can be represented by lines, a borough can be represented by polygons, etc. On the other hand, in the continuous variation data model, a geographical region is divided into spaces, and each attribute forms a layer with different number of spaces associated with it. In general, vector representation is used in the map database due to its flexibility.

In the following sections, we describe the operations involved in supporting the spatial analysis service.

25.3.1 Geocoding and Reverse Geocoding

One of the basic operations to support mobile location services is to search for a certain location in the digital map, with the address input by the user. However, different addressing schemes are applied in different countries and different input formats are possible for the same address. These factors greatly increase the complexity of the process of geocoding, which includes the following steps:

1. Getting user's address input. The complexity of the operation of the geocoding engine is greatly affected by the items, details and correctness of the address input by the user. Generally, the simpler the user's address input, the higher the complexity of the operation performed by the geocoding engine.

2. Standardizing the address input. Since different countries are using different addressing schemes, in order to perform an accurate search, standardization of the address input is required. In this step, the geocoding engine parses the address input into different fields like address, name, suffix and direction. With the standardized formats, the search can be done more easily.

3. Address searching and ranking matches. After the standardization of the address input, a search on the database is performed by the geocoding engine the address input. If it is possible to find a match in the database, the position of the address can then be returned directly. However, an exact match is generally difficult to occur due to user input errors like abbreviations used in the address input, parsing errors due to different addressing schemes, map database errors like unrecorded areas, etc. Therefore, other searching methods are required to assist in searching. A common method used is soundex. Soundex converts letters in the addresses to some codes based on the pronunciation of the letters. The codes are then be for searching. If multiple matches occurs, then an address range search is performed to verify the validity of the matches. Finally, the verified matches are ranked according to some preset priorities in the matching process, for example, matches in the different fields of the address input.

On the other hand, reverse geocoding is the reverse process of the geocoding that we have just discussed. Given a user's position in terms of latitude and longitude, a

map lookup will return the nearest match of the current location of the user on a map. This is useful in figuring out the nearest road segment for directing the driving, and helpful for searching interested points nearby.

25.3.2 Routing

Routing is the process of finding suitable paths from the source location to the destination location based on the criteria given. Basically, the criteria given is turned into costs of links, which are then used in the calculation of the most suitable path by adding the costs of the links involved. With the estimated costs in each link, the routing operations can be captured by one of the following well-known [166]:

Shortest Path Problem: This captures the problem in which the user wants to find the shortest path from his current location to the destination location.

Travelling Salesman Problem: This captures the problem in which the user wants to visit multiple destinations with minimum costs.

Single Depot Multiple Vehicles Routing: This captures the problem in which stocks are stored in a centralized location (i.e., a depot) and multiple vehicles are used to deliver products to the customers with minimum costs.

Multiple Depots Multiple Vehicles Routing: This captures the problem in which multiple stocks are stored in multiple depots and multiple vehicles are used to deliver products to customers with minimum costs.

25.3.3 Other Operations

Other operations of the spatial analysis service include map formation, point of interest search, real-time modification of attributes, etc. These operations are briefly discussed below:

Map Information: There are two main types of map representation, namely vector map and raster map. In vector map representation, a map is represented by points, lines and polygons. This representation method has the advantage of having good modeling of the shape of real objects with smaller amount of data. On the other hand, in raster map representation, a map is represented by pixels. This representation method has the advantage of reducing the complexity of client devices by offloading the complexity to the server. Moreover, the server returns only simple graph images, and the client devices only need to update images after each operation performed on a map (e.g., a zoom-in operation). Currently, raster map representation is commonly used in client devices.

Point of Interest Search: It is a useful operation to help users to identify their targets nearby, so as to direct them to the locations of their targets. There are

a number of search types, such as search by name, search by type of facility (e.g., shops), search by distance from a point, search by route/region, etc.

Map Attributes Modification: Basically, map attributes are seldom modified. However, since some attributes like traffic information along a route or weather information can affect the cost calculation process in routing, these kinds of attributes are required to be changed in a short period of time. This is to ensure that optimal operations of routing can be achieved.

25.4 MOBILE POSITIONING TECHNIQUES

Depending on particular requirements needed, different location based applications require positioning information up to certain degree of granularity with certain response time. For example, most in-vehicle mobile location services like air-bag deployment notification are emergency services which require very fine-grained and accurate location positioning information in a very short time, while personal mobile location services like town and weather information of the user's current location require relatively loose accuracy of information and response time.

As a result, there are various techniques for finding the location of the user. Existing techniques can be divided into three main types—cellular location techniques, GPS (Global Positioning System) location techniques, and the hybrid of the two types. Basically, cellular location techniques can achieve an acceptable accuracy of location information with a fast response time. Most of them allow non-line-of-sight operations and therefore in-door operations are possible. However, their accuracy deteriorates in rural areas due to less BTS/BTC deployment. On the other hand, GPS location techniques allow accurate location information to be derived with an acceptable response time. However, they require the antennae to be pointed to the sky, and thus work poorly in in-door environments. In order to further improve the accuracy and speed of GPS location techniques, some hybrid approaches have been proposed.

25.4.1 Cellular Location Techniques

Figure 25.4 [166] shows an example of the GSM architecture with a positioning server for supporting location tracking . In the figure, the Gateway Mobile Location Center (GMLC) is present to accept requests for location information of a user. The HLR of the user is contacted for getting the current Serving Mobile Location Center (SMLC) of the user. The SMLC controls a number of Location Measurement Units (LMUs) which measure radio information from the mobile station (MS). Based on the radio readings from various LMUs, the SMLC returns the result of the estimated location information of the MS.

There are a number of variations of the GSM-based approach. Some of them are summarized below:

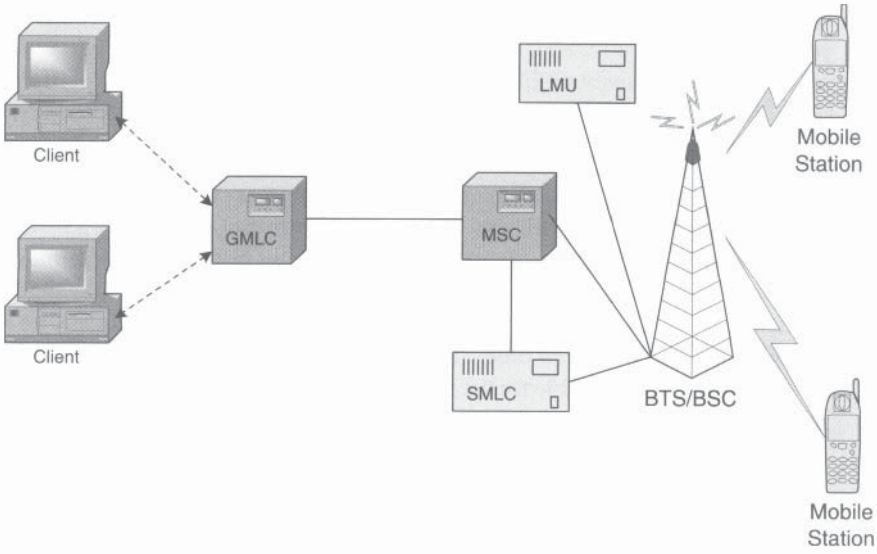


Figure 25.4 GSM architecture for positioning.

Cell of Origin (COO): This approach makes use of the cell concept of the deployment of BTS/BTCs in a mobile system, which usually involves overlapping regions. With the correct position information of each BTS/BTC, the location of the MS can then be estimated through the BTS/BTCs that the MS is currently contacting. The accuracy of this method is on the order of hundred to thousand meters, with a few seconds in response time. This approach is generally referred as a network-based approach since the location is estimated from the cellular network infrastructure.

Time Difference of Arrival (TDOA): This approach makes use of the time differences between the sending signal (uplink) from the MS to at least three BTS/BTCs equipped with LMUs. With the help of the LMUs, the TDOA of the signal from the MS can be estimated. The pairwise time differences and position information of BTS/BTCs can then be used to calculate the position of the MS, with the use of hyperbolic trilateration (discussed in the GPS positioning part in subsequent section). The accuracy of this method is within a hundred meters most of the time. However, it may not be possible to find three BTS/BSCs in some rural areas. Same as COO, this is also a kind of network-based approach.

Angle of Arrival (AOA): Similar to TDOA, this approach makes use of the sending signal (uplink) from the MS to at least three BTS/BTCs to estimate the location of the MS. However, instead of using the arrival time, the angle of arrival to

the base station is used. With this AOA information together with the position information of the BTS/BTCs, the position of the MS can be calculated. Same as TDOA, the accuracy of this method is within a hundred meters most of the time. This approach is also a kind of network-based approach.

Enhanced-Observed Time Differences (E-OTD): This approach is very similar to TDOA. However, instead of using uplink signal arrival times, downlink signal arrival times are used, and the location estimation is performed by the MS. However, this requires the MS to have accurate position information of the neighboring BTS/BTCs and receive synchronized signals from the them. Similar to TDOA and AOA, this approach can achieve an accuracy of within a hundred meters most of the time. It is regarded as a mobile-based approach in which the MS is responsible for the location calculation, or as a mobile-assisted approach in which the MS provides information for helping the network to calculate the estimated location.

Location Pattern Matching: In this approach, the frequency used by the MS and the multipath characteristics of the MS are recorded as "signature" for its location, which is stored in a database. Whenever the current location information of the MS is required, the information stored in the database is compared with the current information to estimate the location of the MS. The accuracy of this approach is around a hundred to several hundreds meters. It is regarded as a mobile-assisted approach.

25.4.2 GPS Location Techniques

Figure 25.4 [166] illustrates the idea of GPS location techniques. As shown in the figure, the GPS receiver should receive at least three satellite signals. By using an approach similar to TDOA, the distances between the GPS receiver and the satellites can be estimated. For example, in Figure 25.4 there are three satellites, and their estimated distances to the GPS receiver are $T1$, $T2$ and $T3$, respectively. With this distance information and the position information of the satellites, three circles are drawn as depicted in the figure. The intersection region of the three circles is the region that the GPS receiver is in. This approach is also known as trilateration, and it can achieve an accuracy of within tens of meters.

25.4.3 Hybrid Location Techniques

Figure 25.6 shows a hybrid location technique known as Assisted GPS (A-GPS) [166]. The location information is estimated by the location server in the cellular network. With the use of this information, the location server returns the satellites that are most suitable to the GPS receiver. As a result, the GPS receiver can measure the readings from these satellites directly without searching, which can reduce the initiation time. This approach can achieve an accuracy of within tens of meters.

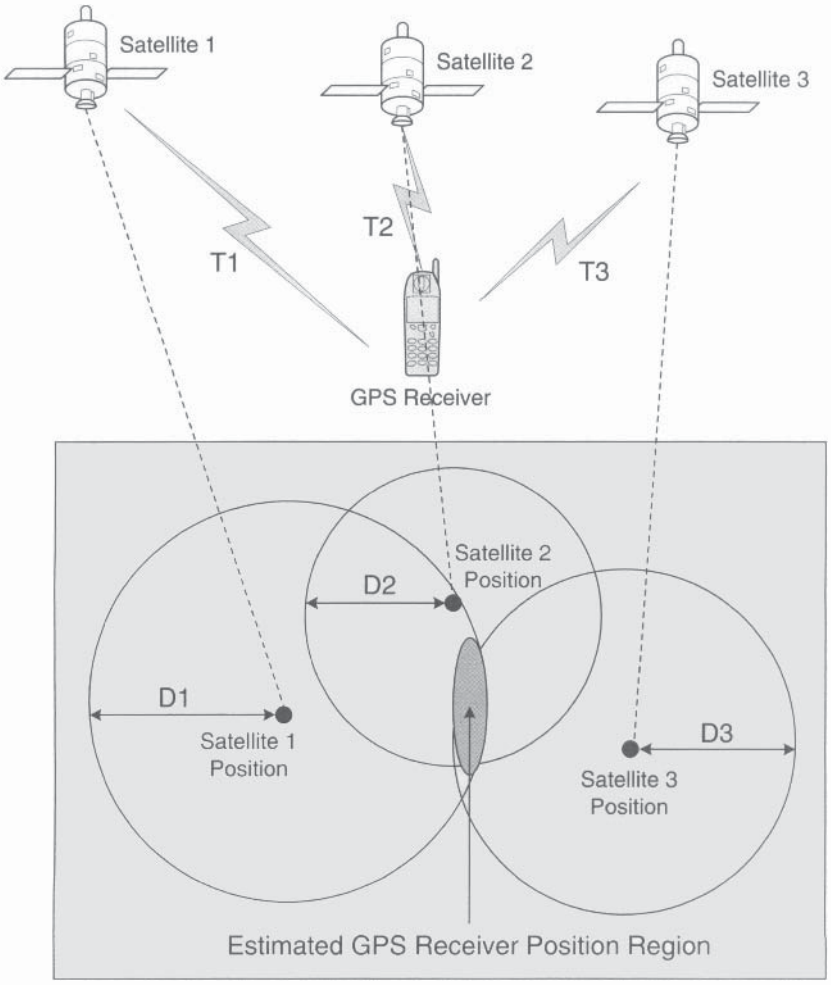


Figure 25.5 GPS architecture for positioning [166].

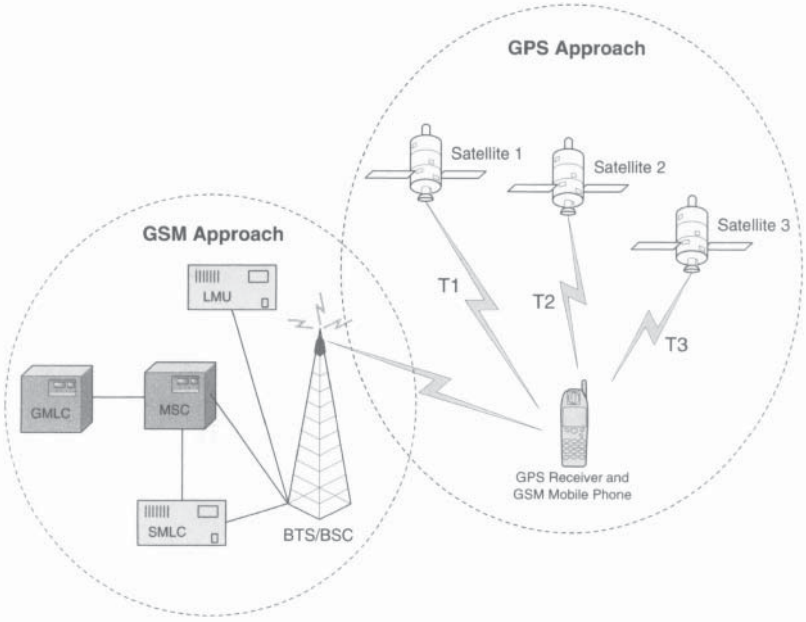


Figure 25.6 Hybrid architecture for positioning.

25.5 PRACTICAL ILLUSTRATIONS—THE CRICKET LOCATION SUPPORT SYSTEM

In previous sections, we have discussed mobile location services including in-vehicle mobile location service and personal mobile location service. In order to support these services, location information of users is always necessary, and the location information is derived with the help of spatial analysis and positioning techniques. In this section, we discus in detail a simple and practical location support system—Cricket [286], which that can help users to obtain their location information within a building.

The basic idea of Cricket is to use beacon devices deployed in various locations within a building to deliver certain information to the listener devices—mobile or static devices attached with a listening interface. The listener devices can then derive their own location from the information broadcast by the beacon devices. The information broadcast by the beacon devices is in fact some strings which can be the identity of the room obtained from the owner, some name servers for other devices to contact with, some authentication messages, etc.

Figure 25.7 illustrates the operation of the Cricket system. In the figure, the listener is located in Room C. It can receive beacon signals from beacon devices A, B and C in Rooms A, B and C, respectively. All the beacon signals carry the corresponding

location information of the beacon devices. Using the beacon signals, the listener then estimates its distances from the beacon devices. Together with the information conveyed in the beacon signals, the listener can figure out the beacon device which is the nearest to it, so as to make decision on its current location. For example, in Figure 25.7, the listener should have the shortest distance from beacon device C. Then it can know that it is currently located in Room C after analyzing the information carried by the corresponding beacon signal.

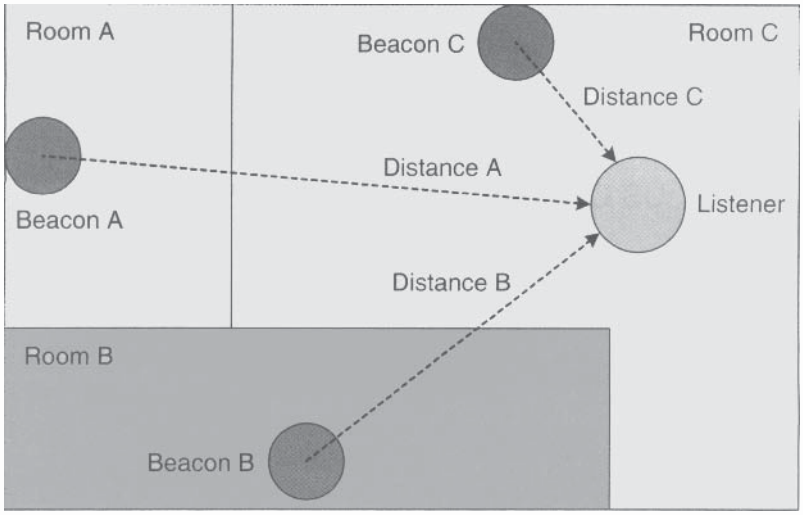


Figure 25.7 Usage scenario of the Cricket location support system.

However, in order to support the above-mentioned operation, a number of issues are required to be handled, e.g., estimation of distances from beacon signals, interferences of beacon devices, nearest beacon device not in current location, etc.

First of all, we will discuss the location estimation process of the Cricket system. Figure 25.8 illustrates the location estimation technique used in the Cricket system. Specifically, each beacon device is equipped with a ultrasonic (US) transmitter and a radio frequency (RF) transmitter. In each transmission, both RF signal and US signal are transmitted simultaneously. However, since the RF signal travels faster than the US signal, with the use of the time differences between the receipt of the first bit encoded in the RF signal and the receipt of the US signal, the distance between the listener and the beacon device can be estimated.

In fact, the absolute distance between the listener and the beacon device is not as important as the relative distances between the listener and other beacon devices. Therefore, the location estimation technique is enough to figure out which beacon device is the nearest from the listener. On the other hand, the coverage area of a

US signal is less than that of an RF signal, and thus with the help of the estimation technique the beacon device can effectively limit its own coverage area.

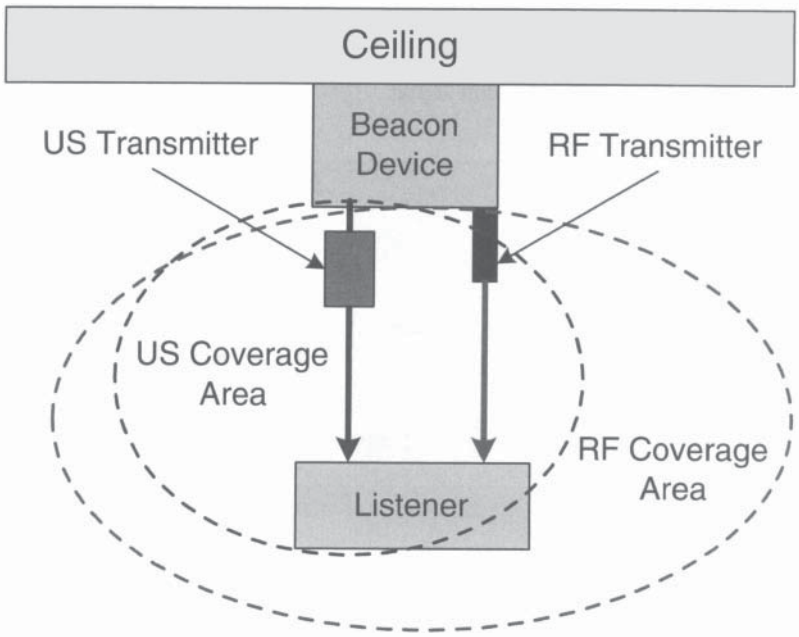


Figure 25.8 Location estimation of the Cricket location support system.

Although the design shown above can limit the coverage area of each beacon device and allow the listener to estimate its distances to beacon devices, it is still possible to have overlapping coverage areas by multiple devices, which can lead to collisions. In order to prevent collisions, a randomization approach is adopted. That is, beacon signals are transmitted randomly within a period of $[R1, R2]$ms. The exact values of $R1$ and $R2$ are chosen based on the situation of proximity, e.g., the bandwidth available, the size of messages delivered by each beacon, etc.

Besides collisions, it is possible for interference to occur between RF and US signals of neighboring beacon nodes. Figure 25.9 shows one of the possible scenarios. In the figure, due to signal attenuation, the listener fails to detect beacon device A's RF signal. However, it can detect beacon device B's RF signal. Before the arrival of beacon device B's US signal, it detects the arrival of beacon device A's US signal. In this case, the distance for beacon device A is induced, instead of that for beacon device B. There are also other possible interferences due to reflection of signals. In order to reduce interferences, proper system parameter values (e.g., size of message, transmission rate, propagation time, etc.) should be chosen. At the same time, some

statistical approaches like majority, minmean and minmode, can be applied to further estimate the nearest beacon device.

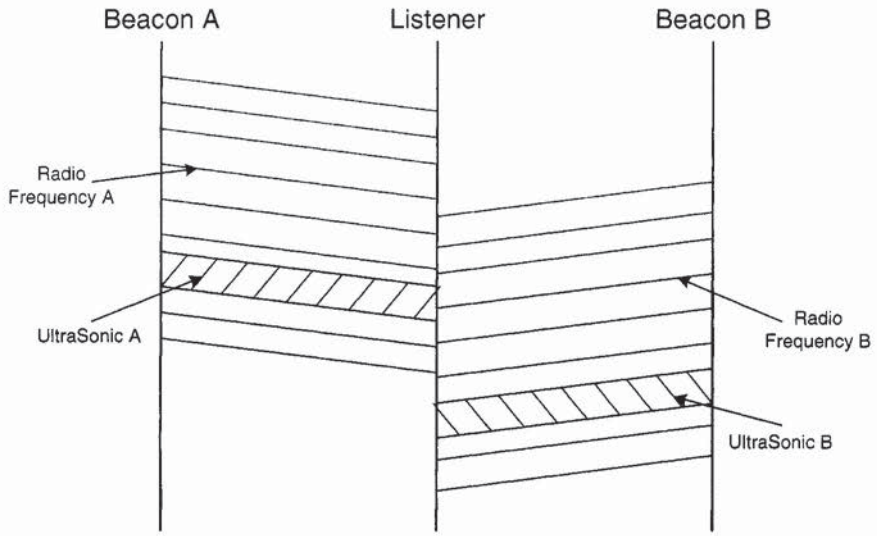


Figure 25.9 Cricket location support system interference problem.

In addition to interference and collision problems, the location of the deployment of beacon devices is also significant to the correctness of the identification of current location of the listener. The main reason is that, in some configuration, the nearest beacon device may not be located in the place where the listener is currently located in. In this case, the listener will get wrong current location information. One possible solution is illustrated in Figure 25.10 The deployment of beacon devices with approximately equal distances between both physical boundary (e.g., different rooms) and virtual boundary (e.g., different areas within the same room) can effectively remedy the problem.

In summary, the Cricket location support system combines several techniques like time differences between two different transmission media, randomized transmission of beacon signals, and certain positioning of beacon devices between the boundary of a location. With the help of these techniques, the system not only can deliver a correct location information up to room-sized level to the user devices with a high accuracy, but also becomes highly practical and can be implemented with low cost. On the other hand, the user devices do not need to transmit any information to the servers. They learn their own location information from the beacon devices instead. As a result, privacy of users can be kept.

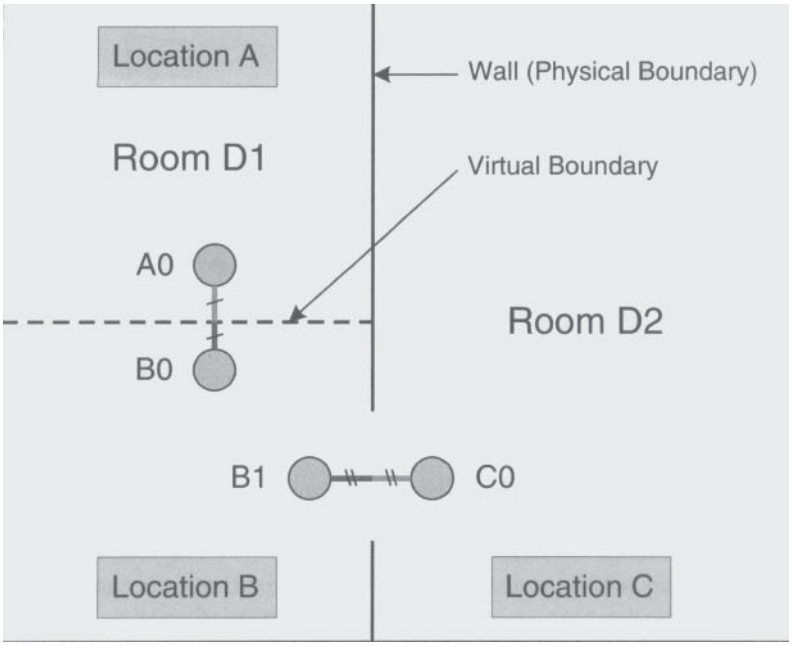


Figure 25.10 Cricket location support system beacon location.

25.6 SUMMARY

In this chapter, the service model of mobile location services has been discussed. Mobile location services can be divided into two main categories—in-vehicle and personal, which are respectively supported by two different operators, namely automobile manufacturers and mobile operators. These services rely on three main components, including the map database service, spatial analysis service and mobile positioning service.

Spatial analysis is required to convert specific locations into some transferable and calculable numbers. This is done by the geocoding process. Sometimes, it is required to have the best route from one location to another, and this requires some routing techniques. Moreover, the routing has to be shown on a map, which requires map generation. Besides some specific locations, users may be interested in some specific point of interests. As a result, point of interest search within a specific location becomes an important component. Finally, the point of interests or the routes within a routing are associated with some attributes. All the above aspects contribute to the spatial analysis service.

The other important issue of mobile location services is the estimation of current location of the user. Existing techniques can be divided into three main types—cellular location techniques, GPS location techniques, and the hybrid of the two types.

Finally, as practical illustrations, we have discussed the Cricket location support system, which is a simple system capable of detecting the location of a user up to room-sized accuracy.

PROBLEMS

25.1 Summarize the types of mobile location services. Discuss briefly which kind of services is more practical to our daily life.

25.2 Briefly discuss the components required in supporting mobile location services and their corresponding functions.

25.3 Discuss the components of the spatial analysis service and their corresponding functions.

25.4 Briefly explain how GPS system works and how GSM system works in tracking the location of a user.

This Page Intentionally Left Blank

CHAPTER 26

TRUST BOOTSTRAPPING IN WIRELESS SENSOR NETWORKS

If you would wish another to keep your secret, first keep it yourself.
—Seneca: Hippolytus, c. 60

26.1 INTRODUCTION

Establishing pairwise symmetric keys is a critical resource management issue in wireless sensor networks. Usually deployed in a hostile environment where malicious users or adversaries are bound to exist, wireless sensors are subject to a general attack model—a sensor node can be captured, re-programmed, and consequently exhibit arbitrary faulty behaviors. Thus, sensor key management is a challenging research issue, attracting a high level of interests in recent years. In general, a key management system works by first pre-allocating some keys to each sensor before deployment. After deployment, neighboring sensors can undergo a discovery process to set up shared keys for secure communications. An efficient key management scheme has to work under severe system constraints including limited memory storage

Wireless Internet and Mobile Computing. By Yu-Kwong Ricky Kwok and Vincent K. N. Lau
Copyright © 2007 John Wiley & Sons, Inc.

and communication overhead in each sensor. In this chapter we provide a detailed survey of state-of-the-art sensor key management techniques that have demonstrated a high degree of effectiveness.

With the ever increasing advancements in miniaturization of processing and communication hardware, wireless sensors and the networking of them have become popularly used in many applications [14]. However, many research challenges remain to be tackled. One of the most important challenges is the protection of sensor data communications. Indeed, wireless sensors are usually deployed in a hostile environment and the sensors have to operate with little or no infrastructure support. For example, sensors may be dropped from an airplane onto the battlefield for target recognition applications [179]. A key problem is that malicious users (e.g., the enemy) may eavesdrop the sensing data transmitted among the sensors. Furthermore, as the sensors are rarely tamper-resistant, some sensors may be captured by the adversary and re-programmed for eavesdropping or other malicious purposes [60, 318].

Interoperability:

A critical step toward protecting sensor data communications is the establishment of encryption keys among sensors so that they can set up secure communication links [47]. As indicated in by Hu *et al.* [157], setting up pairwise keys for secure communications is a necessary prelude to secure routing [182] in wireless ad hoc networks. However, with a small form factor and limited power supply, it is widely considered that a sensor device cannot employ sophisticated cryptographic technologies such as public key cryptosystems [60, 318]. Most notably, the Berkeley Mica Motes are very commonly used by the research community. In a typical mote, a 8-bit 4MHz Atmel ATmega 128L processor is used. Usually, it is equipped with 128K bytes of program memory and 4K bytes of SRAM. For communications, the sensor uses an ISM band RF module with a peak data rate of 40 Kbps and a maximum transmission range of roughly 30m. The operating system, TinyOS [209], is also small and works in an event driven manner. Finally, the wireless sensors are usually immobile after deployment. With such limited capabilities, symmetric cryptography is considered as the most practicable approach [47].

Setting up pairwise symmetric keys among communicating sensors is an important step for *bootstrapping* [47, 105] the mutual trust required for data exchange. The most obvious approach to enable key setup is to have a centralized key distribution center (KDC) [279, 326] which can allocate keys to sensors that need to communicate among each other. However, in a large scale deployment, it is infeasible to have such a KDC to communicate with all the sensors which may be widely dispersed over the field. Indeed, the KDC may be out of the transmission range of some remote sensors. Moreover, such a KDC is a single point of failure. Indeed, in a hostile environment (e.g., battlefield), it is unrealistic to assume the existence of such a KDC.

As to the topology of a wireless sensor network, there are two possible system architectures: (1) hierarchical sensor networks (see Figure 26.1); and (2) system architecture: fully distributed sensor networks (see Figure 26.2). In a hierarchical sensor network, some sensors are more powerful and thus, act as control nodes

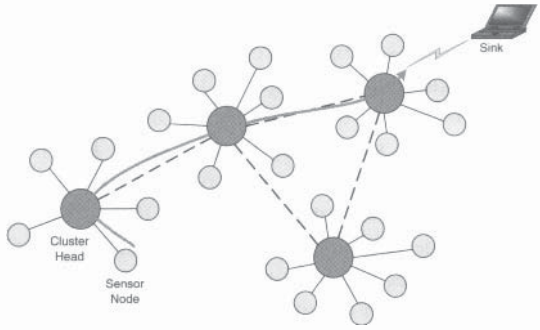


Figure 26.1 A hierarchical sensor network in which nodes are classified as cluster heads and worker sensor nodes.

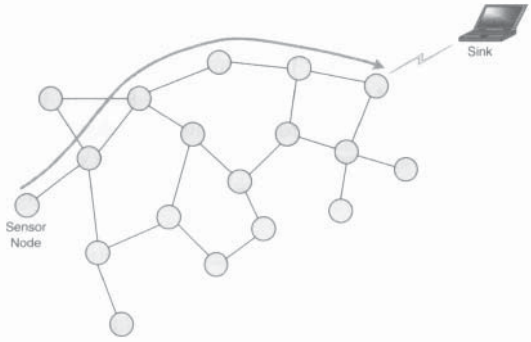


Figure 26.2 A fully distributed sensor network.

(similar to a base station in a cellular network) to provide resource management facilities including encryption keys distribution [47, 174, 374]. These controlling sensor devices are carefully deployed such that they are in the neighborhood of a certain number of "worker" sensors. Indeed, there are many key management schemes designed for hierarchical sensor networks [88, 174, 200, 268, 342, 374] by leveraging the heterogeneity of the sensors. However, in many practical situations [47, 105], it is also infeasible to assume a structured deployment of two types of sensors (e.g., again imagine the situation when a large number of sensors are dropped from the airplane onto a battlefield). In this chapter, we focus on fully distributed sensor networks where the sensors are just randomly deployed and it is very difficult, if not impossible, to control or predict the locations of the sensors after deployment. Hierarchical sensor networks are not considered in this chapter.

In a fully distributed sensor network, the framework that is widely employed in the literature for setting up pairwise encryption keys involve four phases: predistrib-

ution, discovery, path-key establishment, and revocation [105]. Predistribution is the process of storing keys in each sensor by a setup server *prior* to deployment. Usually, the key-sets of different sensors are not disjoint. Thus, after two sensors are deployed in nearby locations and they need to communicate, they can set up a secure key by checking whether they have a shared key in their sets. This is the key discovery phase. Obviously, it is possible that two sensors do not have any common key in their sets and they have to rely on other sensors to enable their communication. That is, they may need some intermediate sensors to set up a path in order to communicate. These intermediate sensors need to have shared keys with the two communicating sensors, respectively, so that each link is made secure by using a different key. This is called the path-key establishment phase. In a hostile environment, some sensors may be captured and the assigned keys are revealed to the adversary. These keys may be used in some communication links in the network and thus, have to be revoked. Key revocation is also a critical component in supporting a highly secure and robust network.

Under such a framework, there is large design space for key management schemes [47]. The major design constraints are the memory requirement and communication overhead involved. As detailed in this chapter, many state-of-the-art techniques are designed based on probabilistic key predistribution and discovery. Specifically, random graph theory [103] is widely used for determining the sizes of the predistributed key-sets in order to achieve a certain level of network connectivity.

Performance:

Wireless sensor networks have many challenging limitations compared to traditional Mobile Ad hoc Networks (MANETs). Specifically, the following constraints make the design of a high performance sensor key management scheme challenging.

1. *Energy limitation*: Energy is the main limitation of sensor nodes. Indeed, effective energy conservation scheme is vital for maximizing the lifetime of operation from a single node to the entire network. Typically, a recent battery-powered sensor node can run for a week (under full operation) to several months.

2. *Memory storage limitation*: Sensor device is very tiny and equipped with only a small amount of memory. Apart from storing keying materials, it is necessary to store the key management algorithm and other programs for operation. For example, a popular sensor device, Berkeley MICA2 mote, has 128 KBytes program memory only.

3. *Computational and transmission power limitation*: The small size of device and limited supply of power restrict the processing and communication capabilities of sensor nodes. As a result, complex asymmetric cryptographic algorithms cannot be implemented efficiently in sensor networks without modifications.

4. *Vulnerability to attacks*: As sensor networks may be deployed in hostile environments, nodes are exposed to physical attacks by potential adversary. In

most practical situations, it is possible to compromise and takes full control of the nodes without being detected.

5. *Lack of post-deployment knowledge*: Without deployment knowledge, the distribution of sensor nodes follows random scattering. This makes it hard to determine the network topology *a priori* so as to optimize the distribution of keys.

In summary, a major barrier for enhancing security in sensor networks is its limited computational and communication capabilities. These inherent properties make traditional protocols impractical. As a case in point, public key algorithms such as Diffie-Hellman key agreement as well as RSA signature techniques are too computational intensive so that it is undesirable to be implemented in scarce resource sensor nodes.

Consequently, performance of a key management scheme for wireless sensor networks is usually quantified by the amount of memory storage needed for storing keys in each sensor and the security level. Specifically, security level is measured by determining the number of compromised links when a certain number of sensor nodes are captured.

Moreover, security performance is also a core metric for measuring the degree of success of a key management scheme. Security performance is defined with respect to a certain attack model. In many existing key management schemes, a widely used attack model is that every sensor node is subject to node capturing. Specifically, if a sensor node is captured, keys in the captured node's key ring are assumed to be exposed completely. The compromised keys can then be used to attack other existing links in the network. Thus, to improve the network resilience, most key management schemes are designed to minimize the impact (i.e., the fraction of communication compromised) of node capturing.

This chapter is organized as follows. In Section 26.2, we survey general techniques proposed for the key predistribution phase. In Section 26.3, we describe some predistribution techniques that are specially designed by exploiting deployment knowledge. In Section 26.4, we introduce algorithms designed for the key discovery, establishment, and revocation processes after the sensors are deployed. We discuss some future research directions in Section 26.5. Section 26.6 provides some concluding remarks.

26.2 KEY PREDISTRIBUTION

A major goal of key predistribution [58, 93, 105, 220] is to judiciously allocate keys to the sensor nodes such that a high degree of connectivity of the network can be achieved without requiring a large key storage in each sensor.

Eschenauer and Gligor [105] performed pioneering work in sensor key predistribution in that they suggested a classical approach to allocating keys to sensors using a

probabilistic method. Their proposed scheme is simple to implement yet effective in a probabilistic sense. Specifically, before deployment, the centralized sensor controller (also called setup server in this chapter) generates a pool of P distinct keys. Then for each sensor, it is assigned a set of k keys that are randomly selected from the pool. As k is typically much smaller than P (e.g., $k = 75$ while $P = 10000$), the probability that two distinct sensors are assigned exactly the same set of keys is small. On the other hand, as described below, there is also a non-zero probability that two distinct sensors share at least one key.

