AMERICAN UNIVERSITY OF BEIRUT

ENERGY-AWARE COOPERATIVE CONTENT DISTRIBUTION WITH DYNAMIC TOPOLOGIES

by
MOSTAFA KHALIL DIKMAK

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
September 2014

AMERICAN UNIVERSITY OF BEIRUT

ENERGY-AWARE COOPERATIVE CONTENT DISTRIBUTION WITH DYNAMIC TOPOLOGIES

by

MOSTAFA KHALIL DIKMAK

Approved by:

Dr. Zaher Dawy, Professor Advisor
Electrical and Computer Engineering

Dr. Hassan Artail, Professor Member of Committee
Electrical and Computer Engineering

Sanaa

Dr. Sanaa Sharafeddine, Associate Professor Member of Committee
Computer Science and Mathematics, Lebanese American University

Date of thesis defense: September 29, 2014

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:	Dikmak	Mostafa	Khalil
	Last	First	Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, **three years after the date of submitting my thesis, dissertation, or project,** to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

Mostafa	03/10/2014
Signature	Date

Acknowledgements

I would like to express my gratitude to my thesis committee Dr. Zaher Dawy, Dr. Hassan Artail, and Dr. Sanaa Sharafeddine. Especially I would like to thank my thesis advisor Dr. Zaher Dawy for his useful comments, remarks, engagement through the learning process of my master's degree thesis. Furthermore, I would like to thank Dr. Lina Al-kanj for introducing me to the topic as well for the support on the way.

An Abstract of the Thesis of

Mostafa Khalil Dikmak for Master of Engineering
Major: Electrical and Computer Engineering

Title: Energy-Aware Cooperative Content Distribution With Dynamic Topologies

Multimedia services such as video streaming and online games have been widely spread with the rapid evolution of wireless communication technologies that provide high data rates. The prolonged usage of such services requires the wireless interfaces to be active for long periods, thus increasing the energy consumption which raises a challenge for the designers of battery-operated mobile terminals (MTs). Optimized cooperative content distribution approaches with device-to-device (D2D) cooperation, have been proposed in the literature to reduce the high energy consumption. However, most of these approaches do not take into consideration the variations in the network (MTs leaving, new MTs joining, and/or MTs moving around). This thesis is divided into two main parts. In the first part, we present pseudo-optimal energy-efficient cooperative content distribution algorithms that account for different types of mobility in the network. The content distribution problem is first formulated using two static optimization problems: minimum spanning tree (MST) formulation and two-hop integer programming optimization formulation. Both formulations return optimal solutions at a given point in time. However, upon any variation in the network, the content distribution solution needs to be updated. Since re-solving the optimization problems is computationally expensive, a re-optimization algorithm that returns sub-optimal solutions is proposed for each type of mobility (node insertion, node deletion, edge weight modification). Simulation results for various scenarios demonstrate performance close to the optimal, with major reduction in computational complexity. In the second part of the thesis, we discuss dynamic cooperative content distribution architectures with the following components: network formation, detection and selection mechanisms, and failure detection and recovery mechanisms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The evolution of technology towards 4G increases the subscribers' interest in multimedia services such as live video streaming, games, file download, etc. Multimedia applications require high data rates to provide quality of service (QoS) and delay guarantees which further increase the energy consumption at the nodes level. Moreover, the prolonged usage of multimedia services requires the wireless interfaces to be active for long periods, thus increasing the energy consumption. Accordingly, battery-operated nodes require batteries with longer lifetime than what existing battery technologies can provide. Therefore, the energy consumption of battery-operated nodes emerges as one of the major limitations in the design of the future wireless communication systems. Note that throughout this thesis, we mean by node any mobile terminal (MT) that could be an ordinary cell phone, a smart phone, or a tablet.

Towards solving the problem of high energy consumption, mobile-to-mobile cooperation has been suggested in the literature to decrease the energy consumption at the nodes [1] [2]. In addition, cooperative wireless networks have proved to bring other advantages such as increasing the network throughput [3] [4], extending the network coverage [5] [6], and decreasing the end user communication cost [7] [8]. Cooperative communications among nodes is expected to play a key role in the deployment of 4G networks [9] [10]. What makes such communication possible is the fact that almost all new mobile terminals (ordinary or smart phones) are equipped with at least two wireless network interfaces: one to communicate with the base station or access point using a Long Range (LR) technology such as GSM/GPRS, UMTS, WiFi, or LTE/LTE-Advanced and one to communicate with other nodes using a Short Range (SR) technology such as Bluetooth, WiFi ad hoc mode, or WiFi Direct.

Extensive work in the literature has tackled the issue of minimizing the energy consumption in a cooperative content distribution scenario, where a set of nodes in a given area are interested in a common content [1] [11] [12]. This content could be a video streaming session, a file download, or any type of multimedia services. In such a scenario, nodes that receive the content or part of it on the

LR exchange it with other nodes on the SR without the interference of the base station. In a non cooperative scenario, each node individually receives the content on the LR. What motivates the cooperation in wireless networks is the fact that SR technologies provide relatively high data rates and consume less energy at the nodes level compared to LR technologies. Therefore, cooperation can help in decreasing the energy consumption per node or the total energy consumption of all the nodes.

In [13], minimizing the energy consumption is formulated as a linear integer optimization problem. The solution of the optimization problem determines which nodes should receive the content on the LR and which should receive it on the SR. Thus, the obtained solution is a clustering of the existing nodes where each cluster contains one node that downloads the content on the LR and the nodes that receive the content from this node on the SR. However, this problem is NP-hard; the computational and time complexity to solve such a problem is high particularly when the number of nodes is large. Moreover, in [13], the network is considered static i.e., the mobility of nodes is not taken into consideration.

Many papers in the literature discuss the content distribution in a cooperative scenario taking into consideration the dynamic behaviour of the nodes [8] [14]. In [14], a cooperating ad-hoc networking protocol called $CHUM$ is presented. Each peer (MT) in a $CHUM$ takes turn serving as a proxy, which sets an Internet connection, downloads multimedia content via a telecommunication link, and broadcasts it to nearby peers using its ad hoc connection. The proxy role is rotated among all peers in a round robin fashion to share the communication cost. In [8], another collaborative streaming protocol called $COSMOS$ is proposed. In $COSMOS$, only a few peers pull video descriptions from base stations. Using a free broadcast channel (such as Wi-Fi and Bluetooth), they share the streams to nearby neighbours. A mechanism is employed to exchange the roles between pullers and passive receivers when a puller has been downloading video data for a certain period of time. When a MT wants to join the cooperative network, $CHUM$ and $COSMOS$ define specific mechanisms to add this new MT to the network. Moreover, recovery mechanisms are proposed to maintain the cooperative network when a MT leave the network. Both $CHUM$ and $COSMOS$ reduce the telecommunication cost taking into consideration the dynamic behaviour of the MTs by defining a mechanism for . However, energy minimization is not considered.

In this work, we consider the problem of finding optimized distribution of a given content among interesting nodes using mobile-to-mobile cooperation taking into consideration the minimization of the energy consumption of the cooperating nodes and the mobility in the network. In a realistic wireless network, nodes are mobile; nodes move in any direction with any speed. Nodes can leave or enter the network at any point in time. Moreover, the network itself is time-variant, i.e., the channel conditions on the LR and on the SR are changing over time. The dynamicity of nodes and the variation of the channel conditions over time make the static modelling of the cooperative content distribution problem unrealistic. Therefore, the aim of this thesis is to define a centralized framework that leads to pseudo-optimal energy-efficient distribution of the content among the nodes taking into consideration the different aspects of mobility in the network. Moreover, the cooperation among the nodes to distribute the common content requires the definition of efficient mechanisms to build the network and to maintain it. Therefore, a dynamic cooperative content distribution architecture, containing network formation, discovery and recovery mechanisms, is proposed.

This report is organized as follows. Chapter 2 provides a literature review and background information on topics of direct relevance to the thesis work. Chapter 3 presents the system model of the cooperative content distribution network and the static formulations of the cooperative content distribution optimization problem. Chapter 4 presents the problem formulation of this thesis. In Chapter 5, re-optimization cooperative content distribution framework is proposed and analysed. Chapter 6 presents different mechanisms of the cooperative content distribution architecture. Finally, Chapter 7 provides some concluding remarks.

Chapter 2

Literature Review

The objective of this literature survey is to investigate some previous work related to the thesis scope. We start in Section 2.1 by summarizing existing work on dynamic optimization schemes such as stochastic and robust network optimization. Then in Section 2.2, we discuss the modeling of energy consumption of the mobile terminals (MTs) needed in the formulation of the cooperative content distribution problem. In a cooperative content distribution network, cooperating mobile terminals (MTs) form an ad-hoc network to distribute the common content using a SR technology. Therefore, the cooperating MTs can be divided into clusters where in each cluster one MT receive the common content using a LR technology and other MTs receive this content from this MT. In Section 2.3, we present existing work on clustering algorithms used in ad-hoc networks. Moreover, in a cooperative scenario, a SR technology is used to support the establishment of ad-hoc networks. Therefore, IEEE 802.11 ad-hoc Mode and WI-FI Direct technology are presented in Section 2.4.

2.1 Optimization under Uncertainty

In a realistic wireless network, a node can move in any direction and with any speed and can at any point in time change its direction and its speed. In such a network, an existent node can leave the network and a new node can enter the network at any point in time. Moreover, in a realistic scenario, channel conditions are time-variant. Therefore, modeling a framework for cooperative content distribution requires us to take into account mobility of the nodes and the variation of the channel conditions over time. Stochastic network optimization and robust network optimization can be utilized to model the dynamic behaviour in wireless networks.

2.1.1 Stochastic Network Optimization

Stochastic networks are networks with random events, time variation, and uncertainty. Stochastic network optimization provides online control strategies for time varying networks with general classes of penalties, rewards, and utility functions. There exist several techniques for stochastic network optimization. Stochastic optimization (SO) methods are used extensively to model the dynamic behaviour in wireless networks [15] [16].

In [17], a technique is proposed to stabilize a multihop packet radio network using Lyapunov drift. In this technique, backpressure routing and maxweight scheduling principles are derived. Many dynamic algorithms for stability in wireless systems [16] [18] and mobile adhoc networks [19] were developed using this technique. In the proposed systems, previous knowledge of of traffic arrival rates or channel probabilities is not needed.

In [15], a dynamic control strategy for minimizing energy expenditure in a time varying wireless network with adaptive transmission rates is developed. It uses a simple Lyapunov drift technique that enables system stability and performance optimization to be achieved simultaneously. The algorithm operates without knowledge of traffic rates or channel statistics, and yields average power that is arbitrarily close to the minimum possible value achieved by an algorithm optimized with complete knowledge of future events.

2.1.2 Robust Network Optimization

Robust optimization is a field of optimization theory that deals with optimization problems in which a certain measure of robustness is sought against uncertainty that can be represented as deterministic variability in the value of the parameters of the problem itself and/or its solution. Most work that uses optimization theory to study communication and networking problems assumes that the network is static and that the information defining the constraints and the objective function of the optimization problem can be obtained precisely. However, in reality, these information are time-varying, or uncertain. Therefore, solving the static optimization problems may lead to unrealistic solutions.

The basic idea of the robust optimization is to seek a solution which remains near-optimal under the modification of parameters in the static optimization problem. Each robust optimization is defined by three-tuple: a nominal formulation, a definition of robustness, and an uncertainty set. In [20], optimization models of wireless sensor network (WSN) subject to distance uncertainty are considered for three classic problems in energy limited WSNs: minimizing the energy consumed, maximizing the data extracted, and maximizing the network lifetime. In a robust optimization model, the uncertainty is represented by considering that the uncertain parameters belong to a bounded, convex uncertainty set. A robust solution is the one with best worst case objective over this set. It is shown that solving for the robust solution in these problems is just as difficult as solving for the problem without uncertainty. Computational experiments show that, as the uncertainty increases, a robust solution provides a significant improvement in worst case performance at the expense of a small loss in optimality when compared to the optimal solution of a fixed scenario.

In [21], a distributed robust optimization model for communication networks is presented. Several models for describing parameter uncertainty sets that can lead to distributed solutions for linearly constrained nominal problems are described. These models include general polyhedron, D-norm, and ellipsoid. For robust rate control under link failures, the authors in [21] designed a fast sequential optimization algorithm based on distributed column generation method and dual decomposition. The algorithm can quickly converge to the optimal solution. The authors of [22] proposed a reliable mathematical model to optimally design a minimum-cost survivable telecommunication network that continues to support a good communication under any node failure scenario.

The stochastic network optimization and robust network optimization require to have an expectation of the variations in the network whether defined using stochastic series or as uncertainty sets. Moreover, the stochastic and robust network optimization problems are generally NP-complete or NP-hard [15] [20]. The time and computational complexities for solving such problems are high. However, the aim of this thesis is to define a realistic cooperative content distribution framework that returns online solutions in response to every modification in the network.

2.2 Energy Consumption Modeling

In order to model the energy consumption in cooperative content distribution architectures, it is required to model the energy consumed by nodes during data transmission and reception. There exist two main types of energy modeling in the literature. The first type of modeling considers that energy consumption in the nodes during sending or receiving follow a linear model that is function of the amount of data sent or received [23] [24]. In [23], the authors show experimentally that the energy consumption follows a linear behavior with the number of bits sent or received over a WLAN interface. The linear model of energy consumption has the following form:

$$E = N_B \cdot E_b + E_{base} \tag{2.1}$$

where N_B is the size of the bits sent or received, E_b is the energy consumed per bit while transmitting or receiving, and E_{base} is independent of the bits sent or received. In [24] experimental results are presented to verify the accuracy of the linear energy model for a WLAN interface.

The other type of energy consumption modeling is more comprehensive. It is shown in [25] that the energy consumption due to the baseband processing circuitry is quite small compared to the energy consumption due to the RF circuitry. Thus, the work in [26] considered only the RF circuit blocks of the transceiver and obtained the following expressions for the energy consumed per unit time during data transmission (P_{Tx}) and data reception (P_{Rx}):

$$P_{Tx} = P_t + P_{ct} \tag{2.2}$$

$$P_{Rx} = P_{cr} \tag{2.3}$$

where P_{ct} and P_{cr} are the circuitry power (energy per unit time) consumed during transmission and reception, respectively, and P_t is the power of the transmitted signal.

It is argued that the energy consumed per unit time during reception can be considered constant for a fixed transmission bandwidth. It is demonstrated in [27] that the energy consumed per unit time during transmission depends on the circuit energy consumption, which is constant for a fixed transmission bandwidth, and on the power of the transmitted signal. Moreover, for a fixed transmit signal power, the energy consumed per unit time during transmission can also be considered constant. Therefore, the transmit and receive energy consumption (E_{Tx} and E_{Rx}) can be formulated as follows:

$$E_{Tx} = P_{Tx} \cdot \tau_{Tx} \tag{2.4}$$

$$E_{Rx} = P_{Rx} \cdot \tau_{Rx} \tag{2.5}$$

where τ_{Tx} and τ_{Rx} are the total transmission and reception times respectively.

