

5 Interprozesskommunikation

5.1 Überblick

5.2 Netzwerke

5.3 Netzwerkgrundlagen

5.4 IPC-Semantiken

5.5 IPC und Fernaufrufe



- Netzwerke – Aspekte für Verteilte Systeme
- Netzwerkgrundlagen
 - Paketvermittlung
 - Daten-Streaming
 - Vermittlungsschemata
 - Protokolle
 - Routing
 - Internetworking
 - Internet-Protokolle
- IPC-Semantiken
- IPC und Fernaufrufe



Netzwerke sind die Grundlage für Verteilte Systeme.

Ihre Eigenschaften und Fähigkeiten bestimmen wesentlich, wie ein Verteiltes System beschaffen ist (sein kann).

■ Leistung

- Interaktion in Verteilten Systemen basiert im Wesentlichen auf Nachrichtenaustausch

- welche Parameter beeinflussen die Geschwindigkeit von Nachrichtenzustellungen?

 - Latenz

 - Zeit von der Sende-Operation bis erste Daten am Ziel ankommen

 - Transferrate (Bits pro Sekunde)

=> $\text{Nachrichtenübertragungszeit} = \text{Latenz} + \text{Nachrichtenlänge} / \text{Transferrate}$

■ Skalierbarkeit



- Zuverlässigkeit
 - wesentliche Grundlage für Strategien in verteilten Anwendungen
 - ist der Fehlerfall die seltene Ausnahme oder der zu erwartende Normalfall?
- Sicherheit
 - worauf kann sich die Anwendung verlassen, was muss sie selbst absichern?
- Mobilität
 - mobile Geräte sind eine stark wachsende Komponente in Verteilten Systemen
 - Folge: wechselnde Erreichbarkeit verbunden mit Schwankungen bei Leistung, Zuverlässigkeit, ...
 - was kann das transparent für die Anwendung machen, was ist inhärent sichtbar?



- Dienstgüte (Quality of Service)
 - erlaubt das Netzwerk die Formulierung von QoS-Anforderungen?
 - wie kann eine Anwendung ihren Bedarf ermitteln und ausdrücken?
- Multicast
 - Gruppenkommunikation ist die Basis für viele Mechanismen in Verteilten Systemen (-> Fehlertoleranz)
 - welche Mechanismen für Broadcast oder Multicast unterstützt bereits das Netzwerk?



Grundlegende Technik in Rechnernetzen ist *Paketvermittlung*. Kommunikation erfolgt *asynchron* – d. h. Nachrichten erreichen ihr Ziel nach einer (variierenden) Verzögerungszeit

■ Paketvermittlung

- Nachrichten werden in Datenpakete begrenzter Länge zerlegt
- Pakete werden mit Adressinformationen versehen und über das Netz übertragen

■ Daten-Streaming

- nicht alle Anwendungen in Verteilten Systemen basieren auf dem Austausch von Nachrichten
- wichtigste Ausnahmen: Audio- und Video-Daten
- spezielle QoS-Anforderungen: Durchsatz, Jitter



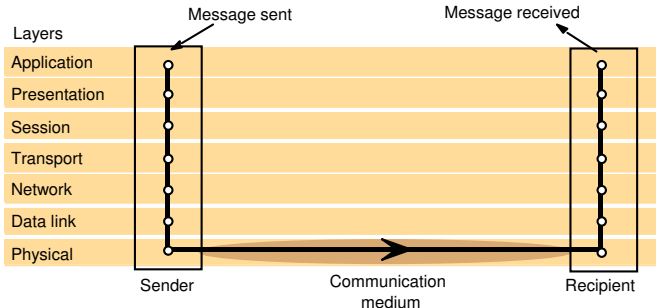
- Vermittlungsschemata
 - Broadcast
 - Leitungsvermittlung (Circuit Switching)
 - Paketvermittlung
 - Frame Relay



Netzwerkgrundlagen (3)

■ Protokolle

- Regeln und Formate für den Austausch von Nachrichten
- Protokollschichten
 - Aufteilung der verschiedenen Aufgaben, die bei einer Nachrichtenübertragung anfallen
 - Schichten können aus Effizienzgründen in der Implementierung verschmolzen werden