After this predistribution phase, the sensors can be deployed onto the field. After deployment, if two sensors (presumably located close to each other) need to set up a secure communication link, they need to check whether they have a common key in their predistributed sets of keys. A simple yet practical method is to allow the two sensors to communicate with each other their list of key-IDs (not the keys themselves, of course) in plaintext. Even an eavesdropper can tap such lists, it cannot get to know the keys themselves. If there is no common key, then they need to set up a multi-hop path between them (albeit their physical locations may be close) via some other nearby sensors that share common keys with them.

Mathematical analysis in [105] indicates that the probability p that two sensors share at least one key is given by:

$$p = 1 - \frac{(1 - \frac{k}{P})^{2(P-k+\frac{1}{2})}}{(1 - \frac{2k}{P})^{(P-2k+\frac{1}{2})}}$$

Suppose $P = 10000$, in order to achieve a probability of $p = 0.5$, only $k = 75$ keys are needed to be stored at each sensor.

It is interesting to note that key revocation and hence rekeying (as triggered by the adversary's capturing of some sensor nodes) can be done by simply executing the key discovery phase again among those nodes that have their shared keys revoked.

Eschenauer and Gligor [105] performed detailed simulations with 10000 sensors. Their simulation results indicated that a value of k smaller than 40 resulted in a disconnected network. Furthermore, with $k = 75$, only half of the nearby sensor nodes were reachable over one single link. A majority of the remaining nodes needed three-hop paths for connections. Finally, out of the 10000 keys generated, only half of them were actually used.

Following up on Eschenauer and Gligor's pioneering work, Chan, Perrig, and Song [58] suggested three clever enhanced schemes which have also been highly cited in the literature. Their first proposed scheme, called q-composite key predistribution, is essentially the same as the original scheme by Eschenauer and Gligor, except that during link setup phase, at least q shared keys are required instead of just one. Specifically, suppose that two sensors have q' shared keys (where $q' \geq q$), then the link key K can be generated by:

$$K = \text{hash}(k_1, k_2, \ldots, k_{q'})$$

where hash is a certain one-way hash function taking all the q' shared keys as input. Analytical results indicated that the q-composite scheme provides greater resilience against node capturing attacks when the number of nodes captured is small. For example, when $q = 2$, suppose 50 nodes have been compromised, the percentage of additional communication links that can compromised is 4.74%. By contrast, this percentage would be 9.52% if the original scheme is used.

The second scheme proposed by Chan *et al.* is based on a multipath transmission of shared keys. The major goal of the multipath key scheme is to update the shared key between two sensors nodes after initial setup. Specifically, suppose nodes A and B want to update their shared key. To do this, they need to find j disjoint paths (where $j \geq 1$) between them. Afterward, node A generates j random values $v_1, \ldots, v_j$, each of which has the same size as the encryption key. Node A then sends these j values along different paths to B. Both nodes A and B can then update their shared key K as follows:

$$K' = K \oplus v_1 \oplus v_2 \oplus \ldots \oplus v_j$$

Consequently, unless a malicious intruder can tap the data on all the j different paths, the updated key can be kept secret.

The third scheme proposed by Chan *et al.* is motivated by the fact that the original Eschenauer-Gligor key-pool scheme cannot provide node-to-node authentication. Indeed, in the original key-pool scheme, two nodes A and B having some shared keys can set up a link between them using a particular shared key. But such a shared key can also be used by one of the two nodes, say A, to set up a different link with another node, say C. Thus, from a data encryption point of view, the two links (and hence the two nodes C and B) are not distinguishable. To address this authentication problem, Chan *et al.* proposed a node-oriented approach (whereas the original key-pool scheme is a key-oriented approach). Suppose that each sensor can store up to m keys and the probability of any two nodes in the network being able to set up a link is set as p (a system parameter). When the key predistribution phase starts, a total of $n = \frac{m}{p}$ unique node IDs are generated. This is the node ID space. Depending on the application, not all IDs will be used up to form a sensor network (i.e., the actual network size may be smaller than n). Each node ID is then associated with m randomly selected distinct node IDs. A distinct pairwise key is generated for each of these m IDs to be shared with the original node ID. Since the keys generated are closely associated with the node ID-pairs, when a link needs to be set up after deployment, node-to-node authentication can be easily carried out.

Liu, Ning, and Li [220, 221, 224, 222] extended the work of both Chan *et al.* [58] and Eschenauer *et al.* [105] by exploiting two new components: location information and bivariate polynomials. Firstly, Liu *et al.* extended the random pairwise key predistribution scheme [58] with the use of location information. Specifically, Liu *et al.* made an important observation in that the setup server, before deployment, should have pretty accurate knowledge about the locations of the sensors to be deployed. Thus, it makes sense for the setup server to arrange the random pairwise predistribution

in such a way that each sensor shares pairwise keys with c other sensors whose expected locations are closest to the expected location of c. Here, the magnitude of c is a system parameter depending on the memory constraint in each sensor. Of course, there is always a non-zero per-sensor deployment error, which is the deviation of the actual location from the expected location. Fortunately, Liu *et al.*'s simulation results indicated that with the same storage capacity, the location aware random pairwise predistribution achieved a higher probability to establish pairwise keys than the original version, especially when the deployment error was small.

Secondly, Liu *et al.* [224] made ingenious use of Blundo *et al.*'s polynomial based group key distribution algorithm [38] to design a general framework of key-pool based random key predistribution. Specifically, before deployment, the setup server first generates a bivariate t-degree polynomial $f(x, y)$ over a finite field F_q, where q is a large prime number (as governed by the security requirement in terms of bit length). The bivariate polynomial has a symmetric property, i.e., $f(x, y) = f(y, x)$. For each sensor i (here i is its unique ID), the setup server computes a "polynomial share" of $f(x, y)$, namely $f(i, y)$. Thus, for any other sensor j with $j \neq i$, sensor i can compute a shared key $f(i, j)$ by evaluating $f(i, y)$ at $y = j$. Symmetrically, sensor j can do the same by computing $f(j, i)$.

There are several important merits of this approach [38]. Consequently, each sensor only needs to store a t-degree polynomial, occupying $(t + 1) \log q$ space. A distinct advantage is that there is no need for the two sensors to communicate, apart from knowing each other's unique ID. Furthermore, it is shown [38] that this polynomial pool based technique is t-collusion resistant in that as long as the number of malicious sensors (forming a collusion) is smaller than t, a shared key among two legitimate nodes cannot be broken.

It is interesting to note that if $t = 0$, then the scheme degenerates to the key pool scheme proposed by Eschenauer and Gligor [105].

Observing that the polynomial based predistribution scheme can deter attacks from fewer than t malicious sensors with t limited by the storage capacity of each sensor, Liu *et al.* [224] improved the scheme by including the location awareness concept. Specifically, the target deployment field is partitioned into square cells as shown in Figure 26.3. As can be seen, sensor u is expected to be deployed inside cell $C_{2,2}$. Thus, cells $C_{2,1}$, $C_{1,2}$, $C_{2,3}$, and $C_{3,2}$ are adjacent to u's home cell. With such knowledge, the setup server then allocates u with the polynomial shares: $f_{2,2}(u, y)$, $f_{2,1}(u, y)$, $f_{1,2}(u, y)$, $f_{2,3}(u, y)$, and $f_{3,2}(u, y)$. Subsequently, sensors deployed in these adjacent cells can easily share keys with sensor u.

As reported in [224], the above two-dimensional partitioning of the field into square cells can be extended to n-dimensional hypercube. The advantage of a higher dimension partitioning is the increased security level, at the expense of a higher storage overhead (storing more polynomial shares at each sensor).

Zhou *et al.* [372] proposed a link-layer key establishment scheme by extending Liu, Ning, and Li's bivariate polynomial-pool scheme [224] with a hexagonal-grid partitioning. The resultant extended scheme has lower communication overhead and

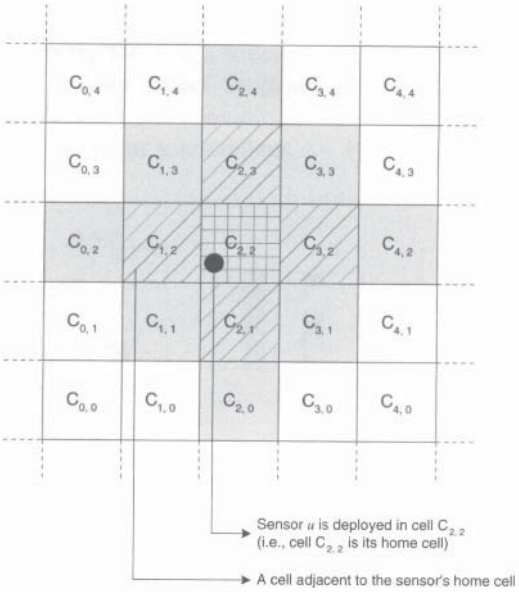


Figure 26.3 The sensor network is partitioned into square shape cells [221].

memory cost. Specifically, suppose D is the inter-cell distance in the grid and ρ is the deployment density, then the number of nodes storing shares of one polynomial in Liu, Ning, and Li's scheme is $5\rho D^2$ whereas in the hexagonal scheme it is only $\sqrt{3}\rho D^2$. Consequently, the memory costs of the two schemes are given by $5(5\rho D^2 + 1)$ and $6(\sqrt{3}\rho D^2 + 1)$, respectively [372].

Du *et al.* [95, 94] exploited the nice properties of a classical key predistribution method proposed by Blom [36]. Specifically, the proposed scheme has an important thresholding feature—as long as no more than λ nodes are compromised, all communication links of uncompromised nodes remain secure.

In Blom's scheme, first of all a $(\lambda + 1) \times N$ matrix G over a finite field GF(q), where N is the size of the network and q is a prime number larger than N. The matrix G is made public. In the key generation process, the setup server produces a random $(\lambda+1) \times (\lambda+1)$ symmetric matrix D over GF(q). It also determines an $N \times (\lambda+1)$ matrix $A = (D \times G)^T$. The matrix D is a secret. Now, as D is symmetric, we have:

$$A \times G = (D \times G)^T \times G = G^T \times D^T \times G = G^T \times D \times G = (A \times G)^T$$

Thus, $A \times G$ is also a symmetric matrix. Now, we denote $A \times G$ by K. Then, we have $K_{ij} = K_{ji}$ (note the similarity between this and the bivariate polynomials used in Liu *et al.*'s scheme), where K_{ij} is an element of the matrix K. Consequently, K_{ij} and K_{ji} can be used as the pairwise key of a pair of sensor nodes. Specifically, the

setup server can assign the following items to sensor node k: the k-th row of matrix A and the k-th column of matrix G. Because G is made public, two sensor nodes can exchange their columns of G to identify each other's index. Blom's scheme, just like the bivariate polynomial scheme by Blundo *et al.*, is λ-collusion resilient. This process is illustrated in Figure 26.4. A Vandermonde matrix [36] can be used as the matrix G.

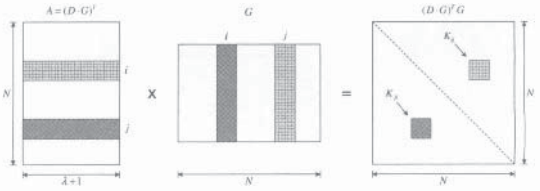


Figure 26.4 Generating sensor pairwise keys based on symmetric matrices [95].

Du *et al.* [95] extended Blom's scheme for sensor networks in the following manner. Du *et al.*'s scheme is called multiple-space key predistribution algorithm. Instead of using one single matrix D, Du *et al.*'s scheme generates ω random symmetric distinct matrices $D_1, \ldots, D_\omega$. Accordingly, there are ω different matrices $A_i = (D_i \times G)^T$. Now, for each sensor, the setup server randomly assigns to it τ (where $2 \leq \tau < \omega$) distinct key spaces from the ω possible choices. Thus, if two sensors share some common key spaces, they can set up shared keys using Blom's scheme.

Du *et al.*'s simulation results indicated that their scheme outperformed Eschenauer-Gligor's and Chan *et al.*'s random key-pool schemes. As a proof of concept, Du *et al.* also implemented their scheme in a real system using MICAz sensor nodes. In their system, $\lambda = 50$ and the matrices are generated over GF(2^{16}). The time for computing a 64-bit key was only 25.67 msec.

Lee and Stinson [201] also proposed a similar scheme as that of Du *et al.* [95] in that the Blom's predistribution method was also used.

Hwang and Kim [161] made an important observation: in practice, the node degree cannot be too large, e.g., 6–8 is optimal for practical considerations (i.e., traffic volume, energy, etc.) instead of the around 20 or more as implied by the random key-pool based or random polynomial-based schemes described above. Indeed, a larger node degree would consume more power in a sensor node. Thus, by using the "giant component" concept in random graph theory [103], Hwang and Kim proposed a slightly modified random key predistribution scheme which is based on judicious adjustment of transmission powers. The resulting network might be partitioned in that a small groups of nodes might be isolated whereas the remaining nodes are connected among each other. But Hwang and Kim argued that this is not a bad result because in practice the isolated nodes can subsequently set up links or paths with the giant component by increasing their transmission powers.

Recently, Traynor *et al.* [335, 335] extended the basic random key-pool predistribution scheme to a hierarchical sensor network. Specifically, they made an observation that the purely peer-to-peer communications among sensors are unrealistic in a practical sensor network. Indeed, they argued that in a real application, usually there are some more powerful sensors acting as control devices. These control devices can then accommodate more keys in the predistribution phase. Thus, accordingly, they proposed an unbalanced random key-pool predistribution scheme, which they found that can give a higher degree of security without increasing the memory cost for the "worker" sensor nodes.

Eltoweissy *et al.* [102] developed a protocol for dynamic re-keying in the post-deployment phase. Under the long life cycle assumption, re-keying is necessary for handling the addition or revocation of nodes. Re-keying can also prevent further compromising of rest of network against node capture. When some nodes are suspected to be compromised, the base station sends re-keying instruction to cluster controllers to trigger the corresponding re-keying operations. The drawback is that their approach requires coordination from a base station and cluster controllers.

Jolly *et al.* [174] proposed a key management protocol based on symmetric keying, primarily for hierarchical sensor networks. It is observed that sensor-to-sensor secure communication is not necessary in many applications. Thus, very few keys (typically two) are enough to establish secure connections between nodes and base station as well as cluster gateways. Their objective is to provide a cost-effective yet sufficient keying infrastructure for security of sensor networks. However, their design requires a centralized key server and the rekeying process is inefficient due to large amount of message exchanges during key renewals.

Table 26.1 gives a qualitative comparison of representative approaches discussed above.

26.3 KEY PREDISTRIBUTION WITH DEPLOYMENT KNOWLEDGE

If the structure of the network after deployment can be predicted or even controlled, predistribution is more efficient because keys can be judiciously allocated to neighboring sensor nodes.

Du *et al.* [92, 94] observed that long distance communications among sensors are not common in many practical applications. Thus, shared keys need only be set up among nodes that are physically close to each other after deployment. Du *et al.* enhanced the original Eschenauer and Gligor's approach by providing a guaranteed probability p of two physically neighboring nodes can set up a shared key. Specifically, N sensor nodes to be deployed are first divided into $t \times n$ groups so that each group $G_{i,j}$ ($i = 1, \ldots, t$ and $j = 1, \ldots, n$) is of equal size. The target deployment location of group $G_{i,j}$ is denoted as (x_i, y_i). For each sensor k in group $G_{i,j}$, its actual deployment location follows the probability density function (pdf) $f(x - x_i, y - y_i)$ where $f(x, y)$ is a Gaussian distribution.

Table 26.1 Comparison of popular key management schemes in wireless sensor networks

	Chan [58]	Du [92]	Eschenauer [105]	Eltoweissy [102]	Jolly [174]	Liu [224]
Description	random pairwise scheme	deployment knowledge scheme	basic scheme	dynamic scheme	low energy scheme	polynomial pool-based scheme
Connectivity	average	high	average	average	N/A (only node to gateways / based station)	average
Resilience	perfect against node capture	impact of attacks is localized	acceptable	good due to controllable rekeying	weak	high due to t-collusion resistant
Scalability	weak as max. network size is limited	simple node addition and revocation	simple node addition and revocation	efficient with help of centralized server	high due to simple rekeying mechanism	simple node addition and revocation
Energy consumption	low	average	average	average	very low	high
Storage cost	high	low	average	low	very low	high
Management	distributed	distributed	distributed	centralized	centralized	distributed
Knowledge assumption	none	location information is available	none	operating in long-lived network	many-to-one communication	none

Now, a set of keys $S_{i,j}$ is allocated to group $G_{i,j}$ during the key predistribution phase. Each set of keys is of the same size, denoted by $|S_c|$. The key feature of Du *et al.*'s scheme is that if two groups' target deployment locations are far away from each other, then the amount of overlap in their respective key sets would be close to zero.

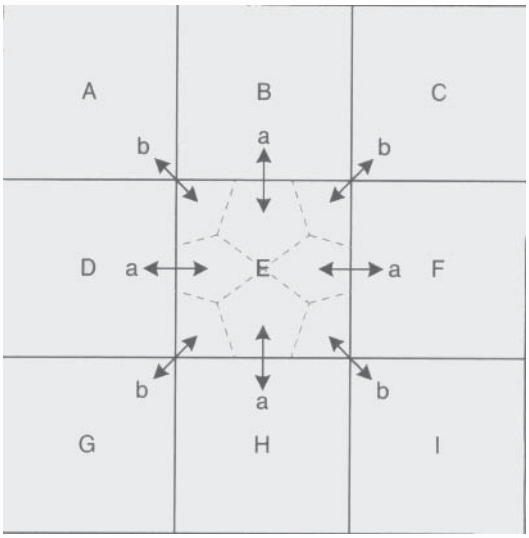


Figure 26.5 A grid based deployment structure illustrating the sharing of keys between neighboring groups [92].

Assuming a grid deployment structure as shown in Figure 26.5, Du *et al.*'s scheme has the following features:

- two horizontally or vertically neighboring groups share exactly $a|S_c|$ keys with $0 \leq a \leq 0.25$;
- two diagonally neighboring groups share exactly $b|S_c|$ keys with $0 \leq b \leq 0.25$ and we have: $4a + 4b = 1$;
- two non-neighboring groups would not share any key.

Figure 26.6 shows a simple example with $t = n = 4$.

With these arrangements, it is shown [92] that the key pool size is given by:

$$|S_c| = \frac{|S|}{tn - (2tn - t - n)a - 2(tn - t - n + 1)b}$$

where $|S|$ is the total number of distinct keys in the system. For example, suppose $|S| = 100000$, $t = n = 10$, $a = 0.167$, and $b = 0.083$, then we have $|S_c| = 1770$.

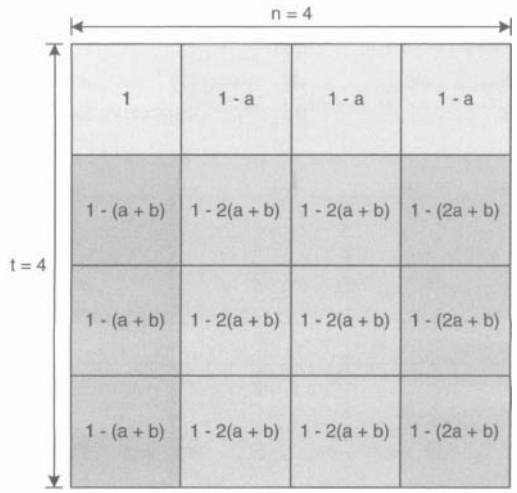


Figure 26.6 An illustrating example ($t = n = 4$) of the sharing among neighboring groups [92].

Huang *et al.* [159] suggested a similar scheme in that it was also based on a grid-group based deployment structure. The major difference is that Huang *et al*'s scheme is based on Blom's key predistribution method [36]. Furthermore, as shown by Huang *et al.*'s detailed analysis, their proposed deployment-knowledge based scheme can effectively tackle some practical malicious attacks such as selective node capturing and node fabrication.

Yu and Guan [367] suggested a predistribution scheme also based on Blom's key allocation method. Their proposed scheme also utilizes the deployment knowledge where the sensor field is divided into hexagonal cells. Their findings indicated that by carefully deciding on the size of the grids, the number of potential neighbors can be significantly reduced. Thus, the resources consumption can be much more efficient.

Ito *et al.* [163] suggested an improved scheme in which they remove Du *et al.*'s assumption [92] that a rectangular grid based deployment structure is used. Essentially, instead of assuming a probability density function for a whole group, Ito *et al.* used a per-node probability density function. Consequently, the key predistribution process is carried on a node by node basis.

Liu, Ning, and Du [223] suggested a scheme similar to that of Du *et al.* [92] in that the sensors are deployed in groups and members of each group are to be deployed to nearby locations. One major distinctive feature is that the target deployment locations (x, y) are not specified as in Du *et al.*'s scheme. As such, Liu *et al.*'s scheme is more flexible. An implication is that the predistribution process cannot leverage on the relationship among the deployment locations, which are unknown. Thus, Liu *et*

al. suggested a novel grouping structure with groups G_i and cross-groups G'_i. An example is shown in Figure 26.7. As can be seen, nodes 1, 2, and 3 belong to the group G_1, whereas nodes 1, 4, 7, and 10 belong to the cross-group G'_1.

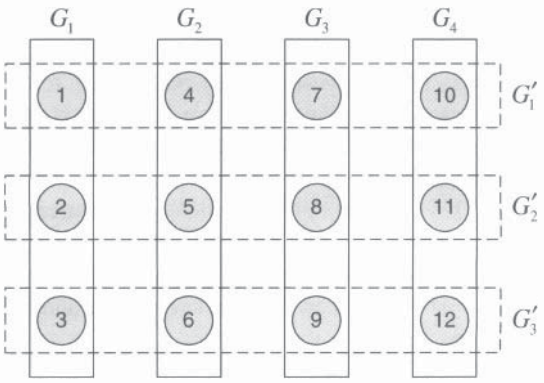


Figure 26.7 Construction of groups and cross-groups [223].

With such a overlapping grouping structure, key predistribution is carried out for both groups and cross-groups. That is, members in each group can set up a key shared key. Similarly, members in each cross-group can also set up a direct shared key. Among nodes that are deployed at physically close locations but do not belong to the same group or cross-group, some "bridge" nodes in a group or cross-group have to be used. For example, if node 1 wants to set up a pairwise key with node 12, the pair could try to use the bridge (1, 10). If this does not work, they can try another bridge (3, 12), and so on.

Liu, Rivera, and Cheng [225] also suggested a deployment knowledge based predistribution method. In their proposed scheme, sensors find out their designated roles and configures themselves automatically based on a pure localized algorithm. Under a hierarchical role-play structure, only service sensors are responsible for key space generation and distribution, with the goal of conserving the energy of other "worker sensors". The core key distribution scheme is, however, also based on bivariate polynomials.

Delgosha and Fekri [84] recently proposed a clever extension to the random bivariate polynomial predistribution scheme. Specifically, each sensor node is assigned a unique ID which is an n-tuple of integers. Each ID then maps to a set of n polynomials that are n-variate to other nodes. As the IDs are organized as the vertices of a hypercube, every two nodes with their IDs differed by a Hamming distance of one can then set up $n - 1$ shared keys. The link key can then be generated by combining these $n - 1$ keys.

26.4 KEY ESTABLISHMENT

Key establishment is a term we use in a narrow sense—the discovery and set up of a shared key between two sensors [338, 160]. As discussed above, after deployment, each sensor has already stored some keys in its memory system. The establishment phase involves the discovery of common keys in the key-sets of two communicating sensors.

Di Pietro *et al.* [281, 280] suggested a key-pool based scheme for setting up pairwise keys among sensors. Specifically, before deployment, a pool of P random keys is generated. Each key is associated with a unique index. Each sensor a is then assigned a set V_a of k keys that are randomly selected from the pool of P keys. The indices of the assigned keys can be deterministically computed by using the ID of the sensor a. Thus, similar to the classical approach suggested by Eschenauer and Gligor [105], there is a non-zero probability that two sensors share some common keys. Precisely, the probability that two distinct sensors a and b share at least one common key is given by:

$$1 - \frac{{}_PC_{P-k}}{{}_PC_k}$$

After deployment, when two sensor nodes a and b want to set up a secure point-to-point communication channel, they independently generate a shared key as follows:

$$\mathcal{K}_{a,b} = \bigoplus_{v_b^i \in V_a} v_b^i$$

That is, the new share key is generated by taking the XOR over all the shared common keys from the key pool.

Referred to as Direct Protocol by Di Pietro *et al.*, it provides only a fixed level of security depending on the system parameters P and k. In view of this, Di Pietro *et al.* suggested an extended version called Cooperative Protocol which works by involving a set of other cooperative sensors, $C = \{c_1, \ldots, c_m\}$ such that $a, b \notin C$, when two sensor nodes a and b want to set up a new shared key. Specifically, the shared key is computed as follows:

$$\mathcal{K}_{a,b}^C = \mathcal{K}_{a,b} \oplus \left(\bigoplus_{\forall c \in C} \text{HMAC}(\text{ID}_a, \mathcal{K}_{c,b}) \right)$$

where $\mathcal{K}_{a,b}$ and $\mathcal{K}_{c,b}$ are generated by using the original Direct Protocol, and HMAC is a one-way hash function. Suppose that sensor a is the one who initiates the key setup procedure, it is responsible for selecting the members of the set C. Then a informs b about the composition of C such that every member $c \in C$ can then set up a pairwise key $\mathcal{K}_{a,c}$ with a and subsequently a pairwise key $\mathcal{K}_{c,b}$ with b. The pairwise key $\mathcal{K}_{a,c}$ is used by a to send the cooperative request to each sensor c for computing the one-way hash function to be sent to b. The security of Cooperative Protocol increases with m.

Chan and Perrig [59] proposed an interesting scheme for key establishment between pairs of sensor nodes. Their proposed scheme is called PIKE (peer intermediaries for key establishment), which works by transmitting a new key via trusted intermediary sensor nodes. The PIKE system critically relies on a regular deployment and logical layout structure, supported by a certain global node-addressing facility. The major distinctive salient feature of PIKE is that the memory cost in each sensor node is $O(\sqrt{n})$, where n is the number of sensor nodes deployed in the system. This represents a significant improvement over many existing schemes which usually have a $O(n)$ memory or communication cost per sensor node.

To illustrate how PIKE works, consider a sensor network with n nodes deployed. We assume that n is a perfect square. Each sensor is then assigned a unique ID (x, y), where $x, y \in \{0, 1, 2, \ldots, \sqrt{n} - 1\}$. Before deployment, each sensor node (x, y) is then allocated $2(\sqrt{n} - 1)$ distinct secret keys which are respectively shared with each member of the two sets of sensor nodes:

$$H = \{(i, y) : i \in [0..\sqrt{n} - 1]\}$$

and

$$V = \{(x, j) : j \in [0..\sqrt{n} - 1]\}$$

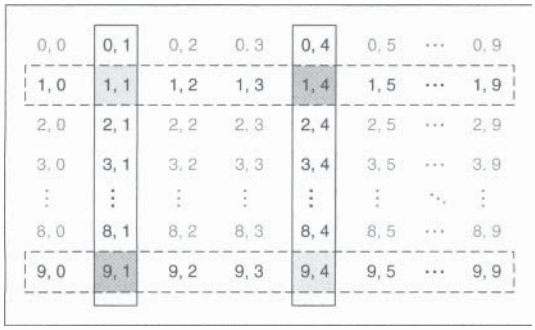


Figure 26.8 Initial key sharing among nodes in both the horizontal and vertical dimensions [59].

Figure 26.8 illustrates the case of $n = 100$. We can see that with the logical layout shown, each pair of nodes in the vertical set $(i, 1)$, where $i = 0, \ldots, 9$, share a distinct secret key which is allocated before deployment. Similarly, each pair of nodes in the vertical set $(i, 4)$ also share a distinct secret key. The same arrangement is carried out for all other vertical sets as well as all horizontal sets, such as $(1, i)$ and $(9, i)$. Consequently, for two nodes that do not have a shared key, they can establish a new shared key via two intermediary nodes. For example, nodes 14 and 91 do not have an *a priori* shared key so that they cannot perform one-to-one secure communications. Now, by transmitting encrypted messages via nodes 11 or 94, nodes 14 and 91 can establish a new key between them.

From the two candidate intermediary nodes, one of them is chosen based on a certain heuristic metric. For example, the candidate that is physically closer to the pair of nodes can be selected. Other metrics such as delay, throughput, or traffic load can also be used.

One critical feature of PIKE is that it relies on a global node-addressing facility which allows a node to locate and reach a node with a shared key. For example, such a facility is required to allow node 14 to locate and reach node 11. This is important because the logical layout has no bearing on the actual physical locations of nodes. Specifically, a geographical routing scheme such as GPSR [184] or a geographic hash table (GHT) based addressing service [297] is needed.

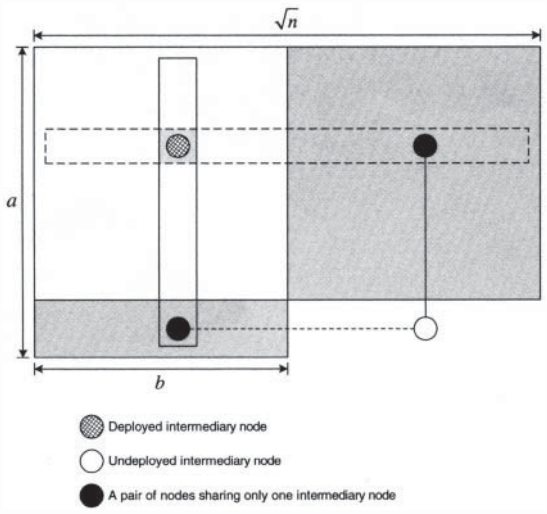


Figure 26.9 A partially deployed sensor network [59].

Furthermore, the sensor nodes need to be deployed in a specific order so that the subsequent establishment of new keys can be carried out with at least one intermediary node to help. Specifically, the sensor nodes should be deployed in the following order: $(0,0), (0,1), (0,2), \ldots, (0,\sqrt{n}), (1,0), (1,1), (1,2), \ldots, (1,\sqrt{n})$, and so on. Figure 26.9 depicts the situation where the last sensor node that has been deployed is $(a-1, b-1)$. Here, it can be shown [59] that the probability that only one intermediary node is available for a pair of nodes (i.e., the two nodes in the shaded regions) wanting to establish a new key is given by:

$$\frac{1}{4(a-1)}$$

Thus, suppose $n = 4900$ and 2000 nodes have already been deployed, then the probability of having only one intermediary node is only around 0.01.

In terms of memory storage requirement, the PIKE scheme is highly efficient. Indeed, based on the description above, each sensor node needs to store $2(\sqrt{n}-1)$ *a priori* keys. This can be further reduced to $(\sqrt{n}-1)$ by using a dynamic key generation method [59].

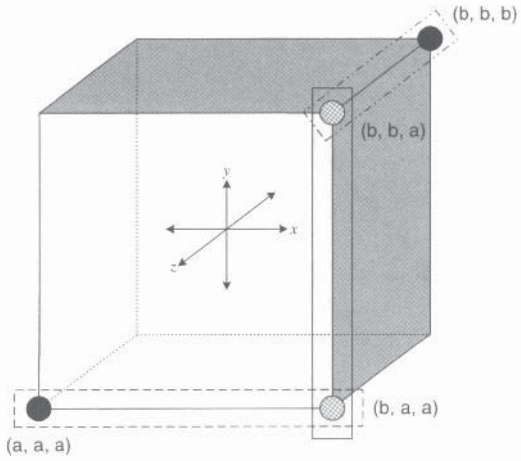


Figure 26.10 A three-dimensional layout structure [59].

There are a couple of useful extensions of the PIKE scheme. Firstly, the 2-D layout structure can be extended to 3-D as shown in Figure 26.10. With three dimensions, each pair of nodes can establish a new key with the help from two intermediary nodes. As each node now shares *a priori* keys with $3(\sqrt[3]{n}-1)$ other nodes, the memory requirement at each node becomes $O(\sqrt[3]{n})$ instead of $O(\sqrt{n})$. The second extension is that each sensor node can share pairwise keys with other nodes on the diagonals, as shown in Figure 26.11. This arrangement increases the number of possible intermediary nodes for any pair of nodes to choose from.

Chan and Perrig [59] conducted a detailed and extensive simulation study in which two versions of PIKE, namely PIKE-2D (with a 2-D layout) and PIKE-3D (with a 3-D layout), are compared against a KDC based approach and a probabilistic approach [93]. Both versions of PIKE produced performance close to that of the KDC approach which represented as an ideal situation. Furthermore, both PIKE-2D and PIKE-3D required significantly less memory and communication overhead compared with the probabilistic approach. PIKE-2D is more resilient against node-capturing attacks while PIKE-3D is more efficient in terms of memory and communication overhead.

Blaßand Zitterbart [35] suggested a novel approach in establishing a shared key between two sensor nodes that belong to the same data aggregation tree. Motivated by the observation that the key-pool or polynomial-pool based schemes [224, 372] require $O(n)$ memory storage, Blaßand Zitterbart argued that key establishment is needed only between nodes on the same data aggregation tree. For example, as shown

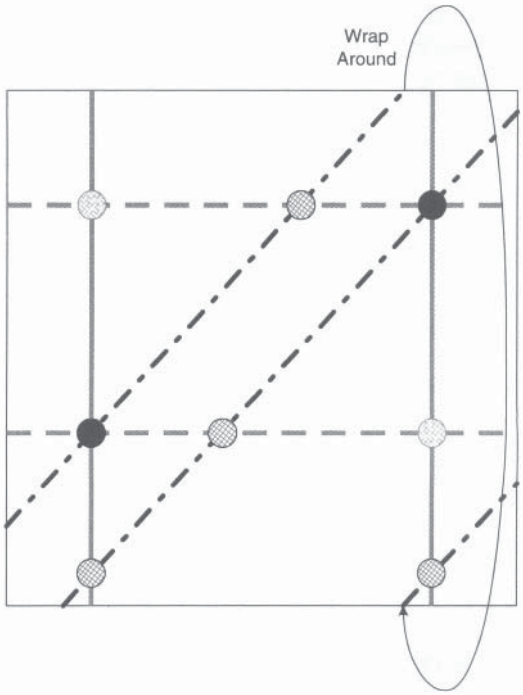


Figure 26.11 Each sensor node shares *a priori* keys with diagonal nodes [59].

in Figure 26.12, the sensor node a only needs to share keys with its ancestors (e.g., x and z) but not other nodes such as b, c, and d. Consequently, the memory cost required is only $O(\log n)$ because the height of the tree is $O(\log n)$.

Specifically, Blaßand Zitterbart argued that for data aggregation applications like the temperature measuring task as depicted in Figure 26.12, sensor devices do not need to communicate in an arbitrary manner to other sensor devices. In this temperature measuring example, it is reasonable to assume that the data sink (i.e., the computer) is interested only in the average temperature of the two rooms. As such, sensor nodes x, y, and z only need to perform the averaging computation. Indeed, there is no need for any communication to happen between a and b, or between c and d.

However, to verify the integrity of the forwarded data (e.g., the average temperature of x and y), an ancestor sensor node in the aggregation tree needs to communicate directly with some nodes on the aggregation paths. For example, the data sink might need to check that the average value as forwarded by z is correct. In order to carry out this checking, the sink needs to communicate with x and y directly. Similarly, z might also need to communicate with a and b directly. The implication of the need

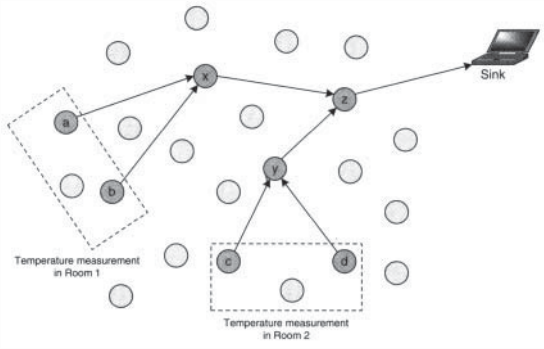


Figure 26.12 Temperature data aggregation tree [35].

for such integrity verification is that a sensor node needs to share a key with each ancestor node on its aggregation path toward the sink.

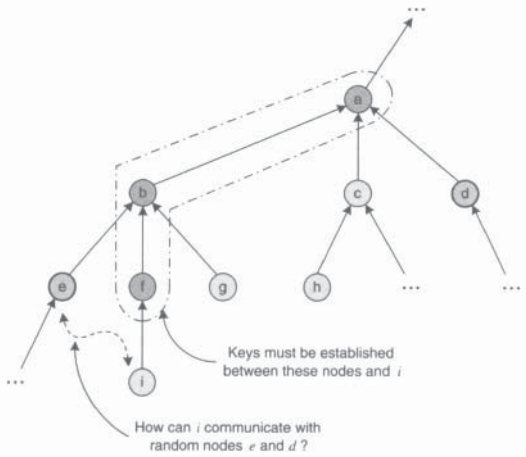


Figure 26.13 A new sensor node i joins the data aggregation tree [35].

The pairwise key establishment protocol proposed by Blaßand Zitterbart can be described *inductively* as follows. Figure 26.13 depicts the situation where a partial aggregation tree has been formed (i.e., the basis of the inductive process). Now, sensor node i is to be added to the aggregation tree with f as its parent. The setup server, called Master Device (MD) in [35], assigns i the following items:

- $K_{\mathrm{MD},i}$: a secret key between i and the MD;

- the IDs of two randomly chosen nodes e and d, together with two secret keys $K_{e,i}$ and $K_{d,i}$ which can be used by i to communicate with these two nodes, respectively; and
- two tickets: $T_e = E_{K_{\text{MD},e}}(i, \text{"is legal player"}, K_{e,i})$ and $T_d = E_{K_{\text{MD,d}}(i,\text{"is legal player"},K_{d,i})}$.

