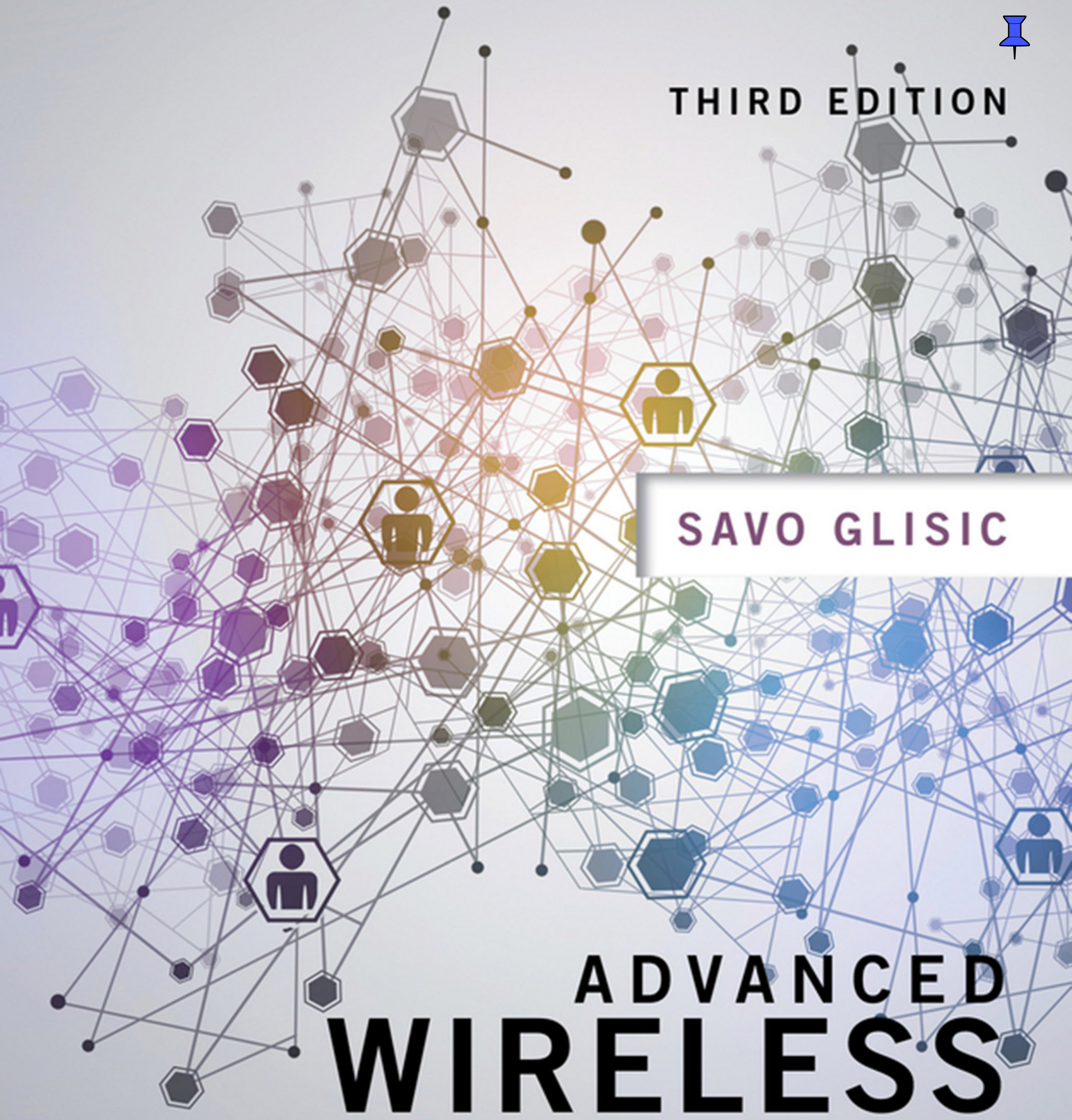


THIRD EDITION

SAVO GLISIC

ADVANCED WIRELESS NETWORKS

TECHNOLOGY AND BUSINESS MODELS

with website

WILEY

ADVANCED WIRELESS NETWORKS

ADVANCED WIRELESS NETWORKS

TECHNOLOGY AND BUSINESS MODELS

Third Edition

Savo Glisic
University of Oulu, Finland

WILEY

This edition first published 2016

First Edition published in 2006

Registered Office
John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Library of Congress Cataloging-in-Publication data applied for

ISBN: 9781119096856

A catalogue record for this book is available from the British Library.

Set in 10/12pt Times by SPi Global, Pondicherry, India

1 2016

Contents

Preface

Wireless communications has been developed so far through generations 1G to 4G with exclusive focus on improving the physical layer. This concept has at least two drawbacks: first, wireless channels cannot compete with optical networks when it comes to network capacity; second, the advantages of user mobility have not been emphasized enough. In the scenarios of future dense networks with a significant increase of user terminals and access points, wireless links in the wireless access concept in 5G will become shorter and shorter, asking for more frequent handoffs which jeopardize the reliability of the connections.

A significant part of the future networks will handle Internet of Things and People (IoTP) communications, where sophisticated physical layer solutions cannot be used. Human body implants will use simple solutions. For these reasons there is a common understanding that 5G will be about wireless networks rather than about wireless access to the networks. In the research of the enabling technologies for 5G, different communities focus on different solutions. Small cell technology, mmWave physical layer, cognitive networks, massive MIMO, spectra and infrastructure sharing in multi-operator network management, dynamic network architecture, user provided networks, and so on.

In the design and analysis of these networks a number of powerful analytical tools are used, like: convex, dynamic and stochastic optimization, stochastic geometry, mean field theory, matching theory, and game theory, as well as a number of tools used in economics/microeconomics.

This book advocates a concept where all these technologies will be simultaneously present in the future wireless networks and focuses on three main issues:

1. Design of heterogeneous networks that include all or a number of these technologies at the same time.
2. Optimization of such complex networks.
3. Design of efficient business models to exploit the limited resources of these networks.

Hence the subtitle of this book: *Technology and Business Models*.

The book is dedicated to the young generation of open-minded researchers, network designers, and managers who will make it happen.

Savo Glisic

1

Introduction

Generalized Model of Advanced Wireless Networks

In the process of evolving towards 5G networks, wireless networks are becoming more complex in both, the number of different functionalities they provide as well as in the number of users they serve [1]. Future 5G networks are expected to be highly heterogeneous (see Chapter 11) and to integrate cognitive network concepts [2, 3] (Chapter 9), heterogeneous solutions for the offload of cellular network traffic to WLANs [4, 5], multi-hop cellular networks (Chapter 8) including combinations of ad hoc (Chapter 4) and cellular networks [6, 7], and mobile to mobile (m2m) communications [8]. In order to analyze and control these networks, evolving towards complex networks structures, *efficient* modeling tools are needed.

Complex network theory (Chapter 14) has emerged in recent years as a powerful tool for modeling large topologies observed in current networks [9]. For instance, the World Wide Web behaves like a power-law node degree distributed network, wireless sensor networks like lattice networks, and relations between social acquaintances like small world networks. The concept of small world networks was first introduced by Watts and Strogatz [10] where a small world network is constructed via rewiring a few links in an existing regular network (such as a ring lattice graph). Later on, Newman-Watt [11] suggested a small world network constructed by adding a few new links (shortcuts) without rewiring existing links. The concept of small world can be introduced to wireless networks, typically to reduce the path length, and thus provide better throughput and end to end delay.

Several works have addressed the question of how to construct a wireless network topology in ad hoc and sensor networks (Chapter 5) in such a way that the small world feature is preserved [12–16]. Long range shortcuts can be created by adding wired links [17], directional beamforming [18] or using multiple frequency channels [19] concepts. In Ref. [9] it was demonstrated that small world networks are more robust to perturbations than other network architectures. Therefore, any network with this property would have the advantage of resiliency

Advanced Wireless Networks: Technology and Business Models, Third Edition. Savo Glisic.

Companion website: www.wiley.com/go/glisic0916

where the random omission of some vertices does not increase significantly the average path length or decrease the clustering coefficient. These features are highly desirable in future wireless networks where the availability of links and nodes can be uncertain. For these reasons, in this book we are interested to redesign heterogeneous wireless networks by including small world properties and frequency channels backups.

The considered network model, that we envision for 5G and further to 6G, includes the multi-hop concept to model future networks with dense user populations and enables mobile to mobile (m2m) connections which are already standardized. We see multi-hop cellular networks as an extension or generalization of the existing m2m concept. The potential users acting as relays may belong to different operators and as such may or may not want to cooperate. Consequently, the existence of those links will be uncertain. Some subareas of the cell will be covered by other technologies such as femto cells, small cells, or WLANs enabling the possibility for the cellular system to offload the traffic. The existence of those links depends on the relaying distance and coverage of the WLAN, as well as the cooperation agreement between the operators. In such a complex network, cognitive links might also be available with limited certainty due to unpredictable activity of the primary user (PU). Complex network theory will be used to aggregate all these characteristics of the network into a unified model enabling a tractable analysis of the overall system performance.

Despite of the extensive work in each of the previous fields, to the best of our knowledge, our book is the first to provide a unified model of the network that will include simultaneously all those technologies. The dynamic characteristics of the network results into a dynamic network topology. The work developed by [20] represents the first attempt to model the link uncertainty by complex networks concepts, although in this work, the uncertainty was a consequence only of fading and dynamic channel access. More specifically, our book emphasizes the following aspects of the design and analysis of complex heterogeneous wireless networks:

1. A unified model for heterogeneous wireless complex networks based on the probabilistic characterization of the node/link uncertainty. The model captures the existence of uncertain and time varying links and nodes inherently present in the latest solutions in wireless networks.
2. Analytical tools for the unified analysis of the multi-operator collaboration, m2m transmission, different traffic offloading options, and channel availability in cognitive heterogeneous networks.
3. Redesign of heterogeneous networks by using specific techniques to systematically add, in a controlled way, network redundancy in order to increase the network robustness to link/node failures.
4. Traffic distribution aware rewiring of the heterogeneous network.
5. A set of new routing protocols for such network.
6. Comprehensive analysis of the network in terms of average path length, clustering, robustness, power consumption, and complexity.

In this introduction we start with a general model of the future wireless network, referred to as *generic network model*, and later in separate chapters we elaborate in more detail each component of such network.

1.1 Network Model

We start by considering a macro cellular network where users transmit uplink by relaying to their adjacent users (neighbors) on the way to the base station (BS). Multi-hop transmission is modeled by considering a virtual cell tessellation scheme presented in Figure 1.1.1, where the macro cell of radius R is divided into inner hexagonal subcells of radius $r < R$. This partition is not physically implemented in the network but rather used to capture the mutual relations between the terminals in the cell that are potentially available for relaying each other's messages. For this purpose, it is assumed that, if available, a potential, ready to cooperate transmitter/receiver is on average situated in the center of each subcell.

We assume that within a cell the BS is surrounded by H concentric rings of subcells. For the example in Figure 1.1.1, $H = 3$. The shortest path (in hop count) between the user location and the BS is given by the hop index h, $h = 1, \ldots, H$. Due to the terminal unavailability, there may be routes towards the BS where the length of the path is longer than h. The number of subcells per ring is $n_h = 6 \cdot h$ and the number of subcells per cell is $N = 3H(H + 1)$.

In the sequel, we present a number of characteristics of heterogeneous networks that lead to the uncertain existence of nodes and links. Node percolation will be used to model and quantify the unavailability of users to relay as a consequence of lack of coverage or terminals belonging to a different operator with no mutual agreement for cooperation. When cognitive links are used, link percolation is used to model the link unavailability due to the return of the PU to the channel. These options will be elaborated in detail in the subsequent subsections.

1.1.1 Node Percolation

1.1.1.1 Multiple Operator Cooperation in Cellular Network

Here we model the scenario where a number of operators coexist in the cellular network. It is assumed that a single operator i has a terminal available in a given subcell with probability p_{o_i}. In a multi-operator cooperative network, a terminal will be available for relaying in the same subcell if at least one operator has a terminal at that location. This will occur with probability $p = 1 - \prod_i (1 - p_{o_i})$.

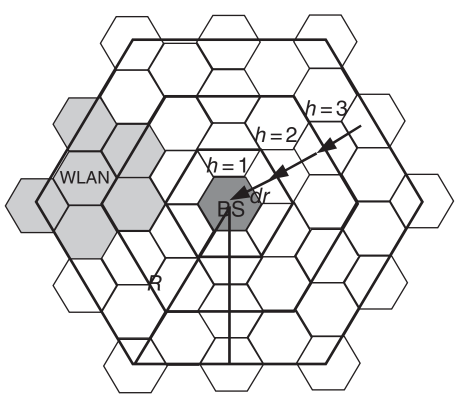


Figure 1.1.1 Macro cell tessellation

This probability is higher for higher number of operators willing to cooperate. In general, this will result into a reduction of the relaying route length. If the operators cooperate and let their users to flexibly connect to the BS that is more convenient to them, the network capacity of both operators will be improved. Thus, a better performance of the network will be obtained in the multi-operator cooperative scenario, as will be shown later in this chapter. The node unavailability for the message forwarding in complex network terminology is referred to as node (or site) percolation.

1.1.1.2 Multiple Operators in Cooperation with Multiple Technologies

In general multiple technologies will be available in a heterogeneous network. Each technology has its own characteristics which enables more appropriate AP choice at a specific place and time based on the users' requirements. Figure 1.1.1 shows an example of a cellular network overlapping in coverage with a WLAN. In the analysis, we will be interested to generalize this model as follows. The relative coverage between the cellular network and other access technologies, that is WLAN will be characterized by probability p_{wlan} which is the probability that in the next hop the connection will have the opportunity to make a handoff to a different technology and so, terminate the route. The probability $p_{wlan} = A/A$ is calculated as the ratio between the coverage areas of other technologies A_h and the coverage area of cellular network A_c. This can be easily generalized to introduce other traffic offloading options like small/femto cells or other multitier elements like micro and pico cells.

1.1.1.3 Modeling *m*2*m* Links

In the analysis, we will consider the possibility that every next relay on the route will be a final destination of an *m*2*m* link with probability p_{m2m}. This parameter depends on the probability that the session is within the same cell and parameter N representing the number of subcells in the network.

The simplest model will assume that for a specific session $p_{m2m} = (N_{m2m}/N)/N_{m2m} = 1/N$, where N_{m2m} is the average number of *m*2*m* connections per cell. N_{m2m}/N represents the probability that the given adjacent node is a sink for an *m*2*m* connection and $1/N$ is the probability that it is a sink for a specific session out of N_{m2m} such sessions.

1.1.2 Link Percolation—Cognitive Links

In the case that cognitive links are used for relaying, which means that we are establishing the routes for the secondary users (SUs; belonging to a secondary operator, SO), there are two related problems that should be considered. The first one is the link availability at the moment when routing/relaying decision is being made and the second one is the PU return probability that will interrupt the ongoing relaying and force the SU user to try it again with a new option.

We assume that spectrum sensing is perfect [3]. Since this problem belongs to the physical layer technology and has been extensively covered in the literature we will not discuss it within this book. We also consider that due to the uncertainty of the PU's activities, the SO cannot obtain spectrum availability information in advance for the entire message transmission period.

We model this uncertainty by defining a probability of return of the PU to the channel currently allocated to the SU, denoted as p_{return}.

Let us assume that call/data session arrivals follow a Poisson distribution with rate λ_p and λ_s for the PU and SU, respectively. The average probability p_{n_p} that in a given moment n_p out of c channels are being used in PO network (the system is in state n_p) can be obtained as a solution of birth death equations for conventional M/M/c system for data session and M/M/c/c system for voice applications [21].

We assume that the average service time of the SU is $1/\mu_s$ so that, the probability of having k_p new PU arriving within that time is [21]

$$p_{k_p}(t = 1/\mu_s) = \frac{(\lambda_p t)^{k_p}}{k_p!} e^{-\lambda_p t} = \frac{(\lambda_p/\mu_s)^{k_p}}{k_p!} e^{-\lambda_p/\mu_s} \tag{1.1.1}$$

The probability that a specific channel among $c - n_p$ channels is allocated to one of the k_p new arrivals is $k_p/(c - n_p)$. So, the average corruption probability due to the PU return will be

$$\begin{aligned} P_r(n_p) &= \sum_{k_p=0}^{c-n_p} \frac{k_p}{c-n_p} p_{k_p}(t = 1/\mu_s) \\ &= \sum_{k_p=0}^{c-n_p} \frac{k_p}{c-n_p} \frac{(\lambda_p/\mu_s)^{k_p}}{k_p!} e^{-\lambda_p/\mu_s}. \end{aligned} \tag{1.1.2}$$

The previous expression can be further averaged out over n_p to give the average PU return probability defined as

$$p_{return} = \sum\nolimits_{n_p} P_r(n_p) p_{n_p}. \tag{1.1.3}$$

The models presented so far capture the uncertainty of nodes and links due to different characteristics of wireless networks. The network connectivity when all the previous phenomena are present in the network is analyzed in the next section by using an absorbing Markov chain.

1.2 Network Connectivity

In modeling network connectivity, we will start with the initial model from Figure 1.1.1 and all components described in the previous section. This initial model is then redesigned later by incorporating the concepts of *small world networks* and systematic introduction of frequency backup channels. In general, we assume that the network is using cognitive links when available. If a cognitive link is used and there is a PU return to the channel, the ongoing transmission will be aborted with probability p_{return}, given by (1.1.3), and the user will try another channel. If there is no PU return to the channel, the user will relay to the receiver of the m2m link if there is such receiver for a specific session in the neighboring subcells (probability p_{m2m}). This joint event will happen with probability p_{m2m} $(1 - p_{return})$. Otherwise, if there is no such receiver,

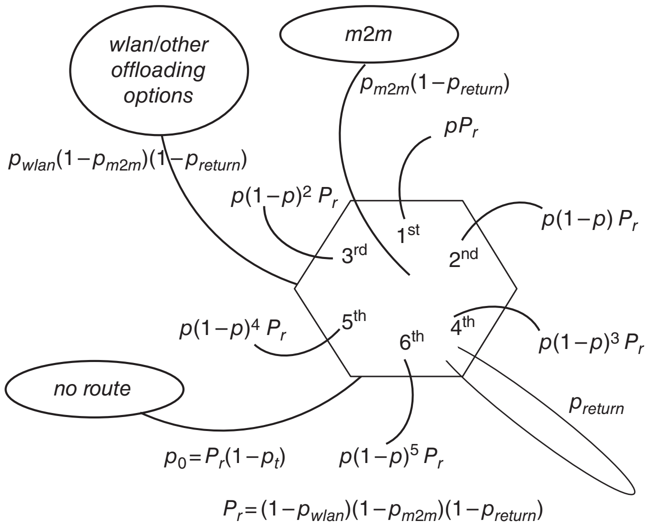


Figure 1.2.1 Connectivity alternatives (the direction of the adjacent users is chosen in increasing order of distance from the BS)

the user will relay to the WLAN if available (probability p_{wlan}) in the neighborhood, with probability $p_{wlan}(1-p_{return})(p_{wlan})$. In addition to WLAN, in general there will be also other options for traffic offloading (small/femto cells, or different tiers of cellular network like pico and micro cells). The offloading decision will be made with certain probability that depends on a number of parameters: AP availability cost of offloading, traffic distribution, terminal interface, and so on. For the purpose of the analysis in this paper all these parameters will be included in p_{wlan}. This is illustrated in Figure 1.2.1. If none of these two options is available and there is no return of the PU, the user will transmit towards BS by relaying to the neighboring subcells that will take place with probability

$$P_r = (1-p_{wlan})(1-p_{m2m})(1-p_{return}). \tag{1.2.1}$$

The probabilities of relaying to a specific adjacent subcell are indicated in Figure 1.2.1 where p is the terminal availability probability. In each subcell, the user checks the adjacent relay that is in the direction with *the shortest distance towards the BS/AP*. The adjacent relay will be available with probability p as shown in Figure 1.2.1, and if available, relaying will take place as indicated with probability pP_r. If this user is not available, then the protocol checks the availability of the next user in the order indicated in Figure 1.2.1. In general the potential relays closer to the direction of the BS are checked up first. More specifically, the protocol checks up the right user, which will be available with probability p, so the probability that this transition will take place is $p(1-p)P_r$. In the case of non-availability the protocol will check the left user. The protocol continues in the same way until it gets to the last adjacent user where relaying will take place with probability $p(1-p)^5P_r$. If none of the above options is available, then the route will not be established with probability p_0 as indicated in Figure 1.2.1. As result, the routing protocol will be referred to as AP location aware routing. Parameter p_0 will be used as a key indicator of the node robustness to link and node failure (unavailability).

In general, we denote by p_n the probability of relaying to adjacent user n obtained as

$$p_n = p(1-p)^{n-1} P_r,\ n = 1, \ldots, 6 \tag{1.2.2}$$

where P_r is given by (1.2.1). Thus, the overall relaying probability to any adjacent subcell is obtained as

$$p_t = \sum\nolimits_n p_n. \tag{1.2.3}$$

In a complex system, the simultaneous impact of the number of factors described in Section 1.1 is included by using the equivalent value of parameter p equal to the product of the individual probabilities characterizing the corresponding phenomena. For example, in the system with two operators with terminal availabilities p_1 and p_2, respectively, the equivalent terminal availability probability is given by

$$p = p_{eq} = 1 - (1-p_1)(1-p_2). \tag{1.2.4}$$

So, the relay will be available if the terminal from at least one operator is available.

1.3 Wireless Network Design with Small World Properties

1.3.1 Cell Rewiring

In the previous section, the network connectivity is considered from the point of view that the BS is the main target (destination) in the routing protocol. This means that most of the traffic is intended for destinations out of the cell. In this section, we focus our interest on the scenarios where most of the traffic remains within the cell and we are primarily interested to improve connectivity among the nodes within the cell. This is typical office scenarios where most of the traffic flows between the interoffice computers, computers and printers, interoffice voice and video communications, and so on. Later on, we will generalize the network model to include multiple cells in the overall complex network.

We start by indexing the subcells along the spiral presented in Figure 1.3.1 and unfolding the spiral into a lattice that will be referred to as *s-lattice*. The lattice obtained this way has similar form as those used in the classic literature of the complex networks theory [10, 11, 22, 23].

In a conventional one-dimensional lattice connections are established between all vertex pairs separated by k or less lattice spacing. The small-world model [10, 22, 23] is created by choosing at random a fraction of the edges in the graph and moving one end of each to a new location, also chosen uniformly at random. In a slight variation on the model in [10, 11] shortcuts are added randomly between vertices, but no edges are removed from the underlying one-dimensional lattice.

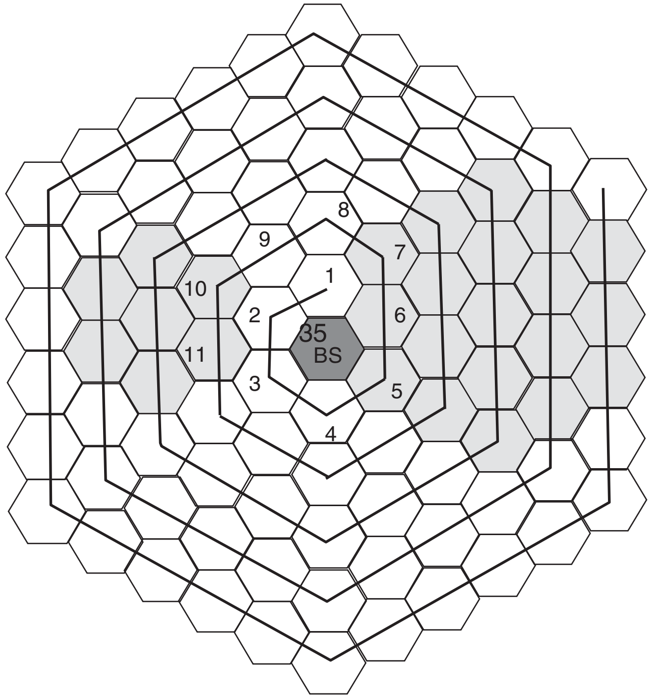


Figure 1.3.1 *s-Lattice* parameters

One can see that in *s-lattice*, obtained by unfolding the spiral from Figure 1.3.1, each vertex is connected to six neighbors. Different from the conventional lattice, the adjacent neighbors on the spiral are not adjacent neighbors in the lattice any more. This fact for itself brings the elements of rewiring or adding additional short cuts. More precisely one can see in Figure 1.3.1 (left hand side in shade) that each node in the h-th round of the spiral, is connected to two adjacent nodes ($k = 1$) with the same h, two adjacent nodes on the $(h-1)$-th round of the spiral and two adjacent nodes on the $(h+1)$-th round of the spiral.

