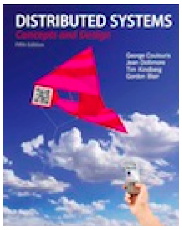

Distributed Systems: Concepts and Design

Edition 5

By George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair

Addison-Wesley ©Pearson Education 2012

Chapter 19 Solutions to Exercises

19.1 What is mobile computing? What are the different problems are associated with wireless communication?

19.1 Ans.

Mobile computing is a paradigm in which users could carry their personal computers and retain some connectivity to other machines.

The different problems that are associated with wireless communication are:

- how to provide continuous connectivity for mobile devices that pass in and out of range of base stations, which are infrastructure components that provide regions of wireless coverage.
- how to enable collections of devices to wirelessly communicate with one another in places where there is no infrastructure.

19.2 Discuss whether it is possible to improve upon the 'pull' model of service discovery by multicasting (or broadcasting) and caching replies to queries.

19.2 Ans.

We start with the 'pull' model described in the chapter but turn it into a hybrid with the 'push' model by having services multicast/broadcast their existence when a query occurs. All potential client devices can then hear and cache the service advertisements. When a component on the device requires a service, it first consults the local cache. If no matching services are found or if a more up-to-date list of matching services is required, then the device still multicasts its query. However, now it includes information about any matching services in its cache. Other potential clients may now learn about those services' existence. Moreover, only services not in the query need reply.

While it is intuitively reasonable, this hybrid scheme needs validation through simultaion or experimentation against the required implementation parameters. The extra multicasts incur a cost.

19.3 Explain the different constraints that mobile and ubiquitous devices have to overcome.

19.3 Ans.

The different constraints are:

a) Limited energy: A device that is portable or embedded in the physical world typically has to run on batteries, and the smaller and lighter the device needs to be, the lower will be its battery capacity. Replacing or re-charging those batteries is liable to be inconvenient in terms of time and physical access.

b) Resource constraints: Mobile and ubiquitous devices have limited computational resources in terms of processor speed, storage capacity and network bandwidth.

19.4 Describe the different forms of volatility exhibited by volatile systems. Give an example of a system, other than mobile or ubiquitous systems, which shows some form of volatility.

19.4 Ans.

The important forms of volatility are:

a. failures of devices and communication links;

b. changes in the characteristics of communication such as bandwidth;

c. the creation and destruction of associations – logical communication relationships – between software components resident on the devices.

Systems that demonstrate one or more forms of volatility but which are neither mobile nor ubiquitous is peer-to-peer computing such as file-sharing applications.

19.5 Describe a situation in which motes can be useful. Why are motes also known as smart dust?

19.5 Ans.

Consider a forest fire; then one or more motes scattered around the forest could sense abnormally high temperatures and communicate those, via their peers, to a more high-powered device capable of communicating the situation to the emergency services.

Motes are also known as 'smart dust', reflecting the tiny size ultimately intended for these devices.

19.6 What is meant by the "physically driven spontaneity of associations"? How is this different from associations that are data-driven?

19.6 Ans.

Physically driven associations are made and broken according to the current physical circumstances of the components, in particular their proximity.

Data driven association often originates from the human but it is the value of the data provided to it that causes a peer to make associations with a peer it may never have associated with before and whose address was not formerly stored by it.

19.7 Discuss the issue of how the scope of an event system can and should be related to the physical extent of a smart space in which it is used.

19.7 Ans.

This is a 'boundary principle' question. Components in a hotel guest room probably don't need to know (and shouldn't know) about events occurring in the guest room next door. However, certain hotel management components hosted elsewhere in the building may have an interest.

One approach would be for events to be propagated by default only to components within the smart space. If necessary, those events could also be relayed to components outside the smart space by a proxy component in the smart space.

But how to ensure that propagation is physically limited in scope? The techniques introduced for physically scoping asociation (Section 16.2.2) can all be applied to this case, too.

19.8 Compare device discovery and service discovery services. Give an example of a system that includes both device discovery and service discovery services.

19.8 Ans.

With device discovery, clients discover the names and addresses of co-present devices. Typically, they then choose an individual device on the basis of out-of-band information (such as selection by a human) and query it for the services it offers. On the other hand, a service discovery service is used where clients are not concerned with which device provides the service they need, but on the attributes of the service alone.

Bluetooth includes both device discovery and service discovery services.

19.9 Describe the five issues to be dealt with in the design of a discovery service. When is a discovery service known as a network discovery service?