With the above items, node i can then establish a shared key with node f with the help from e and d, based on an operation called "key splitting". Specifically, a key K can be split into two *key shares* K_1 and K_2 as follows:

$$\begin{aligned} K_1 &= r \\ K_2 &= K \oplus r \end{aligned}$$

where r is a randomly generated number. With key splitting, the original key can be restored ($K = K_1 \oplus K_2$) if and only if both K_1 and K_2 are available.

Now, i generates a shared key $K_{i,f}$ to be used with node f. Obviously, i does not want to reveal this key to any other nodes including e and d. Thus, i splits $K_{i,f}$ into two key shares K_1 and K_2 first. To start communicating with e and d, node i needs to send them the two tickets T_e and T_d, respectively. Given these tickets, nodes e and d trust i and communicate with it. Afterward, i sends the following "request to forward K_1 to f" to node e:

$$C_1 = (i, E_{K_{e,i}}(i, K_1, f))$$

Similarly, i also sends the following forwarding request to d:

$$C_2 = (i, E_{K_{d,i}}(i, K_2, f))$$

Up to this point, if e already possesses a shared key with f (i.e., a key $K_{e,f}$ exists), then e can simply send the following to f:

$$\gamma_1 = (e, E_{K_{e,f}}(i, K_1))$$

Unfortunately, in this example as depicted in Figure 26.13, there is no such shared key between e and f (as well as between d and f) because they are not on the same aggregation path.

In this situation, e (as well as d) iteratively inquires its ancestor nodes until it can find an ancestor node that is also an ancestor node of f. Such an ancestor must exist because ultimately the sink is the ancestor of every node in the tree. Suppose that e finds that b and a to be such suitable ancestors. Similarly, d finds that a and some ancestor of a are such suitable ancestors. Consequently, e splits K_1 (from i) into two key shares K_1^1 and K_1^2 and then sends the following two requests to b and a, respectively:

$$\begin{aligned} C_1^1 &= (e, E_{K_{e,b}}(f, K_1^1)) \\ C_1^2 &= (e, E_{K_{c,a}}(f, K_1^2)) \end{aligned}$$

Such forwarding actions are illustrated in Figure 26.14.

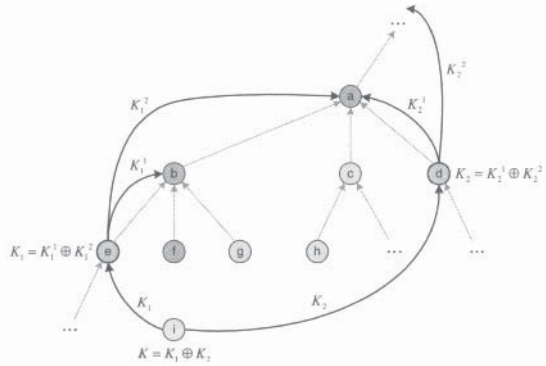


Figure 26.14 Sensor node i sends the key shares to ancestor nodes in the data aggregation tree [35].

Node d performs similar splitting and forwarding actions. Specifically, its share of $K_{i,f}$, K_2, is split into two key shares K_2^1 and K_2^2, which are forwarded to a and some ancestor of a, respectively.

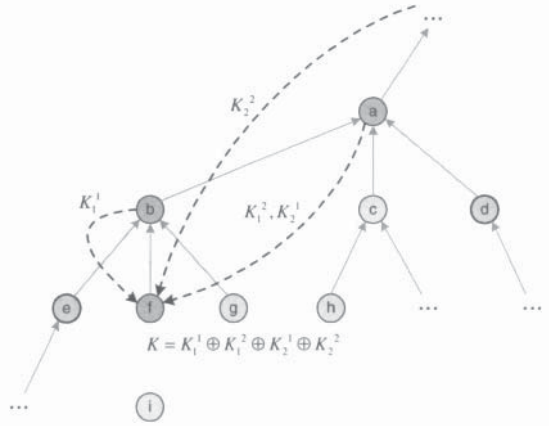


Figure 26.15 Ancestor nodes forward the key shares to the destination node f in the data aggregation tree [35].

As indicated in Figure 26.15, after decrypting the requests from e, a sends the following items to f:

$$\gamma_1^2 = (a, E_{K_{a,f}}(i, K_1^2))$$

Similarly, b sends the following item to f:

$$\gamma_1^1 = (b, E_{K_{b,f}}(i, K_1^1))$$

The ancestor of a and a itself also send similar items to f on behalf of d. Thus, f gets K_1^1, K_1^2, K_2^1, and K_2^2 so that it can restore $K_{i,f}$ by:

$$K_{i,f} = K_1^1 \oplus K_1^2 \oplus K_2^1 \oplus K_2^2$$

This ingenious key establishment protocol can deter attacks from one single malicious node. If there are potentially k malicious nodes ($k \geq 2$), then the protocol can be easily extended by splitting each key into $k + 1$ shares.

Anderson, Chan, and Perrig [17] suggested an interesting idea called "key infection" which is designed based on a practical attacking model. Specifically, Anderson *et al.* argued that it is impractical for the adversary (e.g., enemy in the battlefield) to capture all the traffic flows among all sensors when they are being deployed. The implication of this attacking model is that it becomes more economical and practical to send shared keys among sensors in plaintext when they are being deployed, instead of preloading them with large number of keys from a key pool (as in the classical approach suggested by Eschenauer and Gligor [105]). Their simulation results indicated that the key infection approach worked well for situations where there were up to 3% of malicious sensors in the field.

Miller and Vaidya [241] suggested a clever and comprehensive scheme for practical key establishment by exploiting channel diversity. There are two notable distinctive features in their proposed scheme. First, the sets of keys allocated to the sensors in the predistribution phase are *disjoint*. That is, there is no overlapping between any two sets of keys allocated. Second, in the key material exchange process during key discovery phase, keys are broadcast to neighbor sensors in *plaintext*. The first feature implies that the sensors have to exchange their keys by communicating with each other because they do not share any key implicitly. The second feature implies that during the mandatory keys exchange process, a malicious sensor or an eavesdropper can receive the keys easily because they are sent in plaintext. An important merit of the first feature is that if a sensor is compromised, only its neighborhood would become insecure because the set of keys stored in the compromised sensor would only be possibly used by its neighbors and, as such, other regions of the sensor network are unaffected. By contrast, in a traditional random key-pool predistribution scheme [105], a compromised sensor could possess keys that are used in many regions in the network.

Channel diversity is leveraged to tackle the second issue, namely the broadcasting of plaintext keys during the key discovery phase. Specifically, each sensor is assumed to have an RF module that has multiple channels: a common channel plus a number of other data channels. After the keys predistribution phase, sensors are deployed onto the field and their locations are assumed to be uniformly distributed. Furthermore, the density of the sensors, with respect to the RF transmission range, is assumed to be high enough so that there will be no shortage of one-hop neighbors (e.g., more than ten). When the key discovery phase begins, each sensor u first broadcasts to its neighbors a set data structure S_u, which represents the set of α keys it is allocated by the setup server. The set data structure has the property that it supports checking of

membership of a key but it does not reveal the values of the keys. An authentication data structure M_u is also broadcast. Specifically, a neighbor sensor v can use M_u to verify that the received S_u represents a legal set of keys allocated by the trusted setup server. We will describe the details about the data structures S_u and M_u below. After these two data structures are broadcast, each sensor u then switches to a randomly selected channel. The node u sends each of its α keys in plaintext over the channel and listens to the channel for a randomly selected amount of time. Specifically, the sensor u also listens to the broadcasts from its neighbors who happen to use the same channel at the same time (it is assumed that CSMA/CA is used for each channel). Thus, sensor u can record the keys it receives for subsequent key establishment. Note that the key broadcast time is independent of the channel listening time. As such, during the time a channel is used, the sensor u may only be able to broadcast only some of its α keys. The sensor u then switches to another randomly selected channel and repeat the process.

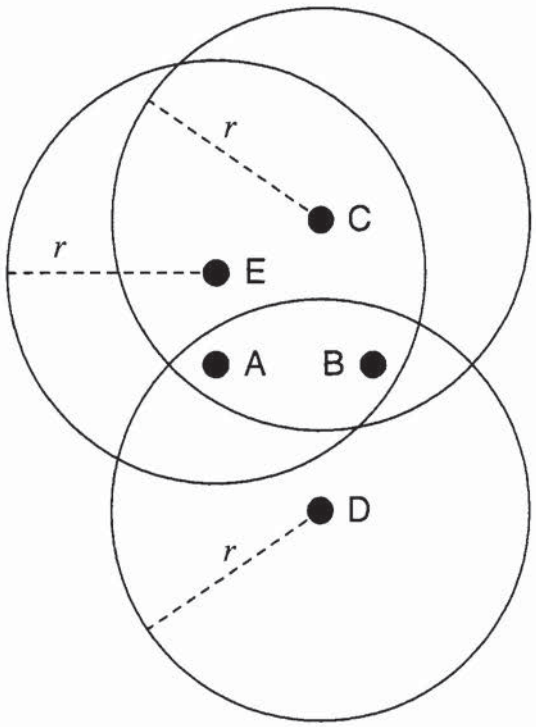


Figure 26.16 Sensor nodes A and B are one-hop neighbors of C, D, and E but nodes D is outside the transmission ranges of C and E. [241].

Channel diversity can probabilistically tackle eavesdropping, as illustrated in Figure 26.16. Suppose each sensor has c data channels. When sensor node C broadcasts

a key, the probability that both A and B receive it is equal to $1/c^2$. Furthermore, the probability that E does not eavesdrop the key is equal to $(c-1)/c$. Thus, A and B can possibly set up a secure link even though E is a malicious sensor. Similarly, D may not be able to eavesdrop the key.

After the plaintext key broadcasting phase is over, each sensor u possesses a combined set of $\alpha+\gamma$ keys. That is, contains its own α preallocated keys plus γ keys that it receives from all its neighbors. Note that the total number of received keys from the neighbors may be larger than γ. Nevertheless, only γ of them are randomly chosen as governed by storage limitations. Switching to the common channel again, each sensor u broadcasts a data structure C_u representing the combined set of keys. It is important to note here that the data structure does not reveal the values of the keys themselves (the same is true in the initial broadcasting of S_u). Upon receiving the broadcast from a neighbor, sensor u can then check, for each of its known keys (i.e., the $\alpha+\gamma$ keys), whether it exists in the received combined set of the neighbor. This process is repeated for each neighbor.

Now, the key establishment phase can begin. For each neighbor v of the sensor u, it creates a potential link key k_{uv} by using η of their common keys:

$$k_{uv} = \text{hash}(k_1||k_2||\ldots||k_\eta)$$

Sensor u then sends a Link Request (LREQ) message to v:

$$\text{LREQ} = (v||u||E_{k_{uv}}(RN)||K_{uv})$$

where K_{uv} is again the set data structure representing the set of η common keys, RN is a random nonce (which is encrypted using the potential link key k_{uv}). When v receives this request, it checks its combined set to see if it really has the η common keys. Suppose this is successful and, thus, v can decrypt the nonce. Sensor v then sends u a Link Reply (LREP) message:

$$\text{LREP} = (u||v||E_{k_{uv}}(RN+1))$$

A link key is then successful set up.

Miller and Vaidya's scheme [241], as described above, critically relies on the set data structure and the authentication of it. Specifically, the set data structure has to support the query of membership without revealing the values of the member keys. Furthermore, the data structure has to be compact so that it does not lead to a heavy storage and communication burden. Miller and Vaidya proposed to use the Bloom filters [37] data structure. Specifically, a Bloom filter is an s-bit vector (initially set to all 0), associated with h one-way has functions: $H^1_{BF}, H^2_{BF}, \ldots, H^h_{BF}$. Given an input value (i.e., a key in our context), each hash function is applied. Every hash function generates an output value between 0 and $s-1$. Suppose hash function H^i_{BF} generates an output value of x, where $0 \leq x \leq s-1$. Then bit x of the vector is set to 1. This is done for all h hash functions. Thus, to check for the membership of

a given value, all the hash functions are applied to it and see if the h bits are really set to 1 in the bit vector. The Bloom filter is a compact data structure in that, with constant physical storage (i.e., s bits), it can "store" many input values by just setting the corresponding bit positions to 1 (if they have not already been set to 1 by some earlier members). However, we can see that a Bloom filter can generate false positives because the bits indicating membership of a certain value may actually be set by some other members.

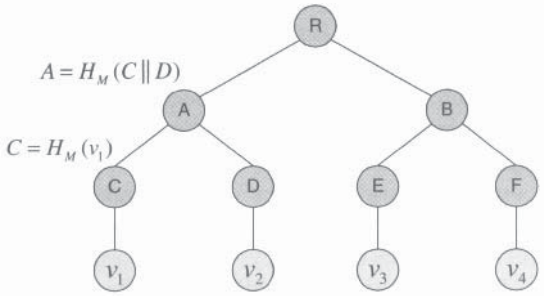


Figure 26.17 An example Merkle tree [241].

For the authentication data structure, Miller and Vaidya [241] used the Merkle trees [240]. Figure 26.17 shows a Merkle tree for authentication of four values: v_1, v_2, v_3, v_4. Notice that the circular nodes represent the tree nodes and leaves, whereas the square nodes represent the objects and are not stored as part of the tree. Each leave node is the output value of applying a one-way hash function to an object. For example, $C = H_M(v_1)$. Each internal node is the output value of applying the hash function to a concatenation of the children values. For example, $B = H_M(E||F)$. Now, each object representing a Bloom filter for a set of keys, can be verified by a receiving sensor with the Merkle tree nodes. For example, given v_2 together with the values of the internal Merkle tree nodes B, C, and R, a receiving sensor can verify the authenticity of v_2 (that is it is an authentic Bloom filter representing the set of keys) by checking the equality:

$$R = H_M(H_M(C||H_M(v_2))||B)$$

Miller and Vaidya [241] performed a detailed simulation study using the NS-2 simulator. With $\alpha = 100$, $\gamma = 100$, and $\eta = 10$, their results indicated that even with just one extra channel (i.e., $c = 2$), the proposed protocol can attain over 90% connectivity among neighboring sensors with link keys that are uncompromised even when 80% of the nodes are malicious.

26.5 DISCUSSIONS AND FUTURE WORK

In the above survey, we can see that techniques proposed for key predistribution and key establishment are practical yet theoretically sound. Researchers have designed their schemes that can cater for various practical constraints such as memory overhead, communication costs, and RF transmission range.

Having reviewed the state-of-the-art techniques proposed for key management in sensor networks, we believe that there are at least two important directions for future research. First of all, while the conventional wisdom is that public key schemes are impractical in a sensor network, Malan *et al.* [231] suggested an ingenious method for practical public key implementation in a resource constrained sensor network. It is likely that further research will make public key schemes more practicable. As such, new predistribution schemes may also be needed for preallocation and authentication of private keys.

Secondly, we consider "post-distribution" also an important topic. Indeed, from a fundamental perspective, the basic problem we need to tackle in key management for sensor networks is the allocation of keys to neighboring nodes so that they can setup secure links. While predistribution followed by discovery is a viable method, it is by no means the only practical approach. We believe that we could also design some post-distribution schemes where keys are allocated or generated *after* the sensor nodes have been deployed and have settled down in the field (i.e., topology of the neighborhood has been identified). The potential merit of a post-distribution approach is that there would be less severe wastage of keys. Toward this research direction, recently Law *et al.* [199] proposed a key redistribution scheme for sensor networks. Their proposed scheme is depicted in Figure 26.18 below. In simple terms, their redistribution scheme works by borrowing keys from neighbors. This is illustrated in Figure 26.19. Obviously, this is just an initial attempt and more work has to be done in this direction. Specifically, the question that needs to be tackled is how a sensor node can compute a key with a neighbor, potentially without predistribution, in a sense that post-deployment computation replaces pre-deployment key allocation.

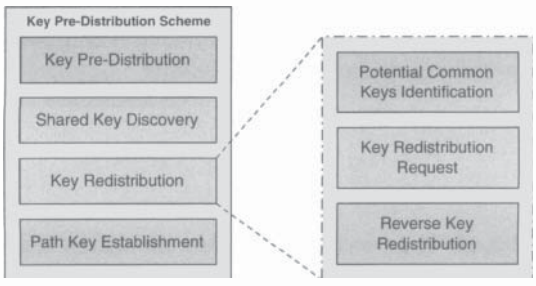


Figure 26.18 Key redistribution approach [199].

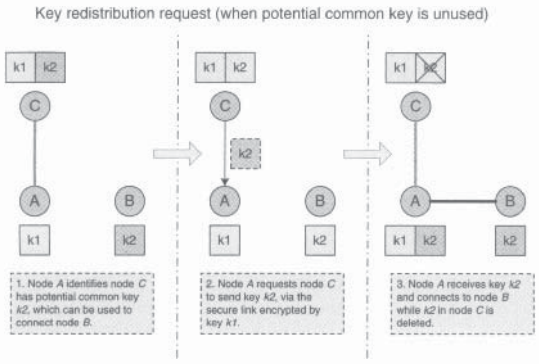


Figure 26.19 Key borrowing concept [199].

26.6 SUMMARY

More often than not, a sensor network is deployed in a hostile environment, where the attack model of an adversary can be very general. Key establishment among sensor nodes is a critical component in protecting the integrity and accuracy of a sensor network application. In this chapter, we have reviewed state-of-the-art efficient techniques proposed for key predistribution and the subsequent phase of key establishment. Many of these schemes are probabilistic in nature and are derived from judicious key space allocation. Despite the demonstrated effectiveness of these existing schemes, sensor network key management still offers a rich space for further research.

PROBLEMS

26.1 Compare and contrast the basic key pool based scheme and the bivariate polynomial based scheme.

26.2 How can deployment knowledge help in optimizing the performance of a key management scheme?

This Page Intentionally Left Blank

CHAPTER 27

PEER-TO-PEER COMPUTING OVER WIRELESS

A friend in need is a friend indeed.

—English Proverb, traced by Apperson to the XIII century

27.1 INTRODUCTION

Modern computing technologies have decentralized data processing power in an unprecedented manner. An important implication is that user machines, be it a desktop computer or a handheld PDA (personal digital assistant), have data processing power, in terms of instruction processing rate, amount of storage, and reliability, that is inconceivable merely a decade or two ago. Indeed, few people would now argue that computing occurs largely at the "edge" of networks, which have also made tremendous strides thanks to the ever improving communications technologies. Advancements in computing and communication, coupled together, enable a recent unstoppable trend in a new form of distributed processing—peer-to-peer (P2P) computing.

Wireless Internet and Mobile Computing. By Yu-Kwong Ricky Kwok and Vincent K. N. Lau
Copyright © 2007 John Wiley & Sons, Inc.

As its name implies, P2P computing involves users (or their machines) on equal footing—there is no designated server or client, at least in a persistent sense. Every participating user can be a server and be a client depending on the context. Some people have referred this to as a "democratic computing environment" [18] because users are free from centralized authorities' control. This new paradigm of distributed computing has spurred many high profile applications, most notably in file-sharing, such as: BitTorrent [70], Freenet [68], Gnutella [134], and, of course, Napster [255].

Interoperability:

In contrast to the traditional client/server file transfer model, peer-to-peer (P2P) systems introduce a new file transfer paradigm—peer-to-peer data transmission. Within this P2P model, there are several types of topology involved, including centralized, decentralized structured and decentralized unstructured. In fact, each kind of topology is developed in view of the disadvantage of others. As a result, they are not interoperable. This is further discussed in Section 27.2 and Section 27.3.

In this chapter, the pioneering idea of wireless P2P systems is our main focus. The wireless medium has actually brought new challenges to the P2P world. Unlike in a traditional wired network where the clients are within a fixed backbone (i.e., the clients connect to ISPs and the ISPs interconnect to form the Internet), in an ad hoc wireless network the mobile clients themselves form the backbone of the system. The formation of the wireless backbone itself forms a topology. As a result, it is required to have a topology control on it. We discuss this in Section 27.4.

There are two topology formations, namely application layer topology (wired P2P systems) and MAC layer topology (wireless ad hoc networks). These two concepts seem to be unrelated and non-interoperable to each other as the former is targeted on wired environments while the latter is targeted on wireless environments. However, the pioneering wireless P2P concept brings these two concepts together. In order to support this concept, it is necessary to integrate these two topology formations together and make them interoperable with each other. The introduction of the TCP2P layer is one of the solutions, and this is further discussed in Section 27.6.

Performance:

It has been suggested that the wireless topology formation can affect the P2P file sharing topology formation. On the other hand, there are some other factors that can severely affect the performance of wireless P2P systems. For example, the file size, the maximum display time of files, the bandwidth required for streaming, the popularity of files, the number of copies replicated and their corresponding locations, etc. In Section 27.5, metrics on file size, delay time and required bandwidth are discussed with the consideration of the access popularity.

27.2 A TAXONOMY OF P2P SYSTEMS

Figure 27.1 illustrates a taxonomy of P2P systems [208]. A P2P system includes several components—peer searching, peer selection, peer membership management

and file sharing protocols. On the other hand, as for file sharing protocols, several generations have been developed and they can be classified into three main types, including centralized, decentralized structured and decentralized unstructured. We further discuss this in Section 27.3.

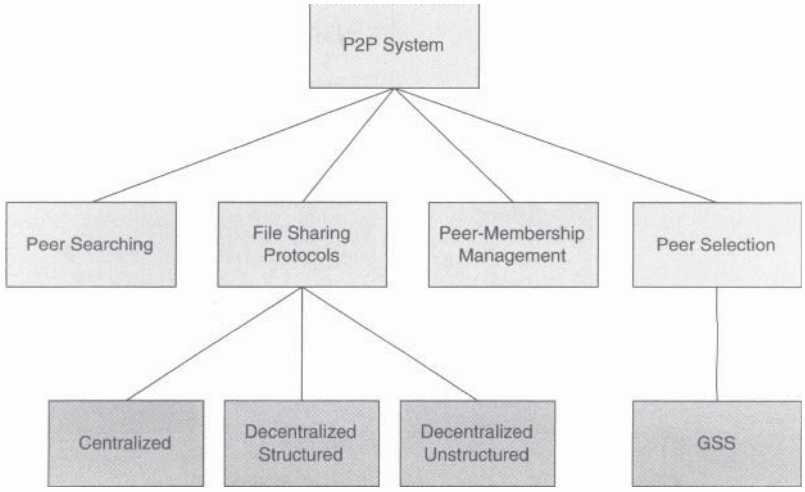


Figure 27.1 Taxonomy of P2P systems.

On the other hand, Figure 27.2 summarizes topology control techniques in the wireless ad hoc network environment. First of all, topology control techniques can be divided into two main categories—graph theoretical approach and non-graph theoretical approach. The graph theoretical approach is not our main focus in this chapter and we will briefly discuss it in Section 27.4. On the other hand, for non-graph theoretical approach, the topology formation is not considered as a graph with edges and vertices. In this case, some techniques are required to control the topology, such as the construction of neighbor sets, transmission power control, nodes sleeping schedule, etc. With the use of these techniques, the topology of a wireless ad hoc network can be restricted to be certain topology. This is further discussed in Section 27.4.

27.3 WIRED P2P SYSTEMS

Wired P2P systems are developed in view of great demand and popularity of file sharing between groups of users. Within a group, users take the responsibility of server and client at the same time. Due to this reason, every user becomes a "peer" in the network. There are many advantages of using this model. The main advantage is that the bandwidth utilization is more evenly distributed among the peers who have replicated the whole copy or chunks of the copy of a shared file. By doing so, no

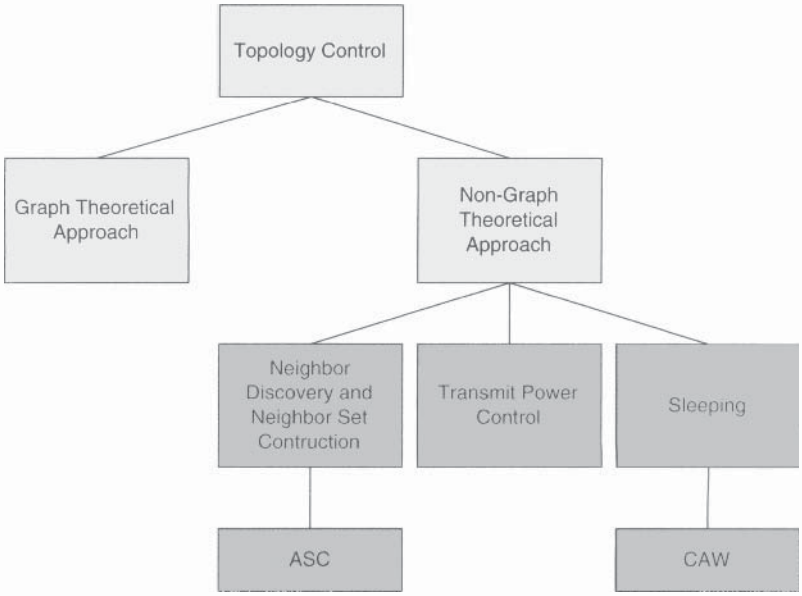


Figure 27.2 Taxonomy of topology control techniques.

single peer will be easily overcrowded since some other peers will also have the same replicated copies.

P2P file sharing systems we encounter in daily life can be classified into three generations [208]. The first generation uses a centralized approach in which a centralized directory is used to store the information of the files that have been shared and the locations of those files. Figure 27.3 illustrates this centralized approach. The clients first update with the server on the list of files that they are willing to share. In the figure, client 1 would like to request an object X. It then sends a query to the centralized directory server. The server then replies with a list of locations where the object X is shared. In this case, client 1 chooses client 3 to download the object and client 3 then sends the required object to client 1 directly.

The centralized approach is adopted by a famous music P2P file sharing company called Napster [255]. Napster keeps a centralized server as a global directory and users share their files using Napster services. However, due its centralized involvement in the file sharing process, Napster has been sued by several music producer companies and was forced to close down its server because of copyright infringement problems.

In view of the legal action against Napster, the second generation of P2P file sharing protocol is developed with the use of decentralized directory approach. Gnutella [134] is one of the most popular examples. In Gnutella, the file query task is distributed to all the nodes within a certain group by flooding the query to neighboring nodes until one of the nodes replies with the presence of the required file or no one replies.

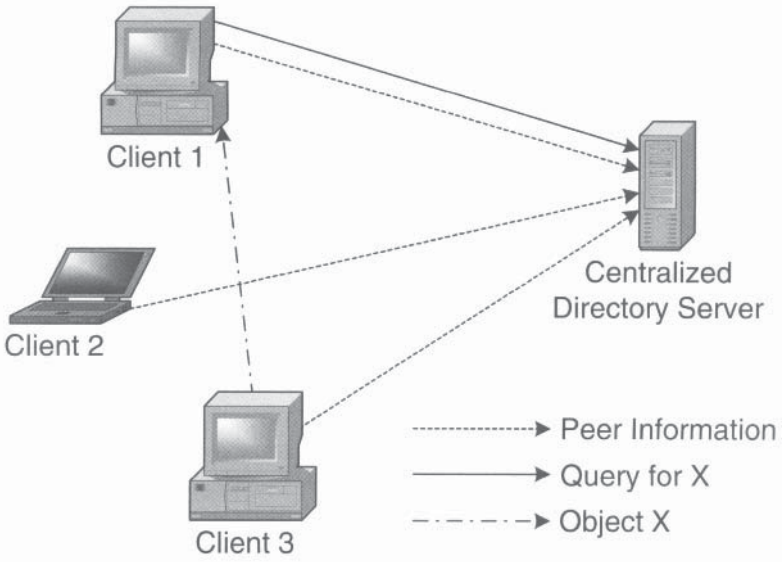


Figure 27.3 A centralized P2P system.

With the use of this idea, there is a bootstrap node to contact with, so that the user can join the group. However, since the bootstrap node itself does not store any file information, it evades the legal action similar to the case of Napster. At the same time, this idea does not guarantee that users must be able to join a group with the files they want, and the users can join and leave the group at any time they want. As a result, the system becomes unstructured as shown in Figure 27.4 and the peer-searching task usually involves flooding.

BitTorrent (BT) [34] can be regarded as the third generation P2P file sharing protocol. From a structural point of view, it is still the same as the second generation file sharing protocol. However, they are different in the sense that BT completely combines the idea of server and client. As a result, a user within a group becomes a "peer". The whole requested file is broken into chunks of smaller size. When a user is requesting chunks that he does not have from other users, other users will also request the chunks that the user has. Due to this reason, a user will download and upload at the same time and the network utilization is highly distributed. With the use of this kind of P2P system, it seems easier to evade from legal actions. However, in Hong Kong, there is a successful legal action against a BT user who had initiated the sharing of movie files.

After the discussion of the evolution of wired P2P systems together with daily life examples, readers may figure out that one type of P2P topology is missing. It is the decentralized structured P2P system. The structure is commonly formed with the use

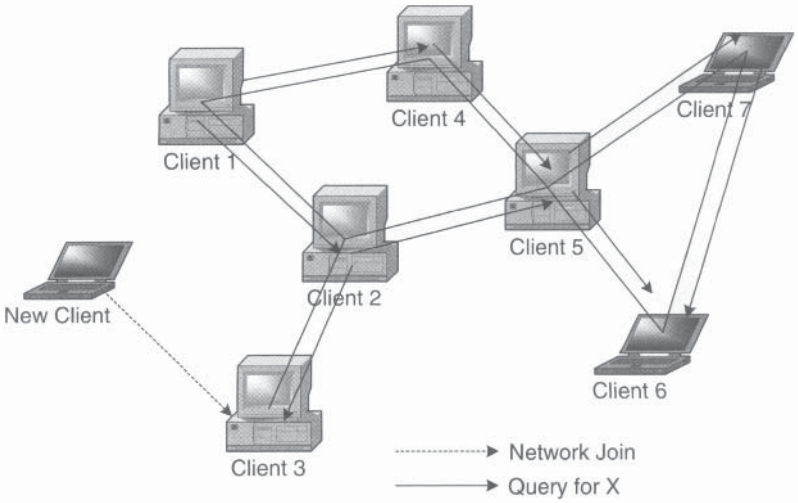


Figure 27.4 A decentralized unstructured P2P system.

of distributed hash tables (DHTs). A DHT can be used to search for files and find the location of the peer in a distributed environment. In other words, a DHT can be used for file searching and peer searching. One of the popular examples is Chord [328], which is developed by MIT. In Chord, nodes are conceptually connected as a ring as illustrated in Figure 27.5

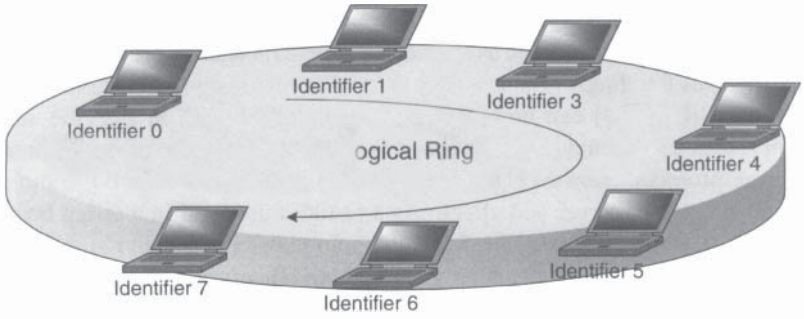


Figure 27.5 A decentralized structured P2P system.

The decentralized unstructured topology is more likely to dominate in the wireless P2P environment. It is because its decentralized and unstructured nature is likely to happen in the wireless ad hoc environment. For example, when you are walking with your mobile phone in a shopping mall, there may be some other people doing

the same thing. In this decentralized environment, it is unlikely to have a centralized server around nor structured topology built. However, due to the unstructured nature, there are certain difficulties required to be handled, such as selecting which peer to download/upload. On the other hand, the formation of a wireless ad hoc network also brings out new difficulties, as certain topologies will be formed during the formation of the network. Some of the topology formation ideas are discussed in subsequent section. A practical illustration is given in Section 27.6 to demonstrate the possibility of integrating different topologies to form a network dedicated to wireless P2P file sharing.

27.4 TOPOLOGY CONTROL IN WIRELESS AD HOC NETWORKS

After the discussion of the evolution of different wired P2P topologies, we start to discuss topology control techniques that are commonly adopted in wireless ad hoc networks. Recall from Figure 27.2 that there are two main categories of topology control in wireless ad hoc networks, namely graph theoretical approach and non-graph theoretical approach.

27.4.1 Graph Theoretical Approach

Generally speaking, graph theoretical approach involves the consideration of the whole wireless ad hoc network as a graph G. The graph is composed of a set of nodes (i.e., vertices) V. Between vertices, the nodes may be connected with each other, and a set of edges E is formed. However, in a wireless environment, it is not necessary to have a mutual communication between two nodes. In this case, communication is represented as a directed edge to represent one-way communication ability. Based on this information, a topology control algorithm can be applied to the graph G so as to construct a subgraph T of the graph G. After the application of the topology control algorithm, the nodes that communicate as indicated in subgraph T should experience certain performance improvement. Figure 27.6 illustrates the idea discussed.

There are a number of graph theoretical topology control algorithms proposed, such as Minimum-Energy Communication Network (MECN) [300], SMECN (Small MECN) [212], algorithms based on relative neighborhood graph (RNG) [104] and minimum spanning tree (MST) [214], etc. In MECN, it takes into account the energy needed to transmit a packet to the destination node. Specifically, certain nodes are assigned as relay nodes if it is found that less energy is required to deliver the packet to the destination node through relay nodes. SMECN, which is considered as an extended and improved version of MECN, is proposed to enable the support to mobile scenarios. However, Cheng *et al.* [64] discovered that both MECN and SMECN may not be able to find a minimum-energy topology if a node has two duties simultaneously, i.e., being both a relay node and a sink node. As a result, other approaches are required to tackle the problem.

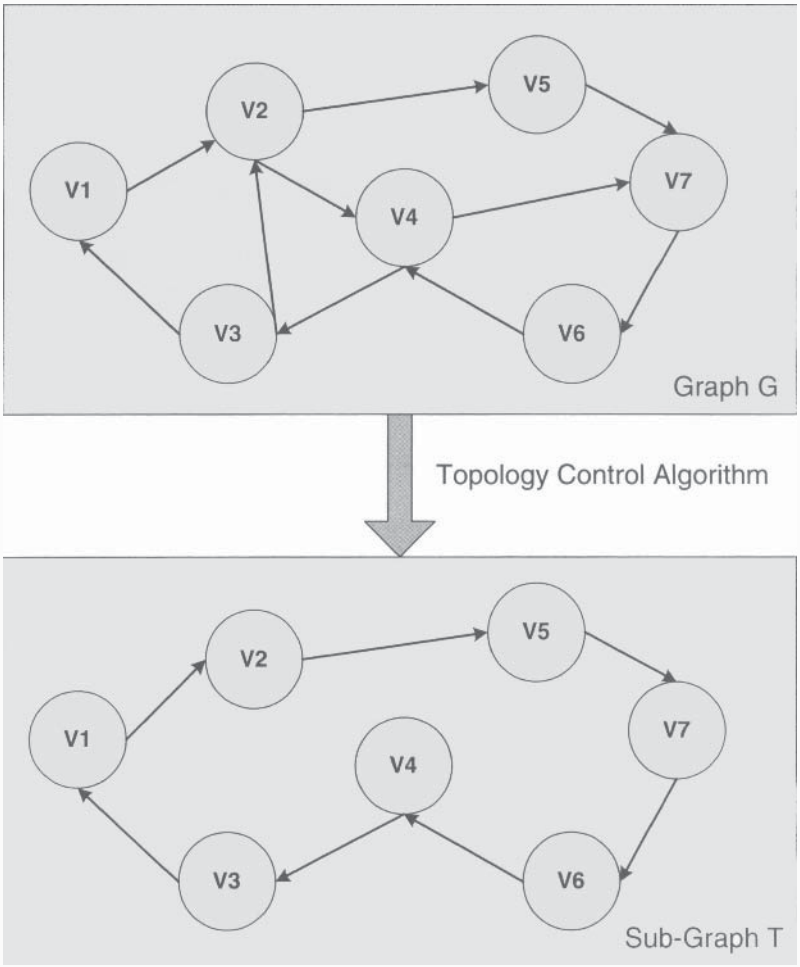


Figure 27.6 Example of graph theoretical approach.

27.4.2 Non-Graph Theoretical Approach

Most of the topology control algorithms of graph theoretical approach may require a node to have all the information about the whole network, so as to calculate the mathematical information required and to form the subgraph. However, this is usually not feasible in a wireless ad hoc network. As a result, it comes the non-graph theoretical approach. In this approach, it considers the nodes' connectivity with their neighbors, which in return controls the topology of the network.

Recall from Figure 27.2 that the non-graph theoretical approach involves the use of three main methods—neighbor discovery, transmit power control and sleeping. These three methods are discussed below:

Neighbor Discovery: The topology of a wireless ad hoc network is greatly varied with the possibility of the discovery of nodes around and the neighbor set construction criteria. For example, the minimum requirement for considering nodes around to be neighbors is that a node should be able to detect the present of other nodes. Generally, radio signals from a node are transmitted omnidirectionally and therefore nodes within certain radius of the node should be able to detect the present of the node. Due to this reason, the coverage area can be changed with the use of directional antennae and this may increase the chance of finding neighbors. On the other hand, certain criteria can be added into the neighbor construction process, e.g., minimum energy level. With the inclusion of different criteria, the topology is greatly varied because the number of potential neighboring nodes may be increased or decreased, depending on the criteria chosen.