If the coverage of transmission is extended to include two layers of subcells (lattice range $k = 2$) around each node (see the right hand side of Figure 1.3.1 in shade) then each node in the h-th round of the spiral, is connected to four adjacent nodes with the same h, four adjacent nodes on the $(h \pm 1)$-th round of the spiral and three adjacent nodes on the $(h \pm 2)$-th round of the spiral. One should notice that for the nodes located at the corners of the spiral ($\theta = 30 + 60n$, $n = 1,\ldots,6$ with respect to the BS), the size of the clusters at the rounds $h + \Delta h$ and $h - \Delta h$ are not equal. This is illustrated in Figure 1.3.2 for nodes 2 and 3 of the spiral in Figure 1.3.1.

Formally, parameter k for *s-lattice* means that each node will be connected to the $2k + 1$ clusters located on adjacent rounds of the spiral within distance $\Delta h \leq k$ with each individual cluster size $\leq k$.

Let us denote by $u(h, \theta)$ the user (network vertex) located in hop h and angle θ. In vector representation, its location is given as $\vec{u}\,(h,\theta) = h \cdot d_r \cdot e^{j\theta}$ where d_r is the relaying distance. The locations of its adjacent relaying users connected for certain lattice range k are given in the Appendix A.1. The *s-lattice* with shortcuts will be referred to as *s(sc)-lattice*.

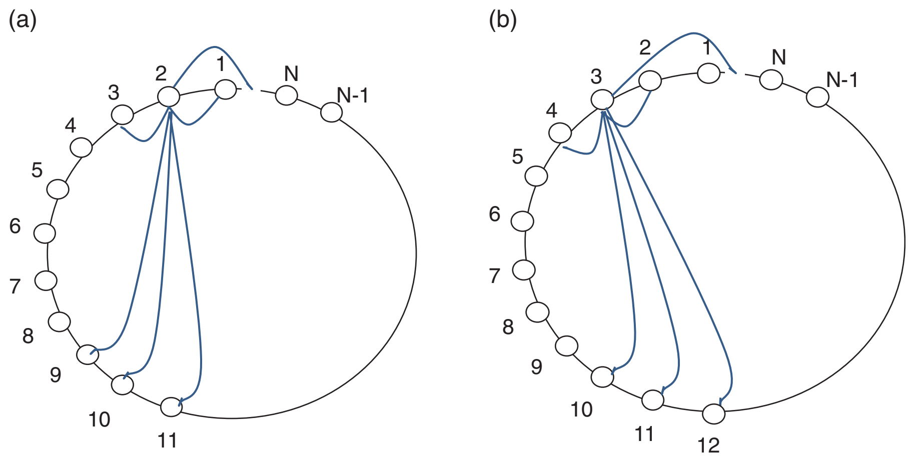


Figure 1.3.2 *s-Lattice* connection model for: (a) user 2 and (b) user 3

1.3.2 Traffic Distribution Aware Rewiring

It is intuitively clear that from the routing and delay point of view we will need a shortcut between nodes with high traffic density. On the other hand, a direct link between nodes far away from each other would require high power to maintain it. In order to accommodate these contradictory requirements, we suggest a traffic distribution aware rewiring where the shortcuts are established, following one of the options provided below, with probability

$$p_{ij} \propto \lambda_{ij}. \tag{1.3.1}$$

By considering the power consumption, (1.3.1) can be modified as

$$p_{ij} \propto \lambda_{ij}/P_{ij}, \tag{1.3.2}$$

or equivalently,

$$p_{ij} \propto \begin{cases} \lambda_{ij}; P_{ij} \le P_{threshold} \\ 0; P_{ij} > P_{threshold} \end{cases}. \tag{1.3.3}$$

These probabilities may be also obtained as a solution of the more sophisticated optimization problem with more complex utility function.

In practice, the shortcuts can be implemented by using separate m2m channels from the macrocell or equivalently, by considering channel reuse factor 1 and scheduling the transmissions in different slots.

In the case of rewiring, referred to as *s(r)-lattice*, the rewired link will be removed and reconnected randomly to another node. For both, the *s(sc)-* and *s(r)-lattices*, a new set of protocols will be developed later.

1.3.3 Multicell Rewiring

Multiple cells can be interconnected by using two way spiral 2*ws-lattice* with 2*N* nodes, as shown in Figure 1.3.3. The rewiring (or adding shortcuts) is performed between the two randomly chosen nodes from the whole network. Physically, this can be implemented by using the network backholes and direct link (macrocell or WLAN) from the nodes to the nearest backhole access point for rewiring.

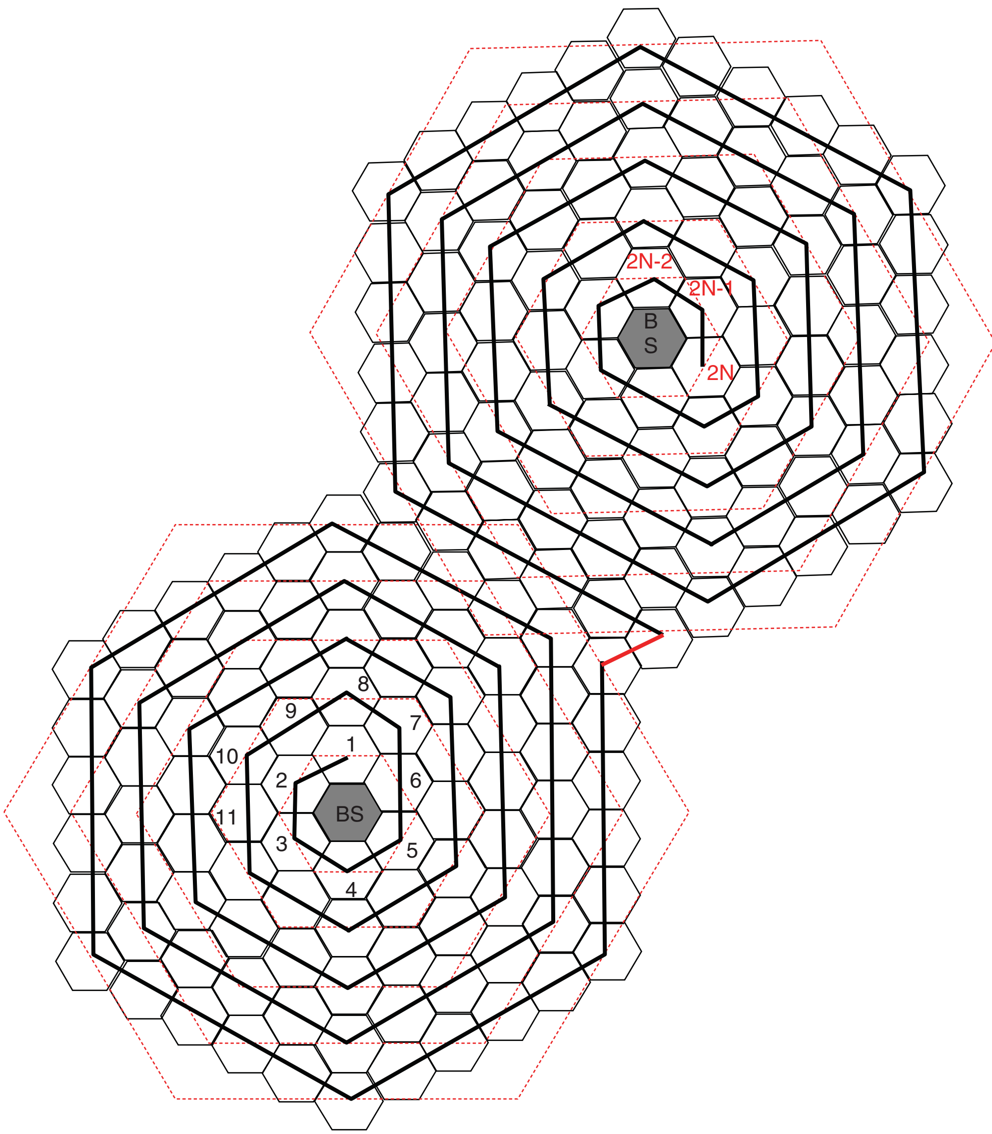


Figure 1.3.3 *2ws-Lattice*

1.4 Frequency Channels Backup

In this section, in addition of the small world properties we consider the possibility that a number of additional channels (either cognitive or purchased licensed channels) will be available for relaying. There are a number of ways how additionally purchased licensed channels can be made available to increase the overall network robustness to the link and node failure. The PO can sell the channel with respect to:

Area (A sell)
*per **m**acro cell*
*per constalletion unit (**s**ubcell)*

Number of frequency channels (F sell)
*one (**1**) or*
***k**$_f$ channels*

*Time the contract is valid (**t** sell)*
*temporal (per **s**ession) or*
***f**ixed time sell*

In the sequel, we will use *A/F/t* notation for an *A* sell / F sell / t sell contract. As an example, a $m/k_f/s$ contract refers to the sell on the **m**acro cell area $\boldsymbol{k_f}$ channels for the duration of a given **s**ession. Depending on the type of the sell, different effects will be achieved with respect to the network robustness enhancement.

1.4.1 m/k$_f$/s *Contract*

We characterize the network state with (n_p,n_s) where n_p is the number of temporally active users in the primary network and n_s is the same parameter for the secondary network. For a given overall number of available channels c, PO will keep b_p channels as its own backup and is ready to temporally sell to SO $c-(n_p+b_p)$ channels. The SO will buy b_s channels for its own back up and the rest of the free channels will be used as cognitive channels. Parameter b_s is limited to $b_s < k_f$ and can be represented as

$$b_s = \begin{cases} k_f, & c-\left(n_p+b_p\right) \ge k_f \\ c-\left(n_p+b_p\right), & c-\left(n_p+b_p\right) \le k_f \end{cases}. \tag{1.4.1}$$

1.4.2 Random Redundancy Assignment (R^2A)

In this case, the backup channel is randomly assigned to n_s users resulting in backup probability in secondary network defined as $p_{bs} = b_s/n_s$.

1.4.3 On Demand Redundancy Assignment

In this case, the redundant channel is assigned to the terminal after s successive returns of PU to the channel. This can be modeled as

$$p_1 = p_{return}^{s} \tag{1.4.2}$$

$$p_i = \binom{n_s}{i} p_1^i (1-p_1)^{n_s - i} \tag{1.4.3}$$

$$p'_{bs} = \sum_{i=0}^{k_f - 1} p_i \tag{1.4.4}$$

where (1.4.2) defines the probability that s successive returns have occurred after which the subcell demands for a backup channel. Parameter p_i is the probability that out of n_s active SU, i users are using the backup channel. Finally, (1.4.4) defines the probability that at least one out of k_f leased channels is free to be allocated to the new demand. The optimum value of parameter s is obtained as

$$\begin{aligned} s &= \arg \max_s p'_{bs}/s \\ &= \arg \max_s \left[\frac{1}{s} \sum_{i=0}^{k_f - 1} \binom{n_s}{i} p_{return}^{is} \left(1 - p_{return}^{s}\right)^{n_s - i} \right] \end{aligned} \tag{1.4.5a}$$

Equation 1.4.5a searches for the value of s that maximizes the probability that at least one out of k_f leased channels is free to be allocated to the new demand. For higher s, SUs will need to wait longer and hope that there will be no additional returns of the PU so that they can finally transmit without asking for the backup channel. It is intuitively clear that higher s will reduce the probability of having i SUs needing backup channels, which is defined by (1.4.3), and thus increase the probability, once the backup channel is requested, that there will be a backup channel to meet such a request as given by (1.4.4). On the other hand, we cannot allow s to be too high since this will increase the overall delay of message delivery to the access point. Therefore, the utility function in (1.4.5a) is obtained by dividing P'_{bs} by s. This utility function can be further modified to obtain s as

$$\begin{aligned} s &= \arg \max_s p'_{bs}/sl_r \\ &= \arg \max_s \left[\frac{1}{sl_r} \sum_{i=0}^{k_f - 1} \binom{n_s}{i} p_{return}^{is} \left(1 - p_{return}^{s}\right)^{n_s - i} \right] \end{aligned} \tag{1.4.5b}$$

$$\begin{aligned} s &= \arg \max_s p'_{bs}/sl \\ &= \max_s \left[\frac{1}{sl} \sum_{i=0}^{k_f - 1} \binom{n_s}{i} p_{return}^{is} \left(1 - p_{return}^{s}\right)^{n_s - i} \right] \end{aligned} \tag{1.4.5c}$$

In (1.4.5b), s is optimized for each route of length l_r separately and in (1.4.5c) for the whole network by using the average value of the route length l. The joint optimization of (s,k_f) is obtained by

$$\begin{aligned}\{s,k_f\} &= \underset{s,k_f}{\arg max} p'_{bs}/k_f sl \\ &= \underset{s,k_f}{\arg max} \left[\frac{1}{k_f sl}\sum_{i=0}^{k_f-1}\binom{n_s}{i} p_{return}^{is}\left(1-p_{return}^{s}\right)^{n_s-i}\right]\end{aligned} \quad (1.4.5d)$$

The optimization problems defined by (1.4.5d) maximize the probability that there will be a backup channel available once the user asks for it. The alternative optimization can be defined as minimizing the time to get the backup channel τ_{asq} after the first return hits the SU. The terminal will ask for the backup channel after s successive hits of PU return. If there is no backup channel available, it will repeat the procedure. This can be defined by

$$\begin{aligned}\{s,k_f\} &= \min_{s,k_f} \tau_{acq} \\ &= \min_{s,k_f} \left(s\left(1\cdot p'_{bs}\right) + 2s\left(1-p'_{bs}\right)p'_{bs} + 3s\left(1-p'_{bs}\right)^2 p'_{bs} + \dots\right) \\ &= \min_{s,k_f} s/p'_{bs}\end{aligned} \quad (1.4.5e)$$

The previous optimization problem will favor high values of k_f which is economically inefficient. A modified version defined as

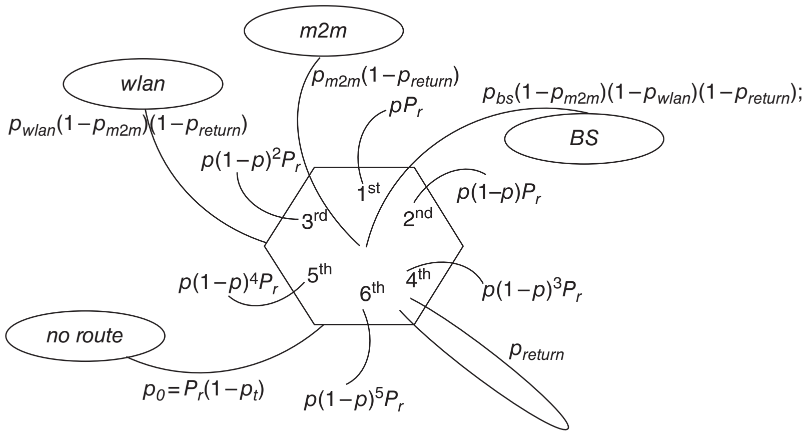
$$\{s,k_f\} = \underset{s,k_f}{\arg min}\, k_f \tau_{acq} \quad (1.4.5f)$$

will minimize the channel acquisition time with acceptable number of channels leased for backup purposes. One should notice that although different initial objectives have been set in the definition of the optimization problem, we ended up that utility function in (1.4.5f) is the reciprocal value of the one in (1.4.5d). Since the former searches for the minimum value of the utility and the latter for its maximum, the optimum values of the parameters are the same.

1.5 Generalized Network Model

In the model described in Section 1.4, we have to precisely define subcell transition probabilities for each subcell and solve the complete Markov model. The next level of abstraction is to randomize the position of the subcell with respect to the BS. This can be modeled by introducing an absorbing state labeled by BS as shown in Figure 1.5.1. The probability for a subcell of being a neighbor to the BS is $p_{bs} = 6/N$. Then, relaying to the neighboring subcells will now take place with probability

$$P_r = (1-p_{wlan})(1-p_{m2m})(1-p_{return})(1-p_{bs}). \quad (1.5.1)$$



$$P_r = (1-p_{wlan})(1-p_{m2m})(1-p_{return})(1-p_{wlan})(1-p_{bs});\ p_{bs} = 6/N$$

Figure 1.5.1 Connectivity alternatives for the generalized model (the direction of the adjacent users is chosen randomly)

This graph can be used for the system analysis when the terminal does not know the BS/AP position. For the simplicity of the terminals, the message is forwarded randomly to a neighbor unless an AP is available. This can be justified in the network with high density of the access points. We refer to this option as *blind* (*or hot potato*) *routing*.

If the routing protocol has the necessary information to preselect the access point, the generalized graph from Figure 1.5.1 will be reduced. The message is intended for the preselected access point and none of the other access points is of interest. So, they are removed from the graph. This may be either m2m final destination or a closest access point selected by the routing protocol in accordance with some optimization criteria. We will refer to this option as *context aware routing*.

1.6 Routing Protocols Over *s*-Lattice Network

Modeling *s-lattice* with shortcuts, referred to as *s(sc)*-lattice, requires modifications in the relaying probabilities from Section 1.3. For these modifications, we introduce the Two Layer Routing (2LR) protocol defined below where *i* refers to the index of the user (vertex) and *j* to the AP.

Protocol 1: 2LR

```
1.  for i=1,..,N
2.    set destination node index j=0
3.    if there is a shortcut between user i and j=0,
    transmit directly to j=0,
4.  otherwise
    transmit to the adjacent users j, j≠0, by 1 layer
    protocol (1L) as described in Fig.1.2.
5.  end
```

The state transition probabilities for such protocol are given as

$$p_{ij}^{(2)} = \begin{cases} p_{i0} = p_{sc}/N, & j = 0 \\ (1 - p_{i0}) p_{ij}^{(1)}, & j \neq 0 \end{cases} \tag{1.6.1}$$

where $p_{ij}^{(1)}$ are the state transition probabilities that correspond to the one layer routing ($1LR$) protocol and p_{sc} is the probability of a shortcut.

An enhancement of the previous protocol, referred to as e2LR protocol, is designed for the situation where it is not possible to have a shortcut directly to the AP. Instead, we will consider the possibility of having a shortcut between user (vertex) i and any user (vertex) w located in hop, $h_w < h_i$.

Protocol 2: e2LR

```
1.  for i=1,..,N
2.      set destination node index j=0
3.      find the set S = {w/h_w < h_i} of candidate users
     w, located in hop h_w<h_i
4.  if S≠∅ then, establish a shortcut between user i
      and each w, w∈S
5.  otherwise
     transmit to the adjacent users j, j≠0, by 1 layer
     (1L) protocol as described in Fig.1.2.
6.  end
```

If we denote by $C_w = |S|$ the number of candidate users w for shortcut, then the overall probability that there will be shortcut is $C_w p_{sc}/N$. The state transition probabilities for such protocol are given as

$$p_{ij}^{(e2)} = \begin{cases} p_{iw} = p_{sc}/N, & j = w \in S \\ \sum_w (1 - p_{iw}) p_{ij}^{(1)}, & j \neq w \end{cases} \tag{1.6.2}$$

where p_{iw} is the probability of user i having a shortcut with a particular user w and S is the set of nodes with hopping distance $h_w < h_i$.

A third option for the two layer routing protocol is the sequential protocol, s2LR, where we also consider first the possibility of having a shortcut directly to the AP. In the case that there is no such option, we check the possibility to have a shortcut to any user located in hop $h_w = h_j + 1$. If it is not possible to have such shortcut then, we check the users in $h_w = h_j + 2$. The protocol continues in the same fashion by using $h_w = h_j + \varepsilon$ where $h_w < h_i$. After that, one layer protocol applies.

Protocol 3: s2LR

```
1.  for i=1,..,N
2.       set destination node index j=0
```

3. *if there is a shortcut between user i and j=0, transmit directly to j=0,*
4. *otherwise*
5. $\varepsilon = 1$
6. *find the set* $S = \{w/h_w = h_j + \varepsilon,\ h_w < h_i\}$ *of candidates users w, located in hop* $h_w = h_j + \varepsilon,\ h_w < h_i$
7. *if* $S \neq \emptyset$ *then, establish a shortcut between user i and each w,* $w \in S$
8. *otherwise,* $\varepsilon = \varepsilon + 1$ *and go to 6.*
9. *If the previous options were not available then transmit to the adjacent users j,* $j \neq 0$*, by 1 layer (1L) protocol as described in Fig.2.*
10. *end*

The state transition probabilities for s2LR protocol are given as

$$p_{ij}^{(s2)} = \begin{cases} p_{iw} = \bar{p}_w p_{sc}/N, & j = w \in S \\ \sum_w (1 - p_{iw}) p_{ij}^{(1)}, & j \neq w \end{cases}$$
$$\text{with } \bar{p}_w = 1 - \sum_{\varepsilon=0}^{h_w - 1} p_{i(j+\varepsilon)} \tag{1.6.3}$$

where S is the set of nodes with hopping distance $h_w < h_i$.

The protocols e2LR and s2LR can be further modified by limiting the number of candidate users in hop $h_w < h_i$ to those located at distance $d(i,w) < d(i,j = 0)$. In this way, the relaying route will always go towards the destination and backwards segments are avoided. When the previous protocols consider this issue will be referred to as e2LRm and s2LRm, respectively.

1.6.1 Application Specific Routing Protocol

For delay sensitive traffic, the algorithm should reach the AP as soon as possible. In this case, s2LR will use $h_w = h_j + 1$ and continue incrementing h_w as $h_w = h_j + \varepsilon$. In the case of power limited terminals, the algorithm should use the closest neighbor for relaying the traffic. Now, s2LR will start with $h_w = h_i - 1$ and continue by negative increments of h_w as $h_w = h_i - \varepsilon$. For differentiated delay sensitivity service, a combination of the two previous options will be used.

1.7 Network Performance

For the analysis of the relaying process in the network, we map the tessellation scheme into an absorbing Markov chain depending on the targeted destination. The absorbing states represent the end of the route when the user has reached the BS, WLAN/off loading node, the end of the m2m communication or due to no route availability point (*nr*).

In general, relaying from subcell i to subcell j will take place with probability p_{ij} which can be arranged in a subcell relaying probability matrix $\mathbf{P} = \|p_{ij}\| = \|p(h,\theta;h',\theta')\|$ where the first set of indexes (h, θ) refers to the location of the transmitter and the second one (h', θ') to the location of the receiver. The mapping $i \to (h,\theta)$ and $j \to (h',\theta')$ is illustrated in Figure 1.3.1.

Following the schemes presented in Figures 1.2.1 and 1.3.1, in the sequel we derive general expressions for the subcell transition probabilities under the assumption that the scheduling protocol imposes constant dwell time in each subcell. Depending on the protocol, these probabilities are given by (1.2.2) and (1.6.1–1.6.3).

The probability that the user does not relay to any other user is p_0 and the process with probability $p_0 = 1 - p_t$ is transferred to an additional absorbing state *nr* (no route). Then, we reorganize the transmission matrix into $(N+1) \times (N+1)$ matrix of the form [24]

$$\mathbf{P}^* = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{1.7.1}$$

where N is the number of subcells. $\mathbf{I}$ is the $(N_A + 1) \times (N_A + 1)$ diagonal unitary matrix corresponding to the number of absorbing states including N_A BS/APs plus no route state *nr*. $\mathbf{0}$ is the $(N_A + 1) \times (N - N_A)$ all zero matrix, $\mathbf{R}$ is the $(N - N_A) \times (N_A + 1)$ matrix of transition probabilities from transient states to absorbing states and $\mathbf{Q}$ is the $(N - N_A) \times (N - N_A)$ matrix of transition probabilities between the transient states. By using notation $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$, the mean time for the process to reach any absorbing state starting from transient state i is [24]

$$\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{N-N_A})^T = T(\mathbf{I}-\mathbf{Q})^{-\mathbf{1}}\mathbf{1} = T\mathbf{N}\mathbf{1} \tag{1.7.2}$$

when the dwell time $T_i = T$ for each state i is the same. Otherwise, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{N-N_A})^T = (\mathbf{I}-\mathbf{Q})^{-\mathbf{1}}\boldsymbol{\nu} = \mathbf{N}\boldsymbol{\nu}$ where $\boldsymbol{\nu} = \mathit{columnvec}\{T_i\}$ and $\mathbf{1}$ is $(N - N_A) \times 1$ column vector of all ones.

