

Communication

Pertemuan 4

Overview

- ▶ Interprocess communication atau komunikasi antar proses adalah inti dari sistem terdistribusi dan komunikasi antar proses-proses pada system-sistem komputer yang berbeda dapat terjadi jika sistem-sistem tersebut terhubung dalam suatu jaringan.
- ▶ Komunikasi antar komputer diatur oleh aturan yang disebut protokol komunikasi.
- ▶ Beragamnya pembuat perangkat keras dan aplikasi komunikasi menghasilkan produk-produk yang tidak saling kompatibel sehingga mempersulit upaya komunikasi.

Dasar-dasar Komunikasi Jaringan

1.1 Layered Protocols

- Interprocess communication atau komunikasi antar proses adalah inti dari sistem terdistribusi dan komunikasi antar proses-proses pada system-sistem komputer yang berbeda dapat terjadi jika sistem-sistem tersebut terhubung dalam suatu jaringan. Komunikasi antar komputer diatur oleh aturan yang disebut protocol komunikasi.
- Beragamnya pembuat perangkat keras dan aplikasi komunikasi menghasilkan produk-produk yang tidak saling kompatibel sehingga mempersulit upaya komunikasi.
- Sebuah standar protokol kemudian dibuat oleh ISO (*International Standard Organization*) untuk mengatasi hal tersebut. Standar tersebut dikenal sebagai OSI (*Open System Interconnection*) model dengan aritektur sebagai berikut.



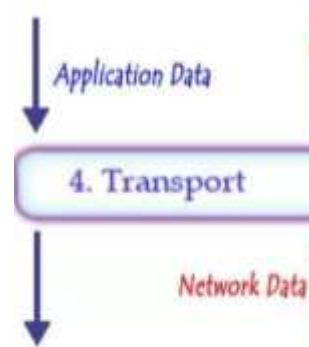
Gambar 3.1 OSI Layer

- ▶ Arsitektur OSI dibuat berlapis–lapis dengan fungsi yang berbeda pada setiap lapisannya. Lapisan yang lebih tinggi menyembunyikan kerumitan dari operasi di lapisan yang lebih rendah dan suatu lapisan hanya dapat diakses oleh lapisan yang ada di atas atau di bawahnya.
- ▶ Hal tersebut dimaksudkan untuk memberi kemudahan kepada para pembuat perangkat keras dan perangkat lunak komunikasi dalam mengembangkan berbagai protokol yang berbeda sesuai kebutuhan namun dapat saling kompatibel dengan syarat produk mereka mengikuti standar yang telah ditentukan dalam OSI.
- ▶ Lapisan–lapisan (Layer) protokol tersebut dapat digolongkan lebih jauh menjadi :
 1. Low –level Layers
 2. Transport Layer
 3. Application Layer
 4. Middleware Layer

1.2 Low-level Layers

Yang termasuk low-level Layer adalah lapisan-lapisan sebagai berikut:

1. **Physical Layer** : berisi spesifikasi dan implementasi dari bit-bit dan proses transmisi dari pengirim ke penerima.
2. **Data link Layer** : bertanggung jawab mengurus pengubahan bit-bit data menjadi frame untuk mengatasi error dan mengatur arus pengiriman frame (*flow control*).
3. **Network Layer** : mengatur bagaimana paket-paket diarahkan agar mencapai tujuannya berdasarkan alamat logik.




