8

Energy-Efficient Operation and Management for Mobile Networks

Zhisheng Niu and Sheng Zhou
Department of Electronic Engineering, Tsinghua University, Beijing, China

8.1 Principles

8.1.1 NM Should Be in a Holistic Manner

The continuously growing demands for ubiquitous and broadband access to the Internet brings the explosive development of information and communication technology (ICT) industry, which has become one of the major sources (responsible for 2–10%) of worldwide energy consumption and is expected to increase further in the future. In the meantime, we have witnessed a consistent increase in the number of mobile terminals, especially in the coming era of Internet of Things, which has triggered more complex and higher energy-consumed signal processing technologies. According to the portfolio analysis of the total energy consumption in a typical mobile network[1], it is reported that nearly 75% comes from the base station (BS) side and, inside a BS, nearly 70% of energy is consumed by baseband processing, power amplifiers, and air conditioners in order to keep the BS working (i.e., providing the coverage) even though there is no any traffic in the cell. Therefore, only through the reduction of transmitting power does not help too much for the total energy savings, and so are the incremental approaches such as slim base stations or smart cooling technologies. A more ambitious and system-wide solution is expected if some lightly loaded BSs can be turned into sleep mode or completely

[1] https://www.ict-earth.eu/default.html

Green Communications: Principles, Concepts and Practice, First Edition.
Edited by Konstantinos Samdanis, Peter Rost, Andreas Maeder, Michela Meo and Christos Verikoukis.

switched off so that the corresponding power amplifiers and air conditioners can also be shut down during that time.

In the contrary, the existing wireless networks are usually dimensioned for performance optimization without enough consideration of energy efficiency. Specifically, the so-called worst-case network planning philosophy has been widely adopted in order to provide quality-of-services (QoS) guarantee even during the period with peak traffic. As a result, during low traffic periods such as nights or holidays or in some sparse spots where the traffic load is temporarily getting very low due to the user mobility, many BSs are under-utilized but still, by being active, consume a great amount of power. Considering the fact that the nonworking time (including holidays and nighttime) is in fact more than half of the year, the wasted energy of the existing cellular networks is remarkable. This is even more severe for future mobile communication networks where the size of cells will be getting smaller and smaller (e.g., micro- or picocellular) in order to accommodate more high data rate users and increase the frequency reuse factor, which will further increase the dynamics of the traffic in a specific cell. Therefore, it will be very important to have the transmitting power (and, therefore, energy consumed) of network nodes adapt to the traffic variation, including completely switching off some BSs when the traffic load is lower than a threshold.

Furthermore, the dominant traffic in wireless networks has been shifting from mobile voice to mobile data and further to mobile video in the future [1]. This transition is in fact one of the key drivers of the evolution to new mobile broadband standards like 3G, WiMAX, LTE, and LTE-Advanced, resulting in the coexistence of macro-, micro-, pico-, and femtocells, that is, heterogeneous cellular networks. That is to say the scarce spectrum resources have to be segmented into many independent pieces and each cellular network has to provide full coverage to their corresponding users by itself. Such a redundant deployment will definitely waste the spectrum as well as energy resources furthermore. In addition, with the development of mobile data and video, it is predicted that the traffic volume of mobile services in 2015 could be as much as 100–1000 times of the existing ones, and among which two-third (2/3) could be mobile video traffic. As the spectrum in wireless networks is limited in nature, this will lead to the widespread use of complex channel coding and modulation techniques for advanced interference mitigation. However, using such techniques typically implies accepting higher power consumption not only from the transceiver but also from the complete radio access network. As a result, the contribution of wireless networks to the global carbon footprint is forecast to double over the next ten (10) years.

To deal with this challenge, the traditional physical- and MAC-layer capacity-enhancement approaches are no more sufficient and efficient. A system- or network-level approach is needed, including rethinking about the existing cellular structure. Specifically, the network management in the future should be able to optimize the usage of the scarce spectrum segmented into heterogeneous networks (e.g., 2G, 3G, 4G, WiFi) in a global way and, meanwhile, shift the optimization goal from spectrum efficiency to energy efficiency, i.e., GREEN. Alternatively speaking, green communications should not simply mean lowering down the transmitting power or improving the energy efficiency of a transmission link, but a holistic approach from the whole network scope. By introducing more collaborations among the neighboring nodes in a wireless network as well as among the heterogeneous networks, the radio resources that are segmented into different nodes and networks can be more efficiently utilized.

8.1.2 NM Should Involve More Cognition and Collaboration

As shown earlier subsection, to overcome the scarcity of wireless spectrum and meet the ever-growing user demand, convergence of multiple wireless networks is a promising way to efficiently utilize wireless resources. Rather than the costly "clean slate" redesign of a one-fit-all wireless system, it is more practical to realize collaboration across these independently developed wireless systems. As a result, many academic entities and international standard organizations have been improving the spectrum efficiency of single-radio access networks with collaboration technology. In other words, the radio resources (even though it is licensed) should be used in a more open and collaborative way so that a harmonized radio ubiquitous system could be realized. As a result, the network managements in the heterogeneous radio systems should CHORUS together rather than sing independently. Otherwise, extra energy will be needed to combat with the noises and/or interferences.

For instance, the seminal work [2] has suggested to use other nodes to relay the signal of a dedicated transmission. The additional diversity can reduce transmission power and increase data rate. Through another route, multi-base station (Multi-BS) collaboration technology [3] is proposed by designing downlink signals collaboratively across BSs, where signals from other cells are used to assist transmission instead of acting as interference, and thus the spatial degrees of freedom (DoFs) are fully utilized. However, one must face excessive signaling and backhaul overhead to collect and exchange channel state information (CSI), user data, and synchronization information among multicells. Therefore, collaboration schemes with limited scale are proposed [4] as compromises.

Research efforts have also triggered the standardization of the collaboration techniques, such as CoMP (coordinated multipoint processing) and Relaying in 3GPP [5]. Nevertheless, their performance in large network is unknown, and small-scale collaboration schemes still suffer from boundary effects, that is, performance of users at the edge of collaboration regions will degrade. Scalable approaches applicable for large systems are valuable and yet to be discovered. Moreover, the benefits of seamless interworking of heterogeneous radio access networks are inherent because of the diverse advantages of heterogeneous networks. Network selection can be triggered by the user terminals through vertical handover, or admission control managed by the network side [6]. People also consider implementing new types of BSs with different coverage capabilities, that is, femtocell and picocell, to bring heterogeneity into single radio networks [7], and this improves coverage performance, interference suppression, and radio resource allocation flexibility. For heterogeneous networks, current studies mainly provide access flexibility, while ultimately it is desired to optimally choose the radio network for each packet so that the system resource is efficiently managed in a finer granularity. The challenges of promoting this kind of "smooth" networking lay in the optimal design in both the vertical (protocol) and the horizontal (heterogeneous cells) directions, and this requires a large amount of information cognition and exchange.

8.1.3 NM Should Be More Adaptive to Traffic Variations

As shown in Figure 8.1 [8], the traffic in a cellular network is typically unbalanced, changing not only in time domain but also in spatial domain. Generally speaking, holiday or weekend

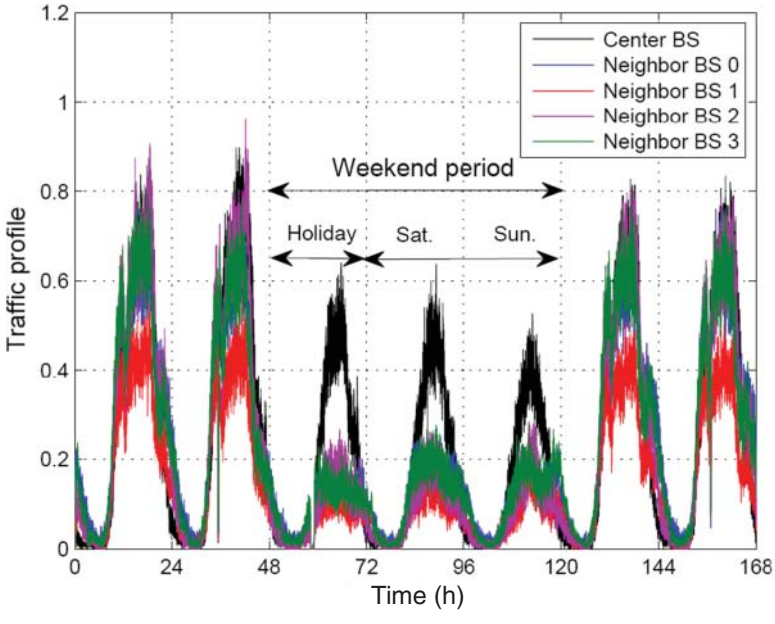

Figure 8.1 Traffic dynamics both in time domain and spatial domain

traffic is lower than weekdays and nighttime traffic is much lower than the daytime. During the daytime, the so-called peak traffic period is only a small portion of the whole day. On the other hand, the traffic in different regions may also be very different due to the user mobility and bursty nature of data and video applications. For example, the business areas may be very heavily loaded during the daytime but lightly loaded during the nighttimes, which leads to unbalanced traffic load among neighboring BSs even during the daytime. Therefore, if the capacity is planned based on the peak traffic load for each cell, there will always be some cells under light load, while others are under heavy load. In this case, any static cell deployment will not be optimal as traffic load fluctuates. Such kind of unbalanced traffic distribution can be even more serious as the next-generation cellular networks move toward smaller cells such as microcells, picocells, and femtocells.

Another trend of mobile traffic is that mobile data and video will dominate the whole networks [1]. On one hand, compared with voice traffic, data and video traffic are typically more bursty and dynamic and, therefore, consume more spectrum and energy resources. On the other hand, they can tolerate some delay in general and, furthermore, are more in point-to-multipoint fashion (many people may be interested in the same content in a short time period) rather than just point-to-point communication mode. As a result, it will not be energy-efficient to provide mobile data and video services in a real-time and point-to-point way as for voice traffic.

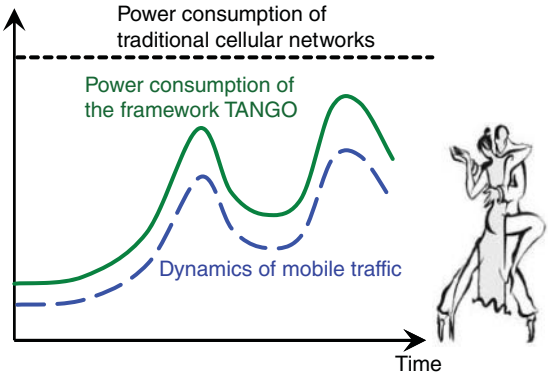

Figure 8.2 Conceptual figure of the framework TANGO

In short, the traffic dynamics can in fact provide some opportunities for energy savings. As shown in Figure 8.2, if we can trace the traffic variation and make the network resources (including transmitting power and other equipment's power) in a cell or the whole cellular network adapted to it, a great amount of energy could be saved. Also, if we can know a priori that many users are requesting the same data or video content from the server, a multicast session could be initiated accordingly so that the transmission energy could be saved by transmitting the content only in one time. This is to say the network panning and operation in green cellular networks should be more traffic-aware, i.e., TANGO (Traffic-aware Network planning and Green Operation). This just looks like a gentleman (traffic needs) and a lady (power and other radio resources) are dancing together in a harmonious way. In reality, more and more BSs have been equipping the self-organizing functionality or even with sleep mode. For instance, Verizon has started a Technical Purchasing Requirement (TPR) guideline by applying the Telecommunication Equipment Energy-Efficiency Rating (TEEER) methodology to all the network components since 2009. Alternatively speaking, if the network components cannot meet the TEEER criteria, they will not be purchased no matter how cheap they are and how excellent the performance is.

8.2 Architectures

8.2.1 Paradigm Shift to CHORUS

The most existing works for either inter or intrasystem collaboration are only applicable to limited scale, or merely provide access flexibility among heterogeneous networks. Generic solutions to fully exploit the features of different radio technologies in accordance with the channel and traffic variations are desired. However, realizing such scalable collaboration with fine resource granularity faces following challenges. First, collecting a large amount of system

information across network entities and heterogeneous networks is difficult, not only due to the huge mass, but also different system architectures. Second, it is hard to find the collaboration opportunities, which also tightly relates to how the system information is collected. Last but not least, as different radio networks are independently designed, their resources, such as power, frequency, time, and space, may have different forms and thus their control interfaces also vary.

One solution to solve above challenges is to incorporate smart cognition methods, and jointly operate it with node collaboration. The concept of cognition is originally applied for cognitive radio [9], built on software-defined radio. Cognitive radio is mainly targeted at sensing the spectrum hole to find transmission opportunities [10] for unlicensed users, and then according to the CSI and interference constraints, adapts the transmit power with spectrum management mechanism to reconfigure the transceivers. In the context of cognitive radio, cognition methods are exploited to learn the channel and interference conditions, and thus various spectrum sensing technologies are proposed [10]. Moreover, the cognition behavior is also characterized from information theoretical perspectives [11] as side information that comprises knowledge of transceiver activities, channels, codebooks, and messages of other transceivers that share the spectrum.

Inspired by these applications, we *broaden* the cognition targets from spectrum and channels in cognitive radio to the whole heterogeneous networks, crossing physical-layer resources and upper-layer traffics. In our design, the core enabling method to realize scalable network collaboration that harmonizes different wireless resources is *cognitive synergy*. Based on the philosophy of smart cognition, as what is generally required in cognitive radio, *distributed* inspection of the system state across multiple layers and *self* organization/control of network entities can be realized with low overhead, which is the key to large scale networking. Moreover, cognitive synergy provides unified virtualization of the resources from heterogeneous networks, which is valuable for traffic offloading and collaborative communications among heterogeneous networks. We also make cognition interactions with collaboration so that

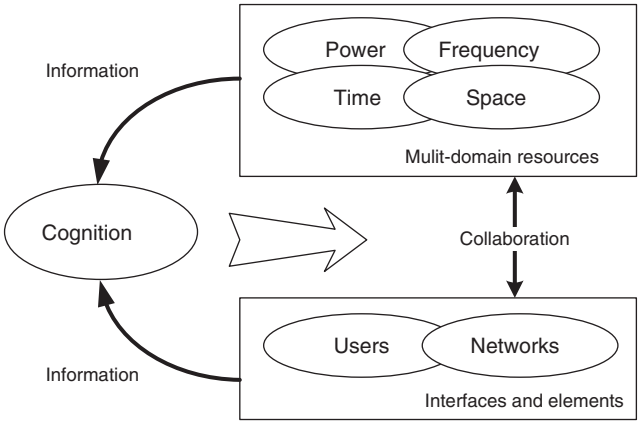

Figure 8.3 Our vision on the new paradigm of a ubiquitous radio network

node collaboration assists cognition, and is also reconfigured according to the cognition results.

Our basic vision on the new paradigm of a ubiquitous radio network is depicted in Figure 8.3, which is explicitly embodied with the framework design CHORUS. All the features shown in Figure 8.3 are reflected in the framework, and our focus is on how to achieve the optimal collaboration gain and environmental friendliness by cognitive synergy in ubiquitous radio network environments. To overcome the difficulties of collaborating transceivers and networks over large scale, cognitive synergy is exploited to inspect the status of multiple radios and layers. Based on the cognition result, harmonic collaboration control is applied to optimize the global performance. The CHORUS framework is designed to work with all available wireless access networks, such as LTE/LTE-A, 3G, WiMAX, WLAN, and so on. This enables CHORUS to make maximum coverage and synthetically enroll various permitted services.

8.2.1.1 Architecture of CHORUS

The architecture of CHORUS is presented in Figure 8.4. The core is a CHORUS server, which controls the procedure of CHORUS. The CHORUS server is a conceptual body in the network and can be either implemented in the gateway, such as radio network controller (RNC), or distributed in the BSs, or in a mixed way. Note that the way of implementation also indicates that the algorithms of CHORUS are realized in a centralized or distributed way. The CHORUS server has two engines. One is the **cognitive engine**, and it communicates with the entities in the wireless access network and collects information about the environment and users. It also analyzes this information, and stores the results in the other engine: **profile database**. The profile database stores the processed information, provided by the cognitive engine, about user terminals, access networks, wireless resources, and environment. In addition, the cognitive engine consists of the **data analyzer** and **cognition controller**. The data analyzer reprocesses the system information based on its raw version gathered from the network side, of which the results are utilized by the cognition controller for making network control decisions.

The network entities sense the environment, such as channel conditions, spectrum occupancy, and so on. They also interact with user terminals, collect their QoS requirements, and provide admission control and service regulations. The collected information will be first locally saved at the network entities. Then these information will be collected by the **cognitive engine** assisted by network node collaboration, which reduces the overhead of collecting the large amount of system information. On the other information route, the profile database delivers back the cognition results for collaboration control and resource reconfiguration to the network side. The contents delivered to the network also include how the network should sense the environment and get the user status more efficiently. The detailed operation flow of CHORUS, especially the cognition part, is presented as follows.