Instructor's Guide for Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design Edn. 5
© Pearson Education 2012



- ... Protokolle
 - Protokollstack
 - Vollständige Menge von Protokollschichten
 - Beispiel: CORBA - TCP/IP - Ethernet
 - Paketzusammenstellung
 - Paketgrößen in verschiedenen Protokollschichten / Teilnetzen ggf. unterschiedlich
 - Pakete einer Nachricht können zerteilt und verschmolzen werden
 - Adressierung
 - erfolgt auf verschiedenen Protokollschichten unterschiedlich: Anwendungs-spezifische Adressen, Prozesse, Rechner, Vermittlungsknoten
 - Abbildungsmechanismen innerhalb des Protokollstacks (z. B. IP-Adresse -> Ethernet-Adresse)
 - Paketzustellung
 - Datagramme – analog zu *Telegrammen*, jedes Datagramm enthält alle notwendigen Adressdaten und wird für sich eigenständig zugestellt
 - Virtuelle Verbindungen – analog zu *Telefonverbindungen* zunächst ein Verbindungsaufbau, die Datenpakete werden dann über die Verbindung übertragen.



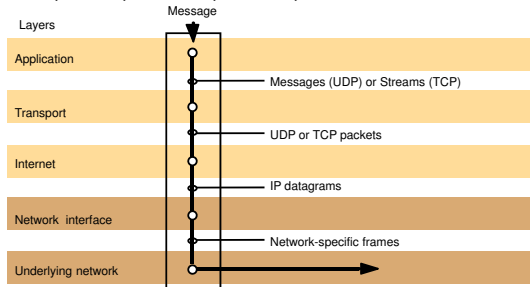
- Routing
 - Zustellung eines Datenpakets durch Übertragung über mehrere Zwischen-/Vermittlungsrechner
 - grundlegendes Konzept in allen Netzen (außer sehr einfachen lokalen Netzen)
 - spezielle Protokolle legen die Übertragungswege fest

- Internetworking
 - Aufbau eines Netzwerks durch den Zusammenschluss vieler Netzwerke
 - lokale Netze, Funknetze, Weitverkehrsnetze
 - unterschiedliche Basistechnologien, unterschiedliche Vermittlungsknoten und -Technologien (Router, Switches, Bridges, Hubs, Tunnel, ...)



Netzwerkgrundlagen (6)

- Internet-Protokolle
 - Basis des heutigen Internet
 - erste Entwicklungen in den 1970er Jahren im ARPAnet (NCP - *Network Control Protocol*)
 - Entwicklung der heutigen Kommunikationsprotokolle ab 1980 (TCP/IP - *Transmission Control Protocol / Internet Protocol*)
 - zunächst einfache Anwendungsprotokolle für Dateitransfer (FTP), Rechnerzugang (rlogin), Mail (SMTP), ...



Instructor's Guide for Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design Edn. 5
© Pearson Education 2012



Grundkonzepte für die Kommunikation zwischen Prozessen durch Nachrichtenaustausch

- **no-wait send** der Sendeprozess wartet, bis die Nachricht im Transportsystem zum Absenden bereitgestellt worden ist
 - *Pufferung* oder *Signalisierung* (dass der übergebene Puffer wieder frei ist)
- **synchronization send** der Sendeprozess wartet, bis die Nachricht vom Empfangsprozess angenommen worden ist
 - *Rendezvous* zwischen Sende- und Empfangsprozess
- **remote-invocation send** der Sendeprozess wartet, bis die Nachricht vom Empfangsprozess verarbeitet und beantwortet worden ist
 - *Fernaufruf* einer vom Empfangsprozess auszuführenden Funktion



- IPC bildet die Grundlage für den Austausch von Nachrichten zwischen Prozessen
- die Nutzung der TCP/IP-Protokolle bietet zwei Alternativen
 - Verpacken von Nachrichten in UDP-Paketen
 - Ein Paket pro Nachricht
 - Nachrichtenzustellung ist zunächst genauso unzuverlässig wie UDP
 - Versenden von Nachrichten über TCP-Verbindungen
 - Nachrichten müssen im Datenstrom der Verbindung kodiert werden (Anfang und Ende kenntlich machen)
 - Nachrichtenübertragung ist zuverlässig – *so lange die Verbindung steht!*