2.3 Ad-hoc Clustering Algorithms

In a typical cooperative content distribution scenario, a set of mobile terminals (MTs) download the common content directly from the base station (BS) using a LR technology link. Each of these MTs forwards this content to a number of MTs in its vicinity using SR technology links. Each MT that receives the content over a LR link forms a cluster with the MTs that receive this content from this MT. Therefore, each cooperative content distribution scenario can be divided into a number of clusters. Most of the static optimization problems [13] and the stochastic and robust network optimization problems [15] [20] are generally NP-complete or NP-hard. To overcome this drawback, the problem originally formulated as an optimization problem can be solved heuristically. In a heuristic, the original problem is simplified into many sub-problems that can be solved with relatively low time and computational complexity. Specifically for wireless networks, one way to simplify such a problem is to group the nodes into clusters. Clustering is an important research topic in the field of mobile ad-hoc networks (MANETs) and wireless sensor networks (WSNs). Clustering in MANETs and

WSNs is the assignment of a set of nodes into subsets called clusters in order to guarantee basic level of system performance such as throughput, delay, and energy minimization. A large variety of approaches for ad hoc clustering have been presented, whereby different approaches typically focus on different performance metrics.

Clustering protocols in MANETs and WSNs can be classified based on their objectives. Therefore clustering protocols can be classified into six different categories. Dominating-Set-based (DS-based) clustering [28] [29] tries to find a Dominating-Set (DS) for a network so that the number of mobile nodes that participate in route search or routing table maintenance can be reduced. Low-maintenance clustering schemes [30] [31] aim at providing stable cluster architecture for upper-layer protocols with little cluster maintenance cost. Mobility-aware clustering [32] [33] takes the mobility behaviour of mobile nodes into consideration. This is because the mobile nodes' movement is the main cause of changes to the network topology. Energy-efficient clustering [34] [35] manages to use the battery energy of mobile nodes more wisely in a MANET. Load-balancing clustering schemes [36] attempt to limit the number of mobile nodes in each cluster to a specified range so that clusters are of similar size. Thus, the network loads can be more evenly distributed in each cluster. Combined-metrics-based clustering [37] usually considers multiple metrics, such as node degree, cluster size, mobility speed, and battery energy, in cluster configuration, especially in clusterhead decisions.

Generally, MANETs and WSNs clustering algorithms can be divided according to different criteria: Clusterhead-based or Non-Clusterhead-based algorithms and One-hop or Multi-hop algorithms. Moreover, clustering algorithms can be categorized into centralized algorithms and distributed algorithms. Centralized algorithms require powerful centralized devices (Base stations or Access Points) to run the clustering algorithms whereas in distributed algorithms all the nodes participate in the algorithm. Centralized algorithms need small communication overheads while distributed algorithms require large communication overheads.

Several work used hierarchical clustering algorithms for ad-hoc networks [38] [39]. Hierarchical clustering is a cluster analysis algorithm that groups data over a variety of scales by creating a cluster tree or dendrogram. Hierarchical clustering algorithm is proved to have low time and computational complexity.

2.4 IEEE 802.11 Ad-hoc Mode and WI-FI Direct Technology

In a cooperative content distribution network, a set of mobile terminals (MTs) receives the common content using the LR technology and forwards this content to other MTs using the SR technology. Therefore, the MTs that communicates with each other over SR links need to establish an ad-hoc network without the administration of any access point.

The IEEE 802.11 standard [40] provides two different modes of operations: Ad-hoc mode and infrastructure mode. The infrastructure mode requires the direct administration of an 802.11 access point. However, using the ad-hoc mode, a set of MTs are able to establish and use an ad-hoc network without the intervention of any access point. The use of ad-hoc mode only affects the protocols, so there is no impact on the Physical Layers (i.e., 802.11a and 802.11b). Within the MAC Layer, all of the carrier sensing and most of the frame types and corresponding usage are the same regardless of which mode you choose. The absence of an access point, however, means that an ad-hoc wireless LAN must take on more of the MAC Layer responsibilities.

Using the ad hoc mode, the first active MT establishes an Independent Basic Service Set (IBSS) and starts sending beacons, which are needed to maintain synchronization among the MTs. (With infrastructure mode, only the access point sends beacons.) Other MTs can join the network after receiving a beacon and accepting the IBSS parameters (e.g., beacon interval) found in the beacon frame. All MTs that join the ad-hoc network must send a beacon periodically if it doesn't hear a beacon from another MT within a very short random delay period after the beacon is supposed to be sent. The random delay minimizes the transmission of beacons from multiple MTs by effectively reducing the number of MTs that will send a beacon. If a MT doesn't hear a beacon within the random delay period, then the MT assumes that no other MTs are active and a beacon needs to be sent. After receiving a beacon, each MT updates their local internal clock with the timestamp found in the beacon frame, assuming the timestamp value is greater than the local clock. This ensures that the all MTs are able to perform operations, such as beacon transmissions and power management functions, at the same time.

Wi-Fi Direct [41] [42] is a new technology defined by the Wi-Fi Alliance wherein capable devices can connect directly to each other quickly, securely and conveniently to do tasks such as printing, synchronization, and sharing of data. Wi-Fi Direct technology, as described in Wi-Fi Peer-to-Peer (P2P) Technical Specification, takes a different approach, to enhance device to device connectivity. Instead of leveraging the ad-hoc mode of operation, Wi-Fi Direct builds upon the successful IEEE 802.11 infrastructure mode and lets devices negotiate who will take over the AP-like functionalities. Thus, enables legacy Wi-Fi devices to connect to the Wi-Fi Direct network that may have not been possible otherwise.

Chapter 3

Cooperative Content Distribution: System Model and Static Problem Formulation

In this chapter, the system model for cooperative content distribution over wireless network is presented. The cooperative content distribution scenario is formulated using two different static approaches: Minimum spanning tree (MST) integer programming optimization problem and two-hop tree integer programming optimization problem. The re-optimization algorithms proposed in Chapter 5 to account for the mobility in the network are based on the static formulations presented in this chapter.

3.1 System Model

The proposed system model consists of a cellular base station (BS) connected by a LR link to a group of mobile nodes as shown in Figure 3.1 . Additionally, the nodes are connected to each other through a SR links using a short-range high bit-rate wireless technology such as WLAN or Bluetooth. The nodes can move in any direction and with any speed. Moreover, existent nodes can leave the network and new nodes can enter it. It is assumed that the BS is connected to the server that holds the content via a wired infrastructure.

The network consists of a single BS and K cooperating nodes in the range of the BS. The base station (BS) is responsible for transmitting a given content to a single receiver (unicast) over wireless fading channels. The content could be live video streaming, game download, or file download.

In a non-cooperative scenario, the server separately unicasts the content to each requesting node. With unicast, the network resources over the LR link are used redundantly for each node. In general, a non-cooperative scenario is believed to be more costly in terms of throughput and energy consumption than an optimized

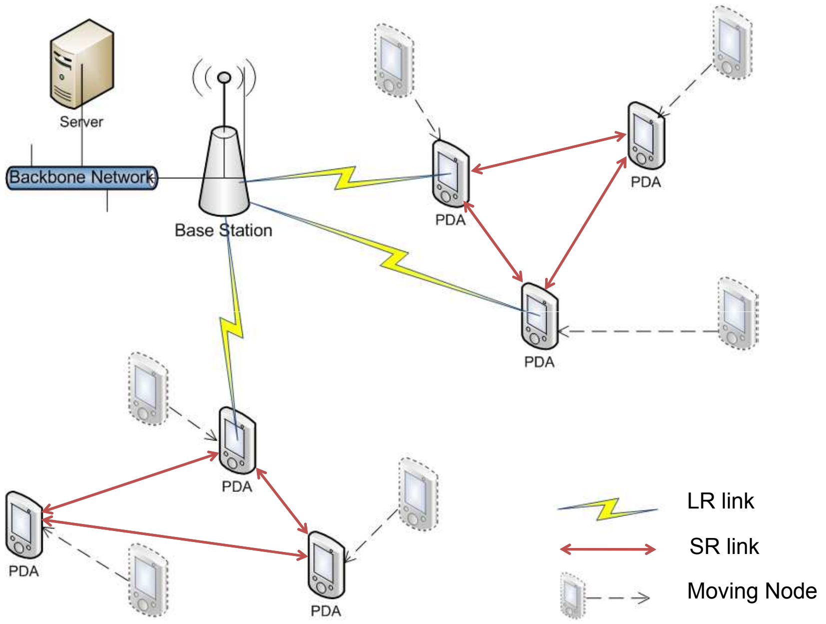


Figure 3.1: System model.

cooperative scenario.

In a cooperative scenario, nodes cooperate with each other over SR wireless links that are more energy efficient than the LR links. In this scheme, some nodes receive the content using the LR technology. Other nodes receive the content from other cooperating nodes in the mobile-to-mobile (M2M) network over the SR links. Since SR links are more energy efficient than LR links, the exchanged parts require lower reception power than receiving them directly on the LR. However, in this case, each node has to spend additional energy to transmit its data parts to the other cooperating nodes. Therefore, it is not directly clear whether the aggregate energy consumed at the nodes would be less or more when compared to traditional content streaming architectures.

Assumptions

In this system model, we assume that channels of the LR and SR technologies are orthogonal and are modelled by pathloss and shadowing. In [43], the received power P_r and the transmitted power P_t are represented using the following formula:

$$\frac{P_r}{P_t}(dB) = 10\log_{10}\kappa - 10\upsilon\cdot\log_{10}\frac{d}{d_0} + h_{\mathrm{dB}} \qquad (3.1)$$

where κ is a constant that depends on the average channel attenuation, υ is path loss exponent, d_0 is a reference distance, d is the distance where the received power is calculated (in this model this parameter represents the distance between the MT and the BS or between two communicating MTs based on the wireless technology used), and h is a Gaussian random variable representing shadowing or slow fading having a variance $\sigma^2_{h_{dB}}$.
The bit rates are estimated as follows: Given target P_e and the signal to noise ratio γ, in [43], the highest order $M-QAM$ modulation scheme that can be used is presented in the following formula:

$$P_e \leq 0.2e^{\frac{-1.5\gamma}{M-1}} \tag{3.2}$$

Assuming the the symbol rate $R_S = 1/W$ where W is the passband bandwidth of the channel, the bit rate $R = \log_2(M) \cdot R_S$.

3.2 Minimum Spanning Tree Static Formulation

To account for the dynamicity of nodes in the network, we start by formulating the problem as a Minimum Spanning Tree (MST) problem. The MST problem is a celebrated problem, useful to model any kind of networks in transports, communications, energy, etc. The following is the minimum spanning tree integer programming formulation of the content distribution problem with a unicast-unicast scenario.
Given an undirected graph G(V,E) connecting the nodes and the BS where V is the set of vertices and E is the set of edges connecting the nodes with each other and with the BS. The number of vertices in this graph is K ($K-1$) nodes and the BS). The total number of edges is $K(K-1)/2$: $K-1$ edges between the BS and the nodes and $(K-1)(K-2)/2$ edges among the nodes. A weight is assigned to each edge in the graph. In this formulation, the weights reflect the energy consumed by the nodes while transmitting and receiving data. The definition of the weights follows the energy model presented in Section 2.2. The weight of an edge between a node M_k and the BS $w_{k,LR}$ is the energy consumed by the node to download the content on the LR.

$$w_{k,LR} = \frac{S_T}{R_{L,k}} \cdot P_{Rx,L} \tag{3.3}$$

where S_T is the size of the content in Mbits, $R_{L,k}$ is the transmission rate (Mbits/s) on the LR links from the BS to M_k and $P_{Rx,L}$ is the power (Joules/s) consumed by the node during reception on the LR.
The weight of an edge between node M_k and node M_j $w_{kj,SR}$ is the energy consumed by the transmitting node to transmit the content and the receiving node to receive it.

$$w_{kj,SR} = \frac{S_T}{R_{S,kj}} \cdot P_{Tx} + \frac{S_T}{R_{S,kj}} \cdot P_{Rx,S} \tag{3.4}$$

where $R_{S,kj}$ is the transmission rate on the SR links from M_k to M_j, P_{Tx} is the power consumed by M_k while transmitting to M_j on the SR interface, and $P_{Rx,S}$ is the power consumed by the node during reception on the SR.

The decision variables for the integer programming (IP) formulation of MST are: $x_e = 1$ if edge $e \in T$ and $x_e = 0$ otherwise ($e \in E$ where E is the set of all edges in graph E). The content distribution is represented by a vector $\mathbf{x} = (x_1, x_2, ..., x_{|E|})$ where $x_e = x_{k,LR}$ if the edge e is connecting a node k with the BS over a LR link, and $x_e = x_{kj,SR}$ if the edge e is connecting a node k to node j over a SR link ($x \in Z^+_{|E|}$ and $|E|$ is the number of edges in set E). The vector $\mathbf{w} = (w_1, w_2, ..., w_{|E|})$ is the vector of weights of each edge in set E where $w_e = w_{k,LR}$ if the edge e is connecting a node k with the BS over a LR link, and $w_e = w_{kj,SR}$ if the edge e is connecting a node k to node j over a SR link.

In the case of unicasting, the total energy consumed in a cooperative scenario using the MST formulation is

$$E_{coop,U} = \sum_{e \in E} x_e w_e = \sum_{k=1}^{K} x_{k,LR} w_{k,LR} + \sum_{k=1}^{K} \sum_{j=1, j \neq k}^{K} x_{kj,SR} w_{kj,SR} \tag{3.5}$$

The IP formulation of MST is as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \sum_{\text{e} \in \text{E}} w_e x_e \\ \text{subject to} \quad & \sum_{e \in E} x_e = K \\ & \sum_{e \in (S,S)} x_e \leq |S| - 1, \ \forall S \subseteq V \\ & x_e \in \{0, 1\}, \ \forall e \in E \end{aligned}$$

where (S, S) denotes all edges that go from a node in the set S to another node in the set S where S could be any set of nodes of V ($S \subseteq V$).

The solution of this problem is a minimum weight spanning tree T. T must satisfy the following tree conditions: have $(K + 1) - 1$ edges (first condition in the IP formulation), be connected and be acyclic (second condition in the IP formulation).

The most famous greedy algorithms to solve the minimum spanning tree problem are Kruskal's algorithm [44] and Prim's algorithm [45].

Following is the Kruskal's algorithm pseudo-code to find a MST for an undirected graph G(V,E).

1. let A be the MST of G, $A = \emptyset$

2. sort the edges in E of the graph G into non-decreasing order by weight
3. for each edge $e(u, v) \in E$ connecting vertices u and v taken in non-decreasing order by weight
    - if vertices u and v are not already connected, set $A = A \cup e(u, v)$
4. return A

The complexity of the Kruskal's algorithm is $O(|E| \log(|E|))$ and the complexity of Prim's algorithm is $O(|E| + |V| \log |V|)$ where $|E|$ is the number of edges in the graph and $|V|$ is the number of vertices in the graph [46]. Prim's algorithm is significantly faster in dense graphs. However Kruskal's algorithm performs better is sparse graphs. Therefore, Kruskal's algorithm is used throughout this work to find the static optimal solution for the MST formulation.