19.9 Ans.

The issues to be dealt with in the design of a discovery service are as follows:

a. Low-effort, appropriate association. Ideally, appropriate associations would be made without any human effort. First, the set of services returned by the queryoperation (Figure 19.3) would be appropriate – they would be precisely the services existing in the smart space that matched the query. Second, service selection could be made programatically or with minimal human input so as to meet the users' needs.

b. Service description and query language. The overall goal is to match services to clients' requests for services. That presupposes a language for describing available services, and one for expressing service requirements. The query and description languages have to agree (or be translatable); and their expressiveness has to keep pace with the development of new devices and services.

c. Smart-space-specific discovery. We require a mechanism for devices to access an instance (or scope) of the discovery service that is appropriate to their current physical circumstances – a mechanism that doesn't rely on the device knowing the particular name or address for that service a priori. In practice, discovery services are related to a particular smart space only through the limited reach of multicast over a subnet that intersects with it, as we shall explain.

d. Directory implementation. Logically, each instance of a discovery service involves a queryable directory of available services. There are several ways of implementing such a directory, with varying implications for network bandwidth, timeliness of service discovery, and energy consumption.

e. Service volatility. Any service in a volatile system has to efficiently and gracefully handle the disappearance of a client. A discovery service has services as clients, and it needs to handle service disappearance appropriately.

Discovery services based only on network reach are known as network discovery services.

19.10 Explain the data-oriented interoperability features of tuple space systems with an example.

19.10 Ans.

Tuple space systems allow application-specific tuples to be exchanged, and the basis for association and interoperation is the components' agreement about structures for tuples and values contained within them.

Example: a digital camera could discover the tuple space for the local smart space – a hotel room, say – and place its images in the tuple space.

19.11 The Speakeasy project applied the same design principles as JetSend to device-device interoperation, but with the difference that mobile code is utilized. What are the motivations for using mobile code in the Speakeasy project?

19.11 Ans.

There are two motivations for using mobile code:

a. The first is that a device such as a printer can send any user interface to a user of another device such as a smart phone. A mobile-code implementation of a user interface can perform local processing such as input validation; and it can provide interaction modes not available in user interfaces that have to be specified in a mark-up language.

b. The second motivation for mobile code in device interoperation is optimization of data transfer. While Speakeasy's mobile code has to work within the constraints of the host device's API, it may perform arbitrary interactions with the remote device that sent it.

19.12 An important aspect of a sensor is its error model. Explain the different approaches to state a sensor's error behavior.

19.12 Ans.

A useful way to state a sensor's error behavior is to quote an accuracy that it reaches for a specified proportion of measurements, for example: 'Within the given area, the satellite navigation unit was found to be accurate to within 10m for 90% of measurements'.

Another approach is to state a confidence value for a particular measurement – a number (usually between 0 and 1) chosen according to uncertainties encountered in deriving the measurement.

19.13 Researchers have devised various software architectures to support context-aware applications. Explain some of the functional challenges to be overcome in designing context-aware systems.

19.13 Ans.

Functional challenges to be overcome in designing context-aware systems:

a. Integration of idiosyncratic sensors.

b. Abstracting from sensor data.

c. Sensor outputs may need to be combined.

d. Context is dynamic

19.14 We described several technologies that provide receive-constrained channels for use in secure spontaneous device association. Which of those technologies also provide send-constrained channels?

19.14 Ans.

Physical contact. Clearly this also provides send-constrained channels.

Infrared. By default, this provides us with a channel that is send-constrained only up to the boundary of a room, say, since an infrared receiver on, say, a PDA or mobile phone could receive data from many possible sources. However, it is possible to mount a receiver behind a lens to prevent signals from sources outside a required direction.

Audio. This provides us with a channel that is send-constrained only up to the boundary of an insulated room, say, since received audio could emanate from many possible sources.

Laser. By default, this provides us with a channel that is send-constrained only up to the nearest optical barrier. However, by mounting a receiver at the end of a narrow, opaque, non-reflective tube one could limit sources to lie in a given direction.

Barcode and camera. This combination can provide useful send-constrained channels. A camera can be focused and positioned so as to read data only from symbols presented from a given orientation and distance. For example, a user can place her camera phone so as to be sure of reading just the symbol presented by a given device.

19.15 Show how to construct a send-constrained channel from a receive-constrained channel, and vice versa. Hint: use a trusted node connected to the given channel.

19.15 Ans.

Let rc be a receive-constrained channel. We shall show how to construct a channel s(rc), which is send-constrained.