This expression will be used in the next section in the definition of the network robustness. In general, the variance of that time is

$$\mathrm{var}\boldsymbol{\tau} = 2(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{T}\mathbf{Q}(\mathbf{I}-\mathbf{Q})^{-1}\boldsymbol{\nu} + (\mathbf{I}-\mathbf{Q})^{-1}\left(\boldsymbol{\nu}_{sq}\right) - \left[(\mathbf{I}-\mathbf{Q})^{-1}\boldsymbol{\nu}\right]_{sq} \tag{1.7.3}$$

where $\mathbf{T} = \mathit{diag\ matrix}\ \{T_i\}$, and if the dwell times are the same

$$\begin{gathered} \mathrm{var}\boldsymbol{\tau} = \left[(2\mathbf{N}-\mathbf{I})\mathbf{N}\mathbf{1} - (\mathbf{N}\mathbf{1})_{sq}\right]T^2 \\ (\mathbf{N}\mathbf{1})_{sq} = \textit{square of each component of } \mathbf{N}\mathbf{1} \end{gathered} \tag{1.7.4}$$

The average time to reach an absorbing state is

$$\tau_a = \mathbf{f}\boldsymbol{\tau} \tag{1.7.5}$$

where $\mathbf{f}$ is a row vector of the probabilities of the users' initial positions and $\boldsymbol{\tau}$ is a column vector given by (1.7.2). The probability that the Markov process starting in a transient state i ends up in an absorbing state j is b_{ij} where

$$\mathbf{B} = [b_{ij}] = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R} \tag{1.7.6}$$

The average probabilities of, accessing the BS, hand off to WLAN, reaching the m2m destination and no route are given as

$$\bar{\mathbf{p}}_{\mathbf{ac}} = (\bar{p}_{bs}, \bar{p}_{wlan}, \bar{p}_{m2m}, \bar{p}_{nr}) = \mathbf{fB} \tag{1.7.7}$$

where **f** is the vector of probabilities of initial user positions.

1.7.1 Average Path Length

The average path length is defined as the average length (in hop count) of the shortest path between any two nodes in the network. It is calculated as

$$l = \frac{1}{N(N-1)/2} \sum_{i,j} l_{i,j} \tag{1.7.8}$$

where $l_{i,j}$ is the shortest distance (in hop count) between nodes i and j, and N is the number of subcells. Parameter $l_{i,j}$ can be obtained by (1.7.2) for each particular destination j modeled as AP and represented as absorbing state in the Markov model. The entries of the relaying probability matrix **P**, $p(h,\theta;h',\theta')$ for that analysis are given by (1.2.2) and (1.6.1–1.6.3), depending on the protocol used. This result will be compared with the same result for the small world network model where $l \sim logN$ [10, 11, 22].

1.7.2 Clustering

In many networks it is found that if vertex i is connected to vertex w and vertex w to vertex j, then there is a heightened probability that vertex i will also be connected to vertex j. In the field of social networks, this is often illustrated by the interpretation "*a friend of my friend is also my friend*." In terms of network topology, clustering means the presence of a heightened number of triangles in the network (sets of three vertices each of which is connected to each other's) [23]. Based on this, in our model, we can quantify the clustering coefficient C as

$$C = \frac{\sum_i \sum_w \sum_j p_{iw} p_{wj} p_{ji}}{\sum_i \sum_w \sum_j p_{iw} p_{wj}}, \; i \neq w \neq j \tag{1.7.9}$$

where the numerator indicates the average number of triangles in the network and the denominator, the average number of connected triples of vertices defined as a single vertex with edges running to an unordered pair of others. In simple terms, C is the mean probability that two vertices that are neighbors of the same other vertex will themselves be neighbors as well.

1.8 Node, Route, Topology, and Network Robustness

Here, we explicitly define the node, route, topology, and network robustness to physical node and link failure as well as to the channel corruption.

- *Node robustness* is defined as

$$\xi = 1 - p_0 = p_t \tag{1.8.1}$$

  where p_0 is the probability that none of the adjacent users is available (probability of no route) and p_t is given by (1.2.3).
- *Route robustness* is defined as the probability that a specific route from node i to the access point will physically exist and is given by (1.2.5).
- *Topology robustness* is the average probability of physical existence of the routes in the network. It can be defined either by averaging (1.7.6) or alternatively as

$$\xi^{(n)} = \sum_i \sum_j \xi^{l_{ij}} p\left(l_{ij}\right) \tag{1.8.2}$$

  The alternative definition will be also used in the numerical analysis

$$\xi^{(N)} = \xi^{l}, \tag{1.8.3}$$

where l is the average path length in the network defined by (1.7.8).

It will be shown in the numerical results that the network resilience is significantly improved with the small world network properties.

- *Network robustness* includes the impact of the channel corruption p_{return} and is defined again by (1.8.2) and (1.8.3) where instead of l parameter τ is used.

When frequency channels are available for backup as explained in Section 1.4, (1.8.1) becomes

$$\xi = 1 - p_0 + p_0 p_{bs}\left(1 - p_0^{nc}\right) \tag{1.8.4}$$

where p_{bs} is the backup probability in secondary network and p_0^{nc} is no route probability in non-cognitive network. One should keep in mind that $\xi = \xi\left(n_p, n_s\right)$ depends now on the system state (n_p, n_s) and the results so far obtained for the network topology robustness defined by (1.8.2) and (1.8.3) should be averaged using the state distribution function $p(n_p, n_s)$. The modification of the analysis for different type of the contracts can be derived in similar way.

If we consider on demand redundancy assignment (ODRA) then, the node robustness can be obtained as

$$\xi = 1 - p_0 + p_0 p_1 p'_{bs}\left(1 - p_0^{nc}\right) \tag{1.8.5}$$

where p'_{bs} is given by (1.4.4). By using (1.8.5) in (1.8.2) or (1.8.3) we can obtain again the network robustness.

1.9 Power Consumption

Let us now discuss the average power consumption for the connection between two nodes in the network. We will use notation *node* 0 for the destination node and *node i* to denote any source node on the hoping distance h_i. Parameter α_{iw} will represent the probability that there is direct link (shortcut) between nodes i and w, and P_{iw} the power needed for direct transmission from node i to node w. We use P_i to denote the average power consumption for the transmission from node i (on the hopping distance h_i) to the destination node 0. For any user located in hop 1, its power consumption is obtained as

$$\begin{aligned} P_1 &= \alpha_{10}P_{10} + (1-\alpha_{10})\alpha_{11}(P_{11}+P_1) \\ &\quad + (1-\alpha_{10})(1-\alpha_{11})\alpha_{12}(P_{12}+P_2) \end{aligned} \tag{1.9.1}$$

where the first term indicates the power needed to transmit directly to the destination denoted by P_{10} multiplied by the probability α_{10} that this transmission will happen. As the user is located in the first hop, the destination is in hop $j = i - 1 = 0$. If there is not such option then, the power needed to transmit to an adjacent user located in the same hop is given by the second term where $(1 - \alpha_{10})\alpha_{11}$ is the probability that this transmission will occur, P_{11} is the power needed to transmit to the adjacent user located in $w = i = 1$ and P_1 the transmission power for a user in hop 1 to reach the destination. As before, this transmission will not happen with probability $(1 - \alpha_{10})(1 - \alpha_{11})$. In that case, the user will relay to any adjacent user located in hop $w = i + 1 = 2$ with probability α_{12} and the total power needed to reach the adjacent user is P_{12} plus the transmission power for any user located in hop 2, P_2, to reach the destination. In general, the power consumption for any user i can be obtained by the following recursion,

$$P_i = \alpha_{i0}P_{i0} + \sum_{w=1}^{i+1} \alpha_{iw}(P_{iw}+P_w)\prod_{\xi=0}^{w-1}(1-\alpha_{i\xi}) \tag{1.9.2}$$

where

$$\alpha_{iw} = \begin{cases} p, & w = i-1 \\ p + p(1-p), & w = i \\ p + p(1-p) + p^2(1-p), & w = i+1 \\ p_{sc}/N, & |w-i| > 1 \end{cases} \tag{1.9.3}$$

The transmission power from a user located in hop i to a user located in w is $P_{iw} = (d_{iw})^{\alpha}P$ where d_{iw} is the transmission distance between both users, α is the propagation constant and P is the power needed for one hop transmission.

The overall power consumption in the network can be obtained as $P_t = \sum_h P_h n_h$ where P_h is obtained by (1.3.3) and n_h is the number of users in hop h.

1.10 Protocol Complexity

At the beginning of this section, we derived general expressions for the subcell transition probabilities under the assumption that the scheduling protocol imposes constant dwell time in each subcell. In this subsection, we are interested in analyzing the number of iterations Δ that the

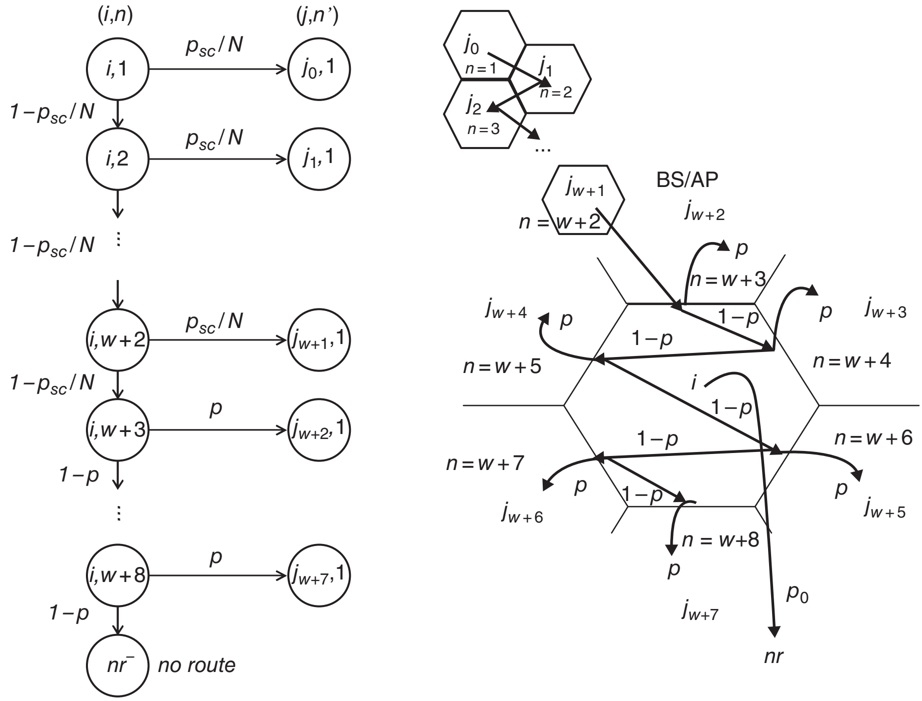


Figure 1.10.1 Transitions of the route discovery protocol from a given subcell i to its neighboring cells jk

protocol needs to find the route for a given user to the access point. We assume *s2LR* protocol as it is the one that considers more candidate users to establish the shortcuts. The extension to obtain the complexity for any other protocol is straightforward. The *s2LR* protocol considers that there may be the option to establish one of the $w+1$ possible shortcuts or otherwise, the protocol will be searching the relaying opportunities to the neighbors in the order indicated in Figure 1.2.1. The first time that the protocol finds such an opportunity it will progress to the next subcell. As a result it will spend different times in different subcells. To model this process, we need a separate state in the Markov model for each iteration in each subcell. Thus, the transition probabilities $p(i,j)$ defined in the previous section should be now modified into $p(i,n;j,n')$ as indicated in Figure 1.10.1, where $n' = 1$ indicates that the new transmission in the adjacent cell j will start from state 1 (shortest distance towards the BS/AP).

The rest of the analysis remains the same, and the average number of iterations Δ (complexity) to find the route can be obtained by (1.7.4) with $\boldsymbol{\Delta} = \boldsymbol{\tau}$.

1.11 Performance Evaluation

1.11.1 Average Path Length

Figure 1.11.1 shows the average path length in the redesigned network with shortcuts. The results are presented for $p = 0.5$ and different 2LR protocols with respect to the number of subcells N. The furthest reduction in l is obtained by s2LR and it is about 50% less than with 1L

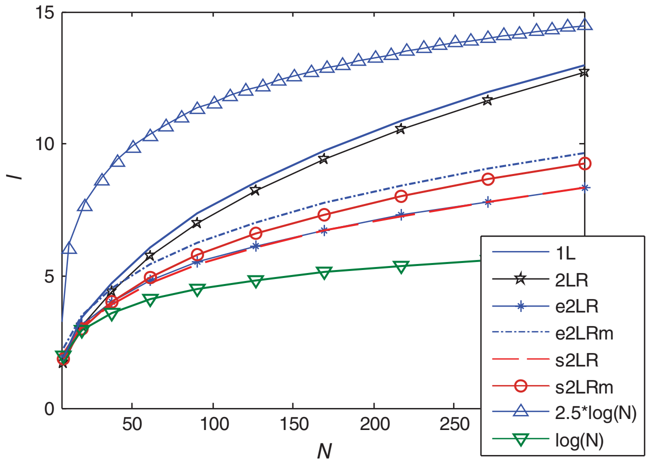


Figure 1.11.1 l given by (1.7.8) versus N for $p = 0.5$ and different 2 layer (2L) protocols

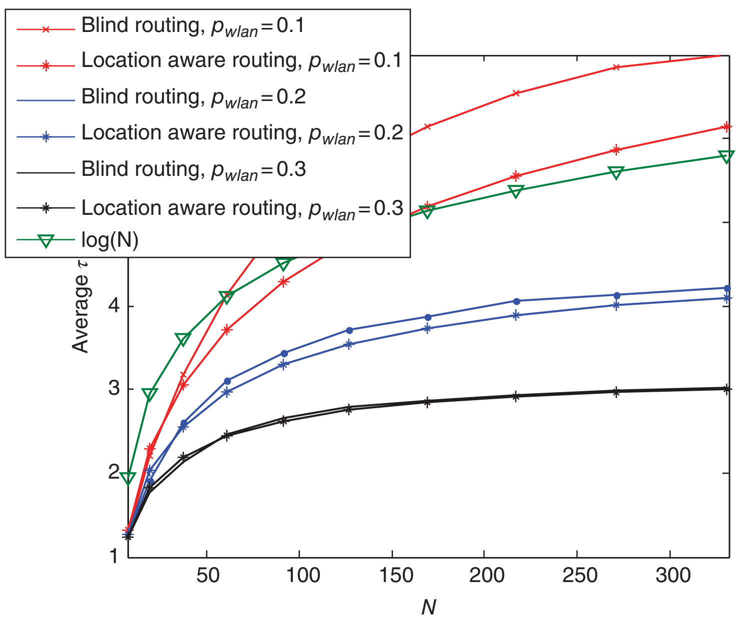


Figure 1.11.2 Average τ versus N for different p_{wlan}

protocol for $N = 300$. The modified versions of e2LR and s2LR, which select the candidate users to establish the shortcut so that backwards segments are avoided, provide similar results. As the number of candidate users in the modified protocols is reduced, l is slightly higher.

In Figure 1.11.2, it is assumed that with certain probability, p_{wlan}, there will be a WLAN AP. We assume $p = 0.5$. In this case, we can see that for small value of p_{wlan}, there is still a difference in τ between blind and location aware routing. If p_{wlan} increases, then the

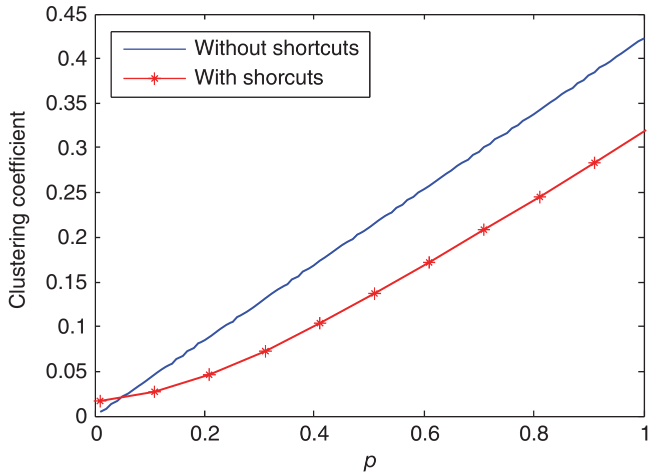


Figure 1.11.3 Clustering coefficient versus p and $N = 200$

performance of both protocols is almost the same as the probability that the adjacent user is an AP increases. Blind routing is intended for big networks where it is not possible to know the location of the APs.

1.11.2 Clustering

Figure 1.11.3 shows the clustering coefficient C versus p. We can see that, for the same value of p, C is higher for the network without shortcuts. In the figure one can see that the difference between the clustering coefficient for both networks increases with p. C for the network with shortcuts decreases about 20% when $p = 1$.

1.11.3 Node Robustness

In Figure 1.11.4, the node robustness to node and link failure, defined as $1 - p_{noroute}$, is shown versus $p > 0.5$ for different protocols. We can see that the resilience increases with p as the probability of finding the route increases. The highest resilience is obtained for the small world network by s2LR protocol. In this case, the node resilience increases by 3% compared to the network without shortcuts, and this difference remains for any value of $p > 0.5$.

1.11.4 Network Robustness

In Figure 1.11.5, the probability B that the user in subcell i reaches the BS/AP is shown for different protocols when $p = 0.5$. We can see that if 1L protocol is used, B significantly decreases for larger number of hops H. On the other hand, if the small world network is considered, the value of B increases about 10% in average and is more uniform through the different hops.

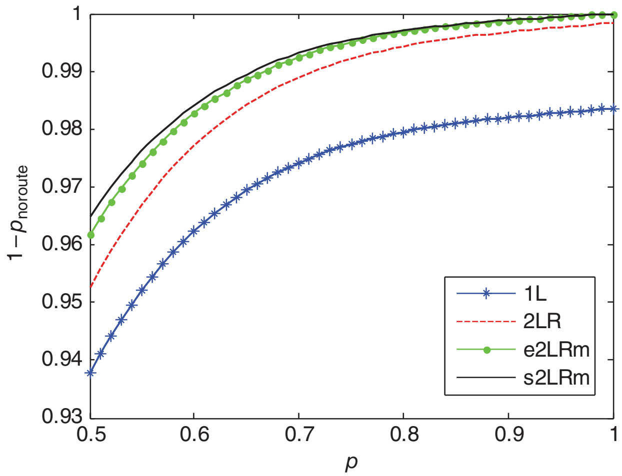


Figure 1.11.4 Average node robustness versus p for one layer and two layer protocols ($H = 4$)

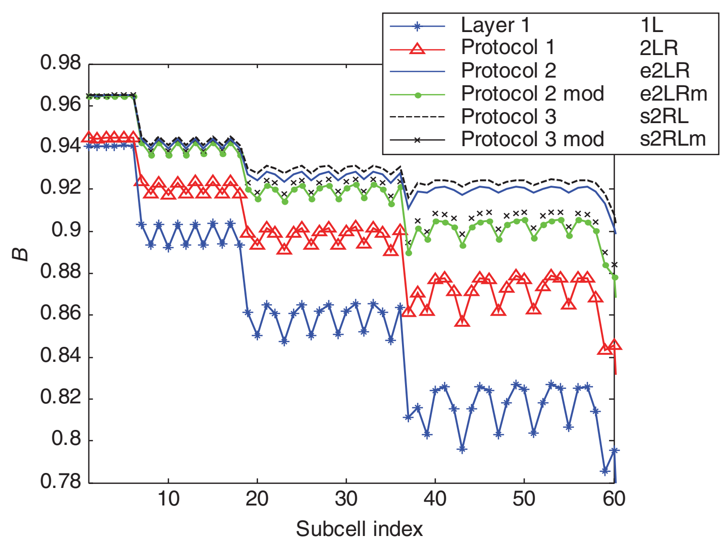


Figure 1.11.5 Route robustness B versus the subcell index for $p = 0.5$

The network robustness as defined by (31) with $l \rightarrow \tau$ is shown in Figure 1.11.6. As expected the robustness increases with k but goes to saturation, since for $p < 1$ there is still probability that there will be no relay available even if we can use an additional channel.

1.11.5 Power Consumption

In Figure 1.11.7, the power consumption is shown versus N for different routing protocols for the network with and without shortcuts. For a given p, the lowest power consumption is obtained when there are no shortcuts in the network (1L protocol). If protocol s2LR is used which considers the highest number of available shortcuts compared to 2LR and e2LR,

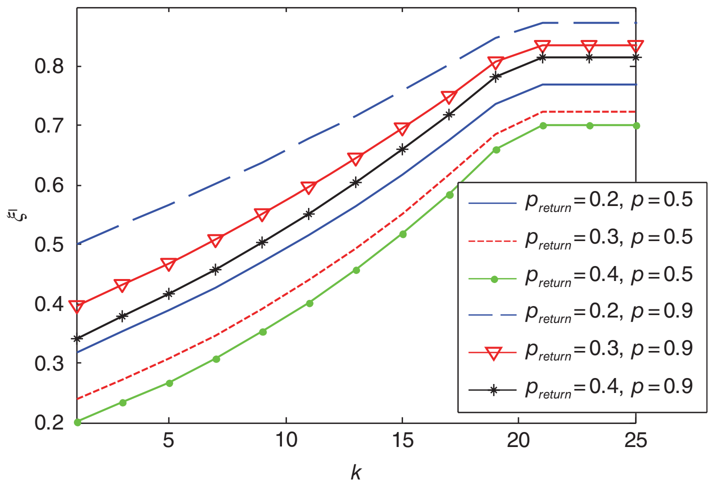


Figure 1.11.6 Network robustness ξ^l versus the number of frequency channels k_f where ξ is given by (1.3.1). $c=50$, $n_p=20$, $b_p=10$, $n_s=25$, b_s variable, $p=0.5$ and 0.9, $l=5.72$, $p=0.5$, 2LR protocol

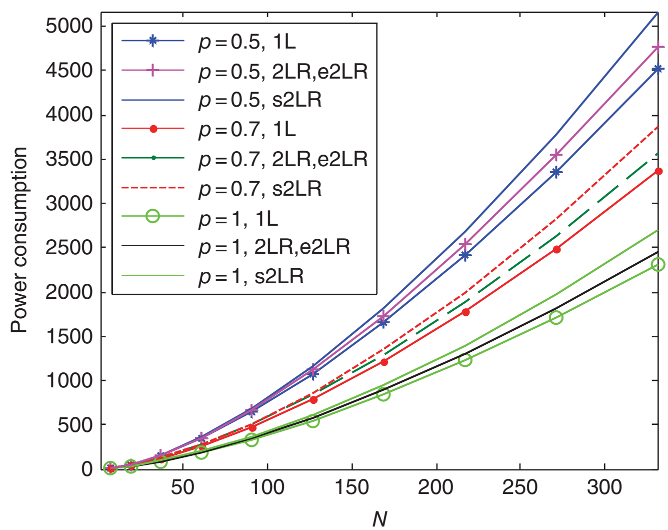


Figure 1.11.7 Power consumption versus N

the power consumption increases about 10%. We can also see that the power consumption increases for lower p, as the length of the route to reach the destination increases.

1.11.6 Protocol Complexity

In Figure 1.11.8, the complexity is compared for different protocols when $p=0.5$. The complexity significantly increases for protocols e2LR and s2LR. As the average path length for both protocols and their modified version are similar and their complexity is almost double compared to the modified version, it will be more efficient to implement the modified versions.

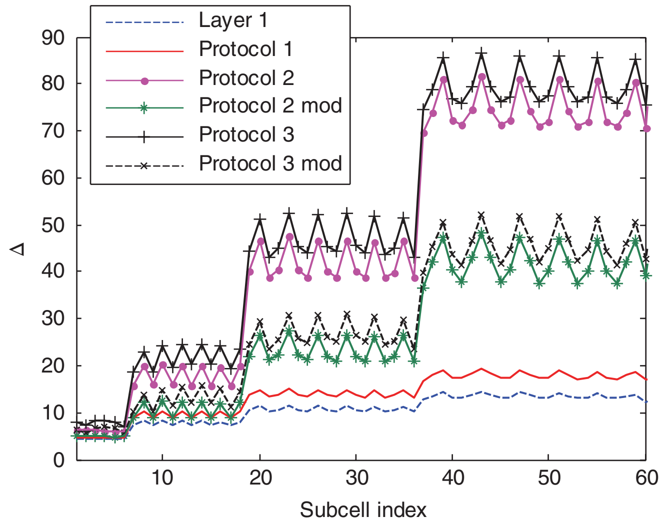


Figure 1.11.8 Complexity Δ versus the subcell index for $p = 0.5$

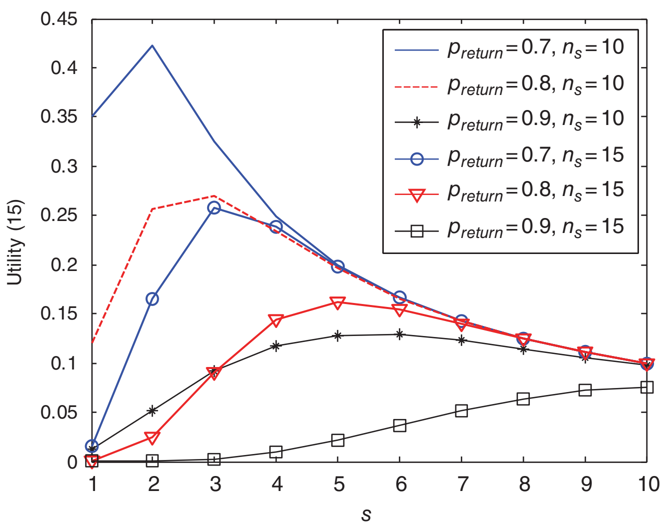


Figure 1.11.9 Utility versus s for different values of p_{return} and $k_f = 7$

Finally, Figures 1.11.9 and 1.11.10 represent the utility defined by (1.4.5a) versus s for different values of p_{return} and k_f, respectively. All these curves have an explicit maximum which indicates the possibility that for every state of the network an optimum value of s can be chosen. If the delays across the network are limited then, for a given s the needed number of backup channels k_f can be obtained.