Transmit Power Control: The topology of a wireless ad hoc network can also be varied by controlling the transmit power of the nodes. This technique in fact varies the length of the link between two connected nodes. Some examples of topology control with this technique are PCDC [251] and STPM [16].

Sleeping: Sleeping is necessary in some wireless ad hoc networks (e.g., a sensor network) since energy conservation is one of their main consideration and sleeping of a node can effectively save energy and prolong the battery life. With the presence of sleep nodes, the topology of the network keeps on changing. It is because the sleep nodes are not able to respond to any signal transmitted in the proximity. As a result, some of the transmission links will be temporarily disabled, which changes the topology. Due to this reason, sleep nodes must be integrated with some wakeup protocol to wake up from time to time to maintain the basic network connectivity.

27.5 MEDIA OBJECTS REPLICATION TECHNIQUES

In previous sections, various topics related to wired P2P systems and wireless ad hoc network topology control have been discussed. The main purpose of a P2P system is to share files between peers. Obviously, the number of replication of files within the system has a significant effect to the performance of the P2P system. However, typically there are a large amount of files in a P2P system, and it is therefore impossible for every peer to store the whole copies of all files in the P2P system. Due to this reason, a good object replication technique or strategy is crucial to the performance of a P2P system.

In this section, a wireless P2P media objects replication scenario is focused to discuss the impact of replication techniques on the performance of a P2P system [133]. Figure 27.7 illustrates the scenario. In the figure, there is a base station which is supposed to have all the media objects and would like to distribute the objects to various home-to-home on-line (H2O) devices. These devices have the ability of creating or producing media objects to be displayed in other H2O devices, displaying media objects, and relaying media objects data from one H2O device to another H2O device.

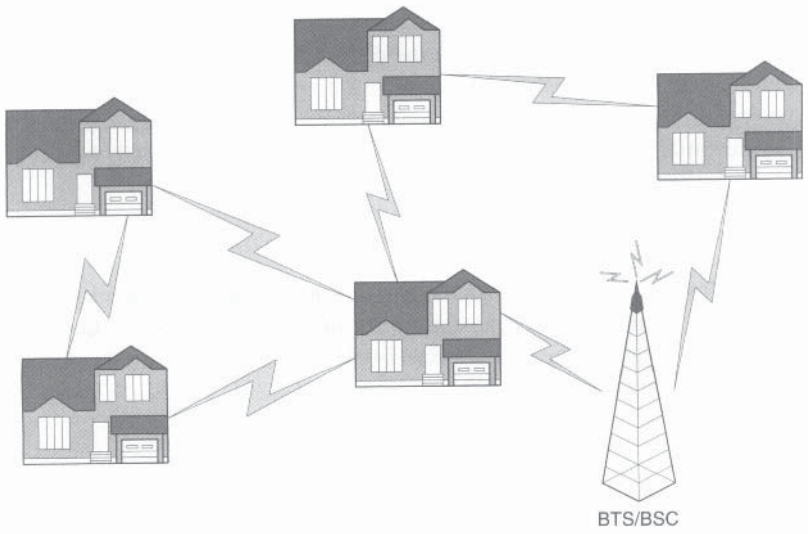


Figure 27.7 Architecture home-to-home on-line (H2O) devices.

However, there are several aspects that can greatly affect the performance of the system:

- media objects require continuous bandwidth for steady display;

- the bandwidth between each link depends on the number of transmitting devices in the corresponding region; and

- the topology.

In order to level out the above issues, assumptions on string topology and fixed bandwidth between each link are made. With these assumptions, there are three different kinds of replication strategies. They are based on the size of objects, display time of objects and the bandwidth requirement of objects. These strategies are tested against uniform distribution of objects, proportional to the object access frequency and square root of object access frequency, respectively. With these configurations, the performance comparison is determined by the number of simultaneously displayed media objects that the system can support.

In the configuration of 100 video clips with bandwidth requirement of 4 Mbps and display time of 120 minutes (this is a homogeneous setting in which every clip is the same), the three replication techniques obtained similar performance.

In the heterogeneous setting scenario, over 400 video clips are mixed with over 3500 audio clips. Video clips have four different display times ranging from 30 minutes to 120 minutes and have bandwidth requirement of 4 Mbps. On the other hands, audio clips have three different play times ranging from 2 minutes to 8 minutes and have bandwidth requirement of 340 kbps. Under these configurations, if the bandwidth replication technique is used, video clips are replicated 12 times more than audio clips (this is roughly the ratio between bandwidth requirements of video clips and audio clips). However, when size is used as the replication technique, with the average size of over two gigabytes for video clips and average size of over 10 megabytes for audio clips, video clips are replicated over 100 times more than audio clips. On the other hand, using the display time replication technique, video clips are replicated 14 times more than audio clips.

The results showed that the replication technique which is based on the size of objects obtains the worst performance, as it replicates many video clips and not much space is left for audio clips. The situation is worsen especially in the case that the references of audio clips are much larger than that of video clips. On the other hand, the replication technique which is based on the bandwidth requirement of objects obtains the best results, and it can replicate objects into the nodes with around 1/10 of the space used.

27.6 PRACTICAL ILLUSTRATIONS—LOCALIZED TOPOLOGY CONTROL

Previously, we have discussed various issues that are required to be considered in a wireless P2P system, including wired P2P systems, wireless ad hoc network topology control, and media objects replication techniques. In this section, we discuss

a practical wireless P2P system, especially on how the system handles the issues of integrating the topology control of a P2P system with a wireless ad hoc network [208].

First of all, we start by talking about the target application scenario. Figure 27.8 illustrates the target application scenario of the wireless P2P system. As shown in the figure, in a shopping mall there are a number of users with portable devices which have short-distance wireless connection capability. The portable devices may be mobile phones, PDAs or laptops. Under this scenario, it is unlikely that they have a common infrastructure for sharing of files between users. As a result, one of the possible ways of accomplishing the file sharing task is to build up a wireless infrastructure as shown in the figure. In this wireless infrastructure, users can join and leave at any time they want. This concept is similar to that of wireless ad hoc networks. However, the network construction process has to consider several parameters which will affect the network topology formed, including energy constraints, incentive, and fairness.

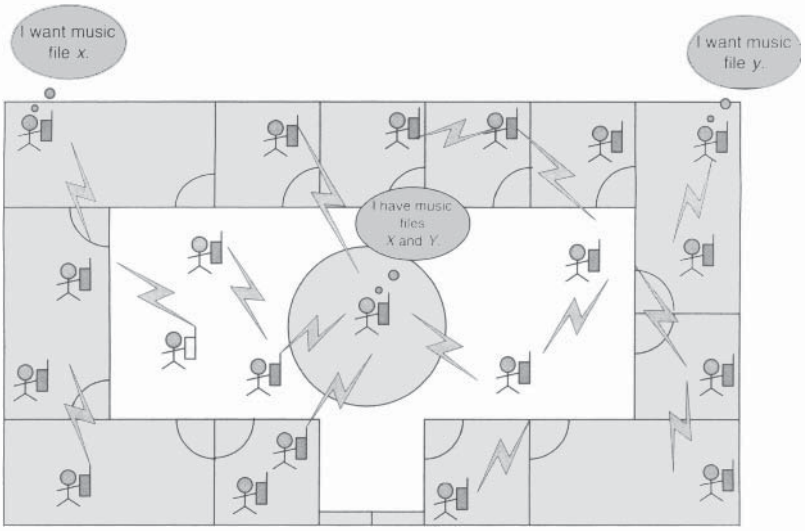


Figure 27.8 Shopping mall scenario of a wireless P2P system.

In view of the differences between wired and wireless P2P file sharing protocols, TCP2P [208] is proposed as illustrated in Figure 27.9 As indicated in the figure, TCP2P can be regarded as a protocol which runs in-between MAC layer and network layer. This protocol layer is mainly used for controlling the topology of the wireless P2P network formed and therefore it is a localized topology control protocol. TCP2P is composed of three main components—Adjacency Set Construction (ASC), Community-Based Asynchronous Wakeup (CAW) and Greedy Server-Peer Selection (GSS).

Recall from Figure 27.2 that one of the main problems in wireless ad hoc network topology control is the neighbor discovery and neighbor set construction process.

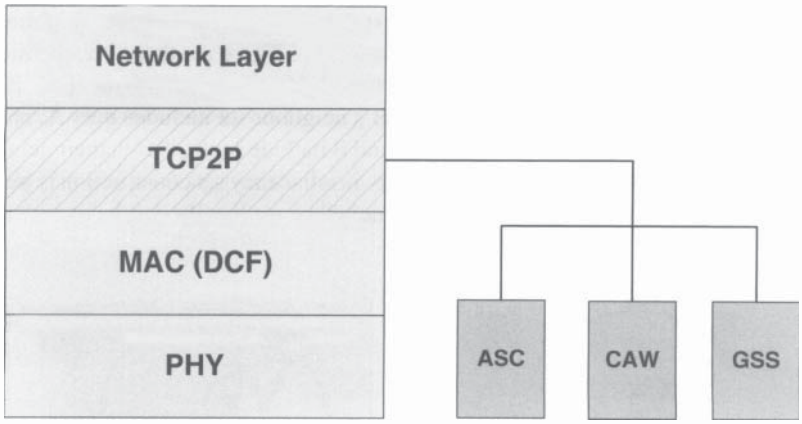


Figure 27.9 TCP2P protocol stack.

ASC is proposed to solve the problem "who is my neighbor?" [208]. In other words, ASC is proposed to construct next-hop set for peers locally, so that the upper layer—network layer can use the neighbor set information provided to route packets from the source node to the destination node.

As discussed previously, the minimum requirement to construct the neighbor set is that a node should be able to detect the signal transmitted by other peers (i.e., neighbors). After the fulfillment of this minimum requirement, the problem becomes why the neighbors are willing to become next-hop neighbors of the node. There are several factors that a peer needs consider in order to decide whether it should be the next-hop of another peer. The factors include the following:

Fairness: If a peer has been used as a relay node for a long time and it has also contributed a lot in routing control packets for maintaining the network connectivity, it should be exempted from being a next-hop neighbor of other peers.

Aggressiveness: If a peer always asks for some files from other peers without contributing much to the whole P2P system, other peers will have less will to be its neighbor.

Access Popularity: If a peer holds lots of files or a certain amount of very popular files, there is a high chance that other peers will request files from this peer. In this case, the workload will be very high for being a next-hop neighbor of the peer.

Remaining Energy: If the remaining energy of a peer is low, it is unlikely that the peer is willing to be the neighbor of other peers. It is because quite a large amount of energy will be consumed in acting as a relay node.

After quantifying the above factors against some threshold values, a neighbor set for wireless P2P file sharing can then be constructed. There are three possibilities as shown in Figure 27.10 The first case is that peer A's neighbor set includes peer B, but not vice versa. The second case is that peer B's neighbor set includes peer A, but not vice versa. The last case is that both peers A and B include each other in their neighbor sets. The existence of the asymmetric property in adjacency set construction is mainly due to the fact that every peer constructs its neighbor set locally and independently.

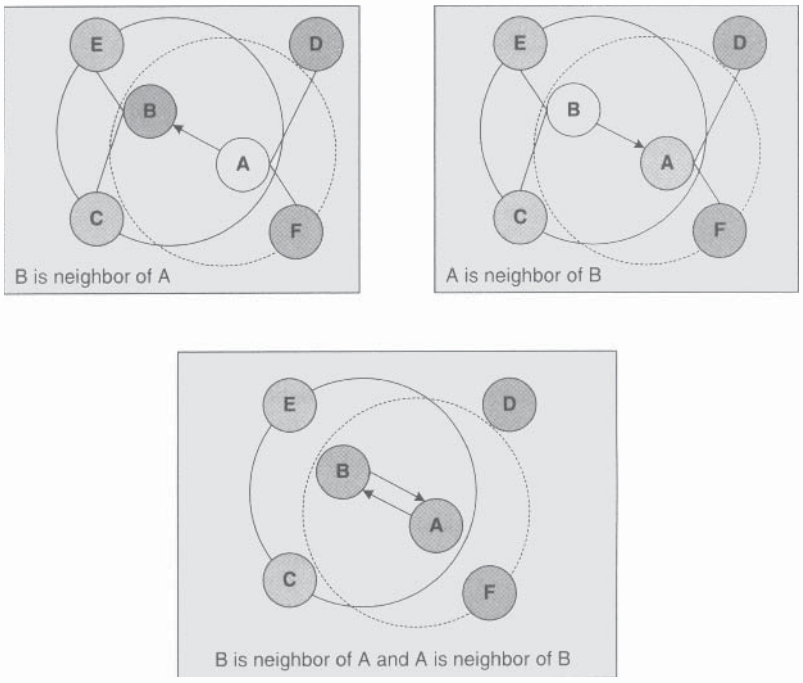


Figure 27.10 TCP2P ASC neighbor set.

The use of ASC in the wireless P2P file sharing scenario has some advantages, including a higher file request success ratio, shorter delays, more even distribution of relay and server nodes, and less deviated energy levels of nodes.

The second component of the TCP2P protocol is Community-Based Asynchronous Wakeup (CAW). One of the expected usages of the TCP2P protocol is to share and exchange some music files. Usually, the fans of a particular singer will have the tendency of downloading music files of this particular singer. As a result, the files stored and shared by fans of this particulars singer are in fact very similar. In this case, these users should have similar interest and they have the potential to form a

community [1] as shown in Figure 27.11 However, some measurement techniques have to be adopted in order to measure the similarity between two peers. One of the viable solutions is Information Retrieval (IR) [307]. With the use of techniques like IR, peer nodes can be grouped into different communities.

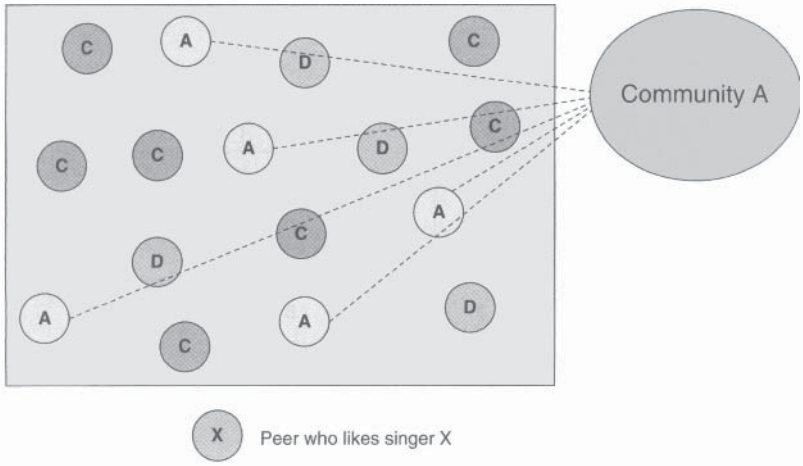


Figure 27.11 Example of community formation.

The problem becomes why should these communities be formed. Recall from Figure 27.2 that another possible technique of controlling the topology of a P2P system is to adopt sleeping in the network. One of the sleeping techniques is to apply asynchronous wakeup scheduling with the use of Wakeup Schedule Function (WSF) [371]. With the use of this function, it is guaranteed that mobile nodes using different schedules will have at least one active time slot overlapping with mobile nodes using other schedules. Figure 27.12 shows an example of WSF design and it is known as "(7,3,1)-design". In the figure, there are seven different schedules in which any schedule will have at least one active time slot overlapped with other schedules.

In the TCP2P protocol, CAW adopts the concept of community into sleeping schedules. Different communities take up different schedules and the peers in the same community, which is defined by the similarity of peers, use the same wakeup schedule. The main advantages of this design is that peers with similar interest can wake up at the same time for file sharing. As a result, those who are not interested in files being shared can be put into sleep mode, which eventually results in a shorter delay for transferring files.

After discussing the questions on "who is my neighbor?" and "when should I sleep and wake up?", we discuss the third component of the TCP2P protocol—Greedy Server-Peer Selection (GSS), which is mainly designed to solve the problem "who

[1] A community refers to at least two entities which have particular preferences.

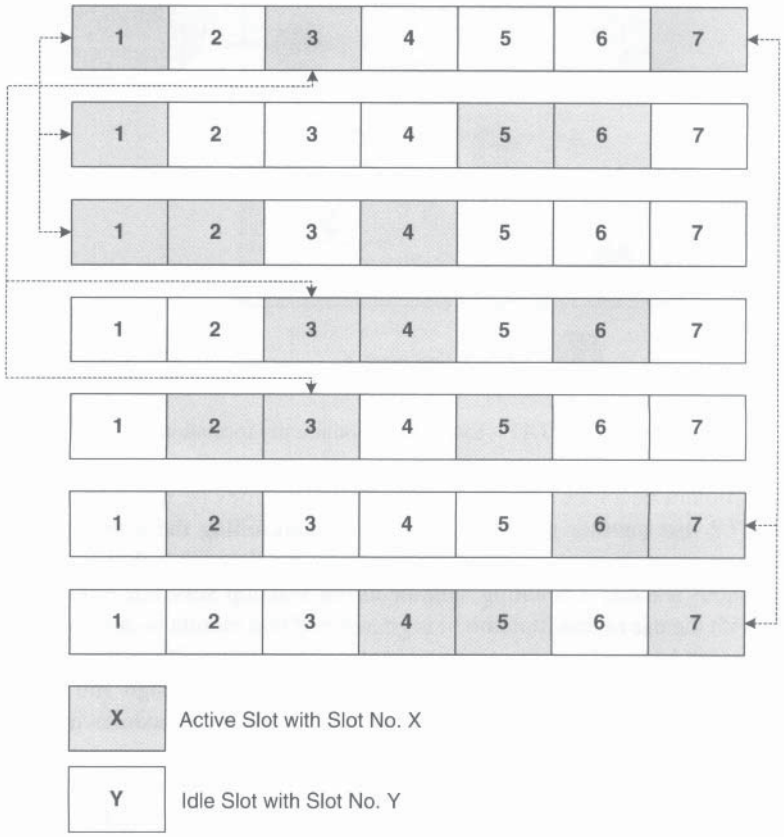


Figure 27.12 TCP2P CAW "(7,3,1)" WSF example.

should I download files from?". Recall from Figure 27.1 that peer selection is one of the main concerns in the P2P system design. It is because a peer usually has not much information on other peers within the system, and therefore it may be difficult to make decision on which peer should be selected to download the file required so that the selection can be beneficial to the whole system. As a result, the idea of Energy-Based Data Availability (EBDA) is proposed [208]. EBDA is a performance metric which takes into account the remaining energy of mobile nodes, the number of files that each node holds and the popularity of files.

In the peer selection process, it is better to select a peer which can make the EBDA metric more positive or at least less negative. Since the EBDA metric takes the remaining energy of peer nodes into consideration, a high value of the the sum of EBDA of all peers is significant to the P2P system. Specifically, the higher the value, the longer the system can survive. The direct consequence is that files can be kept longer in the P2P system.

27.7 SUMMARY

In this chapter, the pioneering concept of wireless P2P systems has been discussed. We have discussed the taxonomy of wired P2P systems and wireless ad hoc networks. Wired P2P systems are very popular nowadays with three different kinds of topology—centralized, decentralized unstructured and decentralized structured. The evolution of these topologies has been discussed.

After that, we have discussed the concept of topology control in wireless ad hoc networks. There are two main types of topology control in wireless ad hoc networks—graph theoretical approach and non-graph theoretical approach. In the non-graph theoretical approach, three main methods can be used to control the topology formation. They are neighbor discovery and neighbor set construction, transmit power control and sleeping.

Besides topology control, replication techniques are also important to the performance of a wireless P2P network sharing media objects. Several parameters are required to be considered, including the size of files, bandwidth required to transmit files at steady rate, display delay of files, and access popularity files. It is found that the performance of the P2P network is sensitive to both bandwidth required and the access popularity.

Lastly, a practical shopping mall scenario of the usage of a wireless P2P network has been illustrated. Specifically, the TCP2P protocol has been discussed. TCP2P includes three components—ASC, CAW, GSS. With the use of TCP2P, a localized topology control on a wireless P2P network can be done.

One major challenging issue in P2P systems is the need to provide incentives so as to entice cooperation among users. We discuss the incentive issues in Chapter 28.

PROBLEMS

27.1 Briefly discuss the wired P2P file sharing protocols.

27.2 Briefly discuss the wireless ad hoc network topology control protocols.

27.3 Summarize the idea of the localized topology control protocol TCP2P and discuss how it can integrate both P2P system topology control and wireless ad hoc network topology control.

CHAPTER 28

INCENTIVES IN PEER-TO-PEER COMPUTING

Even wisdom has to yield to self-interest.

—Pindar: Pythian Odes, III, c. 475 B.C.

It is worse to mistrust a friend than to be deceived by him.

—French Proverb

28.1 INTRODUCTION

As we have introduced in Chapter 27, Peer-to-Peer (P2P) computing has become an important distributed processing paradigm that is pervasive in nature. In a P2P system, users are autonomous and not subject to any central rule. Thus, some users may not be cooperative or altruistic. Indeed, these users usually are just selfishly trying to derive benefits from the P2P community without making any contribution. Such "free-riding" behaviors could render the P2P system ineffective or not viable. To encourage cooperative behaviors, incentives are mandatory. Incentive mechanisms

Wireless Internet and Mobile Computing. By Yu-Kwong Ricky Kwok and Vincent K. N. Lau
Copyright © 2007 John Wiley & Sons, Inc.

manifest themselves as payment based systems, auction based systems, exchange based systems, reciprocity based systems, and reputation based systems. We discuss the usage of various incentive mechanisms in two main P2P applications: file sharing and media streaming. Although most P2P incentive mechanisms designed for the Internet environment can be applied to a wireless system, there is a unique challenge in a wireless environment—the connectivity among peers is by itself a critical sharing problem.

The highly flexible features of P2P computing such as a dynamic population (users come and go asynchronously at will), dynamic topologies (it is impractical, if not impossible, to enforce a fixed communication structure), and anonymity, come at a significant cost—autonomy, by its very nature, is not always in harmony with tight cooperation. Consequently, inefficient or lack of cooperation could lead to undesirable effects in P2P computing. Among them the most critical one is "free-riding" [113] behavior. Loosely speaking, free-riding occurs when some users do not follow the presumed cooperation rules (in order to benefit the whole community) such as sharing files voluntarily, sharing bandwidth voluntarily, or sharing energy voluntarily. Such altruistic actions, presumably, would bring indirect and intangible (and even remote) returns to the users. For instance, if everyone shares files voluntarily, every user would eventually benefit from the high availability of a large and diverse set of selections. Unfortunately, there are always some users that do not believe or buy in to such utopia-like concepts and would, then, "rationally" choose to just enjoy the benefits derived from the community, but not contribute their own resources.

To deter or avoid free-riding behaviors, the P2P community has to provide some *incentives*—returns for resource expenditure that are, more often than not, tangible and immediate. Such incentives would then motivate an otherwise selfish user to rationally choose to cooperate because such cooperation would bring tangible and immediate benefits. Let us consider an analogy: in human society, getting pay from our work is a tangible and immediate incentive to motivate us to devote our energy, which could otherwise be spent on playing, to the work. Indeed, it is important for the incentive to be tangible so that a user can perform a cost-benefit analysis—if benefit outweighs cost, the user would then take a cooperative action [193]. It is also important for the incentive to be immediate (even though this is a relative concept) because any resource is associated with an opportunity cost in that if immediate return cannot be obtained from a cooperative action, then the user might want to save the effort for some other private tasks.

Interoperability:

To provide incentives in a P2P computing system, there are basically five different classes of techniques.

1. **Payment Based Mechanisms:** Users taking cooperative actions (e.g., sharing their files voluntarily) would obtain payments in return. The payment may be real monetary units (in cash) or virtual (i.e., some tokens that can be redeemed for other services). Thus, two important components are needed: (1) currency;

(2) accounting and clearing mechanism. Obviously, if the currency is in the form of real cash, there is a need of a centralized authority, in the form of an electronic bank, that is external to the P2P system. If the currency is in the form of some virtual token, then it might be possible to have a peer-to-peer clearing mechanism. In both cases, the major objective of is to avoid fraud at the expense of usually significant overhead. Proper pricing of cooperative actions is also important—over-priced actions would make the system economically inefficient while under-priced actions would not be able to entice cooperation.

2. **Auction Based Mechanisms:** In some situations, in order to come up with an optimal pricing, auction is an effective mechanism. In simple terms, auction involves bidding from the participating users so that the user with the highest bid get the opportunity to serve (or to be served, depending on the context). An important issue in auction based systems is the valuation problem—how much a user should set in the bid? If every user sets a bid higher than its true cost in providing a service, then the recipient of the service would pay too much that it is deserved. On the other hand, if the bids are too low, the service providers may suffer. Fortunately, in some form of auctions, we can design proper mechanisms to induce bidders to bid at their true costs.

3. **Exchange Based Mechanisms:** Compared to payment based and auction based systems, exchange (or barter) based techniques manifest as a purer P2P interaction. Specifically, in an exchange based environment, a pair of users (or, sometimes, a circular list of users) serve each other in a rendezvous manner. That is, service is exchanged in a synchronous and stateless transaction. For example, a pair of users meet each other and exchange files. After the transaction, the two users can forget about each other in the sense that any future transaction between them is unaffected by the current transaction. Thus, an advantage is that very little overhead is involved. Most importantly, peers can interact with each other without the need of intervention or mediation by a centralized external entity (e.g., a bank). Furthermore, free-riding is impractical. Of course, the downside is that service discovery and peer selection (according to price and/or quality of service) could be difficult.

4. **Reciprocity Based Mechanisms:** While pure barter based interactions are stateless, reciprocity generally refers to stateful and history based interactions. Specifically, a peer A may serve another peer B at time t_1 and does not get an immediate return. However, the transaction is recorded in some history database (centralized in some external entity or distributed in both A and B). At a later time $t_2 > t_1$, peer B serves peer A, possibly because peer B selects peer A as the client due to the earlier favor from A. That is, as peer A has served peer B before, peer B would give a higher preference to serve peer A. A critical problem is how we can tackle a special form of free-riding behavior, namely the "whitewashing" action (i.e., a user leaves the system and rejoins

with a different identity), which enables the free-rider to forget about his/her obligations.

5. **Reputation Based Mechanisms:** A reputation based mechanism is a generalized form of reciprocity. Specifically, while a reciprocity record is induced by a pair of peers (or a circular list of more than two peers), a reputation system records a score for each peer based on the assessments made by many peers. Each service provider (or consumer, depending on the application) can then consult the reputation system in order to judge whether it is worthwhile or safe to provide service to a particular client. Reputation based mechanism is by nature globally accessible and thus, peer selection can be done easily. However, the reputation scores must be securely stored and computed, or otherwise, the scores cannot truly reflect the quality of peers. In some electronic market place such as eBay, the reputation scores are centrally administered. But such an arrangement would again need an external entity and some significant overhead. On the other hand, storing the scores in a distributed manner at the peers would induce problems of fraud. Finally, similar to reciprocity based mechanisms, whitewashing is a low cost technique employed by selfish users to avoid being identified as a low quality users which would be excluded from the system.

Performance:

Different techniques mentioned above are suitable for different applications. Generally speaking, there are two mainstream applications in P2P environments: sharing of discrete data, and sharing of continuous data. Examples of the former include file sharing systems (e.g., Napster), data sharing systems (e.g., sharing of financial or weather reports), etc. A notable example of the latter is P2P video streaming. Indeed, there is an important difference between file sharing and media streaming systems. In the former, a user needs to wait until a file (or a discrete unit of shared information) is completely received before it can be consumed or used. Thus, there could be a significant delay between service request and judgement of service quality. In an extreme case, a user may not discover that a shared file is indeed the one requested or just a piece of junk. By contrast, in a media streaming application, a user would quickly discover if the received information is good enough. The quality of service (QoS) metric used is also different in these two different applications. In a file sharing application, the most important metrics are downloading time and the integrity of the received files. In a media streaming application, the more crucial performance parameters are the various playback quality metrics such as jitter, frame-rate, resolution, etc.

Currently, a majority of P2P systems are implemented over the Internet. However, wireless P2P systems are also proliferating. While most of the incentive techniques designed for a wired environment could be applicable in a wireless system, the wireless connectivity is by itself an important bootstrap sharing problem. Indeed, in the Internet, users seldom pay attention to the connectivity issue because a user can be

reached (or can reach) any other Internet user without noticeable effort. The only concern about communication is the uploading or downloading bandwidth consumption. In a wireless environment, however, the mere action of sending a request message from a client peer to a server peer would probably need several intermediate peers to help doing the message forwarding because the server and client peers may be out of each other's transmission range. Consequently, incentives have to be provided to encourage such forwarding actions.

In Section 28.2, we describe approaches designed for providing incentives in Internet based P2P networks. We discuss both file sharing and media streaming applications. In Section 28.3, we describe solutions suggested for wireless P2P systems. We then provide some of our interpretations and suggestions in Section 28.4. We summarize this chapter by providing some remarks in Section 28.5.

28.2 INCENTIVE ISSUES IN P2P SYSTEMS OVER THE INTERNET

28.2.1 File Sharing Systems

In a file sharing system, users would like to retrieve files from other users, and would expect other users to do the same. Thus, each user would need to expend two different forms of resource:

- Storage: Each user has to set aside some storage space to keep files that may be needed by other users, even though such files may not be useful to the user itself;
- Bandwidth: Each user has to devote some of its outbound bandwidth for uploading requested files to other users;

Users usually perform file selection (and hence, peer selection) with the help of some directory system which may or may not be fully distributed. For example, in Napster [255], the directory is centralized.

Under such a sharing model, the most obvious form of free-riding behaviors is that a selfish user just keeps on retrieving files from others but refuses to share its collections (and thus, no need to expend any outbound bandwidth for file uploading). Interestingly enough, in an empirical study using the Maze file sharing system [238] performed by Yang *et al.* [358], it is found that the more direct indicator of free-riding behaviors is the online time of a user. Specifically, the online time of a selfish user in a P2P file sharing network is on average only one-third of that of a cooperative peer.

In this section, we first briefly overview a contemporary file sharing system called BitTorrent [70]. We then survey techniques suggested for various other P2P file sharing networks.

28.2.1.1 BitTorrent BitTorrent [70] is by far one of the most successful P2P file sharing system. A key feature in BitTorrent is that each shared file is divided into

pieces (of size 256KB each), which are usually stored in multiple different peers. Thus, for any peer in need of a shared file, parallel downloading can take place in that the requesting peer can use multiple TCP connections to obtain different pieces of the file from several distinct peers. This feature is highly effective because the uploading burden is shared among multiple peers and the network can scale to a large size. Closely related to this parallel downloading mechanism is the incentive component used in BitTorrent. Specifically, each uploading peer selects up to four requesting peers in making uploading connections. The selection priority is based on descending order of downloading rates from the requesting peers. That is, the uploading peer selects four requesting peers that have the highest downloading rates. Here, downloading rate refers to the data rate that is used by a requesting peer in sending out pieces of some other file. Thus, the rationale of this scheme is to provide incentive for each participating peer to increase the data rate used in sending out file data (i.e., uploading, or, in BitTorrent's term, *unchoking*). There are other related mechanisms (e.g., optimistic unchoking), which are described in detail in [70, 290].

Qiu and Srikant [290] performed an indepth analysis of BitTorrent's incentive mechanism. By using an intricate and accurate model, it is shown that a Nash equilibrium exists in the upload/download game in BitTorrent. At the equilibrium, each peer sets its uploading data rate to be its physical maximum uploading rate (i.e., each peer is fully cooperative). On the other hand, due to the usage of the optimistic unchoking mechanism (a fifth requesting peer is randomly selected in the uploading process, for details, see [70, 290]), a free-rider can potentially achieve 20% of the possible maximum downloading rate. This theoretical result conforms nicely with the simulation findings by Jun and Ahamad [175] who observed that in BitTorrent, free-riders are not penalized adequately while contributors are not rewarded sufficiently.

28.2.1.2 Hierarchical P2P Systems In some situations, the P2P network may be structured in a hierarchical manner so that some specialized machines (called "super-peers") can take up a more important role for handling resource management tasks such as request forwarding and routing, directory listing, etc.

Singh *et al.* [321] studied the impacts of super-peers in a P2P file sharing network. Simply put, a super-peer is a special network node that serves a hub to provide file indexing service to other nodes. The problem is that there is a lack of incentive for a participating node to act as a super-peer because any node can simply join an existing super-peer to obtain good performance. Singh *et al.* observed that some entities external to the P2P system, such as an Internet Service Provider (ISP) or a content publisher, have business driven incentives for designating some nodes to act as super-peers, and for enhancing the capabilities of super-peers. Specifically, a super-peer can cache meta-data only instead of the files themselves. As such, the cost of acting as a super-peer is lower. Furthermore, with properly designed meta-data, a super-peer can support value-added search commands (e.g., topic based search of files).

With such facilities incorporated in each super-peer, a hierarchical P2P file sharing network is then much more efficient that a flat P2P system which relies on request-flooding. Consequently, all nodes in the system have the incentive in maintaining such a P2P file sharing system. Simulation results also indicate that Singh *et al.*'s proposed super-peer scheme is effective.

28.2.1.3 Payment Based Systems Hauscheer *et al.* [145] suggested a token-based accounting system that is generic and can support different pricing schemes for charging peers in file sharing. The proposed system is depicted in Figure 28.1. The system mandates that each user has a permanent ID authenticated by a certification authority. Each peer has a token account keeping track of the current amount of tokens, which are classified as local and foreign. A peer can spend its local tokens for accessing remote files. The file owner treats such tokens as foreign tokens, which cannot be spent but need to be exchanged with super-peers for new local tokens. Each token has a unique ID so that it cannot be spent multiple times.

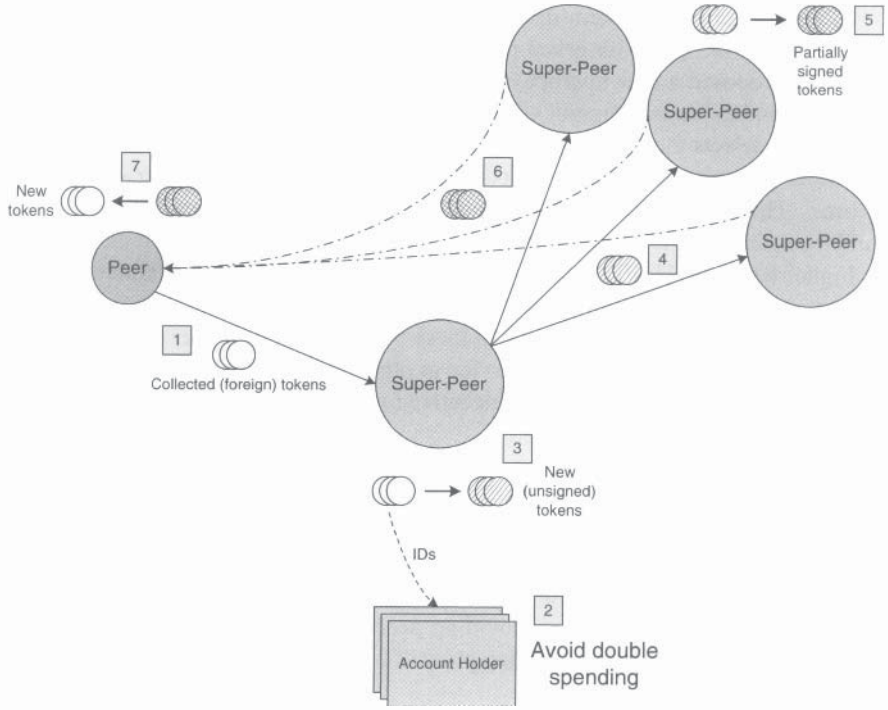


Figure 28.1 A super-peer based token accounting system for P2P file sharing [145].

At the beginning of each file sharing transaction, the file consumer tells the file owner about which tokens it intends to spend. The file owner then checks against file

consumer's account kept at the file owner's machine. If the tokens specified are valid (i.e., they have not been spent before), then the file consumer can send the tokens in an unsigned manner to the file owner. Upon receipt of these unsigned tokens, the file owner provides the requested files to the file consumer. When the files are successfully received, the file consumer sends the signed version of the tokens to the file owner. In this manner, Hauscheer *et al.* argued that there is no incentive for the peers to cheat.

Yang and Garcia-Molina [357] proposed the PPay micropayment system in which each peer can buy a coin from a broker. The peer then becomes the "owner" of the coin and can spend it to some other peer. An important feature is that even after the coin is spent, the original owner still has the responsibility to check the subsequent usage of the coin. For example, suppose A is the owner of a coin which is spent to B. If B wants to spend the coin in turn to C, the original owner A needs to check whether such a transaction is valid (e.g., to avoid double spending of the same coin). If A is offline (e.g., temporarily departed the P2P system), then the broker is responsible to perform such checking.

Although the PPay system described above is a useful tool for supporting P2P sharing, Jia *et al.* [169] observed that PPay can be further improved. Specifically, Jia *et al.* proposed a new micropayment system, called CPay (an improved version of PPay), which has one significant new feature. The new feature is that the broker judiciously selects the most appropriate peer to be the owner of a coin. Specifically, the owner of a coin should be one that is expected to stay in the system for a long period of time. Thus, the broker's potential burden of checking coin owners' transactions can be considerably reduced.