1.3 Transport Layer

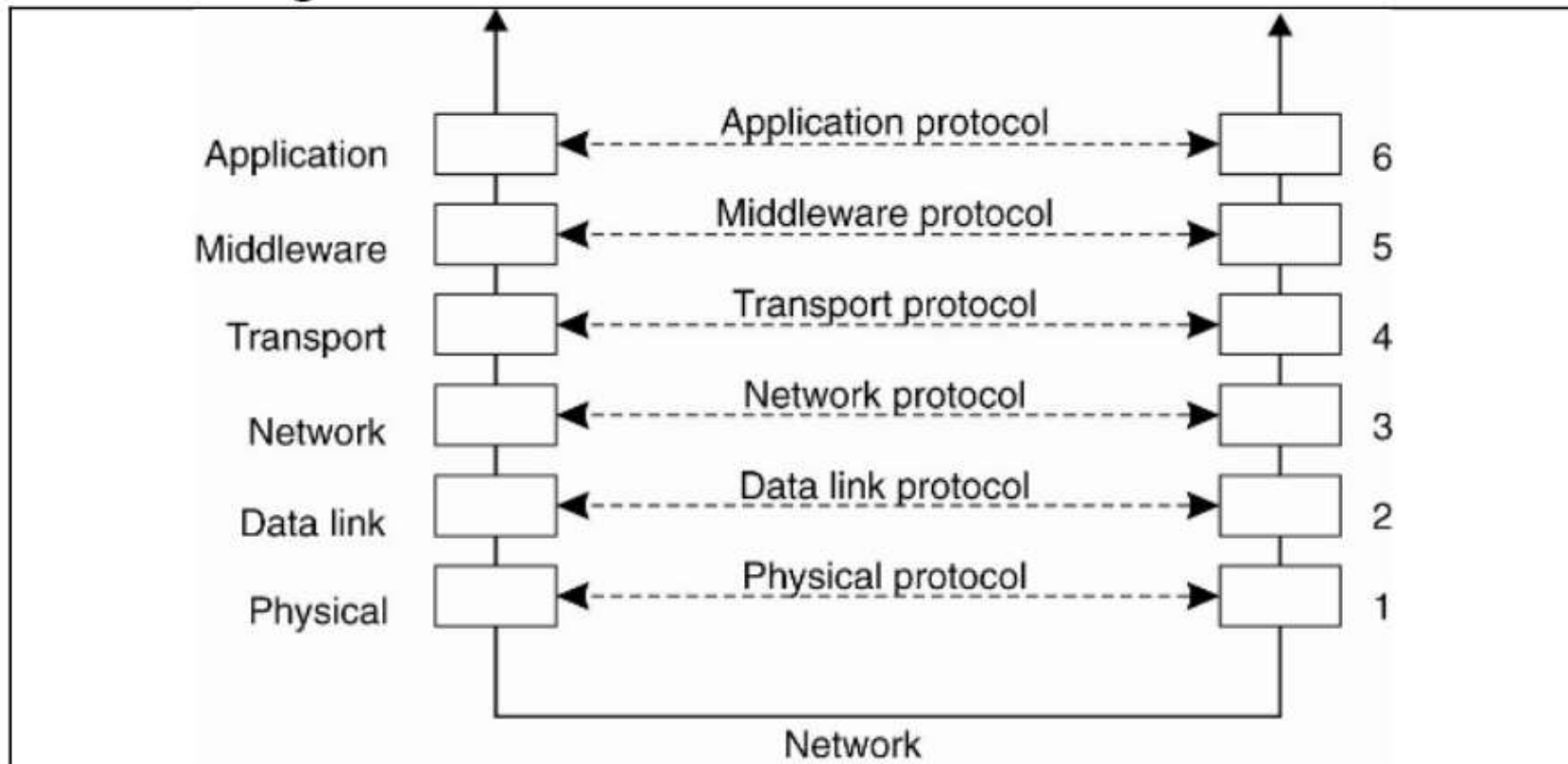
Lapisan ini adalah lapisan sebenarnya yang memberikan fasilitas komunikasi bagi kebanyakan sistem tersebar. Pada lapisan ini ada dua macam protokol yang sering digunakan, yaitu:

1. TCP (*Transport Control Protocol*) dengan karakteristik sebagai berikut:
 - a. *Connection-oriented*, sistem-sistem yang akan berkomunikasi harus terlebih dulu saling mengetahui dan sepakat.
 - b. *Reliable*, tersedia mekanisme yang menjamin paket yang rusak atau hilang dikirim ulang.
 - c. *Stream-oriented communication*.
 - d. Membutuhkan sumber daya komputasi dan jaringan lebih besar dari UDP.
2. UDP (*User Datagram Protocol*) dengan karakteristik sebagai berikut:
 - a. *Connectionless*, tidak memerlukan adanya saling mengetahui dan kesepakatan.
 - b. *unreliable (best-effort) datagram communication*, tidak tersedia mekanisme yang menjamin paket yang rusak atau hilang dikirim ulang.
 - c. Membutuhkan sumber daya komputasi dan jaringan lebih kecil dari TCP

1.4 Middleware Layer

- Lapisan adalah lapisan yang dibuat untuk menyediakan layanan-layanan dan protokol-protokol yang seragam dan dapat digunakan oleh aplikasi-aplikasi yang berbeda-beda.
 - Pada lapisan ini terdapat sekumpulan protokol komunikasi yang beragam yang memungkinkan berbagai macam aplikasi dapat berkomunikasi.
 - Middleware juga memberikan fasilitas marshaling, yaitu proses pengubahan data dalam komunikasi antar proses menjadi bentuk yang siap dikirimkan melalui jaringan sehingga dapat tetap konsisten ketika sampai di sisi penerima, dan proses kebalikannya, unmarshaling.
- 

Gambar 3.3 Layer, interface dan protokol dalam OSI model



Protokol–protokol lain yang tersedia dalam sebuah Middleware biasanya meliputi:

1. Protokol naming yang menangani masalah pemberian nama dan pencarian lokasi nama.
2. Protokol keamanan yang berfungsi mengamankan komunikasi yang bersifat penting dan rahasia.