In summary, in this chapter, we model link and node uncertainties as the result of a number of characteristics we envision to be present in future wireless networks. Those characteristics result from the heterogeneity of networks, operators, and applications where different agreements exist between users and operators. We show that the terminal availability probability

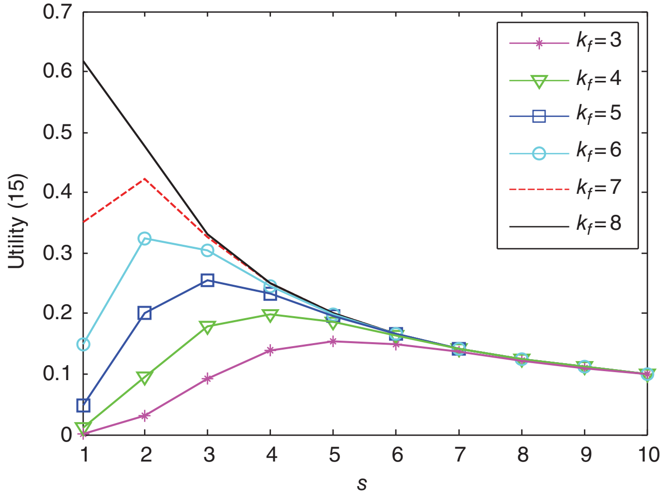


Figure 1.11.10 Utility defined by (1.4.5a) versus s for $p_{return} = 0.7$ and $n_s = 10$

p significantly affects the network performance. In particular, the clustering coefficient decrease about 50% for $p = 0.5$ compared to $p = 1$. This quantifies the importance of multi-operator cooperation in the network.

Then, we show how to redesign heterogeneous networks by introducing *small world* properties. A comprehensive analysis of such a network is provided by considering the average path length, clustering coefficient, node and link resilience, power consumption, and complexity. Illustrations show that, for the redesigned network, the average path length is proportional to $logN$ and stays within the range $logN < l < 2.5\ logN$. The resilience of the network improves for the small world network at the expense of 10% increase in power consumption, slight increase in the scheduling length and average increase in the complexity of factor 2 for a network of $H = 4$ hops.

A number of routing protocols are presented based on the awareness of the existence of different APs in the network. It was shown that when $p_{wlan} > 0.2$, blind routing and location aware routing provide very similar results for large networks since there is high probability that the traffic will be offloaded rather than forwarded to the BS.

It was also demonstrated how the optimal allocation of the backup channels can be performed for each state of the network. The analysis provides an explicit relation between the waiting time s to issue a request for a backup channel and the number of available back up channels k_f. All optimization curves have an explicit maximum which indicates the possibility that for every state of the network an optimum value of s can be chosen. If the delays across the network are limited then, for the given s (delay), the necessary values of the number of backup channels k_f can be obtained.

1.12 Book Layout

In the previous sections a generic model of the future wireless network was presented that integrates a number of different components and tools needed for their analysis. In the rest of the book these components and tools are elaborated in more detail within separate chapters. In this section we briefly summarize the content of these chapters in order to justify the motivations for introducing this material in the book and to relate the chapters to the generic model of the network.

1.12.1 Chapter 1: Introduction: Generalized Model of Advanced Wireless Networks

The chapter presents the generalized networks model anticipated for 5G technology and discuss its components and relevant issues, mainly: node percolation, link percolation – cognitive links, network connectivity, wireless network design with small world properties, frequency channels backup, generalized network model routing protocols over *s-lattice* network, network performance, node, route, topology and network robustness, power consumption, protocol complexity, and performance evaluation.

1.12.2 Chapter 2: Adaptive Network Layer

The chapter on *adaptive network layer* covers: graphs and routing protocols, elements of graph theory, routing with topology aggregation, network and aggregation models.

1.12.3 Chapter 3: Mobility Management

It is anticipated that, in the generic model of the network, the cellular network will be still responsible for the mobility management. For this reason the chapter reviews the mobility management techniques and focuses on cellular systems with prioritized handoff, cell residing time distribution and mobility prediction in pico and micro cellular networks.

1.12.4 Chapter 4: Ad Hoc Networks

As indicated in the generic network model description, some segments of the future networks will be organized on ad hoc principles. For this reason this chapter includes discussion on: routing protocols in ad hoc networks, hybrid routing protocol, scalable routing strategies multipath routing, clustering protocols, cashing schemes for routing and distributed quality-of-service (QoS) routing.

1.12.5 Chapter 5: Sensor Networks

The most of the network protocols will be context aware and data about the network and environment will be collected by sensor networks. For this reason this chapter will include discussions on: sensor networks parameters, sensor networks architecture, mobile sensor networks deployment, directed diffusion, aggregation in wireless sensor networks, boundary estimation, optimal transmission radius in sensor networks, data funneling, and equivalent transport control protocol in sensor networks.

1.12.6 Chapter 6: Security

Security remains an important segment of future wireless networks and for that reason in this chapter we discuss the following topics: authentication, security architecture, key management, security management in *ad hoc* and sensor networks.

1.12.7 Chapter 7: Networks Economy

As indicated in the generic model of the network the significant changes in business models in the field of communications networks should take place already in the very first versions (releases) of 5G/6G technology. This will be visible on both macro (operator) level in spectrum sharing as well as on micro (terminal) level for reimbursing the terminal relaying other users' traffic. For this reason we discuss some basic principles in this field with the focus on: Pricing of services, auctions, bidding for QoS, bandwidth auction, investment incentives, sequential spectrum auctions, and double auction mechanism for secondary spectrum markets.

1.12.8 Chapter 8: Multi-Hop Cellular Networks

As it was indicated earlier, in addition to the massive traffic offloading options the generic network model also includes the option of multi-hop transmission which in a way represents further extension of the existing m2m communications within the macro cell. To elaborate this technology further we discuss in this chapter the following topics: relaying, nanoscale network model, scale free networks, multi-hop multi-operator multi-technology networks, network defading, multi-radio, adaptive relaying in LTE-advanced networks, spectrum auctions for multi-hop secondary networks.

1.12.9 Chapter 9: Cognitive Networks

The generic model of the network includes the options where some of the links are with the status of secondary user. For this reason in this chapter we discuss in more details general principles of cognitive networks including: cognitive small cell networks, power allocation games, data traffic, broadcast protocols, opportunistic spectrum access, spectrum trading, stability analysis, dynamic profit maximization of network operator.

1.12.10 Chapter 10: Stochastic Geometry

Stochastic geometry has become one of the main tools for the analysis of the interference in dense wireless networks. For this reason we present some of the problems in this field that can be modeled and analyzed in this way. The focus in this chapter is on: Stochastic geometry modeling of wireless networks, signal to interference plus noise ratio (SINR) model, point processes, performance metrics, dominant interferers by region bounds or nearest n interferers, approximation of the pdf of the aggregate interference.

1.12.11 Chapter 11: Heterogeneous Networks

The generic model represents the network which is very much heterogeneous. For this reason in this chapter we discuss basics of heterogeneous networks and summarize the experience and results published so far. The material includes discussion on: WLAN macro/femto/small cells, macro to femto network deployment and management, self-organizing femtocell networks, economics of femtocell service provision, femtocells as additional internet gateways, indoor cooperative small cells over Ethernet, cognitive small cell networks, self-organization in small cell networks, adaptive small-cell architecture.

1.12.12 Chapter 12: Access Point Selection

As indicated in the generic model of future wireless networks terminals will have a variety of different access points to choose to connect to. In this chapter we discuss main criteria of how to select the best connection in such environment with emphasis on: Network selection and resource allocation, joint access point selection and power allocation, averaged iterative-water filling algorithm, a non-cooperative game formulation, stability and fairness of AP selection games, a unified QoS-inspired load optimization, a learning-based network selection method in heterogeneous wireless systems.

1.12.13 Chapter 13: Self-Organizing Networks

Self-organization of dense networks becomes important for improving network efficiency in using the available resources. In this chapter we cover the following problems: Conceptual framework for self-organizing networks (SONs), optimization over the user-association policy, introducing load constraints, handover parameter optimization.

1.12.14 Chapter 14: Complex Networks

It is expected that the design tools of complex networks, well established in the fields like Internet, social networks, citations networks, or web networks will be used more and more in the design of future wireless dense networks. The first hints were already given in the description of the generic model of the network. In this chapter we briefly discuss some of the main topics in this field like: Types of networks, social networks, the small-world effect, degree distributions, scale-free networks, network resilience, random graphs, average path length, models of network growth, Price's model, the model of Barabási and Albert, processes taking place on networks, percolation theory and network resilience and epidemiological processes.

1.12.15 Chapter 15: Massive MIMO

Although massive MIMO is a physical layer technology, in this book it is discussed as an option to increase network capacity by spatial reuse of the channels. For this reason in this chapter we include material on: massive MIMO for next generation wireless systems with the focus on precoding algorithms, imperfections, channel measurements and modeling, detection algorithms, resource allocation, performance analysis, robust design, and coordinated point transmission.

1.12.16 Chapter 16: Network Optimization Theory

There is a varity of network optimization tools that enable best network parameters selection in accordance with some objective function, often referred to as utility function. In this chapter we provide a brief overview of these tools. More specifically we cover topics as: layering as optimization decomposition, cross-layer optimization, and optimization problem decomposition methods.

1.12.17 Chapter 17: Network Information Theory

Network optimization theory provides tools to analyze maximum achievable rates (capacity) in the network. Most of the time the performance measure is the network transport capacity. In this chapter we provide a brief overview of these tools. More specifically we cover topics as capacity of ad hoc networks, information theory, and network architectures.

1.12.18 Chapter 18: Network Stability

For delay tolerant networks (DTNs) messages can be temporally stored in a queue in a node before being forwarded to the next node on the route. For such a network it is important to control the congestion and make sure that all queues in the network do not exceed the predetermined value which is referred to as the network stability. In this chapter we briefly summarize the tools for the analysis of the network stability by focusing on: time varying network with queuing, network delay, Lyapunov drift and network stability, Lagrangian decomposition of multi-comodity flow optimization problem, flow optimization in heterogeneous networks, dynamic resource allocation in computing clouds.

1.12.19 Chapter 19: Multi-Operator Spectrum Sharing

As indicated in the generic model of the network the spectrum sharing principle might be more attractive and efficient then classical cognitive network approach. For this reason we cover in this chapter basic principles of spectrum sharing and mutual business relations between multiple operators. Mainly we cover: possible business relations in spectrum sharing, game theory based models, primary/secondary network operator contracts, channel availability, channel corruption, spectra borrowing/leasing, pricing models, modeling user dissatisfaction, multi-operator congestion control in the network.

1.12.20 Chapter 20: Large Scale Networks and Mean Field Theory

We discuss Mean Field Theory (MFT) for Large Heterogeneous Cellular Networks, Macro-BS optimization problem, Mean-Field Game Among Femto-BSs, Interference Average Estimation, Large Scale Network Model Compression, Mean-Field Analysis, Mean Filed Theory

Model of Large Scale DTN Network, Mean Field Modeling of Adaptive Infection Recovery in Multicast DTN Networks, Background Technology, System Model, Recovery Schemes for Multicast DTN, System Performance, Extensions of the Model and Implementation Issues, Illustrations, MFT for Scale Free Random Networks, The Scale-Free Model by Barabasi, Mean Field Network Model, Incomplete BA Network Models, Spectrum Sharing and MFT, Optimal Wireless Service Provider (WSP) Selection Strategy using MFT, WSP Selection Strategy for Finite Number of Terminals, Iterative Algorithm to Solve Systems of Nonlinear ODEs (DiNSE- algorithm), Infection Rate of Destinations for DNCM and Infection Rate for Basic Epidemic Routing.

1.12.21 Chapter 21: mmWave 3D Networks

mmWave technology has become interesting for 5G/6G systems at least for the reason that it enables significant additional spectra and more efficient beamforming, which now becomes feasible for implementations even in portable terminals. For this reason in this chapter we summarize some basic issues regarding this field with the emphasis on: *mmWave* technology in subcellular architecture, limitations of mmWave technology, network model, network performance, performance of dense mmWave networks, microeconomics of dynamic mmWave networks, dynamic small cell networks, DSC network model, and DSC network performance.

1.12.22 Chapter 22: Cloud Computing in Wireless Network

Cloud computing has become a priority in the research community since it provides new concepts, more efficient and more powerful, when it comes to the organization and management of big data. This has generated an equivalent problem in communications and networking. For this reason in this chapter we discuss: technology background, system models, system optimization, dynamic control algorithm, achievable rates, and network stabilizing control policies.

1.12.23 Chapter 23: Wireless Networks and Matching Theory

In this chapter, we discuss the use of matching theory, for resource management in wireless networks. The key solution concepts and algorithmic implementations of this framework are presented. *Matching theory* can overcome some limitations of game theory and optimization discussed in the previous chapters of the book. It provides mathematically tractable solutions for the combinatorial problem of matching players in two distinct sets, depending on the individual information and preference of each player. Within the chapter we discuss Matching Markets, Distributed Stable Matching in Multiple Operator Cellular Network with Traffic Offloading, Many to Many Matching Games for Cashing in Wireless Networks and Many to One Matching with Externalities in Cellular Networks with Traffic Offloading.

1.12.24 Chapter 24: Dynamic Wireless Network Infrastructure

The network infrastructure require significant investments and for this reason a certain attention has been attracted by the latest work on the new paradigms in this field. In general these paradigms are providing solution where the network infrastructure of a particular operator can be temporally expanded or compressed without need for additional investment. We discuss in this chapter two options for this solution: (i) *network infrastructure sharing* and (ii) *user provided connectivity*. In addition in this chapter we discuss Network Virtualization, Software Defined Networks (SDNs), and SDN Network Security.

Appendix A.1

In multi-hop transmission, $u(h, \theta)$ relays the information to any of its adjacent users $u(h', \theta')$. The location of any adjacent relay is calculated in vector form as

$$\begin{aligned}\vec{u}(h',\theta') &= h' \cdot d_r \cdot e^{j\theta'}\\ &= \vec{u}(h,\theta) + \eta \cdot d_r \cdot e^{j\theta_n^{(\eta)}},\ \eta = 1,\ldots,k,\ n = 1,\ldots,n_h\end{aligned}$$

which depends on the lattice range k, the relaying distance d_r, and the location of the transmitter $u(h', \theta')$.

For the lattice of range $k=1$, the set of angles $\Theta^{(1)} = \{\theta_n^{(1)}\}$ is $\theta_1^{(1)} = 30°$; $\theta_n^{(1)} = \theta_{n-1}^{(1)} + \hat{\theta}_1 = \theta_{n-1}^{(1)} + 60°$, $n = 2,\ldots,n_h$, where $\theta_1^{(1)}$ is the first angle of the set. As we can see from Figure 1.3.3, the first adjacent user (vertex) in $k=1$ is located 30° with respect to $u(h, \theta)$, and the separation between users (network vertices) is $\hat{\theta}_1 = 60°/1 = 60°$. The set of angles $\Theta^{(k)}$ for the lattice with range $k = 2,\ldots, H$ is calculated following the same reasoning as

$$\begin{aligned}&\theta_1^{(2)} = 0°;\ \theta_n^{(2)} = \theta_{n-1}^{(2)} + 30°,\ n = 2,\ldots,n_h\\ &\theta_1^{(3)} = 10°;\ \theta_n^{(3)} = \theta_{n-1}^{(3)} + 20°\\ &\theta_1^{(4)} = 0°;\ \theta_n^{(4)} = \theta_{n-1}^{(4)} + 15°\\ &\theta_1^{(5)} = 6°;\ \theta_n^{(5)} = \theta_{n-1}^{(5)} + 12°,\ \ldots\\ &\theta_n^{(k)} = \theta_{n-1}^{(k)} + \hat{\theta}_h = \theta_{n-1}^{(k)} + 60°/k,\quad n = 2,\ldots,n_h\\ &\theta_1^{(k)} = \begin{cases} 30°/k, & \textit{if } k = 2p+1,\ p = 0,1,\ldots,\left\lceil \dfrac{H-1}{2} \right\rceil \\ 0°, & \textit{otherwise}\end{cases}\end{aligned}$$

The set of adjacent relays (nodes) is given by $U = \cup_\eta \left\{\vec{u}\left(\eta, \Theta^{(\eta)}\right)\right\}$, $\eta = 1,\ldots,k$. Once the location of the users (vertices) is known in terms of h and θ, it is straightforward to obtain its index within the spiral.

References

[1] Zorzi, M., Gluhak, A. and Lange, S. (2010) From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view. *IEEE Wireless Communications*, **17** (6), 44–51.

[2] Pabst, R., Walke, B.H., Schultz, D.C. *et al.* (2004) Relay-based deployment concepts for wireless and mobile broadband radio. *IEEE Communications Magazine*, **42** (9), 80–89.

[3] Ganesan, G., Li, Y., Bing, B. and Li, S. (2008) Spatiotemporal sensing in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, **26** (1), 5–12.

[4] Deb, S., Mhatre, V., and Ramaiyan, V. (2008) *WiMAX Relay Networks: Opportunistic Scheduling to Exploit Multiuser Diversity and Frequency Selectivity*. Proceedings of the ACM MobiCom'08, September 2008, pp: 163–174.

[5] Bejerano, Y., Han, S.-J. and Li, L. (2007) Fairness and load balancing in wireless LANs using association control. *IEEE/ACM Transactions on Networking*, **15** (3), 560–573.

[6] Lorenzo, B. and Glisic, S. (2012) Context-aware nanoscale modeling of multicast multihop cellular networks. *IEEE/ACM Transactions on Networking*, **21** (2), 359–372.

[7] Lorenzo, B. and Glisic, S. (2010) *Joint Optimization of Cooperative Diversity and Spatial Reuse in Multi-hop Hybrid Cellular/Ad Hoc Networks*. Proceedings of the IEEE MILCOM 2010, November 2010, pp. 499–506.

[8] Patzold, M., Hogstad, B. and Youssef, N. (2008) Modeling, analysis, and simulation of MIMO mobile-to-mobile fading channels. *IEEE Transactions on Wireless Communications*, **7** (2), 510–520.

[9] Albert, R. and Barabasi, A. (2002) Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74** (1), 47–97.

[10] Watts, D.J. and Strogatz, S.H. (1998) Collective dynamics of 'small-world' networks. *Nature*, **393** (6684), 440–442.

[11] Newman, M.E.J. and Watts, D.J. (1999) Renormalization group analysis of the small-world network model. *Physics Letters A*, **263** (4), 341–346.

[12] Verma, C.K., Tamma, B.R., Manoj, B.S. and Rao, R. (2011) A realistic small-world model for wireless mesh networks. *IEEE Communications Letters*, **15** (4), 455–457.

[13] Guidoni, D., Mini, R., and Loureiro, A. (2008) *On the Design of Heterogeneous Sensor Networks Based on Small World Concepts*. 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 08), 2008, Vancouver, Canada, p. 309314.

[14] Brust, M. and Rothkugel, S. (2007) *Small-Worlds: Strong Clustering in Wireless Networks*. First International Workshop on Localized Algorithms and Protocols for Wireless Sensor Networks, LOCALGOS, USA, 2007.

[15] Agarwal, R., Banerjee, A., Gauthier, V. *et al.* (2011) *Self-Organization of Nodes using Bio-Inspired Techniques for Achieving Small World Properties*. 2011 GLOBECOM Workshops, pp. 89–94.

[16] Afifi, N. and Chung, K.-S. (2008) *Small World Wireless Mesh Networks*. International Conference on Innovations in Information Technology(IIT), Al Ain, December 2008, pp. 500–504.

[17] Sharma, G. and Mazumdar, R. (2005) *Hybrid Sensor Networks: A Small World*. Proceedings of the 2005 ACM Mobihoc, pp. 366–377.

[18] Banerjee, A., Agarwal, R., Gauthier, V. *et al.* (2012) A self-organization framework for wireless ad hoc networks as small worlds. *IEEE Transactions on Vehicular Technology*, **61** (6), 2659–2673.

[19] Bo, L., Muqing, W., Jingrong, W., and Dongyang, W. (2013) *Small Worlds in Multi-channel Wireless Networks: An Analytical Approach*. Proceedings of the IEEE ICC 2013, pp. 1527–1531.

[20] Chen, P.-Y. and Chen, K.-C. (2010) *Information Epidemics in Complex Networks with Opportunistic Links and Dynamic Topology*. GLOBECOM – IEEE Global Telecommunications Conference.

[21] Glisic, S., Lorenzo, B., Kovacevic, I., and Fang, Y. (2013) *Modeling Dynamics of Complex Wireless Networks*. Proceedings of HPCS 2013, July 1–5, 2013, Helsinki, Finland.

[22] Watts, D.J. (1999) *Small Worlds*, Princeton University Press, Princeton, NJ.

[23] Newman, M.E.J. (2003) The structure and function of complex networks. *SIAM Review*, **45**, 167–256.

[24] Bolch, G., Greiner, S., de Meer, H. and Trivedi, K.S. (2006) *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, 2nd edn, John Wiley & Sons, Inc., Hoboken, NJ.

2

Adaptive Network Layer

2.1 Graphs and Routing Protocols

The most important function of the network layer is routing. A tool used in the design and analysis of routing protocols is graph theory. Networks can be represented by graphs where mobile nodes are vertices and communication links are edges. Routing protocols often use shortest path algorithms. In this section we provide a simple review of the most important principles in the field which provides a background to study the routing algorithms.

Elementary Concepts: A graph $G(V,E)$ is two sets of objects, vertices (or nodes), set V and edges, set E. A graph is represented with dots or circles (vertices) interconnected by lines (edges). The magnitude of graph G is characterized by the number of vertices $|V|$ (called the order of G) and the number of edges $|E|$, size of G. The running time of algorithms is measured in terms of order and size.

Directed Graph: An edge $e \in E$ of a directed graph is represented as an ordered pair (u,v), where $u, v \in V$. Here u is the initial vertex and v is the terminal vertex. Also assume here that $u \neq v$. An example with

$$V = \{1, 2, 3, 4, 5, 6\}, |V| = 6$$

$$E = \{(1,2), (2,3), (2,4), (4,1), (4,2), (4,5), (4,6)\}, |E| = 7$$

is shown in Figure 2.1.1.

Advanced Wireless Networks: Technology and Business Models, Third Edition. Savo Glisic.

Companion website: www.wiley.com/go/glisic0916

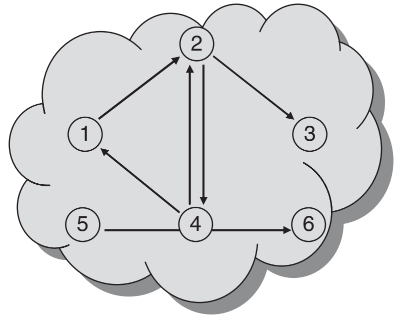


Figure 2.1.1 Directed graph

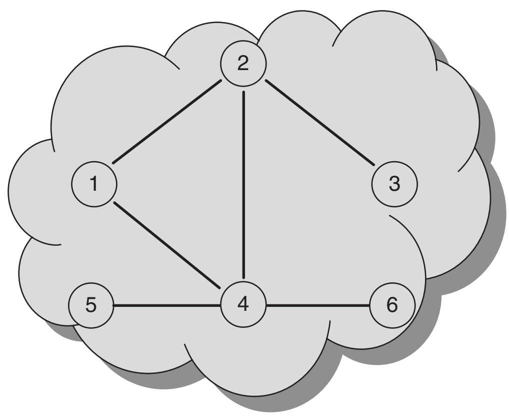


Figure 2.1.2 Undirected graph

Undirected Graph: An edge $e \in E$ of an undirected graph is represented as an unordered pair $(u,v) = (v,u)$, where u, $v \in V$. Also assume that $u \neq v$. An example with

$$V = \{\,1, 2, 3, 4, 5, 6\}, \; |V| = 6$$

$$E = \{(1,2), (2,3), (2,4), (4,1)\}, (4,5)\,(4,6)\; |E| = 6$$

is shown in Figure 2.1.2.

Degree of a Vertex: Degree of a vertex in an undirected graph is the number of edges incident on it. In a directed graph, the *out degree* of a vertex is the number of edges leaving it and the *in degree* is the number of edges entering it. In Figure 2.1.2 the *degree* of vertex 2 is 3. In Figure 2.1.1 the *in degree* of vertex 2 is 2 and the *in degree* of vertex 4 is 1.