IPC und Fernaufrufe (2)

- IPC bildet die Basis, in der „darüberliegenden“ Ebene steht die *Bedeutung* der Nachrichten im Mittelpunkt
 - die Bedeutung kann sich *implizit* durch den Verarbeitungsalgorithmus (das Programm) ergeben oder
 - die Prozesse machen sie sich gegenseitig *explizit* über „Anweisungen“ bekannt
- die Nachrichten enthalten (problemspezifische) Daten und/oder Code:
 - function shipping* der Empfangsprozess interpretiert Programme
 - mobiler Code (Java Bytecode, PostScript) ggf. mit Daten unterfüttert
 - data shipping* der Empfangsprozess interpretiert Daten
- im „Normalfall“ bewirken Nachrichten die Ausführung entfernter Routinen
 - die aufzurufenden Prozeduren/Funktionen sind implizit oder explizit kodiert



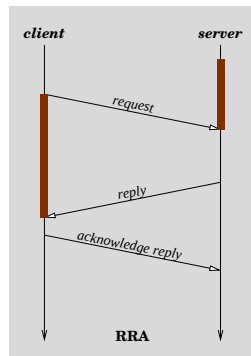
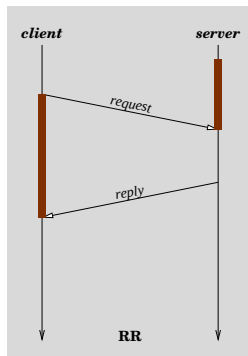
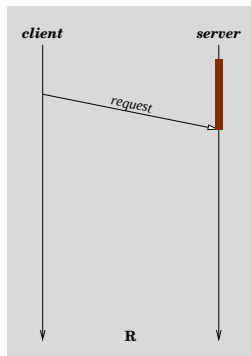
request (R) kann genutzt werden, wenn die entfernte Prozedur/Funktion keinen Rückgabewert liefert und der Sendeprozess keine Bestätigung für die erfolgte Ausführung benötigt 1 Nachricht

request-reply (RR) ist das geläufige Verfahren, da die Antwortnachricht implizit die Anforderungsnachricht bestätigt und dadurch explizite Bestätigungen entfallen 2 Nachrichten

request-reply-acknowledge reply (RRA) gestattet es, die zum Zwecke der *Fehlermaskierung* (beim Server) gespeicherten Antwortnachrichten zu verwerfen, wenn (vom Client) keine weitere Anforderungsnachricht gesendet wird 3 Nachrichten



IPC-Protokolle für Fernaufrufe (2)



- Nachrichtenaustausch unterliegt bestimmten (typischen) *Fehlerannahmen*:
 1. Nachrichten können verloren gehen
 - beim Sender, beim Empfänger oder im Netz
 2. es können Netzwerk-Partitionierungen auftreten
 - ein oder mehrere Rechner (Knoten) werden „abgetrennt“
 3. Prozesse können scheitern (d.h. „abstürzen“)
 - Prozess-, Rechner- oder Netzausfälle sind nicht unterscheidbar
 4. Daten können verfälscht werden
- als Folge sind unterschiedliche (typische) *Protokollvarianten* entstanden



- Anforderungs- und ggf. auch Antwortnachrichten wiederholen
 - nach einer Pause (*time-out*) werden die Nachrichten erneut versendet
 - die „optimale“ Länge der Pause zu bestimmen ist äußerst schwierig
- eingetroffene Nachrichtenduplikate sind zu erkennen und zu ignorieren
 - ggf. bereits versandte Antwortnachrichten wiederholt versenden
 - auf Client- bzw. Server-Seite ist ggf. „*duplicate supression*“ anzuwenden
- *idempotente Operationen*/Zustandsfreiheit tolerieren
Anforderungsduplikate



Fehlermaskierung (2)