Figueiredo *et al.* [119] also considered a payment based system to entice cooperation among peers. Specifically, each peer requiring message forwarding service from other peers needs to pay real money to these peers. Thus, as a peer stays in the P2P network and provides forwarding service to other peers, it can gain money. Indeed, such a monetary gain represents an incentive to enhance the availability of a peer in the network.

Saito [303] proposed an Internet based electronic currency called i-WAT, which can be used by users in a P2P system for "purchasing" services. Each i-WAT message is an electronic ticket signed using OpenPGP. Saito *et al.* [304] then extended the i-WAT system by adding a new feature called "multiplication over time", which means that a requesting peer's debt (in terms of i-WAT units) increases over time. This feature then encourages service providing peers to stay in the system for a longer period of time so as to defer the redemption of i-WAT tickets, thereby increasing the gains from the requesting peers.

28.2.1.4 Cost of Sharing Varian [336] reported a simple but insightful analytical study on disincentives in P2P sharing. Table 28.1 lists the notation used in Varian's analysis.

Table 28.1 Notation used in Varian's analysis [336] on disincentives for P2P sharing.

Symbol	Definition
p	unit price
v	value of the item as perceived by each peer
n	number of peers in the system
D	total cost of producing all the items
$d = \frac{D}{n}$	average development cost
k	number of peers in each group that share an item
t	sharing cost incurred by each member of a group
c	cost imposed by the central authority to those peers who participate in sharing
π	profit derived by the central authority

In Varian's model, a single item (e.g., a single music file) is considered and the system is homogeneous in that all peers have the same valuation on the item. There is an external central authority (e.g., the original producer of the music) that has the incentive to discourage sharing among peers in the system. The issue is then how the central authority can introduce proper disincentives into the system. Firstly, observe that for viability in producing the item, we have:

$$v - \frac{p}{k} - t - c \geq 0 \tag{28.1}$$

$$p\frac{n}{k} \geq D \tag{28.2}$$

The equilibrium (where the item is just viable to be produced) price and profit are then given by:

$$p = (v - t - c)k \tag{28.3}$$

$$\pi = (v - t - c)kn - D \tag{28.4}$$

We can see that profit is, counterintuitively, decreasing in c. An interpretation is that c is not large enough to discourage sharing and price has to be cut for compensation. As an analogy, consider that c represents some copy protection mechanism which merely brings inconvenience to customers but cannot discourage sharing. To compensate for the inconvenience, the price has to be reduced.

On the other hand, for a given value of c, suppose the price is set in such a way that it is marginally unattractive to share. That is, we have:

$$\frac{p}{k} + t + c \geq p \tag{28.5}$$

This in turn implies that:

$$p = \frac{k}{k-1}(t + c) \tag{28.6}$$

Now, as the maximum practical value of p is v, we have:

$$c = v - \frac{k-1}{k}t \tag{28.7}$$

Yu and Singh [366] also investigated the issue of proper pricing in the presence of free-riders in a P2P system. A referral based system is considered in that each peer can either answer a remote query directly (e.g., serving a requested file) or reply with a referral (e.g., pointing to a different peer who may be able to serve the requested file). A requester (i.e., file consumer) needs to pay for both a referral or a direct answer. Each peer keeps track of reserve prices of potential referrals and direct answers from any other peer. These reserve prices are updated dynamically based on transaction experiences in that a satisfactory transaction leads to an increase in the reserve prices while an unsatisfactory one leads to a decrease. From a seller's point of view, these prices are also exponentially decreased as time goes by. Under this model, simulation results indicate that a free-rider will quickly deplete its budget. On the other hand, the price of a direct answer is found to be much higher than that of a referral.

Courcoubetis and Weber [75] recently reported an in-depth analysis of the cost in sharing in a P2P system. In their study, a P2P sharing system is modeled as a community with an excludable public good. Furthermore, the public good is assumed to be nonrivalrous, meaning that a user's consumption of the public good does not decrease the value of the good. Such a model is suitable for a P2P file sharing network, where the excludable public good is the availability of shared files. The model is also considered as suitable for a P2P wireless LAN environment, where the excludable public good is the common wireless channel. With their detailed modeling and analysis, an important conclusion is derived: each peer only needs to pay a fixed contribution, in terms of service provisioning (e.g., a certain fixed number of distinct files to be shared by other peers), in order to make the system viable. Such a fixed contribution is to be computed by some external administrative authority (called a "social planner") by using the statistical distribution of the peers' valuations of the public good.

28.2.1.5 Reciprocity and Reputation Based Systems Feldman *et al.* [114] suggested an integrated incentive mechanism for effectively deterring (or penalizing) free-riders using a reciprocity based approach. Specifically, the proposed integrated mechanism has three core components: discriminating server selection, maxflow based subjective reputation computation, and adaptive stranger policies.

In the discriminating server selection component, each peer is assumed to have a private history of transactions with other peers. Thus, when a file sharing request is initiated, the peer can select a server (i.e., a file owner) from the private history. However, in any practical P2P sharing network, we can expect a high turnover rate of participation. That is, a peer may only be present in the system for a short time. Thus, when a request needs to be served, such a departed peer would not be able to help if it is selected. To mitigate this problem, a shared history is to be implemented.

That is, each peer is able to select a server from a list of global transactions (i.e., not just restricted to those involved the current requesting peer). A practical method of implementing shared history is to use a distributed hash table (DHT) based overlay networking storage system [329].

A problem in turn induced by the shared history facility is that collusion among non-cooperative users may take place. Specifically, the non-cooperative users may give each other a high reputation value (e.g., possibly by reporting bogus prior transaction records). To tackle this problem, Feldman *et al.* suggested a graph theoretic technique. To illustrate, consider the reputation graph shown in Figure 28.2. Here, each node in the graph represents a peer (C denotes a colluder) and each directed edge represents the perceived reputation value (i.e., the reputation value of the node incident by the edge as perceived by the node originating the edge). We can see that the colluders give each other a high reputation values. On the other hand, a contributing peer (e.g., the top node) gives a reputation value of 0 to each colluder because the contributing peer does not have any prior successful transaction carried out with a colluder. With this graph, we can apply the maxflow algorithm to compute the reputation value of a destination peer as perceived by a source peer. For instance, peer B's (the destination) perceived reputation value with respect to peer A (the source) is 0 despite that many colluders give a high reputation value to B.

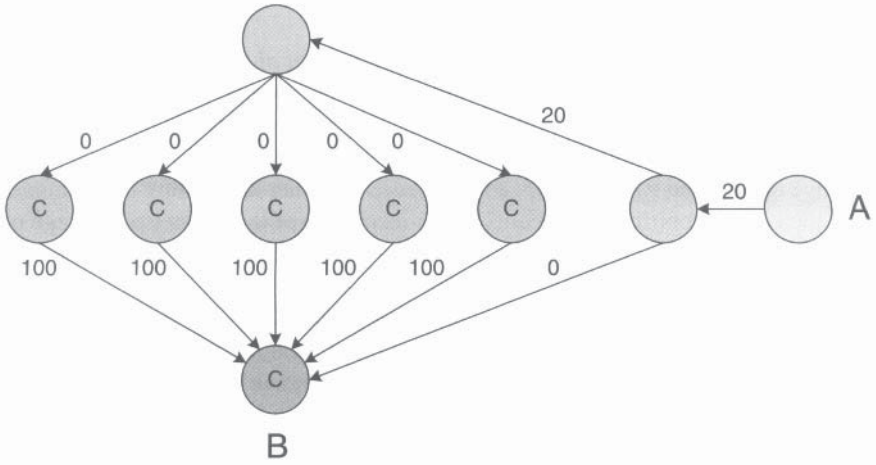


Figure 28.2 A graph depicting the perceived reputation values among peers (C denote a colluder) [114].

Finally, an adaptive stranger policy is proposed to deal with whitewashing. Instead of always penalizing a new user (which would discourage expansion of the P2P network), the proposed policy requires that each existing peer, before deciding whether to do a sharing transaction with a new user, computes a ratio of amount of services provided to amount of services consumed by a new user. If this ratio is great than or

equal to 1, then the existing peer will work with the new user. On the other hand, if the ratio is smaller than 1, then the ratio is treated as a probability of working with this new user.

Sun and Garcia-Molina [330] suggested an incentive system called Selfish Link-based InCentive (SLIC), which is based on pairwise reputation values. Specifically, any peer u maintains a reputation value $W(u, v)$ for each of its neighbor peer v, where the reputation value is normalized such that $0 \leq W(u, v) \leq 1$. Here, "neighbor" means a peer v currently having a logical connection with u and thus, such a peer v can potentially request for service from u. With these reputation values, the peer u can then allocate the uploading bandwidth to any requesting neighbor peer v with a value of $W(u, v)/\sum_i W(u, i)$. The reputation value $W(u, v)$ is updated periodically based on an exponential averaging method.

Under this model, Sun and Garcia-Molina [330] observed that each peer has the incentive to do some or all of the following, in order to increase its reputation values as perceived by other peers (and hence, enjoy a better quality of service).

- Sharing out more file data;
- Connecting to more peers (to increase the opportunities for serving others);
- Increasing its total uploading capacity;

28.2.1.6 Penalty Based Approaches Feldman *et al.* [115] also investigated disincentive mechanisms that can discourage free riding. Specifically, they considered various possible penalty schemes in deterring free-riders. A simple model is used. At the core of the model, each user i in the P2P sharing network is characterized by a positive real-valued *type* variable, denoted as t_i. Another key feature of the model is that the cost of contributing is equal to the reciprocal of the current percentage of contributors, which is denoted as x. Thus, for any rational user with type t_i, the user will choose to contribute if $1/x < t_i$ and free-ride if $1/x \geq t_i$.

Furthermore, the benefit each user derived from the P2P network is assumed to be of the form αx^{β}, where $\beta \leq 1$ and $\alpha > 0$. With this benefit function, the system performance is defined as the difference between the average benefit and the average contribution cost. Specifically, system performance is equal to: $\alpha x^{\beta} - 1$.

Even with the simplistic model described above, Feldman *et al.* provided several interesting conclusions. Firstly, it is found that excluding low type users can improve system performance only if the average type is low and α is large enough. Unfortunately, exclusion is impractical because a user's type is private and thus, cannot be determined accurately by other peers. It is then assumed that free riding behaviors are observable (i.e., free-riders can be identified). Such free-riders are then subject to a reduction in quality of service. Quantitatively, the benefit received by a free rider is reduced by a factor of $(1 - p)$, where $0 < p \leq 1$. A simple implementation of this penalty is to exclude a free rider with a probability of p. The second interesting conclusion is that the penalty mechanism is effective in deterring free-riders when

the penalty is higher than the contribution cost. In quantitative terms, the condition is that $p > 1/\alpha$. Finally, another interesting conclusion is that for a sufficiently heavy penalty, no social cost is incurred because every user will contribute (i.e., choose not to be a free-rider) so that optimal system performance is achieved. In particular, to deal with the whitewashing problem, the analysis suggests that every new user is imposed a fixed penalty. Essentially, this is similar to the case in the eBay system where every new user has a zero reputation and thus, will less likely be selected by other users in commercial transactions. However, this is in sharp contrast to the adaptive stranger policies suggested also by Feldman *et al.* in another study [114] that we have described earlier.

28.2.1.7 Game Theoretic Modeling Ranganathan *et al.* [294] proposed and evaluated three schemes induced by the Multi-Person Prisoner's Dilemma (MPD) [265, 312]. Specifically, the key features of the MPD framework can be briefly summarized as follows:

- The MPD game is symmetric in that each of n players has the same actions, payoffs, and preferences.
- Any player's payoff is higher if other players choose some particular actions (e.g., "quiet" instead of "fink").

The MPD framework is used for modeling P2P file sharing as follows. There are n users in the system, each of which has a distinct file that can be either shared or kept only to the owner. The system is homogeneous in that all files have the same size and same degree of popularity. Now, the potential benefit gained by each user is the access of other users' files. The cost involved is the bandwidth used for serving other users' requests. With this simple model, it can be shown that the system has a unique Nash equilibrium in which no user wants to share. Obviously, this equilibrium is sub-optimal (both at the individual level and at a system-wide level) in that each user could obtain a higher payoff (i.e., a higher value of net benefit) if all users choose to share their files.

Motivated by the MPD modeling, Ranganathan *et al.* proposed three incentive schemes:

- **Token Exchange:** This is a payment based scheme because each file consumer has to give a token to the file owner in the sharing process. Each user is given the same number of tokens initially and each file has the same fixed price.
- **Peer-Approved:** This is a reputation based scheme in that each user is associated with a rating which is computed using metrics such as the number of requests successfully served by the user. A user can download files from any owner who has a lower or the same rating. Thus, to gain access to more files in the system, a user has to actively provide service to other users so as to increase the rating.

- **Service Quality:** This is also a reputation based scheme similar to Peer-Approved. The major difference is that a file owner provides differentiated service qualities to users with different ratings.

Theoretical analysis indicates that the Peer-Approved policy with a logarithmic benefit function (in terms of number of accessible files) can lead to the optimal equilibrium where every user contributes fully to the system. Simulation results also suggest that Peer-Approved generates performance (in terms of total number of files shared) comparable to that of Token Exchange, which entails a higher difficulty in practical implementation as it requires a payment system.

Becker and Clement [29] also suggested a simple but insightful analysis of the sharing behaviors using variants of the classical 2-player Prisoner's Dilemma. Specifically, the P2P file sharing process is divided into three different stages: introduction, growth, and settlement. In the introduction stage, the P2P network usually consists of just a few altruistic users who are eager to make the network viable. Thus, sharing of files is a trusted social norm. The payoffs of the two possible actions (supply files or not supply files) are depicted in Figure 28.3. Here, we have the payoffs ranking as: $R > T > S > P$. Consequently, the Nash equilibrium profile is: (Supply, Supply). Notice that the payoffs ranking in the original Prisoner's Dilemma is: $T > R > P > S$, and as such, the Nash equilibrium is the action profile in the lower right corner of the table.

Player 1 \ Player 2	Supply g_2^1	No Supply g_2^2
Supply g_1^1	R, R	S, T
No Supply g_1^2	T, S	P, P

Figure 28.3 Payoff table in the introduction stage [29].

In the growth stage, we can expect that more and more non-cooperative users join the network. For these users, the payoffs ranking becomes: $T > R > P > S$, which is the same as the original Prisoner's Dilemma. Thus, the Nash equilibrium for such users occurs at the profile: (No Supply, No Supply). As the P2P network progresses to the mature stage (i.e., the size of the network becomes stabilized), we can expect that a majority of users are neither fully altruistic nor fully non-cooperative. For these users, the payoffs ranking is: $R > T > P > S$. As a result, the payoff matrix is depicted in Figure 28.4. As can be seen, there are two equally probable Nash equilibria: (Supply, Supply) and (No Supply, No Supply). Consequently, whether or not the P2P network is viable or efficient depends on the relative proportions of users in these two equilibria. Results obtained in empirical studies [29] using real P2P networks conform quite well to the simple analysis described above.

Ma *et al.* [228, 229] suggested an analytically sound incentive mechanism based on a fair bandwidth allocation algorithm. Indeed, the key idea is to model the P2P sharing

Player 1 \ Player 2	Supply g_2^1	No Supply g_2^2
Supply g_1^1	R R	T S
No Supply g_1^2	S T	P P

Figure 28.4 Payoff table in the settlement stage [29].

as a bandwidth allocation problem. Specifically, the model is shown in Figure 28.5. Here, multiple file requesting peers compete for uploading bandwidth of a source peer. Each requesting peer i sends a bidding message b_i to the source peer N_S. The source peer then divides its total uploading bandwidth W_S into portions of x_i for the peers. However, due to network problems such as congestion, each peer i may receive an actual uploading bandwidth of x'_i which is smaller than x_i.

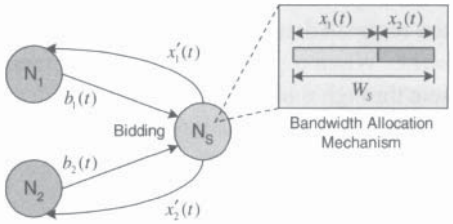


Figure 28.5 Two file requesting peers (N_1 and N_2) compete for uploading bandwidth of a source peer (N_S) [228].

Each bidding message b_i is the requested amount of bandwidth. Thus, we have $x_i \leq b_i$. To achieve a fair allocation, the source peer uses the contribution level C_i of each competing peer i to determine an appropriate value of x_i. Ma *et al.* [228, 229] described several allocation algorithms with different complexities and considerations: simplistic equal sharing, max-min fair allocation, incentive based max-min fair allocation, utility based max-min fair allocation, and incentive with utility based max-min fair allocation. The last algorithm is the most comprehensive and effective. It works by solving the following optimization problem:

$$\max \sum_{i=1}^{N} C_i \log(\frac{x_i}{b_i} + 1) \tag{28.8}$$

where:

$$\sum_{i=1}^{N} x_i \leq W_S \tag{28.9}$$

Here, the logarithmic function represents the utility as perceived by each peer i.

The above optimization problem can be solved by a progressive filling algorithm that prioritizes competing peers in descending order of the marginal utility $C_i/(b_i + x_i)$.

Given values of b_i and C_i, the source peer can compute the allocations in a deterministic manner. However, from the perspective of a requesting peer, a problem remains as to how it should set its bidding value b_i. Using a game theoretic analysis, it is shown that the action profile in which $b_i = \frac{W_S C_i}{\sum_{j=1}^{N} C_j} \forall i$ is a Nash equilibrium. Furthermore, provided that all cooperative peers use their respective strategies as specified in the Nash equilibrium action profile, collusion among non-cooperative peers can be eliminated. Notice that each requesting peer i needs to know the values of W_S and $\sum_{j=1}^{N} C_j$ in order to determine its own bid b_i. In a practical situation, these two values can be supplied by the source peer to every requesting peer.

28.2.1.8 Auction Based Approaches Gupta and Somani [138] proposed an auction based pricing mechanism for P2P file object lookup services. In their model, each resource (e.g., a file object) is stored in a single node. However, the indices for such a file object are replicated at multiple nodes in the network and these nodes are called *terminal nodes*. When a peer initiates a lookup request for a certain file object, the request is sent through multiple paths toward the terminal nodes, as shown in Figure 28.6. The problem here is that the intermediate nodes need some incentives in order to participate in the request forwarding process.

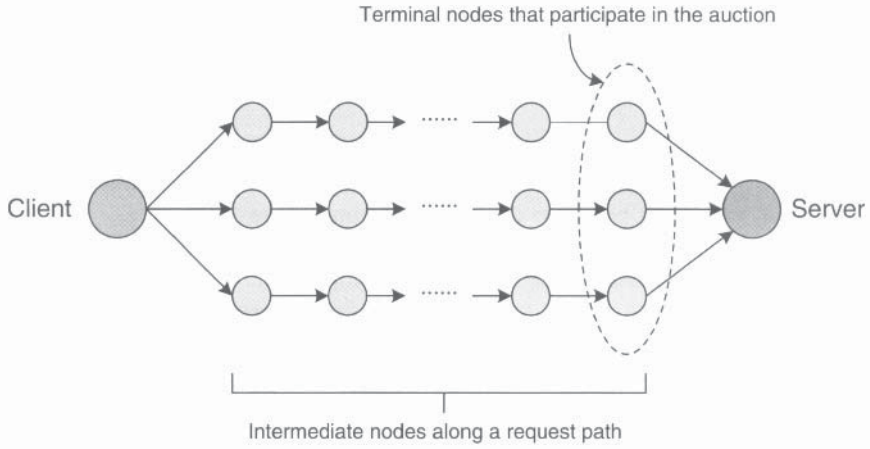


Figure 28.6 The request forwarding process [138].

Gupta and Somani [138] suggested a novel solution to the incentive problem. Specifically, the initiating peer attaches a price in the request message it sends to the first layer of nodes in the request chains. Each intermediate node on the request chains then updates the price by adding its own "forwarding cost". The terminal nodes

also do the same updating before sending the request messages to the data source. Upon receiving all the request messages, the data source then performs a second price sealed bid auction (also referred to as Vickrey auction) [265] to select the highest bid among the terminal nodes. The selected terminal node then needs to pay the price equal to the value of the second highest bid. With this auction based approach, all the intermediate nodes on the request chains have the incentive to participate in the forwarding process because they might eventually get paid by the requester should their respective request chain wins the auction.

For example, consider the lookup process shown in Figure 28.7. We can see that the request chain terminated by node T1 wins the auction process and the payoff to the data source node B is 60. The only intermediate node (node 1) then also gets a payoff. Gupta and Somani [138] also showed that a truthful valuation in is the optimal strategy for each intermediate node. Furthermore, based on the requirement that every message is non-repudiable, it is also shown that the proposed mechanism can handle various potential threats such as malicious auctioneer, collusion between data source and a terminal node, forwarding of bogus request message, etc.

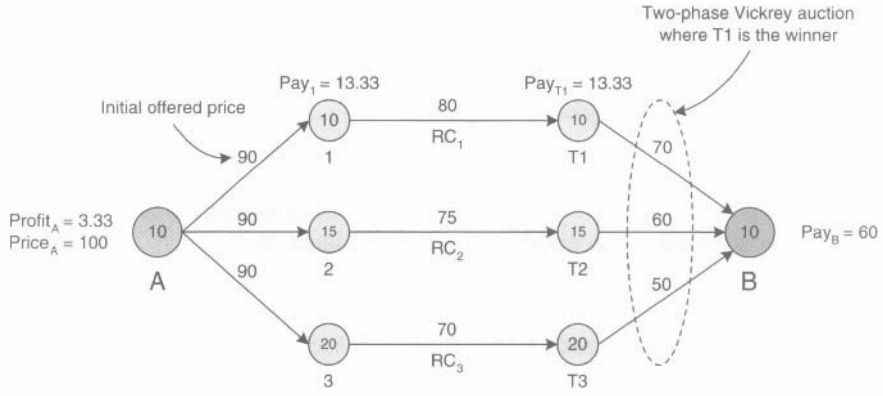


Figure 28.7 An example of the auction process in request forwarding [138].

Wongrujira and Seneviratne [352] also proposed a similar auction based charging scheme for forwarding nodes on a path from a requesting peer to a data source. However, they pointed out an important observation that some potential malicious peers could try to reduce the profits of other truthful peers by dropping the price messages. To mitigate this problem, a reputation system is introduced in that every peer maintains a history of interactions with other peers. The reputation value of a peer is increased every time a message is forwarded by such a peer. On the other hand, if an expected message exhibits a timeout, the responsible peer's reputation value is decreased.

Wang and Li [346] also considered a similar problem in which a peer needs to decide how much to charge for forwarding data. Instead of using auction, a comprehensive utility function is used. The utility function captures many realistic factors: the quantitative benefits of forwarding data, the loss in delivering such data, the cost and the benefit to the whole community. With this utility function, an upstream peer has the incentive to contribute its forwarding bandwidth while a downstream peer is guided toward spending the upstream bandwidth economically. Furthermore, a reinforcement learning component is incorporated so that each peer can dynamically adjust the parameters in its utility function so as to optimally respond to the current market situations.

Sanghavi and Hajek [308] observed that in a typical auction based pricing mechanism as described above, there is a heavy communication burden on the peers. Indeed, the entire set of user preferences has to be communicated from a peer to the auctioneer. Sanghavi and Hajek then analytically derived a class of alternative information mechanisms that can significantly reduce the communication overhead. Specifically, each peer's bid is only a single real number in each case, instead of an entire real-valued function.

Hausheer and Stiller [146] studied a completely decentralized auction approach for electronic P2P pricing of goods in a system called PeerMart (which is built on top of Pastry [301]). The key idea is the usage of a broker set which comprises other peers in the electronic marketplace. Specifically, a broker set consists of peers' whose IDs are closest to the ID of the good in the auction. Each of these peers then potentially acts as the auctioneer in the selling process. The advantage of the broker set based method is that in case a particular peer in the set is faulty (or even malicious in the sense that it does not respond to auction requests), another member in the set can take up the role of auctioneer. An example is shown in Figure 28.8.

28.2.1.9 Exchange Based Systems Motivated by the fact that any payment/credit based system entails a significant transaction and accounting overhead, Anagnostakis and Greenwald [15] proposed an exchange based P2P file sharing system. The fundamental premises is that any peer gives priority to exchange transfers. That is, in simple terms, any peer is willing to send a file to a peer that is able to return a desired file. However, based on this idea, it is incorrect to consider 2-way exchanges only. Indeed, a "ring" of exchange involving two or more peers, as shown in Figure 28.9, is also a proper P2P file transfer.

In the exchange based P2P file sharing system, each peer maintains a data structure called *incoming request queue* (IRQ). Now, a crucial problem is—how each peer can determine whether an incoming request should be entertained, i.e., whether such a request comes from some peer on a ring of exchange requests. It is obviously computationally formidable to determine all the potential multi-peer cycles. Fortunately, Anagnostakis and Greenwald [15] argues that based on simulation results, in practice a peer only needs to check for cycles with up to five peers.

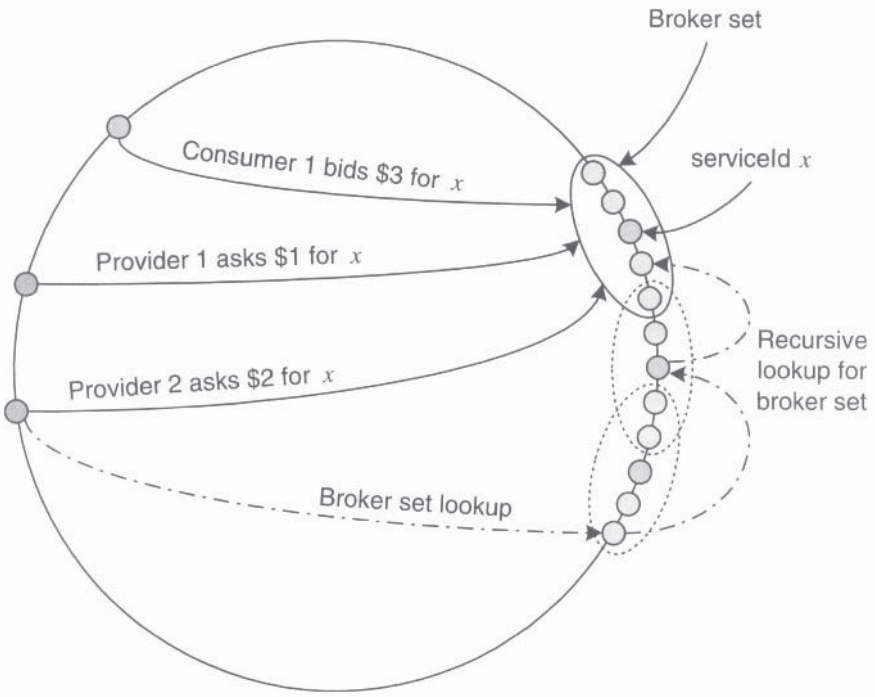


Figure 28.8 An example of fully decentralized auction [146].

Each peer uses a data structure called *request tree* to check for potential request-cycles. For example, as we can see in Figure 28.10, a peer A decides to entertain a request for file object o_2 because A finds that peer P_9 possesses an object that is needed by A. Based on this checking mechanism, the incoming requests are prioritized. Simulation results indicate that the proposed exchange based mechanisms are effective in terms of file object download time.

28.2.2 Media Streaming Systems

In this section, we describe several interesting techniques for providing incentives in a P2P media streaming environment. Broadly speaking, there are two different structures employed in P2P media streaming: asynchronous layered streaming and synchronous multicast streaming.

28.2.2.1 Layered Many-to-One Streaming Xu *et al.* [353] proposed a fully distributed differentiated admission control protocol called DAC_{p2p}. In their model, *each requesting* peer needs multiple supplying peers to send different layers of media data. That is, from a topological perspective, each streaming session involves one

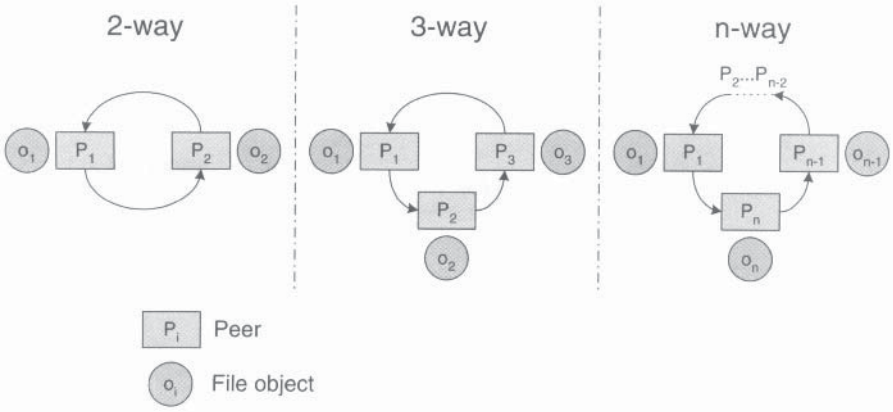


Figure 28.9 Different feasible forms of exchanges [15].

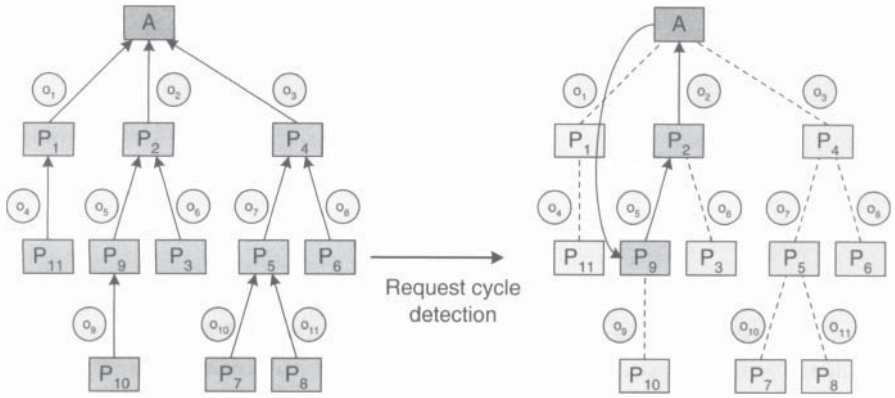


Figure 28.10 Request cycle detection using the request tree data structure maintained at each peer [15].

single recipient and multiple sources, structured as an two-level inverted tree. Of course, each supplying peer may also be a recipient of another logically different streaming session. With this streaming model, each requesting peer needs to actively contact several supplying peers in order to start a session.

This streaming model is based on a practical observation—peer requests are asynchronous and each peer's communication capability is different (i.e., heterogeneous streaming capabilities), as illustrated in Figure 28.11 where peers with different communication supports (e.g., DSL, dial-up, etc.) initiate streaming requests at different times.

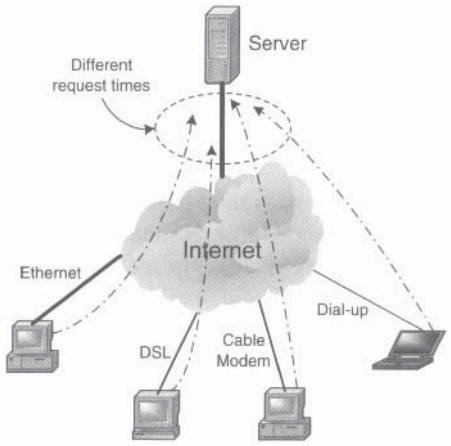


Figure 28.11 Asynchrony and heterogeneity of media streaming peers [80].

A layer-encoded media streaming process is assumed, as shown in Figure 28.12. As can be seen, each peer performs buffered playback so that the received media data are kept in a buffer and thus, can be used for streaming to other later-coming peers. For example, H_1 initiates a streaming session first and thus, it is served solely by the server. Peer H_2 starts its session next and so it can request H_1, which has buffered some media data, together with the server to send it the required data. Similarly, H_3 can stream from H_1 and H_2 without the server as it starts its session just-in-time to use the buffered data from the two earlier peers. On the other hand, H_4, which starts too late, cannot stream from H_1 and H_2. Instead, it receives media data from H_3 and the server.

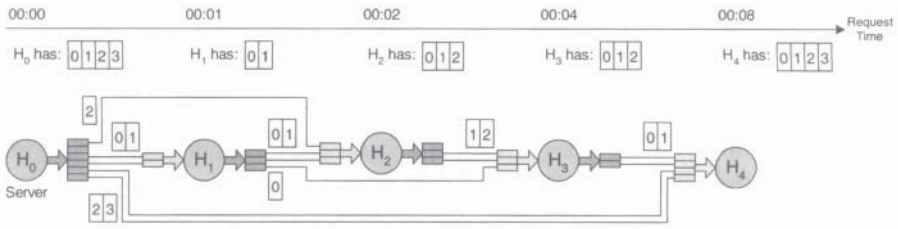


Figure 28.12 Layered streaming with buffering for serving asychronous requests [80].

Each potential supplying peer has only a limited capacity and thus, an admission control mechanism is needed. Peers in the system are classified into N classes according to the different levels of uploading bandwidth available at the peers. Each potential supplying peer P_S maintains an admission probability vector: $(Pr[1], Pr[2], \ldots, Pr[N])$.

Here, a smaller index represents a class with a larger uploading bandwidth. Suppose P_S is itself a class-k peer. Then, its probability vector is initialized as follows:

- For $1 \leq i \leq k$, $Pr[i] = 1.0$;
- For $k < i \leq N$, $Pr[i] = \frac{1}{2^{i-k}}$;

Thus, P_S always grants media streaming requests from a higher class peer (i.e., one that has a larger uploading bandwidth). Notice that this is similar in spirit to the incentive approach used in BitTorrent [70]. For requests from lower class peers, P_S may serve them as governed by the respective probabilities $Pr[i]$ in the vector. If P_S has not served any request during a certain period of time, then the admission probabilities of lower class peers will be increased.

Similar to the case in BitTorrent, each peer has the incentive to report a higher uploading bandwidth because doing so will increase its probability of admission when it needs to initiate a media streaming session. Xue *et al.* [355] extended the DAC_{p2p} to a wireless environment. The key idea in the extension is to exploit the spatial distribution of mobile devices to form clusters. Users in a cluster interact using the DAC mechanism.

Habib and Chuang [139] also explored a similar idea in providing differentiated peer selection to participating peers. Specifically, a peer has only a limited set of choices (with possibly low media quality) if it behaves selfishly in the system. The degree of selfishness is reflected by a score known to other peers. The score is increased if the peer contributes to other peers, and is decreased if it refuses the requests of other peers. Based on a practical emulation study using the PROMISE [148] streaming system implemented on top of PlanetLab [282], Habib and Chuang [139] found that the proposed incentive scheme is effective in enhancing the performance of the system.

28.2.2.2 Multicast One-to-Many Streaming Ngan *et al.* considered an application level multicast system for video streaming. The system is based on SplitStream [55] which in turn is built on top of Pastry [301]. The multicast system considered critically relies on a payment based scheme. Specifically, there are five components:

- **Debt Maintenance:** When a peer A forwards video streaming data to a downstream peer B, B owes A a unit of debt.
- **Periodic Tree Reconstruction:** The multicast tree is reconstructed periodically in order to avoid prolonged unfair connections among peers. An unfair connection is one between a well-behaved peer and a selfish peer.
- **Parental Availability:** Any new peer can obtain location and addressing information about any potential parent peers in the multicast tree. Thus, the new peer can identify a potential selfish parent if the latter consistently refuses connection.

- **Reciprocal Requests:** In the system, any two well-behaved peers are expected to have an equal chance of being parent or child in any given multicast tree.
- **Ancestor Rating:** This is a generalization of the Debt Maintenance component. Here, debts are also accounted for all ancestors of a peer. Specifically, all nodes on a path in forwarding data from the source to a peer are credited or debited in cases where expected data is successfully received or not, respectively.

Simulation results indicate that a selfish free-rider is effectively penalized in terms of the amount of video streaming data received.

Chu and Zhang [158] also considered a multicast based streaming environment. The streaming process is synchronous and is supported by a multiple description codec (MDC), in which a server provides several different stripes of video with different quality. A key feature in this streaming environment is that during each streaming session, multiple multicast trees are used, each of which is for sending different stripes of video. This is illustrated in Figure 28.13. With such a streaming structure, a peer can logically join different trees simultaneously at a *different position* in each tree. Specifically, when a peer joins a certain multicast tree at a higher level (e.g., peer A in the tree for stripe II), it needs to provide a larger uploading bandwidth to serve the lower level peers in the same tree. On the other hand, a peer can also join a tree as a leaf so that it becomes a pure recipient in the tree.

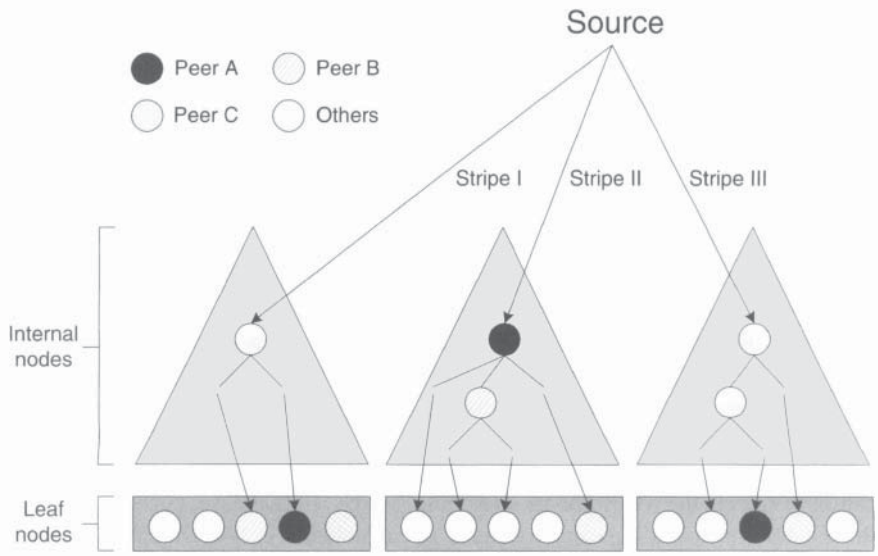


Figure 28.13 Layered video streaming using multiple multicast trees [158].