Weighted Graph: In a weighted graph each edge has an associated weight, usually given by a weight function $w\colon E \rightarrow R$. Weighted graphs from Figures 2.1.1 and 2.1.2 are shown in Figure 2.1.3. In the analysis of the routing problems, these weights represent the cost of using the link. Most of the time this cost would be delay that a packet would experience if using that link.

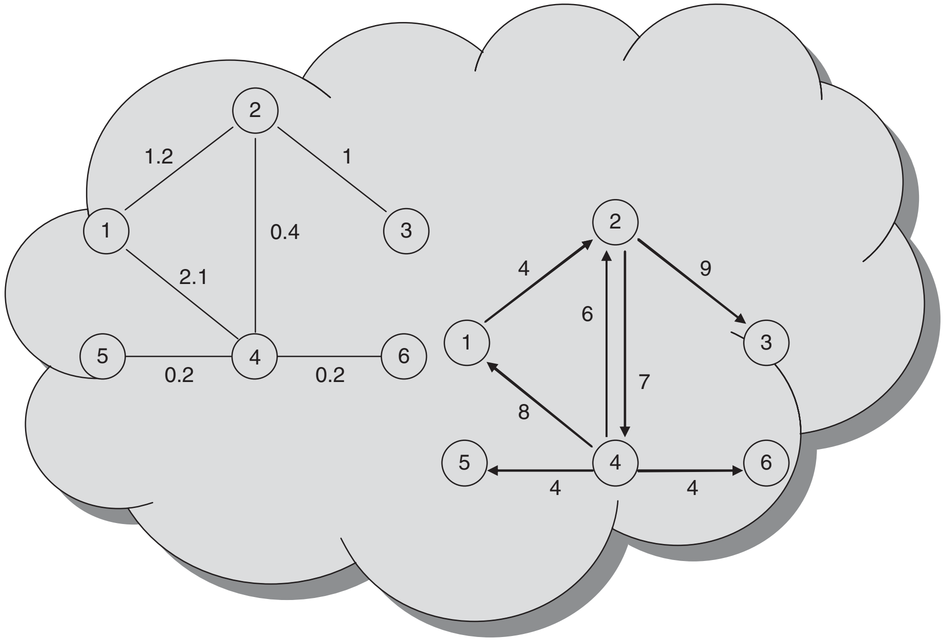


Figure 2.1.3 Weighted graphs

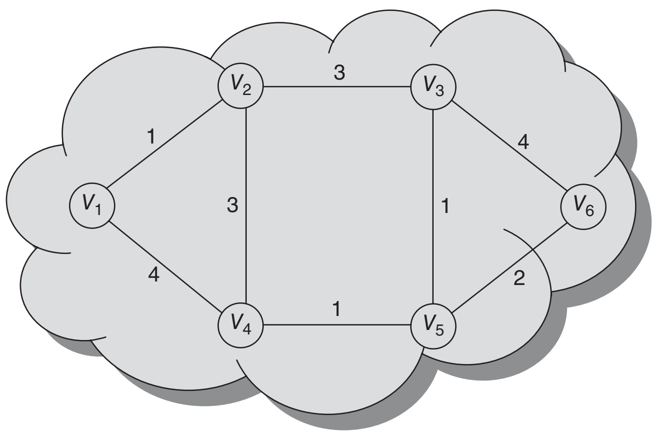


Figure 2.1.4 Illustration of a walk

Walks and Paths: A walk is a sequence of nodes ($v1$, $v2$,…, vL) such that {($v1$, $v2$), ($v2$, $v3$),…, (vL−1, vL)} $\subseteq E$, e.g. ($V2$, $V3$,$V6$, $V5$,$V3$) in Figure 2.1.4.

A *simple path* is a walk with no repeated nodes, e.g. ($V1$, $V4$,$V5$, $V6$,$V3$).

A *cycle* is a walk ($v1$, $v2$,…, vL) where $v1 = vL$ with no other nodes repeated and $L > 3$, e.g. ($V1$, $V2$,$V3$, $V5$,$V4$,$V1$). A graph is called *cyclic* if it contains a cycle; otherwise it is called *acyclic*.

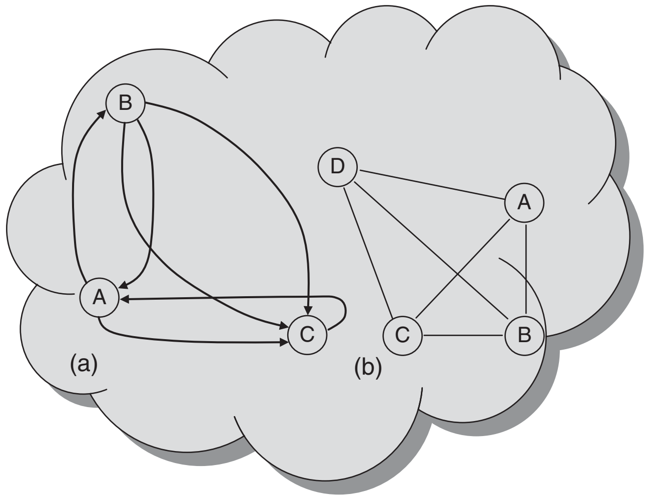


Figure 2.1.5 Complete graphs: (a) V nodes and V(V – 1) edges: three nodes and 3 × 2 edges. (b) V nodes and V(V – 1)/2 edges: four nodes and 4 × 3/2 edges

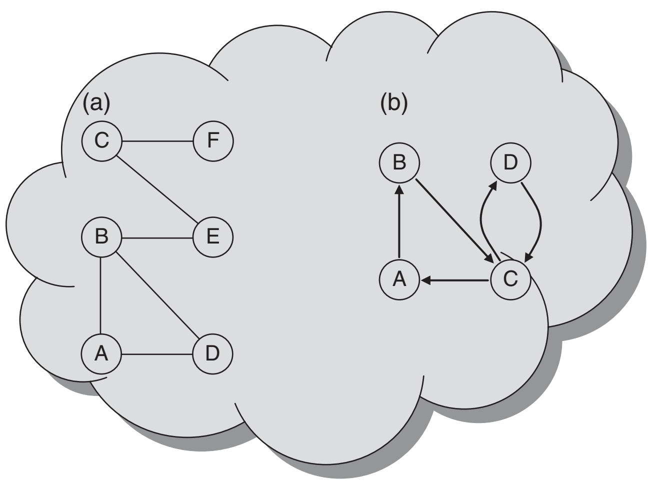


Figure 2.1.6 Connected graphs

A *complete graph* is an undirected/directed graph in which every pair of vertices is *adjacent.* If (u, v) is an edge in a graph G, we say that vertex v is *adjacent* to vertex u.

Connected Graphs: An undirected graph is connected if you can get from any node to any other by following a sequence of edges or any two nodes are connected by a path, as shown in Figures 2.1.5 and 2.1.6. A directed graph is *strongly connected* if there is a directed path from any node to any other node. A graph is *sparse* if $|E| \approx |V|$. A graph is *dense* if $|E| \approx |V|^2$.

A *bipartite graph* is an undirected graph $G = (V, E)$ in which V can be partitioned into two sets, $V1$ and $V2$, such that $(u,v) \in E$ implies either $u \in V1$ and $v \in V2$ or $v \in V1$ and $u \in V2$, see Figure 2.1.7.

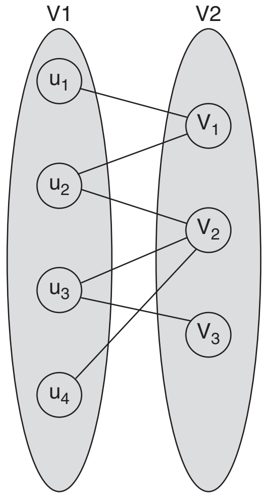


Figure 2.1.7 Bipartite graph

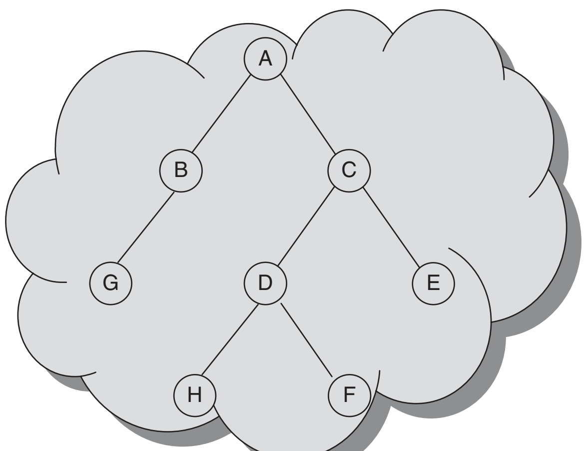


Figure 2.1.8 Tree

Trees: Let G = (V, E) be an undirected graph. The following statements are equivalent:

1. *G* is a tree.
2. Any two vertices in *G* are connected by a unique simple path.
3. *G* is connected, but if any edge is removed from *E*, the resulting graph is disconnected.
4. *G* is connected, and $|E| = |V| - 1$.
5. *G* is acyclic, and $|E| = |V| - 1$.
6. *G* is acyclic, but if any edge is added to *E*, the resulting graph contains a cycle
7. For illustration, see Figure 2.1.8.

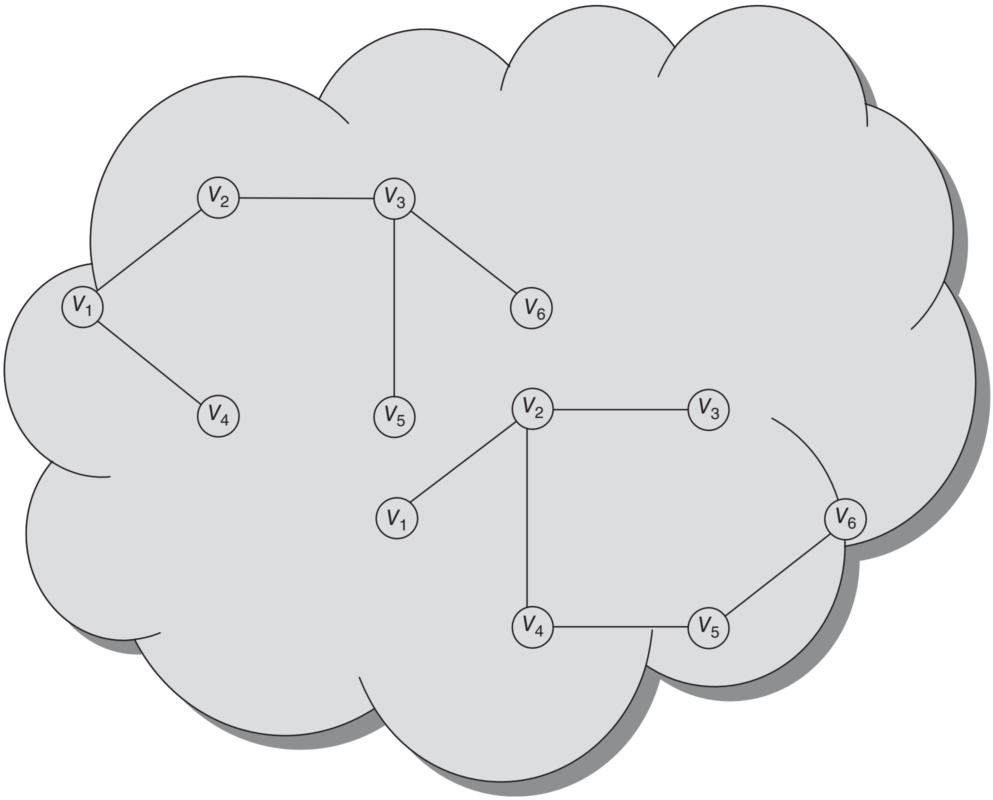


Figure 2.1.9 Spanning trees

Spanning Tree: A tree (T) is said to span $G = (V, E)$ if $T = (V, E')$ and $E' \subseteq E$. For the graph shown in Figure 2.1.4, two possible spanning trees are shown in Figure 2.1.9.

Given connected graph G with real-valued edge weights ce, a Minimum Spanning Tree (MST) is a spanning tree of G whose sum of edge weights is minimized (Figure 2.1.10).

MST Computation:

Prim's Algorithm:

Select an arbitrary node as the initial tree (T).

Augment T in an iterative fashion by adding the outgoing edge (u,v), (i.e., $u \in T$ and $v \in G{-}T$) with minimum cost (i.e., weight). The algorithm stops after $|V| - 1$ iterations. Computational complexity = O (|V|2). An illustration of the algorithm is given in Figure 2.1.11.

Kruskal's Algorithm:

Select the edge $e \in E$ of minimum weight $\rightarrow E' = \{e\}$.

Continue to add the edge $e \in E - E'$ of minimum weight that when added to E', does not form a cycle. Computational complexity = O ($|E| \times \log|E|$). An illustration of the algorithm is given in Figure 2.1.12.

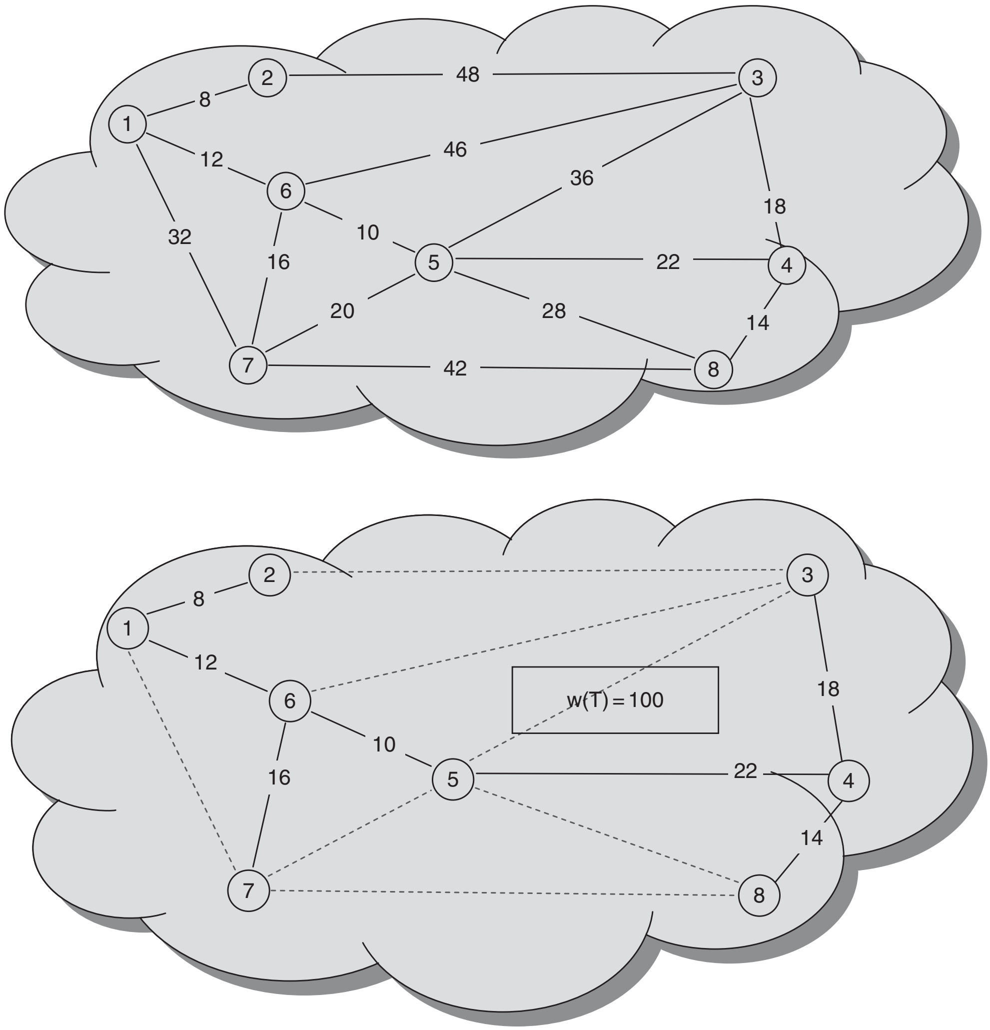


Figure 2.1.10 Minimum spanning tree

Distributed Algorithms: For these algorithms each node does not need complete knowledge of the topology. The MST is created in a distributed manner. The algorithm starts with one or more fragments consisting of single nodes Each fragment selects its minimum weight outgoing edge and using control messaging fragments coordinate to merge with a neighboring fragment over its minimum weight outgoing edge. The algorithm can produce a MST in $O(|V| \times |V|)$ time provided that the edge weights are unique. If these weights are not unique the algorithm still works by using the nodes IDs to break ties between edges with equal weight. The algorithm requires an $O(|V| \times \log|V|) + |E|$ message overhead. An illustration of the distributed algorithm is given in Figure 2.1.13.

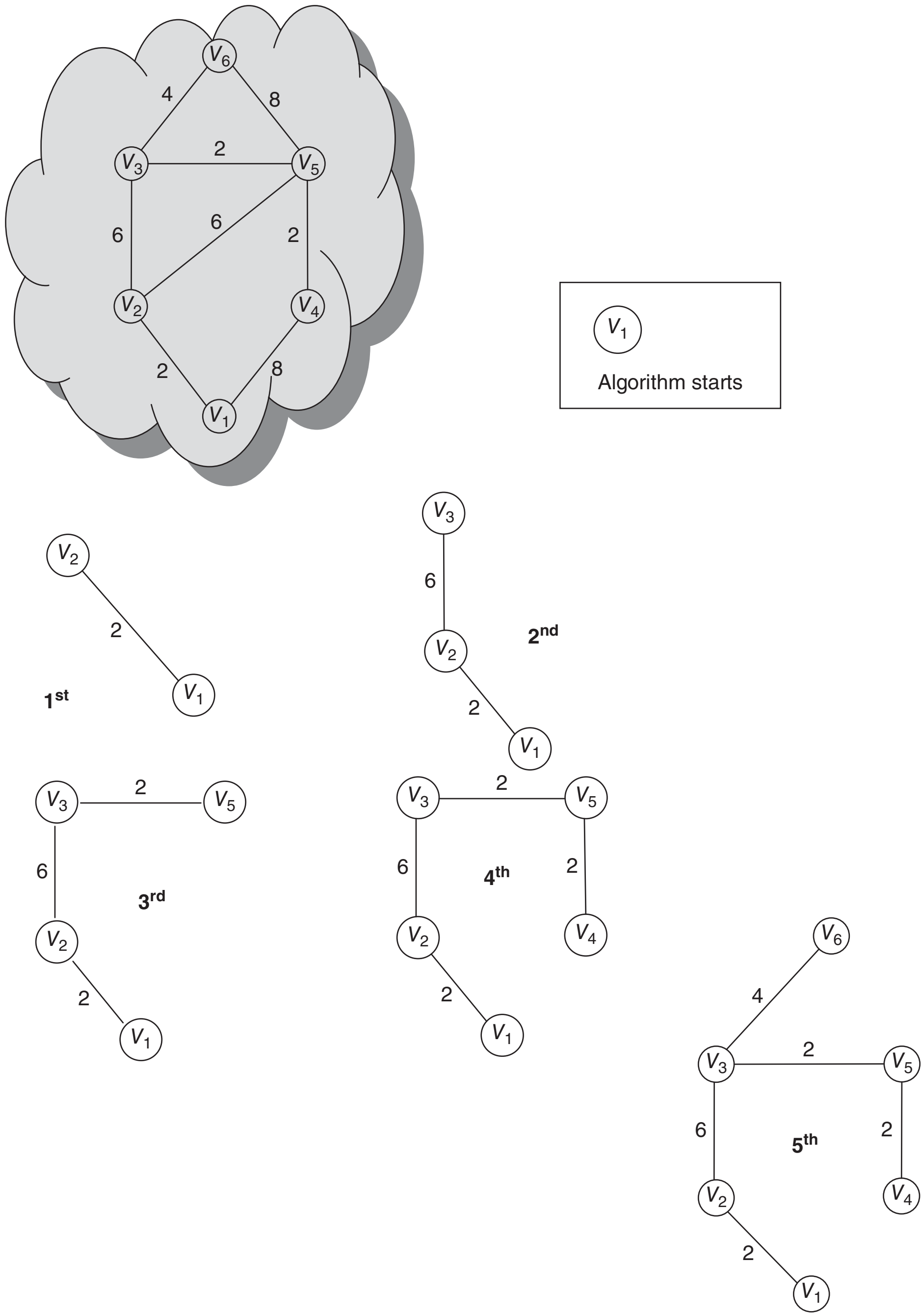


Figure 2.1.11 MST solution via Prim's algorithm

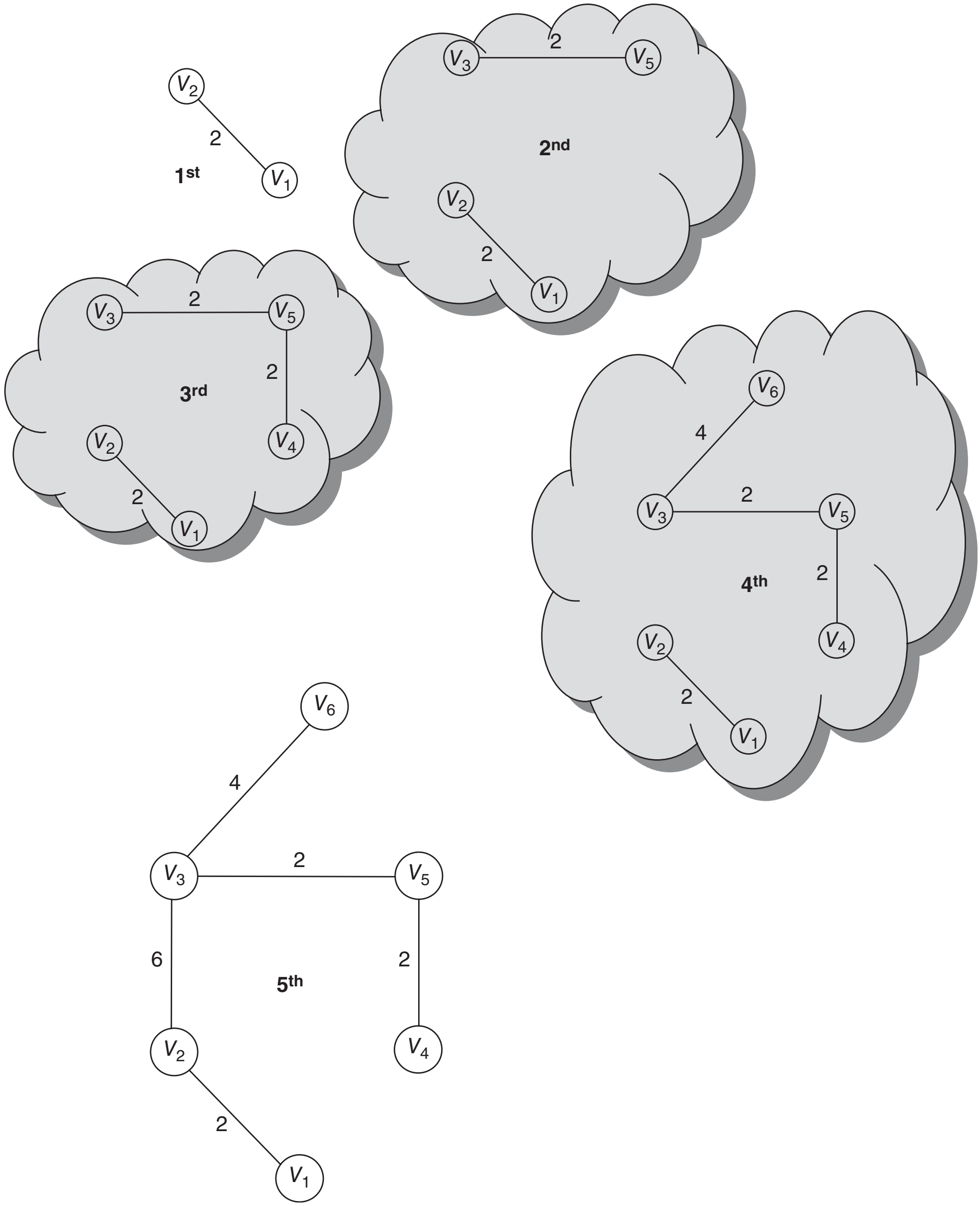


Figure 2.1.12 MST solution via Kruskal's algorithm

Shortest Path Spanning Tree (SPST), *T*, is a spanning tree rooted at a particular node such that the $|V| - 1$ minimum weight paths from that node to each of the other network nodes is contained in *T*. An example of the SPST is shown in Figure 2.1.14. Note that the SPST is not the same as the MST.

SPST trees are used for unicast (one to one) and multicast (one to several) routing.