Fungsi utama dari Middleware Layer adalah membuat kerumitan komunikasi antar proses di sistem yang berbeda menjadi transparan seolah–olah komunikasi hanya terjadi secara lokal. Layanan penting lainnya adalah mekanisme perluasan/penambahan ukuran sistem atau lebih dikenal sebagai scaling mechanism.


Fasilitas ini menyediakan dukungan untuk proses replikasi dan Caching (penyimpanan sementara).

1.5 Jenis-jenis Komunikasi dalam Middleware

Berdasarkan sifat-sifatnya komunikasi dalam sistem tersebar melalui Middleware dapat digolongkan menjadi:


1. **Transient versus persistent,**
2. **Asynchronous versus Synchronous**

Pada **persistent** communication sebuah pesan yang dikirim akan disimpan dalam tempat penyimpanan sementara selama pesan tersebut belum diterima oleh sistem tujuan sedangkan pada **transient** communication pesan segera dibuang oleh sistem komunikasi perantara ketika tidak dapat dikirimkan ke tujuan berikutnya.

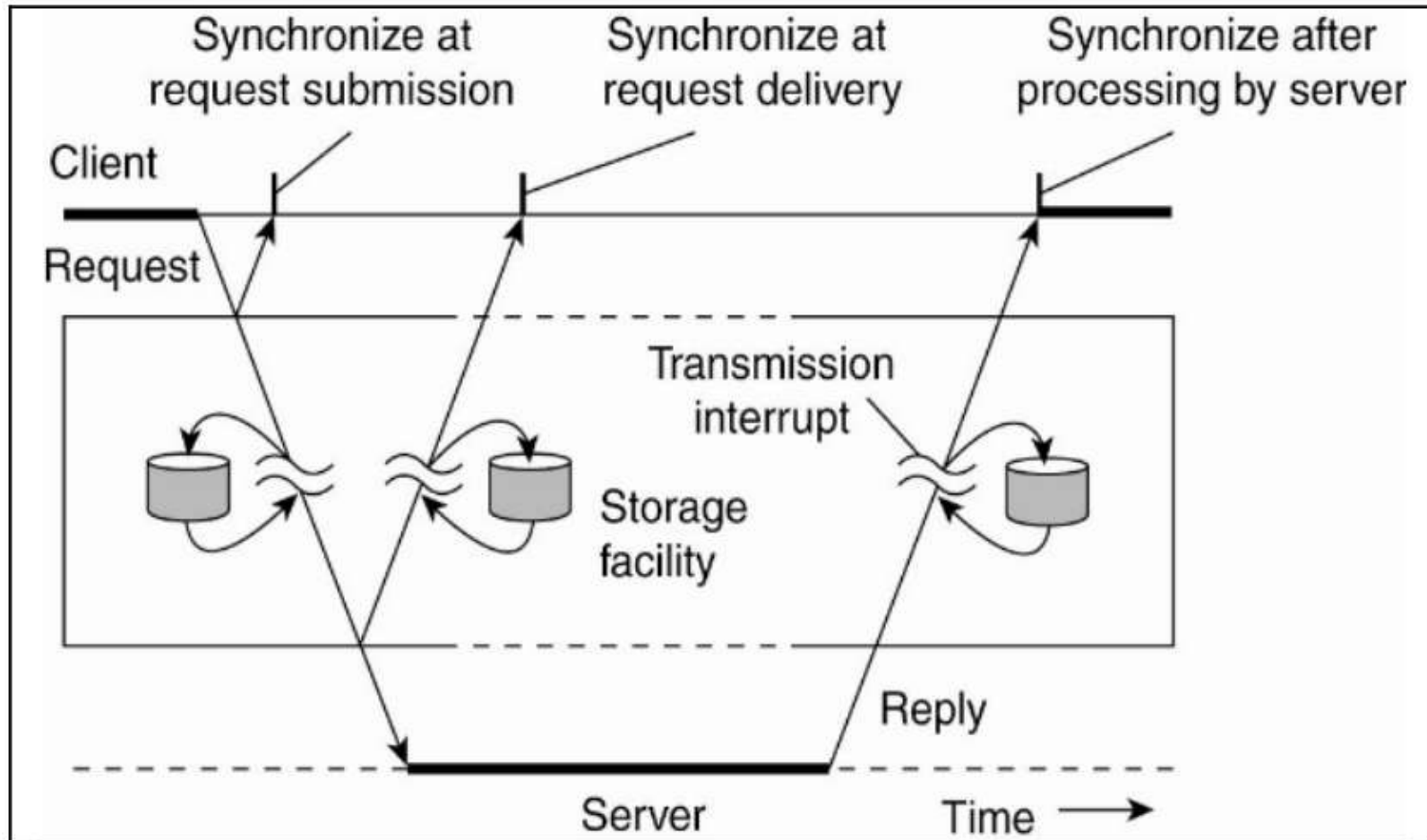


Pada *aSynchronous communication* pengirim dapat langsung mengirimkan pesan berikutnya ketika pesan terdahulu belum sampai di sisi penerima sedangkan *Synchronous communication* mensyaratkan pesan terdahulu diterima sebelum pesan berikutnya dapat dikirimkan.

Pengiriman pesan berikutnya dapat ditunda pada tiga titik berikut:

1. Sampai Middleware menyatakan akan melayani permintaan pengiriman pesan ketika diminta oleh pengirim,
