Distributed Systems

Exercises

Werner Nutt

Networking

HTTP

1. Suppose, a user requests a Web page that consists of some text and two images. Is it the case that for this page, the client will send one request message and receive three response messages?

2. Suppose, you click a URL on a Google page and you type the same URL into the top window of your browser.

   Is it the case that with the response you will receive the same page in the two cases?

3. If you call type “www.google.com” from a university in Italy and from one in Germany, will you get the same page? Explain your answer.

Routing

Consider a packet network using 8-bit host addresses. Suppose a router uses longest prefix matching and has the following forwarding table:

Prefix Match	Interface
1	0
11	1
111	2
otherwise	3

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

Subnetting

Consider a router that connects three subnets S1, S2, S3. Suppose all of the interfaces in each of these three subnets are required to belong to the network 223.1.17.0/24. Also, suppose that S1 is required to support up to 125 interfaces and S2 and S3 are each required to support up to 60 interfaces.

1. Provide three network addresses of the form a.b.c.d/x that satisfy these requirements.

Network Address Translaters

This problem deals with the impact of NATs on P2P applications like internet telephony. Suppose a peer with username Arnold discovers that a peer with username Bernard is online. Also, suppose that Arnold and Bernard are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard.

If you have difficulty devising such a technique, discuss why.

Threads

Threads

Consider bank accounts that are modelled as objects and that each bank account object is identified by its number. Suppose that transfers of money from one account to another one should happen concurrently and are therefore implemented as threads.

- Which problem may arise if two threads access the same account? How do you avoid this problem?
- Describe how this solution may cause deadlocks.
- Describe a technique by which these deadlocks can be avoided.

Threads (contd.)

A file server uses caching, and achieves a hit rate of 80%. File operations in the server cost 5 ms of CPU time when the server finds the requested block in the cache, and take an additional 15 ms of disk I/O time otherwise.

Explaining any assumptions you make, estimate the server's throughput capacity (average requests/sec) if it is:

- single-threaded;
- two-threaded, running on a single processor;
- two-threaded, running on a two-processor computer.

Request-Response Protocols

Communication Types

- Asynchronous: sender continues after submission

- Synchronous: sender is blocked until
  - message is stored at receiver's host
  - message is received
  - response is received

Request/Response Protocols

Basically all client/server communication follows the pattern of a request/response protocol:

- the client *sends* a request message
- the server *executes* the requested operation
- the server *responds* with a response message

1. Give an example of a request/response protocol in the real world.

2. Discuss the pros and cons of synchronous and asynchronous communication for a request/response protocol. Under which conditions is synchronous communication preferable? When asynchronous communication?