8.2.1.2 Work Flow of CHORUS

An exemplary working flow of CHORUS is shown in Figure 8.5. The core procedure of CHORUS operation is synergetic cognition, which guarantees scalable collaborative

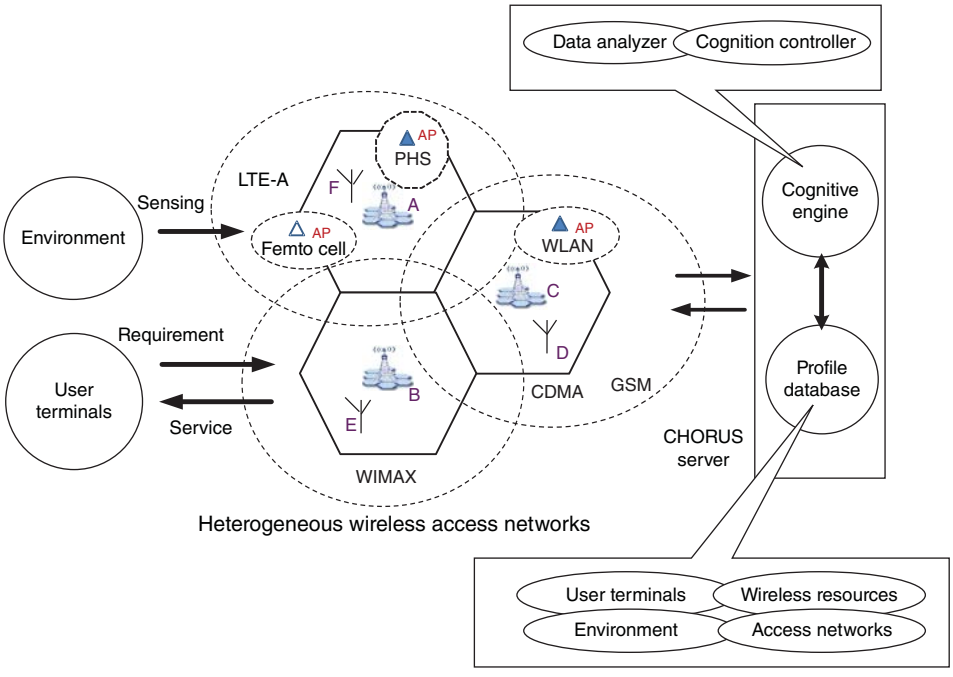

Figure 8.4 The architecture of CHORUS framework

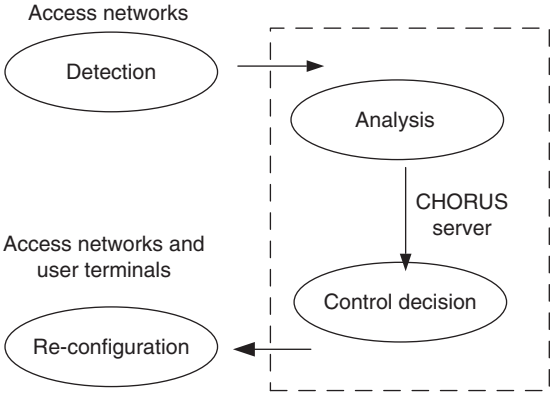

Figure 8.5 Example of CHORUS operation procedures

communications. There are four major steps, that is, **Detection**, **Analysis**, **Decision** and **Reconfiguration**.

- **Detection** is performed by access network entities to get necessary data and information from user terminals or environment, for corresponding cognition. same as the way CHORUS server is implemented, the detection can be locally accomplished by the network elements, or by a central controller with feedback from network elements. The single-link detection aims to adapt the transmission strategy according to channel conditions and spectrum usages, for instance, adaptive modulation and coding. Multiple-link detection is to gather information from interactive links in network, and the information consists of network attributes and parameters that affect the user experience. For heterogeneous networks, the resource occupancy across different networks is detected to provide resource virtualization of networks. Note that in the process of **Detection**, network entities also *collaborate* to provide their information for data analysis efficiently.
- After sufficient information is ready, data **Analysis** is executed by data analyzer based on the profile data saved in the profile database. The data analyzer uses certain techniques, such as data mining and learning algorithms, to find valuable information from the raw version collected from **Detection**. It also investigates the equivalent resource consumption of certain operations for heterogeneous networks. Network entities can also *collaborate* to estimate state information or predict the impact of possible network operations to assist data analysis. The data analyzer also in turn updates the profile database with the new cognition results.
- Afterward, the instructions of network **Control** and **Reconfiguration** decisions are made by cognition controller and sent to networks and user terminals. The on-demand collaborative control is harmonized with the system state. Over time, the cognition controller can refine which information of the system is most valuable for collaboration so that the **Detection** is optimized. This closed-loop for detection and cognition is vital for scalable information collection.
- Based on the decisions made by cognition controller, the network **Reconfiguration** guarantees the efficient collaboration form that fits mostly to the current status of wireless resources and user requirements.

8.2.1.3 Relationship between Cognition and Collaboration

In CHORUS, the unique feature is that cognition methods interact with collaboration. Network entities collaborate to investigate and collect system information, after which the cognition results will help reconfigure the collaboration scale, way of collaboration, andinformation required to perform collaboration, and so on. In other words, collaboration is tied with all the major steps of cognition procedure in CHORUS. This is also the reason why "cognition" appears jointly with "synergy" as the core idea of CHORUS. With node collaboration, CHORUS allows distributed resource control schemes, and the information flow of CHORUS can also have different styles. For example, as will be shown in the next section, *distributed negotiation* for dynamic collaboration is an effective way of transferring information and making joint control decisions. Inherently, due to the distributed feature, the overhead of information exchange is reduced, and it is important to scalable networking.

8.2.2 Paradigm Shift to TANGO

8.2.2.1 Adjusting the Working Mode of Base Stations

As shown in Figure 8.2, the traditional network planning and operation is mainly based upon the assumption that user requests may happen *anytime* and at *anyplace*. This is in fact also the dream that people are having about the mobile communications. As a result, the existing cellular networks were mostly designed to keep the transmitting power always-on in order to guarantee the cell coverage as well as provide the appropriate services if the request happens. This is clearly not energy-efficient because the user requests occur only *sometime* and *someplace* in practical situation. It is, therefore, reasonable to keep the cell coverage by a minimum number of BSs and then adjust the working mode (active or sleep) of the remaining BSs in accordance with the traffic variations. This is equivalent to adjusting the BS density of the cellular networks or, in other words, the network resources should only be provided on demand whenever there is such a need. As a result, the network architecture should be flexible enough to this adaptation. Apparently, this paradigm shift has a great potential for the energy saving in cellular networks. Of course, when some BSs are switched off or in the sleep mode, radio coverage and service provisioning are taken care of by the devices that remain active, i.e., BS cooperation is crucial.

BS sleeping is drawing more and more attention in recent years. Reference [12] gives a static BS sleep pattern according to a deterministic traffic variation pattern over time. However, neither the randomness nor the spatial variation of the traffic is considered. Reference [13] proposes a resource on-demand (RoD) strategy for high-density centralized WLANs, where a cluster-head AP takes care of the whole coverage in the cluster so that other APs in the cluster can be switched off when the traffic load is low. However, the channel model of WLANs is quite different from that of cellular networks where path-loss effect is dominating. Therefore, dynamic clustering algorithm by considering the BS collaboration is needed.

8.2.2.2 Adjusting the Cell Size

Cell size in cellular networks is in general fixed based on the estimated traffic load. However, as discussed earlier, the traffic load can have significant spatial and temporal fluctuations and, therefore, keeping the cell size fixed is not energy efficient. In other words, the cell size should be adjusted dynamically according to traffic conditions as well as the situation of neighboring BSs in a collaborative way. This is the concept of the so-called *cell zooming* proposed in Ref. [14], where two typical zooming patterns by increasing the transmitting power of those BSs that remain active and the corresponding cell planning results are shown. Unlike the power control on link layer, which does not actually change the cell size, cell zooming is a technique on network layer that changes the cell size by adjusting the transmit power of control signals. The detailed description of the cell zooming technique is shown in the following section.

8.2.2.3 Adjusting the Service Mechanism

As mentioned earlier, mobile data and video traffic will dominate the future networks. Unlike the voice traffic, which is typically delay-sensitive and symmetric in uplink and downlink, data and video traffic is in general loss-sensitive and asymmetry in uplink and downlink. But

majority of the existing cellular networks were designed to accommodate voice traffic mainly, i.e., if the capacity of the networks is not enough, they try to increase the capacity anyway (and, therefore, consume more power) or simply reject the requests (and, therefore, deteriorate the QoS). In other words, the existing cellular networks are not so friendly to data and video traffic. As the IP-based data and video traffic can tolerate some delay in nature, they can be served in an opportunistic way in fact. Specifically, they can be served during the period when the channels are in good condition or the network is lightly loaded. For some data and video contents that many users are requesting in a short period, they can in fact be served by using multicast or broadcast [15] so that the same contents are not necessarily transmitted multiple times. Rather, they can be cached in between [16] so that the users who have the same request can get the service locally without going to the source node every time. In such a way by adjusting the service mechanism properly, the energy can be saved dramatically. Consequently, the cellular network architecture should be flexible enough to provide such differentiated services in an appropriate manner.

8.3 Implementation Examples

8.3.1 CHORUS by Scalable Collaboration

As previously mentioned, the idea of cognition in CHORUS is generalized by inspecting status and parameters of multiple scales. The cognition method is also jointly optimized with network collaboration, which enables efficient system resource utilization for information collection and exchange. This is beneficial but the most crucial barrier for network collaboration for both homogeneous and heterogeneous networks is the mass information exchange. In what follows, we firstly show the benefits of exploiting *cognitive synergy* in CHORUS through two case studies that realize **Scalable Collaboration** and **Ubiquitous Access**, correspondingly. Then, some implementation issues will be discussed.

8.3.1.1 A Decentralized BS Dynamic Clustering Scheme

Enabled by smart cognition, the network can harmonize the whole process of collaboration with two key capabilities: reducing the overhead to collect necessary information for collaborative communication and adapting the appropriate form and scale of collaboration. The first one is reflected in the **Detection** and **Analysis** steps, while the second is enabled through the **Decision** and **Reconfiguration** steps. We illustrate how these two capabilities are realized with our research on distributed dynamic BS clustering [17] and dynamic BS sleeping [18] for cellular systems. Specifically, we will show how to exploit cognitive synergy to realize scalable collaboration with high spectrum efficiency and reduced energy consumption.

Implementing collaborative communication is in fact a trade-off between performance gain and resource consumption for information sharing. Taking BS collaboration for example, by designing downlink signals cooperatively among BSs, signals from other cells are used to assist transmission instead of acting as interference, and thus BS collaboration can substantially increase the spectrum efficiency of the cellular network [17]. However, due to practical constraints like synchronization, and backhauling, only a limited number of BSs are allowed to collaborate [17], where the collaborating BSs formulate a BS *cluster*, and the whole network is

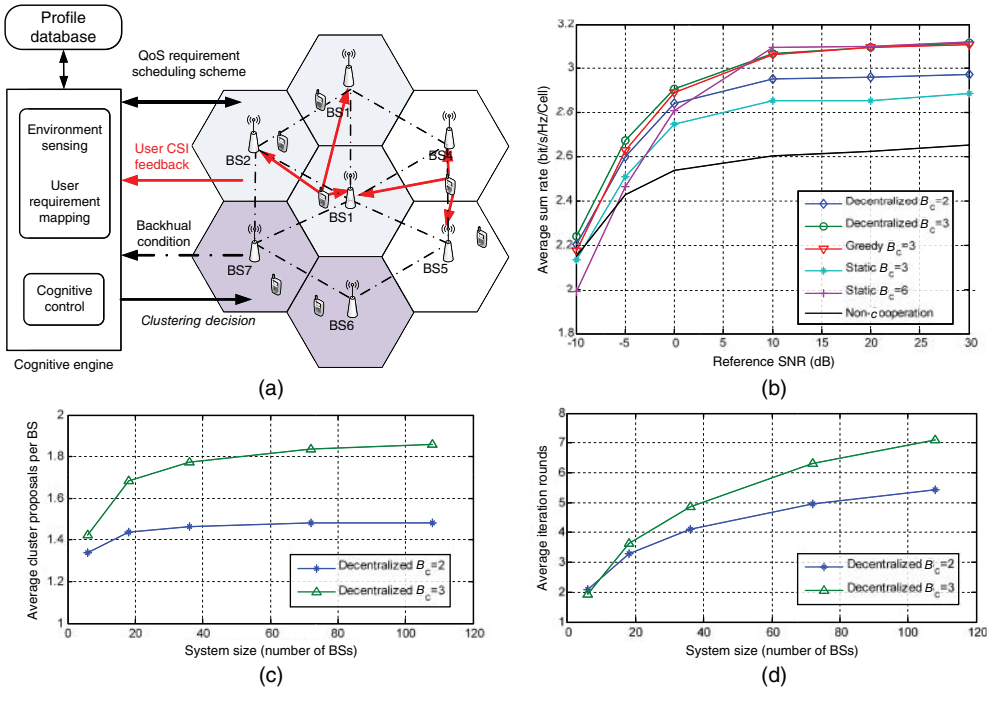

Figure 8.6 CHORUS application for BS collaboration, where B_c is the cluster size. (a) Architecture of dynamic BS clustering under CHORUS framework. (b) Average sum-rate with dynamic BS clustering. (c) Feedback overhead. (d) Calculation complexity. (simulation figures from Ref. [17])

divided into disjointing clusters. Even with fixed BS clustering, users at cluster edge still suffer from severe inter-cluster interference. Therefore, adaptively tuning the BS clustering structure is more preferable. In fact, dynamic clustering itself is challenging as cluster formation relies on scheduling decisions, precision and scale of user CSI feedback, user distribution, and so on, which in turn incur extra overhead. In our work, CHORUS reduces these overhead and provides suitable BS clustering formation structures.

The architecture of BS collaboration with the assistance of CHORUS is shown as Figure 8.6, where the BSs are separated into three difference clusters. The conceptual elements in Figure 8.6 have the following realizations. The cognitive engine is implemented in each BS and according to the CSI and user QoS requirements, for the **Detection** step it abstracts the preferable choices of clustering companions of each BS b into a preference function $\hat{R}(b, c)$, which represents the expected rate of the associated users if b joins cluster c. Then the value of a cluster c, denoted by $V(c)$, is defined as the sum of $\hat{R}(b, c)$ from each component BS. The unique feature of this piece of work is that the cognitive **Analysis** and **Control Decision** are merged in the form of distributed BS negotiation about their preference functions and cluster values. BS clustering can be accomplished with low complexity and in a *distributed* way. Detailed algorithm can be found in Ref. [17]. The sum-rate performance is shown in Figure 8.6. It is observed that BS collaboration enjoys significant throughput gain over single BS transmission, and dynamic clustering scheme outperforms static clustering

substantially. Moreover, from Figure 8.6, one can see that both the feedback and calculation overhead scale very slowly with the network size: The feedback overhead is almost irrelevant to the network size when the network size is large enough, and the calculation overhead scales approximately logarithmically with the network size, while the complexity of conventional centralized greedy algorithm scales quadratically with the network size. Therefore, CHORUS shows its ability of maintaining the scalability of network collaboration. Furthermore, with cognition synergy, scalable CSI feedback optimization is investigated in Ref. [19], where we combine the dynamic clustering with the consideration of limited feedback bits. We design a feedback set adaptation scheme by the cognition of most valuable BS CSI feedback bits, and successfully achieve a good trade-off between collaboration gain and quantization precision, which further reduces collaboration overhead.

BS collaboration can also be used to save energy. Reducing the energy consumption has become one of the key features of future network design, and it has been shown that BSs cost most of the energy of the access network [8]. Therefore, switching on and off BSs *on demand* can be more efficient. In accordance with BS sleeping, cell sizes of active BSs should also be tuned to guarantee the network coverage, namely cell zooming [14]. Both of above highly rely on the precise cognition of the network temporal–spatial traffic conditions. BS sleeping requires BS collaboration so that users in sleeping cells are taken care of by the active BSs. In addition, taking signaling overhead, device lifetime and switching energy consumption into account, frequent BS mode switching should be avoided.