3.3 Two-hop Tree Static Formulation

In [13], the problem of optimizing energy consumption in a content distribution scenario is tackled. In the following formulation, the network is considered static. The optimization problem is formulated as follows. Consider K requesting nodes interested in downloading a content from a server on the internet in a cooperative manner. Assume that the content is divided into N parts considered equal in size and importance. Therefore, the data D_k to send by the BS to node M_k is equal to:

$$D_k = \frac{x_k \cdot S_T}{N} \tag{3.6}$$

where x_k is the number of parts received by node M_k over the LR link, S_T is the size of the content to be sent in one transmission interval, and N is the number of parts the content is divided into.
The time t_k required to send the data D_k over a link with rate R_k is:

$$t_k = \frac{D_k}{R_k} = \frac{x_k \cdot S_T}{N.R_k} \tag{3.7}$$

Given the general energy formula $E = P \cdot t$ with P being the power consumption and t the time spent, the energy $E_{L,k}$ consumed by M_k for receiving over the LR is:

$$E_{L,k} = \frac{D_k}{R_k} \cdot P_{Rx,L} = \frac{x_k \cdot S_T}{N.R_{L,k}} \cdot P_{Rx,L} \tag{3.8}$$

where $R_{L,k}$ is the transmission rate on the LR links from the BS to M_k and $P_{Rx,L}$ is the power consumed by the node during reception on the LR.
The energy $E_{S,Tx,kj}$ to send this data on the SR from device k to device j is

$$E_{S,Tx,kj} = \frac{D_k}{R_k} \cdot P_{Tx} = \frac{x_k \cdot S_T}{N.R_{S,kj}} \cdot P_{Tx} \tag{3.9}$$

where $R_{S,kj}$ is the transmission rate on the SR links from M_k to M_j and P_{Tx} is the power consumed by M_k while transmitting to M_j on the SR interface.
Note that the node M_k should transmit to all the $(K-1)$ devices within its group, it will have to make N_{tr} SR transmissions; Since unicasting is used, N_{tr} is $(K-1)$.
Following the same reasoning for computing the energy consumed by M_k while receiving description x_j from M_j $E_{S,Rx,jk}$ we get,

$$E_{S,Rx,jk} = \frac{x_k \cdot S_T}{N \cdot R_{S,jk}} \cdot P_{Rx,S} \tag{3.10}$$

where $R_{S,jk}$ is the transmission rate on the SR links from M_j to M_k and $P_{Rx,S}$ is the power consumed by the node during reception on the SR.
Thus, the overall energy consumed by node M_k while transmitting and receiving on the SR interface is:

$$E_{S,k} = \sum_{j=1,k\neq j}^{N_{tr}} E_{S,Tx,kj} + \sum_{j=1,k\neq j}^{K} E_{S,Rx,jk} \tag{3.11}$$

The total energy consumed by a node M_k on the LR and SR links is:

$$E_k = E_{L,k} + E_{S,k} \tag{3.12}$$

In the case of unicasting, the energy consumed by a M_k is:

$$E_{k,U} = \frac{x_k \cdot S_T}{N.R_{L,k}} \cdot P_{Rx,L} + \frac{x_k \cdot S_T}{N} P_{Tx} \sum_{j=1,k\neq j}^{K} \frac{1}{R_{S,kj}} + \frac{S_T}{N} P_{Rx,S} \sum_{j=1,k\neq j}^{K} \frac{x_j}{R_{S,jk}} \tag{3.13}$$

The total energy consumed by the requesting nodes is:

$$E_{coop} = \sum_{k=1}^{K} E_k = \sum_{k=1}^{K} [E_{L,k} + E_{S,k}] \tag{3.14}$$

In the case of unicasting, the total energy consumed by the requesting nodes is:

$$E_{coop,U} = \frac{S_T}{N} \cdot P_{Rx,L} \sum_{k=1}^{K} \frac{x_k}{R_{L,k}} + \frac{S_T}{N} (P_{Tx} + P_{Rx,S}) \sum_{k=1}^{K} \sum_{j=1,j\neq k}^{K} \frac{x_k}{R_{S,kj}} \tag{3.15}$$

The total energy consumption spent when no cooperation takes place is:

$$E_{No-coop} = S_T \cdot P_{Rx,L} \sum_{k=1}^{K} \frac{1}{R_{L,k}} \tag{3.16}$$

In this problem, the objective is to minimize total energy consumption of the nodes. A constraint that guarantees that the whole content is transmitted on the LR should be added. Thus, the optimization problem becomes:

$$
\begin{aligned}
\underset{\mathbf{x}}{\text{minimize}} \quad & \mathrm{E}_{\mathrm{coop}} \\
\text{subject to} \quad & \sum_{k=1}^{K} x_k = N \\
& x \succeq 0 \\
& x : int.
\end{aligned}
$$

Solving this optimization problem, the solution determines the number of parts of the content received by each node on the LR and the number of parts of the content received from every other node on the SR. This problem is NP-hard [13]. It has high time and computational complexities. The solution of this optimization problem is a set of two-hop clusters where in each cluster one node downloads the content from the BS using a LR link and then forwards the content to other nodes in the cluster using SR links. To reduce the complexity of this formulation, we can determine the number of clusters and the distribution of nodes into these clusters. Then for each cluster, a low complexity linear programming optimization problem is solved to minimize energy consumption of the nodes of the cluster. To define the number of clusters, we use a rule of thumb that sets the number of clusters for n nodes is $\sqrt{n/2}$ [47]. Distributing the nodes among clusters using the rule of thumb and solving the linear programming optimization problem for each cluster return a near-optimal solution compared to original NP-hard integer programming formulation.

Chapter 4

Dynamic Cooperative Content Distribution: Problem Definition

In this chapter, the different aspects of mobility in the network are presented and analysed in the light of the static formulations in Chapter 3. Moreover, the problem addressed in this thesis is defined and presented.

4.1 Mobility Analysis

In a realistic scenario, three types of mobility are differentiated:

- Node mobility in the network: This mobility is mainly caused by the movement of the nodes and the variation of the channel conditions. As mentioned in Section 2.2, the energy consumption of the nodes is related to the data transmission and reception time which is inversely proportional to the transmission and reception rate of the nodes. The latter depends on the distance between the nodes and the channel conditions. Since the nodes are moving and the channel conditions are varying with time, it is required to account for these variations in the formulation of the optimization problem. Starting from the static formulation of MST presented Section 3.2 or from the two-hop IP formulation presented in Section 3.3, this type of mobility affects two parameters: The transmission rate on the LR links $R_{L,k}$ and the transmission rate on the SR links $R_{S,kj}$. Therefore, it affects the weights of the edges of the MST produced by solving the MST optimal formulation and the two-hop tree produced by solving the two-hop IP formulation. This type of dynamicity is well known in the maintenance of MST as weight edge modification.

- Node leaving the network: At any point in time, an existing node may leave the network. This node will affect the distribution of the content especially if this node was downloading the content from the BS on the LR and sharing it with other nodes. This type of mobility is well known in the maintenance of MST as node deletion.

- Node joining the network: At any point in time, a new node interested in the same content may join the network. This type of dynamicity is well known in the maintenance of MST as node insertion.

4.2 Problem Definition

Since the network is dynamic, the optimal solution produced by solving the MST formulation or the two-hop IP formulation at a point in time will no longer be optimal if any modification occurs in the network. To account for the mobility of the network in the optimal solution, two options are suggested: (1) Either to re-solve the optimization problem from scratch using the static formulation or (2) to define a maintenance algorithm that re-optimizes the minimum spanning tree. The first option requires to resolve the static formulation of the optimization problem periodically. Therefore, it imposes high delay and computational overheads. However, regarding the second option, maintenance algorithms such as those defined for the clustering algorithms presented in Section 2.3 can be defined. The concept of re-optimization can be defined as follows: Given an instance of the problem for which we already know an optimal solution and given some modifications on the initial instance, it is possible to compute a new near optimal solution without resolving the static formulation of the optimization problem from scratch. Although this option reduces the delay and the computational overhead, it does not return an optimal solution but a near-optimal one. Consequently, a comparison is needed to be conducted between the optimal solution of the static formulation and the solutions of re-optimization algorithms. For each type of mobility presented in Section 4.1, a re-optimization algorithm that finds pseudo-optimal solution without the need to resolve the optimal formulation from scratch needs to be suggested.

To account for the movement of the nodes and the variations of the channel conditions, a mechanism to track the variations in the network is needed. An optimization framework is needed to re-solve the static optimization problem or to run greedy algorithms to re-optimize the modified network. The optimization framework can either run on a server (centralized) or run among the nodes (distributed). For the centralized option, nodes should periodically update the server with the changes in the network. Therefore, the delay for sending the update is very high and the variation in the network is very rapid. For the distributed option, nodes have limited resources to solve such a complex optimization problem. Moreover, solving this problem in a distributed manner imposes high overhead among the nodes.

The cooperative content distribution in a dynamic network requires the definition of a network formation and a failure detection mechanisms. An application layer architecture that defines how cooperative network is initialized and how the common content is requested and distributed among the node must be proposed. This architecture includes efficient mechanisms to account for different types of mobility in the network.

Chapter 5

Dynamic Cooperative Content Distribution Framework: Proposed Re-optimization Algorithms and Performance Analysis

The problem of cooperative content distribution is formulated in Chapter 3 using MST IP formulation and two-hop tree IP formulation. In these two formulations, the network is considered static. However, in a realistic scenario, the network is dynamic and three aspects of mobility in the network are differentiated: Node insertion (new nodes entering the network), node deletion (nodes leaving the network) and edge weight modification (mobility of the nodes and variations of the channel conditions). The Concept of re-optimization is introduced in Chapter 4 to overcome the overheads of resolving the static formulation from scratch upon any modification in the network. In this chapter, re-optimization algorithms are proposed for each aspect of mobility in the network starting from the MST IP formulation and the two-hop tree IP formulation. Simulations and performance analysis are conducted throughout this chapter to compare the proposed algorithms with the optimal formulations.

5.1 Approach I: Minimum Spanning Tree Re-Optimization

For each type of mobility in the network, a greedy re-optimization algorithm is suggested and tested to maintain pseudo-optimal solutions. Simulations are conducted to compare the energy consumption and the computational complexity of the suggested greedy algorithms with Kruskal's algorithm used to find the optimal solution of MST IP formulation.

5.1.1 Nodes Joining the Network (Node Insertion): Proposed Algorithms

Starting from the MST formulation in Section 3.2, consider an initial instance $I = G(V, E)$ of an undirected graph G(V,E) connecting the nodes and the BS where V is the set of vertices and E is the set of edges connecting the nodes with each other and with the BS. A set X of n nodes $(x_1, x_2, ..., x_n)$ enter the network. The new instance I_x consists of a graph $G_x(V_x, E_x)$ where $V_x = V \cup X$ and $E_x = E \cup E(X)$ where $E(X)$ is the set of edges adjacent to the nodes of X. In [48], a pseudo-optimal node insertion algorithm called $REOPT1+$ is proposed. $REOPT1+$ is defined as follows. Consider T^* is the initial minimum spanning tree before the addition of n nodes. The following algorithm extends T^* to a tree T_{new} spanning the whole V_x:

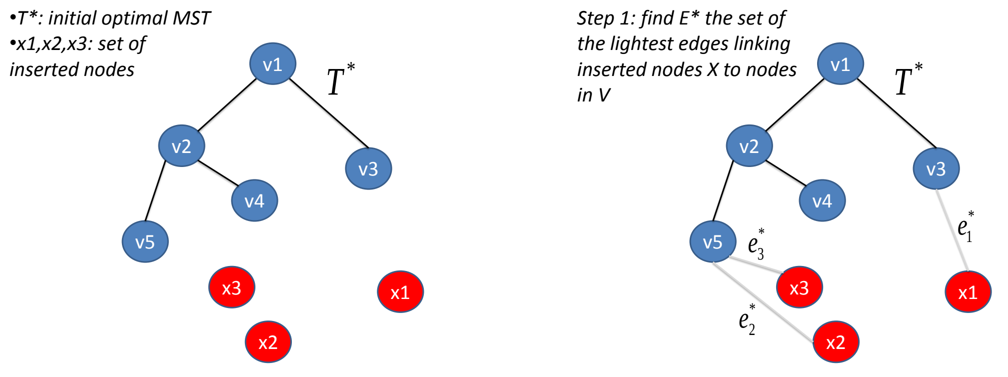
1. for each x_i in X, let $e_i^* = \{x_i, v_i^*\}$ be the lightest edge linking x_i to a node of V; set $E^* = \{e_1^*, ..., e_n^*\}$, $V^* = \{v_1^*, ..., v_n^*\}$
2. build an artificial node v' as the contraction of all nodes in V^*, so that, $\forall i$, $e_i' = (v', x_i)$, and $w(e_i') = w(e_i^*)$; set $E' = \{e_1', ..., e_n'\}$.
3. run Kruskal's algorithm on the graph $H = (X \cup v', (X \times X) \cup E')$; let T_1' be the resulting tree.
4. replace each edge e_i' in T_1' with the corresponding edge e_i^* and denote by T_1^* the resulting set.
5. output $T_{new} = T^* \cup T_1^*$

Figure 5.1 describes the different steps of $REOPT1+$ algorithm. In the presented example, three new nodes x_1, x_2, x_3 are added to a minimum spanning tree T^* consisting of the vertices $v_1, .., v_5$.

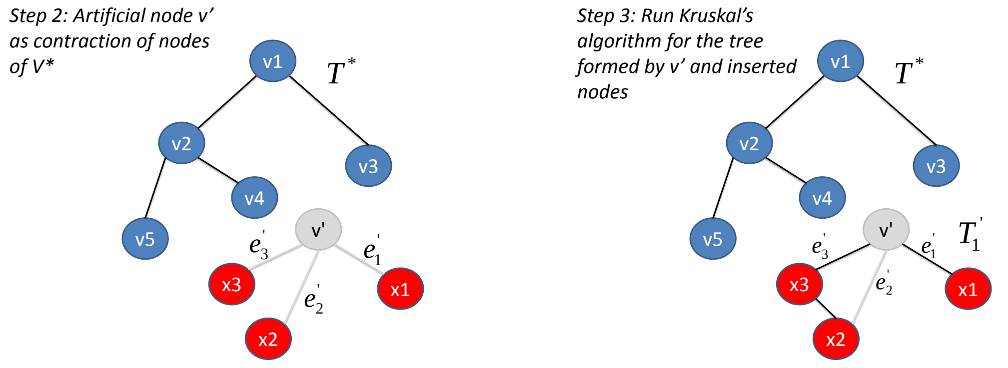
It is easy to see that due to steps 1 and 3 the running time of this algorithm is $O(Kn + n \log n)$ where $K = |V|$ is the number of vertices in T^*. It is better than the complexity of the Kruskal's algorithm run on the entire graph G_x, $O((n + K)^2 \log(n + K))$.

In this thesis, two new node insertion algorithms $MST-INS1$ and $MST-INS2$ are proposed. The $MST-INS1$ algorithm is defined as follows: Consider T^* is the initial minimum spanning tree before the addition of a set X of n nodes. The following algorithm extends T^* to a tree T_{new} spanning the whole V_x:

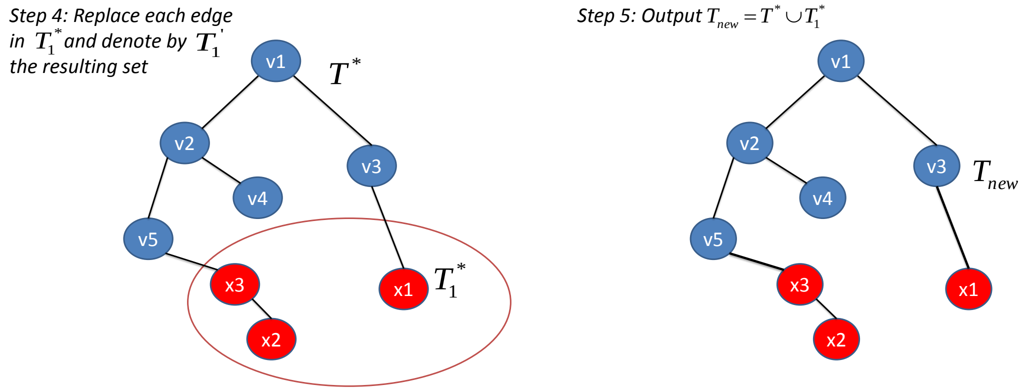
1. let $T' = T^*$ and $V' = V$



(a) Initial distribution of nodes (b) Step 1



(c) Step 2 (d) Step 3



(e) Step 4 (f) Step 5

Figure 5.1: REOPT1+ algorithm.