 2. Sampai pesan sebelumnya terkirimkan ke penerima, atau
 3. sampai penerima pesan memberikan respon terhadap pesan yang dikirimkan.
- 

Gambar 2.1 layanan pada application-level communication Remote Procedure Call (RPC)




2.1 Procedure Calling (RPC)

- ▶ Banyak sistem tersebar dibuat berdasarkan pertukaran pesan antarproses secara eksplisit namun prosedur send dan receive yang biasa digunakan tidak menyediakan fasilitas untuk menyembunyikan kerumitan proses komunikasi antar proses pada sistem tersebar.
- ▶ Hal ini berakibat pada kerumitan mengembangkan aplikasi sistem tersebar bagi para pengembang aplikasi.
- ▶ Untuk menyembunyikan kerumitan ini dibuatlah konsep Remote
- ▶ Procedure Calling atau RPC sedemikian sehingga sebuah proses pada suatu mesin dapat memanggil prosedur yang berada di mesin lainnya. Pada saat memanggil prosedur pada mesin lain proses yang memanggil akan ditunda eksekusinya hingga mendapatkan balasan hasil eksekusi dari mesin lawannya.
- ▶ Pesan dapat dikirimkan sebagai nama dan parameter prosedur kemudian kembali diterima sebagai keluaran prosedur.

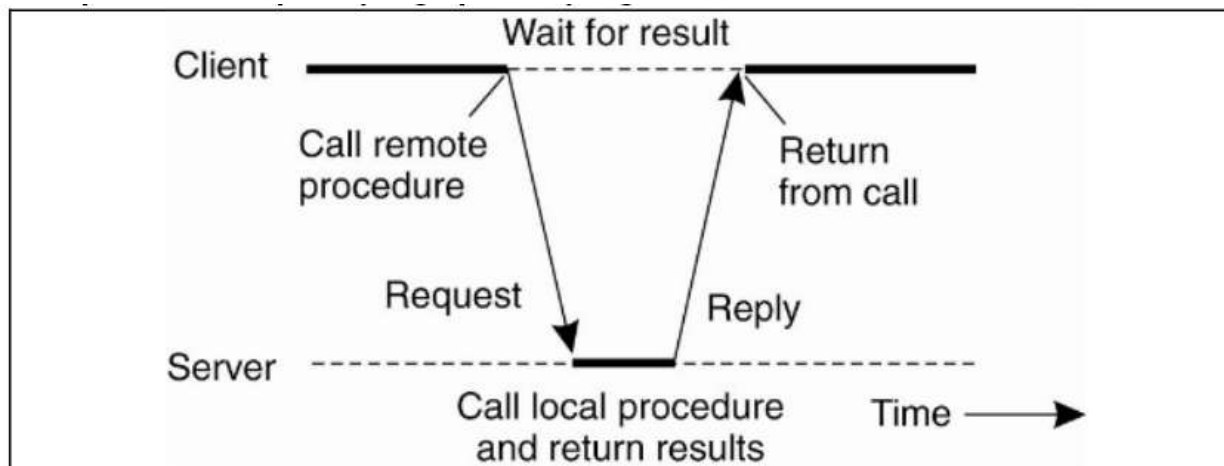
1. Operasi Dasar RPC

Pada RPC komputer yang memanggil remote procedure berstatus sebagai client sedangkan yang menyediakan prosedur yang dipanggil berstatus sebagai server. Berikut ini adalah rangkaian kejadian ketika sebuah remote procedure call terjadi:

1. Aplikasi yang berada di client memanggil prosedur yang disediakan oleh server.
2. Nama prosedur yang dipanggil dan parameter masukannya (jika diperlukan) sebelum dikirimkan melalui jaringan komputer terlebih dahulu diubah menjadi format yang sesuai untuk pengiriman melalui jaringan atau lebih dikenal sebagai proses marshalling. Proses marshalling ini dilakukan oleh suatu perangkat lunak yang dikenal sebagai client stub.
3. Client stub kemudian memanggil prosedur mengirim paket melalui jaringan yang disediakan dan menjadi tanggung jawab sistem operasi client.

4. Sistem operasi client mengolah pesan dari client stub dan menjadi paket jaringan dan dikirimkan ke komputer server.
 5. Di sisi server pesan diterima oleh sistem operasi dan diberikan kepada server stub yang melakukan proses kebalikan dari marshalling yaitu unmarshalling. Proses unmarshalling mengubah pesan dari client stub menjadi bentuk awal berupa panggilan untuk prosedur tertentu dengan parameter masukannya.
 6. Server akan memanggil prosedur yang diinginkan dengan parameter masukan yang diterima dari server stub kemudian mengembalikan hasil pemanggilan prosedur kepada server stub.
 7. Server stub melakukan marshalling pesan berisi hasil dari prosedur yang dipanggil, kemudian memanggil layanan pengiriman data dari sistem operasi server.
- 

8. Sistem operasi server mengolah pesan dari server stub menjadi paket jaringan dan dikirimkan ke komputer client.
9. Sistem operasi client menerima paket dari sistem operasi server dan menyampaikannya pada client stub.
10. Client stub melakukan unmarshalling dan memberikan hasil pemanggilan prosedur kepada yang aplikasi yang melakukan RPC.



Gambar 3.5 Basic RPC operation

RPC: Parameter Passing

2. RPC: Parameter Passing

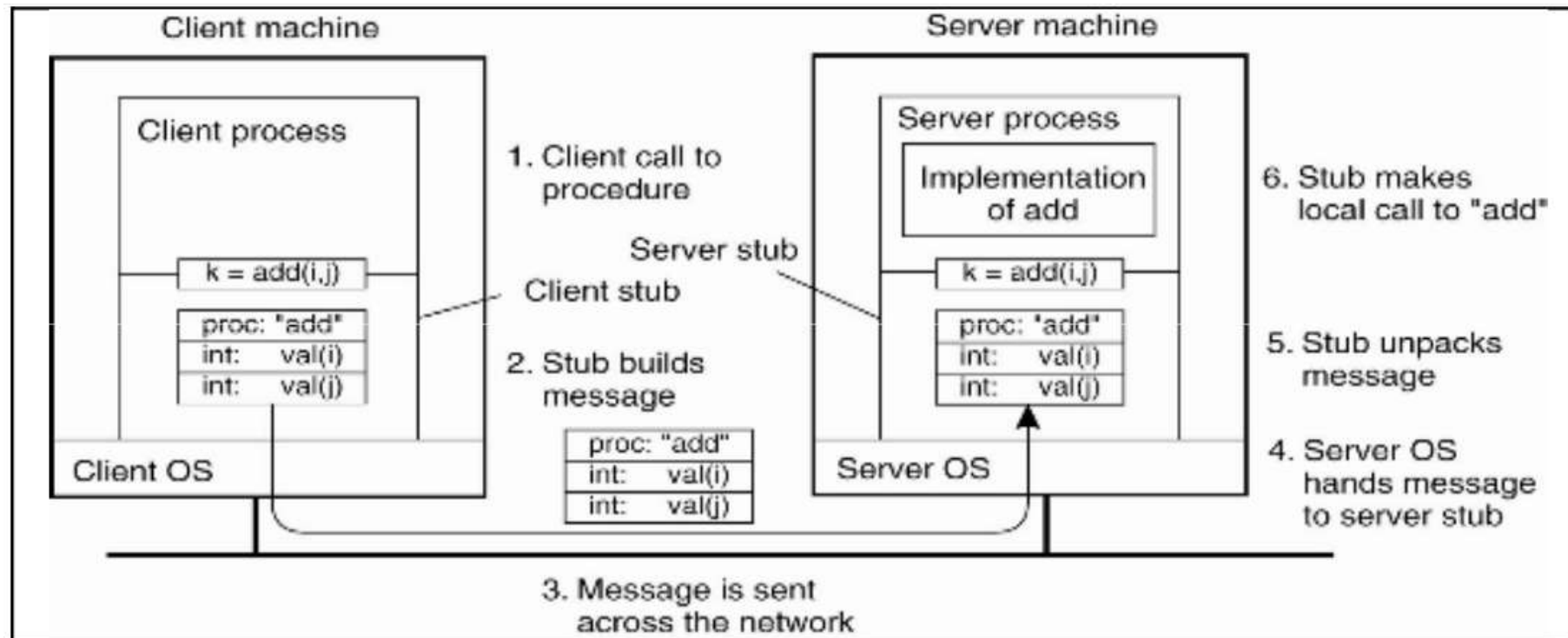
Pada saat nama prosedur dan parameter masukan diubah menjadi pesan ada beberapa hal yang perlu diperhatikan, yaitu:

1. Client dan server sangat mungkin memiliki cara pembacaan data dengan urutan yang berbeda (little endian dan bigendian).
2. Jika terjadi proses encoding maka client dan server harus sepakat mengenai encoding yang digunakan.
3. Representasi tipe data dasar seperti integer, float, dan string dalam bentuk biner dapat berbeda-beda antar komputer dan bahasa pemrograman.
4. Representasi tipe data kompleks seperti array, union, objek, dan sebagainya ke dalam bentuk biner juga dapat berbeda-beda.

Untuk menyelesaikan masalah-masalah tersebut client dan server harus mengubah pesan menjadi format representasi data yang tidak bergantung mesin maupun bahasa pemrograman.



Gambar 3.6 Remote computation through RPC

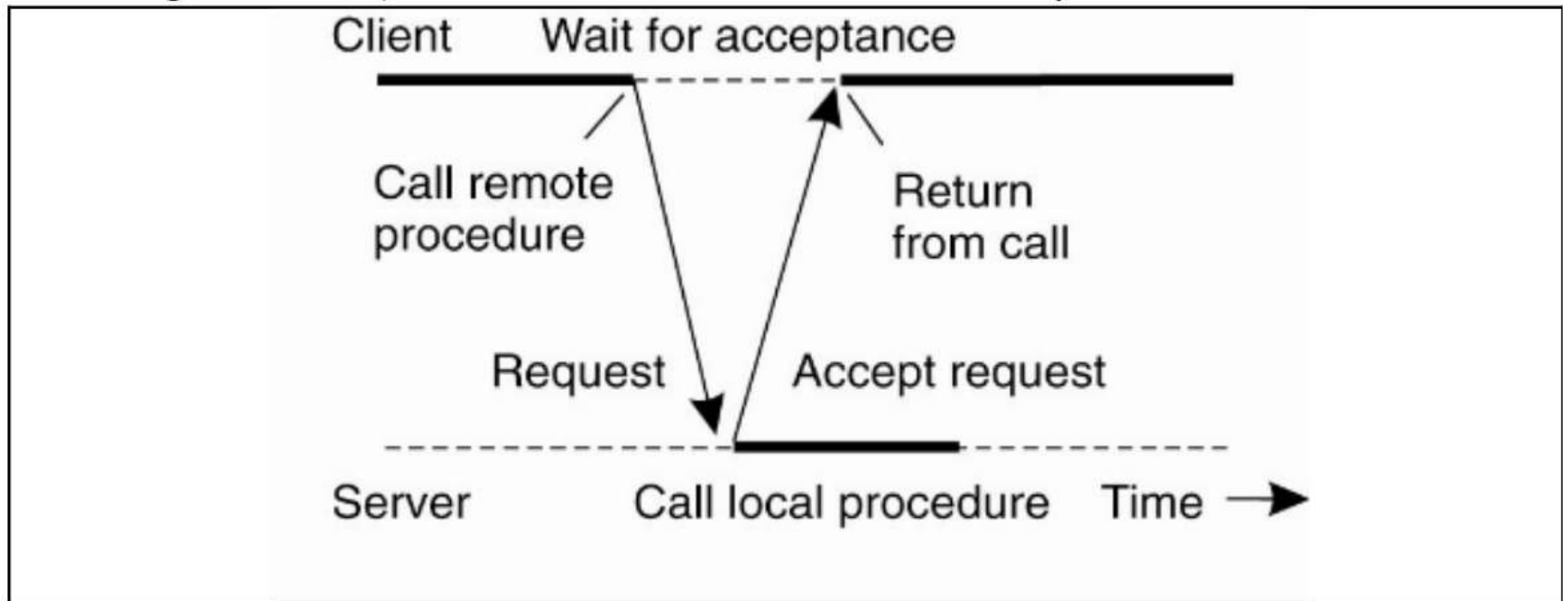


3. A Synchronous RPC