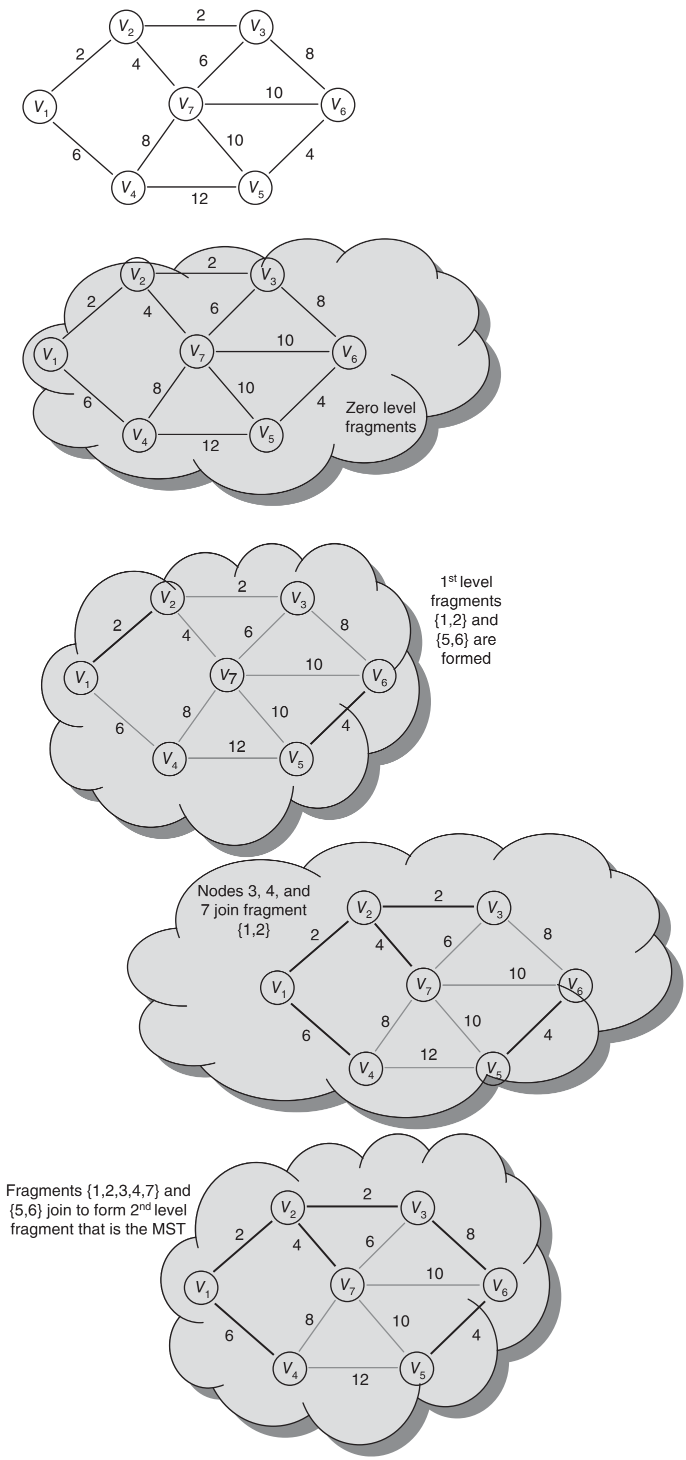


Figure 2.1.13 Example of the distributed algorithm

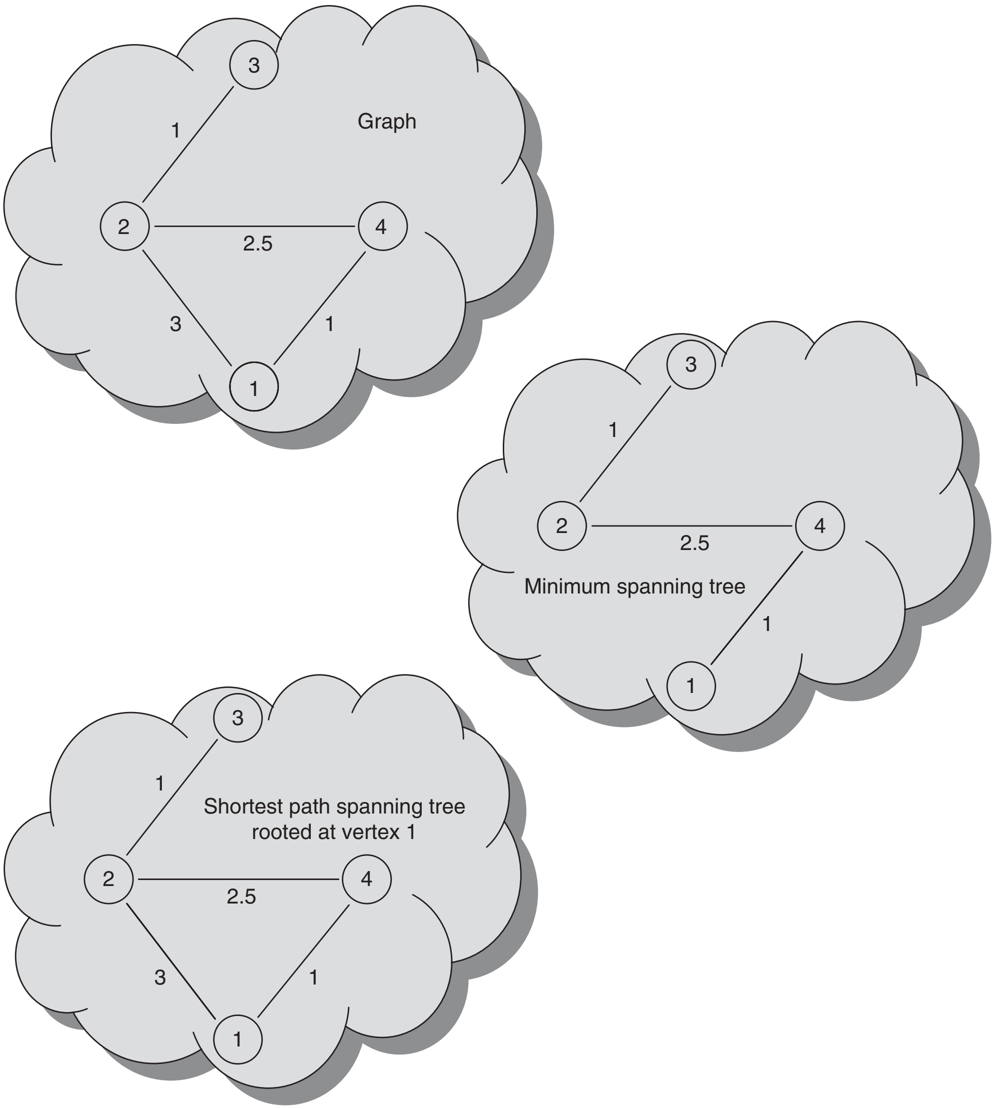


Figure 2.1.14 Examples of minimum spanning tree and shortest path spanning tree

Shortest Path Algorithms: Let us assume non-negative edge weights.

Given a weighted graph (G, W) and a node s (*source*), a shortest path tree rooted at s is a tree T such that, for any other node $v \in G$, the path between s and v in T is the shortest path between the nodes. Examples of the algorithms that compute these shortest path trees are Dijkstra and Bellman–Ford algorithms as well as algorithms that find the shortest path between all pairs of nodes, e.g. Floyd–Warshall.

Dijkstra Algorithm: For the source node s the algorithm is described with the following steps:

```
V' = {s}; U = V-{s};
E' = ϕ ;
For v ∈ U do
  D_v = w(s,v);
  P_v = s;
EndFor
While U ≠ ϕ do
  Find v ∈ U such that D_v is minimal;
  V' = V' ∪ {v}; U = U − {v};
  E' = E' ∪ (P_v,v);
  For x ∈ U do
    If D_v + w(v,x) < D_x then
       D_x = D_v + w(v,x);
       P_x = v;
    EndIf
  EndFor
EndWhile
```

An example of Dijkstra algorithm is given in Figure 2.1.15. It is assumed that $V1$ is s and D_v is the distance from node s to node v. If there is no edge connecting two nodes x and $y \rightarrow w(x,y) = \infty$.

The algorithm terminates when all the nodes have been processed and their shortest distance to node 1 has been computed. Note that the tree computed is not a minimum weight spanning tree. A MST for the given graph is given in Figure 2.1.16.

The *Bellman–Ford Algorithm* finds the shortest walk from a source node s to an arbitrary destination node v subject to the constraints that the walk consist of at most h hops and goes through node v only once. The algorithm is described with the following steps:

```
D_v^{-1} = ∞ ∀ v ∈ V;
D_s^0 = 0 and D_v^0 = ∞ ∀ v ≠ s, v ∈ V;
h = 0;
Until (D_v^h = D_v^{h-1} ∀ v ∈ V ) or (h = |V|) do
  h = h + 1;
  For v ∈ V do
     D_v^{h+1} = min{D_v^h + w(u,v)} u ∈ V;
  EndFor
EndUntil
```

An illustration for the *Bellman–Ford Algorithm* is given in Figure 2.1.17a.

The *Floyd–Warshall Algorithm* finds the shortest path between all ordered pairs of nodes (s,v), $\{s,v\}$, $v \in V$. Each iteration yields the path with the shortest weight between all pair of nodes under the constraint that only nodes $\{1,2,\ldots n\}$, $n \in |V|$, can be used as intermediary nodes on the computed paths. The algorithm is defined by the following steps.

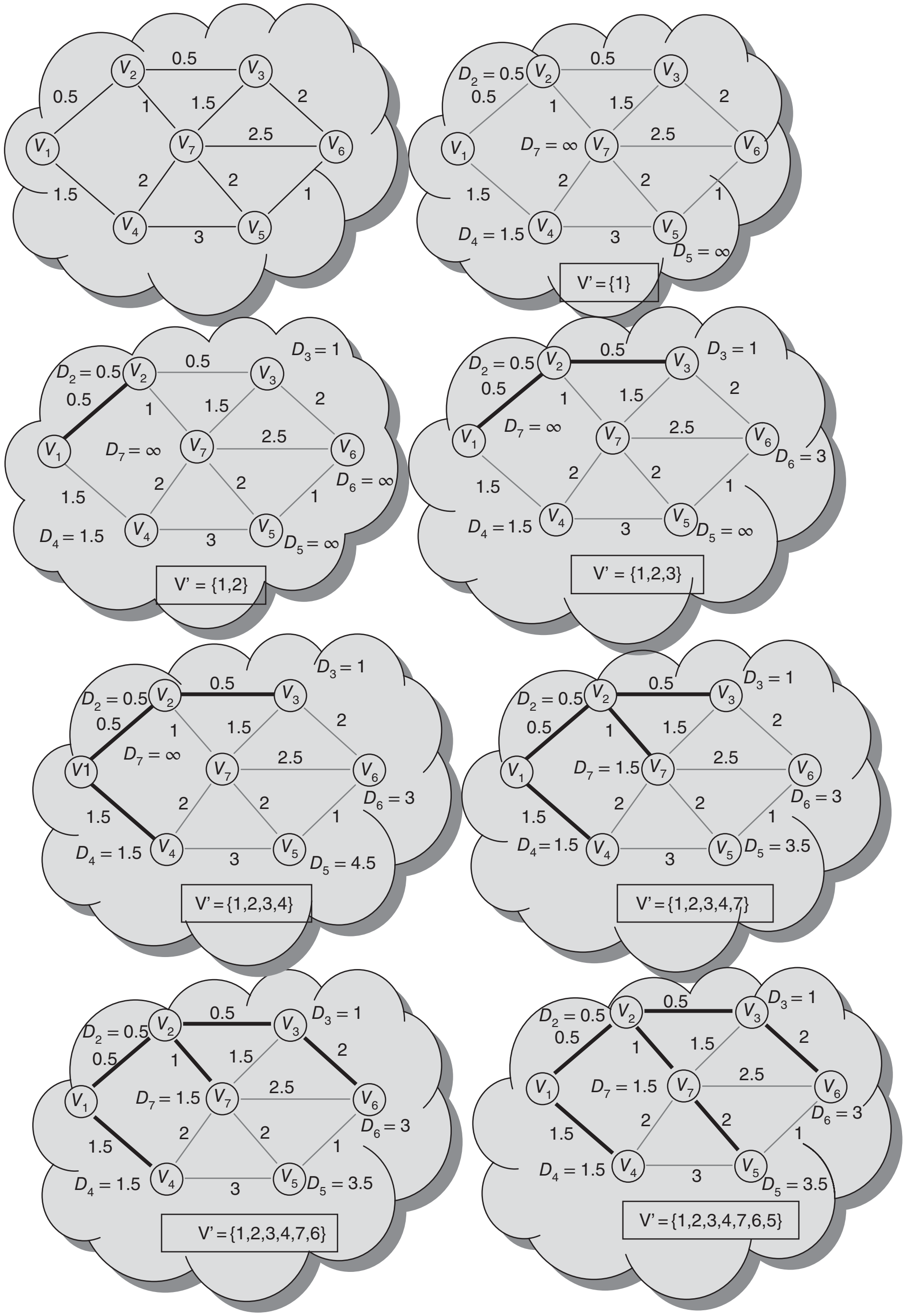


Figure 2.1.15 Example of Dijkstra algorithm

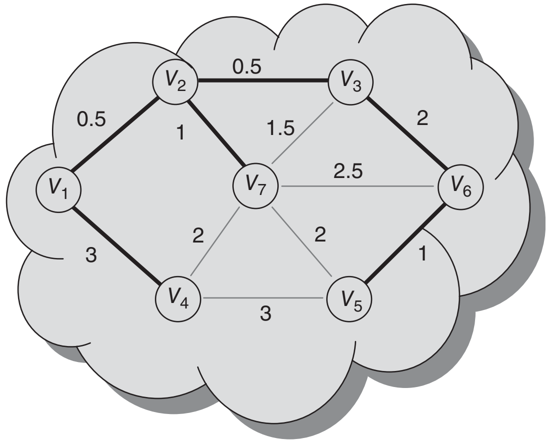


Figure 2.1.16 A MST for the basic graph in Figure 2.1.15

```
D = W; (W is the matrix representation of the edge weights)
For u = 1 to |V| do
  For s = 1 to |V| do
    For v = 1 to |V| do
        D_{s,v} = min{D_{s,v}, D_{s,u} + W_{u,v}}
    EndFor
  EndFor
EndFor
```

The algorithm completes in O(|V|3) time. An example of the *Floyd–Warshall Algorithm* is given in Figure 2.1.17b with $D = W$; (W is the matrix representation of the edge weights).

Distributed Asynchronous Shortest Path Algorithms: In this case each node computes the path with the shortest weight to every network node. There is no centralized computation. The control messaging is also required for distributed computation, as for the distributed MST algorithm. Asynchronous means here that there is no requirement for inter-node synchronization for the computation performed at each node or for the exchange of messages between nodes.

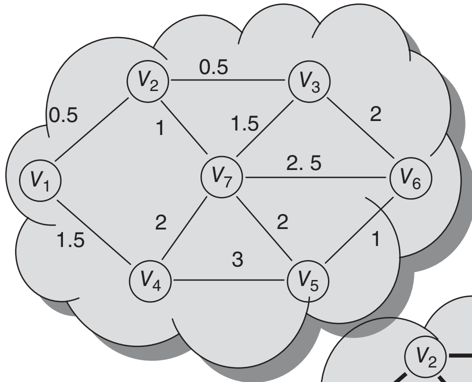
Distributed Dijkstra Algorithm: There is no need to change the algorithm. Each node floods periodically a control message throughout the network containing link state information. Transmission overhead is $O(|V| \times |E|)$. Entire topology knowledge must be maintained at each node. Flooding of the link state information allows for timely dissemination of the topology as perceived by each node. Each node has typically accurate information to be able to compute the shortest paths.

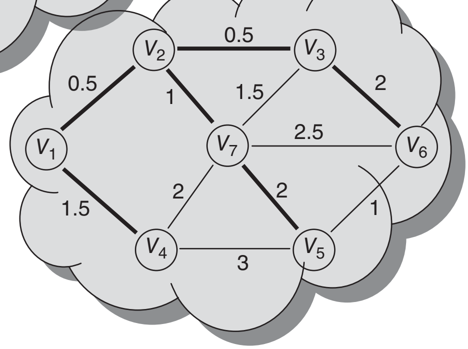
Distributed Bellman–Ford Algorithm: Assume G contains only cycles of *non-negative* weight. If $(u,v) \in E$ then so is (v,u). The update equation is

$$D_{s,v} = \min_{u \in N(s)} \{w(s,u) + D_{u,v}\} \forall v \in V - \{s\} \tag{2.1.1}$$

where $N(s)$ = Neighbors of $s \rightarrow \forall u \in N(s), (s,u) \in E$. Each node only needs to know the weights of the edges that are incident to it, the identity of all the network nodes and estimates

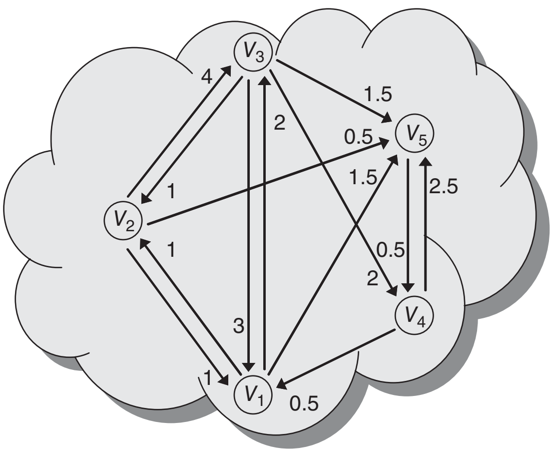
(a)





	$h=1$	$h=2$	$h=3$	$h=4$
D_2^h	0.5	0.5	0.5	0.5
D_3^h	∞	1	1	1
D_4^h	1.5	1.5	1.5	1.5
D_5^h	∞	4.5	3.5	3.5
D_6^h	∞	∞	3	3
D_7^h	∞	1.5	1.5	1.5

(b)

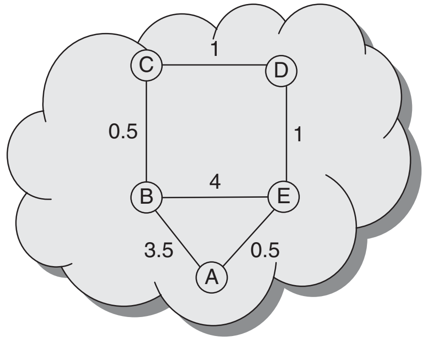


$$D_0 = \begin{bmatrix} 0 & 1 & 2 & \infty & 1.5 \\ 1 & 0 & 4 & \infty & 0.5 \\ 3 & 1 & 0 & 2 & 1.5 \\ 0.5 & \infty & \infty & 0 & 2.5 \\ \infty & \infty & \infty & 0.5 & 0 \end{bmatrix} \quad D_1 = \begin{bmatrix} 0 & 1 & 2 & \infty & 1.5 \\ 1 & 0 & 4 & \infty & 0.5 \\ 3 & 1 & 0 & 2 & 1.5 \\ 0.5 & 1.5 & 2.5 & 0 & 2 \\ \infty & \infty & \infty & 0.5 & \infty \end{bmatrix} \quad D_2 = \begin{bmatrix} 0 & 1 & 2 & \infty & 1.5 \\ 1 & 0 & 3 & \infty & 0.5 \\ 2 & 1 & 0 & 2 & 1.5 \\ 0.5 & 1.5 & 2.5 & 0 & 2 \\ \infty & \infty & \infty & 0.5 & 0 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} 0 & 1 & 2 & 4 & 1.5 \\ 1 & 0 & 3 & 5 & 0.5 \\ 2 & 1 & 0 & 2 & 1.5 \\ 0.5 & 1.5 & 2.5 & 0 & 2 \\ \infty & \infty & \infty & 0.5 & 0 \end{bmatrix} \quad D_4 = \begin{bmatrix} 0 & 1 & 2 & 4 & 1.5 \\ 1 & 0 & 3 & 5 & 0.5 \\ 2 & 1 & 0 & 2 & 1.5 \\ 0.5 & 1.5 & 2.5 & 0 & 2 \\ 1 & 2 & 3 & 0.5 & 0 \end{bmatrix} \quad D_5 = \begin{bmatrix} 0 & 1 & 2 & 2 & 1.5 \\ 1 & 0 & 3 & 1 & 0.5 \\ 2 & 1 & 0 & 2 & 1.5 \\ 0.5 & 1.5 & 2.5 & 0 & 2 \\ 1 & 2 & 3 & 0.5 & 0 \end{bmatrix}$$

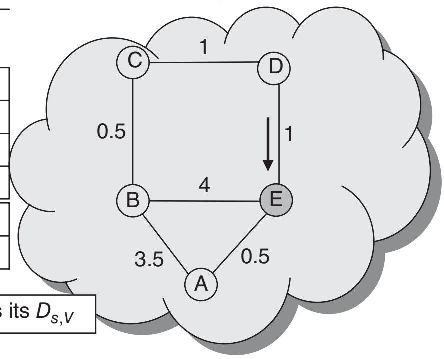
Figure 2.1.17 (a) An illustration for Bellman–Ford algorithm (b) An example of Floyd–Warshall algorithm

Initial $D_{s,V}$					
s	A	B	C	D	E
A	0	3.5	∞	∞	0.5
B	3.5	0	0.5	∞	4
C	∞	0.5	0	1	∞
D	∞	∞	1	0	1
E	0.5	4	∞	1	0



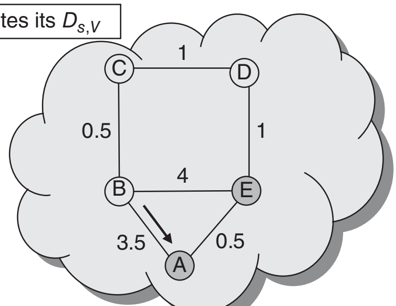
$D_{s,V}$					
s	A	B	C	D	E
A	0	3.5	∞	∞	0.5
B	3.5	0	0.5	∞	4
C	∞	0.5	0	1	∞
D	∞	∞	1	0	1
E	0.5	4	2	1	0

E receives D's routes and updates its $D_{s,V}$



A receives B's routes and updates its $D_{s,V}$

$D_{s,V}$					
s	A	B	C	D	E
A	0	3.5	4	∞	0.5
B	3.5	0	0.5	∞	4
C	∞	0.5	0	1	∞
D	∞	∞	1	0	1
E	0.5	4	2	1	0



A receives E's routes and updates its $D_{s,V}$

$D_{s,V}$					
s	A	B	C	D	E
A	0	3.5	2.5	1.5	0.5
B	3.5	0	0.5	∞	4
C	∞	0.5	0	1	∞
D	∞	∞	1	0	1
E	0.5	4	2	1	0

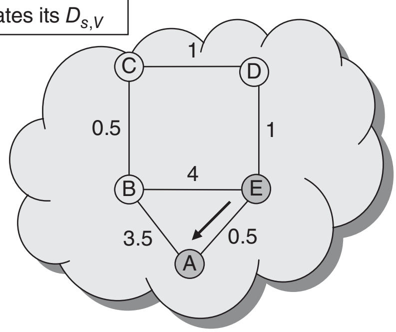
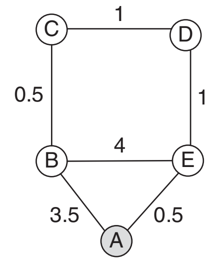


Figure 2.1.18 Distributed Bellman–Ford algorithm example

A's routing table		
Destination	Next Hop	Distance
B	E	3
C	E	2.5
D	E	1.5
E	E	0.5



E's routing table		
Destination	Next Hop	Distance
A	A	0.5
B	D	2.5
C	D	2
D	D	1

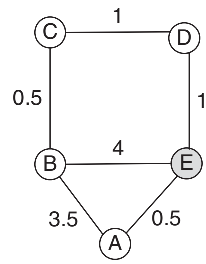


Figure 2.1.18 (Continued)

(received from its neighbors) of the distances to all network nodes. The algorithm includes the following steps:

- Each node s transmits to its neighbors its current distance vector $D_{s,V}$.
- Likewise each neighbor node $u \in N(s)$ transmits to s its distance vector $D_{u,V}$.
- Node s updates $D_{s,v}$, $\forall\, v \in V - \{s\}$ in accordance with (2.1.1). If any update changes a distance value then s sends the current version of $D_{s,v}$ to its neighbors.
- Node s updates $D_{s,v}$ every time that it receives a distance vector information from any of its neighbors.
- A periodic timer prompts node s to recompute $D_{s,V}$ or to transmit a copy of $D_{s,V}$ to each of its neighbors.

An example of the Distributed Bellman–Ford Algorithm is given in Figure 2.18.