With this model, Chu and Zhang [158] then studied the effects of different degrees of altruism. They used a parameter $K = f/r$, where f is the total uploading band-

width provided by a peer and r is its total downloading bandwidth. Thus, a larger value of K indicates a higher degree of altruism for the peer. Similar to many other P2P sharing systems described above in this chapter, a peer with a higher value of K can enjoy a better performance (in terms of media quality in the streaming application). Simulation results indicate that a small average value of K (e.g., 1.5) can already improve the overall performance of the whole system.

Shrivastava and Banerjee [341] demonstrated that streaming based on a multicast structure could be a result of natural selection. The key idea is depicted in Figure 28.14. The left part of the figure illustrates a situation where multiple peers are sharing the capacity of a single server. As a result, each peer can only enjoy a small downloading data rate. However, when peers are organized as a multicast tree, based on strategic *natural selection* (detailed below), each peer can enjoy a much larger downloading rate.

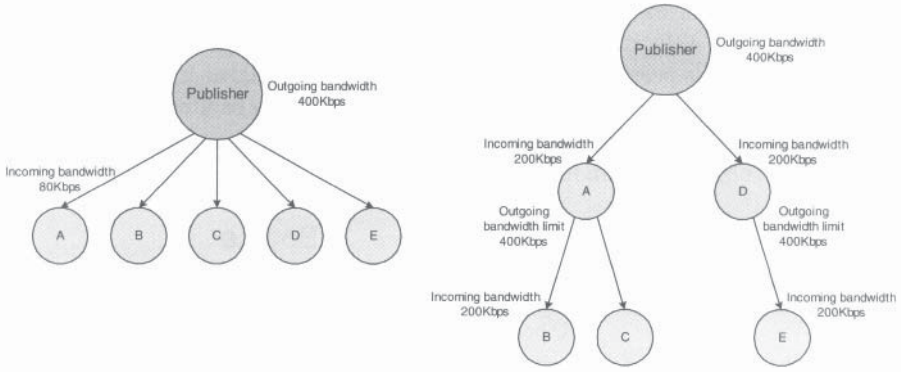


Figure 28.14 A multicast streaming structure is better off for every peer [341].

The natural selection process can be illustrated in Figure 28.15. Here, initially the root is the only source in the system and thus, peer A selects the root as the source, enjoying a downloading rate of 500Kbps. Now, when a new peer B joins the system, peer A has basically two choices: (1) serve peer B; or (2) do not serve peer B. To implement the second choice which seems to be a more favorable one, peer A can declare to the BSE that its uploading rate is 0 or a value smaller than 250Kbps (which is half of the capacity of the root). However, in doing so, peer B has no choice but naturally selects the root to be its streaming source. In that case, peers A and B will share the root's uploading capacity and thus, each obtains only 250Kbps data rate. On the other hand, in anticipation of such an actually unfavorable outcome, peer A should instead strategically declare its uploading bandwidth to be 300Kbps, which is slightly higher than the capacity declared by the root. Consequently, peer A can continue to enjoy a high downloading rate from the root, at the expense of its uploading of data to peer B at a rate of 300Kbps.

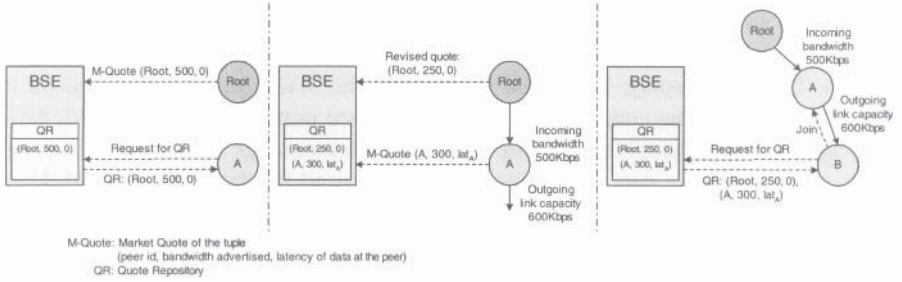


Figure 28.15 An illustration of the strategic natural selection process in connecting streaming sources and destinations (BSE is the bootstrap entity providing service information) [341].

Ye and Makedon [361] proposed a useful detection and penalty scheme to tackle the existence of selfish peers in a multicast streaming session. They observed that a selfish peer may lie to other peers in that it claims its uploading bandwidth is large so that it can enjoy a higher probability of being admitted into a streaming session or enjoy a higher quality of media data. The key of the detection mechanism is that a downstream peer in a multicast tree returns a "streaming certificate" back to its parent peer. For example, as shown in Figure 28.16, peers P_4, P_5, and P_6, send streaming certificates **SCert**(P_i,P_3) to the parent peer P_3 ($i = 4, 5, 6$). The certificates are sent periodically and are time-stamped with authentication. Thus, a higher level peer, e.g., P_1, can periodically check whether its children peers (i.e., P_2 and P_3) are selfish by asking for certificates they have received (if any) from their own children peers. If a peer cannot produce such a certificate, the higher level peer can then remove such a potentially selfish peer from the tree. The removal process is manifested as a termination of media data transmission.

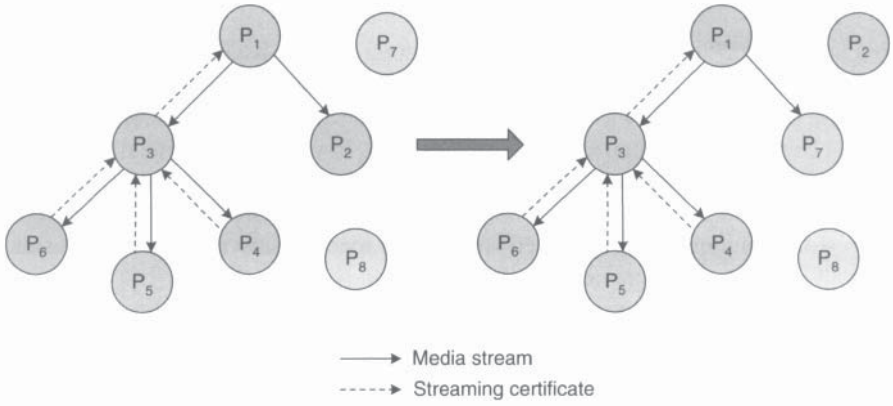


Figure 28.16 Detection and removal of a selfish peer from the streaming multicast tree [361].

Jun *et al.* [176] also explored a similar idea in their proposed Trust-Aware Multicast (TAM) protocol. Targeted for detecting and deterring uncooperative peers which can modify, fabricate, replay, block and delay data, the TAM protocol is based on a message structure that contains four fields: sequence number, timeout period, data payload, and cryptographic signature. The sequence number is used for detecting duplicated or missing data. The timeout period is used for detecting delayed data. Thus, a selfish (or even malicious) peer can be identified by its children peers in the multicast tree. Different from the approach suggested by Ye and Makedon [361], the children peers are responsible for reporting such suspicious selfish peers to the root (or the server) in the tree. Jun *et al.*'s scheme is also more flexible in that even upon receiving such "negative reports", the root may not discard such suspected selfish peers immediately. Instead, the root keeps track of a trust metric for each peer in the tree. A negative report only decreases the trust value. Only when the trust value falls below some threshold, the suspected selfish peer is discarded from the tree and the peers under its sub-tree are relocated.

28.3 INCENTIVE ISSUES IN WIRELESS P2P SYSTEMS

Wireless P2P systems [153] are proliferating in recent years. Thanks to the widely available hot-spot wireless environments, users handheld devices can work with each other in an *ad hoc* and impromptu manner. As will be evident in this section, many techniques designed for wired environments are also applied in a wireless system in a similar manner. Nevertheless, there is a unique challenge in a wireless environment, namely the connectivity issue. Furthermore, there is also one more dimension of cost incurred in each wireless P2P user—the energy expenditure, which is of prime concern to the user as wireless devices are largely powered by batteries.

28.3.1 Routing and Data Forwarding

In a wireless P2P system, the connectivity among peers is itself a bootstrap sharing problem. Indeed, if wireless users are unwilling to cooperate in performing routing and data forwarding, the wireless network can be partitioned so that service providers cannot be reached by potential service consumers. In view of this critical challenge, there has been a plethora of important research results related to incentive issues for ad hoc routing and data forwarding. In the following, we briefly cover several interesting techniques that are based on payment mechanisms, auction mechanisms, reputation systems, and game theoretic modeling.

Ileri2005 *et al.* [162] proposed a payment based scheme for enticing devices to cooperate in forwarding data for other devices in the network. The payment is not in monetary terms but in terms of of bits-per-Joule. Specifically, the utility of a user i in the network is defined as:

$$u_i(p_i) = \frac{T_i(p_i)}{p_i} \tag{28.10}$$

where u_i is the utility, p_i is the transmit power, and T_i is the throughput. That is, the utility is equal to the average amount of data received per unit energy expended, also in bits-per-Joule.

The payment system also involves an access point in the wireless network. Specifically, the access point also tries to maximize its revenue by using two parameters, μ and λ, judiciously. Here, λ is the unit price of service provided by the access point to any device in the network. On the other hand, μ is the unit reimbursement the access point provides to any device which has helped forwarded other devices' traffic. The access point's revenue is therefore given by:

$$\rho = \sum_{\text{all users } i} \lambda T_{\text{in}_i} - \sum_{\text{all forwarders } j} \mu T_{ja}^{\text{eff-for}} \tag{28.11}$$

where T_{in_i} is the service provided by the access point to user i and $T_{ja}^{\text{eff-for}}$ is the service forwarded by a forwarder j to the access point. The situation is as shown in Figure 28.17.

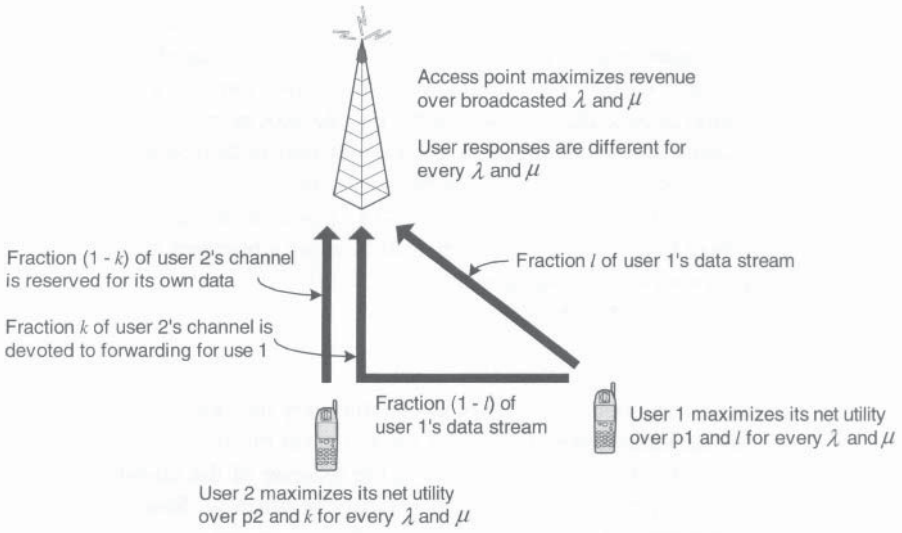


Figure 28.17 Charging of network service and reimbursement of data forwarding [162].

Simulation results indicate that the proposed service reimbursement scheme generally improves the network aggregate utility.

Salem *et al.* [305] also considered a payment based scheme for encouraging cooperation. However, their scheme involves real monetary costs and a payment clearance infrastructure (e.g., a billing account for each user). The charging and rewarding scheme is similar to that we described above—a forwarder will get reimbursed and a normal user using network service will get charged.

Marbach and Qiu [234] investigated a similar problem based on individual device pricing. However, the main differences are that each device is allowed to freely decide how much to charge for forwarding traffic (in previous researches, usually the unit price is the same for all device) and there is no budget constraint on all the devices.

Wang and Li [346] proposed an auction based scheme similar to the work by Gupta and Somani [138] that we described in Section 28.2.1.8. Specifically, each device in a wireless ad hoc network declares its cost for forwarding data when some other device wants to initiate a multihop transmission. After considering all possible paths that are able to reach the destination device, the least cost path is chosen and the devices on the path are paid for their forwarding. Again using a VCG based analysis, it is shown that the dominant action for each device is to report its true cost of forwarding. Wang and Li also showed that no truthful mechanism can avoid collusion between two neighboring devices in the forwarding auction game.

Buchegger and Le Boudec [45] observed that while economic incentives such as payment approaches can entice selfish users to help in routing and forwarding data but may not be able to handle other types of misbehaviors such as packet dropping, modification, fabrication, or timing problems. Thus, they proposed to use a reputation system in which every user provides "opinion" data to the network based on observing the behaviors of neighboring devices. After a user device has gathered such opinions (both from itself or from others, i.e., second-hand information), it can carry out a Bayesian estimation so as to classify the neighboring devices as malicious or normal. A neighboring device that is identified as a malicious user is then isolated from the network by rejecting its routing and forwarding requests.

Felegyhazi *et all.* [116] reported an interesting game theoretic analysis of the forwarding problem in ad hoc networks. Instead of using a payment based strategy, the model employs a purely utility concept in that a device's utility is equal to its payoff when it acts as a data source (i.e., the sender of a multihop traffic), minus the cost when it acts an intermediate device (i.e., a forwarder of other sender's traffic) in any time-slot. Here, both the payoff and cost are defined in terms of data throughput. Thus, an important assumption in this model is that only the sender has a positive payoff, while all the intermediate devices enjoy no payoff but just incur forwarding costs. Specifically, the destination device (i.e., the receiver of the multihop traffic) also enjoys no payoff. This may not conform to a realistic situation. Simulations were done to estimate the probability that the conditions for a cooperative equilibrium hold in randomly generated network scenarios.

28.3.2 Wireless Information Sharing Systems

Wolfson *et al.* [350] investigated an interesting opportunistic wireless information exchange problem in which a moving vehicle transmits the information it has collected to encountered vehicles, thereby obtaining other information from those vehicles in exchange. The incentive mechanisms suggested are based on virtual currency. Specifically, each mobile user carries some virtual currency in the form of a protected

counter. Two different mechanisms are considered: producer-paid and consumer-paid. In the former, the producer of information pays while the consumer pays in the latter. The price P of a piece of information item (e.g., availability information about a parking lot) is given by:

$$P = E - t - \frac{d}{v} \tag{28.12}$$

where E is the gross valuation of the information item, t is the time elapsed since the information item is created, d is the distance to travel before the user of the information can reach the relevant location (e.g., the parking space), and v is the speed of the user. Simulation results under a simple situation where there are only one consumer and two parking spaces indicate that the proposed incentive mechanisms are effective.

Yeung and Kwok [363] considered an interesting scenario in wireless data access: a number of mobile clients are interested in a set of data items kept at a common server. Each client independently sends requests to inform the server of its desired data items and the server replies in the broadcast channel. Yeung and Kwok investigated the energy consumption characteristics in such a scenario.

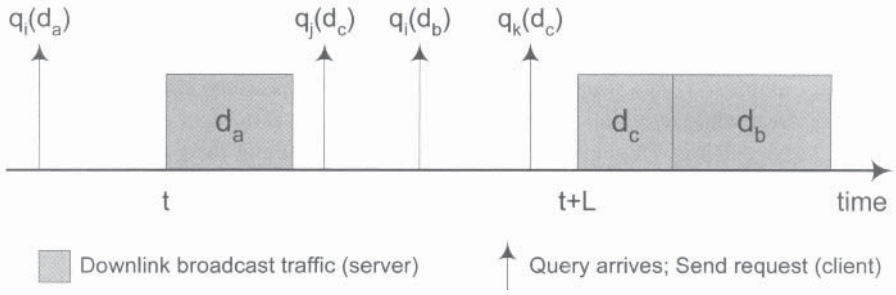


Figure 28.18 System model for wireless data access [363].

Figure 28.18 depicts the system model for wireless data access. It consists of a server and a set of clients, N. The clients are interested in a common set of data items, D, which are kept at the server. To request a specific data item, d_a, client i is required to inform the server by sending an uplink request, represented by $q_i(d_a)$. The server then replies with the content of the requested data item, d_a, in the common broadcast channel. This allows the data item to be shared among different clients. As illustrated in Figure 28.18, both clients j and k request the same data item, d_c, in the second interval. However, the server is required to broadcast the content of d_c only *once* in the next broadcast period, which reduces the bandwidth requirement.

To successfully complete a query, a client expends its energy in two different parts: (1) informing the server of the desired data item, E_{UL}; and (2) downloading the content of the data item from the common broadcast channel, E_{DL}. The energy cost of sending a request to the server is represented by E_s. If a client sends a request for each query, then E_{UL} would be the same as E_s. However, we show that clients

need not to send requests for each query even without caching. This implies that E_{UL} does not necessarily equal E_s. In general, the total energy required to complete a query, E_Q, is given by:

$$E_Q = E_{UL} + E_{DL} \tag{28.13}$$

It is assumed that E_{DL} is the same for all clients, but its value depends on the size of a data item. In practice, E_s is a function of various quantities, including spatial separation, speed, instantaneous channel quality, bit-error rate requirement, etc. For simplicity, however, E_s is also assumed to be a fixed quantity.

Based on this model of wireless data access, a novel utility function for quantifying performance is defined as follows:

$$U_i = \frac{E_{total}}{E_Q^i} \tag{28.14}$$

The objective is then to reduce the amount of energy consumed in the query process such that *every* client's utility (Equation (28.14)) is increased.

Based on the utility function, Yeung and Kwok formulate the wireless data access scheme as a non-cooperative game—*wireless data access (WDA) game*. Their game theoretic analysis shows that even without using client caching, the clients do not need to send requests to the server. Simulation results also indicate that the proposed scheme, compared with a simple always-request one, increases the utility and lifetime of *every client* while reducing the number of requests sent, at the cost of slightly larger average query delay. They also compared the performance of the proposed scheme with two popular schemes that employ client caching. The simulation results show that caching only benefits clients with high query rates while resulting in both shorter lifetime and smaller utility in other clients.

28.3.3 Network Access Sharing

Efstathiou and Polyzos [101] studied the problem of building a federation of wireless networks using a fully autonomous P2P approach. Specifically, in their system model, there are multiple WLANs, each of which is considered to be completely autonomous. When a user of one WLAN enters the domain of a different nearby WLAN, the latter would also admit the user based on a reciprocity idea. To achieve this, each WLAN is equipped with a domain agent (DA) which is responsible for managing the roaming of foreign users. Each DA maintains a counter of tokens, which is increased when a foreign user is admitted to its WLAN, and is decreased when a local user travels to a foreign WLAN. The DAs of different WLANs interact with each other in a pure P2P fashion. Thus, the advantage of this approach is that there is no need to set up prior *pairwise* administrative agreements among different WLANs.

Based on a prototype built using Cisco WLANs, it is found that the proposed P2P based roaming scheme is efficient.

Kang and Mutka [181] considered an interesting problem in which peers share the access cost of wireless multimedia contents. Specifically, one peer in the network

serves as a proxy which pays a network server in order to download some multimedia contents in a wireless fashion. Other peers in the system then share the contents without incurring any cost. This is achieved by having the proxy broadcast the received multimedia data to the peers within its transmission range. For other distant peers, rebroadcasting by the edge peers is employed to serve them. This is illustrated in Figure 28.19 (left side).

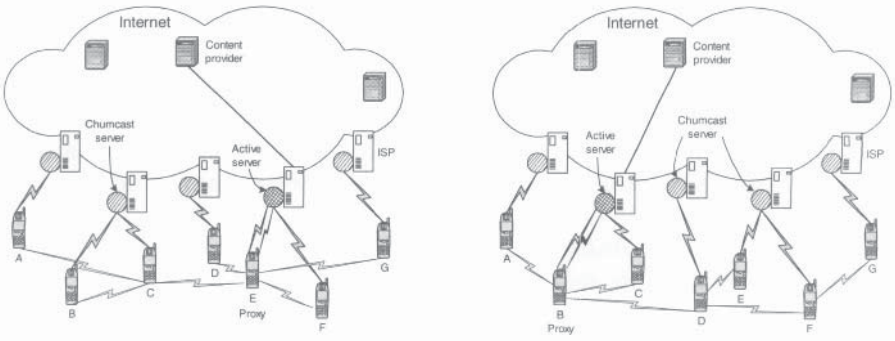


Figure 28.19 Illustration of network access cost sharing (left), and round-robin scheduling of proxy (right) [181].

This idea of cost sharing in wireless data access is called CHUM (cooperating ad hoc networking to support messaging). A key component in such sharing is the cost sharing mechanism. In the CHUM system, the peers take turn, in a round-robin manner, to serve as the proxy. This is illustrated in Figure 28.19 (right side). It is assumed that the peers have the incentive to follow this round-robin rule based on the reciprocity concept. Simulation results indicate that 80% of network access cost is saved even with just six peers in the system.

28.4 DISCUSSION AND FUTURE WORK

We have seen that in both wired and wireless systems, the major techniques for providing incentives are: payment (virtual or real), exchange (barter), reciprocity (pairwise), reputation (global), and game theoretic utility. Payment based systems work by exploiting a user's incentive in increasing or even maximizing its "revenue". However, such an incentive may not be appropriate in some practical situations. For instance, a cellular phone user may not be interested in his or her income (from such sharing) but care more about the quality of service derived from the device. Exchange based systems fit very well in file sharing applications because users have strong incentives for trading interested files (e.g., music files). For other applications such as forwarding of data (e.g., in wireless ad hoc networks), it is debatable as to whether in practice a user would be interested in exchanging data forwarding

capabilities. Reciprocity (in a pairwise manner) is similar in spirit as a exchange based mechanism. The major difference is that exchange is usually memoryless in that every exchange transaction is treated as a rendezvous event, while a general reciprocity is achieved when devices help each other during different points in time. Thus, such a difference leads to the requirement that users have to keep memory about the prior transactions so that users can "pay back" each other. Due to this history based feature, reciprocity mechanism suffers from one drawback—a user may not be still in the system when he/she needs to pay back. Such a possible "future loss" may deter a user from genuinely contributing to the community for the fear of not getting deserved pay back. Reputation based systems can be seen as a generalized form of reciprocity. Specifically, while reciprocity is about a particular user pair, a reputation value is a global assessment perceived by all users in the sense that the reputation value is computed by using observations made by many different users. By nature, similar to reciprocity mechanism, reputation systems require substantial memory, centralized or distributed, for recording the reputation values. Thus, it seems that such a system is more suitable for situations where there is a persistent entity, which is logically external to the P2P system, for keeping track of the reputation values. For example, such an entity may be a centralized auctioneer in a electronic auction community, or the base station (access point) in a wireless network. Game theoretic incentive mechanisms are convincing in the sense that the utility function used can usually cover a multitude of important metrics. But the problem is that it is sometimes difficult to achieve an efficient distributed implementation of the resultant protocols.

Perhaps except in a exchange based system which involves "stateless" interactions, all other incentive mechanisms could suffer from tampering or fabrication of the "incentive parameter" used: the money (virtual or real) in a payment based system, the reciprocity metric, and the reputation value. Thus, additional mechanisms, usually based on cryptographic techniques, are needed to guard against such potential malicious attacks to the incentive mechanisms. In particular, whitewashing is widely considered as a very low cost technique for a selfish or malicious user to work around the incentive scheme.

Obviously, there are plenty of room for future research about incentive mechanisms in P2P sharing environments. Most notably, revenue maximizing [364] in a hybrid P2P system (e.g., the so-called converged wireless architecture where an infrastructure based cellular network is tightly coupled with P2P WLANs) is of a high practical interest because we have witnessed more and more cellular subscribers try to share their resources without the intervention of the cellular service provider. In a data sharing environment, server peer selection [204] is another important direction because we believe that users care more about the quality of service achieved than about the revenue or cost they incur in participation. In economics terms, people, especially wired or wireless game players, are quite inelastic about the costs. Nevertheless, energy conservation [206, 205] is still of a prime concern in any wireless P2P sharing network because energy depletion cannot be compensated in any way

by increased revenue generated in a payment based sharing system. Thus, perhaps in a game theoretic setting, we should incorporate energy expenditure in the utility function. Finally, topology control [207] in a wired (overlay networks) or wireless (ad hoc networks) is also important in the sense that sharing is usually interest-based, meaning that users naturally form clusters with similar interests, and as such, related users would be more cooperative in following the incentive protocols. Consequently, building an interest-based sharing topology could be helpful in enhancing the effectiveness of sharing.

28.5 SUMMARY

In this chapter, we have presented a detailed survey of incentive techniques for promoting sharing (discrete or continuous data) in a peer-to-peer system. We have considered well known mechanisms that are proposed and have been deployed in the Internet environment. The techniques used can be classified into: payment based, exchange-based, reciprocity, reputation, and game theoretic. These techniques are also applicable in general to a wireless environment. However, a wireless P2P system, while still in its infancy, has a unique challenge—the connectivity among devices is by itself a crucial "sharing" problem (sharing of energy and bandwidth). Much more work needs to be done in related problems such as revenue maximizing (or pricing) in a hybrid wireless P2P system, intelligent peer selection in a data sharing network, energy aware incentive mechanisms, and interest based topology control.

PROBLEMS

28.1 What is the major difference between a barter based incentive scheme and a reputation based incentive scheme?

28.2 What is the key difficulty in implementing a payment based incentive scheme?

28.3 Explain why a wireless P2P system needs incentives in maintaining connectivity among the devices.

This Page Intentionally Left Blank

APPENDIX A

OPTIMALITY OF MINIMUM DISTANCE DECODER

In this appendix, we shall prove that minimum distance decoding minimizes the error probability. Consider a demodulator for a M-ary modulator. Let $\mathcal{S}$ denote the signal space containing the M constellation points and $\mathcal{R} = \mathcal{R}_1 \bigcup \mathcal{R}_2 \bigcup ... \bigcup \mathcal{R}_M$ be a *decision region* of the demodulator. Note that the decision region $\mathcal{R}$ is in fact a partition of the signal space into M regions where $\mathcal{R}_i \bigcap \mathcal{R}_j$ = (no intersection between regions). Figure A.1 illustrates one example of a decision region.

Given a decision region, the decision rule is to pick $\vec{s}_m$ if the received vector $\vec{y}_n$ falls in region $\mathcal{R}_m$. Hence, any specific decoding scheme can be characterized by a corresponding decision region $\mathcal{R}$. Given that $\vec{s}_1$ is transmitted, the error probability is given by:

$$P_e\left(\vec{s}_1\right) = \int_{R_1^c} p\left(\vec{y}|\vec{s}_1\right) d\vec{y} = 1 - \int_{R_1} p\left(\vec{y}|\vec{s}_1\right) d\vec{y}$$

Wireless Internet and Mobile Computing. By Yu-Kwong Ricky Kwok and Vincent K. N. Lau
Copyright © 2007 John Wiley & Sons, Inc.

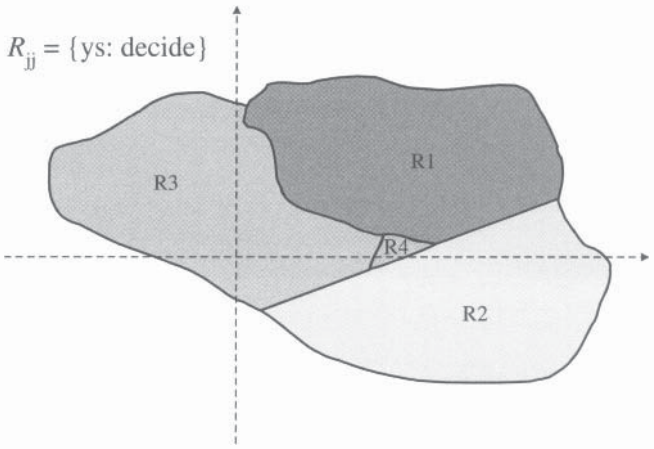


Figure A.1 Illustration of decision regions for a digital demodulator with respect to quartenary modulator.

where $\mathcal{R}_1^c = \mathcal{S} \backslash \mathcal{R}_1$. Hence, the average error probability (averaged over all possible transmitted constellation points) is given by:

$$P_e = \sum_{j=1}^{M} p\left(\vec{s}_j\right) \int_{R_j^c} p(\vec{y}|\vec{s}_j) d\vec{y} = 1 - \sum_{j=1}^{M} p\left(\vec{s}_j\right) \int_{R_j} p(\vec{y}|\vec{s}_j) d\vec{y}$$

Since each received vector $\vec{y}$ must belong to one and only one decision region, we should assign $\vec{y}$ to $\mathcal{R}_j$ (for minimizing the average error probability) if $p(\vec{y}|\vec{s}_j)p(\vec{s}_j)$ is maximized. If we assume equi-probable constellations (i.e., $P(\vec{s}_j) = 1/M$), the optimal demodulation rule is given by:

$$\hat{s} = \underset{\vec{s} \in \left\{\vec{s}_1, \ldots, \vec{s}_M\right\}}{\arg\max} \; p\left(\vec{y}\,|\vec{s}\right).$$

This is called the *maximal likelihood demodulation*. Since the channel is AWGN, the received vector is given by:

$$\vec{y} = \vec{s} + \vec{z}$$

and

$$p\left(\vec{y}|\vec{s}\right) = K \exp - \left(\frac{1}{2\sigma_n^2} \left\|\vec{y} - \vec{s}\right\|^2\right)$$

where K is some constant and σ_n^2 is the variance of the channel noise. Hence, the solution of the maximal likelihood decoding becomes:

$$\begin{aligned}\hat{s} &= \underset{\vec{s}\in\left\{\vec{s}_1,\ldots,\vec{s}_M\right\}}{\arg\min} \left\|\vec{y}-\vec{s}\right\|^2 = \underset{\vec{s}\in\left\{\vec{s}_1,\ldots,\vec{s}_M\right\}}{\arg\max} \left[2\vec{y}\bullet\vec{s}-\left\|\vec{s}\right\|^2\right] \\ &= \underset{\vec{s}\in\left\{\vec{s}_1,\ldots,\vec{s}_M\right\}}{\arg\max} \left[2\int_0^{T_s} y(t)\,s(t)\,dt - \int_0^{T_s} s^2(t)\,dt\right] \end{aligned} \tag{A.1}$$

In other words, the minimum distance decoding is equivalent to the maximal likelihood decoding which is equivalent to the correlator structure as illustrated by the above equation.

This Page Intentionally Left Blank

REFERENCES

1. Bluetooth Specification Version 1.1. *Specification of the Bluetooth System*. Bluetooth SIG, February 2001.
2. Bluetooth Specification Version 1.2. *Specification of the Bluetooth System*. Bluetooth SIG, November 2003.
3. IrDA Serial Infrared Physical Layer Specification Version 1.4. *Infrared Data Association Serial Infrared Physical Layer Specification*. Infrared Data Association, May 2001.
4. 3GPP. Physical channels and mapping of transport channels onto physical channels (FDD). *3GPP Technical Specifications, TS 25.211*, 2004.
5. 3GPP. Spreading and modulation (FDD). *3GPP Technical Specifications, TS 25.213*, 2004.
6. IEEE Standard 802.11. *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*. IEEE, 1999.
7. IEEE Standard 802.11a. *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band*. IEEE, April 1999.
8. IEEE Standard 802.11b. *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher speed physical layer (PHY) extension in the 2.4G Hz band*. IEEE, 1999.

Copyright © 2007 John Wiley & Sons, Inc.

9. IEEE Standard 802.11g. *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz Band*. IEEE, 2003.

10. L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala. Bluetooth and WAP Push Based Location-Aware Mobile Advertising System. In *Proc. MobiSys*, pages 49–58, June 2004.

11. Y. Adam, B. Fillinger, I. Astic, A. Lahmadi, and P. Brigant. Deployment and Test of IPv6 Services in the VTHD Network. *IEEE Communications Magazine*, 42(1):98–104, January 2004.

12. H. Afifi and D. Zeghlache (eds.). *Applications and Services in Wireless Networks*. Kogan Page, 2002.

13. S. Aign and K. Fazel. Temporal & Spatial Error Concealment Techniques for Hierarchical MPEG-2 Video Codec. In *Proc. Globecom*, 1995.

14. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, March 2002.

15. Kostas G. Anagnostakis and Michael B. Greenwald. Exchange-Based Incentive Mechanisms for Peer-to-Peer File Sharing. In *Proc. 24th Int'l Conference on Distributed Computing Systems*, 2004.

16. M. Anand, E. B. Nightingale, and J. Flinn. Self-Tuning Wireless Network Power Management. In *ACM MOBICOM*, 2003.

17. Ross Anderson, Haowen Chan, and Adrian Perrig. Key Infection: Smart Trust for Smart Dust. In *Proc. 12th IEEE Int'l Conf. Network Protocols (ICNP 2004)*.

18. Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys*, 36(4):335–371, December 2004.

19. E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin. AIRMAIL: A link-layer protocol for wireless networks. *ACM Wireless Networks*, 1(1):47–60, February 1995.

20. A. Bakre and B. R. Badrinath. Implementation and Performance Evaluation of Indirect TCP. *IEEE Transactions on Computers*, 46(3):260–278, March 1997.

21. B. Bakshi, P. Krishna, H. H. Vaidya, and D. K. Pradhan. Improving Performance of TCP over Wireless Networks. In *Proc. ICDCS*, pages 365–373, May 1997.

22. H. Balakrishnan, S. Seshan, and R. H. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networksa. *ACM Wireless Networks*, 1(4):469–481, December 1995.

23. Constantine A. Balanis. *Antenna Theory: Analysis and Design (2nd edition)*. John Wiley and Sons, 1997.

24. J. Bannister, P. Mather, and S. Coope. *Convergence technologies for 3G networks: IP, UMTS, EGPRS, and ATM*. John Wiley & Sons, 2004.

25. C. Barakat and A. Al Fawal. Analysis of Link-level Hybrid FEC/ARQ-SQ for Wireless Links and Long-lived TCP Traffic. *Performance Evaluation*, 57(4):453–476, August 2004.

26. D. Barbará and T. Imieliński. Sleepers and Workaholics: Caching Strategies in Mobile Environments. In *Proc. 1994 SIGMOD International Conference on Management of Data*, pages 1–12, May 1994.
27. L. Barken. *How Secure Is Your Wireless Network? Safeguarding Your Wi-Fi LAN*. Prentice Hall PTR, 2003.
28. S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile Ad Hoc Networking*. John Wiley & Sons, Ltd., 2004.
29. Jan U. Becker and Michel Clement. The Economic Rationale of Offering Media Files in Peer-to-Peer Networks. In *Proc. 37th Hawaii Int'l Conference on System Sciences*, 2004.
30. B. Bellur, R. G. Ogier, and F. L. Templin. Topology Broadcast Based on Reverse-Path Forwarding (TBRPF). *IETF Internet Draft*, 2001.
31. P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi. CDMA/HDR: A Bandwidth Efficient High-Speed Wireless Data Service for Nomadic Users. *IEEE Communications Magazine*, pages 70–88, July 2000.
32. P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi. Using Channel State Dependent Packet Scheduling to Improve TCP Throughput over Wireless LANs. *ACM/Baltzer Wireless Networks*, 3(1):91–102, 1997.
33. V. Bharghavan, S. Lu, and T. Nandagopal. Fair queuing in wireless networks: Issues and approaches. *IEEE Personal Communications*, pages 44–53, February 1999.
34. BitTorrent. http://www.bittorrent.com/, 2006.
35. Erik-Oliver Blaßand Martina Zitterbart. An Efficient Key Establishment Scheme for Secure Aggregating Sensor Networks. In *Proc. ACM ASIACCS 2006*, pages 303–310, March 2006.
36. R. Blom. An Optimal Class of Symmetric Key Generation Systems. In *Proc. Advances in Cryptology (EUROCRYPT'84), Lecture Notes in Computer Science Volume 209*, pages 335–338.
37. B. H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*, 13(7), July 1970.
38. C. Blundo, A. De Santis, Amir Herzberg, S. Kutten, U. Vaccaro, , and M. Yung. Perfectly Secure Key Distribution for Dynamic Conferences. In *Proc. Advances in Cryptology (CRYPTO 1992), Lecture Notes in Computer Science Volume 740*, pages 471–486, 1993.
39. K. Boman, G. Horn, P. Howard, and V. Niemi. Umts security. *IEEE Electronics & Communication Engineering Journal*, October 2002.
40. B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet. *RFC 2309*, April 1998.
41. J. W. Branch, Jr. N. L. Petroni, L. Van Doorn, and D. Safford. Autonomic 802.11 wireless lan security auditing. *IEEE Security & Privacy Magazine*, May 2004.
42. K. Brown and S. Singh. M-TCP: TCP for Mobile Cellular Networks. *Computer Communications Review*, 27(5):19–43, October 1997.