- ❑ Pada umumnya proses yang melakukan remote procedure call akan berhenti sementara hingga mendapat hasil dari sisi server. Proses-proses tertentu seperti: memulai layanan di komputer lain, menransfer uang ke rekening lain, melakukan entri data pada basis data tidak perlu menunggu hasil tertentu dan dapat melanjutkan eksekusi intruksi berikutnya.
- ❑ RPC asinkron (a Synchronous RPC) adalah teknik sedemikian sehingga proses yang memanggil remote procedure tidak menunggu hasil dari sisi server untuk melanjutkan mengeksekusi instruksi berikutnya.
- ❑ Pada RPC asinkron server langsung memberikan balasan ketika pesan dari client diterima tanpa memproses pesan terlebih dahulu dan begitu balasan dari server diterima client segera melanjutkan eksekusi instruksi berikutnya.

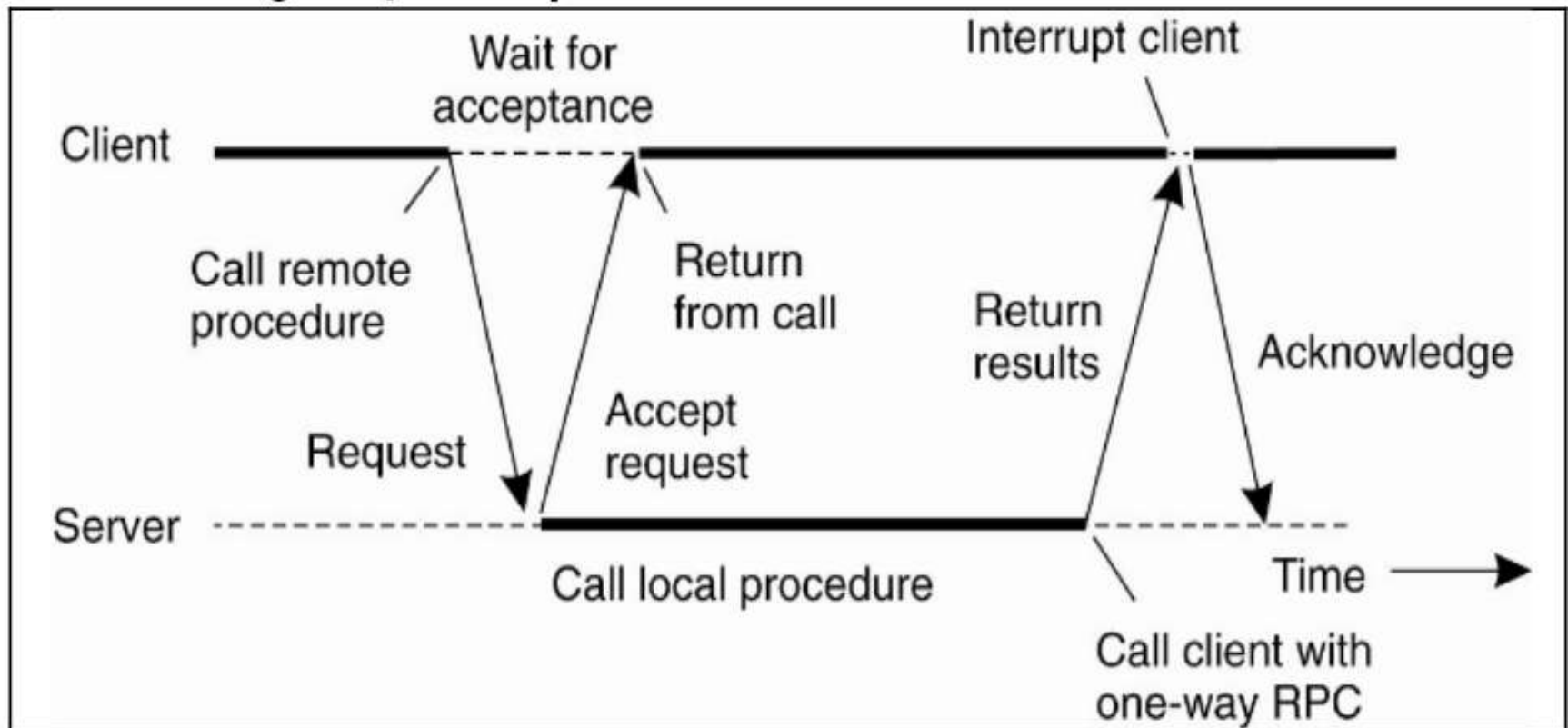
Gambar 3.7 Asynchronous RPC.