In CHORUS, we combine the traffic cognition and the BS sleep control. As shown in Figure 8.7, they are implemented in the cell zooming server. For cognition operation, here the neighboring BSs negotiate to exchange traffic information and working state, and then collaborate to determine the optimal BS working mode. The neighboring BS negotiation based **Detection** and **Analysis** steps greatly reduce the complexity and the cost for working mode switch: The full consideration of all network states has the exponential complexity with the network size, while with CHORUS, we are able to reduce it to linear information scaling [18]. Next, the **Control Decision** step is realized as a dynamic programming with the per-stage cost

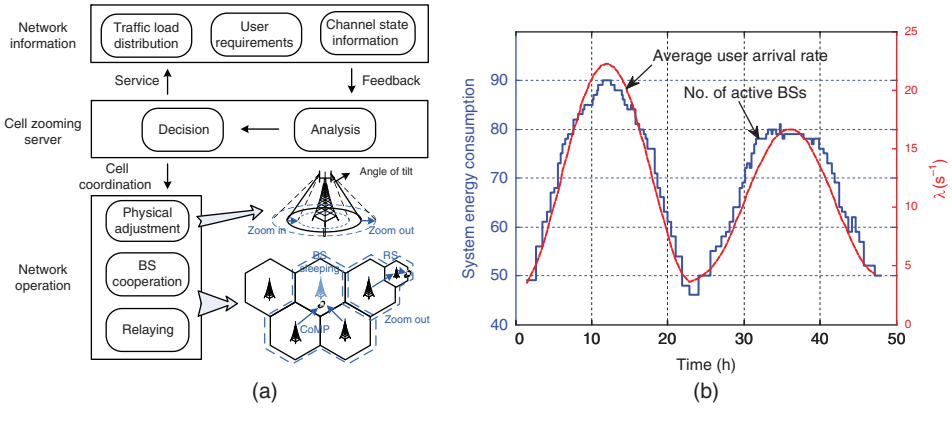

Figure 8.7 CHORUS application for smart dynamic BS energy saving. (a) Architecture of cell zooming under CHORUS framework. (b) Number of active BSs compared with average traffic intensity in time space with different traffic configurations

at stage i

$$g_i = \sum_{m=1}^{M} [|s_i^{(m)} - u_i^{(m)}|E_i + C_s u_i^{(m)} + h(\tilde{P}_i^{(m)})] \tag{8.1}$$

where $s_i^{(m)} \in \{0, 1\}$ is the BS working state, $u_i^{(m)} \in \{0, 1\}$ is the action of BS m, and $\tilde{P}_i^{(m)}$ is the blocking probability of a new call arrived in cell m. So, the per-stage cost is a combination of operation energy $s_{i+1}^{(m)} E_i$, switching cost $C_s u_i^{(m)}$, and blocking probability penalty $h(\tilde{P}_i^{(m)})$ of all the cells. The object is to find the optimal policy that minimizes the total cost of all stages

$$\{u_{i,\,\mathrm{opt}}^{(m)}\}_{i=0,\ldots,N-1}^{m=1,\ldots M} = \arg \min_{\{u_i^{(m)}\}_{i=0,\ldots,N-1}^{m=1,\ldots M}} \sum_{i=0}^{N-1} g_i \tag{8.2}$$

The optimal decision of each BS is approximated as a function of local cognition results of the traffic in its own coverage as well as its first-tier neighboring BSs. An iterative algorithm is proposed to find the suboptimal decisions. The network energy consumption, represented by the number of active BSs, well matches the traffic intensity as shown in Figure 8.7.

8.3.1.2 A Ubiquitous Heterogeneous Radio Access Scheme

Scalable collaboration can happen not only intra-system, but more importantly in an inter-system way, of which ubiquitous access is one of the key functions. Ubiquitous is identified and articulated as a new computing paradigm, where the network is connected at any place, any time, and with any object. Current user equipments are capable of reconfiguration to access multiple networks with different protocols. While at the network side, the independent design hinders the ubiquitous access. The implementation of such network highly depends on the *virtualization* of the network resources and dynamic reconfiguration/reorganization of terminals and networks. When the collaboration happens between different networks, traffic can be split and conveyed over heterogeneous networks smoothly. In this section, we show how the **Detection**, **Analysis**, and **Decision** steps in CHORUS support the **Reconfiguration** of existing networks with minimum cost and with fine time granularity so that ubiquitous interconnection is realized with optimal radio, power, time-slot allocation at any time, any place. With the unified internetwork interface and protocol, it is flexible to design subnetwork satisfying new application demand and joining in CHORUS system.

One of CHORUS applications in heterogeneous network is the integrated communication and broadcast networks (ICBN) architecture [15]. The key issue of ICBN is how to combine advantages of the two types of radio networks so that an efficient provision of high-quality multimedia services is ensured. The cognitive engine identifies the states of both networks, and the equivalent resource occupancy of delivering certain amount of information bits from any of the two networks, it then intelligently delivers downlink data transmission through broadcast network or retransmission through communication network. One can thus achieve higher spectrum efficiency and thus system capacity.

In ICBN, contents can be distributed to different transmission nodes closer to users. The contents are firstly cached at these nodes before an appropriate choice of the time to broadcast/mulitcase contents to users within its coverage. For example, when relay stations (RSs) are introduced, caching with multicast can also help reduce the transmission cost, when the need for the same content of different users spreads over time [16]. In some content delivery

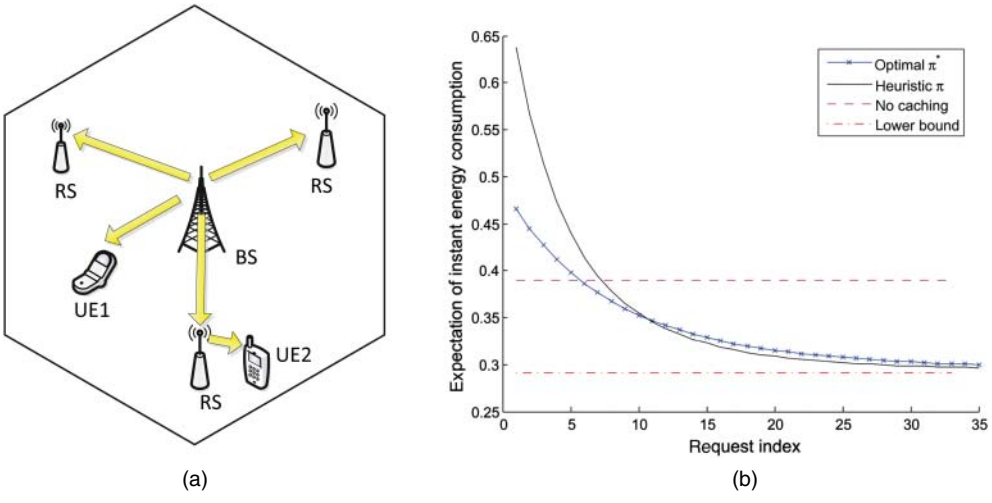

Figure 8.8 CHORUS application for smart caching assisted by relays. (a) Relay caching scheme in cellular network. (b) Cumulative average of transmission energy consumption, where we compare the optimal policy and a proposed heuristic with the baseline policy without relay caching, and the energy consumption lower bound to any policy is achieved when all the necessary contents are pre-stored in the relay

services such as video-on-demand service, multiple requests of the same content induce abundance of insignificant traffic if the delivery is naive unicast. However, as users may appear or request the content at different time, introducing relay caching can replace costy repeating re-broadcast with low power short distance transmission. A single cell in a cellular network with RSs is depicted as Figure 8.8. When a user requests a piece of content, the BS may transmit the content to the user directly or may broadcast the content to both the RSs and the UE. Contents can be cached into the buffer of RSs in order to serve the nearby users with future requests of the same content. When a piece of content that has already been cached in an RS is requested by a user in the RS's coverage, it can be transmitted from the RS to the user directly.

For this kind of dynamic caching, the key problem is to identify whether or not to cache the requested content in the RS, as the buffer size of RSs is limited. The probability of a piece of content being requested, that is, the popularity, is introduced to **Analyze** the content. The popularities of contents in the buffer are stored in the profile database, and the cognitive engine makes the **Control Decision** to determine whether to cache a newly requested content or not. As the system runs, the cognitive engine learns and **Reconfigures** the optimal caching policy through stochastic dynamic programming. The average energy consumptions of transmission with the optimal policy and the derived heuristic policies are shown in Figure 8.8. Energy consumption is saved by 15.3% after the caching period.

8.3.2 TANGO by Cell Zooming

In this section, the concept of cell zooming is introduced, which is to adaptively adjust the cell size according to the traffic load fluctuations. Cell zooming can not only solve the problem

of traffic imbalance, but also reduce the energy consumption in cellular networks. Techniques such as physical adjustments, BS cooperation, and relaying can be used to implement cell zooming. Through the numerical examples, we show that the proposed cell zooming algorithms can leverage the trade-off between energy saving and blocking probability. The algorithms also save a large amount of energy when traffic load is light, which can achieve the purpose of green cellular network in a cost-efficient way.

8.3.2.1 Concept and Challenges

Cell size in cellular networks is in general fixed based on the estimated traffic load. However, as mentioned earlier, the traffic load can have significant spatial and temporal fluctuations, which bring both challenges and opportunities to the planning and operating of cellular networks. This subsection introduces a concept of *cell zooming*, which adaptively adjusts the cell size according to traffic load, user requirements, and channel conditions. The implementation issues of cell zooming are then presented. Finally, a use case of cell zooming for energy saving is investigated. Centralized and distributed cell zooming algorithms are developed, and simulation results show that the proposed algorithms can greatly reduce the energy consumption, which leads to green cellular networks.

An example of cell zooming is illustrated in Figure 8.9. It is a cellular network with five cells. One central cell is surrounded by four neighboring cells. BSs are located at the respective center of the cells, denoted by hollow squares; MUs are randomly distributed in the cells, denoted

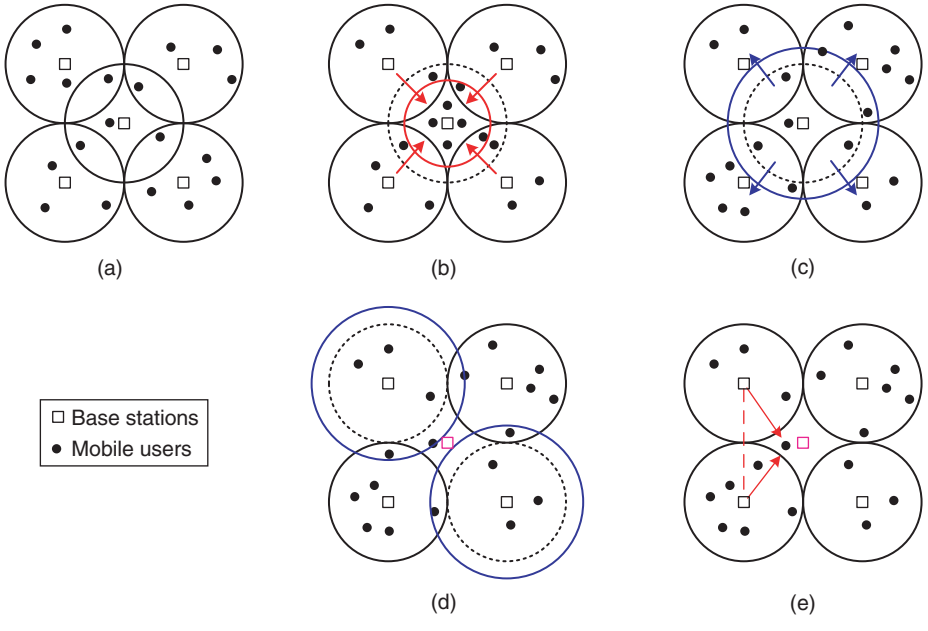

Figure 8.9 Cell zooming operations in cellular networks: (a) Cells with original size; (b) Central cell zooms in when load increases; (c) Central cell zooms out when load decreases; (d) Central cell sleeps and neighboring cells zoom out; (e) Central cell sleeps and neighboring cells transmit cooperatively

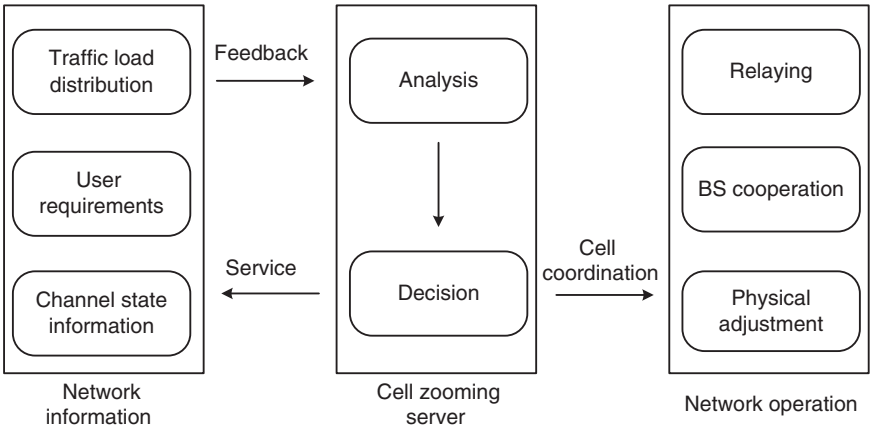

Figure 8.10 Framework of cell zooming

by solid dots. When some MUs move into the central cell and make it congested, the central cell can zoom in to reduce the cell size and, therefore, release from the congestion (Figure 8.9(b)). On the contrary, if some MUs move out of the central cell and cause the neighboring cells congested, the neighboring cells can zoom in and the central cell zooms out to avoid any possible coverage hole. If the neighboring cells are designed to have high capacity, and therefore not necessarily zoom in, the central cell can also choose to sleep to reduce the energy consumption. In this case, the neighboring cells can either zoom out to take care of the coverage as in Figure 8.9(d), or serve the left MUs by transmitting cooperatively as in Figure 8.9(e). This example shows that cell zooming has the potential to achieve green cellular networks.

Implementing cell zooming in cellular networks needs to introduce some new components and corresponding functionalities to current network architecture. The framework of cell zooming is illustrated in Figure 8.10. There is a cell zooming server (CS), which controls the procedure of cell zooming. The CS is a virtual entity in the network, which can be either implemented in the gateway or distributed in the BSs. The CS will first sense the network state information for cell zooming, such as traffic load, channel conditions, user requirements, and so on. The sensing process can be realized by specific control messages. After collecting the information, the CS will analyze whether there are opportunities for cell zooming and make decisions. If a cell needs to zoom in or zoom out, it will coordinate with its neighbor cells with the help of CS. Then these cells will either zoom in or zoom out by network operations such as physical adjustment, BS cooperation, and relaying.

Techniques

Many techniques can be used to implement cell zooming. A simple and straightforward way is to adjust the physical parameters such as the transmit power of BSs. Besides physical adjustment, other techniques can also be used for cell zooming, as illustrated in Figure 8.11. A detailed discussion of the techniques used for cell zooming is given as follows.

Physical Adjustment: Adjusting physical parameters of network deployment can help to implement cell zooming. Cells can zoom out by increasing the transmit power of BS, and

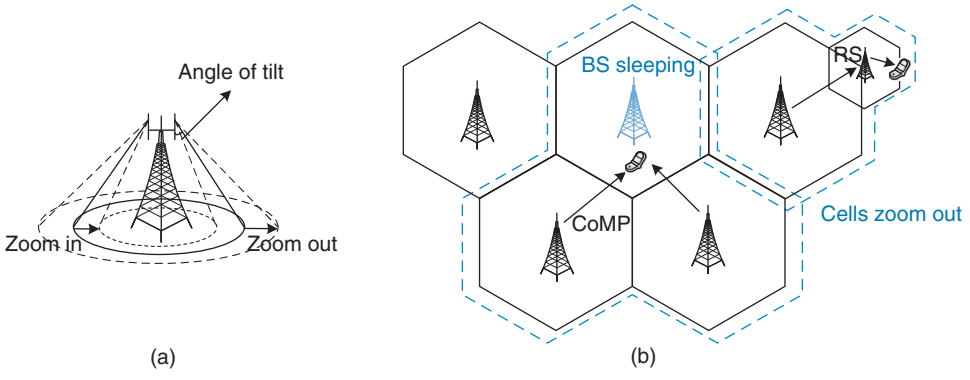

Figure 8.11 Techniques to implement cell zooming: (a). Cell zooms in or zooms out with physical adjustments; (b). Cells zoom out through BS cooperation and relaying

vice versa. Furthermore, antenna height and antenna tilt of BSs can also be adjusted for cells to zoom in or zoom out (Figure 8.11(a)). Such adjustments need the help of additional mechanical instruments.

BS Cooperation: BS cooperation means multiple BSs form a cluster, and cooperatively transmit to or receive from MUs, which is also named as CoMP (Coordinated Multipoint transmit/receive) in 3GPP LTE-A (Long-Term Evolution-Advanced) [5]. The newly formed cluster is a new cell from MUs' perspective, whose cell size is the sum of the original size of the BSs in cooperation. The size can be even larger, as BS cooperation can reduce inter-cell interference. In this case, cells zoom out to improve the coverage (Figure 8.11(b)).

