Weg durch das Netzwerk verlorengehen, in Routern fragmentiert oder dupliziert werden und schließlich den Empfänger sogar (über verschiedene Wege) in vertauschter Reihenfolge erreichen. Die IP-Schicht muß nun die eintreffenden IP-Pakete wieder zu dem ursprünglichen Datenblock zusammensetzen, der von der Schicht 4 aus beim Sender auf die Reise ging. Treffen alle zum originalen Datenblock gehörenden IP-Pakete nicht innerhalb einer bestimmten Zeitspanne ein, so werden die bis dahin empfangenen IP-Pakete ohne weitere Maßnahmen verworfen. Ansonsten wird mit dem Eintreffen aller IP-Pakete der komplette Datenblock der UDP-Schicht überreicht. Diese stellt ihn nach Kontrolle der Prüfsumme den höheren Schichten 5–7 zur Verfügung.

Die maximale Größe eines per UDP zu versendenden Datenblocks ist prinzipiell auf 64 kByte begrenzt, kann jedoch je nach IP-Implementation auch deutlich kleiner sein. Als Faustwert sind in der Praxis 8 kByte anzusehen [2, 4]. Garantiert werden in der Protokollspezifikation [3] jedoch nur 548 Byte (unter der Voraussetzung, daß keine IP-Optionen mit im IP-Paket transportiert werden müssen). ACPLT/KS muß jedoch insbesondere bei Archivabfragen oder dem Browsen durch den Objektbaum eines KS-Servers mehr als 8 kByte (oder 64 kByte) lange Antworten transportieren können.

Ein weiteres Problem bei UDP stellt der unquittierte Versand der Datenblöcke dar. Erhält ein RPC-Klient innerhalb einer bestimmten Zeit keine Antwort vom RPC-Server, so weiß der Klient nicht, ob seine Anfrage überhaupt beim Server angekommen ist, oder ob dessen Antwort verlorengegangen. Bei der ONC/RPC-Implementation von Sun verschickt daher der Klient standardmäßig alle fünf Sekunden eine neue Anforderung an den RPC-Server und bricht ohne eine Server-Antwort nach spätestens 25 Sekunden ab. Überträgt nun das Netz eine Server-Antwort erst verzögert, so kann es passieren, daß der Klient bereits die RPC-Anfrage wiederholt gesendet hat. Um zu verhindern, daß die gewünschte Funktion im Server irrtümlich mehrfach aufgerufen wird, muß der Server daher in einem Cache die bereits bearbeiteten Anfragen speichern.

5.2 TCP

Das „Transmission Control Protocol“ (TCP) ist ein verbindungsorientierter Dienst, der einen gesicherten Vollduplex-Datenstrom zur Verfügung stellt. Die beiden Kommunikationspartner eröffnen dabei je einen Kommunikationsendpunkt („socket“), in den hinein – beziehungsweise aus dem heraus – sie ohne Beachtung von Blockgrenzen Daten schreiben oder lesen können. Die TCP-Transportschicht kümmert sich dann darum, daß die Daten in der richtigen Reihenfolge beim Empfänger ankommen und verlorengegangene IP-Pakete erneut gesendet werden. Durch die stromartige Verbindung kennt TCP im Gegensatz zu UDP keine obere Grenze für die Datenblockelänge, außerdem kann ein vom Sender übertragener Datenblock beim Empfänger auch in mehreren Blöcken von der Transportschicht abgeholt werden.

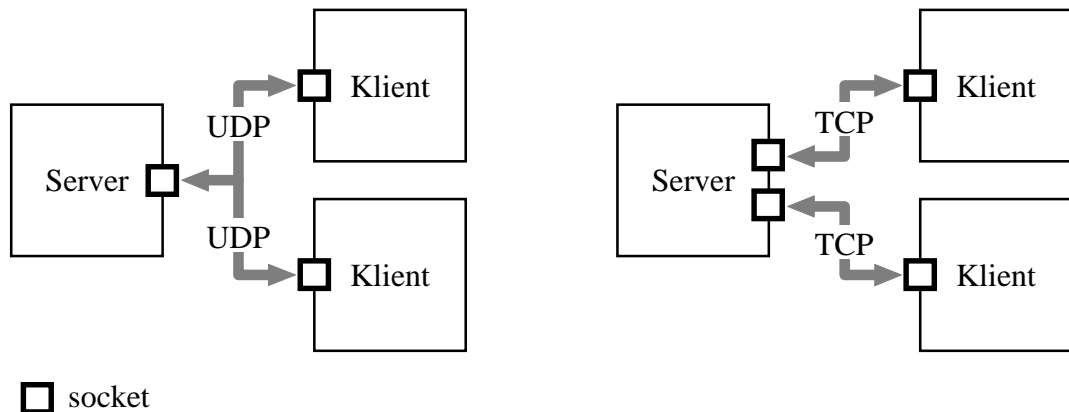


Abbildung 5.1: Vergleich eines UDP- und eines TCP-basierten Servers, der jeweils mehrere Klienten bedient.

Der gesicherte Datentransport via TCP erfordert, daß beide Kommunikationsteilnehmer Statusinformationen über jede offene Verbindung führen. Ein Server muß daher bei Einsatz von TCP/IP für jeden Klienten eine gesonderte Verbindung verwalten, wohingegen er bei UDP/IP nur einen Kommunikationsendpunkt benötigt (siehe Abbildung 5.1). Dieser erhöhte Ressourcenbedarf für TCP/IP-Verbindungen wird jedoch durch den Vorteil einer gesicherten Datenübertragung und das Fehlen von Blockgrößenbeschränkungen kompensiert. Aus diesen Gründen verwendet ACPLT/KS als Transportprotokoll TCP/IP für die Kommunikation zwischen KS-Servern und KS-Klienten.

6 AuA – Abkürzungen und Akronyme

ACPLT/KS	Kommunikationssystem des Lehrstuhls für Prozeßleittechnik der RWTH Aachen
IP	Internet Protocol
ISO	International Standards Organization
ONC	Open Network Computing
OSI	Open Systems Interconnect
RPC	Remote Procedure Call/Calling
TCP	Transmission Control Protocol
TIRPC	Transport Independent Remote Procedure Calling
UDP	User Datagram Protocol
XDR	External Data Representation

7 Literaturverzeichnis

- [1] Arnold, Matthias:
Werkzeugintegration in verteilten Systemen der Prozeßleittechnik.
Dissertation (in Vorbereitung), Lehrstuhl für Prozeßleittechnik, RWTH Aachen.
- [2] Bloomer, John:
Power Programming with RPC.
O'Reilly & Associates, Inc., Sebastopol, CA, 1992
- [3] Braden, R.:
RFC 1122: Requirements for Internet Hosts – Communication Layers.
<http://www.internic.net/rfc/rfc1122.txt>, 1989
- [4] N.N.:
Programming with ONC RPC.
Digital Equipment, http://sawyer.wustl.edu/du4-docs/AA-Q0R5B-TET1_html/TOC.html, 1996
- [5] Stevens, Richard W.:
UNIX Network Programming.
Prentice Hall, 1990.