2. for each x_i in X
    (a) Let $e_i^* = \{x_i, v_i^{'}\}$ be the lightest edge linking x_i to a node of $V^{'}$ ($V^{'}$ includes the vertices of V in addition to the inserted vertices added consecutively throughout the execution of the algorithm)
    (b) Add e_i^* to the edges of $T^{'}$ and x_i to vertices of $T^{'}$
    (c) Repeat Steps 2(a) and 2(b) consecutively for each the inserted nodes x_i
3. output $T_{new} = T^{'}$

Figure 5.2 describes the different steps of $MST - INS1$ algorithm. In the presented example, three new nodes x_1, x_2, x_3 are added to a minimum spanning tree T^* consisting of the vertices $v_1, .., v_5$.
The worst case complexity of finding the minimum value of a list of N entries is $O(N)$. Therefore the worst case complexity of $MST - INS1$ algorithm for the insertion of n nodes of set X to a graph G with K vertices is:

$$O(K) + O(K+1) + O(K+2) + ... + O(K+n-1) \tag{5.1}$$

$$\approx O(K) + O\left(\frac{n(n-1)}{2}\right) \approx O(K+n^2) \tag{5.2}$$

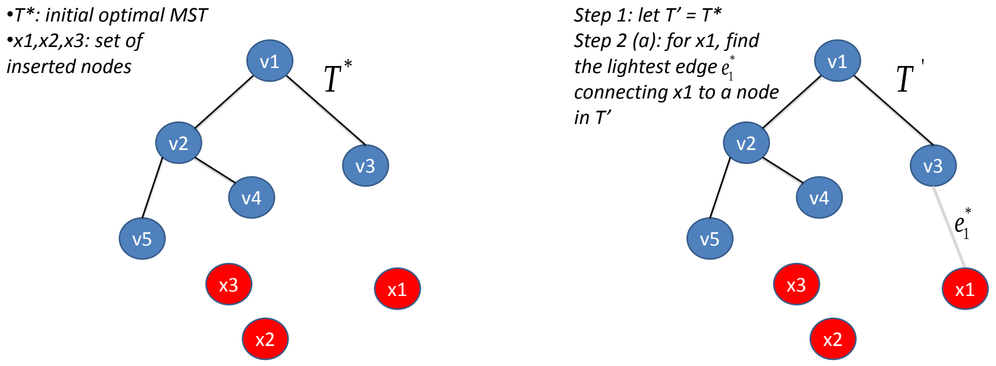
The $MST - INS2$ algorithm is defined as follows:

1. build the sub-graph $H(V \cup X, T * \cup_{i=1}^{n} E(x_i)$ of G_x. H contains , in addition to the edges that connect nodes in X to the nodes in V, the edges of G that belong to T^*.
2. run the Kruskal's Algorithm on the sub-graph H

$MST - INS2$ simply removes all these edges from the set of candidate edges, and runs Kruskals Algorithm, on the surviving graph H that is smaller than G_x. Henceforth it returns an optimal MST for the graph G_x. The complexity of $MST - INS2$ is $O((n^2 + nK)\log(n+K))$, that is better than the complexity of Kruskal's algorithm run on the entire graph G_x, $O((n+K)^2 \log(n+K))$. However, for large number of inserted nodes, both Kruslal and $MST - INS2$ algorithms converge to have similar computational complexity.

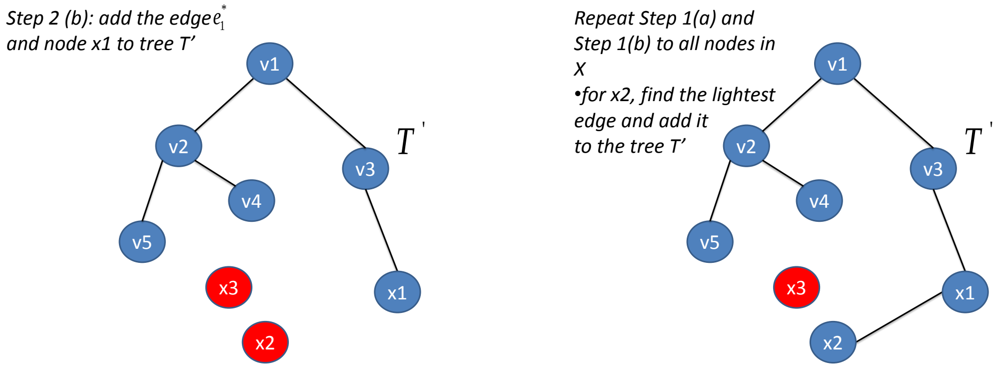
5.1.2 Nodes Joining the Network (Node Insertion): Performance Results and Analysis

To extract performance results for the node insertion re-optimization algorithms of the MST IP formulation, a simulation model is proposed. We deploy a number of MTs within a $200m \times 200m$ square area with the BS located at the left bottom


•T^*: initial optimal MST
•x1,x2,x3: set of inserted nodes

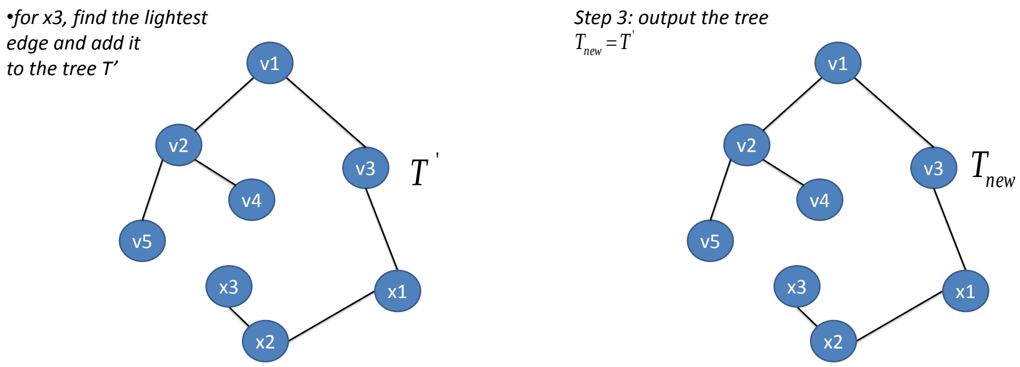
Step 1: let T′ = T*
Step 2 (a): for x1, find the lightest edge e_1^* connecting x1 to a node in T′

(a) Initial distribution of nodes
(b) Step 1 and Step 2(a) for node x1


Step 2 (b): add the edge e_1^* and node x1 to tree T′

Repeat Step 1(a) and Step 1(b) to all nodes in X
•for x2, find the lightest edge and add it to the tree T′

(c) Step 2(b) for node x1
(d) Repeat Steps 2(a) and 2(b) for node x2


•for x3, find the lightest edge and add it to the tree T′

Step 3: output the tree $T_{new} = T'$

(e) Repeat Steps 2(a) and 2(b) for node x3
(f) Step 3

Figure 5.2: MST-INS1 algorithm.

corner of the area as shown in Figure 5.3. Using this simulator, we will capture the formulations of the proposed algorithms with the general energy consumption expressions that are a function of the various transmission rates. Since, using this simulator, the MTs are deployed randomly in the defined area, the distances between each MT and the BS and among the MTs are random. Therefore, the bit rates of the links, calculated based on the assumptions of the system model presented in Section 3.1, depend on the distances of the links.

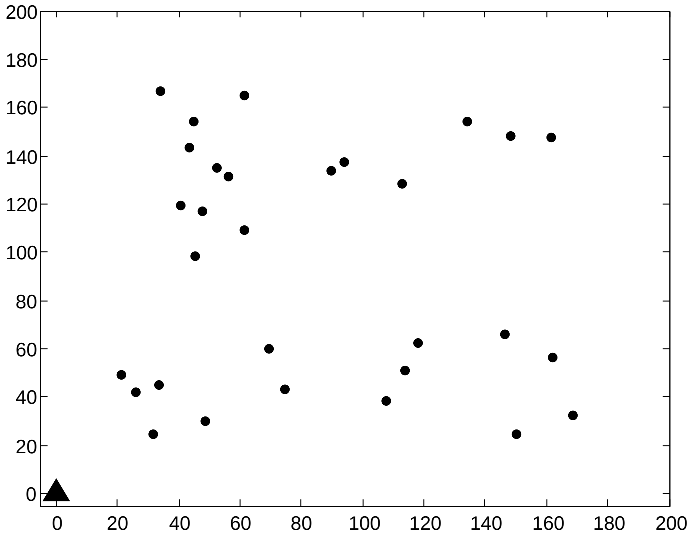


Figure 5.3: Network snapshot for random distribution of 30 nodes using the proposed simulator.

However, there are limited studies, in the literature, that associate for a typical range of transmission rates, the corresponding energy consumed during transmission and reception for various LR and SR technologies. Due to the lack of data sheets that relate the transmission and reception energy, for various LR and SR technologies, at different transmission rates, we will follow the model of the work in [23] [49], which was verified using experimental measurements. According to [23] [49], the energy consumed per unit time can be considered almost constant for various transmission rates.

We consider that the characteristics of the MTs are analogous to the HP iPAQ Pocket PC h6300 Series [50]; thus, we set the transmitting power $P_{Tx} = 1.015\ J/s$ and the receiving power $P_{Rx,S} = 0.66\ J/s$ of a MT using SR links. Without loss of generality, we assume that the energy consumed per second during reception is the same on the LR and SR links; thus, the receiving power of a MT using a LR link $P_{Rx,L} = P_{Rx,S} = 0.66\ J/s$. The thermal noise σ^2 is considered to be 10.6 mW. The channel parameters are defined as follows:

- κ = -31.54 dB, a unitless constant which depends on on various link parameters.

Parameter	Value
P_{Tx}	1.015 J/s
$P_{Rx,S}$	0.66 J/s
$P_{Rx,L}$	0.66 J/s
σ^2	10.6 mW
κ	-31.54 dB
υ	3.71
$d0$	10 m
$\sigma^2_{h_{dB}}$	13.29 dB
ST	4 Mb

Table 5.1: List of parameters used in the proposed simulator.

- The path loss exponent $\upsilon = 3.71$.
- $d0 = 10\ m$, a reference distance.
- $\sigma^2_{h_{dB}} = 13.29\ dB$, a variance for the Gaussian random variable that represents shadowing.

The content size ST is set to 4 Mb. The set of possible values of bit rates R_s is chosen from 1 Mb/s with increments of 1 Mb/s until a maximum of 12 Mb/s. The parameters used in the proposed simulator are summarized in Table 5.1.

The three proposed re-optimization algorithms ($REOPT1+$, $MST-INS1$, and $MST-INS2$) are implemented and tested using this simulator. Moreover this simulator is used to test the performance of all proposed re-optimization algorithms for both MST and two-hop tree formulation.

Figure 5.4 shows the results of node insertion algorithms for a snapshot network scenario with $K = 7$ nodes that were originally deployed in the network (Figure 5.4(a)). Figure 5.4(b) shows the MST generating by running Kruskal's algorithm on the initial distribution of MTs. Then a set of $n = 7$ nodes are inserted in the network. In Figure 5.4(c), Kruskal's algorithm is re-solved from scratch to find the optimal MST spanning all the nodes in the vicinity of the BS. The trees generated by the node insertion algorithms $REOPT1+$, $MST-INS1$, and $MST-INS2$ are shown in Figures 5.4(d),5.4(e), and 5.4(f) respectively. The re-optimization algorithms benefit from the original MST spanning in Figure 5.4(b) to add the new inserted nodes. The black dots represent the nodes that were existent in the initial distribution of nodes and the red dots represent the nodes inserted in the network. We denote by EC the energy consumption of all the nodes in a cooperative content distribution scenario and EN the energy consumption of all the nodes a non-cooperative scenario where each node downloads the content individually on the LR. The optimal solution of the Kruskal's algorithm in Figure 5.4(c) has the smallest energy consumption. Moreover, as expected, $MST-INS2$

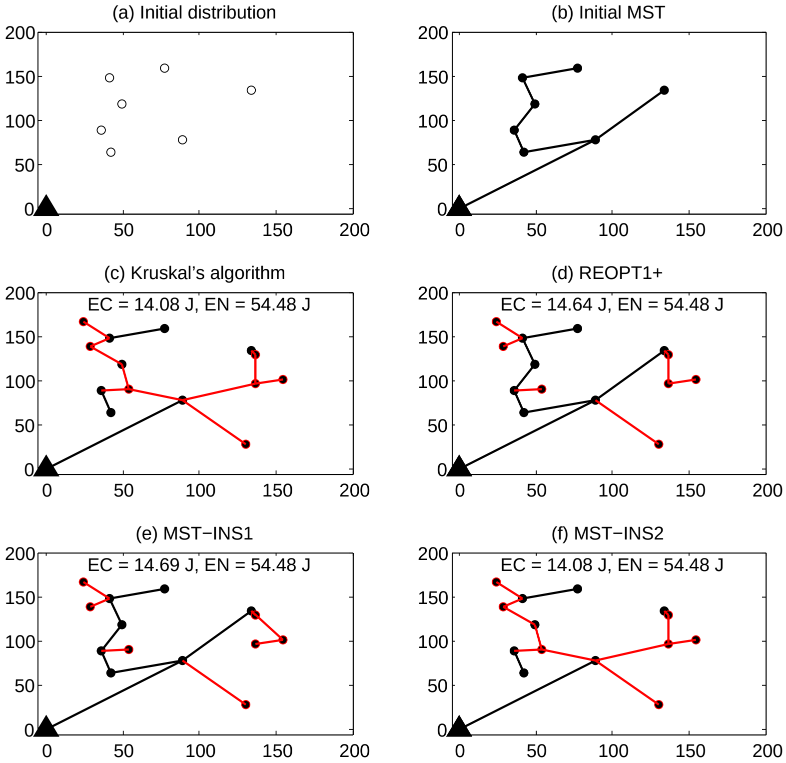


Figure 5.4: Snapshot of MST node insertion algorithms.

algorithm returns an optimal solution in Figure 5.4(f) with same distribution of nodes and EC as Kruskal's algorithm.

To simulate the three alternative algorithms and compare the results with the Kruskal's algorithm, we start with an initial distribution of 15 nodes. Then a set of n nodes is randomly deployed in the network where the value of n varies between 1 and 30. For each inserted node, the distances of this node with the BS and with other MTs are used to estimate the bit rates following the assumptions presented in Section 3.1.