Relaying: Relay stations (RSs) are deployed in cellular networks to improve the performance of cell-edge MUs, which is also an important technique in 3GPP LTE-A. The cell with RSs zooms out as shown in Figure 8.11(b). RSs can also be deployed near the boundary of two neighboring cells. In this case, RSs can relay the traffic from the cell under heavy load to the cell under light load. The former cell zooms in, and the latter cell zooms out.

BS Sleeping: When a BS is working in sleep mode, the air conditioner and other energy-consuming equipments can be switched off. BS sleeping can largely reduce the energy consumption of cellular network. In this case, the cell with BS working in sleep mode zooms in to 0, and its neighbor cells will zoom out to guarantee the coverage.

Benefits

Cell zooming can provide various benefits in cellular networks. Firstly, cell zooming can be used for load balancing by transferring traffic from cells under heavy load to cells under light load. Secondly, cell zooming can be used for energy saving. Contrary to the usage for load balancing, here cells zoom in to zero when the traffic load is light enough. Some BSs work in sleep mode, and the neighbor cells zoom out accordingly to guarantee the coverage. Therefore, cell zooming can both disperse load for load balancing and concentrate load for energy saving. In both cases, the resources are allocated to match the traffic distribution, however, the load transfer direction is opposite. It is a challenging problem to decide when to disperse load for load balancing and when to concentrate load for energy saving.

User experience can also improved by cell zooming, such as throughput, battery life, and so on. Techniques like BS cooperation and relaying can reduce the inter-cell interference, mitigate impact of shadowing and multipath fading, and reduce handover frequency. The techniques can also be jointly used. For example, in the scenario of isolated cell coverage, when cells zoom out by adjusting physical parameters such as antenna tilt, there will be more overlap among the cells. This provides opportunities for BS cooperation so that more MUs can achieve higher diversity gain, and coverage is also improved. As user requirements are better satisfied, there is no need for upgrading the network frequently, and this will reduce the operational cost of network operators.

Power control in cellular networks has been studied extensively in the literature (see the references in Ref. [20]). Power control can help to ensure efficient spatial reuse and minimize energy consumption. These functionalities is quite similar to that of cell zooming. However, cell zooming is different from power control in many ways. Power control focuses on the link-level performance and transmit power consumption, while cell zooming techniques focus on the network-level performance and energy consumption of the whole network. Power control does not actively change the cell size, while cell zooming actively changes the cell size by adjusting the transmit power of control signals.

Challenges

There exist many challenges to implement cell zooming. To make cell zooming efficient and flexible, traffic load fluctuations should be exactly traced and fed back to the CS. However, significant spatial and temporal fluctuations make it a challenging problem. One possible way to model the fluctuations is to divide it into long-term scale fluctuations and short-term scale fluctuations. The long-term scale fluctuations reflect the variation of traffic arrival rate, whose timescale is hours or days. The short-term scale fluctuation reflects the random arrival of users, whose timescale is seconds or minutes. It would be an interesting topic to find other models for the spatial and temporal traffic load fluctuations.

Compatibility is another challenging issue. Some of the techniques of cell zooming are not supported by current cellular networks, such as the additional mechanical equipments to adjust the antenna height and tilt, BS cooperation, and relaying techniques. Implementing cell zooming also needs to change current structure of network management. For example, feeding back the network information for cell zooming requires special control channels.

Cell zooming may cause other problems, such as inter-cell interference and coverage holes. When some neighboring cells zooms out together, there will be more inter-cell interference among them. If BS cooperation is infeasible, additional interference management schemes are needed to reduce the interference. Cell zooming may also produce coverage holes. When cells zoom in and zoom our, some areas in the network are possible have no coverage. In order to provide service to newly arrival MUs, the neighboring cells need to zoom in so as to cover these areas.

8.3.2.2 Centralized and Distributed Algorithms

In this subsection, a usage case of cell zooming for energy saving in cellular network is investigated. When traffic load is light, some cells can work in sleep mode to save energy, and other cells take care of the coverage. There have been many related studies about BS sleeping in

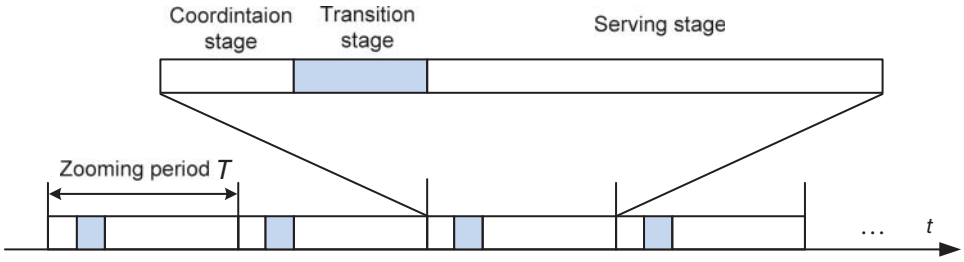

Figure 8.12 The process of cell zooming algorithms

cellular networks. In Ref. [12], a predefined BS sleeping scheme is presented according to a deterministic traffic variation pattern over time. Another similar work considers switching off some microcells at night hours while guaranteeing the blocking probability below a given target [21]. In these solutions, the sleeping pattern is fixed and the traffic intensity is assumed to be uniformly distributed over the whole network. In this article, we consider cellular networks with spatial and temporal traffic load fluctuations, and develop dynamic cell zooming algorithms for energy saving.

Consider a densely deployed cellular network in which the coverage of BSs overlaps and traffic load fluctuates over time and space. Assume there are M BSs, and all the BSs are assumed to have the same energy consumption. Each BS has two working modes: *active* mode with energy consumption P^a and *sleeping* mode with power consumption P^s, where P^a is usually much larger than P^s. MUs arrive at the network according to a Poisson process, and each MU will be associated with one BS upon its arrival. The sojourn time for each MU is exponentially distributed, and the rate requirement is fixed for each MU, denoted by r_i for MU i. The spectral efficiency is ω_{ij} when MU i is associated with BS j. Therefore, the bandwidth needed is given by $b_{ij} = r_i/\omega_{ij}$. We assume the spectral efficiency is independent of the associations among other BSs and MUs. The total bandwidth for BS j is B_j. When a new MU arrives, if there is not enough bandwidth to be allocated, the MU will be blocked. We are interested in two objectives, minimizing the energy consumption and minimizing the blocking probability. If there are more cells working in sleep mode, more energy will be saved, however, it also leads to larger blocking probability. Therefore, there is a trade-off between the two objectives.

As the mode transition of BSs will last for a period of time, during which the cells cannot provide service to MUs, thus frequent mode transition is infeasible in practice. In our cell zooming algorithms, time is divided into cell zooming periods, and the length of each period is T. Each period consists of three stages: coordination stage, transition stage, and serving stage, as shown in Figure 8.12. In the coordination stage, the CS collects necessary network state information for cell zooming, and makes decisions. Our proposed cell zooming algorithms will also work during this stage. In the transition stage, cells change their working modes, and complete the handoff process if needed. In the serving stage, cells fix their working mode, and provide service to current and newly arrival MUs in the network. We assume the length of coordination stage and transition stage are much shorter than serving stage, so the energy consumption depends on the work mode of cells in the serving stage.

Intuitively, in order to minimize the number of active BSs, traffic load should be concentrated to a few BSs so the left BSs under light load can be switched off. Following the intuition,

two cell zooming algorithms are proposed. The first one is a centralized algorithm, in which all the channel conditions and user requirements in the network are collected by the CS, and resource allocation and cell zooming operations are performed in a centralized way. The second one is a distributed algorithm. Each MU will select the BS to be associated with by itself based on the information provided broadcasted by the BSs. Generally speaking, the centralized algorithm requires more signaling overhead, but can achieve better performance compared with the distributed one. The details of the two cell zooming algorithms are given as follows.

Centralized Algorithm

In the centralized cell zooming algorithm, MUs feed back channel conditions and rate requirements to the BSs during the coordination stage. The CS will collect all these information together with BSs' bandwidth limitation. After receiving updates from all the MUs and BSs, CS will generate a 0–1 matrix $\mathbf{X} = [x_{ij}]$, where $x_{ij} = 1$ means MU i is associated with BS j, otherwise $x_{ij} = 0$. As each MU can only be served by one BS, the sum of each column in $\mathbf{X}$ is 1. The main idea of the algorithm is to switch off the BSs under light load as far as possible. As many MUs arrive during the serving stage, each active BS will reserve some bandwidth for the newly arrived MUs. Denote the proportion of bandwidth reserved in BS j as α_j, where $\alpha_j \in [0, 1]$. Initially, the idle bandwidth for BS j is given by

$$\tilde{B}_j = (1 - \alpha_j)B_j \tag{8.3}$$

Denote the set of MUs associated with BS j as $\mathcal{M}_j$. The traffic load of BS j is given by

$$L_j = \sum_{i \in \mathcal{M}_j} \frac{b_{ij}}{B_j} \tag{8.4}$$

The detailed procedure of the algorithm is described as follows.

- Step 1: Initialize all the L_j to be 0 and all the elements in matrix $\mathbf{X}$ to be 0.
- Step 2: For each MU i, find the set of BSs who can serve MU i without violating the bandwidth constraints, which means $L_j B_j + b_{ij} \leq \tilde{B}_j$. If the set is empty, MU i is blocked. Otherwise, associate MU i with a BS j that has the highest ω_{ij} in the set. Update L_j and $\mathbf{X}$ after each association.
- Step 3: Sort all the BSs by the ratio of $L_j B_j$ to $\tilde{B}_j$ by increasing order. All the BSs with the ratio 0 will zoom in to zero and work in sleep mode in the following serving period. For other BSs, find the BS j with the smallest ratio, and reassociation the MUs in $\mathcal{M}_j$ to other BSs in the network. If no MU is blocked, update $\mathbf{X}$ and go to Step 3. Otherwise, output $\mathbf{X}$ and end the procedure.

Distributed Algorithm

To reduce the information exchange and signaling overhead, we also propose a distributed cell zooming algorithm, in which each MU will select the BS by itself according to the measured channel conditions and BSs' traffic load. In the distributed algorithm, BSs also reserve bandwidth for newly arrival MUs as in centralized algorithm. In practice, traffic load information and bandwidth reservation parameters can be obtained by broadcasting control signals from BSs. Intuitively, each MU will select the BS with high load and high spectral efficiency. We

define a preference function if MU i is to be associated with BS j as

$$U(\omega_{ij}, L_j, \alpha_j) = \begin{cases} \dfrac{\omega_{ij}(L_j B_j + b_{ij})}{\tilde{B}_j} & L_j B_j + b_{ij} \le \tilde{B}_j \\ 0 & L_j B_j + b_{ij} > \tilde{B}_j, \end{cases} \tag{8.5}$$

which means MUs prefer those BSs with high load and high spectral efficiency, but the load cannot exceed a predefined threshold. The procedure of distributed cell zooming algorithm is described as follows.

- Step 1: Initialize all the L_j to be 0 and all the elements in matrix $\mathbf{X}$ to be 0.
- Step 2: For each MU i, find the set of BSs who can serve MU i without violating the bandwidth constraints, which means $L_j B_j + b_{ij} \le \tilde{B}_j$. If the set is empty, MU i is blocked. Otherwise, associate MU i with a BS j that has the highest $U(\omega_{ij}, L_j, \alpha_j)$ in the set. Update L_j and $\mathbf{X}$ after each association.
- Step 3: Repeat Step 2 until there is no update of $\mathbf{X}$, then output $\mathbf{X}$ and end the procedure.

In the distributed algorithm, no coordination among BSs is needed; therefore, much signaling overhead is reduced. The distributed algorithm works in an iterative way. The convergence of the distributed algorithm is guaranteed if any two MUs take no action simultaneously. This is because the BS selection set of each MU is finite. After the algorithm converges, the BSs with no association will work in sleep mode during the serving stage.

8.3.2.3 Performance Evaluation

The proposed dynamic cell zooming algorithms are evaluated in a scenario with time-varying traffic distribution. The simulation layout is 10 by 10 hexagon cells wrapped up to avoid boundary effect (Figure 8.13). The cell radius is set to 200 m, and assume that each BS can extend its coverage to at most 400 m. We only consider path loss for the channels between BSs and MUs, according to ITU microcell test environment [5]. Power consumption is 400 W for BSs in active mode and 10 W for BSs in sleep mode. The bandwidth of each BS is 5 MHz. MUs arrive in the network according to a Poisson process, and the average sojourn time of each MUs is 1 minute. To evaluate the algorithms in cellular networks with spatial traffic load fluctuations, three hotspots with relatively higher load than other areas are generated, as shown in Figure 8.13. A new MU arrives in each hotspot with probability 5% respectively, and their locations follow normal distribution with mean at central point of each hotspot and standard deviation R. The others are uniformly placed in the whole area. The rate requirement of each MU is 122 kbps. The cell zooming period T is set to be 1 hour, and all the simulation results are averaged over 100 cell zooming periods.

In the simulation, we set the reservation parameters the same for all BSs with $\alpha_i = \alpha$, then tune the value of α and calculate the average energy consumption. When α increases, more bandwidth is reserved and the cell zooming algorithm becomes more conservative. This will result in more BSs working in the active mode, and less blocking probability can be achieved. Therefore, by tuning α, we can leverage the trade-off between energy consumption and quality of service. The simulation results in Figure 8.14 verify our analysis. For a given arrival rate, there is a trade-off curve of energy consumption versus outage probability for each algorithm.

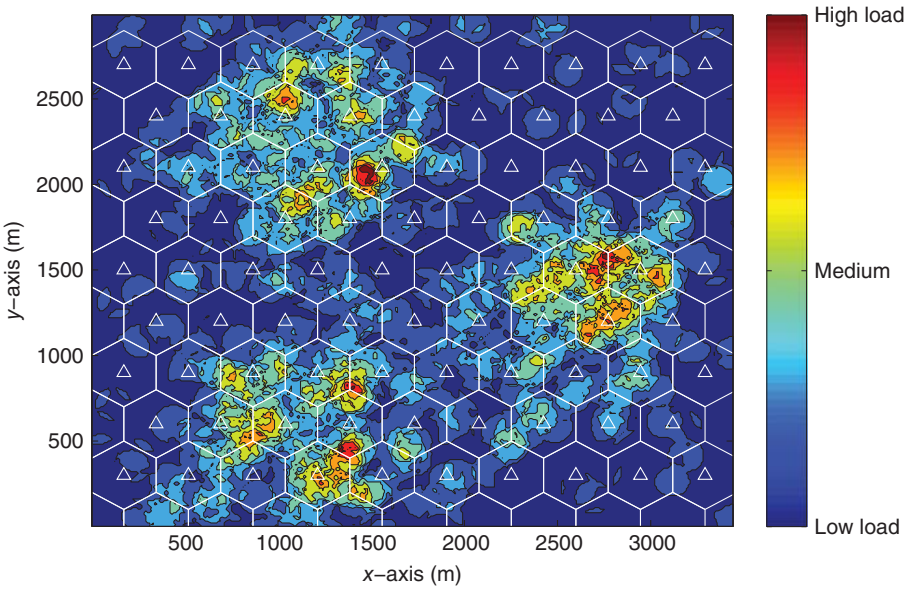

Figure 8.13 Traffic distribution in the tested cellular network layout

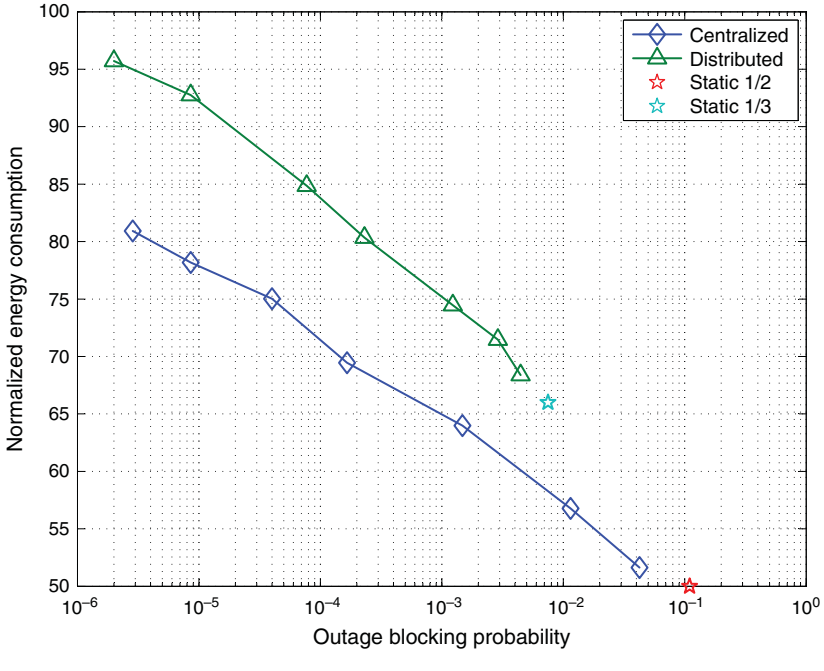

Figure 8.14 Energy outage trade-off of centralized and distributed cell zooming algorithms

The figure also shows that our algorithms can save a lot of energy (the energy consumption is normalized to 100 if all the BSs are active). The centralized algorithm can achieve a better trade-off than distributed algorithm. We also compare the cell zooming algorithms with the static BS sleeping algorithm, which switches off 1/2 or 1/3 of all the BSs. The results show that our centralized algorithm performs better than the static algorithm. Our algorithms are also more flexible as they can freely leverage the trade-off between energy consumption and outage probability.