43. S. Buchegger and J.-Y. Le Boudec. CONFIDANT Protocol: Cooperation of Nodes—Fairness in Dynamic Ad-hoc Networks. In *Proc. IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing*, 2002.

44. S. Buchegger and J.-Y. Le Boudec. Self-Policing Mobile Ad Hoc Networks by Reputation Systems,. *IEEE Communications Magazine*, 2005.

45. Sonja Buchegger and Jean-Yves Le Boudec. Self-Policing Mobile Ad Hoc Networks by Reputation Systems. *IEEE Communications Magazine*, pages 101–107, July 2005.

46. J. Cai and K. L. Tan. Energy-Efficient Selective Cache Invalidation. *Wireless Networks*, 5(6):489–502, December 1999.

47. Seyit A. Çamtepe and Bülent Yener. Key Distribution Mechanisms for Wireless Sensor Networks: A Survey. In *Technical Report TR-05-07, Department of Computer Science, Rensselaer Polytechnic Institute*, March 2005.

48. G. Cao. A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments. In *Proc. 6th Annual International Conference on Mobile Computing and Networking*, pages 200–209, August 2000.

49. G. Cao. On Improving the Performance of Cache Invalidation in Mobile Environments. *Mobile Networks and Applications*, 7(4):291–303, August 2002.

50. G. Cao. Proactive Power-Aware Cache Management for Mobile Computing Systems. *IEEE Transactions on Computers*, 51(6):608–621, June 2002.

51. G. Cao. A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1251–1265, September 2003.

52. Y. Cao and V. O. K. Li. Scheduling Algorithms in Broadband Wireless Networks. *Proc. IEEE*, 89(1):76–87, January 2001.

53. M. Cardei, I. Cardei, and D. Du (eds.). *Resource Management in Wireless Networks*. Springer, January 2005.

54. Jonathan P. Castro. *All IP in 3G CDMA Networks: The UMTS Infrastructure and Service Platforms for Future Mobile Systems*. John Wiley and Sons, 2004.

55. M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in cooperative environments. In *Proc. 19th ACM Symposium on Operating Systems Principles*, October 2003.

56. R. Chakravorty, S. Banerjee, P. Rodriguez, J. Chesterfield, and I. Pratt. Performance Optimizations for Wireless Wide-area Networks: Comparative Study and Experimental Evaluation. In *Proc. MOBICOM*, pages 159–173, September 2004.

57. R. Chakravorty, S. Katti, I. Pratt, and J. Crowcroft. Flow Aggregation for Enhanced TCP over Wide-Area Wireless. In *Proc. INFOCOM*, volume 3, pages 1754–1764, March 2003.

58. H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In *Proc. IEEE Sym. Security and Privacy*, May 2003.

59. Haowen Chan and Adrain Perrig. PIKE: Peer Intermediaries for Key Establishment in Sensor Networks. In *Proc. INFOCOM 2005*.

60. Haowen Chan and Adrian Perrig. Security and Privacy in Sensor Networks. *IEEE Computer*, pages 103–105, October 2003.

61. C. H. Chek. *Coexistence Techniques for Heterogeneous Mobile Devices Operating in Uncoordinated Radio Spectrum*. MPhil Thesis, The University of Hong Kong, 2003.

62. Hsiao-Hwa Chen. *Next Generation CDMA Technologies*. John Wiley and Sons, 2007.

63. T.-W. Chen and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. In *IEEE International Conference on Communications*, 1998.

64. M. X. Cheng, M. Cardei, J. Sun, X. Cheng, L. Wang, Y. Xu, and D. Z. Du. Topology Control of Ad Hoc Wireless Networks for Energy Efficiency. *IEEE Transactions on Computers*, 53(12):1629–1635, December 2004.

65. C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla. Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel. In *Proc. IEEE Singapore International COoference on Networks*, 1997.

66. S. Choi, B. Kim, J. Park, C. Kang, and D. Eom. An implementation of wireless sensor network for security system using bluetooth. *IEEE Transactions on Consumer Electronics*, February 2004.

67. Mooi Choo Chuah and Qinqing Zhang. *Design and Performance of 3G Wireless Networks and Wireless LANs*. Springer Verlag, 2005.

68. I. Clarke, O. Sandberg, B. Wiley, and T. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Proc. Workshop on Design Issues in Anonymous and Unobservability*, July 2000.

69. T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol. In *Proc. IEEE INMIC*, 2001.

70. Bram Cohen. Incentives Build Robustness in BitTorrent. In *Proc. Workshop on Economics of Peer-to-Peer Systems*, June 2003.

71. Atheros Communicatons. 802.11ag: The "clear" choice. *Atheros Communications White Paper*, December 2004.

72. Atheros Communicatons. Atheros extended range xr. *Atheros Communications White Paper*, April 2004.

73. Atheros Communicatons. Super g: Maximizing wireless performance. *Atheros Communications White Paper*, March 2004.

74. C. M. Cordeiro, D. P. Agrawal, and D. H. Sadok. Interference modeling and performance of bluetooth mac protocol. *IEEE Transactions Wireless Communications*, November 2003.

75. Costas Courcoubetis and Richard Weber. Incentives for Large Peer-to-Peer Systems. *IEEE Journal on Selected Areas in Communications*, 24(5):1034–1050, May 2006.

76. D. De Couto, D. Aguayo, J. Bicket, and R. Morris. High-Throughput Path Metric for Multi-Hop Wireless Routing. In *ACM MobiCom*, 2003.

77. Thomas Cover. An achievable rate region for the broadcast channel. *IEEE Transactions on Information Theory*, 31(3):399 – 404, 1975.

78. Thomas Cover and Jay Thomas. *Elements of Information Theory*. John Wiley and Sons, 2006.

79. P. Cuenca, L. Orozco-Barbosa, A. Garrido, D. Quiles, and T. Olivares. A Survey of Error Concealment Schemes for MPEG-2 Video Communications over ATM Network. In *CCECE*, 1997.

80. Yi Cui and Klara Nahrstedt. Layered Peer-to-Peer Streaming. In *Proc. NOSSDAV'03*, pages 162–171, June 2003.

81. G. Davies. *Designing and Developing Scalable IP Networks*. John Wiley and Sons, first edition, 2004.

82. S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 1883*, December 1995.

83. S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. *RFC 2460*, December 1998.

84. Farshid Delgosha and Faramarz Fekri. Threshold Key Establishment in Distributed Sensor Networks Using a Multivariate Scheme. In *Proc. IEEE INFOCOM 2006*.

85. J. Deng and Z. J. Hass. Dual Bust Tone Multiple Access (DBTMA). In *ACM MOBICOM*, 2003.

86. K. Dhananjay and N. Guo. Optimal power allocation for the reverse link in a multimedia ds-cdma system. *Proceedings of IEEE Vehicular Technology Conference VTC*, 2:981–984, 1999.

87. S. N. Diggavi, N. Al-Dhahir, A. Stamoulis, and A. R. Calderbank. Great expectations: The value of spatial diversity in wireless networks. *Proceedings of IEEE*, 92(2):219–270, 2004.

88. Tassos Dimitriou and Ioannis Krontiris. A Localized, Distributed Protocol for Secure Information Exchange in Sensor Networks. In *Proc. 19th IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS 2005)*.

89. NTT DoCoMo. http://www.nttdocomo.com/, 2006.

90. R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. In *ACM SIGCOMM*, 2004.

91. R. Draves, J. Padhye, and B. Zill. Routing in Multi-radio, Multi-hop Wireless Mesh Networks. In *ACM MobiCom*, 2004.

92. Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, and Pramod K. Varshney. A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. In *Proc. INFOCOM 2004*, pages 586–597.

93. Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. In *Proc. 10th ACM Conf. Computer and Communications Security (CCS 2003)*, pages 42–51, October 2003.

94. Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge. *IEEE Transactions on Dependable and Secure Computing*, 3(1):62–77, Jan.-Mar. 2006.

95. Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod K. Varshney, Jonathan Katz, and Aram Khalili. A Pairwise Key Predistribution Scheme for Wireless Sensor Networks. *ACM Transactions on Information and System Security*, 8(2):228–258, May 2005.

96. V. A. Dubendorf. *Wireless data technologies*. John Wiley & Sons, 2003.

97. V. A. Dubendorf. *Wireless Data Technologies*. John Wiley & Sons, 2003.

98. R. E. Van Dyck and D. J. Miller. Transport of wireless video using separate, concatenated and joint source-channel coding. *Proc. IEEE*, October 1999.

99. Joerg Eberspaecher, Hans-Joerg Voegel, and Christian Bettstetter. *GSM Switching, Services, and Protocols*. John Wiley and Sons, 2001.

100. D. A. Eckhardt and P. Steenkiste. Effort-Limited Fair (ELF) Scheduling for Wireless Networks. In *Proc. INFOCOM*, pages 1097–1106, 2000.

101. Elias C. Efstathiou and George C. Polyzos. A Peer-to-Peer Approach to Wireless LAN Roaming. In *Proc. WMASH*, pages 10–18, September 2003.

102. M. Eltoweissy, M. Moharrum, and R. Mukkamala. Dynamic Key Management in Sensor Networks. *IEEE Communications Magazine*, 44(4):122–130, April 2006.

103. P. Erdös and A. Rényi. On the Evolution of Random Graphs. In *Institute of Mathematics Hungarian Academy of Sciences*, 1959.

104. O. Escalante, T. Perez, J. Solano, and I. Stojmenovic. RNG-Based Searching and Broadcasting Algorithms over Internet Graphs and Peer-To-Peer Computing Systems. In *Proc. 3rd ACS/IEEE International Conference on Computer Systems and Applications*, pages 17–24, January 2005.

105. L. Eschenauer and V. Gligor. A Key Management Scheme for Distributed Sensor Networks. In *Proc. 9th ACM Conf. Computer and Communications Security (CCS 2002)*, pages 41–47, November 2002.

106. R. Dube et al. Signal Stability based Adaptive Routing (SSA) for Ad Hoc Mobile Networks. *IEEE Personal Communication Magazine*, 1997.

107. ETSI. Telecommunications and Internet Protocol Harmonization over Networks (TIPHON): Description of Technical Issues. *ETSI Technical Report 101 300*, October 1999.

108. ETSI. Broadband Radio Access Networks (BRAN): HIPERLAN Type 2: Data Link Control (DLC) Layer: Part 1: Basic Data Transport Functions. *ETSI Technical Specifications*, 2001.

109. E.V. Eyuboglu and S.U. Qureshi. Reduced state sequence estimation with set partitioning and decision feedback. *IEEE Transactions on Communications*, 38:13–20, 1998.

110. F. LING F, R. LOVE, and H. XU. Behavior and performance of power controlled is-95 reverse link under soft handoff. *IEEE Transactions on Vehicular Technology*, 49(9):1697 – 1704, 1999.

111. L. M. Feeney. An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks. *Mobile Networks and Applications*, 6(3):239–249, June 2001.

112. L. M. Feeney and M. Nilsson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. In *Proc. 20th International Conference on Computer Communications*, volume 3, pages 1548–1557, April 2001.

113. Michal Feldman and John Chuang. Overcoming Free-Riding Behavior in Peer-to-Peer Systems. *ACM SIGccom Exchanges*, 5(4):41–50, July 2005.

114. Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. Robust Incentive Techniques for Peer-to-Peer Networks. In *Proc. 5th ACM conference on Electronic Commerce*, pages 102–111, May 2004.

115. Michal Feldman, Christos Papadimitriou, John Chuang, and Ion Stoica. Free-Riding and Whitewashing in Peer-to-Peer Systems. In *Proc. 2004 SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems*, page 228–235, August 2004.

116. Mark Felegyhazi, Jean-Pierre Hubaux, and Levente Buttyan. Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 5(5):463–476, May 2006.

117. W. Feng, K. S. Shin, D. D. Kandlur, and D. Saha. The BLUE Active Queue Management Algorithms. *IEEE/ACM Transactions on Networking*, 10(4):513–528, August 2002.

118. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol—HTTP/1.1. *RFC 2068*, January 1997.

119. Daniel Figueiredo, Jonathan Shapiro, and Don Towsley. Incentives to Promote Availability in Peer-to-Peer Anonymity Systems. In *Proc. 13th IEEE Int'l Conference on Network Protocols*, 2005.

120. S. Floyd. Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-Way Traffic. *ACM Computer Communications Review*, 21(5):30–47, October 1991.

121. S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communications Review*, 24(5):10–23, October 1994.

122. S. Floyd and T. Henderson. The NewReno Modification to TCP¡¦s Fast Recovery Algorithm. *RFC 2582*, April 1999.

123. S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

124. S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky. An Extension to the Selective Acknowledgement (SACK) Option for TCP. *RFC 2883*, July 2000.

125. A. Ganz, Z. Ganz, and K. Wongthavarawat. *Multimedia Wireless Networks: Technologies, Standards, and QoS*. Prentice Hall, 2003.

126. A. Ganz, Z. Ganz, and K. Wongthavarawat. *Multimedia Wireless Networks: Technologies, Standards, and QoS*. Prentice Hall PTR, 2003.

127. Vijay K. Garg. *IS-95 CDMA and cdma 2000: Cellular/PCS Systems Implementation*. Prentice Hall, 1999.

128. M. S. Gast. *802.11 Wireless Networks*. O'Reilly, 2002.

129. Matthew Gast and Matthew S. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2002.

130. C. Gehrmann, J. Persson, and B. Smeets. *Bluetooth Security*. Artech House, 2004.

131. P. R. Gejji. Forward-link-power control in cdma cellular systems. *Proceedings of IEEE Vehicular Technology Conference VTC*, 2:981–984, 1992.

132. Wolfgang H. Gerstacker, Frank Obernosterer, Robert Schober, Alexander Lehmann, Alexander Lampe, and Peter Gunreben. Widely linear receivers for space-time block coded transmission over frequency-selective fading channels. *url = "citeseer.ist.psu.edu/459853.html.*

133. S. Ghandeharizadeh and T. Helmi. Consistency and Replication: An Evaluation of Alternative Continuous Media Replication Techniques in Wireless Peer-To-Peer Networks. In *Proc. 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access*, 2003.

134. Gnutella. http://www.gnutella.com/, 2006.

135. Gerardo Gomez and Rafael Sanchez. *End-to-End Quality of Service over Cellular Networks: Data Services Performance Optimization in 2G/3G*. John Wiley and Sons, 2005.

136. O. Granberg. GSM on the Net. *Ericsson Review*, 4:184–191, 1998.

137. J. Gruber and L. Strawczynski. Subjective Effects of Variable Delay and Speech Clipping in Dynamically Managed Voice Systems. *IEEE Transactions on Communications*, 8:801–808, August 1985.

138. Rohit Gupta and Arun K. Somani. A Pricing Strategy For Incentivizing Selfish Nodes To Share Resources In Peer-to-Peer (P2P) Networks. In *Proc. IEEE International Conference on Networks*, November 2004.

139. Ahsan Habib and John Chuang. Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media Streaming. *IEEE Transactions on Multimedia*, 8(3):601–621, June 2006.

140. A. Hac. *Mobile Telecommunications Protocols for Data Networks*. John Wiley & Sons, Ltd., 2003.

141. Timo Halonen, Javier Romero, and Juan Melero. *GSM, GPRS and EDGE Performance: Evolution Towards 3G/UMTS*. John Wiley and Sons, 2003.

142. T. Hamann and J. Walrand. A New Fair Window Algorithm for ECN Capable TCP (New-ECN). In *Proc. INFOCOM*, volume 3, pages 1528–1536, March 2000.

143. Lawrence Harte. *Introduction to GSM: Physical Channels, Logical Channels, Network, and Operation*. Althos, 2004.

144. Lawrence Harte, Morris Hoenig, Daniel McLaughlin, and Roman Kikta. *CDMA IS-95 for Cellular and PCS: Technology, Applications, and Resource Guide*. McGraw Hill, 1999.

145. David Hausheer, Nicolas C. Liebau, Andreas Mauthe, Ralf Steinmetz, and Burkhard Stiller. Token Based Accounting and Distributed Pricing to Introduce Market Mechanisms in a Peer-to-Peer File Sharing Scenario. In *Proc. Third International Conference on Peer-to-Peer Computing*, 2003.

146. David Hausheer and Burkhard Stiller. Decentralized Auction-based Pricing with PeerMart. In *Proc. 9th IFIP/IEEE International Symposium on Integrated Network Management*, May 2005.

147. Simon Haykin and Michael Moher. *An Introduction to Digital and Analog Communications*. John Wiley and Sons, 2001.

148. M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava. PROMISE: Peer-to-Peer Media Streaming Using CollectCast. In *Proc. ACM Multimedia*, November 2003.

149. R. Hinden and S. Deering. Internet Protocol Version 6 (IPv6) Addressing Architecture. *RFC 3513*, April 2003.

150. J. C. Hoe. Improving the Start-Up Behavior of a Congestion Control Scheme for TCP. In *Proc. ACM SIGCOMM*, pages 270–280, August 1996.

151. H. Holma and A. Toskala (eds.). *WCDMA for UMTS: Radio access for third generation mobile communications*. John Wiley & Sons, third edition, 2004.

152. Harri Holma and Antti Toskala. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. John Wiley and Sons, 2001.

153. Hung-Yu Hsieh and Raghupathy Sivakumar. On Using Peer-to-Peer Communication in Cellular Wireless Data Networks. *IEEE Transactions on Mobile Computing*, 3(1):57–72, Jan.-Mar. 2004.

154. J. H. Hu and K. L. Yeung. FDA: A Novel Base Station Flow Control Scheme for TCP over Heterogeneous Networks. In *Proc. INFOCOM*, volume 1, pages 142–151, April 2001.

155. J. H. Hu, K. L. Yeung, S. C. Kheong, and G. Feng. Hierarchical Cache Design for Enhancing TCP over Heterogeneous Networks with Wired and Wireless Links. In *Proc. GLOBECOM*, volume 1, pages 338–343, November 2000.

156. Q. Hu and D. L. Lee. Adaptive Cache Invalidation Methods in Mobile Environments. In *Proc. 6th IEEE International Symposium on High Performance Distributed Computing*, pages 264–273, August 1997.

157. Yih-Chun Hu, Adrain Perrig, and David B. Johnson. ARIADNE: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proc. 8th Int'l Conf. Mobile Computing and Networking (MOBICOM 2002)*, pages 12–23, 2002.

158. Yang hua Chu and Hui Zhang. Considering Altruism in Peer-to-Peer Internet Streaming Broadcast. In *Proc. NOSSDAV'04*, pages 10–15, June 2004.

159. Dijiang Huang, Manish Mehta, Deep Medhi, and Lein Harn. Location Aware Key Management Scheme for Wireless Sensor Networks. In *Proc. ACM SASN 2004*, pages 29–42, October 2004.

160. Qiang Huang, Johnas Cukier, Hisashi Kobayashi, Bede Liu, and Jinyun Zhang. Fast Authenticated Key Establishment Protocols for Self-Organizing Sensor Networks. In *Proc. WSNA 2003*, pages 141–150, September 2003.

161. Joengmin Hwang and Yongdae Kim. Revisiting Random Key Predistribution Schemes for Wireless Sensor Networks. In *Proc. ACM SASN 2004*, pages 43–52, October 2004.

162. Omer Ileri, Siun-Chuon Mau, and Narayan B. Mandayam. Pricing for Enabling Forwarding in Self-Configuring Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 23(1):151–162, January 2005.

163. Takashi Ito, Hidenori Ohta, Nori Matsuda, and Takeshi Yoneda. A Key Predistribution Scheme for Secure Sensor Networks Using Probability Density Function of Node Deployment. In *Proc. ACM SASN 2005*, pages 69–75, November 2005.

164. V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM*, pages 314–329, August 1988.

165. V. Jacobson. Modified TCP Congestion Avoidance Algorithm. *Technical Report 30*, April 1990.

166. A. Jagoe. *Mobile Location Servies: The Definitive Guide*. Prentice Hall PTR, December 2002.

167. William C. Jakes. *Microwave Mobile Communications*. John Wiley and Sons, 1975.

168. A. Jalali, R. Padovane, and R. Pankaj. Data Throughput of CDMA-HDR: A High Efficiency High Data Rate Personal Communication Wireless System. In *Proc. VTC*, 2000.

169. Zou Jia, Si Tiange, Huang Liansheng, and Dai Yiqi. A New Micropayment Protocol Based on P2P Networks. In *Proc. 2005 IEEE Int'l Conference on e-Business Engineering*, 2005.

170. M. Jiang, J. Li, and Y. C. Tay. Cluster Based Routing Protocol,. *IETF Internet Draft*, 1999.

171. J. Jing, A. Elmagarmid, A. S. Helal, and R. Alonso. Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments. *Mobile Networks and Applications*, 2(2):115–127, October 1997.

172. D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. *RFC 3775*, June 2004.

173. D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Network. *Mobile Computing*, 1996.

174. Gaurav Jolly, Mustafa C. Kuscu, Pallavi Kokate, and Mohamed Younis. A Low Energy Key Management Protocol for Wireless Sensor Networks. In *Proc. 8th IEEE Int'l Symposium on Computers and Communications*, 2003.

175. Seung Jun and Mustaque Ahamad. Incentives in BitTorrent Induce Free Riding. In *Proc. ACM SIGCOMM 2005 Workshop*, August 2005.

176. Seung Jun, Mustaque Ahamad, and Jun (Jim) Xu. Robust Information Dissemination in Uncooperative Environments. In *Proc. 25th IEEE Int'l Conference on Distributed Computing Systems*, 2005.

177. E. Jung and N. H. Vaidya. A Power Control MAC Protocol for Ad Hoc Networks. In *ACM MOBICOM*, 2002.

178. E. Jung and N. H. Vaidya. An Energy Efficient MAC Protocol for Wireless LANs. In *IEEE INFOCOM*, 2002.

179. J. M. Kahn, R. H. Katz, and K. S. Pister. Mobile Networking for Smart Dust. In *Proc. Int'l Conf. Mobile Computing and Networking (MOBICOM 1999)*, August 1999.

180. A. Kahol, S. Khurana, S. K. S. Gupta, and P. K. Srimani. A Strategy to Manage Cache Consistency in a Disconnected Distributed Environment. *IEEE Transactions on Parallel and Distributed Systems*, 12(7):686–700, July 2001.

181. Seung-Seok Kang and Matt W. Mutka. A Mobile Peer-to-Peer Approach for Multimedia Content Sharing Using 3G/WLAN Dual Mode Channels. *Wireless Communications and Mobile Computing*, 5:633–645, 2005.

182. C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. In *Proc. 1st IEEE Int'l Workshop on Sensor Network Protocols and Applications*, pages 113–127, May 2003.

183. P. Karn and C. Partridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. In *Proc. ACM SIGCOMM*, pages 2–7, August 1987.

184. B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. 6th Int'l Conf. Mobile Computing and Networking (MOBICOM 2000)*, pages 243–254, August 2000.

185. H. Kawai, H. Suda, and F. Adachi. Outer-loop control of target sir for fast transmit power control inturbo-coded w-cdma mobile radio. *Electronics Letters*, 35:699–701, 1999.

186. R. Keenan. Caching technique eases wlan roaming, available at: http://www.commsdesign.com/news/tech_beat/showarticle.jhtml?articleid=26806681/, 2006.

187. F. Kelly. Charging and Rate Control for Elastic Traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

188. S. Keshav and S. P. Morgan. SMART Retransmission: Performance with Overload and Random Losses. In *Proc. INFOCOM*, volume 3, pages 1131–1138, April 1997.

189. H Kim, S Bang, and Y Han. Performance of fast forward power control using coherent sir estimation. *Proceedings of 4th CIC Conference*, 2:981–984, 1999.

190. Kyoung Il Kim. *Handbook of CDMA System Design, Engineering and Optimization*. Prentice Hall, 1999.

191. J. J. Kistler and M. Satyanarayanan. Disconnected operation in the coda file system. *ACM Transactions on Computer Systems*, Feburary 1992.

192. J. Korhonen. *Introduction to 3G mobile communications*. Artech House, second edition, 2003.

193. Ramayya Krishnana, Michael D. Smith, and Rahul Telang. The Economics of Peer-to-Peer Networks. *Journal of Information Technology Theory and Application*, 5(3):31–44, 2003.

194. J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, third edition, 2005.

195. Y. K. Kwok and V. K. N. Lau. A Novel Channel-Adaptive Uplink Access Control Protocol for Nomadic Computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(11):1150–1165, November 2002.

196. M.A. Lagunas, A. I. Neira, M. G. Amin, and J. Vidal. Spatial processing for frequency diversity schemes. *IEEE Transactions on Signal Processing*, 48(2):353–362, 2000.

197. T. V. Lakshman, A. Ortega, and A. R. Reibman. VBR Video: Tradeoffs and Potentials. *Proc. IEEE*, 86(5):952–973, May 1998.

198. V. K. N. Lau. Performance of Variable Rate Bit-Interleaved Coding for High Bandwidth Efficiency. In *Proc. VTC*, volume 3, pages 2054–2058s, May 2000.

199. Chun Fai Law, Ka Shun Hung, and Yu-Kwong Kwok. A Key Redistribution Scheme for Distributed Sensor Networks. In *Technical Report HKUEEE-2006-12, The University of Hong Kong*, October 2006.

200. Yee Wei Law, Ricardo Corin, Sandro Etalle, and Pieter H. Hartel. A Formally Verified Decentralized Key Management Architecture for Wireless Sensor Networks. In *Technical Report TR-CTIT-03-07 Centre for Telematics and Information Technology, University of Twente, Enschede*, 2003.

201. Jooyoung Lee and Dou glas R. Stinson. Deterministic Key Predistribution Schemes for Distributed Sensor Networks. In *Proc. SAC 2004, Lecture Notes in Computer Science Volume 3357*, pages 294–307.

202. William C. Y. Lee. *Mobile Cellular Telecommunications: Analog and Digital Systems*. McGraw Hill, 1995.

203. William C. Y. Lee. *Lee's Essentials of Wirelesss Communications*. McGraw Hills, 2000.

204. Andrew Ka Ho Leung and Yu-Kwong Kwok. An Efficient and Practical Greedy Algorithm for Server-Peer Selection in Wireless Peer-to-Peer File Sharing Networks. In *Proceedings of the International Conference on Mobile Ad-hoc and Sensor Networks (MSN'2005), (Lecture Notes in Computer Science, Volume 3794, Xiaohua Jia, Jie Wu, Yanxiang He (eds.), Springer Berlin/Heidelberg)*, pages 1016–1025, December 2005.

205. Andrew Ka Ho Leung and Yu-Kwong Kwok. Community-Based Asynchronous Wakeup Protocol for Wireless Peer-to-Peer File Sharing Networks. In *Proceedings of the IEEE Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'2005)*, pages 342–350, July 2005.

206. Andrew Ka Ho Leung and Yu-Kwong Kwok. Energy Conservation by Peer-to-Peer Relaying in Quasi-Ad Hoc Networks. In *Proceedings of IFIP International Conference on Network and Parallel Computing (NPC'2005), (Lecture Notes in Computer Science, Volume 3779, Hai Jin, Daniel Reed, Wenbin Jiang (eds.), Springer Berlin/Heidelberg)*, pages 451–460, November 2005.

207. Andrew Ka Ho Leung and Yu-Kwong Kwok. On Topology Control of Wireless Peer-to-Peer File Sharing Networks: Energy Efficiency, Fairness and Incentive. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'2005)*, pages 318–323, June 2005.

208. K. H. Leung. *Localized Topology Control in Wireless Peer-to-Peer File Sharing Networks*. MPhil Thesis, The University of Hong Kong, 2005.

209. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehous, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An Operating System for Sensor Networks. In *Ambient Intelligence (W. Weber, JM Rabaey, and E. Aarts)*. Springer Verlag, 2005.

210. Chenyang Li and Sumit Roy. Performance of Frequency-Time MMSE Equalizer for MC-CDMA over Multipath Fading Channel. *Wireless Personal Communications*, 18:179–192, 2001.

211. J. Li, D. Cordes, and J. Zhang. Power-aware routing protocols in ad hoc wireless networks. *IEEE Wireless Communications*, December 2005.

212. L. Li and J. Y. Halpern. A Minimum-Energy Path-Preserving Topology Control Algorithm. *IEEE Transactions on Wireless Communications*, 3(3):910–921, May 2004.

213. L. Li, H. B. Li, and Y. D. Yao. Time diversity and equalization for frequency selective fading channels. *Proceedings of IEEE Vehicular Technology Conference (VTC)*, 92(2):1673–1678, 2001.

214. X. Y. Li, Y. Wang, and W.-Z. Song. Applications of k-Local MST for Topology Control and Broadcasting in Wireless Ad Hoc Networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(12):1057–?1069, December 2004.

215. P. Lin, B. Bensaou, Q. L. Ding, and K. C. Chua. A Wireless Fair Scheduling Algorithm for Error-Prone Wireless Channels. In *Proc. WoWMoM*, pages 11–20, 2000.

216. X. Lin, Y. Kwok, and V. K. N. Lau. Power Control for IEEE 802.11 Ad Hoc Networks: Issues and A New Algorithm. In *IEEE ICPP*, 2003.

217. Yi Bing Lin. *Wireless and Mobile Network Architectures*. John Wiley and Sons, 2001.

218. C. Liu and R. Jain. Delivering Faster Congestion Feedback with the Mark-Front Strategy. In *Proc. WCC-ICCT*, volume 1, pages 665–672, August 2000.

219. Chia-Liang Liu and Kamilo Feher. Pilot-symbol aided coherent m-ary psk in frequency-selective fast rayleigh fading channels. *IEEE Transactions on Communications*, 42(1):54–62, 1994.

220. Donggang Liu and Peng Ning. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proc. 10th ACM Conf. Computer and Communications Security (CCS 2003)*, pages 52–61, October 2003.

221. Donggang Liu and Peng Ning. Location Based Pairwise Key Establishments for Static Sensor Networks. In *Proc. 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 72–82, 2003.

222. Donggang Liu and Peng Ning. Improving Key Predistribution with Deployment Knowledge in Static Sensor Networks. *ACM Transactions on Sensor Networks*, 1(2):204–239, November 2005.

223. Donggang Liu, Peng Ning, and Wenliang Du. Group Based Key Predistribution in Wireless Sensor Networks. In *Proc. ACM WiSE 2005*, pages 11–20, September 2005.

224. Donggang Liu, Peng Ning, and Rongfang Li. Establishing Pairwise Keys in Distributed Sensor Networks. *ACM Transactions on Information and System Security*, 8(1):41–77, February 2005.

225. Fang Liu, Major Jose "Many" Rivera, and Xiuzhen Cheng. Location Aware Key Establishment in Wireless Sensor Networks. In *Proc. ACM IWCMC 2006*, pages 21–26, July 2006.

226. Su-Lin Low and Ron Schneider. *CDMA Internetworking: Deploying the Open A-Interface*. Prentice Hall, 2000.

227. S. Lu, V. Bharghavan, and R. Srikant. Fair Scheduling in Wireless Packet Networks. *ACM Transactions on Networking*, 7(4):473–489, August 1999.

228. Richard T. B. Ma, Sam C. M. Lee, John C. S. Lui, and David K. Y. Yau. A Game Theoretic Approach to Provide Incentive and Service Differentiation in P2P Networks. In *Proc. SIGMETRICS*, pages 189–198, June 2004.

229. Richard T. B. Ma, Sam C. M. Lee, John C. S. Lui, and David K. Y. Yau. An Incentive Mechanism for P2P Networks. In *Proc. 24th Int'l Conference on Distributed Computing Systems*, 2004.

230. W. Ma and Y. Fang. Dynamic Hierarchical Mobility Management Strategy for Mobile IP Networks. *IEEE Journal on Selected Areas in Communications*, 22(4):664–676, May 2004.

231. David J. Malan, Matt Welsh, and Michael D. Smith. A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. In *Proc. First IEEE International Conference on Sensor and Ad Hoc Communications and Networks*, October 2004.

232. Giridhar D. Mandyam and Jersey Lai. *Third Generation CDMA Systems for Enhanced Data Services*. Academic Press, 2006.

233. Z. Mao and C. Douligeris. A distributed database architecture for global roaming in next-generation mobile networks. *IEEE/ACM Transactions on Networking*, Feburary 2004.

234. Peter Marbach and Ying Qiu. Cooperation in Wireless Ad Hoc Networks: A Market Based Approach. *IEEE/ACM Transactions on Networking*, 13(6):1325–1338, December 2005.

235. S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *ACM MobiCom*, 2000.

236. M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. *RFC 2018*, October 1996.

237. M. Maxim and D. Pollino. *Wireless Security*. McGraw-Hill, 2002.

238. Maze. http://maze.pku.edu.cn, 2006.

239. Asha Mehrotra. *GSM System Engineering*. Artech House Publishers, 1997.

240. R. C. Merkle. A Certified Digital Signature. In *Advances in Cryptology (CRYPTO 1989)*, August 1989.

241. Mathhew J. Miller and Nitin H. Vaidya. Leveraging Channel Diversity for Key Establishment in Wireless Sensor Networks. In *Proc. IEEE INFOCOM 2006*.

242. S. S. Miller. *WI-FI Security*. McGraw-Hill, 2003.

243. Ajay R. Mishra. *Advanced Cellular Network Planning and Optimisation: 2G/2.5G/3G...Evolution to 4G*. John Wiley and Sons, 2007.

244. WAP User Agent Caching Model. http://www.openmobilealliance.org/release_program/browsing_archive.html, 2006.

245. J. W. Modestino and D. G. Daut. Combined source-channel coding of images. *IEEE Transactions on Communications*, November 1979.

246. J. W. Modestino, D. G. Daut, and A. L. Vickers. Combined source-channel coding of images using the block consine transform. *IEEE Transactions on Communications*, September 1981.

247. P. Mohapatra and S. V. Krishnamurthy. *Ad Hoc Networks Technologies and Protocols*. Springer, 2005.

248. J. P. Monks, V. Bharghavan, W. Mei, and W. Hwu. A Power Controlled Multiple Access Protocol for Wireless Packet Networks. In *IEEE INFOCOM*, 2001.

249. L. B. Mummert, M. R. Ebling, and M. Satyanarayanan. Exploiting Weak Connectivity for Mobile File Access. In *Proc. 15th ACM Symposium on Operating Systems Principles*, December 1995.

250. Y. Mun and H. K. Lee. *Understanding IPv6*. Springer, first edition, 2005.

251. A. Muqattash and M. M. Krun. A Distributed Transmission Power Control Protocol for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 3(2):113–?128, April 2004.

252. S. Murthy and J. J. Garcia-Luna-Aceves. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proc. ACM First International Conference on Mobile Computing & Networking (MobiCom)*, 1995.

253. S. Nanda, R. Ejzak, and B. T. Doshi. A Retransmission Scheme for Circuit-Mode Data on Wireless Links. *IEEE Journal on Selected Areas in Communications*, 12(8):1338–1352, October 1994.

254. T. Nandagopal, S. Lu, and V. Bharghavan. A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms. In *Proc. MOBICOM*, pages 132–142, 1999.

255. Napster. http://www.napster.com/, 2006.

256. Mesh Networking. http://research.microsoft.com/mesh/, 2006.

257. T. S. E. Ng, I. Stoica, and H. Zhang. Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors. In *Proc. INFOCOM*, pages 1103–1111, 1998.

258. NOT-YET-ENTERED2. NOT-YET-ENTERED2. *NOT-YET-ENTERED2*, 2006.

259. NOT-YET-ENTERED3. NOT-YET-ENTERED3. *NOT-YET-ENTERED3*, 2006.

260. T. Ojanperä and R. Prasad. An overview of air interface multiple access for IMT-2000/UMTS. *IEEE Communications Magazine*, 36(9):82–86,91–95, September 1998.

261. Y. Okumura. Field strength and its variability in vhf and uhf land-mobile radio service. *Review of the Electrical Communication Laboratories*, 16(9):825–873, 1968.

262. Open Mobile Alliance (OMA). http://www.openmobilealliance.org/, 2006.

263. OnStar. http://www.onstar.com/us_english/jsp/index.jsp, 2006.

264. L. Ophir and Y. Bitran. 802.11 Over Coax—A Hybrid Coax-Wireless Home Network Using 802.11 Technology . In *Proc. Consumer Communications and Networking Conference*, January 2004.

265. Martin J. Osborne. *An Introduction to Game Theory*. Oxford University Press, 2004.

266. Connecting IPv6 Islands over IPv4 MPLS using IPv6 Provider Edge Routers (6PE). http://www.ietf.org/internet-drafts/draft-ooms-v6ops-bgp-tunnel-06.txt/, 2006.

267. Teredo Overview. http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.mspx, 2006.

268. Taejoon Park and Kang G. Shin. LiSP: A Lightweight Security Protocol for Wireless Sensor Networks. *ACM Transactions on Embedded Computing Systems*, 3(3):634–660, August 2004.

269. V. Park and M. Scott Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proc. INFOCOM*, 1996.

270. J. D. Parsons. *The Mobile Radio Propagation Channel*. John Wiley & Sons, second edition, 2000.

271. M. Patzold. *Mobile Fading Channels: Modelling, Analysis, and Simulation*. John Wiley and Sons, 2002.

272. M. R. Pearlman and Z. Haas. Determining the Optimal Configuration for the Zone Routing Protocol. *IEEE Journal on Selected Areas in Communications*, August 1999.

273. G. Pei, M. Gerla, and T.-W. Chen. Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks. In *Proc. IEEE International Conference on Communications*, 2000.