The new graph produced by the insertion of each set of nodes is solved by each of the four algorithms (Kruskal, $REOPT1+$, $MST-INS1$ and $MST-INS2$ algorithms). For each value of n, 10000 iterations of the simulation are run. For each iteration, n nodes are randomly deployed in the network and the four tested algorithms are solved. For each algorithm, the average energy consumption and the average CPU runtime are computed and presented in figures 5.5 and 5.6 respectively.

Figure 5.5 shows the energy efficiency ratio of Kruskal's algorithm, $REOPT1+$ algorithm, $MST-INS1$ algorithm, and $MST-INS2$ algorithm versus the number of inserted nodes. The energy efficiency ratio is the ratio of the energy consumption of the nodes in a cooperative scenario to the energy consumption in a non-cooperative scenario. The energy consumption of the solution of $MST-$

$INS2$ algorithm proposed in this thesis is smaller than that of the pseudo-optimal algorithm $REOPT1+$ proposed in [48]. The energy consumption of $MST-INS1$ is slightly higher than that of $REOPT1+$. Moreover, the cooperative energy consumption of all the proposed algorithms is higher than that of the Kruskal's algorithm expect the $MST-INS2$ algorithm since it returns an optimal solution.

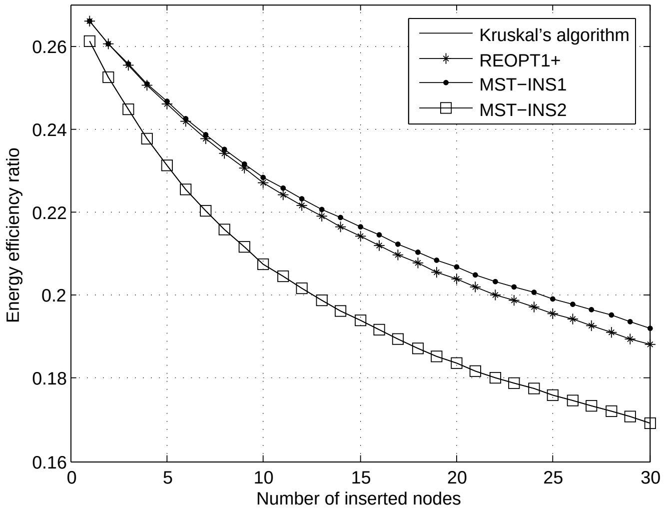


Figure 5.5: Energy efficiency ratio vs. number of inserted nodes for MST algorithms.

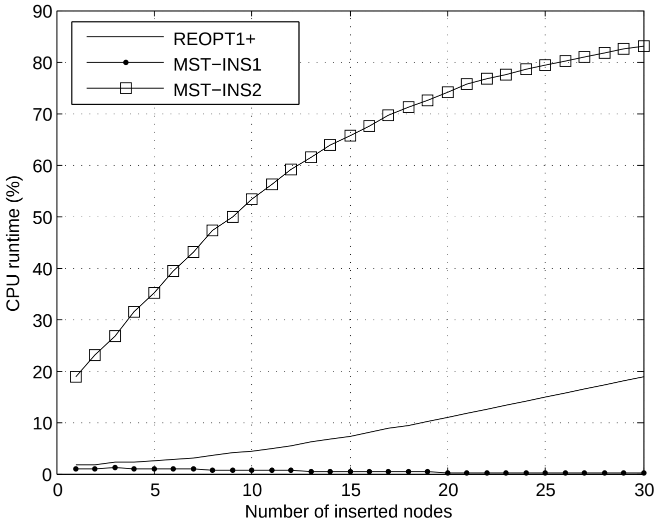


Figure 5.6: CPU runtime percentage vs. number of inserted nodes for MST algorithms.

Figure 5.6 shows the CPU runtime percentage of $REOPT1+$ algorithm, $MST-INS1$ algorithm, and $MST-INS2$ algorithm versus the number of nodes inserted. The CPU runtime percentage is the percentage of the CPU runtime of an algorithm to the CPU runtime of the Kruskal's algorithm. All the algorithms have better CPU runtime than the Kruskal's algorithm. The pseudo-optimal algorithm $MST-INS1$ has smallest CPU runtime than the pseudo-optimal algorithm $REOPT1+$. However the algorithm $MST-INS2$ has the worst CPU runtime percentage since it is an optimal algorithm based on the Kruskal's algorithm. As the number of inserted nodes increases, the CPU runtime of $MST-INS2$ converges to that of Kruskal's algorithm.

Simulations show that the proposed algorithm $MST-INS2$ has the smallest energy consumption among the suggested algorithms. However, it has the highest computational complexity among these algorithms. Its computational complexity is very close to that of the Kruskal's algorithm especially for large number of inserted nodes. $MST-INS1$ has the smallest computational complexity among the suggested algorithms.

Table 5.2 compares the MST node insertion algorithms. $Kruskal$ and $MST-INS2$ algorithms return optimal solution but both algorithms have relatively high computational complexity. $REOPT1+$ and $MST-INS1$ has slightly higher energy consumption that $Kruskal$ and $MST-INS2$ algorithms. However, they have better computational complexity. Node insertion algorithms require knowledge of all of the information of all the nodes. Therefore, the implementation of these algorithms requires high computational and communications capabilities. Implementing the node addition algorithms in a distributed way imposes additional delay and computational overheads on the nodes. Thus, node insertion algorithm must be implemented within the service provider in a centralized manner.

Algorithm	Performance	Complexity	Implementation
$Kruskal$	Optimal	$O((n+K)^2\log(n+K))$	Centralized
$REOPT1+$	Pseudo-optimal	$O(Kn+n\log n)$	Centralized
$MST-INS1$	Pseudo-optimal	$O(K+n^2)$	Centralized
$MST-INS2$	Optimal	$O((n^2+nK)\log(n+K))$	Centralized

Table 5.2: Comparison of MST node insertion algorithms.

5.1.3 Nodes Leaving the Network (Node Deletion): Proposed Algorithms

As described in Section 4.1, a node can leave the network at any point in time. To account for this aspect of network mobility without resolving the MST problem from scratch, a pseudo-optimal algorithm $MST-DEL$ is proposed in this thesis. Starting from the MST formulation in Section 3.2, consider an initial instance $I = G(V, E)$ of an undirected graph G(V,E) connecting the nodes and the BS where V is the set of vertices and E is the set of edges connecting the nodes with each other and with the BS. The set X is a subset of V of n nodes $(x_1, x_2, ..., x_n)$ that leave the network. Let T^* be the initial minimum spanning tree before the deletion of n nodes. V can be split into the following distinct subsets:

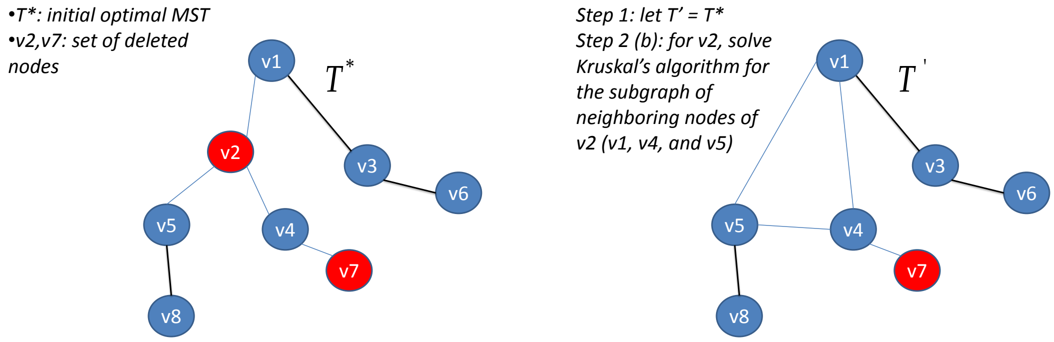
- V^0, the subset of nodes of T^* that are leaf nodes in T^*.
- V^1, the subset of edges of T^* that have at least one children node in the MST T^*

The following algorithm computes a solution of the node deletion problem:

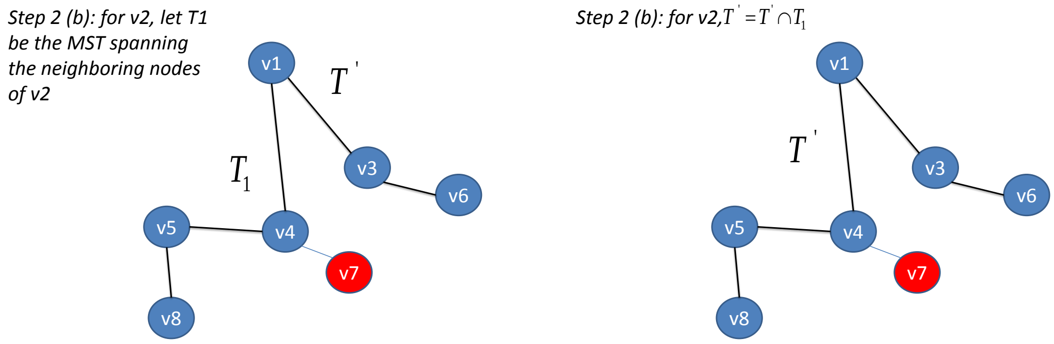
```
1. let $T^{'} = T^*$
2. for each $x_i$ in $X$
    (a) if $x_i \subset V^0$
        i. let $e_i^*$ be the edge linking $x_i$ to a node of $V^1$
        ii. remove node $x_i$ and $e_i^*$ from $T^{'}$
    (b) if $x_i \subset V^1$
        i. if $x_i$ has two neighbouring nodes in $T^{'}$, connect these two nodes
           and add the corresponding edge to $T^{'}$.
        ii. if $x_i$ has more than two neighbouring nodes in $T^{'}$, solve the Kruskal's
            algorithm for the subgraph connecting the neighbouring nodes of
            $x_i$ and let $T_1$ be the solution. Set $T^{'} = T^{'} \cap T_1$.
3. output $T_{new} = T^{'}$
```

Figure 5.7 describes the different steps of $MST - DEL$ algorithm. In the presented example, two nodes v_2, v_7 are deleted from the minimum spanning tree T^*.

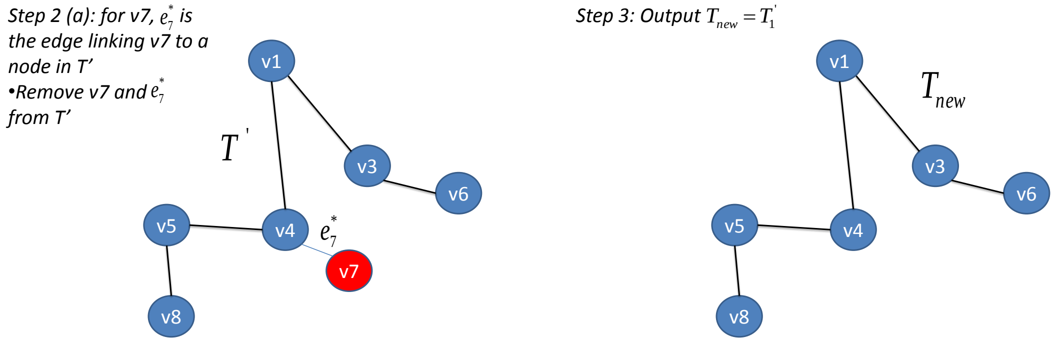
Kruskal's algorithm requires the aggregation of the information of all the nodes in the network to formulate and solve the optimization problem. Therefore, it is better to be implemented in a centralized manner. However, $MST-DEL$ can be implemented in a distributed way: When a node leaves the network, $MST-DEL$ finds an local optimal solution for the neighbouring nodes of the deleted node. Thus, $MST - DEL$ can be implemented on the level of the node and run by one of the neighbouring nodes.



(a) Initial distribution of nodes (b) Step 1 and Step 2(b) for node v2



(c) Step 2(b) for node v2 (d) Step 2(b) for node v2



(e) Step 2(b) for node v7 (f) Step 3

Figure 5.7: MST-DEL algorithm.

5.1.4 Nodes Leaving the Network (Node Deletion): Performance Results and Analysis

To simulate the proposed algorithm $MST - DEL$, we use the simulator defined in Section 5.1.2. Figure 5.8 compares the results of Kruskal's algorithm and $MST - DEL$ node deletion algorithm for a snapshot network scenario with $K = 15$ nodes forming a MST generated by Kruskal's algorithm.

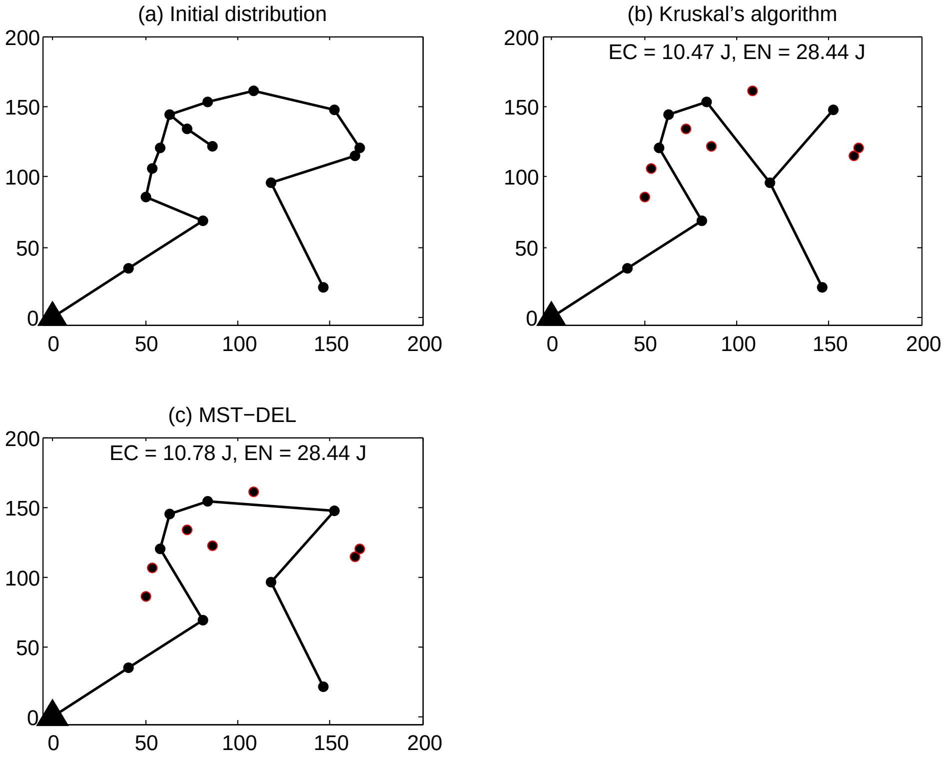


Figure 5.8: Snapshot of MST node deletion algorithms.

Then a set of $n = 7$ nodes are deleted from the network. In Figure 5.8(b), Kruskal's algorithm is re-solved from scratch to find to the optimal MST spanning all the nodes remaining in the vicinity of the BS after deletion of other nodes. However, the node deletion algorithm $MST - DEL$ benefits from the original MST to remove the deleted nodes. The black dots represent the nodes that were existent in the initial distribution of nodes and the red dots represent the nodes removed from the network. Since the algorithm $MST - DEL$ is based on the initial MST solution, the MST generated by the $MST - DEL$ is similar to the initial distribution. However, since using Kruskal's algorithm the MST is generated from scratch and not based on the initial solution, the difference is clear between the solution based on the Kruskal's algorithm and the initial distribution. We denote by EC the energy consumption of all the nodes in a cooperative content distribution scenario and EN the energy consumption of all the nodes a non-cooperative scenario where each node downloads the content individually via a LR technology. $MST-DEL$ algorithm has the slightly higher energy consumption than Kruskal's algorithm. However, $MST-DEL$ maintains the gain of cooperation.