8.3.3 TANGO by Adaptive BS Sleeping

As mentioned earlier, in cellular networks, the energy consumption of base stations (BSs) contributes 60% to 80% of the whole network [12], and will increase as network structure migrating from macrocell to micro- and picocells to meet the increasing demand of radio resources. As a result, the energy consumption of BSs becomes a major portion of the whole network energy consumption. As the energy consumption of a BS mainly comes from base-band processor, power amplifier, air-conditioner, and so on, rather than transmit power which only takes the ratio of 3.1% [22], turning as many as possible BSs into sleep mode whenever possible is considered as a promising technique to reduce the energy consumption.

In fact, due to the variation in time domain and the dynamic distribution among cells in space domain [23], there are opportunities for some BSs to turn to sleep mode when the traffic load in their coverage is low. However, when BSs turn to sleep mode, radio coverage and quality of service (QoS, e.g., blocking probability) must still be guaranteed. Thanks to the concept of *cell zooming* [14], the users in the sleeping cells can be served by the neighboring active BSs by transmit power adjustment, antenna reconfiguration, wireless relay, and BS cooperation technologies. As a consequence, BS sleeping is a feasible approach for energy saving in cellular networks.

To design efficient BS sleeping schemes, the following issues must be carefully studied.

- On the one hand, BS mode switching decision cannot be made by each BS individually. Not only the load condition of a BS itself, but also the load of its neighbors needs to be considered. For instance, a BS may not turn to sleep while its neighboring BSs are over loaded, even if its own traffic load is low. For this reason, each BS should make its mode switching decision via BS cooperation.
- On the other hand, although cooperation among all the BSs can achieve the optimal sleep policy, it is not applicable in real system due to the high complexity. Suboptimal solution obtained by *local* cooperation among neighboring BSs is preferable.
- Finally, taking signaling overhead, device lifetime, and switching energy consumption into account, frequent BS mode switching should be avoided. That is, BSs should try to minimize the number of switching actions or, in other words, maximize the BS *mode holding time*, which is defined as the holding duration between two successive switching actions.

In this section, we exploit the traffic variation feature to design an energy-efficient BS sleeping algorithm, which is then formulated as a dynamic programming (DP) problem with a combined cost function of energy consumption, switching cost, and blocking probability penalty. To reduce the dimension of state space and that of action space, *per-cell Q-factor* based on the cooperation among neighboring BSs is introduced, and a low-complexity algorithm is

proposed to find the suboptimal policy. In addition, to match the system with BS sleeping behavior, user association and handover algorithms are redesigned. Simulations demonstrate that, with the proposed algorithm, the active BS pattern well meets the time variation and the nonuniform spatial distribution of system traffic. Besides, the trade-off between the energy saving from BS sleeping and the cost of switching is well balanced by the proposed scheme.

8.3.3.1 System Model

Consider a downlink cellular network consisting of M BSs with universal frequency reuse. Let $\mathcal{M} = \{1, \ldots, M\}$ denote the set of BSs. The *maximum coverage* of BS m is the area where BS m can provide the required data rate, and the *cell* m is defined as the area that is nearest to BS m compared with other BSs. As depicted in Figure 8.15, the cell radius is R_c and the BS maximum coverage radius is R_b, which indicates that each BS is able to cover its neighbor cells. In the traditional cellular networks where all BSs are active, each cell is taken care of by its own BS. When some BSs turn to sleep, the actual BS coverage extends from their own cells to the neighbors with sleeping BSs. This is reasonable in urban scenarios, where BSs are densely deployed. The neighbors of BS m are denoted as $m(1), \ldots, m(B)$, where B is the number of neighbors ($B = 6$ in hexagonal cellular system). Denote $\mathcal{B}_m = \{m, m(1), \ldots, m(B)\}$ as the set of BSs, which can provide service to the users in cell m.

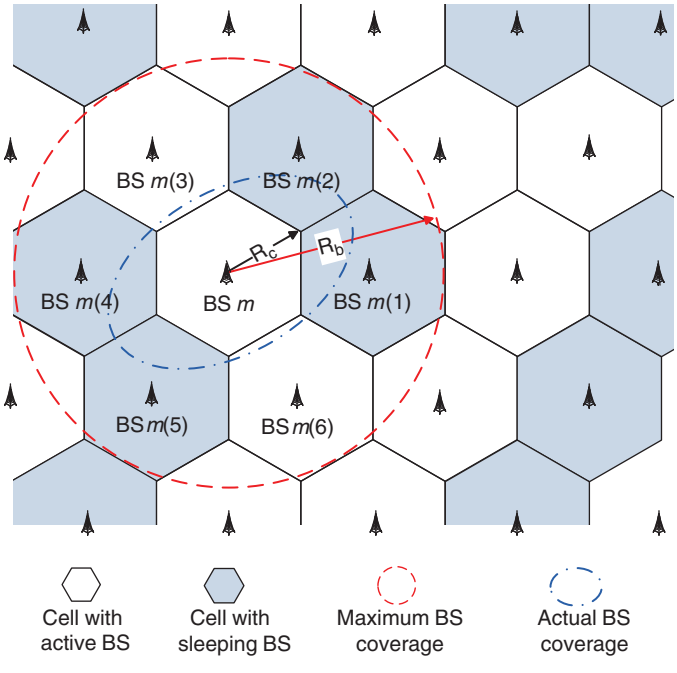

Figure 8.15 Cellular network architecture. The cell radius is R_c and the maximum coverage radius is R_b, which indicates the overlapped network structure. When some BSs turn to sleep, the active BSs extend their actual coverage from their own cells to the neighbors with sleeping BSs

The traffic arrives in cell m at time t as a Poisson process with intensity $\lambda_m(t)$. The traffic is assumed uniformly distributed in each cell, but asymmetric among different cells. Assume that the system has the statistic traffic information, that is, the average arrival rate $\boldsymbol{\lambda}(t) = \{\lambda_m(t)\}_{m=1}^{M}$, which is a periodic function with period T (e.g., 24 hours). Each user has a minimum rate requirement r_0. All users arrive randomly and then remain stationary until the transmission is finished. The transmission duration of each user follows exponential distribution with mean $1/\mu$.

Assume that each active BS m has limited radio resource, that is, the maximum bandwidth $W_m^{\max}$. Notice that the bandwidth here is the generalization of wireless resources that the BS can allocate, that is, subcarriers or time-slots, and so on. If user k is associated to BS m, the corresponding bandwidth demand is

$$w_{mk}(t) = \frac{r_0}{C_{mk}(t)} \tag{8.6}$$

where $C_{mk}(t)$ is the spectrum efficiency (instantaneous peak rate per unit bandwidth). It varies over time due to shadowing, multipath fading, and so on. Nevertheless, as the long timescale performance is considered here, we ignore the fast fading effects, that is, we assume that the spectrum efficiency is constant during the transmission, which is determined by the large-scale path loss. The expressions are then simplified to C_{mk} and w_{mk} without time index. The inter-cell interference is assumed already being taken care of by certain reuse or interference management schemes. Interference power is averaged over all possible user positions assuming all the BSs are in active mode. It is a conservative estimation as interference is reduced when some BSs go to sleep mode. Based on the aforementioned assumptions, C_{mk} depends only on the distance l_{mk} from BS m to user k

$$C(l_{mk}) = \begin{cases} \log_2\left(1 + \dfrac{P_t \beta l_{mk}^{-\alpha}}{N_0}\right), & 0 < l_{mk} \le R_b, \\ 0, & l_{mk} > R_b, \end{cases} \tag{8.7}$$

where P_t is the transmit power; β is the path-loss constant, and α is the path-loss exponent; N_0 is the noise-plus-interference power.

The time period T is divided into $N+1$ time intervals (each with index i) as shown in Figure 8.16. Note that to be able to track the system traffic variation, the length of time interval $\tau^{(i)}$ varies with $\boldsymbol{\lambda}(t)$ so that on average, constant number of users, K_τ, arrive during $\tau^{(i)}$. Then we have

$$K_\tau = \int_{t^{(i)}}^{t^{(i)}+\tau^{(i)}} \sum_{m=1}^{M} \lambda_m(t)\mathrm{d}t, \tag{8.8}$$

where $\tau^{(i)}$ can be calculated by some numerical method.

Assume that each BS $m \in \mathcal{M}$ can work in two modes: *active* mode (denoted as $s_m^{(i)} = 1$) and *sleep* mode (denoted as $s_m^{(i)} = 0$). In active mode, BS works with full power consumption $P_{\max}$ including signal processing, air-conditioning, power amplifier, and so on. In sleep mode, BS works with minimum power $P_{\min}$ to be able to wake up. In each time interval $\tau^{(i)}, i = 1, \ldots, N$, the system works in the fixed state $\boldsymbol{s}^{(i)} = \{s_m^{(i)}\}_{m=1}^{M}$. The state space is

$$\mathcal{S} = S_1 \times S_2 \times \ldots \times S_M, \tag{8.9}$$

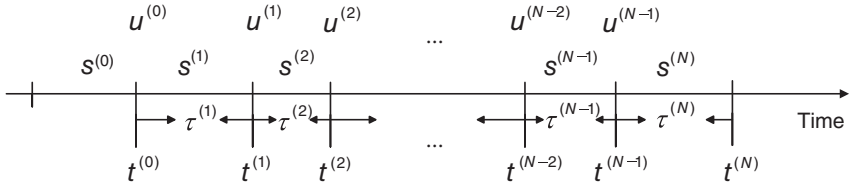

Figure 8.16 System operation over time. The network keeps a constant state $\boldsymbol{s}^{(i)}$ in each time interval $\tau^{(i)}$, and operates action $\boldsymbol{u}^{(i)}$ at each time spot $t^{(i)}$

where $S_m = \{0, 1\}, m = 1, \ldots, M$ is the set of working modes of BS m.

At each time spot $t^{(i)}, i = 0, \ldots, N$, the BSs take the action $\boldsymbol{u}^{(i)} = \{u_m^{(i)}\}_{m=1}^{M}$. The action space is

$$\mathcal{U} = U_1 \times U_2 \times \ldots \times U_M, \tag{8.10}$$

where $U_m = \{0, 1\}, m = 1, \ldots, M$ is the set of actions of BS m. Denote $u_m^{(i)} = 1$ as the action that BS m switches its working mode and $u_m^{(i)} = 0$ otherwise, as that BS m maintains its working mode.

The overview of the system operation is as follows. The BSs work in the constant state $\boldsymbol{s}^{(i)}$ during the time interval $\tau^{(i)}, i = 1, \ldots, N$, and the users are served by the currently active BSs. At each time spot $t^{(i)}, i = 0, 1, \ldots, N-1$, the BSs decide whether to switch their working modes or not according to the action $\boldsymbol{u}^{(i)}$. If a BS switches from active mode to sleep mode, the associated users are shifted to the active neighbors. After the BS mode switching process is finished, the system goes into the next time interval $\tau^{(i+1)}$ with updated state $\boldsymbol{s}^{(i+1)}$.

Generally speaking, the object of BS sleeping algorithm is to determine the action $\boldsymbol{u}^{(i)}, i = 0, \ldots, N-1$ to minimize the system energy consumption given the initial state $\boldsymbol{s}^0$ and statistic traffic information $\boldsymbol{\lambda}(t)$, as well as to maintain a predefined blocking probability and avoid from frequent mode switching. Specifically, we formulate it as a dynamic programming problem, which is detailed in the next section.

8.3.3.2 Problem Formulation

A standard dynamic programming problem contains the following elements: state, action, state transition, and per-stage cost [24]. Note that the system state is actually $(\boldsymbol{s}^{(i)}, \boldsymbol{\lambda}^{(i)})$, where $\boldsymbol{\lambda}^{(i)} = \{\lambda_m^{(i)}\}_{m=1}^{M}$, $\lambda_m^{(i)} = 1/\tau^{(i+1)} \int_{t^{(i)}}^{t^{(i+1)}} \lambda_m(t)\mathrm{d}t$. We denote $\boldsymbol{s}^{(i)}$ as the system state for notation convenience, as $\boldsymbol{\lambda}^{(i)}$ is system-determined parameter and do not change with any action.

Given the current system state $\boldsymbol{s}^{(i)}$ and the control action $\boldsymbol{u}^{(i)}$, the state transition is determined by

$$\boldsymbol{s}^{(i+1)} = f(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) = \{|s_m^{(i)} - u_m^{(i)}|\}_{m=1}^{M}. \tag{8.11}$$

The per-stage cost function $g(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)})$ composed of three parts. The first part is the energy consumption of BS operation, which is calculated as

$$g_{\mathrm{e}}^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) = \sum_{m=1}^{M} [s_m^{(i+1)} P_{\max} + (1 - s_m^{(i+1)}) P_{\min}] \tau^{(i+1)} \tag{8.12}$$

The second part is the BS mode switching cost

$$g_s^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) = \sum_{m=1}^{M} E_s u_m^{(i)} \tag{8.13}$$

where E_s is the cost of switching between active mode and sleep mode.

Finally, according to the optimization problem, blocking probability penalty should be integrated into the cost function. To relate the BS sleeping action with blocking probability, we define *system blocking probability* and *area blocking probability* next.

Definition 8.1 *The* system blocking probability *at state* $\boldsymbol{s}$ *is the probability that a newly arrived user* k' *is blocked, that is, none of the active BSs can provide the required bandwidth to this user:*

$$P_{\text{sys}}(\mathbf{s}) = \Pr\left\{ \bigcap_{\substack{m:\, s_m=1,\\ w_{mk'}<\infty}} \left(w_{mk'} + \sum_k x_{mk} w_{mk} > W_m^{\max} \right) \right\} \tag{8.14}$$

where binary variable $x_{mk} = 1$ *if user* k *is associated to BS* m*, and equals 0 otherwise. The summation is over all the existing users in the system.*

Definition 8.2 *The* area blocking probability *of area* A *is the conditional probability that a user* k' *arrived in* A *is blocked:*

$$P_a(A, \boldsymbol{s}) = \Pr\left\{ \bigcap_{\substack{m:\, s_m=1,\\ w_{mk'}<\infty}} \left(w_{mj} + \sum_k x_{mk} w_{mk} > W_m^{\max} \right) | k' \in A \right\}. \tag{8.15}$$

As the network is divided into multiple cells, the relationship between system blocking probability and area blocking probability is given by the law of total probability

$$P_{\text{sys}}(\boldsymbol{s}) = \sum_{m=1}^{M} \Pr(k' \in A_m) P_a(A_m, \boldsymbol{s}) \tag{8.16}$$

where A_m is the area of cell m. As cell m can only be covered by the BSs in $\mathcal{B}_m$, the area blocking probability of A_m can be approximated as

$$P_a(A_m, \boldsymbol{s}) \approx P_a(A_m, \tilde{\boldsymbol{s}}_m) \tag{8.17}$$

where $\tilde{\boldsymbol{s}}_m = \{s_m, s_{m(1)}, \ldots, s_{m(B)}\}$ is the *local state* of cell m. It can be easily verified that a sufficient condition for $P_{\text{sys}}(\boldsymbol{s}) \le P_{\text{thr}}$ is

$$P_a(A_m, \tilde{\boldsymbol{s}}_m) \le P_{\text{thr}}, \quad \forall\, m \tag{8.18}$$

where P_{thr} is a given threshold.