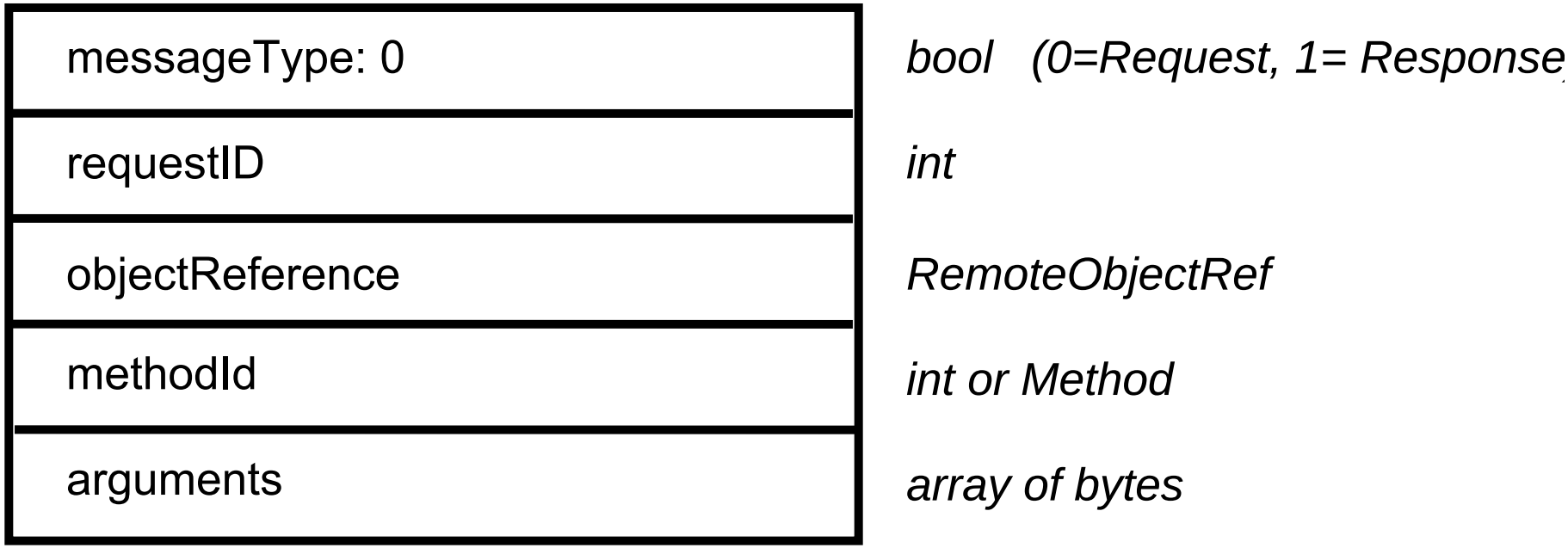
Request/Response Protocols

2. Discuss the pros and cons of synchronous and asynchronous communication for a request/response protocol.
   i. What are the advantages of synchronous communication?
   ii. What are the advantages of asynchronous communication?

Request-response Message Structure

Imagine a request-response protocol where a request is an invocation of a method of a remote object.

Suppose requests have the following format:



Request-Response Message Structure (cntd)

3. Under which circumstances is a requestID needed?
   (Hint 1: Remember that messages are sent over networks;
   Hint 2: Remember Question 2)

4. What elements should a response message have?

Request-Response at the Transport Level

5. Which transport protocol would be more suitable to implement a request response protocol, UDP or TCP?

6. Does the answer depend on additional circumstances?

7. Over which transport-level protocol is the request-response protocol implemented that you mentioned as answer to Question 1?

Datagram-based RRP

Suppose a Request-Response Protocol is implemented using UDP.

8. What can go wrong in a message exchange?
   Describe 3 possible problems.
   (Think of message loss and its consequences.)

9. For each problem above, describe a (simple) refinement of the basic request-response protocol that overcomes it.

Invocation Semantics

For request/response protocols, one distinguishes between three semantics that are distinguished by the guarantees given for the execution of the *server's operation*:

- Maybe Semantics: the server may execute the request once, several times, or not at all
- At-least-once Semantics: the server executes the request at least once, but may execute it more often, until the client receives an answer
- At-most-once Semantics: for each request, the server executes the operation at most once; the invoking application receives the result or an exception

Invocation Semantics (cntd)

11. Explain: for which of the three semantics is it necessary that the client transmits requests more than once?

12. Explain: for which of the three semantics is it necessary that the server remembers the requests it has answered?

13. For each of the three semantics, explain what the server should do when it receives a request.
    Hint: Find out, whether the server should remember something, and if so, what.

Idempotent Operations

14. Suppose a server receives the same client request more than once. How should it react the second time?

An operation of the server is called “idempotent” if it leads to the same result, independent of how many times it is executed.

15. When running a request/response protocol, does it make a difference whether server operations are idempotent or not? Does this depend on the invocation semantics? Explain your answer.

Request-Response with Acknowledgements

In the Request-Response-Acknowledgement (RRA) protocol the client acknowledges the server's response messages, and the acknowledgement message contains the ID in the response message being acknowledged.