To simulate the proposed algorithm $MST-DEL$ and compare the results with the Kruskal's algorithm, we start with an initial distribution of 31 nodes. Then a set of n nodes is deleted from the network where n varies between 1 and 30. The new graph produced by the deletion of the sets of nodes is solved by Kruskal's algorithm and $MST-DEL$ algorithm. We repeat this scenario for 10000 iterations and for each iteration we record the energy consumption and CPU runtime for each of the implemented algorithms. For each algorithm, the average energy consumption and average CPU runtime are computed and presented in Figure 5.8 and 5.10 respectively.

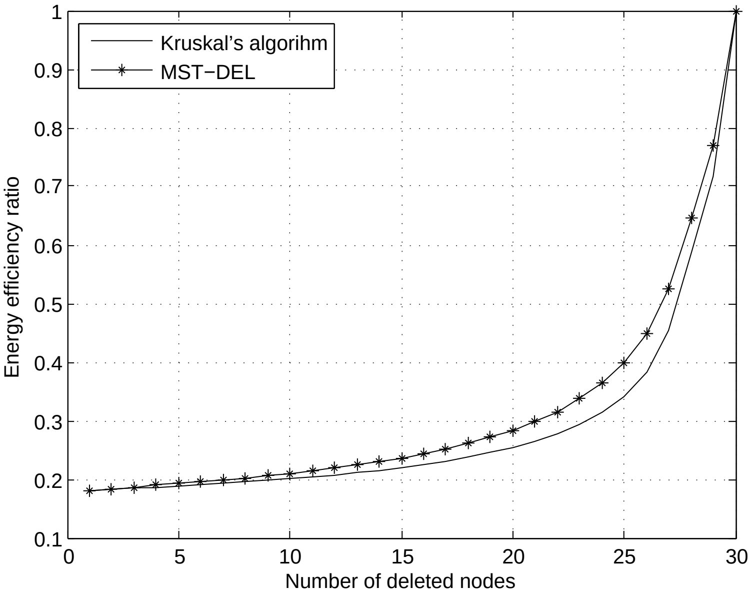


Figure 5.9: Energy efficiency ratio vs. number of deleted nodes for MST algorithms.

Figure 5.9 shows the energy efficiency ratio versus the number of deleted nodes. The energy consumption of the MST solution generated by the $MST-DEL$ algorithm is relatively close to the energy consumption of the MST solution generated by kruskal's algorithm. Although the energy consumption of $MST-DEL$ algorithm is barely higher than Kruskal's algorithm, it still maintains the gain of the cooperative content distribution among the nodes.

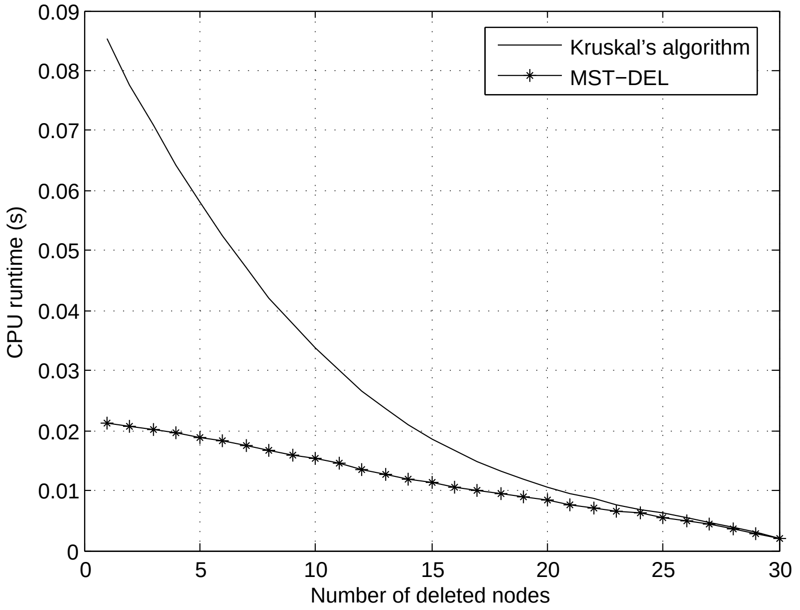


Figure 5.10: CPU runtime (s) vs. number of deleted nodes for MST algorithms.

Figure 5.10 shows the CPU runtime (in seconds) of the $MST - DEL$ algorithm versus the number of deleted nodes. The CPU runtime of $MST - DEL$ algorithm is smaller than that of the Kruskal's algorithm. Therefore, $MST - DEL$ algorithm has better computational complexity that Kruskal's algorithm especially if the number of the deleted nodes is relatively small. When the number of deleted nodes increases (number of nodes remaining in the network decreases), both Kruskal's algorithm and $MST - DEL$ algorithm converges to have close computational complexity.

5.1.5 Nodes Moving in the Network (Edge Weight Modification): Proposed Algorithms

In a dynamic scenario, nodes are moving in the network. Moreover, the channel conditions are varying over time. This type of mobility in the network cause the modification of the weights of the edges of the solution of the MST. The weight of the edge is defined in the static formulation of MST in Section 3.2. In this work, an edge weight modification pseudo-optimal algorithm called $MST - MOV1$ is proposed.

Consider an initial instance $I = G(V, E)$ of IP formulation of the content distribution problem. A set X of n nodes $(x_1, x_2, ..., x_n)$ is the subset of V where their edges are modified. The new instance I_x consists of a graph $G_x(V, E_x)$ where $E_x = E \cap E(X)$ where $E(X)$ is the set of modified edges. We propose the following re-optimization algorithm:

1. let $T^{'} = T^{*}$

2. for each x_i in X
    (a) if x_i is a leaf node
        - let $e_i^{'} = \{x_i, v_i^*\}$ be the lightest edge linking x_i to a node of G
        - replace e_i^* by $e_i^{'}$ in $T^{'}$
    (b) if x_i is not a leaf node
        - re-solve the Kruskal's algorithm for the nodes that are connected to x_i and let T_{x_i} be the solution
        - $T^{'} = T^{'} \cup T_{x_i}$
3. output $T_{new} = T^{'}$

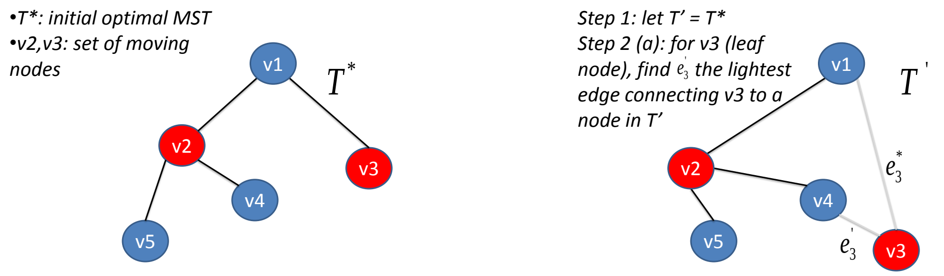
Figure 5.11 describes the different steps of $MST - MOV$ algorithm. In the presented example, two nodes v_2, v_3 moves in the network. Therefore, the weights of the edges connecting these two nodes with other nodes in the network vary accordingly.

5.1.6 Nodes Moving in the Network (Edge Weight Modification): Performance Results and Analysis

The $MST-MOV$ algorithm is tested using the simulator defined in Section 5.1.2. To simulate this algorithm and compare the results with the Kruskal's algorithm, we start with an initial distribution of 30 nodes. Then a number of nodes are consecutively moving in the network. The number of nodes vary between 1 and 30. When a node moves in the network, the distances with BS and with other nodes vary. Therefore, the bit rates to communicate with the BS and with other nodes vary according to the definition of bit rate in Section 3.1. The new graph produced by the movement of the nodes is solved by Kruskal's algorithm and $MST - MOV$ algorithm. This scenario is repeated for 10000 iterations. For each iteration, the energy consumption and the CPU runtime are recorded for Kruskal's algorithm and for $MST - MOV$ algorithm. For each solution, the average energy consumption and average CPU runtime are computed and presented in Figure 5.12 and 5.13 respectively.

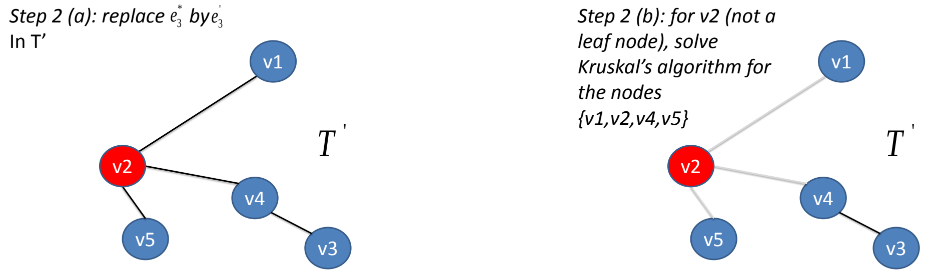
Figure 5.12 shows the energy efficiency ratio versus the number of moving nodes. The energy efficiency ratio is the ratio of the energy consumption of the nodes in cooperative scenario to the energy consumption in the non-cooperative scenario. The energy consumption of the MST solution generated by the $MST - MOV$ algorithm proposed in this thesis is relatively close to the energy consumption of the MST solution generated by kruskal's algorithm. Although the energy consumption of $MST - MOV$ algorithm is barely higher than Kruskal's algorithm, it still maintains the gain of the cooperative content distribution among nodes.

Figure 5.13 shows the CPU runtime (in seconds) of the $MST - MOV$ algorithm versus the number of deleted nodes. The CPU runtime of $MST-MOV$ algorithm



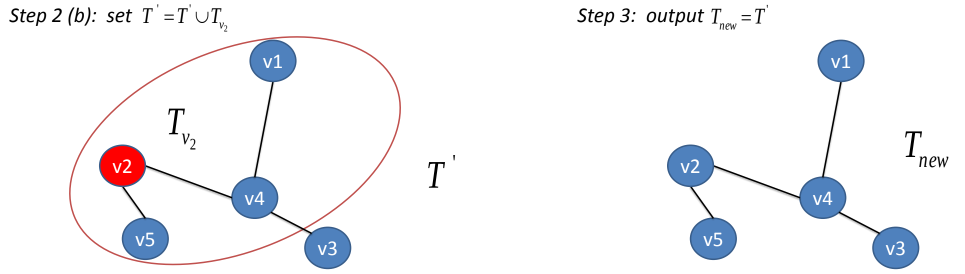
(a) Initial distribution of nodes

(b) Step 1 and Step 2(a) for node v3



(c) Step 2(a) for node v3

(d) Step 2(b) for node v2



(e) Step 2(b) for node v2

(f) Step 3

Figure 5.11: MST-MOV algorithm.

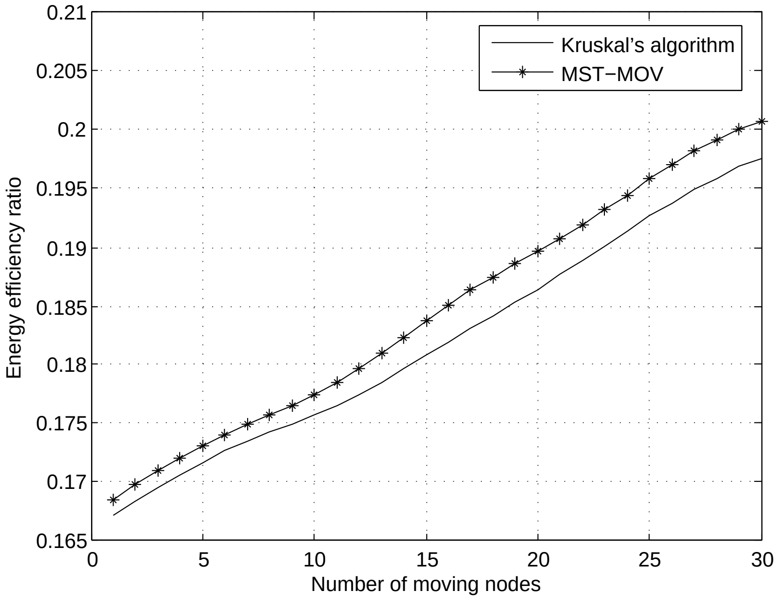


Figure 5.12: Energy efficient ratio vs. number of moving nodes for MST algorithms.

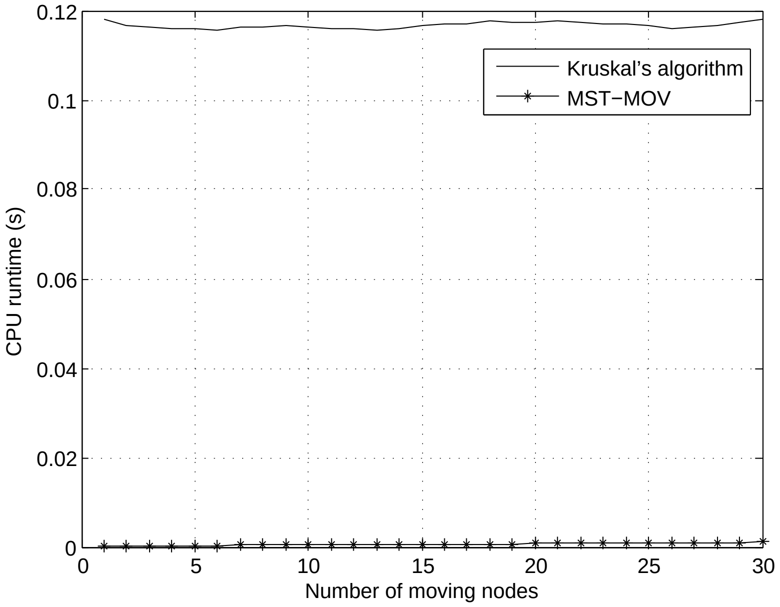


Figure 5.13: CPU runtime (s) vs. number of moving nodes for MST algorithms.

is very small compared to that of the Kruskal's algorithm (The CPU runtime of $MST - MOV$ algorithm is less that 1 percent of that of the Kruskal's algorithm). Therefore, $MST - MOV$ algorithm has better computational complexity than the Kruskal's algorithm.

5.2 Approach II: Two-hop Tree Re-optimization

In this section, three pseudo-optimal re-optimization algorithms that account for the three different types of mobility are proposed for the static IP formulation [13] presented in Section 3.3. The solution of the IP formulation is a two-hop network where the nodes are either downloading the content from the BS directly or downloading it from one the nodes that communicate directly with the BS.

5.2.1 Nodes Joining the Network (Node Insertion): Proposed Algorithms

To account for node insertion in the network, a two-hop pseudo-optimal algorithm $2HOP-INS$ is proposed.

Consider an initial instance $I = G(V, E)$ of IP formulation of the content distribution problem. A set X of n nodes $(x_1, x_2, ..., x_n)$ enter the network. The new instance I_x consists of a graph $G_x(V_x, E_x)$ where $V_x = V \cup X$ and $E_x = E \cup E(X)$ where $E(X)$ is the set of edges adjacent to the nodes of X. Let V_{LR} be the subset of nodes that communicate directly with the BS.