Bandingkan dengan operasi dasar RPC pada gambar 3.5




- ❑ Pada kasus lain sebuah server dapat saja membutuhkan waktu untuk menjalankan prosedur yang dipanggil oleh Client. Contohnya adalah permintaan melakukan pencarian data tertentu pada server yang dapat membutuhkan waktu cukup lama.
- ❑ Client tidak perlu menghabiskan waktu dengan menunggu jawaban hasil pencarian melainkan dapat langsung melanjutkan eksekusi instruksi selanjutnya. Pada saat jawaban sudah selesai disusun oleh server dan dikirimkan ke client proses yang memanggil prosedur pencarian akan diinterupsi agar mengetahui bahwa jawaban dari server telah datang dan dapat mengolah jawaban tersebut sesuai kebutuhannya.
- ❑ Teknik ini dikenal sebagai deferred Synchronous RPC.

Gambar 3.8 Client dan server berinteraksi dengan 2 a Synchronous RPC



4. RPC pada Tataran Praktis: DCE

- ▶ DCE atau Distributed Computer Environment adalah sebuah Middleware yang berfungsi untuk menjadi sebuah lapisan yang menyembunyikan kerumitan lapisan sistem operasi dan lapisan komunikasi jaringan dalam RPC.
 - ▶ DCE awalnya dikembangkan untuk sistem operasi UNIX namun saat ini telah dibuat agar dapat dijalankan di berbagai sistem operasi terkenal termasuk Windows dan VMS.
- 


3.3.1 Data stream

- ▶ Data stream atau aliran data adalah rangkaian unit-unit data yang dapat disampaikan secara diskrit (dalam paket-paket) maupun kontinyu (aliran tidak terputus/terbagi-bagi). Pengaturan waktu sangat penting ketika aliran data dikirimkan secara kontinyu. Untuk memberi perlakuan yang sesuai berdasarkan pengaturan waktu transmisi data dapat dibedakan menjadi beberapa jenis, yaitu:

1. Transmisi asinkron (aSynchronous transmission)
2. Transmisi sinkron (Synchronous transmission)
3. Transmisi isokron (isochronous transmission).


1. Transmisi asinkron

Transmisi asinkron (aSynchronous transmission) Pada transmisi jenis ini data dalam stream dialirkan secara berurutan tetapi tidak ada batas waktu kapan transmisi data berlangsung. Transmisi ini biasanya berlaku pada pengiriman data yang bersifat diskrit seperti file atau pesan.



2. Transmisi sinkron

Transmisi sinkron (Synchronous transmission) Kebalikan dari transmisi asinkron transmisi jenis ini memiliki batas waktu maksimum jeda pengiriman dari ujung ke ujung untuk setiap unit aliran data. Pada transmisi ini tidak penting apakah data dikirimkan lebih cepat dari batas maksimum jeda. Contoh aplikasinya adalah pada sensor suhu yang mengirimkan sampling suhu kepada suatu pengendali. Data sampling yang tiba lebih cepat dari batas maksimum tidak memberikan pengaruh apapun terhadap pekerjaan pengendali.



3. Transmisi isokron

Transmisi isokron (isochronous transmission). Pada transmisi jenis ini terdapat batas maksimum dan minimum jeda pengiriman dari ujung ke ujung atau yang dikenal sebagai filter. Transmisi ini berlaku pada pengiriman data audio, video dan multimedia.

Terima Kasih