The approximated area blocking probability $P_{\mathrm{a}}(A_m, \tilde{\boldsymbol{s}}_m)$ is derived in 8.3.3 and is summarized next:

$$P_{\mathrm{a}}(A_m, \tilde{\boldsymbol{s}}_m) = \prod_{n \in \mathcal{B}_m, s_n = 1} \left(\frac{\lambda_n'}{\lambda_n' + \mu} \right)^{K_n^{\max}} \tag{8.19}$$

where $\lambda_n' = \lambda_n + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} \lambda_j / I_j M_j^{\mathrm{on}}, R_j = R_c \sqrt{(B/I_j M_j^{\mathrm{on}} + 1)}, M_j^{\mathrm{on}} = \sum_{j' \in \mathcal{B}_j \cap \mathcal{B}_m} s_{j'}$, and $I_j = 1$ if $j = m$ and $I_j = 2$ if $j = m(b), b = 1, \ldots, B$, and $K_n^{\max}$ is expressed as

$$K_n^{\max} = \lceil x \rceil \tag{8.20}$$

The operator $\lceil \cdot \rceil$ rounds the real number to the nearest integer no smaller than it, where x stands for

$$x = \frac{R_c^2 W_n^{\max}}{2 r_0} \left(1 + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} \frac{\lambda_j I_n}{I_j M_j^{\mathrm{on}} \lambda_n} \right) \left(\int_0^{R_c} \frac{l \mathrm{d} l}{C(l)} + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} \frac{\lambda_j I_n}{B \lambda_n} \int_{R_c}^{R_j} \frac{l \mathrm{d} l}{C(l)} \right)^{-1} \tag{8.21}$$

Then the blocking probability penalty is calculated as a sum of area blocking probability penalty of all cells

$$g_b^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) = \sum_{m=1}^{M} h(P_{\mathrm{a}}(A_m, f(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)})), P_{\mathrm{thr}}) \tag{8.22}$$

where $\tilde{\boldsymbol{u}}_m = \{u_m, u_{m(1)}, \ldots, u_{m(B)}\}$ is *local action* of cell m, and the penalty function h is defined as

$$h(P_{\mathrm{a}}(A_m, \tilde{\boldsymbol{s}}_m), P_{\mathrm{thr}}) = \begin{cases} E_b P_{\mathrm{a}}(A_m, \tilde{\boldsymbol{s}}_m), & \text{if } P_{\mathrm{a}}(A_m, \tilde{\boldsymbol{s}}_m) > P_{\mathrm{thr}} \\ 0, & \text{else} \end{cases} \tag{8.23}$$

where E_b is a very large number.

In summary, the per-stage cost is given by

$$g^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) = g_e^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) + g_s^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) + g_b^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}), i = 0, 1, \ldots, N-1 \tag{8.24}$$

$$g^N(\boldsymbol{s}^N) = 0 \tag{8.25}$$

In this paper, given the traffic variation function $\boldsymbol{\lambda}(t)$ and the initial state $\boldsymbol{s}^{(0)}$, we seek to minimize the total cost of all stages

$$\min_{\{\boldsymbol{u}^{(0)}, \ldots, \boldsymbol{u}^{(N-1)}\}} \sum_{i=0}^{N-1} g^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}), \tag{8.26}$$

and find an optimal control policy $\boldsymbol{\nu} = \{\nu^{(0)}, \nu^{(1)}, \ldots, \nu^{(N-1)}\}$ that satisfies

$$\boldsymbol{\nu} = \arg \min_{\{\boldsymbol{u}^{(0)},\ldots,\boldsymbol{u}^{(N-1)}\}} \sum_{i=0}^{N-1} g^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) \tag{8.27}$$

In the next section, we first present the standard DP algorithm. Then by reducing the state size using per-cell Q-factor, a low-complexity algorithm is proposed. Also, BS sleeping related user association and handover is discussed, as well as implementation issues.

8.3.3.3 Dynamic Programming Algorithm

A. General Solution

The cost minimization problem (8.26) can be solved by the standard DP algorithm [24] taking the form

$$J^N(\boldsymbol{s}^{(N)}) = 0 \tag{8.28}$$

$$J^{(i)}(\boldsymbol{s}^{(i)}) = \min_{\boldsymbol{u}^{(i)} \in \mathcal{U}} [g(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) + J^{(i+1)}(f(\boldsymbol{s}^{(i)}\boldsymbol{u}^{(i)}))] \tag{8.29}$$

where $i = 0, 1, \ldots, N-1$. Proceeding backward induction of Eq. (8.29) from $N-1$ to 0, the optimal cost is equal to $J^0(\boldsymbol{s}^0)$ for the given $\boldsymbol{s}^{(0)}$. Furthermore, if $\boldsymbol{\nu}^{(i)} = \boldsymbol{u}^{(i)}(\boldsymbol{s}^{(i)})$ minimizes the right side of Eq. (8.29) for each $\boldsymbol{s}^{(i)}$ and i, the policy $\boldsymbol{\nu} = \{\nu^{(0)}, \nu^{(1)}, \ldots, \nu^{(N-1)}\}$ is optimal.

Note that the cardinalities of $\mathcal{S}$ and $\mathcal{U}$ are both 2^M, which increase exponentially with the number of BSs M. Due to the *curse of dimensionality* (as termed in Ref. [24]), the computational requirement to obtain the optimal control policy is overwhelming if the network size is large. As a consequence, it is very difficult to implement the standard DP algorithm in practical systems. In the following, we introduce per-cell Q-factor estimation to reduce the size of state space and propose a low-complexity decision algorithm to simplify the process of action.

B. Q-factor and Space Reduction

Define the Q-factor as follows:

$$\boldsymbol{Q}^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) = g^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) + J^{(i+1)}(f(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)})) \tag{8.30}$$

where $i = 0, 1, \ldots, N-1$. According to (8.29) and (8.30), we have

$$J^{(i)}(\boldsymbol{s}^{(i)}) = \min_{\boldsymbol{u}^{(i)} \in \mathcal{U}} \boldsymbol{Q}^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) \tag{8.31}$$

$$\boldsymbol{Q}^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) = g^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) + \min_{\boldsymbol{u}^{(i+1)} \in \mathcal{U}} \boldsymbol{Q}^{(i+1)}(f(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}), \boldsymbol{u}^{(i+1)}) \tag{8.32}$$

The Q-factor $\boldsymbol{Q}^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)})$ represents the cost of applying the action $\boldsymbol{u}^{(i)}$ at the current state $\boldsymbol{s}^{(i)}$ and applying the action $\arg \min_{\boldsymbol{u}^{(i+1)} \in \mathcal{U}} \boldsymbol{Q}^{(i+1)}(f(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}), \boldsymbol{u}^{(i+1)})$ at the next state $\boldsymbol{s}^{(i+1)} = f(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)})$. To reduce the size of state space, we approximate the Q-factor as a sum of per-cell Q-factors, that is,

$$\boldsymbol{Q}^{(i)}(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)}) \approx \sum_{m=1}^{M} Q_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}) \tag{8.33}$$

where the per-cell Q-factor is

$$Q_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}) = g_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}) + \min_{\tilde{\boldsymbol{u}}_m^{(i+1)}} Q_m^{(i+1)}(f(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}), \tilde{\boldsymbol{u}}_m^{(i+1)}) \tag{8.34}$$

where the per-cell per-stage cost is

$$g_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}) = \frac{1}{B+1} \sum_{n \in \mathcal{B}_m} \{[s_n^{(i+1)} P_{\max} + (1 - s_n^{(i+1)}) P_{\min}] \tau^{(i+1)} + E_s u_n^{(i)}\} + h(P_{\mathrm{a}}(A_m, f(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)})), P_{\mathrm{thr}}) \tag{8.35}$$

which includes the energy consumption and the switching cost of the BSs in $\mathcal{B}_m$, and the area blocking probability of cell m. The per-cell Q-factor $Q_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)})$ only needs the information of the BSs in $\mathcal{B}_m$, which indicates the limited cooperation among neighboring BSs. It can be recursively calculated for each BS $m \in \mathcal{M}$. The suboptimal control policy is then given by

$$\boldsymbol{\nu}^{(i)}(\boldsymbol{s}^{(i)}) = \arg \min_{\boldsymbol{u}^{(i)} \in \mathcal{U}} \sum_{m=1}^{M} Q_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}). \tag{8.36}$$

Algorithm 1 Q-factor Recursion

1: Calculate $P_{\mathrm{a}}(A_m, \tilde{\boldsymbol{s}}_m^{(i)})$ for all $i = 1, 2, \ldots, N, \tilde{\boldsymbol{s}}_m^{(i)} \in \{0, 1\}^{B+1}$.
2: Calculate $Q_m^{(N-1)}(\tilde{\boldsymbol{s}}_m^{(N-1)}, \tilde{\boldsymbol{u}}_m^{(N-1)}) = g_m^{(N-1)}(\tilde{\boldsymbol{s}}_m^{(N-1)}, \tilde{\boldsymbol{u}}_m^{(N-1)})$ for all $\tilde{\boldsymbol{s}}_m^{(N-1)}, \tilde{\boldsymbol{u}}_m^{(N-1)} \in \{0, 1\}^{B+1}$.
3: **for** $i = N - 2$ to 0 **do**
4: Calculate $Q_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}) = g_m^{(i)}(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}) + \min_{\tilde{\boldsymbol{u}}_m^{(i+1)}} Q_m^{(i+1)}(f(\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)}), \tilde{\boldsymbol{u}}_m^{(i+1)})$ for all $\tilde{\boldsymbol{s}}_m^{(i)}, \tilde{\boldsymbol{u}}_m^{(i)} \in \{0, 1\}^{B+1}$.
5: **end for**

Remark 1 (State Space Reduction): The size of state space in each stage is substantially reduced from 2^M (exponential growth with respect to the number of BSs M) to $M2^{B+1}$ (linear growth w.r.t. M).

Although the size of state space is reduced by introducing per-cell Q-factor, the minimization progress (8.36), which requires exhaustive search over the action space $\mathcal{U}$, is still of high complexity. Based on the per-cell Q-factor, we propose an iterative decision-making algorithm. In each iteration, the BSs find the optimal local actions and update the global action one by one. Until the sum of per-cell Q-factors does not decrease, the iteration terminates. It is summarized in **Algorithm 2**.

Remark 2 (Action Space Reduction): In the greedy search step 6, the size of decision space is 2^{B+1}. Obviously, the iteration (from step 3 to step 10) converges in a finite number of iterations. Simulations show that the number of iterations is no more than 4 mostly. As a result, the action search complexity in each stage is reduced from $O(2^M)$ to $O(M2^{B+1})$.

Algorithm 2 Action Iteration

for $i = 0$ to $N-1$ **do**
  Set $\boldsymbol{u}^* = 0, Q_{\min} = \infty, \hat{Q} = E_b$.
  while $\hat{Q} < Q_{\min}$ **do**
    Set $Q_{\min} = \hat{Q}$.
    for $m = 1$ to M **do**
      Find the optimal solution of the following problem

$$\tilde{\nu}_m^* = \arg\min_{\tilde{\boldsymbol{\nu}}_m} \sum_{n=1}^{m} Q_n^{(i)}(\tilde{\boldsymbol{s}}_n^{(i)}, \tilde{\boldsymbol{u}}_n),\ \tilde{\boldsymbol{\nu}}_m \in \{0,1\}^{B+1}$$

      where $\boldsymbol{u}$ is determined as: $u_n = u_n^*$, if $n \notin \mathcal{B}_m$; $u_n = |u_n^* - \nu_n|$, if $n \in \mathcal{B}_m$.
    Update $\boldsymbol{u}^*$ as: $u_n^* = u_n^*$, if $n \notin \mathcal{B}_m$; $u_n^* = |u_n^* - \nu_n^*|$, if $n \in \mathcal{B}_m$.
    end for
    Update $\hat{Q} = \sum_{n=1}^{M} Q_n^{(i)}(\tilde{\boldsymbol{s}}_n^{(i)}, \tilde{\boldsymbol{u}}_n^*)$.
  end while
  Set $\boldsymbol{u}^{(i)} = \boldsymbol{u}^*, \boldsymbol{s}^{(i+1)} = f(\boldsymbol{s}^{(i)}, \boldsymbol{u}^{(i)})$.
end for

C. Systematic Design

Recall that during the time interval $\tau^{(i)}, i = 1, \ldots, N$, the users are served by the currently active BSs. At each time spot $t^{(i)}, i = 0, 1, \ldots, N-1$, if a BS switches from active mode to sleep mode, the associated users are shifted to the active neighbors. As the accessible BSs provide different signal strength and are of various load conditions, user association and handover should be carefully designed to optimize resource allocation.

User Association: Load balancing scheme is implemented for the user association to reduce system blocking probability. In the literature, load balancing has been extensively studied and some efficient scheduling methods have been proposed. We make use of the *load-aware cell-site selection scheme* [25]. If the user k arrives in cell m, its candidate serving BS set is $\mathcal{C}_k = \{n \in \mathcal{B}_m | s_n = 1\}$. Notice that the remaining bandwidth of the BS $n \in \mathcal{C}_k$ maybe not enough to accept user k, that is, $W_n^{\max} - W_n < w_{nk}$, where W_n is the allocated bandwidth of BS n. In this case, the user should try the BSs in $\mathcal{C}_k$ one by one. The proposed user association algorithm is stated in **Algorithm 3**.

Note that $|\cdot|$ is the cardinality of a set. This scheme can be operated in a distributed manner. User selects the serving BS with high channel quality and low traffic load as well. Consequently, the asymmetric load across the whole system is implicitly balanced.

User Handover: At each time spot $t^{(i)}, i = 0, 1, \ldots, N-1$, the users which associate with the BSs turning from active mode to sleep mode, should change their association to neighboring active BSs. To minimize the number of droppings, the handover algorithm also takes the idea of load balancing. Initialize $\mathcal{H}^{(i)} = \{k | x_{mk} = 1, s_m^{(i)} = 1, s_m^{(i+1)} = 0, m \in \mathcal{M}\}$. The handover algorithm is presented in **Algorithm 4**.

After the algorithm terminates, the remaining users in $\mathcal{H}^{(i)}$ are dropped. Due to the nature of load balancing, the number of dropped users is minimized.

Algorithm 3 User Association

Set up a list $L_k = \{n_1, n_2, \ldots | n_j \in C_k\}$ with $c_{n_1} \geq c_{n_2} \geq \ldots$, where $c_n = C(l_{nk})W_n^{\max}/W_n$.
Set flag = 0.
for $j = 1$ to $|C_k|$ **do**
  if $W_{n_j}^{\max} - W_{n_j} \geq w_{n_j k}$ **then**
    $x_{n_j k} = 1, W_{n_j} = W_{n_j}^{\max} - w_{n_j k}$, flag = 1; break.
  end if
end for
if flag $\neq 1$ **then**
  No BS is selected, user k is blocked.
end if

Algorithm 4 User Handover

Set up a BS-user pair list $L^{(i)} = \{(m_1, k_1), (m_2, k_2), \ldots | s_{m_j}^{(i+1)} = 1, k_j \in \mathcal{H}^{(i)}, C(l_{m_j k_j}) > 0\}$ with $c_{(m_1,k_1)} \geq c_{(m_2,k_2)} \geq \ldots$, where $c_{(m,k)} = C(l_{mk})W_m^{\max}/W_m$.
Set $j = 0$.
while $\mathcal{H}^{(i)} \neq \emptyset$ and $j < |L^{(i)}|$ **do**
  Set $j = j + 1$.
  if $W_{m_j}^{\max} - W_{m_j} \geq w_{m_j k_j}$ and $k_j \in \mathcal{H}^{(i)}$ **then**
    $x_{m_j k_j} = 1, W_{m_j} = W_{m_j}^{\max} - w_{m_j k_j}, \mathcal{H}^{(i)} = \mathcal{H}^{(i)} \backslash \{k_j\}$.
  end if
end while

D. Implementation Issue

As the proposed algorithm offers an off-line solution, different policies are implemented for different traffic variation pattern. A typical example is that policies for workday and weekend should be distinguished.

In real networks, the statistic features of traffic distribution and variation may change. For instance, the increase of the total number of subscribers enhances the average traffic intensity; a newly opened business center will become a new hotspot in the daytime, which changes the traffic distribution in space domain. To be able to track the long-term variation of traffic, the system should establish a dataset and record the number of calls in each cell. Depending on the statistic information obtained from the dataset, the system can operate the proposed algorithm to update the BS sleep pattern whenever necessary.