Distance Vector Protocols: With this protocol each node maintains a routing table with entries {Destination, Next Hop, Distance (cost)}.

Nodes exchange routing table information with their neighbors: (a) whenever the table changes, (b) periodically.

Upon reception of a routing table from a neighbor, a node updates its routing table if it finds a "better" route. Entries in the routing table are deleted if they are too old, that is, they are not "refreshed" within a certain time interval by the reception of a routing table.

Link Failure:

1. Simple rerouting case shown in Figure 2.1.19:
   - F detects that link to G has failed.
   - F sets a distance of ∞ to G and sends update to A.
   - A sets a distance of ∞ to G since it uses F to reach G.

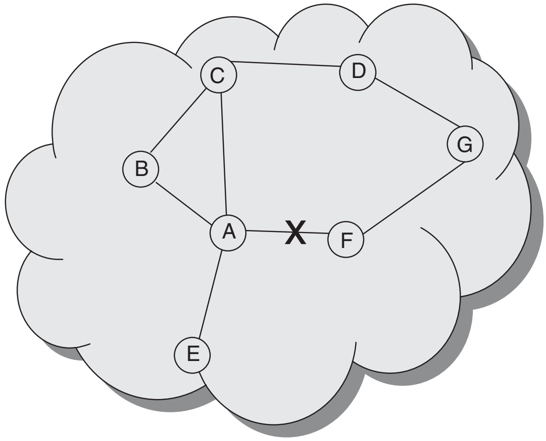


Figure 2.1.19 Simple rerouting case

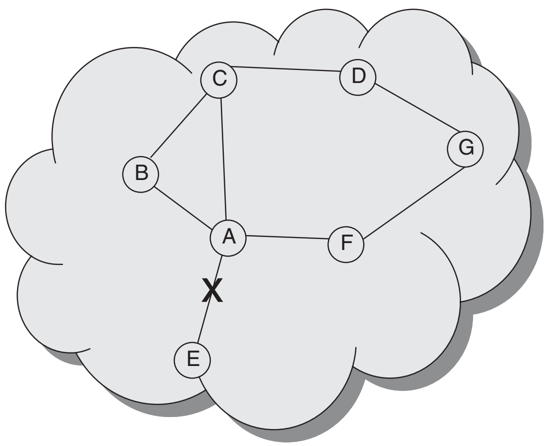


Figure 2.1.20 Routing loop case

- A receives periodic update from C with two-hop path to G (via D).
- A sets distance to G to three and sends update to F.
- F decides it can reach G in four hops via A.

2. Routing loop case shown in Figure 2.1.20
   - Link from A to E fails.
   - A advertises distance of ∞ to E.
   - B and C had advertised a distance of two to E (prior to the link failure).
   - Upon reception of A's routing update B decides it can reach E in three hops; and B advertises this to A.
   - A decides it can reach E in four hops; A advertises this to C.
   - C decides that it can reach E in five hops.

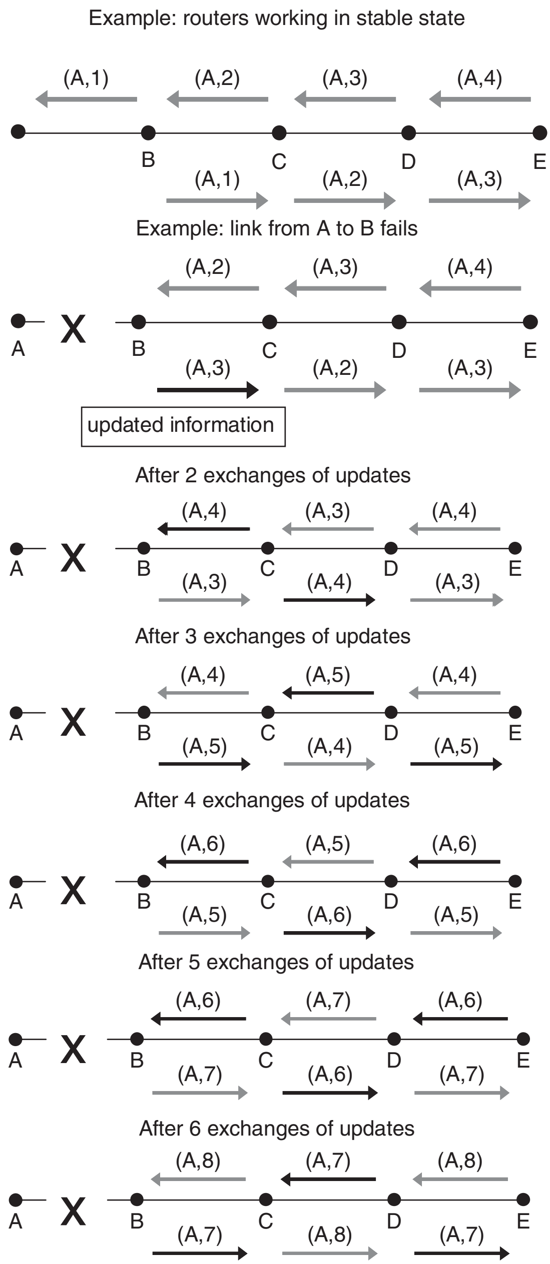


Figure 2.1.21 Count to infinity problem

This behavior is called *count to infinity.* This problem is further elaborated in Figure 2.1.21. In the figure routing updates with distance to A are shown. When link from A to B fails, B can no longer reach A directly, but C advertises a distance of two to A and thus B now believes it can reach A via C and advertises it. This continues until the distance to A reaches infinity.

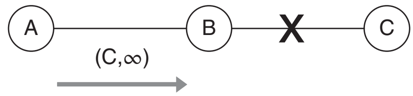


Figure 2.1.22 Split Horizon algorithm

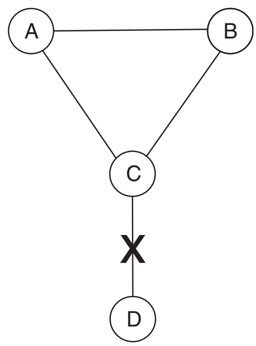


Figure 2.1.23 Example where Split Horizon fails

Split Horizon Algorithm: Used to avoid (not always) the count to infinity problem. If A, in Figure 2.1.22, routes to C via B, then A tells B that its distance to C is ∞.

As result, B will not route to C via A if the link B to C fails. Works for two node loops, but not for loops with more than two nodes.

An example where *Split Horizon* fails is shown in Figure 2.1.23.

When link C to D breaks, C marks D as unreachable and reports that to A and B. Suppose A learns it first. A now thinks best path to D is through B. A reports D unreachable to B and a route of cost three to C.

C thinks D is reachable through A at cost four and reports that to B. B reports a cost five to A who reports new cost to C, and so on.

Routing Information Protocol (*RIP*): RIP, was originally distributed with BSD Unix. Widely used on the Internet (internal gateway protocol). RIP updates are exchanged in ordinary IP datagrams. RIP sets infinity to 16 hops (cost $\in$ [0–15]).

RIP updates neighbors every 30 s, or when routing tables change.

2.2 Graph Theory

The previous section summarizes the basic relations and definitions in graph theory. A number of references that cover graph theory in depth is available. The books [1–10] all discuss the various aspects of graph theory in general, not merely from a communication network point of view. Reference [7] has become a standard in this field. Reference [3] contains a lot of material related to the hop distance and graph invariants defined in terms of distance properties, much of which may have applications to communications networks. Reference [8] has a comprehensive discussion of connectivity in directed networks. Reference [4] presents a number of illuminative examples of relations holding among common graph invariants, such as node and edge connectivity, including counter examples which indicate the extent to which these results are best possible.

Graph theoretic algorithms are given good coverage in Ref. [11] and in the two books [12, 13], but [14] is probably the best introduction to this subject. Additional references are [11–28]. Colbourn's book [18] is the only general text available at this time which covers network reliability. McHugh [22] includes a chapter on the implementation of graph algorithms for concurrent and parallel computation. Tenenbaum *et al.* [27, 28] discuss the implementation of some of these algorithms in the Pascal and C languages. See also Refs. [19, 20] for more discussion of using data structures and graph algorithms in these computer languages. Christofides [17] discusses some important graph algorithms in much more detail, including the traveling salesman and Hamiltonian tour problems. A number of these works also treat some topics involved with flows in networks, a subject which we have not discussed so far but one which does have important applications to network connectivity, as discussed in Chapter 13 of this book (see also article by Estahanian and Hakimi [21]).

References [29–38] discuss the efficiency and complexity of computer algorithms in general, not merely the graph theoretic algorithms. Here the two books by Aho *et al.* [29,30] and Knut's three volume series [36] are the best general references to the theory and practice of computer algorithms. Garey and Johnson [32] is the best overall guide to Nondeterministic Polynomial (NP) completeness and provides a compendium of many of those problems that were known to be NP-complete as of 1979. A problem is NP-hard if an algorithm for solving it can be translated into one for solving any other NP problem (NP time) problem. NP-hard therefore means "at least as hard as any NP problem," although it might, in fact, be harder. There are now more than 1000 known NP-complete problems, many of them in graph theory, and dozens more are discovered every year, so this catalog has rapidly become out of date. As a result there is now an ongoing column on NP-complete problems by Johnson [35] which appears several times a year in the journal *Algorithms.* A number of these columns have discussed NP completeness for problems in communication networks and reliability. Harel [33] is a very good and exceptionally readable overall account of the current state of the art in algorithmics and has a good account of the problems that arise in designing and verifying algorithms for parallel processing. The book by Sedgewick [37] also comes in two other editions which give more details of the implementations of these algorithms in either Pascal or C.

Problems concerning the statistical dependence of the network component failures are treated in the Refs. [39–46]. One should note that the assumption of independent failures can lead to either an overly pessimistic or an overly optimistic estimate of the true network reliability. The paper by Egeland and Huseby [39] gives some results as to how one might determine which of these is the case.

Most of the probabilistic measures of network connectivity lead to computability problems that are NP-hard, so there has been considerable effort in searching for restricted classes of networks for which there are reliability algorithms with a smaller order of complexity [47–57]. For example, the papers of Boesch [50] and Pullen [56] consider only the constant probability of edge failures. This may reduce the problem to one in graph enumeration, but this problem still has nonpolynomial complexity. Similarly, the article by Bienstock [48] considers only planar networks, and he proves the existence of an algorithm whose complexity grows exponentially in the square root of p, rather than p itself. This is still very far from having polynomial growth, and to obtain that complexity even more drastic restrictions are necessary, as shown in the articles by Agrawal and Satayanarana [47] and by Politof and Satyanarayana [55]. Even very regular grid networks in the *xy* plane yields NP-hard problems, as shown by the Clark and Colbourn article [52].

Due to the computational intractability of the network reliability calculations for general probabilistic graphs, there are a very large number of papers devoted to the problem of obtaining bounds and approximations of the reliability [58–68]. For the same reason Monte Carlo simulations are also used in this field [69–75].

The node connectivity factor (NCF) and the link connectivity factor (LCF) were introduced in the papers [76–82] as possible alternatives to the usual reliability measure. They are indicators of how close the network is to being totally disconnected. Unfortunately, the NCF at least is computationally difficult to compute and does not seem to be amenable to simplifying techniques such as factorization or edge reduction used by othermethods. Thus it is not yet clear how useful a concept this will prove to be, although if these connectivity factors are available they can be used to identify the most vulnerable components of the network and to adapt the network so as to equalize the vulnerability over its components.

The papers [83–96] are concerned with some aspects of graph connectivity other than the usual path oriented one, primarily with those deriving from the notion of the diameter of a graph (i.e., the maximum node to node hop distance across the graph) or the average node to node hop distance. Of special interest here is the notion of leverage, as described in the papers of Bagga *et al.* [83], which is a general method of quantifying changes in graph invariants due to the loss of some network components.

A number of Refs. [97–109] are concerned with a number of other graph invariants that have an obvious connection with the notions of vulnerability and survivability of communications networks. The main concepts here are those of dominance, independence, and covering of a graph with respect to either a set of nodes or a set of edges of the underlying graph. These quantities have already been applied to problems involving networks used in scheduling and service facilities, though their applications and usefulness to communications networks remains to be determined. Also, the calculation of some of these quantities can be NP-hard (some in the deterministic sense, others from the probabilistic point of view). This is also an area of very active research.

2.3 Routing with Topology Aggregation

The goal of quality of service (QoS) routing is to find a network path from a source node to a destination node, which has sufficient resources to support the QoS requirements of a connection request. The execution time and the space requirement of a routing algorithm increase with the size of the network, which leads to the scalability problem. For very large networks, it is impractical to broadcast the whole topology to every node for the purpose of routing. In order to achieve scalable routing, large networks are structured hierarchically by grouping nodes into different domains [111,112]. The internal topology of each domain is then aggregated to show only the cost of routing across the domain, that is, the cost of going from one *border node* (a node that connects to another domain) to another border node. This process is called *topology aggregation*. One typical way of storing the aggregated topology is for every node to keep detailed information about the domain that it belongs to, and aggregated information about the other domains.

Since the network after aggregation is represented by a simpler topology, most aggregation algorithms suffer from distortion, that is, the cost of going through the aggregated network deviates from the original value [113]. Nevertheless, [114] showed that topology aggregation reduces the routing overhead by orders of magnitude and does not always have a negative impact on routing performance. Some aggregation approaches have been proposed. Reference [115] presented algorithms that find a minimum distortion-free representation for an *undirected* network with either a single additive or a single bottleneck parameter. Examples of additive metrics are delay and cost, while an example of a bottleneck parameter is bandwidth. For an additive constraint, it may require $O(|B|^2)$ links to represent a domain in the distortion-free aggregation, where $|B|$ is the number of border nodes in the domain. Reference [116] proposed an algorithm that aggregates *directed* networks with a single additive parameter by using $O(|B|)$ links. The algorithm achieves bounded distortion with a worst-case distortion factor of $O\left(\sqrt{\rho}\log|B|\right)$, where ρ is the *network asymmetry constant*, defined as the maximum ratio between the QoS parameters of a pair of opposite directed links.

In this section, we discuss networks with two QoS parameters, *delay* and *bandwidth*. Some related work can be found in Refs. [117–119]. Reference [117] presented an aggregation method that aggregates an undirected delay bandwidth sensitive domain into a spanning tree among border nodes. Therefore, there is a unique path between each pair of border nodes after aggregation and the space complexity is $O(|B|)$. The paper showed that a spanning tree can provide a distortion-free aggregation for bandwidth, but not for delay. Reference [118] studied the problem of topology aggregation in networks of six different QoS parameters. The aggregated topology follows the ATM Private Network–Network Interface (PNNI) standard [111]. The authors proposed to minimize the distortion by using a linear programming approach. Both [117] and [118] assumed certain precedence order among the parameters, so that among several paths that go between the same pair of border nodes, one path can be selected as the "best" path. The state of a path in a delay bandwidth sensitive network can be represented as a delay bandwidth pair [119]. If there are several paths going across a domain, a single pair of values, which is a point on the delay bandwidth plane, is not sufficient to capture the QoS parameters of all those paths [120].

Referfence [119] was the first to use a curve on the delay bandwidth plane to approximate the properties of multiple physical paths between two border nodes, without assuming any precedence among the parameters. A curve is defined by three values: the minimum delay, the maximum bandwidth, and the smallest stretch factor among all paths between two border nodes. The stretch factor of a path measures how much the delay and the bandwidth of the path deviate from the best delay and the best bandwidth of all paths. The curve provides better approximation than a single point, but this approach has several shortcomings. First, the paper did not provide a routing algorithm with polynomial complexity to *find* a feasible path based on the aggregated topology. Instead, it provided an algorithm to *check* if a given path is likely to be feasible. Essentially, the algorithm determined whether the point, defined by the delay/bandwidth requirement, is within the curve defined by the delay, bandwidth, and stretch factor of the path. Second, although the paper provided an aggressive heuristic to find the stretch factor of an interdomain path, there are cases where only one QoS metric will contribute to the value, and the information about the other metric is lost.

In this section, we discuss a way of representing the aggregated state in delay bandwidth sensitive networks by using line segments. The approach solves some problems in Ref. [119] and other traditional approaches by introducing a specific QoS parameter representation, a specific aggregation algorithm, and the corresponding routing protocol. The algorithm outperforms others due to smaller distortion.

2.3.1 Network and Aggregation Models

A large network consists of a set of domains and links that connect the domains. It is modeled as a directed graph, where link state can be asymmetric in two opposite directions. Figures 2.3.1 and 2.3.2 are examples of a network with four domains. There are two kinds of nodes in a domain. A node is called a *border node* if it connects to a node of another domain. A node is an *internal node* if it is not a border node. A domain is modeled as a tuple (V,B, E), where V is the set of nodes in the domain, $B \subseteq V$ is the set of border nodes, and E is the set of directed links among the nodes in V. The entire network is modeled as (G, L), where $G = \{g_i | g_i = (V_i, B_i, E_i), 1 \leq i \leq \eta\}$ is the set of domains, L is a set of links that connect border nodes of different domains, and η is the number of domains in G.

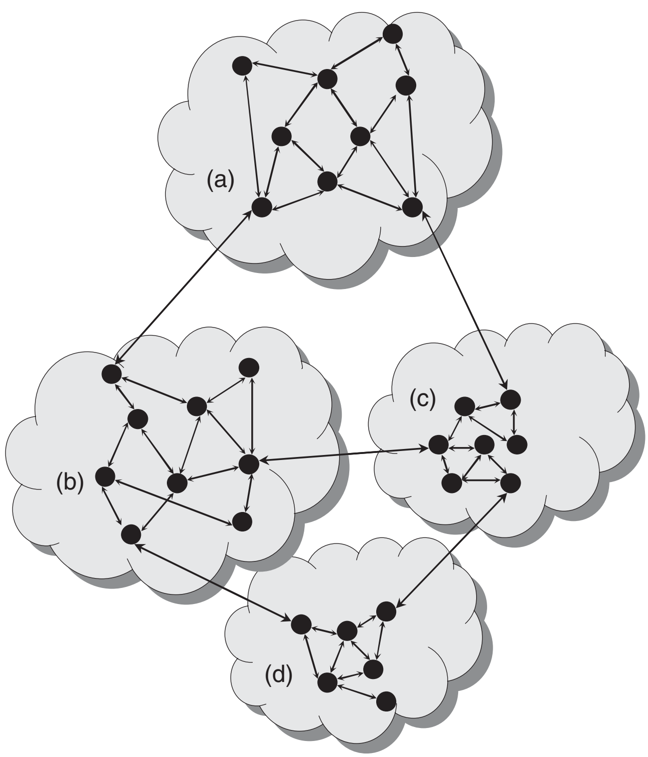


Figure 2.3.1 Network example

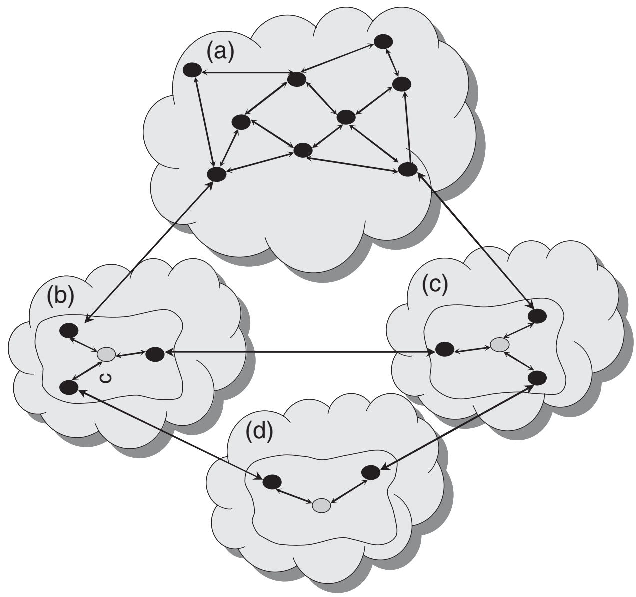


Figure 2.3.2 Aggregated network from Figure 2.3.1 with a complete view of domain A

There are several aggregation models for large networks. In this section, we use the topology aggregation model proposed by the PNNI [111,121]. One of the representative topologies in PNNI is the *star* topology. Other popular ones are *simple node* and *mesh*. In a simple node topology, a domain is collapsed into one virtual node. This offers the greatest reduction of information as the space complexity after aggregation is $O(1)$, but the distortion is large. The mesh topology is a complete graph among the border nodes. The complexity of this topology is $O(|B|^2)$ and its distortion is much smaller. The star topology is a compromise between the above two. It has a space complexity of $O(|B|)$ and the distortion is between those of a simple node and a mesh. Reference [122] compares the performance of the above three aggregation methods. It shows that the star topology outperforms the simple-node and performs slightly worse than the mesh in a uniform network.

Let us consider the domain in Figure 2.3.3a, where nodes *a, b, c,* and *d* are the border nodes. The mesh aggregation is shown in Figure 2.3.3b, and the star aggregation is shown in Figure 2.3.3c. In a star topology, the border nodes connect via links to a virtual *nucleus*. These links are called *spokes*. Each link is associated with some QoS parameters. To make the representation more flexible, PNNI also allows a limited number of links connected directly between border nodes. These links are called *bypasses*.

Figure 2.3.3d is an example of a star with bypasses. We call the links in an aggregated topology as *logical links* since they are not real.

After aggregation, a node in a domain sees all other nodes in the same domain, but only aggregated topologies of the other domains. For example, for the network in Figure 2.3.1, the aggregated view of the network stored at a node in Domain A is shown in Figure 2.3.2. In such a view, the topology of Domain A is exactly the same as the original one but the topologies of the other domains are now represented by border nodes, nuclei, and spokes (without

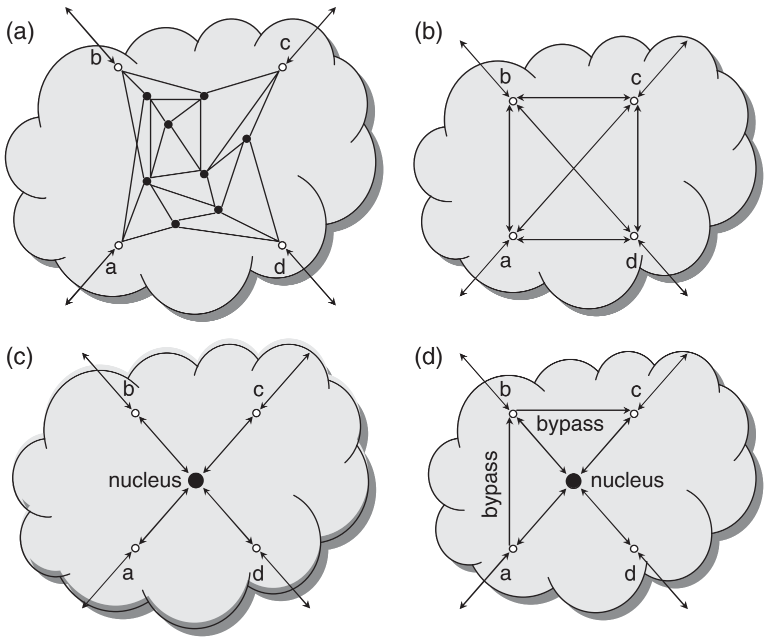


Figure 2.3.3 Topology aggregation. (a) Domain F. (b) Mesh of the borders. (c) Star representation. (d) Star representation with bypasses

bypasses in this example). For a large network, this aggregated view is significantly smaller than the original topology and thus scalability is achieved. However, for the purpose of QoS routing, it is extremely important to develop solutions on how to represent the state information in this aggregated topology and how to control the information loss due to aggregation. For these details see Ref. [110].

References

[1] Berge, C. (1985) *Graphs*, North-Holland, New York.
[2] Bollobas, B. (1979) *Graph Theory: An Introductory Course*, Springer, New York.
[3] Buckley, F. and Haraly, F. (1990) *Distance in Graphs*, Addison-Wesley, Reading, MA.
[4] Capobianco, M. and Molluzzo, J.C. (1978) *Examples and Counterexamples in Graph Theory*, North-Holland, New York.
[5] Chen, W.-K. (1990) *Theory of Nets. Flows in Networks*, John Wiley & Sons, Inc., New York.
[6] Deo, N. (1974) *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ.
[7] Harary, F. (1969) *Graph Theory*, Addison-Wesley, Reading, MA.
[8] Harary, F., Norman, R.Z. and Cartwright, D. (1965) *Structural Models: An Introduction to the Theory of Directed Graphs*, John Wiley & Sons, Inc., New York.
[9] Harary, F. and Palmer, E.M. (1973) *Graphical Enumeration*, Academic Press, New York.
[10] Robinson, D.F. and Foulds, L.R. (1980) *Digraphs: Theory and Techniques*, Gordon and Breach, New York.
[11] Tarjan, R.E. (1983) *Data Structures and Network Algorithms*, SIAM, Philadelphia, PA.
[12] Even, S. (1979) *Graph Algorithms*, Computer Science Press, Rockville, MD.