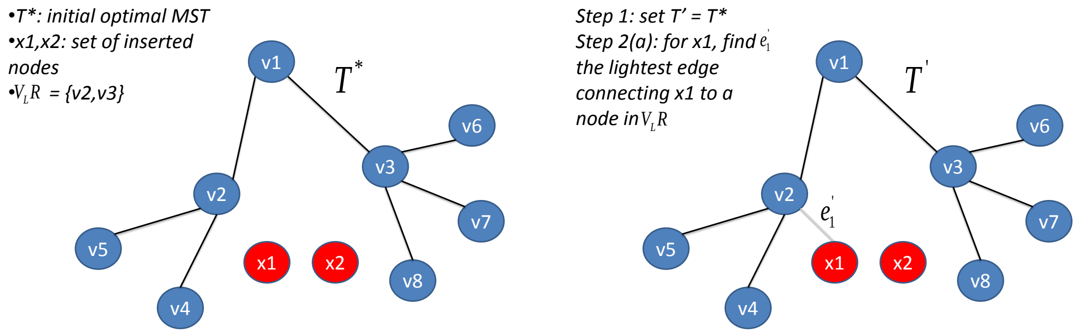
```
1. let $T^{'} = T^*$
2. for each $x_i$ in $X$
    (a) let $e_i^{'} = \{x_i, v_i^{'}\}$ be the lightest edge linking $x_i$ to a node of $V_{LR}$
    (b) add node $x_i$ and edge $e_i^{'}$ to the $T^{'}$ such that $T^{'} = T^{'} \cup (x_i, e_i^{'})$
3. output $T_{new} = T^{'}$
```

Figure 5.14 describes the different steps of $2HOP - INS$ algorithm. In the presented example, two new nodes x_1, x_2 are added to a two-hop tree T^* consisting of the vertices $v_1, .., v_8$.

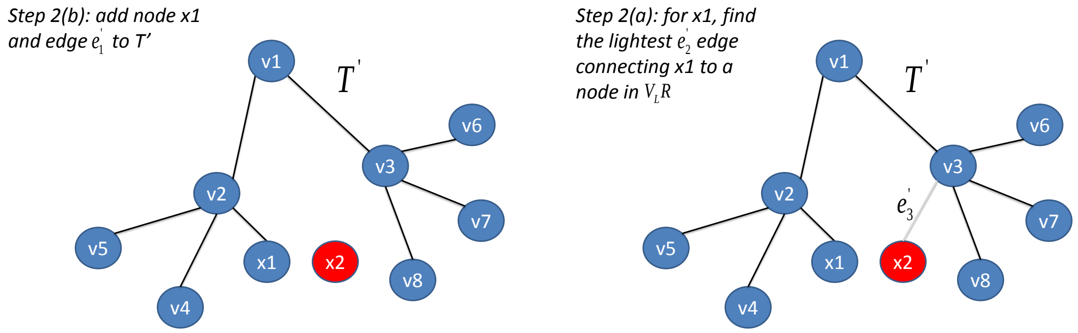
5.2.2 Nodes Joining the Network (Node Insertion): Performance Results and Analysis

To simulate the proposed algorithm $2HOP - INS$ and compare the results with the two-hop IP algorithm presented in Section 3.3, we use the simulator defined in Section 5.1.2. We start with an initial distribution of 15 nodes. Then a set of n nodes is randomly deployed in the network where the value of n varies between 1 and 30. For each inserted node, the distances of this node with the BS and with other MTs are used to estimate the bit rates following the assumptions presented in Section 3.1.

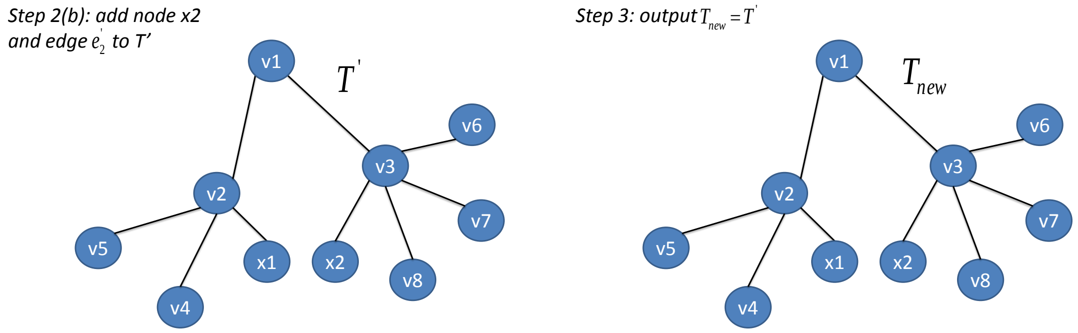
The new graph produced by the insertion of each set of nodes is solved by $2HOP-INS$ and two-hop IP algorithm. For each value of n, 10000 iterations of the



(a) Initial distribution of nodes (b) Step 1 and Step 2(a) for node x1



(c) Step 2(b) for node x1 (d) Step 2(a) for node x2



(e) Step 2(b) for node x2 (f) Step 3

Figure 5.14: 2HOP-INS algorithm.

simulation are run. For each iteration, n nodes are randomly deployed in the network and the two tested algorithms are solved. For each algorithm, the average energy consumption and the average CPU runtime are computed and presented in figures 5.15 and 5.16 respectively.

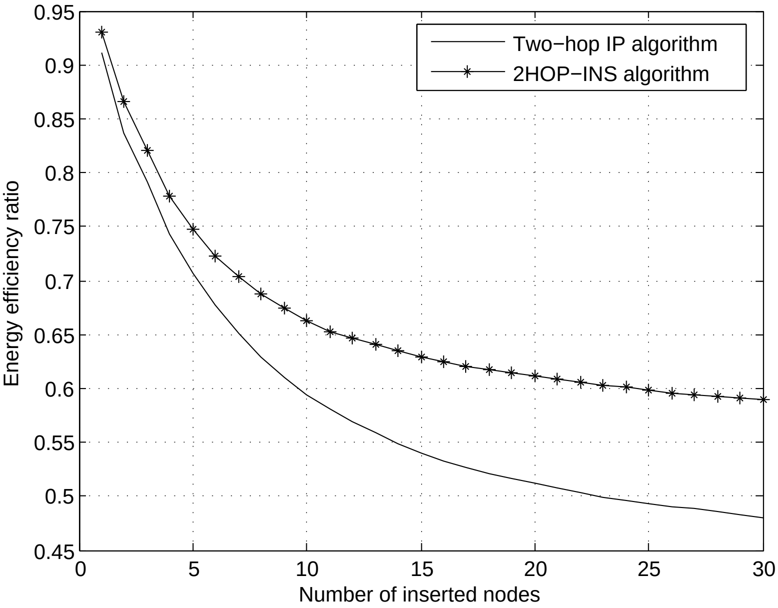


Figure 5.15: Energy efficiency ratio vs. number of inserted nodes for two-hop algorithms.

Figure 5.15 shows the energy efficiency ratio versus the number of deleted nodes. The energy efficiency ratio is the ratio of the energy consumption of the nodes in cooperative scenario to the energy consumption in the non-cooperative scenario. The energy consumption of the two-hop pseudo-optimal solution generated by the $2HOP - INS$ algorithm proposed in this thesis is relatively close to the energy consumption of the solution of the two-hop IP formulation presented in Section 3.3. Although the energy consumption of $2HOP - INS$ algorithm is barely higher than the two-hop IP formulation algorithm, it still maintains the gain of the cooperative content distribution among nodes.

Figure 5.16 shows the CPU runtime (seconds) of the $2HOP - INS$ algorithm versus the number of deleted nodes. The CPU runtime of $2HOP - INS$ algorithm is very small compared to that of the two-hop IP formulation. Therefore, $2HOP - INS$ algorithm has better computational complexity that the two-hop IP formulation.

In Figure 5.17, we compare the performance of MST algorithms and two-hop algorithms. The performance results show that MST algorithms have better energy efficiency that the two-hop algorithms. This is due to the additional constraint imposed in the formulation of the two-hop static algorithm. In the

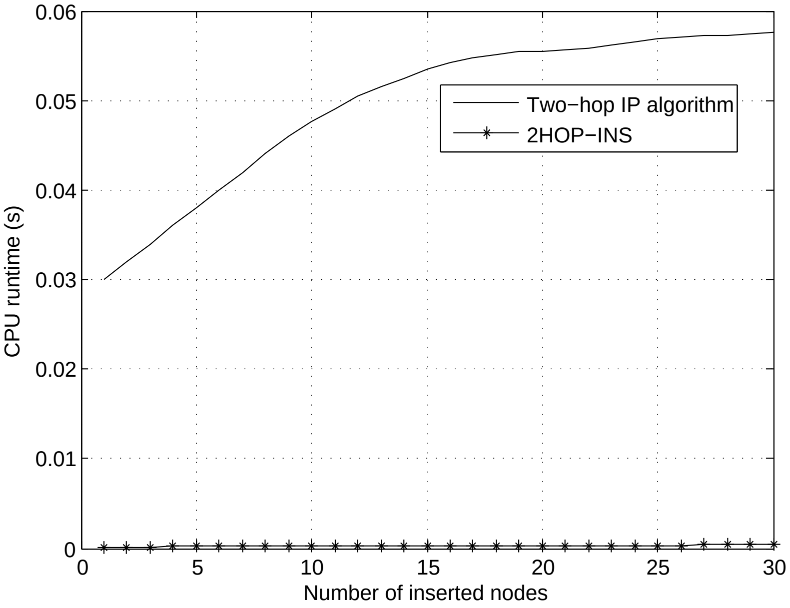


Figure 5.16: CPU runtime (s) vs. number of deleted nodes for two-hop algorithms.

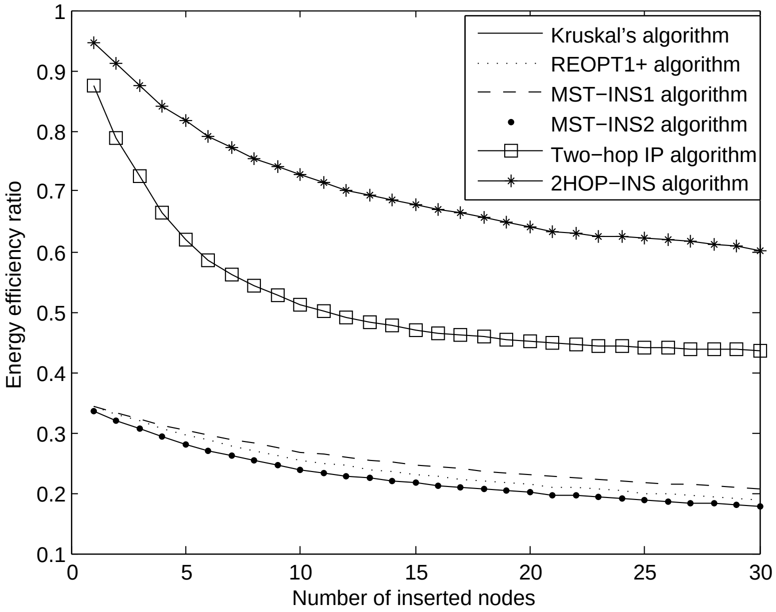


Figure 5.17: Energy efficiency ratio vs. number of inserted nodes for MST and two-hop algorithms.

two-hop static formulation presented in Section 3.3, the number of hops in the resulted tree is limited to two. However, in the MST static formulation presented in Section 3.2, this optimization constraint is relaxed and the resulting tree can have any number of hops. Because of this additional constraint in the two-hop static formulation, the two-hop formulation has higher computational complexity than Kruskal's algorithm.

In a cooperative content distribution scenario, the processing and delay overheads are higher in a MST than a two-hop tree. In practice, the delay between the reception of the content by a leaf node and the reception of the same content by a parent node (that communicates with the BS) in a MST is relatively high compared to that in a two-hop tree. For instance, in a MST, a parent node could start receiving a new content packet while a leaf node did not yet receive the previous packet. This imposes a critical challenge for the designers of the cooperative content distribution architecture and needs additional mechanisms to avoid the high delay.

5.2.3 Nodes Leaving the Network (Node Deletion): Proposed Algorithms

In a mobile network, a node can leave the network at any point in time. To account for this aspect of network mobility without resolving the two-hop IP formulation from scratch, two pseudo-optimal algorithm $2HOP-DEL1$ and $2HOP-DEL2$ are proposed.

Following is the formulation of $2HOP-DEL1$ algorithm. Consider the set X of n nodes $(x_1, x_2, ..., x_n)$ leave the network. Let T^* be the initial minimum spanning tree before the deletion of n nodes. T^* can be split into the following distinct subsets:

1. V_{LR}, the subset of nodes that communicate directly with the BS..
2. V_{SR}, the subset of nodes that communicate a node of V_{LR}.

We proposed the following deletion re-optimization algorithm for the IP formulation of the content distribution problem:

1. let $T^{'} = T^*$
2. for each x_i in X
    (a) if $x_i \subset V_{SR}$
        i. let e_i^* be the edge linking x_i to a node of V_{LR}
        ii. remove node x_i and e_i^* from $T^{'}$
    (b) if $x_i \subset V_{LR}$
        i. let e_i^* be the edge linking x_i to the BS.
        ii. remove node x_i from $T^{'}$
        iii. re-solve the two-hop IP formulation for the nodes in V_{SR} that were connected to x_i and let T_{x_i} be the solution

```
        iv. $T^{'} = T^{'} \cup T_{x_i}$
3. output $T_{new} = T^{'}$
```

$2HOP - DEL2$ algorithm is similar to $2HOP - DEL1$. The only difference between the two algorithms is in Step 2(b)-iii: If a deleted node was a cluster-head of a cluster in the two-hop tree, instead of resolving the two-hop IP formulation for all the nodes in the cluster, find the node that is connected with the lightest edge to BS and set it as cluster-head. Following is the description of the $2HOP-DEL2$ algorithm.

```
1. let $T^{'} = T^{*}$
2. for each $x_i$ in $X$
    (a) if $x_i \subset V_{SR}$
        i. let $e_i^*$ be the edge linking $x_i$ to a node of $V_{LR}$
        ii. remove node $x_i$ and $e_i^*$ from $T^{'}$
    (b) if $x_i \subset V_{LR}$
        i. let $e_i^*$ be the edge linking $x_i$ to the BS.
        ii. let $V_{SR}(i)$ the nodes in $V_{SR}$ that were connected to $x_i$
        iii. remove node $x_i$ from $T^{'}$
        iv. find the node in $V_{SR}(i)$ that is connected with the lightest edge
            to BS and set it as the cluster-head of $V_{SR}(i)$ and let $T_{x_i}$ be the
            solution
        v. $T^{'} = T^{'} \cup T_{x_i}$
3. output $T_{new} = T^{'}$
```

5.2.4 Nodes Leaving the Network (Node Deletion): Performance Results and Analysis

To simulate the proposed algorithms $2HOP - DEL1$ and $2HOP - DEL2$ and compare the results with the two-hop IP formulation presented in Section 3.3, we use the simulator defined in Section 5.1.2. We start with an initial distribution of 31 nodes. Then a set of n nodes is deleted from the network where n varies between 1 and 30. The new graph produced by the deletion of the sets of nodes is solved two-hop IP algorithm, $2HOP-DEL1$, and $2HOP-DEL2$ algorithms. We repeat this scenario for 10000 iterations and for each iteration we record the energy consumption and CPU runtime for each of the implemented algorithms. For each algorithm, the average energy consumption and average CPU runtime are computed and presented in Figure 5.18 and 5.19 respectively.