8.3.3.4 Simulation Study

The simulation layout is 10 by 10 hexagon cells with wrap-up to avoid boundary effect, which is shown in Figure 8.17. The cell radius is $R_c = 200$ m and the BS's maximum coverage is $R_b = 520$ m. We set the power consumption $P_{\max} = 1 \times 10^3$ W and $P_{\min} = 0$. The maximum bandwidth is $W_m^{\max} = 5$ MHz, which is identical for all BSs. User rate requirement is

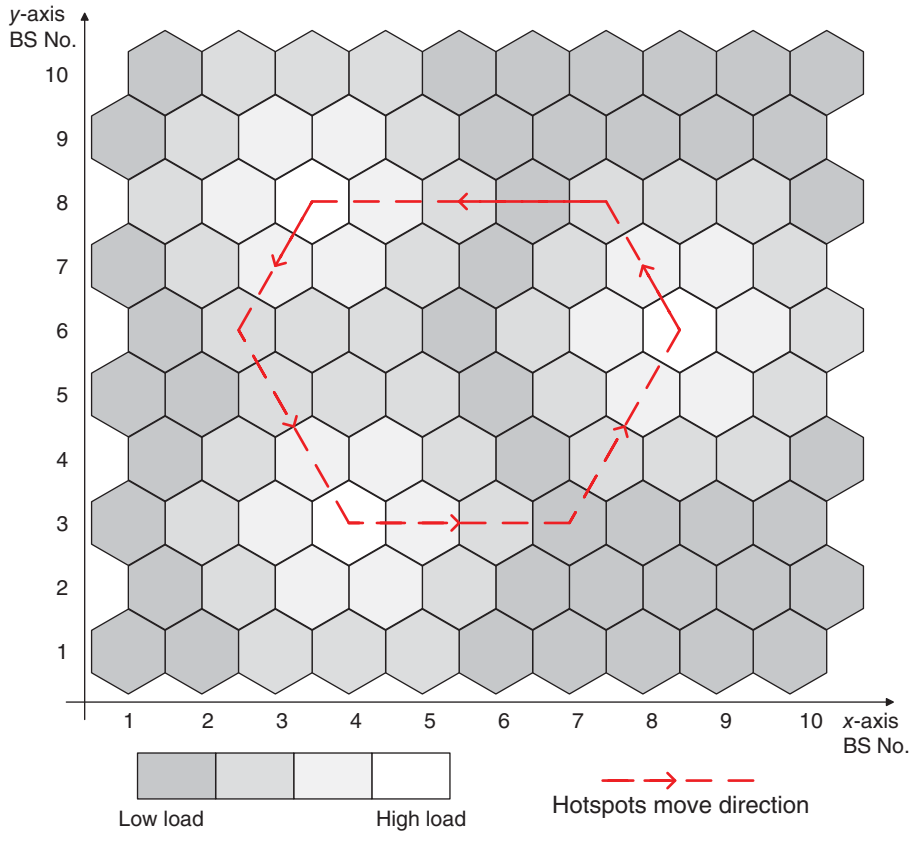

Figure 8.17 Simulation layout and traffic distribution. Three hotspots are formed and move along the dashed line anticlockwise every 24 hours a cycle. The highest load is $\lambda_{\mathrm{h}}(t)$, and the others are $\alpha_l \lambda_{\mathrm{h}}(t)$, $l = 1, 2, 3, 0 \leq \alpha_3 \leq \alpha_2 \leq \alpha_1 \leq 1$, respectively

$r_0 = 122$ kbps. Transmission duration parameter is $\mu = 1/180s^{-1}$. The link parameters are set according to ITU microcell test environment [26]. The transmit power is $P_t = 41$ dBm. The noise-plus-interference power N_0 is calculated by setting the reference SNR at distance 200 m to be 0 dB. Path-loss model is $PL^{\mathrm{dB}}(l_{ik}) = 33.05 + 36.7\log_{10}(l_{ik})$. The number of user arrivals in each time interval is $K_\tau = 1 \times 10^4$. The blocking probability penalty $E_b = 1 \times 10^8$ J and the threshold $P_{\mathrm{thr}} = 1\%$. To simulate the asymmetric traffic distribution, three *hotspots* are formed in the network. The traffic distribution is configured as follows:

- Average arrival rate (or traffic intensity) of the whole network $\lambda_{\mathrm{w}}(t) = \sum_{m=1}^{M} \lambda_m(t)$ varies according to the red-dashed line in Figure 8.18. The period of arrival rate is $T = 24$ hours.
- Three hotspots are generated and move along the dashed line shown in Figure 8.13 anti-clockwise every 24 hours a cycle.
- Set the arrival rates of the hotspot center cells as $\lambda_m(t) = \lambda_{\mathrm{h}}(t)$. Then the arrival rates of the lst-tier of hotspot center cells are $\lambda_m(t) = \alpha_l \lambda_{\mathrm{h}}(t), l = 1, 2$, and the others are $\lambda_m(t) = \alpha_3 \lambda_{\mathrm{h}}(t)$, where $0 \leq \alpha_3 \leq \alpha_2 \leq \alpha_1 \leq 1$.

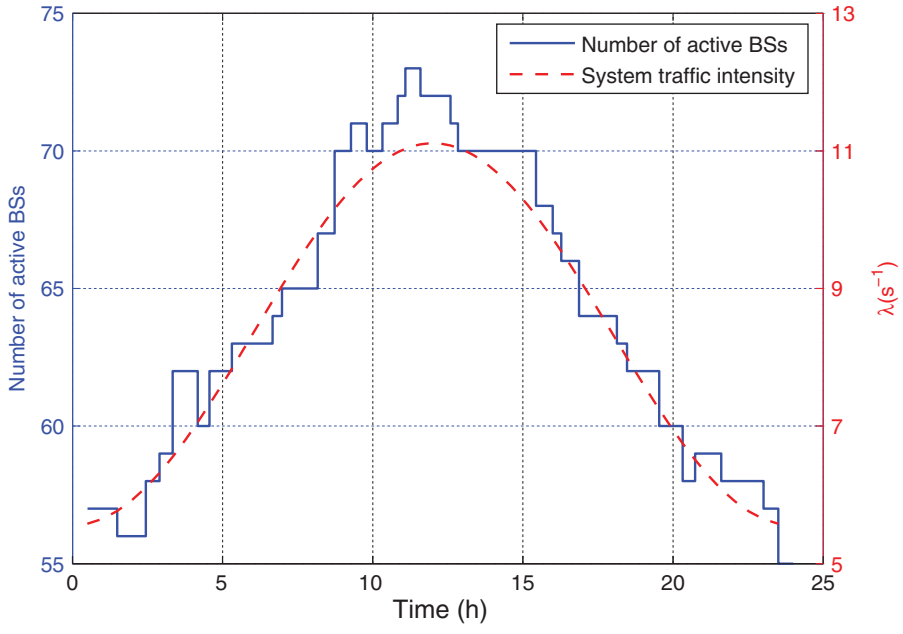

Figure 8.18 Number of active BSs compared with average traffic intensity in time space

The simulation is performed as follows. We first calculate the sleeping policy with respect to the statistic traffic information $\boldsymbol{\lambda}(t)$ and the given initial state $\boldsymbol{s}^{(0)}$. Then the random user arrival is generated in accordance with $\boldsymbol{\lambda}(t)$ to test the performance of the policy obtained by the proposed algorithm. The initial state $\boldsymbol{s}^{(0)}$ is set by activating half of the BSs in the network uniformly and then opening two more BSs in each hotspot.

A. BS Sleeping Pattern

In this part of simulation, parameter settings are: $E_s = 2.5 \times 10^5$ J/switch, $\alpha_1 = 0.88, \alpha_2 = 0.63, \alpha_3 = 0.50$.

BS mode switching behavior along with the average traffic intensity is presented in Figure 8.18. It is shown that our proposed DP algorithm tracks the variation of the average traffic intensity well in time domain. To illustrate the spatial consistency between the traffic distribution and the number of active BSs, we take the stage $i = 20$ as an example, and calculate $s'_m = s_m^{(i)} + \sum_{m(j),j=1,\ldots,B} s_{m(j)}^{(i)}/2$ to imply the number of active BSs around each cell. Comparing Figure 8.19 with Figure 8.20, we can see that more BSs are active in the highly loaded area, while less BSs are active in the area with low load. Still, in the low load area, there generally are some active BSs in order to guarantee the network coverage. As a result, the active BS distribution well meets the spatial distribution of traffic intensity. In addition, the blocking probability in each time interval is maintained below the target (1%) almost all the time (see Figure 8.21). On average, more than 36% energy consumption is reduced. The average blocking probability over 24 hours is 0.3%, which shows that the area blocking probability estimation is conservative. More elaborate area blocking probability analysis can be performed to further improve the energy saving performance.

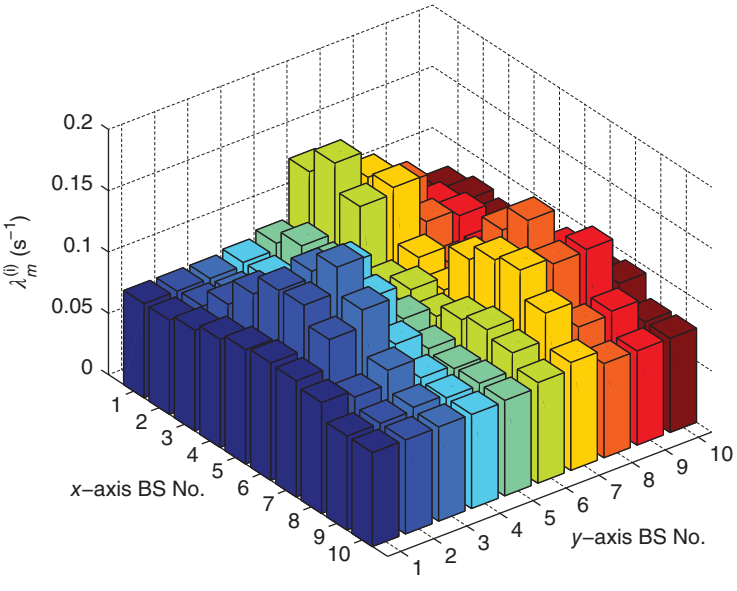

Figure 8.19 Traffic distribution in spatial domain

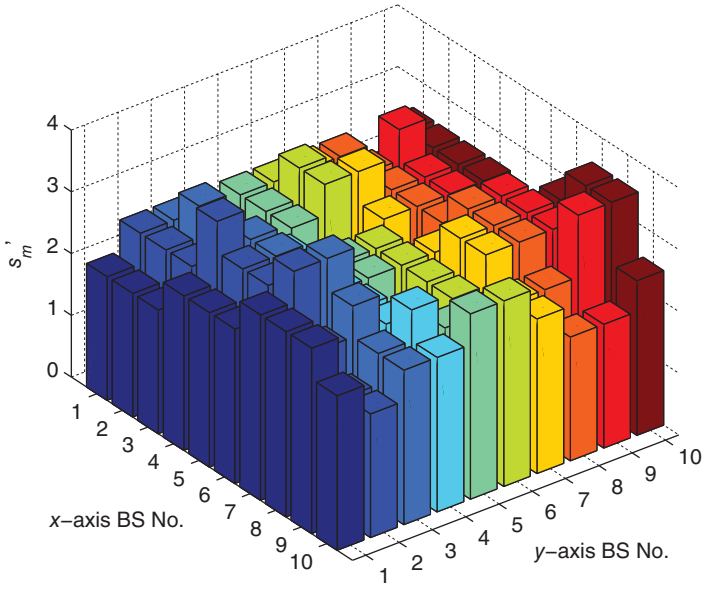

Figure 8.20 BS state distribution in spatial domain

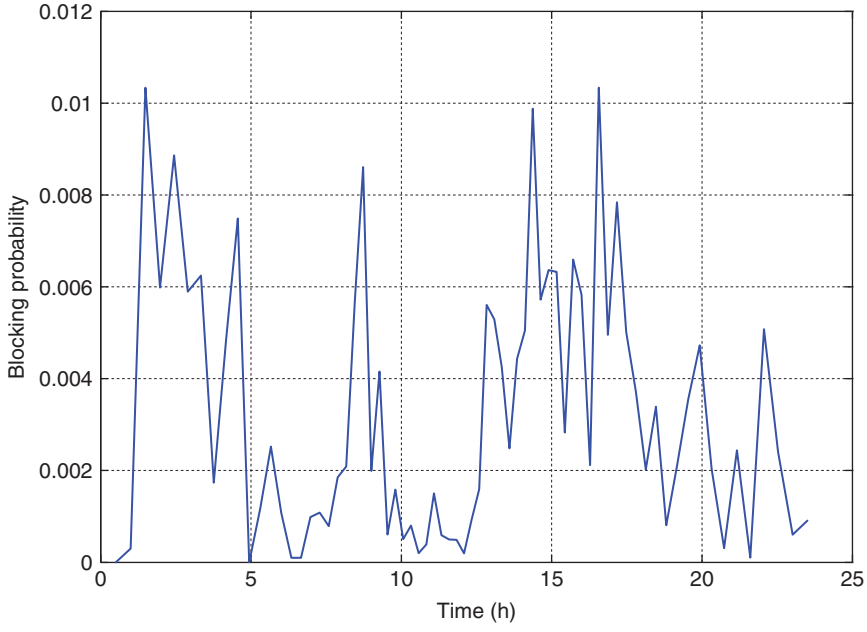

Figure 8.21 System blocking probability variation versus time

B. Comparing with Uniform BS Sleeping

We compare the proposed DP algorithm with the *uniform BS sleeping* approach proposed in Ref. [12], where active BSs are uniformly located in the network. We modify the uniform algorithm from binary pattern to multiple pattern. The number of active BSs are determined according to

$$N_{\text{on}}(t) = 55 + 5 \times \left\lfloor \frac{5(\lambda_w(t) - \lambda_{\min})}{\lambda_{\max} - \lambda_{\min}} \right\rfloor \tag{8.37}$$

which is a function of average traffic intensity $\lambda_w(t)$, $\lambda_{\min} < \lambda_w(t) < \lambda_{\max}$. As a result, $N_{\text{on}}(t) \in \{55, 60, \cdots, 75\}$. Here $\lfloor \cdot \rfloor$ rounds the real number to the nearest integer no larger than it. The switch cost of the DP algorithm is fixed $E_s = 2.5 \times 10^5$ J/switch.

In Figure 8.22, the average number of active BSs and the average blocking probability are compared. We decrease the value of α_1, α_2, and α_3 to enhance the degree of asymmetric traffic distribution. By matching BS sleeping pattern with traffic intensity distribution from both time domain and space domain, the proposed DP algorithm outperforms the uniform one. Specifically, as the traffic distribution becomes more and more asymmetric, the performance gap becomes larger. That is, the gap of the average number of active BSs increases from 1.27 to 3.65, and the average blocking probability ratio of the uniform algorithm to the DP algorithm increases from 3.63 to 41.3.

The figure also shows the following result, which is a little bit surprising: both the number of active BSs and the blocking probability of the proposed DP algorithm decrease as the parameters α_1, α_2, and α_3 decrease. As a matter of fact, the blocking events mainly come from two aspects: (i) blocking caused by overloading in the hotspot cells; (ii) blocking caused

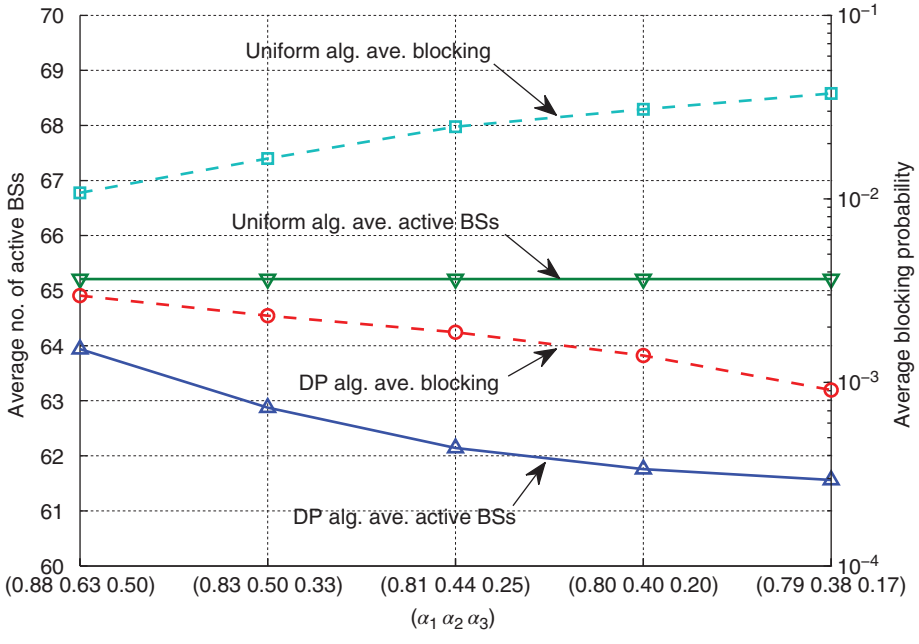

Figure 8.22 Comparison of proposed DP BS sleeping algorithm and uniform BS sleeping algorithm

by the high bandwidth requirement of the users in sleeping cells, which are denoted as *coverage edge* users. In current simulation settings, the system is in low traffic scenario. As a consequence, the blocking events caused by coverage edge users outweigh those caused by overloaded hotspot. As the traffic distribution becomes more and more asymmetric, more and more users are taken care of in the hotspots, and the number of coverage edge users becomes less. Hence, the blocking probability becomes lower and more BSs can sleep. Note that if the system traffic is extremely high on the contrary, the blocking events happened in hotspots outweigh those caused by coverage edge users, the blocking probability might increase.

C. Switching Cost

Extensive simulations are run to test the influence of the switching cost E_s on the performance. We set $\alpha_1 = 0.88, \alpha_2 = 0.63, \alpha_3 = 0.50$ for the simulations.