[13] Even, S. (1973) *Algorithmic Combinatorics*, Macmillan, New York.
[14] Gibbons, A. (1985) *Algorithmic Graph Theory*, Cambridge University Press, Cambridge.
[15] Ammeraal, L. (1987) *Programs and Data Structures in C*, John Wiley & Sons, Inc., New York.
[16] Arnsbury, W. (1985) *Data Structures: From Arrays to Priority Queues*, Wadsworth, Belmont, CA.
[17] Christofides, N. (1975) *Graph Theory: An Algorithmic Approach*, Academic Press, New York.
[18] Colbourn, C.J. (1987) *The Combinatorics of Network Reliability*, Oxford University Press, London.
[19] Dale, N. and Lilly, S.C. (1988) *Pascal Plus Data Structures*, 2nd edn, D. C. Heath, Lexington, MA.
[20] Esakov, J. and Weiss, T. (1989) *Data Structures: An Advanced Approach Using C*, Prentice-Hall, Englewood Cliffs, NJ.
[21] Estahanian, A.H. and Hakimi, S.L. (1984) On computing the connectivity of graphs and digraphs. *Networks*, **14**, 355–366.
[22] McHugh, J.A. (1990) *Algorithmic Graph Theory*, Prentice Hall, Englewood Cliffs, NJ.
[23] Manber, U. (1989) *Introduction to Algorithms*, Addison-Wesley, Reading, MA.
[24] Minieka, E. (1978) *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, New York.
[25] Papadimitriou, C.H. and Steiglitz, K. (1982) *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ.
[26] Swarny, M.N.S. and Thulasiraman, K. (1981) *Graphs, Networks, and Algorithms*, John Wiley & Sons, Inc., New York.
[27] Tenenbaum, A.M. and Augenstein, M.J. (1986) *Data Structures Using Pascal*, 2nd edn, Prentice-Hall, Englewood Cliffs, NJ.
[28] Tenenbaum, A.M., Langsarn, Y. and Augenstein, M.J. (1990) *Data Structures Using C*, Prentice-Hall, Englewood Cliffs, NJ.
[29] Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1974) *The Design and Analysis of Computer Programs*, Addison-Wesley, Reading, MA.
[30] Aho, A.V., Hopcroft, J.E. and Ullrnan, J.D. (1983) *Data Structures and Algorithms*, Addison-Wesley, Reading, MA.
[31] Cormen, T.H., Leiserson, C.E. and Rivest, R.L. (1990) *Introduction to Algorithms*, MIT Press, Cambridge MA.
[32] Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA.
[33] Hare, D. (1987) *Algorithmics The Spirit of Computing*, Addison-Wesley, Reading, MA.
[34] Horowitz, E. and Sahni, S. (1978) *Fundamentals of Computer Algorithms*, Computer Science Press, Rockville, MD.
[35] Johnson, D.S. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, **2**, 393–405, 1981 (this column appeared several times per year).
[36] (a) Knuth, D.E. (1973) *The Art of Computing: Fundamental Algorithms*, vol. **1**, 2nd edn, Addison-Wesley, Reading, MA; (b) Knuth, D.E. (1981) *The Art of Computing: Seminumerical Algorithms*, vol. **2**, 2nd edn, Addison-Wesley, Reading, MA; (c) Knuth, D.E. (1973) *The Art of Computing: Sorting and Searching*, vol. **3**, Addison-Wesley, Reading, MA.
[37] Sedgewick, R. (1988) *Algorithms*, 2nd edn, Addison-Wesley, Reading, MA.
[38] Wilf, H.S. (1986) *Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ.
[39] Egeland, T. and Huseby, A.B. (1991) On dependence and reliability computation. *Networks*, **21**, 521–546.
[40] Heffes, H. and Kumar, A. (1986) Incorporating dependent node damage in deterministic connectivity analysis and synthesis of networks. *Networks*, **l6**, Sl–S65.
[41] Lam, Y.F. and Li, V. (1977) *On Network Reliability Calculations with Dependent Failures*. Proceedings of the IEEE 1983 Global Telecommunications Conference (GTC'83), November, 1977, San Diego, CA, pp. 1499–1503.
[42] Lam, Y.F. and Li, V. (1985) *Reliability Modeling and Analysis of Communication Networks with Dependent Failures*. Proceedings IEEE INFOCOM, pp. 196–199.
[43] Lam, Y.F. and Li, V. (1986) Reliability modeling and analysis of communication networks with dependent failures. *IEEE Transactions on Communications*, **34**, 82–84.
[44] Lee, K.V. and Li, V.O.K. (1990) *A Path-Based Approach for Analyzing Reliability of Systems with Dependent Failures and Multinode Components*. Proceedings IEEE INFOCOM, pp. 495–503.
[45] Page, L.B. and Perry, J.E. (1989) A model for system reliability with common-cause failures. *IEEE Transactions on Reliability*, **38**, 406–410.
[46] Zemel, E. (1982) Polynomial algorithms for estimation of network reliability. *Networks*, **12**, 439–452.

[47] Agrawal, A. and Satayanarana, A. (1984) An OIE time algorithm for computing the reliability of a class of directed networks. *Operations Research*, **32**, 493–515.
[48] Bienstock, D. (1986) An algorithm for reliability analysis of planar graphs. *Networks*, **16**, 411–422.
[49] Beichelt, F. and Tittman, P. (1991) A generalized reduction method for the connectedness probability of stochastic networks. *IEEE Transactions on Reliability*, **40**, 198–204.
[50] Boesch, F.T. (1988) On unreliability polynomials and graph connectivity in reliable network synthesis. *Journal of Graph Theory*, **10**, 339–352.
[51] Bobbio, A. and Premoli, A. (1982) Fast algorithm for unavailability and sensitivity analysis of series-parallel systems. *IEEE Transactions on Reliability*, **31**, 359–361.
[52] Clark, B.N. and Colbourn, C.L. (1990) Unit disk graphs. *Discrete Mathematics*, **86**, 165–177.
[53] Colbourn, C.L. (1987) Network resilience. *SIAM Journal of Algebra and Discrete Mathematics*, **8**, 404–409.
[54] Debany, W.H., Varshney, P.K. and Hartman, C.R.P. (1986) Network reliability evaluation using probability expressions. *IEEE Transactions on Reliability*, **35**, 161–166.
[55] Politof, T. and Satyanarayana, A. (1990) A linear-time algorithm to compute the reliability of planar cube-free networks. *IEEE Transactions on Reliability*, **39**, 557–563.
[56] Pullen, K.W. (1986) A random network model of message transmission. *Networks*, **16**, 397–409.
[57] Yang, O.W.W. (1991) *Terminal Pair Reliability of Tree-Type Computer Communication Networks*. Proceedings of the IEEE 1991 Military Communications Conference (MILCOM'91), November, 1991.
[58] AboElFotoh, H.M. and Colbourn, C.J. (1989) Computing 2-terminal reliability for radio-broadcast networks. *IEEE Transactions on Reliability*, **38**, 538–555.
[59] Ball, M.O. and Provan, J.S. (1983) Calculating bounds on reachability and connectedness in stochastic networks. *Networks*, **13**, 253–278.
[60] Ball, M.O. and Provan, J.S. (1981) Bounds on the reliability polynomial for shellable independence systems. *SIAM Journal of Algebra and Discrete Mathematics*, **3**, 166–181.
[61] Brecht, T.B. and Colbourn, C.J. (1986) Improving reliability bounds in computer networks. *Networks*, **16**, 369–380.
[62] Brecht, T.B. and Colbourn, C.J. (1989) Multiplicative improvements in network reliability bounds. *Networks*, **19**, 521–529.
[63] Brecht, T.B. and Colbourn, C.L. (1988) Lower bounds on two-terminal network reliability. *Discrete Applied Mathematics*, **21**, 185–198.
[64] Colbourn, C.J. and Ramanathan, A. (1987) Bounds for all-terminal reliability by arc packing. *Ars Combinatoria*, **23A**, 229–236.
[65] Colboum, C.J. (1988) Edge-packings of graphs and network reliability. *Discrete Mathematics*, **72**, 49–61.
[66] Colbourn, C.J. and Harms, D.D. (1988) Bounding all-terminal reliability in computer networks. *Networks*, **18**, 1–12.
[67] Torrieri, D. (1991) *Algorithms for Finding an Optimal Set of Short Disjoint Paths in a Communication Network*. Proceedings of the IEEE 1991 Military Communications ConJ (MILCOM'91), November, 1991.
[68] Wagner, D.K. (1990) Disjoint (s, t)-cuts in a network. *Networks*, **20**, 361–371.
[69] Fishman, G.S. (1986) A comparison of four Monte Carlo methods for estimating the probability of st-connectedness. *IEEE Transactions on Reliability*, **35**, 145–154.
[70] Fishman, G.S. (1986) A Monte Carlo sampling plan for estimating network reliability. *Operations Research*, **34**, 581–594.
[71] Fishman, G.S. (1989) Estimating the st-reliability function using importance and stratified sampling. *Operations Research*, **37**, 462–473.
[72] Fishman, G.S. (1987) A Monte Carlo sampling plan for estimating reliability parameters and related functions. *Networks*, **17**, 169–186.
[73] Karp, R.M. and Luby, M.G. (1985) *Monte-Carlo Algorithms for Enumeration and Reliability Problems*. Proceedings of the IEEE 24th Annual Symposium on Foundations of Computer Science, November 7–9, Tucson, AZ, pp. 56–64.
[74] Kubat, P. (1989) Estimation of reliability for communication/computer networks—simulation/analytic approach. *IEEE Transactions on Communications*, **37**, 927–933.
[75] Nd, L.D. and Colbourn, C.J. (1990) Combining Monte Carlo estimates and bounds for network reliability. *Networks*, **20**, 277–298.
[76] Newport, K.T. and Schroeder, M.A. (1987) *Network Survivability through Connectivity Optimization*. Proceedings of the 1987 IEEE International Conference on Communications, Vol. **1**, pp. 471–477.

[77] Newport, K.T. and Varshney, P. (1991) Design of communications networks under performance constraints. *IEEE Transactions on Reliability*, **40**, 443–439.
[78] Newport, K.T. and Schroeder, M.A. (1990) *Techniques for Evaluating the Nodal Survivability of Large Networks*. Proceedings of the IEEE 1990 Military Communications Conference (MILCOM'90), Monterey, CA, pp. 1108–1113.
[79] Newport, K.T., Schroeder, M.A., and Whittaker, G.M. (1990) *A Knowledge Based Approach to the Computation of Network Nodal Survivability*. Proceedings of the IEEE 1990 Military Communications Conference (MILCOM'90), Monterey, CA, pp. 1114–1119.
[80] Schroeder, M.A. and Newport, K.T. (1987) *Tactical Network Survivability through Connectivity Optimization*. Proceedings of the 1987 Military Communications Conference (MILCOM'87), Monterey, CA, Vol. **2**, pp. 590–597.
[81] Whittaker, G.M. (1990) *A Knowledge-Based Design Aid for Survivable Tactical Networks*. Proceedings of the IEEE 1990 Military Communications Conference (MILCOM'90), Monterey, CA, Sect 53.5.
[82] Schroeder, M.A. and Newport, K.T. (1989) *Enhanced Network Survivability Through Balanced Resource Criticality*. Proceedings of the IEEE 1989 Military Communications Conference (MILCOM'89), Boston, MA, Sect 38.4.
[83] Bagga, K.S., Beineke, L.W., Lipman, M.J., and Pippert, R.E. (1988) *The Concept of Leverage in Network Vulnerability*. Conference on Graph Theory, Kalamazoo, MI, pp. 29–39.
[84] Bagga, K.S., Beineke, L.W., Lipman, M.J., and Pippert, R.E. (1988) *Explorations into Graph Vulnerability*. Conference on Graph Theory, Kalamazoo, MI, pp. 143–158.
[85] Bienstock, D. and Gyori, E. (1988) Average distance in graphs with removed elements. *Journal of Graph Theory*, **12**, 375–390.
[86] Boesch, F.T. and Frisch, I.T. (1986) On the smallest disconnecting set in a graph. *IEEE Transactions on Circuit Theory*, **15**, 286–288.
[87] Buckley, F. and Lewinter, M. (1988) A note on graphs with diameter preserving spanning trees. *Journal of Graph Theory*, **12**, 525–528.
[88] Chung, F.R.K. (1988) The average distance and the independence number. *Journal of Graph Theory*, **12**, 229–235.
[89] Exoo, G. (1982) On a measure of communication network vulnerability. *Networks*, **12**, 405–409.
[90] Favaron, O., Kouider, M. and Makeo, M. (1989) Edge-vulnerability and mean distance. *Networks*, **19**, 493–509.
[91] Harary, F., Boesch, F.T. and Kabell, J.A. (1981) Graphs as models of communication network vulnerability: connectivity and persistence. *Networks*, **11**, 57–63.
[92] Haraty, F. (1983) Conditional connectivity. *Networks*, **13**, 347–357.
[93] Lee, S.M. (1988) Design of e-invariant networks. *Congressus Nurner*, **65**, 105–108.
[94] Oellermann, O.R. (1991) Conditional graph connectivity relative to hereditary properties. *Networks*, **21**, 245–255.
[95] Plesnik, J. (1984) On the sum of all distances in a graph or digraph. *Journal of Graph Theory*, **8**, 1–21.
[96] Schoone, A.A., Bodlaender, H.L. and van Leeuwer, J. (1987) Diameter increase caused by edge deletion. *Journal of Graph Theory*, **11**, 409–427.
[97] Ball, M.O., Provan, J.S. and Shier, D.R. (1991) Reliability covering problems. *Networks*, **21**, 345–357.
[98] Caccetta, L. (1984) Vulnerability in communication networks. *Networks*, **14**, 141–146.
[99] Doty, L.L. (1989) Extremal connectivity and vulnerability in graphs. *Networks*, **19**, 73–78.
[100] Ferguson, T.J., Cozzens, J.H., and Cho, C. (1990) *SDI Network Connectivity Optimization*. Proceedings of the IEEE 1990 Military Communications Conference (MILCOM'90), Monterey, CA, Sect 53.1.
[101] Fink, J.F., Jacobson, M.S., Kinch, L.F. and Roberts, J. (1990) The bondage number of a graph. *Discrete Mathematics*, **86**, 47–57.
[102] Gunther, G. (1985) Neighbor-connectedness in regular graphs. *Discrete Applied Mathematics*, **11**, 233–242.
[103] Gunther, G., Hartnell, B.L. and Nowakowski, R. (1987) Neighbor-connected graphs and projective planes. *Networks*, **17**, 241–247.
[104] Hammer, P.L. (1991) Cut-threshold graphs. *Discrete Applied Mathematics*, **30**, 163–179.
[105] Hobbs, A.M. (1989) Computing edge-toughness and fractional arboricity. *Contemporary Mathematics, American Mathematical Society*, **89**, 89–106.
[106] Jiang, T.Z. (1991) *A New Definition on Survivability of Communication Networks*. Proceedings of the IEEE 1991 Military Communications Conference (MJLCOM'91), November, 1991.
[107] Lesniak, L.M. and Pippert, R.E. (1989) On the edge-connectivity vector of a graph. *Networks*, **19**, 667–671.

[108] Miller, Z. and Pritikin, D. (1989) On the separation number of a graph. *Networks*, **19**, 651–666.
[109] Wu, L. and Varshney, P.K. (1990) *On Survivability Measures for Military Networks*. Proceedings of the IEEE 1990 Military Communications Conference (MILCOM'90), Monterey, CA, pp. 1120–1124.
[110] K.-S. Lui, K. Nahrstedt, and S. Chen, Routing With Topology Aggregation in Delay–Bandwidth Sensitive Networks, *IEEE/ACM Transactions On Networking*, Vol. **12**, No. 1, pp.17–29, 2004.
[111] IEEE (1996) Private Network–Network Interface Specification Version 1.0, Mar. 1996, IEEE, Washington, D.C.
[112] Y. Rekhter and T. Li (1995) A Border Gateway Protocol 4 (BGP-4), Network Working Group, RFC 1771, Mar. 1995.
[113] Guerin, R. and Orda, A. (1999) QoS-based routing in networks with inaccurate information: theory and algorithms. *IEEE/ACM Trans. Networking*, **7**, 350–364.
[114] F. Hao and E. W. Zegura (2000) On scalable QoS routing: performance evaluation of topology aggregation, *Proc. IEEE INFOCOM*, 2000, pp. 147–156.
[115] W. Lee (1999) Minimum equivalent subspanner algorithms for topology aggregation in ATM networks, *Proc. 2nd Int. Conf. ATM (ICATM)*, June 1999, pp. 351–359.
[116] Awerbuch, B. and Shavitt, Y. (2001) Topology aggregation for directed graphs. *IEEE/ACM Trans. Networking*, **9**, 82–90.
[117] W. Lee (1995) Spanning tree method for link state aggregation in large communication networks, *Proc. IEEE INFOCOM*, 1995, pp. 297–302.
[118] A. Iwata, H. Suzuki, R. Izmailow, and B. Sengupta (1998) QoS aggregation algorithms in hierarchical ATM networks, in *IEEE Int. Conf. Communications Conf. Rec.*, 1998, pp. 243–248.
[119] T. Korkmaz and M. Krunz, Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks, *ACM Trans. Modeling Comput. Simulation*, vol. **10**, no. 4, pp. 295–325, 2000.
[120] Chen, S. and Nahrstedt, K. (1998) An overview of quality-of-service routing for the next generation high-speed networks: problems and solutions. *IEEE Network*, **12**, 64–79.
[121] Lee, W. (1995) Topology aggregation for hierarchical routing in ATM networks. *ACMSIGCOMM Comput. Commun. Rev*, **25**, 82–92.
[122] L. Guo and I. Matta (1998) On state aggregation for scalable QoS routing, *IEEE Proc. ATM Workshop*, May 1998, pp. 306–314.

3

Mobility Management

3.1 Cellular Networks

As already indicated in Chapter 1 the generic model of 5G/6G wireless networks will integrate services of different segments, such as cellular networks, WLAN, WPAN, and even LEO satellites. Several alternative backbone networks will be used like the public land mobile networks (PLMNs), mobile Internet protocol (Mobile IP) networks, wireless asynchronous transfer mode (WATM) networks, and low Earth orbit (LEO) satellite networks. The backbone network will be most probably organized on the software defined network (SDN) principle. Regardless of the network, one of the most important and challenging problems for wireless communication and computing is mobility management [1–62]. Mobility management enables communication networks to locate roaming terminals for call delivery and to maintain connections as the terminal is moving into a new service area, process known as *handoff.* The handoff may be executed between different segments (cells) of the same or different systems. *The handoff* event is caused by the radio link degradation or initiated by the system that rearranges radio channels in order to avoid congestion. Our focus in this section is on the first kind of handoff, where the cause of handoff is poor radio quality due to a change in the environment or the movement of the wireless terminal. For example, the mobile user might cross cell boundaries and move to an adjacent cell while the call is in process. In this case, the call must be handed off to the neighboring cell in order to provide uninterrupted service to the mobile subscriber. If adjacent cells do not have enough channels to support the handoff, the call is forced to be blocked. In systems where the cell size is relatively small (microcellular systems), the handoff procedure has an important effect on the performance of the system. Here, an important issue is to limit the probability of forced call termination, because from the point of view of a mobile user forced termination of an ongoing call is less desirable than blocking a new call. Therefore, the system must reduce the chances of

Advanced Wireless Networks: Technology and Business Models, Third Edition. Savo Glisic.

Companion website: www.wiley.com/go/glisic0916

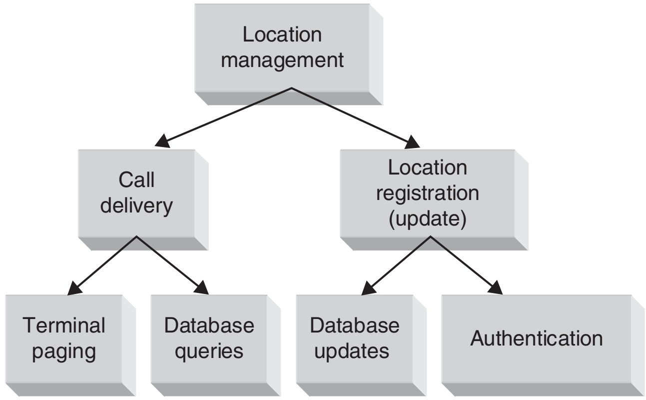


Figure 3.1.1 Components of location management process

unsuccessful handoffs by reserving some channels explicitly for handoff calls. For example, handoff prioritizing schemes are channel assignment strategies that allocate channels to handoff requests more readily than new calls.

Thus, mobility management supports mobile terminals (MTs), allowing users to roam while simultaneously offering them incoming calls and supporting calls in progress. Mobility management consists of location management and handoff management.

Location management: a process that enables the network to discover the current attachment point of the mobile user for call delivery. The main components of the process are shown in Figure 3.1.1.

The first segment is location registration (or location update). In this stage, the MT periodically notifies the network of its new access point (AP), allowing the network to authenticate the user and revise the user's location profile. The second segment is call delivery. Here the network is queried for the user location profile and the current position of the mobile host is found. The main issues in location management involve database architecture design, design of messaging procedures and the transmission of signaling messages between various components of a signaling network. Other issues include: security, dynamic database updates, querying delays, terminal paging methods, and paging delays.

Handoff (*or handover*) *management*: enables the network to maintain a user's connection as the MT continues to move and change its AP to the network. The three-stage process for handoff first involves *initiation,* where either the user, a network agent, or changing network conditions identify the need for handoff. The second stage is new *connection generation,* where the network must find new resources for the handoff connection and perform any additional routing operations. Under network-controlled handoff (NCHO), or mobile-assisted handoff (MAHO), the network generates a new connection, by finding new resources for the handoff and performing any additional routing operations. For mobile-controlled handoff (MCHO), the MT finds the new resources and the network approves. The final stage is *data-flow control,* where the delivery of the data from the old connection path to the new connection path is maintained according to agreed-upon QoS. The segments of handoff management are presented in Figure 3.1.2.

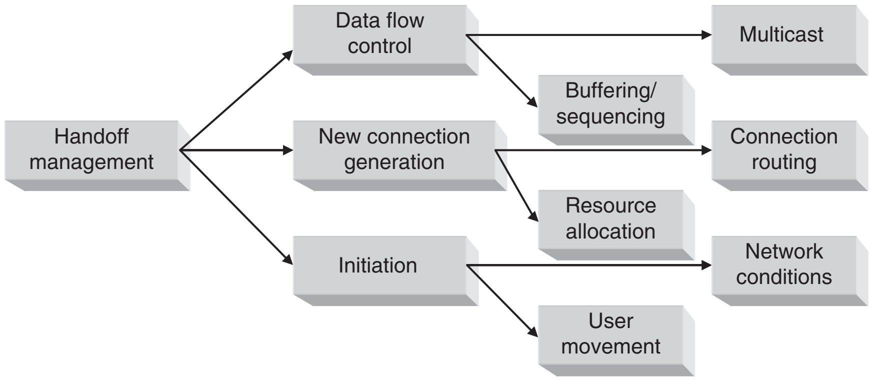


Figure 3.1.2 Components of handoff management

Handoff management includes two conditions: intracell handoff and intercell handoff. Intracell handoff occurs when the user moves within a service area (or cell) and experiences signal strength deterioration below a certain threshold that results in the transfer of the user's calls to new radio channels of appropriate strength at the same base station (BS). Intercell handoff occurs when the user moves into an adjacent cell and all of the terminal's connections must be transferred to a new BS. While performing handoff, the terminal may connect to multiple BSs simultaneously and use some form of signaling diversity to combine the multiple signals. This is called *soft handoff*. On the other hand, if the terminal stays connected to only one BS at a time, clearing the connection with the former BS immediately before or after establishing a connection with the target BS, then the process is referred to as *hard handoff*. Handoff management issues are: efficient and expedient packet processing, minimizing the signaling load on the network, optimizing the route for each connection, efficient bandwidth reassignment, and refining quality of service for wireless connections.

In the sequel we will discuss the handoff management in some of the component networks of 5G integrated wireless network concept as suggested by Figure 3.1.1.

3.1.1 Mobility Management in Cellular Networks

Mobile terminals are free to travel and thus the network AP of an MT changes as it moves around the network coverage area. As a result, the ID of an MT does not implicitly provide the location information of the MT and the call delivery process becomes more complex. The current systems for PLMN location management strategies require each MT to register its location with the network periodically. In order to perform the registration, update, and call delivery operations described above, the network stores the location information of each MT in the location databases. Then the information can be retrieved for call delivery.

Current schemes for PLMN location management are based on a two-level data hierarchy such that two types of network location database, the home location register (HLR) and the visitor location register (VLR), are involved in tracking an MT. In general, there is an HLR