16. For which of the three invocation semantics does RRA allow for an optimisation? Is the optimisation on the client or on the server side?

    Describe how the implementation of client and/or server changes for this semantics when moving from RR to RRA.

Request-Response with Piggy-Backed Acknowledgements

17. Design a variant of the RRA protocol in which the acknowledgement is piggy-backed on, that is, transmitted in the same message as, the next request where appropriate, and otherwise sent as a separate message. Which changes are necessary on the server and on the client side?

Coordination and Agreement

Bully Algorithm

1. Sketch the Bully Algorithm. Remember there are 3 types of messages:
   - election, vote, coordinator.
2. What happens if two processes notice at the same time that the leader has crashed?
3. In the Bully algorithm, a recovering process starts an election and will become the new coordinator if it has a higher identifier than the current leader.
   - Is this a feature that is necessary for the algorithm to work correctly?
   - Under which circumstances can it be dropped?

Synchronous vs. Asynchronous Distributed Systems

1. Define: When is a distributed system
   - synchronous
   - asynchronous?

2. Which of the following distributed algorithms functions only in a synchronous system?
   - Ring-based leader election
   - Bully algorithm

3. Does one of them only function in an asynchronous system?

Lamport's Logical Clocks

Consider three distributed processes P1 , P2 , and P3 . The processes are involved in the events a, b, . . . , k listed below, which happen at specific points in time, specified as "wall-clock time" (WCT) and measured in ms:

At 0ms WCT:

a: P2 sends message m1 to P1

b: P3 reads a startup file

At 100ms WCT:

c: P1 opens a file containing a user profile

d: P3 sends message m2 to P2

At 200ms WCT:

e: P1 receives message m1

f : P2 receives message m2

Lamport's Logical Clocks (cntd)

At 300ms WCT:

g: P1 sends message m3 to P2 and P3

At 400ms WCT:

h: P2 receives message m3

At 500ms WCT:

i: P2 sends message m4 to P1

At 600ms WCT:

j : P1 receives message m4

At 600ms WCT:

k: P3 receives message m3

Lamport's Logical Clocks (cntd)

1. Which of these events are related by Lamport's "happened before" relation (in the lecture denoted as "E1 → E2 " if event E1 happened before event E2)? Draw a directed graph where the events are vertices and where there is an edge from E1 to E2 if E1 happened before E2.

2. Associate to each event a logical timestamp according to the logical clock algorithm. Use process IDs to break possible ties. What is the linear order of events induced by these timestamps?

Concurrent Events wrt “Happens-Before”

Consider two processes P1, P2 where sending and receiving messages are the only events.

- Can you construct an example of two events that are concurrent? That is, none “happened before” the other?
- Is it possible that two events are concurrent and have different logical timestamps?

Ring-based Mutual Exclusion and Ordering Property

Consider the ring-based algorithm for mutual exclusion.

Assume that the following are possible events in a process:

- Sending/receiving a message
- Entering a state where a shared resource needs to be accessed ("request" event).

- Can it happen that process P1 needs to access the shared resource before process P2 (in "happens-before" order), but gets access only after P2?

Ordering and Single-Threaded Processes

Consider a scenario where the central-server algorithm for mutual exclusion is applied. Suppose that all processes requesting tokens are single-threaded.

- Is it possible that the order in which requests are served contradicts the “happens-before” ordering?
- Does this change if the server orders the entries in its queue according to their Lamport timestamps?

Ricart-Agrawala's Algorithm

Given a distributed system with three processes P1, P2, and P3, in three different sites where mutual exclusion is enforced using Ricart-Agrawala's algorithm. Consider the scenario below and list in detail all the events that follow until both processes complete their critical sections.

- Both P1 and P2 are requesting the critical section while P3 has no need at this time.
- The timestamp of P1's request is 47, and that of P2's request is 32.