With different switching cost $E_s = 0, 2.5 \times 10^5, \ldots, 1 \times 10^6$ (J/switch), the average numbers of active BSs in the period of 24 hours are 62.9, 63.9, 65.2, 66.0, and 67.2 respectively. The blocking probability and the dropping probability are depicted in Figure 8.23. The proposed load balancing based algorithms for user association and handover are compared with the strongest signal based ones, where the selection criterion is simply $C(l_{mk})$ and the selection process is similar with **Algorithms 3** and **4**. The result shows that by effectively utilizing the wireless resources, load balancing is helpful for reducing the number of blocking and dropping events. It also illustrates that with the increase of E_s, the energy consumption increases, while both the blocking probability and the dropping probability decrease. It can be concluded that

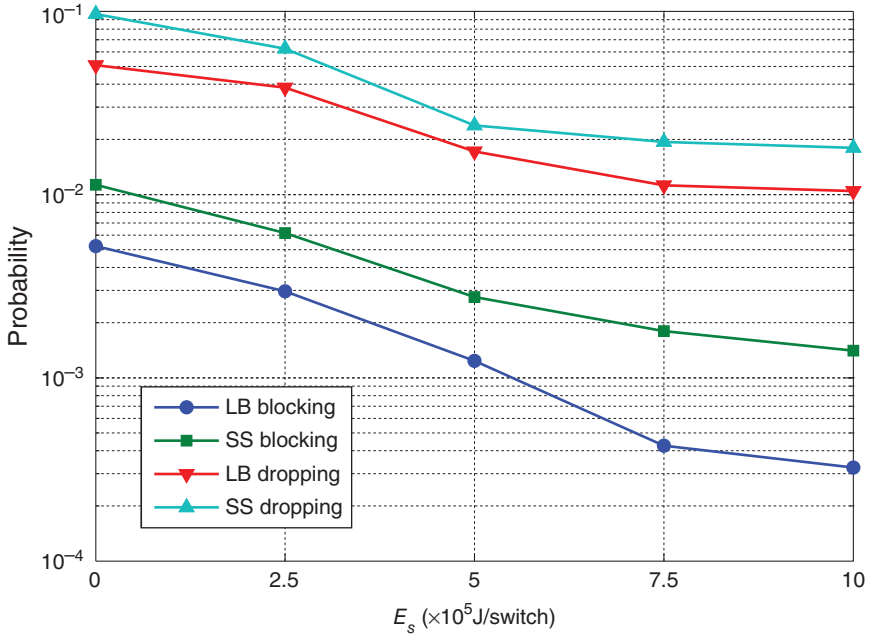

Figure 8.23 System blocking probability and dropping probability versus switching cost E_s. LB: proposed load balancing based BS selection and user handover algorithm; SS: strongest signal based algorithm

Table 8.1 Handover performance with different switching cost

E_s (J/switch)	0	2.5×10^5	5×10^5	7.5×10^5	1×10^6
No. of switching/stage	32.6	10.0	4.06	1.85	1.63
No. of handover/stage	385.7	108.5	41.74	17.73	16.03
No. of dropping/stage	19.6	4.16	0.72	0.20	0.17

there is a trade-off between the energy saved from turning BSs into sleep mode and the energy cost of BSs' mode switching. Because high switching cost prevents the BS switching from active to sleep, blocking probability is reduced.

Figure 8.24 shows the cumulative distribution function (CDF) of BS mode holding time. Without the switching penalty, more than 70% of BSs' measured mode holding time is less than 1 h. Such frequent mode switching maybe not acceptable for BS equipments in the real system. It not only consumes additional energy, but also brings large amount of handover, which causes exploding signaling overhead and user QoS degradation (shown in Table 8.1). This result explains the necessity of integrating switching cost into the total cost. As the value of E_s increases, the BS mode holding time increases accordingly, which shows that our algorithm well balances the trade-off between energy saving from sleep and cost from switching.

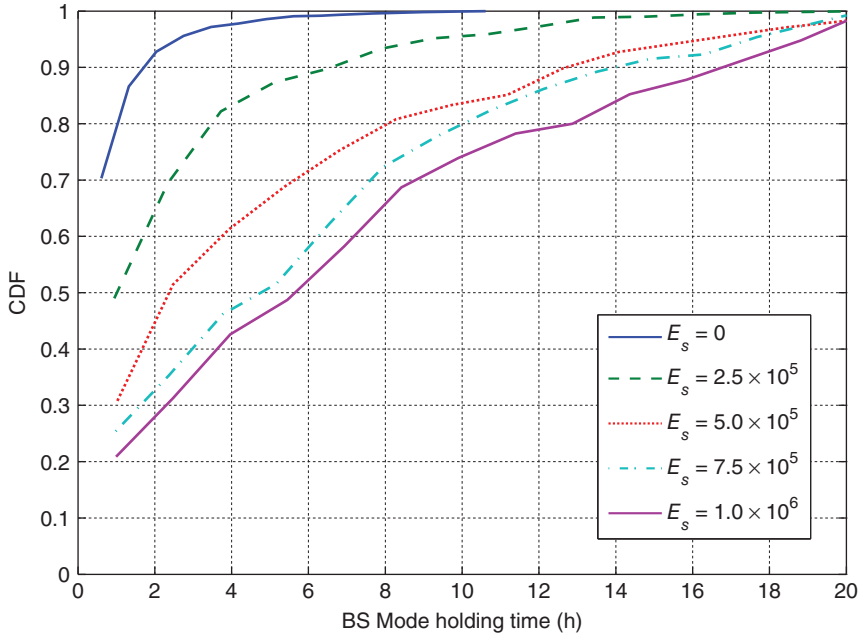

Figure 8.24 Cumulative distribution function of mode holding time with different switching cost E_s (J/switch)

8.4 Derivation of Area Blocking Probability

In this section, we ignore the stage index i for simplicity. According to the fact that a newly arrived user in cell m is blocked if all the active BSs in $\mathcal{B}_m$ are out of bandwidth, we have

$$\begin{aligned} P_a(A_m, \tilde{\boldsymbol{s}}_m) &= \Pr\left(\bigcap_{n\in\mathcal{B}_m, s_n=1} W_n \geq W_n^{\max}\right) \\ &= \prod_{n\in\mathcal{B}_m, s_n=1} \Pr(W_n \geq W_n^{\max}) \end{aligned} \tag{8.38}$$

where the second equality holds because the bandwidth utilizations of BSs are independent of each other.

If $s_n = 1, n \in \mathcal{B}_m$, the users in cell n can be served by BS n. The density of users in cell n is $K_n/\pi R_c^2$, where K_n is the number of users in cell n. Under the assumption that the user distribution in each cell is uniform, the bandwidth utilization of BS n for the users in cell n is calculated as

$$\begin{aligned} W_{nn} &= \int_0^{R_c} w_{nk} \frac{K_n}{\pi R_c^2} 2\pi l \mathrm{d}l \\ &= K_n \frac{2r_0}{R_c^2} \int_0^{R_c} \frac{l \mathrm{d}l}{C(l)} \end{aligned} \tag{8.39}$$

If $s_j = 0, j \in \mathcal{B}_m$, due to the nature of load balancing technique, the area of cell j is evenly assigned to its neighbor active BSs. Assume that the shifted traffic are uniformly distributed in the $1/B$ annulus sector with larger radius R_j and smaller radius R_c. The bandwidth requirement of the shifted traffic is

$$\begin{aligned} W_{jn} &= \int_{R_c}^{R_j} w_{nk} \frac{K_j}{\pi R_c^2} \frac{2\pi}{B} l \mathrm{d}l \\ &= K_j \frac{2r_0}{BR_c^2} \int_{R_c}^{R_j} \frac{l \mathrm{d}l}{C(l)} \end{aligned} \tag{8.40}$$

where R_j is set to evenly assign the cell area to its neighbor active BSs, i.e.,

$$R_j = R_c \sqrt{\frac{B}{I_j M_j^{\mathrm{on}}} + 1} \tag{8.41}$$

where $M_j^{\mathrm{on}} = \sum_{j' \in \mathcal{B}_j \cap \mathcal{B}_m} s_{j'}$ is the number of active neighbor BSs of sleeping BS j, $I_j = 1$ if $j = m$ and $I_j = 2$ if $j = m(b), b = 1, \ldots, B$. Note that local information sets $\tilde{\boldsymbol{s}}_m$ and $\tilde{\boldsymbol{\lambda}}_m$ do not contain the full local information of BS $m(b), b = 1, \ldots, B$. Therefore, we introduce the parameter I_j assuming that the local information of BS $m(b)$ is symmetric, that is, the state and the arrival rate of BS $j' \in \mathcal{B}_{m(b)} \backslash \mathcal{B}_m$ are the same as those of BS $j' \in \mathcal{B}_{m(b)} \cap \mathcal{B}_m$.

In summary, the bandwidth utilization of active BS n is

$$\begin{aligned} W_n &= W_{nn} + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} W_{jn} I_n \\ &= K_n' \gamma_n \end{aligned} \tag{8.42}$$

where $K_n' = K_n + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} I_n K_j / (I_j M_j^{\mathrm{on}})$ is the total number of user served by BS n, and

$$\gamma_n = \frac{\frac{2r_0}{R_c^2} \left(\int_0^{R_c} \frac{l \mathrm{d}l}{C(l)} + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} \frac{\lambda_j I_n}{B \lambda_n} \int_{R_c}^{R_j} \frac{l \mathrm{d}l}{C(l)} \right)}{1 + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} \frac{\lambda_j I_n}{I_j M_j^{\mathrm{on}} \lambda_n}}, \tag{8.43}$$

where we make use of the fact that $K_n / \lambda_n = K_j / \lambda_j$.

At the same time, the traffic load in cell n is evenly shifted to its neighbor active BSs. Similarly, we assume that half of the traffic of BS $m(b), b = 1, \ldots, B$ is shifted to accessible active BSs in $\mathcal{B}_m$. As a result, the traffic load of BS $n (s_n = 1)$ is

$$\lambda_n' = \lambda_n + \sum_{j \in \mathcal{B}_n \cap \mathcal{B}_m, s_j = 0} \frac{\lambda_j}{I_j M_j^{\mathrm{on}}} \tag{8.44}$$

As the radio resource is shared by active users, the number of users K_n' associated with BS n evolves like the number of customers in a processor-sharing queue with Poisson arrivals and i.i.d. service times [27]. The key property of the processor-sharing queue is that the stationary distribution of the number of customers is insensitive to the distribution of service times. Hence, the stationary distribution of the number of active users is given by

$\Pr(K'_n = k) = (\rho_n)^k(1-\rho_n)$ with mean $E[K'_n] = \rho_n/(1-\rho_n)$, where ρ_n is the average traffic load of BS n. Applying Little's law [28], we get $E[K'_n] = \lambda'_n/\mu$, which results in $\rho_n = \lambda'_n/(\lambda'_n + \mu)$. Finally, the area blocking probability is expressed as

$$\begin{aligned} P_a(A_m, \tilde{\mathbf{s}}_m) &= \prod_{n\in\mathcal{B}_m, s_n=1} \Pr(K'_n \geq W_n^{\max}/\gamma_n), \\ &= \prod_{n\in\mathcal{B}_m, s_n=1} \rho_n^{\lceil W_n^{\max}/\gamma_n \rceil} \end{aligned} \tag{8.45}$$

Summarizing the equations derived earlier, we obtain the expression of the approximated area blocking probability.

References

[1] Cisco, "Global mobile data traffic forecast update 2010–2015."

[2] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity, part i: system description," IEEE Trans. Commun., vol. 51, no. 11, pp. 1927–1938, 2003.

[3] S. Shamai and B. M. Zaidel, "Enhancing the cellular downlink capacity via co-processing at the transmitter end," in *Proceedings of VTC 2001 Spring*, 2001.

[4] H. Huang, M. Trivellato, A. Hottinen, M. Shafi, P. J. Smith, and R. Valenzuela, "Increasing downlink cellular throughput with limited network mimo coordination," IEEE Trans. Wireless Commun., vol. 8, no. 6, pp. 2983–2989, 2009.

[5] 3GPP, "Further advancements for E-UTRA: physical layer aspects," Technical Specification on Group Radio Access Network (Release 9), TR 36.814 V 1.2.1, Third Generation Partnership Project (3GPP), 2012.

[6] W. Song and W. Zhuang, "Multi-service load sharing for resource management in the cellular/WLAN integrated network," IEEE Trans. Wireless Commun., vol. 8, no. 2, pp. 725–735, 2009.

[7] A. Yeh, S. Talwar, G. Wu, N. Himayat, and K. Johnsson, "Capacity and coverage enhancement in heterogeneous networks," IEEE Wireless Commun. Mag., vol. 18, no. 3, pp. 132–139, 2011.

[8] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Towards dynamic energy-efficient operation of cellular network infrastructure," IEEE Commun. Mag., vol. 49, no. 6, pp. 56–61, 2011.

[9] S. Haykin, "Cognitive radio: brain-empowered wireless communications," IEEE J. Sel. Areas Commun., vol. 23, no. 2, pp. 201–220, 2005.

[10] A. Ghasemi and E. S. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," IEEE Commun. Mag., vol. 46, no. 4, pp. 32–39, 2008.

[11] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: an information theoretic perspective," Proc. IEEE, vol. 97, no. 5, 2009.

[12] M. A. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Optimal energy savings in cellular access networks," in *Proceedings of the IEEE ICC'09 Workshop, GreenComm.*, June 2009.

[13] A. P. Jardosh, K. Papagiannaki, E. M. Belding, K. C. Almeroth, G. Iannaccone, and B. Vinnakota, "Green WLANs: on-demand WLAN infrastructures," Mob. Networks Appl., vol. 14, no. 6, pp. 798–814, 2009.

[14] Z. Niu, Y. Wu, J. Gong, and Z. Yang, "Cell zooming for cost-efficient green cellular networks," IEEE Commun. Mag., vol. 48, no. 11, pp. 74–79, 2010.

[15] Z. Niu, L. Long, J. Song, and C. Pan, "A new paradigm for mobile multimedia broadcasting based on integrated communication and broadcast networks," IEEE Commun. Mag., vol. 46, no. 7, pp. 126–132, 2008.

[16] X. Wang, Y. Bao, X. Liu, and Z. Niu, "On the design of relay caching in cellular networks for energy efficiency," in *IEEE INFOCOM'11 workshop*, April 2011.

[17] S. Zhou, J. Gong, Z. Niu, Y. Jia, and P. Yang, "A decentralized framework for dynamic downlink base station cooperation," in *Proceedings of IEEE Globecom'09*, December 2009.

[18] J. Gong, S. Zhou, and Z. Niu, "A dynamic programming approach for base station sleeping in cellular networks," IEICE Trans. Commun., vol. 95B, no. 2, 2012.

[19] S. Zhou, J. Gong, and Z. Niu, "Distributed adaptation of quantized feedback for downlink network mimo systems," IEEE Trans. Wireless Commun., vol. 10, no. 1, pp. 61–67, 2011.

[20] M. Chiang, P. Hande, T. Lan, and C. W. Tan, "Power control in wireless cellular networks," *Foundations and Trends*® in Networking, vol. 2, no. 4, pp. 381–533. Now Publishers, 2008.
[21] L. Chiaraviglio, D. Ciullo, M. Meo, and M. A. Marsan, "Energy-aware UMTS access networks," in *Proceedings of the 11th International Symposium on Wireless Personal Multimedia Communications*, September 2008.
[22] H. Karl, "An overview of energy-efficiency techniques for mobile communication systems." Report of AG Mobikom WG7, September 2003. [Online]. Available: http://www.tkn.tu-berlin.de/\break fileadmin/fg112/Papers/TechReport_03_017.pdf.
[23] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz, "Primary user behavior in cellular networks and implications for dynamic spectrum access," IEEE Commun. Mag., vol. 47, no. 3, pp. 88–95, 2009.
[24] D. P. Bertsekas, Dynamic Programming and Optimal Control, 3rd ed., Massachusetts: Athena Scientific, 2005.
[25] A. Sang, X. Wang, M. Madihian, and R. Gitlin, "A load-aware handoff and cell-site selection scheme in multi-cell packet data systems," in *Proceedings of Globecom'04*, pp. 3931–3936, December 2004.
[26] Ericsson, "Radio characteristics of the ITU test environments and deployment scenarios," [Online]. Available: http://ftp.3gpp.org/tsg_ran/WG1_RL1/TSGR1_56b/Docs/R1-091320.zip.
[27] T. Bonald and A. Proutiere, "Wireless downlink data channel: user performance and cell dimensioning," in *Proceedings of MobiCom'03*, pp. 339–352, September 2003.
[28] L. Kleinrock, Queueing Systems, John Wiley & Sons, 1976.