We use a trusted node N. When it receives a message m, it uses the receive-constrained channel to return to the sender a signed hash sig{h(m, t)}, where t is the time by N's clock.

Let c be any (possibly unconstrained) channel connecting the parties that we wish to be able to communicate. We construct s(rc) from c and N. The rules for sending and receiving on s(rc) are as follows:

To send m on s(rc):

send m to N

receive < t, sig{h(m, t)}> from N over rc

send <m, t, sig{h(m, t)}> on c.

To receive m on s(rc):

receive <m, t, h> on c

verify h = sig{h(m, t)}

verify currency of t and freshness of h

Discard m if verification fails, else receive m.

We assume that the receivers' clocks are synchronised to N's clock. The timestamp (and hence state of the sender) is deemed current if it is within a given bound of the time on the receiver's clock. To prevent replay attacks, the receiver need remember the hashes for only a limited time: older messages will have a non-current timestamp.

In a similar fashion, we can implement a receive-constrained channel from a send-constrained channel. All messages are sent (over any channel) to a trusted node, which stores them. Receivers must use a particular send-constrained channel to reach that node, which responds with the next message for them.

19.16 A group of smart spaces are connected only by a space between them such as a hallway or square. Discuss the factors that determine whether that intervening space can act as a mix zone

19.16 Ans.

A mix zone is an area in which users do not access location-aware services (i.e., do not give up an identifier correlated to a location). The idea is that users change their (pseudonymous) identifiers in the mix zone, and the mix zone acts to prevent correlations between the identifiers used by a particular individual.

To serve as a mix zone, there needs to be sufficiently many travelling through it at any given time. Moreover, it should be impossible to make an inference based on distances and likely speeds of travel. If, for example, two spaces are known to be close to one another, then an absence of an identifier in one correlated with a new presence in another shortly after may provide evidence that those two identifiers belong to the same person.

19.17 The devices in the volatile systems are very much heterogeneous in nature. What are the objectives of developing adaptive systems for such an environment?

19.17 Ans.

The aim of adaptive systems is to accommodate heterogeneity by allowing software reuse across contexts that vary in factors such as device capabilities and user preferences, and to accommodate changing runtime resource conditions by adapting application behavior without sacrificing crucial application properties.

19.18 What does the association problem comprise? What are the two important aspects of the association problem? What is the boundary principle?

19.18 Ans.

Once a device can communicate in the smart space, it is faced with the problem of how to associate appropriately within it. This problem is known as association problem.

The two important aspects of the association problem are scale and scope.

The boundary principle is that smart spaces need to have system boundaries that correspond accurately to meaningful spaces as they are normally defined territorially and administratively. Those ‘system boundaries’ are system-defined criteria that scope but do not necessarily constrain association.

19.19 A Cooltown user squirts the URL of a sound file or a streaming radio station to an Internet radio. Suggest a modification to the eSquirt protocol that would enable the user to control the volume from their portable device. Hint: consider what extra data the squirting device should provide.

19.19 Ans.

One approach would be to provide a transport address along with the URL of the content to be rendered. The recipient device could then make a callback to the sending device, supplying a URL for obtaining the recipient device's control web page.

19.20 Discuss the applicability to mobile and ubiquitous systems of techniques drawn from the areas of (a) peer-to-peer systems (Chapter 10); (b) coordination and agreement protocols (Chapter 15) and (c) replication (Chapter 18).

19.20 Ans.

This is mainly a question for class discussion but some key observations are:

(a) peer-to-peer systems are volatile but they are not necessarily physically integrated, e.g. devices involved might be conventional PCs and typically there is no need to apply the boundary principle. Some of the p2p algorithms assume there is a connected infrastructure.

(b) the coordination and agreement protocols are in general designed on the assumption that failure is the exception (e.g. election algorithms) whereas in volatile systems it is the rule. Many of these techniques are too expensive in communication or too slow to converge to be run frequently. Notions such as consensus are too strong to implement and have to be replaced by more pragmatic, weaker abstractions such as soft state. However, multicast provides a good level of indirection as long as relatively cheap-to-implement semantics suffice.

(c) Chapter 15 deals with disconnected operation. However, in general the techniques considered in this chapter tend to assume a redundancy of resources whereas in ubiquitous and mobile systems the opposite is generally true.

19.21 Volatile systems give rise to different types of resource sharing which require security designs. Give an example of spontaneous interoperation and explain.

19.21 Ans.

Example that raises security and/or privacy issues:

Two employees of the same company who encounter one another at a conference wirelessly exchange a document between their mobile phones or other portable devices.