274. Jonathan Leary Pejman Roshan. *Wireless Local-Area Network Fundamentals*. Cisco Press, 2003.

275. F. Peng, S. Cheng, and J. Ma. An Effective Way to Improve TCP Performance in Wireless/Mobile Networks. In *Proc. EUROCOMM*, volume 3, pages 250–255, May 2000.

276. C. E. Perkins. Mobile IP. *IEEE Communications Magazine*, pages 84–99, May 1997.

277. C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance Vetcor Routing. In *Proc. ACM SIGCOMM*, 1994.

278. C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing. In *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

279. A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In *Proc. 7th Annual Int'l Conf. Mobile Computing and Networks (MOBICOM 2001)*, pages 189–199, July 2001.

280. Roberto Di Pietro, Luigi V. Mancini, and Alessandro Mei. Efficient and Resilient Key Discovery Based on Pseudo-Random Key Pre-Deployment. In *Proc. 18th Int'l Parallel and Distributed Processing Symposium (IPDPS 2004)*.

281. Roberto Di Pietro, Luigi V. Mancini, and Alessandro Mei. Random Key Assignment for Secure Wireless Sensor Networks. In *Proc. 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 62–71, 2003.

282. PlanetLab. http://www.planet-lab.org, 2006.

283. David Poole. *Linear Algebra: A Modern Introduction*. Brooks Cole, 2002.

284. R. Prasad and L. Munoz. *WLANs and WPANs towards 4G Wireless*. Artech House, 2003.

285. R. Prasad and M. Ruggieri. *Technology Trends in Wireless Communications*. Artech House, 2003.

286. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *Proc. 6th Annual International Conference on Mobile Computing and Networking*, 2000.

287. John G. Proakis. *Digital Communications*. McGraw Hill, 2000.

288. M. B. Pursley. Performance evaluation for phase-coded spread-spectrum multiple-access communication–part i: System analysis. *IEEE Transactions on Communications*, 25(8):795 – 799, 1977.

289. Mu Qin. *Space-time-frequency: Diversity for wireless communication*. ProQuest, 2006.

290. Dongyu Qiu and R. Srikant. Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In *Proc. SIGCOMM*, pages 367–377, September 2004.

291. Qualcomm. Cdma 1xrtt security overview. *Qualcomm White Paper*, August 2002.

292. K. Ramakrishnan and S. Floyd. A Proposal to Add Explicit Congestion Notification (ECN) to IP. *RFC 2481*, January 1999.

293. P. Ramanathan and P. Agrawal. Adapting Packet Fair Queueing Algorithms to Wireless Networks. In *Proc. MOBICOM*, pages 1–9, 1998.

294. Kavitha Ranganathan, Matei Ripeanu, Ankur Sarin, and Ian Foster. To Share or not to Share: An Analysis of Incentives to Contribute in Collaborative File-Sharing Environments. In *Proc. Workshop on Economics of Peer-to-Peer systems*, June 2003.

295. H. C. H. Rao, Y.-B. Lin, and S.-L. Cho. iGSM: VoIP Service for Mobile Networks. *IEEE Communications Magazine*, 38(4):62–69, April 2000.

296. Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2001.

297. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A Geographic Hash Table for Data-Centric Storage. In *Proc. 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, September 2002.

298. Siegmund Redl, Matthias K. Weber, and Malcolm Oliphant. *An Introduction to GSM*. Artech House Publishers, 1995.

299. C.W. Rhodes and P. Crosby. Measuring peak and average power of digitally modulated advanced television systems. *IEEE Transactions on Broadcasting Technology*, 37(8):885–890, Dec 1992.

300. V. Rodoplu and T. H. Meng. Minimum Energy Mobile Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333?–1344, August 1999.

301. A. Rowstron and P. Druschel. Pastry: Scalable, distributed object address and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM Int'l Conf. on Distributed Systems Platforms*, November 2001.

302. E. M. Royer and C. K. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Networks. *IEEE Personal Communications*, April 1999.

303. Kenji Saito. Peer-to-Peer Money: Free Currency over the Internet. In *Proc. 2nd Int'l Conference on Human.Society@Internet*, June 2003.

304. Kenji Saito, Eiichi Morino, and Jun Murai. Multiplication Over Time to Facilitate Peer-to-Peer Barter Relationship. In *Proc. 16th Int'l Workshop on Database and Expert Systems Applications*, 2005.

305. Naouel Ben Salem, Levente Buttyan, Jean-Pierre Hubaux, and Markus Jakobsson. Node Cooperation in Hybrid Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 5(4):365–376, April 2006.

306. J. Eric Salt and Surinder Kumar. Effects of filtering on the performance of qpsk and msk modulation in d-s spread spectrum systems using rake receivers. *IEEE Journal on Selected Areas in Communications*, 12(4):707–715, 1994.

307. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, first edition, 1983.

308. Sujay Sanghavi and Bruce Hajek. A New Mechanism for the Free-Rider Problem. In *Proc. SIGCOMM 2005 Workshop*, August 2005.

309. K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A Secure Routing protocol for Ad Hoc Networks. In *Proc. 10th IEEE International Conference on Network Protocol*, 2002.

310. M. Satyanarayanan. The evolution of coda. *ACM Transactions on Computer Systems*, May 2002.

311. M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere. Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on Computers*, April 1990.

312. T. C. Schelling. *Micromotives and Macrobehavior*. W. W. Norton & Company, 1978.

313. J. Schiller. *Mobile Communications*. Addison Wesley, 2003.

314. M. Schlager, B. Rathke, S. Bodenstein, and A.Wolisz. Advocating a Remote Socket Architecture for Internet Access using Wireless LANs. *Mobile Networks and Applications*, 6(1):23–42, January 2001.

315. Shunsuke SEO, Tomohiro DOHI, and Fumiyuki ADACHI. Sir-based transmit power control of reverse link for coherent ds-cdma mobile radio. *IEICE TRANSACTIONS on Communications*, E81-B(2):1508–1516, 1998.

316. John S. Seybold. *Introduction to RF propagation*. John Wiley and Sons, 2005.

317. Y. Shaked and A. Wool. Cracking the Bluetooth PIN. In *Proc. 3rd International Conference on Mobile Systems, Applications, and Services*, 2005.

318. Elaine Shi and Adrian Perrig. Designing Secure Sensor Networks. *IEEE Wireless Communications*, pages 38–43, December 2004.

319. J. Shin, D. C. Lee, and C.-C. J. Kuo. *Quality of Service for Internet Multimedia*. Prentice Hall PTR, 2003.

320. Marvin K. Simon and Mohamed-Slim Alouini. *Digital Communication over Fading Channels*. John Wiley and Sons, 2004.

321. Sumeet Singh, Sriram Ramabhadran, Florin Baboescu, and Alex C. Snoeren. The Case for Service Provider Deployment of Super-Peers in Peer-to-Peer Networks. In *Proc. Workshop on Economics of Peer-to-Peer Systems*, June 2003.

322. David Soldani, Man Li, and Renaud Cuny. *QoS and QoE Management in UMTS Cellular Systems*. John Wiley and Sons, 2006.

323. W. Stallings. *Wireless Communications And Networks*. Prentice Hall, 2002.

324. Raymond Steele and Lajos Hanzo. *Mobile Radio Communications*. John Wiley and Sons, 1999.

325. Raymond Steele, Chin Chun Lee, and Peter Gould. *GSM, CDMAOne and 3G Systems*. John Wiley and Sons, 2001.

326. J. Steiner, C. Neuman, and J. Schiller. Kerberos: An Authentication Service for Open Network Systems. In *Proc. USENIX Winter Conf.*, pages 191–202, January 1988.

327. W. R. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. *RFC 2001*, January 1997.

328. I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A Scalable Peer-To-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, February 2003.

329. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. ACM SIGCOMM 2001*, pages 149–160, August 2001.

330. Qixiang Sun and Hector Garcia-Molina. SLIC: A Selfish Link-based Incentive Mechanism for Unstructured Peer-to-Peer Networks. In *Proc. 24th Int'l Conference on Distributed Computing Systems*, 2004.

331. K. L. Tan. Organization of Invalidation Reports for Energy-Efficient Cache Invalidation in Mobile Environments. *Mobile Networks and Applications*, 6(3):279–290, June 2001.

332. K. Tang, M. correa, and M. Gerla. Isolation of Wireless Ad Hoc Medium Access Mechanisms under TCP. In *Proc. Computer Communications and Networks*, pages 77–82, October 1999.

333. Using UAProf (User Agent Profile) to Detect User Agent Types and Device Capabilities. http://www.developershome.com/wap/detection/detection.asp?page=uaprof, 2006.

334. C. K. Toh. Associativity-Based Routing For Mobile Ad-Hoc Networks. *Journal on Wireless Personal Communications*, 1997.

335. Patrick Traynor, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas La Porta. Establishing Pairwise Keys in Heterogeneous Sensor Networks. In *Proc. IEEE INFOCOM 2006*.

336. Hal R. Varian. The Social Cost of Sharing. In *Proc. Workshop on Economics of Peer-to-Peer systems*, June 2003.

337. E. P. Vasilakopoulou, G. E. Karastergios, and G. D. Papadopoulos. Design and implementation of the hiperlan/2 protocol. *Mobile Computing and Communications Review*, 2003.

338. Srdjan Čapkun, Jean-Pierre Hubaux, and Levente Buttyán. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 5(1):43–51, January 2006.

339. A. J. Viterbi. *CDMA: Principles of Spread Spectrum Communication*. Prentice Hall, 1995.

340. A. J. VITERBI, A. M. VITERBI, and K. GILHOUSEN. Soft handoffs extends cdma coverage and increase reverse link capacity. *IEEE Journal of Selected Areas on Communications*, 12(8):1281 – 1288, 1994.

341. vivek Shrivastava and Suman Banerjee. Natural Selection in Peer-to-Peer Streaming: From the Cathedral to the Bazaar. In *Proc. NOSSDAV'05*, June 2005.

342. Ashraf Wadaa, Stephan Olariu, Larry Wilson, and Mohamed Eltoweissy. Scalable Cryptographic Key Management in Wireless Sensor Networks. In *Proc. 24th IEEE Int'l Conf. Distributed Computing Systems Workshops*, 2004.

343. B. Walke, P. Seidenberg, and M. P. Althoff. *UMTS: The Fundamentals*. John Wiley & Sons, 2003.

344. L. Wang, Y. K. Kwok, W. C. Lau, and V. K. N. Lau. Efficient Packet Scheduling Using Channel Adaptive Fair Queueing in Distributed Mobile Computing Systems. *ACM/Kluwer Mobile Networks and Applications*, 9(4):297–309, August 2004.

345. W. Wang, S. C. Liew, and V. O. K. Li. Solutions to Performance Problems in VoIP over a 802.11 Wireless LAN. *IEEE Transactions on Vechicular Technology*, 54(1):366–384, January 2005.

346. Weihong Wang and Baochun Li. Market-driven Bandwidth Allocation in Selfish Overlay Networks. In *Proc. IEEE INFOCOM 2005*, March 2005.

347. Y. Wang, S. Ye, and X. Li. Understanding Current IPv6 Performance: A Measurement Study. In *Proc. 10th IEEE Symposium on Computers and Communications*, pages 71–76, June 2005.

348. Y. Wang and Q.-F. Zhu. Error control and concealment for video communication: A review. *Proc. IEEE*, May 1998.

349. S. Williams. Irda: Past, present and future. *Personal Communications*, Feburary 2000.

350. Ouri Wolfson, Bo Xu, and A. Prasad Sistla. An Economic Model for Resource Exchange in Mobile Peer to Peer Networks. In *Proc. 16th Int'l Conference on Scientific and Statistical Databased Management*, 2004.

351. W. C. Wong, R. Steele, B. Glance, and D. Horn. Time diversity with adaptive error detection to combat rayleigh fading in digital mobile radio. *IEEE Transactions on Communications*, 31(3):378–387, 1983.

352. Krit Wongrujira and Aruna Seneviratne. Monetary Incentive with Reputation for Virtual Market-Place Based P2P. In *Proc. CoNEXT'05*, October 2005.

353. Dongyan Xu, Mohamed Hefeeda, Susanne Hambrusch, and Bharat Bhargava. On Peer-to-Peer Media Streaming. In *Proc. 22nd Int'l Conference on Distributed Computing Systems*, 2002.

354. S. Xu and T. Saadawi. Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks? *IEEE Communications Magazine*, 39(6):130–137, June 2001.

355. Guang-Tao Xue, Ming-Lu Li, Qian-Ni Deng, and Jin-Yuan You. Stable Group Model in Mobile Peer-to-Peer Media Streaming System. In *Proc. IEEE Int'l Conference on Mobile Ad Hoc and Sensor Systems*, pages 334–339, 2004.

356. M. D. Yacoub. *Foundations of Mobile Radio Engineering*. CRC Press, 1993.

357. B. Yang and H. Garcia-Molina. PPay: Micropayments for Peer-to-Peer Systems. In *Proc. 10th ACM Conference on Computer and Communication Security*, pages 300–310, 2003.

358. Mao Yang, Zhengyou Zhang, Xiaoming Li, and Yafei Dai. An Empirical Study of Free-Riding Behavior in the Maze P2P File-Sharing System. In *Proc. IPTPS'05*, February 2005.

359. Y. J. Yang and J. F. Chang. A strength and sir combined adaptive power control for cdma mobile radio channels. *IEEE Transactions on Vehicular Technology*, 48(6):1996–2004, 1999.

360. R. Yavatkar and N. Bhagawat. Improving End-to-end Performance of TCP over Mobile Internetworks. In *Proc. Workshop on Mobile Computing Systems and Applications*, pages 146–152, December 1994.

361. Song Ye and Fillia Makedon. Collaboration-Aware Peer-to-Peer Media Streaming. In *Proc. MM'04*, pages 412–415, October 2004.

362. M. K. H. Yeung and Y. K. Kwok. Wireless Cache Invalidation Schemes with Link Adaptation and Downlink Traffic. *IEEE Transactions on Mobile Computing*, 4(1):68–83, January 2005.

363. Mark Kai Ho Yeung and Yu-Kwong Kwok. A Game Theoretic Approach to Power Aware Wireless Data Access. *IEEE Transactions on Mobile Computing*, 5(8), August 2006.

364. Mark Kai Ho Yeung and Yu-Kwong Kwok. On Maximizing Revenue for Client-Server Based Wireless Data Access in the Presence of Peer-to-Peer Sharing. In *Proc. 17th Annual IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'2006)*, September 2006.

365. H. K. Yip. *Packet Scheduling Techniques for Coordinating Colocated Bluetooth and IEEE 802.11b in a Linux Machine*. MPhil Thesis, The University of Hong Kong, 2004.

366. Bin Yu and Munindar P. Singh. Incentive Mechanisms for Peer-to-Peer Systems. In *Proc. 2nd International Workshop on Agents and Peer-to-Peer Computing*, July 2003.

367. Zhen Yu and Yong Guan. A Robust Group-Based Key Management Scheme for Wireless Sensor Networks. In *Proc. IEEE WCNC 2005*, pages 1915–1920.

368. J. Zander. Performance of optimum transmitter power control in cellular radio system. *IEEE Journal of Selected Area on Communications*, E81-B(2):51–62, 1992.

369. S. Zeadally, R. Wasseem, and I. Raicu. Comparison of End-System IPv6 Protocol Stacks. *IEE Proceedings—Communications*, 151(3):238–242, June 2004.

370. L. Zhao, J. Shin, and C.-C. J. Kuo J. Kim. http://citeseer.ist.psu.edu/zhao01fgs.html, 2001.

371. R. Zheng, C. Hou, and S. Lui. Asynchronous Wakeup for Ad Hoc Networks. In *Proc. ACM MobiHoc*, pages 35–45, June 2003.

372. Yun Zhou, Yanchao Zhang, and Yuguang Fang. LLK: A Link-Layer Key Establishment Scheme for Wireless Sensor Networks. In *Proc. IEEE WCNC 2005*, pages 1921–1926.

373. Q.-F. Zhu, Y. Wang, and L. Shaw. Coing and cell-loss recovery in dct-based packet video. *IEEE Transactions on Circuits and Systems for Video Technology*, June 1993.

374. Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. LEAP: Efficient Security Mechanisms for Large Scale Distributed Sensor Networks. In *Proc. ACM CCS 2003*, pages 62–72, October 2003.

375. M. Zorzi and R. R. Rao. The Role of Error Correlations in the Design of Protocols for Packet Switched Services. In *Proc. 35th Annual Allerton Conference on Communications, Control, and Computing*, pages 749–758, September 1997.

This Page Intentionally Left Blank

TOPIC INDEX

3G, 14, 101
  See also UMTS
  3G1X, 88, 114, 119, 121
  CDMA2000, 114, 164, 174
  QoS differentiation, 128, 132
ad hoc networks, 287
ad hoc routing
  ABR, 493
  AODV, 487, 489
  CBRP, 493
  CSGR, 485
  DSDV, 483
  DSR, 487
  Fisheye state routing, 485
  flooding, 485
  GSR, 485
  OLSR, 485
  power aware, 497
  proactive, 476, 483, 485, 487, 494–495, 500
  reactive, 476, 487, 489, 491, 493, 495
  RICA, 491
  RREP, 489, 492
  RREQ, 487, 489, 491–492
  security, 496
  SSA, 493
  system model, 480
  TBRPF, 485
  TORA, 493
  WRP, 485
AES, 520, 536
AMPS, 39, 44, 113–114, 116–118, 131
  IS54, 131
ARP, 389
ARQ, 420
AVSG, 583–584, 588
Barker sequence, 272, 275
binary exponential backoff, 270
Bluetooth, 285–287, 289–290, 292, 295, 304, 330–331, 335, 339, 341, 354–355, 359, 410–412, 520, 529, 531–532, 540–541
  ACL, 286, 294–295, 299, 304
  baseband, 289, 292
  cable replacement, 285, 287, 305, 307
  channels, 297
  encryption, 531
  eSCO, 294, 296

hold mode, 299
IAC, 297
L2CAP, 289
LMP, 289
MAC, 286, 292, 302, 307
master-slave, 293–294, 296–302
OBEX, 286, 290–291
packet types, 294
DH1, 304
DH3, 304
DH5, 304
DM1, 296
pairing, 529
park mode, 299
piconet, 289, 293–294, 296–299, 301–304, 307, 312, 315–316, 318, 321, 323
bridge node, 296, 299–300, 302–303
power classes, 292
profiles, 291
protocol stack, 287–288, 292
RFCOMM, 289, 291
scatternet, 296, 299–300, 302, 307
BlueConstellation, 301
Bluenet, 300
BlueRing, 302
BTCP, 300
formation, 300–303, 307
randomized formation, 301
routing, 302
scheduling, 303
search tree, 301
SCO, 286, 294, 296, 299
SDP, 289
SIG, 285–287
sniff mode, 299
specifications, 287
TDD, 302
time slots, 293–294
CDMA, 13, 43–44, 48, 54–55, 57, 59, 61, 63, 66, 81, 83–84, 86, 112, 114, 116–118, 131, 134, 136, 143–144, 154, 164–166, 174, 520, 524, 526, 540, 546
D-CDMA, 61–65, 67, 69, 73–74, 77–78, 81, 84, 86, 125, 133, 136–138, 140, 173
despreading, 59, 62, 66, 68, 84, 101, 103–106, 136, 143
HDR, 451
IS95, 83–85, 88, 101, 114, 116–117, 121, 131, 133–134, 136–140, 144, 146, 148, 154–156, 158, 160–161, 164–165, 167, 173–174, 176
long code, 136–137, 139, 144, 146, 149, 173
power control, 134, 144, 154–156, 158, 160–161, 172–174
RSSI, 156, 161, 173
processing gain, 68–70, 84
R-CDMA, 66–70, 72–73, 76–78, 81, 84, 86, 89, 133, 136–138, 140, 154, 173
RAKE receiver, 63, 88, 101, 103–106, 166, 170
soft handover, 134, 139, 144, 146, 161, 164–168, 170, 172, 174
active set, 167–168, 170, 172, 174
candidate set, 167–168, 170, 174
neighbor set, 167–168, 170, 172, 174
PSMM, 168, 170, 174
remaining set, 167–168, 170, 174
spreading, 58–59, 62–63, 65–66, 68–69, 86
spreading factor, 59, 61–62, 64, 69, 74, 84–85, 125, 127, 132
Walsh code, 137–139, 143–144, 148, 156, 173
wideband, 114, 125, 220, 225
cellular location techniques
AOA, 605
COO, 605
E-OTD, 606
TODA, 605
cellular networks, 111–114, 116–119, 121, 127, 130–131
cell splitting, 115, 131
MSC, 111, 130–131
channel, 1–3, 6, 9, 12–14, 16, 19
assignment, 80
fixed, 78
AWGN, 2, 12, 20, 30, 44, 87–89, 96, 106
coherence bandwidth, 7, 9, 12, 14, 16
coherence time, 9, 12–13, 96, 107
Doppler spread, 9, 12–13, 16
fading, 9, 12–13, 87–88, 90, 93–97, 105, 107
flat, 87–90, 92, 94, 106
frequency flat, 12, 14, 16
frequency selective, 12–14, 16, 87, 89–90, 97–101, 103–104, 106–107
ISI, 14
microscopic, 3, 7, 9, 12–14, 16, 155–156, 160
Rayleigh, 7, 88
ISI, 57, 63, 90, 98–99, 101, 103, 106
multiaccess, 50, 52
multipath, 2–3, 5, 7, 9, 12–14, 16, 143, 156, 166
multipath dimension, 7
multipath
resolvable, 7, 12–14, 16, 89–90, 98, 104, 106
path loss, 3, 5–7, 13, 16, 155–156
exponent, 5, 7, 13, 16
Hata model, 6
Okumura model, 6
shadowing, 3, 6–7, 13–14, 16, 155–156
Shannon's capacity, 49
spatial, 70, 73

time varying dimension, 7, 9
Coda, 580–581, 583–584, 587–588, 590, 595
coexistence, 309–310, 313–316, 318, 320, 322, 327–328
AFH, 310, 316, 318, 320, 322–324, 327–328
collaborative mechanism, 310, 313, 324, 327–328
MDMS, 310, 318, 322–324, 327–328
non-collaborative mechanism, 310, 313, 315–316, 318–319, 322–323
contention window, 271
CSI, 443, 491, 505
CSMA/CA, 269, 278
CSMA/CD, 264, 266
CTS, 268, 270
data caching, 501, 503, 517
consistency, 502–503, 507, 509–510, 515, 518
invalidation, 502, 505–515, 518
data replication, 579–580, 582–584, 587–588, 595
DHCP, 362
distributed file system, 580–583, 595
disconnected operations, 580, 584, 587–590, 595
weakly connected operations, 580, 588–589, 595
diversity, 88, 92–94, 96–98, 106
combining, 92, 95–96, 104, 106
equal gain, 92, 94–96, 106
maximal ratio, 92–94, 96, 106
frequency, 88, 96–97, 99, 101, 104, 106
receive, 97
selection, 92, 95–96, 106
spatial, 88, 96–97
time, 96
transmit, 97
doppler spread, 442
DSL, 499
DSSS, 272, 275, 283, 309
Ericsson, 285, 397
error concealment, 562–564, 571, 574–575, 577
Ethernet, 259, 262, 264
exposed terminal problem, 266–267
FDMA, 48, 54–57, 59, 62–64, 73–74, 77–78, 82, 86, 176, 184, 216
FEC, 294–295, 420
FHSS, 272–273, 283, 309, 312
frequency hopping, 310, 312, 315–316, 319–320, 322–323
Bluetooth, 289, 292, 294, 298
game theory, 668, 675, 678, 688, 690, 692–695
Nash equilibrium, 668, 675–676, 678
GPS, 604–606, 613
assisted GPS, 606
GSM, 13, 39, 42, 44, 49, 81–84, 86, 114, 116–117, 121–123, 131, 134, 154, 164–166, 175–176, 178–179, 181–182, 184, 186–189, 191–192, 194, 196–197, 199–200, 202–205, 207, 209, 212, 216–217, 289, 403, 520, 522–524, 526, 540–541, 546–549, 551–553, 555–556, 560, 590, 592, 595, 599, 604, 613
architecture, 176
AuC, 179
BSS, 176, 180–182, 184, 194
channels
AGCH, 188–189, 194
BCCH, 186, 191, 194
CCCH, 188
DCCH, 188
FACCH, 189, 191
PAGCH, 192–193
PBCCH, 192, 194
PCH, 188–189
PDCCH, 192
PPCH, 192, 194
PRACH, 192–193
RACH, 188–189, 193–194
SACCH, 188–189, 191
SDCCH, 189, 191
EDGE, 39, 42, 44, 114, 121, 123, 125, 128, 130, 132, 176, 217
GMSC, 196–197, 203, 205, 207, 218
GPRS, 114, 121–123, 125, 128, 130, 132, 175–176, 179–182, 184, 186, 192, 194, 209–210, 212, 214–217, 403, 413, 419
GGSN, 130–131, 180–181, 184, 194, 209, 211–212, 214–217, 428
mobility management, 211
SGSN, 130–131, 180–182, 184, 194, 209, 211–212, 214–217, 427
handover, 178, 191
HLR, 179–181, 184, 194, 196–197, 200, 203–205, 207, 212, 217, 548–549, 551–552, 555, 590–592, 595
hyperframe, 187, 191
IMSI, 182, 184, 197, 199–200, 205, 207, 210–212, 522
IWF, 194, 196, 217
LMU, 604
MAP, 179–180, 196, 199–200, 202–203, 205, 207
mobility management, 179
MSC, 176, 178–180, 182, 184, 194, 196–197, 199–200, 202–203, 205, 207, 212, 217
power control, 191
radio interface
AB, 186

DB, 186
FB, 186
NB, 186
roaming, 176, 179, 184, 197, 204–205, 207, 217–218
RLC, 205, 207, 209, 214–215
tromboning, 205, 207, 218
SIM, 176, 178, 182
SMLC, 604
SMS, 178
speech encoding, 178
superframe, 187
time slot, 178, 181, 185–188, 191–192, 216
TMSI, 182, 197, 199–200, 202, 212, 522
TRAU, 178, 182
VLR, 179–181, 184, 194, 196–197, 199–200, 202–205, 207, 212, 217, 548–549, 551–552, 555, 590–592, 595
Hamming code, 295
hidden terminal problem, 266, 268
HiperLAN, 330, 343–345, 347–348, 351–352, 354–355
home network, 282
HomeRF, 330, 338–339, 341–342, 344, 354–355
HTML, 395, 403
HTTP, 395
ICMP, 360–361
IEEE 802.15 TG2, 318, 320, 327
See also coexistence
IETF, 290, 379
infrared, 283
interframe space, 269
See also WLAN
IPng, 379
IPv4, 377–382, 384–387, 389–391, 394
IPv6, 377–382, 384–387, 389–391, 394
addressing, 381, 387, 394
autoconfiguration, 380
care-of address, 387
dual stack, 377–378, 384–385, 391, 394
headers, 378, 380, 382, 384, 387, 389, 394
mobility, 378, 380, 387, 389, 394
QoS, 380
security, 380, 391
tunneling, 377–378, 380, 384, 386, 391, 394
IrDA, 329–332, 334–335, 339, 354–355
ISM band, 309–313, 316, 327
Bluetooth, 286, 292, 303
IEEE 802.11a, 274
IEEE 802.11b, 275
IEEE 802.11g, 275
WLAN, 272–273
JavaScript, 403
Kasumi, 527
layered coding, 563, 566, 575, 577
Linux, 326
location based services, 597
personal, 597–600, 604, 608, 613
service model, 598
spatial analysis, 597, 600, 602–603, 608, 613
geocoding, 600, 602, 613
vehicle, 597–600, 604, 608, 613
macroblock, 566, 568–569, 577
message authentication code, 406
MMSE, 100–101
Mobile IP, 360, 365–366, 371–373
agent advertisement, 361–363
care-of address, 360–372
encapsulation, 363, 366, 369, 374
foreign agent, 360–366, 368–372
handoff, 361, 371
home address, 362, 366–367, 370
home agent, 360–361, 363–369, 371–372, 374
tunneling, 368–369, 373
modulation, 19, 25, 27, 36, 38, 44
8PSK, 42, 70, 78, 123, 125, 132
adaptive, 43
BPSK, 28, 37–38, 42, 69–70, 88–89, 93, 275
CCK, 275
DBPSK, 273
DQPSK, 273
GFSK, 273
GMSK, 42
O-QPSK, 39, 41–42
QAM, 36, 70, 78, 98, 275
QPSK, 36, 39, 41–42, 70, 78, 86, 98, 275
quarternary modulator, 37
time-orthogonal, 43
Motorola, 397
MPEG-4, 574–575, 577
multi-user communication, 47, 49, 61, 73, 81, 85
multi-user interference, 61, 67, 72–73, 77
multiaccess coding, 50, 52
multimedia, 562, 575
NAP, 326
NAT, 379, 386–387
NCP, 380
Nokia, 397
Nokia 6230i, 408
NS-2, 480
OFDM, 272, 274–276, 283–284
OMA, 397
OPNET, 480
P2P computing, 664, 666, 668, 670, 672–673, 686, 692, 694–695
free-riding, 663–665, 667
BitTorrent, 649, 667–668, 684
unchoking, 668

DHT, 650, 673
file sharing, 646–649, 651, 656, 658–659, 662, 664, 666–669, 672, 675, 680, 693
Gnutella, 648
incentives, 663–664, 667–668, 670–671, 678, 681, 690, 693, 695
auction, 664–665, 678–680, 688, 690, 694
barter, 665, 695
exchange, 664–665, 680, 690, 693–695
payment, 664, 670, 675–676, 684, 688–690, 693–695
pricing, 665, 669, 672, 678, 680, 690, 695
reciprocity, 664–666, 672, 692–695
reputation, 664, 666, 672–676, 679, 688, 690, 693–695
Napster, 648, 666–667
topology control, 646–647, 651, 653–656, 661–662
packet scheduling, 433–435, 437, 439, 443–445, 447, 449–454
channel adaptive fair, 452
effort fair, 446–448
fairness, 433–434, 437, 445–454
HSDPA, 220, 226, 236, 244, 247, 249–250, 252, 254–255, 453
general, 220
HS-DSCH, 252
HS-SSCH, 252
HSUPA, 254
outcome fair, 446–448, 450–451, 453
proportional fair, 434, 451–452
virtual compensation, 453
PAN
Bluetooth, 303
pathrater, 496
PCS, 117–119, 121
Phone.com, 397
polling
Bluetooth, 293, 302
WLAN, 270–272, 279
power conservation, 455–456, 458, 460, 468, 470, 472–473
power control, 458, 460, 468, 470–473
power management
idle mode, 455–456, 458, 463, 470, 472–473
sleep mode, 456, 458, 463–465, 467
PPM, 273
PPP, 290–291
PSNR, 562
QAM, 274–275
QoS, 260, 271, 276–279, 282–284
Bluetooth, 302
Bluetooth scatternets, 303
random backoff, 269–272, 280, 297
Rayleigh fading, 442
See also channel
RC4, 533–534
RED, 426
RFC 1883, 379
RFC 2068, 409
RFC 2460, 379
RFC 3775, 387
RTS, 268, 270
SDMA, 54, 70, 72, 86, 112
interference nulling, 72–73
MMSE, 72–73
sensor networks
Bluetooth, 304–305
Shannon C. E., 573
signal, 1–3, 5–7, 9, 12–14, 16
narrow band, 7, 9, 12, 16
wideband, 16
spread spectrum system, 101, 103, 105
SS7, 179
superframe, 342
TACS, 113, 116–117
TCP, 290, 371
congestion avoidance, 416
cumulative ACK, 414
duplicate ACK, 415
EBSN, 427
ECN, 426
ELN, 427
mark font, 426
mark tail, 426
New-ECN, 426
NewReno, 425
Reno, 415, 425, 430
retransmission, 414
slow start, 415
Tahoe, 425
wireless
AIRMAIL, 420
end-to-end approaches, 425
I-TCP, 422
link layer approaches, 419
M-TCP, 423
MTCP, 423
NewSnoop, 421
ReSoA, 423
RLP, 421
SACK, 425
Snoop, 421
split-connection, 422
TDMA, 48, 54–57, 59, 62–64, 73–74, 77–78, 82, 86, 176, 181, 184, 186–188, 192, 216
TKIP, 520, 533, 535–536
TLS, 395

UDP, 290, 363
UMTS, 14, 88, 121, 125–126, 128, 131–132, 164, 174, 176, 215, 413
  architecture, 221
  BCCH, 227
  BCH, 229
  CCCH, 227
  CN, 221
  CPCH, 229
  CPICH, 230
  CSCF, 223
  CTCH, 227
  DCCH, 227, 241
  DCH, 229
  DPCH, 232
  DSCH, 229
  DTCH, 227
  FACH, 229
  GGSN, 222
  HLR, 223
  MGCF, 223
  MRF, 223
  multi-code transmission, 125–126, 132
  OVSF, 125, 226
  P-CCPCH, 230
  PCCH, 227
  PCH, 229
  PCPCH, 232
  PDCP, 224
  PDSCH, 232
  PRACH, 232
  RACH, 229
  Rel 3, 223
  Rel 4, 223
  Rel 5, 223, 226, 255
  Rel 6, 131
  Rel 99, 131, 220–223, 226, 243, 252, 254–255
  RNC, 220
  RRC, 225
  RRC layer, 239
  S-CCPCH, 230
  SCH, 230
  SGSN, 222
  UE, 220
  UTRAN, 221
  VLR, 223
UWB, 307
VoIP, 545–546, 549, 555–557, 559–560
  H.323, 546–549, 551–553, 555–556, 560
  iGSM, 546–549, 551–553, 555–556, 560
    misrouting, 546, 553, 555, 560
    tromboning, 553, 555, 560
VSG, 583
WAP, 395–398, 400–404, 406–410, 412
  Bluetooth, 286, 290
  gateway, 396, 400–401, 406, 409, 411–412
  OTA, 401, 411
  profiles, 396, 408
  server push, 401–402, 404, 406, 409–412
  WAE, 290, 403–404
  WCMP, 407
  WDP, 404, 406–407
  WML, 396, 398–400, 404, 409, 412
  WMLScript, 398–400, 404, 409, 412
  WSP, 404–405
  WTA, 402, 404
  WTAI, 398, 402
  WTLS, 406, 409
  WTP, 396, 404–406
WCDMA, 527
WCMP, 407
wired-cum-wireless, 419
wireless sensor networks, 615–617, 642
  Berkeley Motes, 616
  bivariate polynomials, 621–623, 629, 643
  fully distributed, 616
  hierarchical, 616
  key establishment, 618, 631, 633, 635, 638, 640, 642–643
  key management, 615, 617, 619, 642–643
    deployment knowledge, 619, 629, 643
  key predistribution, 618–625, 627–629, 638, 642–643
  random key pool, 622, 630, 638, 643
  re-keying, 625
  TinyOS, 616
  trust bootstrapping, 616
WLAN, 14, 259–261, 274–277, 279–284, 359, 372, 497, 499, 517, 546, 556–557, 560
  AIFS, 278–279
  BSS, 260–263, 266, 271, 278
  DCF, 261, 269–272, 278, 283–284
  DIFS, 270, 278–280
  distribution system (DS), 260, 262–263, 266
  EDCF, 278–279, 283–284
  ESS, 262–263, 266
  HCF, 278–279, 283–284
  IEEE 802.11, 260–263, 266, 269, 272–274, 277–281, 330, 332, 338, 342, 348, 354–355, 520, 534, 540, 546
    DIFS, 559–560
    IFS, 556, 559
    SIFS, 560
  IEEE 802.11a, 260, 274–276, 281, 330, 347, 352, 355
  IEEE 802.11ag, 276, 284
  IEEE 802.11b, 49, 85, 260, 275–276, 533–534
  IEEE 802.11e, 260, 277–278, 282

IEEE 802.11g, 14, 275–276, 280, 284
IEEE 802.11i, 533, 536
IEEE 802.11n, 282
IEEE 802.11, 259–260, 273, 281
IFS, 269–272, 280
PCF, 261, 269–271, 277–279, 283–284
PIFS, 270–272, 279
SIFS, 270–271, 280
WEP, 520, 533–536, 541
Wi-Fi, 14, 84–85, 534
WPA, 520, 534–536, 541
XML, 395, 398, 403, 408, 411

This Page Intentionally Left Blank

WILEY SERIES ON INFORMATION AND COMMUNICATIONS TECHNOLOGIES

Series Editors: Russell Hsing and Vincent K. N. Lau

Wireless Internet and Mobile Computing: Interoperability and Performance
Yu-Kwong Ricky Kwok and Vincent K. N. Lau

The Wiley Series on Information and Communications Technologies (ICT) focuses on bridging the gap between advanced communication theories, practical designs, and end-user applications with respect to various next-generation wireless systems. Specifically, the ICT book series examines the practical difficulties of applying various advanced communication technologies to practical systems such as WiFi, WiMax, B3G, etc. In addition to covering how things are designed in the standards, we look at why things are designed that way so as to link up theories and applications. The ICT book series also addresses application-oriented topics such as service management and creation, end-user devices, as well as the coupling between end devices and infrastructure.

Series Editors contact information:

Dr. Russell Hsing
Executive Director
Telcordia Technologies Inc.
Email: thsing@telcordia.com

Professor Vincent K.N. Lau
Associate Professor, Dept. of EEE
Hong Kong University of Science and Technology
Email: eeknlau@ee.ust.hk

This Page Intentionally Left Blank