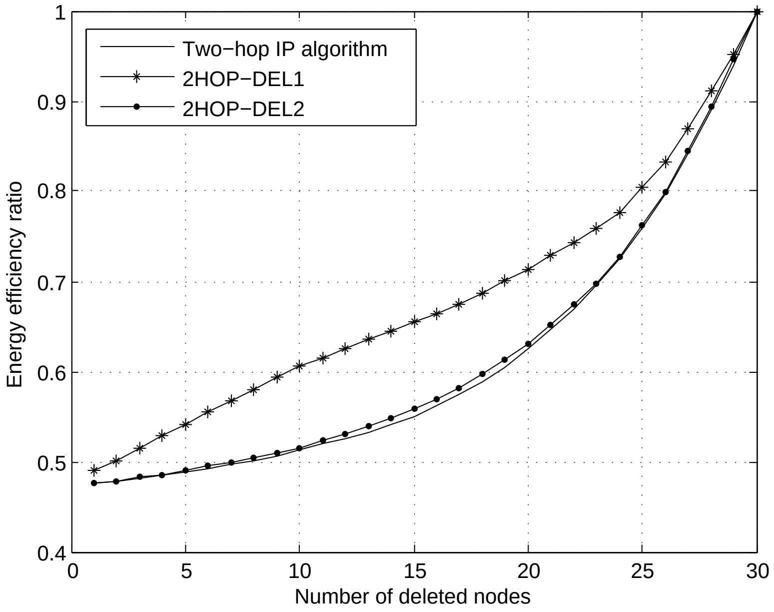


Figure 5.18: Energy efficiency ratio vs. number of deleted nodes for two-hop algorithms.

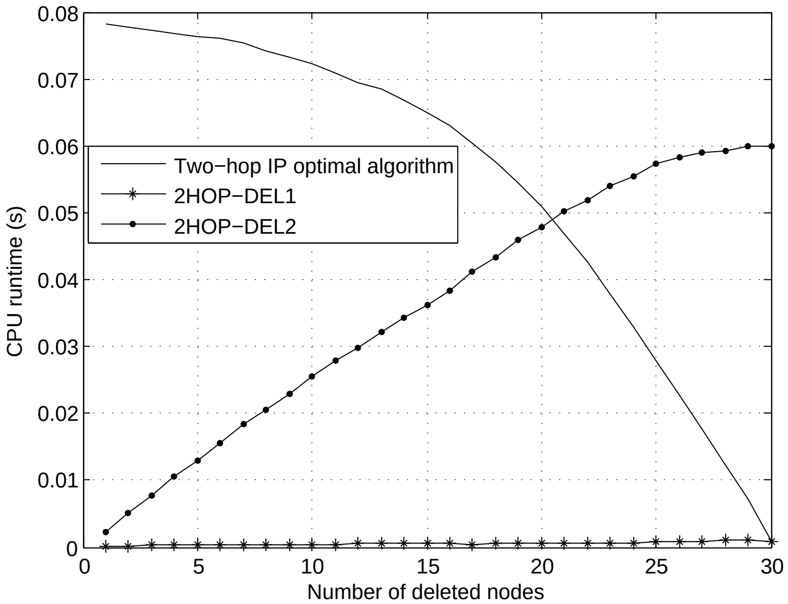


Figure 5.19: CPU runtime (s) vs. number of deleted nodes for two-hop algorithms.

Figure 5.18 shows the energy efficiency ratio versus the number of deleted nodes. The energy efficiency ratio is the ratio of the energy consumption of the nodes in cooperative scenario to the energy consumption in the non-cooperative scenario.

The energy consumption of the two-hop solution generated by the $2HOP-DEL1$ algorithm proposed in this thesis is very close to the energy consumption of the MST solution generated by the two-hop IP formulation. The energy consumption of $2HOP-DEL2$ is higher than of $2HOP-DEL1$ and two-hop IP formulation. However, $2HOP-DEL2$ maintains the gain of the cooperative content distribution among the nodes.

Figure 5.19 shows the CPU runtime (in seconds) of the two-hop IP formulation, $2HOP-DEL1$ algorithm, and $2HOP-DEL2$ algorithm versus the number of deleted nodes. The CPU runtime of $2HOP-DEL2$ algorithm is very small compared to that of two-hop IP formulation and $2HOP-DEL1$. Therefore, $2HOP-DEL2$ has the best computational complexity among the three tested algorithms. $2HOP-DEL1$ has better computational complexity than the two-hop IP solution when the number of the deleted nodes is small.

5.2.5 Nodes Moving in the Network (Edge Weight Modification): Proposed Algorithms

To account for the edge weight modification in a dynamic network without re-solving the two-hop IP formulation, an edge weight modification two-hop pseudo-optimal algorithm called $2HOP-MOV$ is proposed.

Consider an initial instance $I = G(V, E)$ of IP formulation of the content distribution problem. A set X of n nodes $(x_1, x_2, ..., x_n)$ is the subset of V where their edges are modified. The new instance I_x consists of a graph $G_x(V, E_x)$ where $E_x = E \cap E(X)$ where $E(X)$ is the set of modified edges. Let V_{LR} be the subset of nodes that communicate directly with the BS and V_{SR} be the set of nodes that are not directly connected to the BS.

We propose the following re-optimization algorithm:

1. for each x_i in X, if $x_i \subset V_{SR}$
    (a) let $e_i^{'}$ be the edge linking x_i to a node of V_{LR}
    (b) replace e_i^* by $e_i^{'}$ in T^* and let $T_{new} = T^*$
2. if $x_i \subset V_S L$
    (a) re-solve the IP formulation for the nodes in V_{SR} that are connected to x_i and let $T_{x(i)}$ be the solution
    (b) $T_{new} = T^* \cup T_{x(i)}$
3. Repeat Steps 1 and 2 for all the nodes in X
4. output T_{new}

5.2.6 Nodes Moving in the Network (Edge Weight Modification): Performance Results and Analysis

The $2HOP - MOV$ algorithm is tested using the simulator defined in Section 5.1.2. To simulate this algorithm and compare the results with the two-hop IP algorithm, we start with an initial distribution of 30 nodes. Then a number of nodes are consecutively moving in the network. The number of nodes vary between 1 and 30. When a node moves in the network, the distances with BS and with other nodes vary. Therefore, the bit rates to communicate with the BS and with other nodes vary according to the definition of bit rate in Section 3.1. The new graph produced by the movement of the nodes is solved by the two-hop IP algorithm and $2HOP - MOV$ algorithm. This scenario is repeated for 10000 iterations. For each iteration, the energy consumption and the CPU runtime are recorded for the two-hop IP algorithm and for $2HOP - MOV$ algorithm. For each solution, the average energy consumption and average CPU runtime are computed and presented in Figure 5.20 and 5.21 respectively.

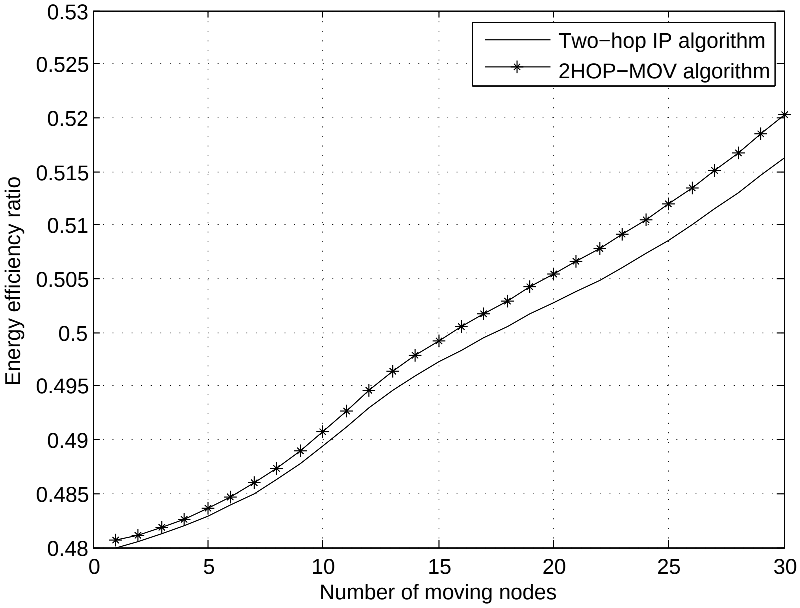


Figure 5.20: Energy efficient ratio vs. number of moving nodes for two-hop algorithms.

Figure 5.20 shows the energy efficiency ratio versus the number of moving nodes. The energy efficiency ratio is the ratio of the energy consumption of the nodes in cooperative scenario to the energy consumption in the non-cooperative scenario. The energy consumption of the MST solution generated by the $2HOP - MOV$ algorithm proposed in this thesis is relatively close to the energy consumption of the two-hop solution generated by the two-hop IP formulation. Although the energy consumption of $2HOP-MOV$ algorithm is barely higher than the two-hop IP formulation it still maintains the gain of the cooperative content distribution among nodes.

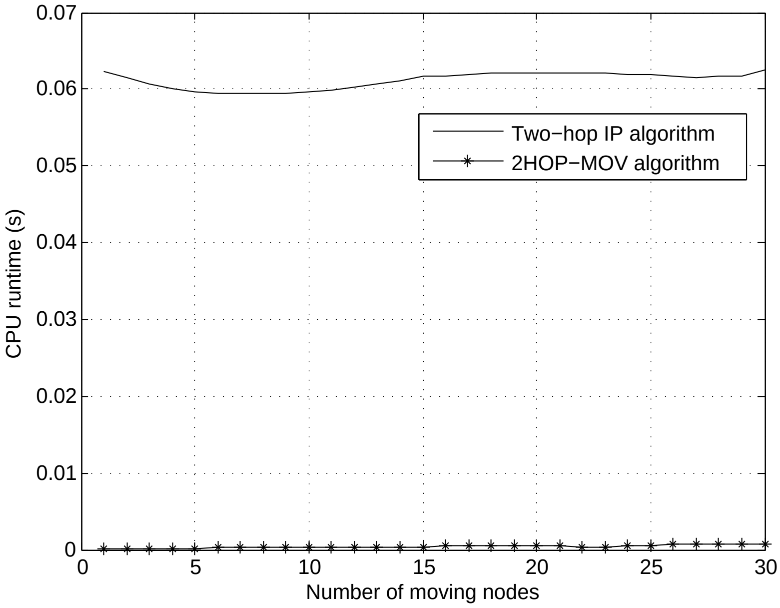


Figure 5.21: CPU runtime (s) vs. number of moving nodes for two-hop algorithms.

Figure 5.21 shows the CPU runtime (in seconds) of the $2HOP-MOV$ algorithm versus the number of deleted nodes. The CPU runtime of $2HOP-MOV$ algorithm is very small compared to that of the two-hop IP formulation (The CPU runtime of $2HOP-MOV$ algorithm is less that 1 percent of that of the two-hop IP formulation). Therefore, $2HOP-MOV$ algorithm has better computational complexity than the two-hop IP formulation.

Chapter 6

Dynamic Cooperative Content Distribution Architecture

In this chapter, a dynamic application layer architecture for the cooperative content distribution is proposed. This architecture includes the intelligence of the cooperation in a dynamic content distribution scenario.

The proposed intelligence defines generic efficient mechanisms for network formation and recovery. The main components of the architecture are (1) the network formation, (2) the detection and selection mechanisms, and (3) the failure detection and recovery mechanisms.

6.1 General Overview

In a cooperative content distribution scenario, MTs interested in the same content cooperate with each other to reduce the total energy consumption of downloading this content. When a MT requests to download a content from a multimedia service provider, it has the option to cooperate with other MTs or to individually download this content. The option of cooperation can be configured at the level of the operating system of the MT (for example, a smart phone can be configured to cooperate via WiFi connections and not to cooperate while using other packet data networks) or in the settings of the application used to request the service. Moreover, the user can be asked by the service provider while requesting the service if he is willing to cooperate or not.

Similarly, each multimedia service provider has the choice to enable cooperative content distribution or not. Enabling cooperation requires the service provider to maintain a framework dedicated to manage the content distribution among the MTs. This framework should include the cooperative content distribution model that determines how the MTs cooperate while downloading a given content. Moreover, it should have a dedicated database to manage the cooperation and special communication mechanisms to coordinate the content distribution among the MTs. The service provider can enable cooperation in dense areas only such as train stations, airports, malls and disable it in other areas. To locate the MTs and determine in which area and in the vicinity of which AP each MT is located, MTs must send location information (GPS for example) while requesting the service. Moreover, service providers can benefit from the information provided by the internet service providers (ISPs) and mobile network operators (MNOs) to determine the locations of the MTs. The cooperative content distribution framework could be integrated within the service provider server or could be itself a service provided by a third party service provider or a mediated service agent. For simplicity, we will consider in this work that the cooperative content distribution framework is installed within the service provider server. The design of this framework is out of the scope of this thesis. We will refer to its components generally throughout this chapter.

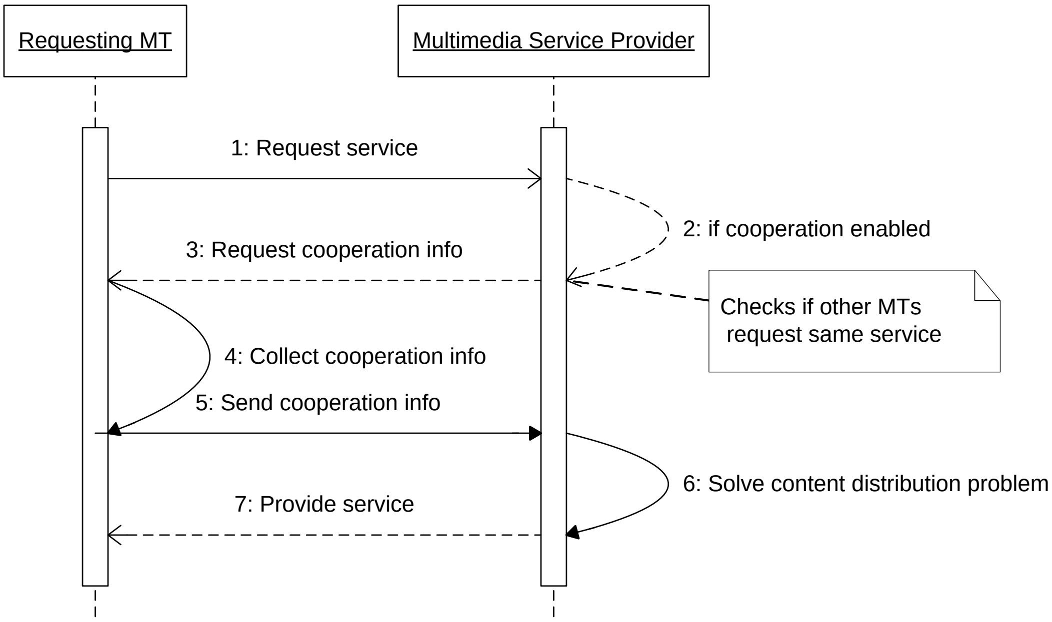
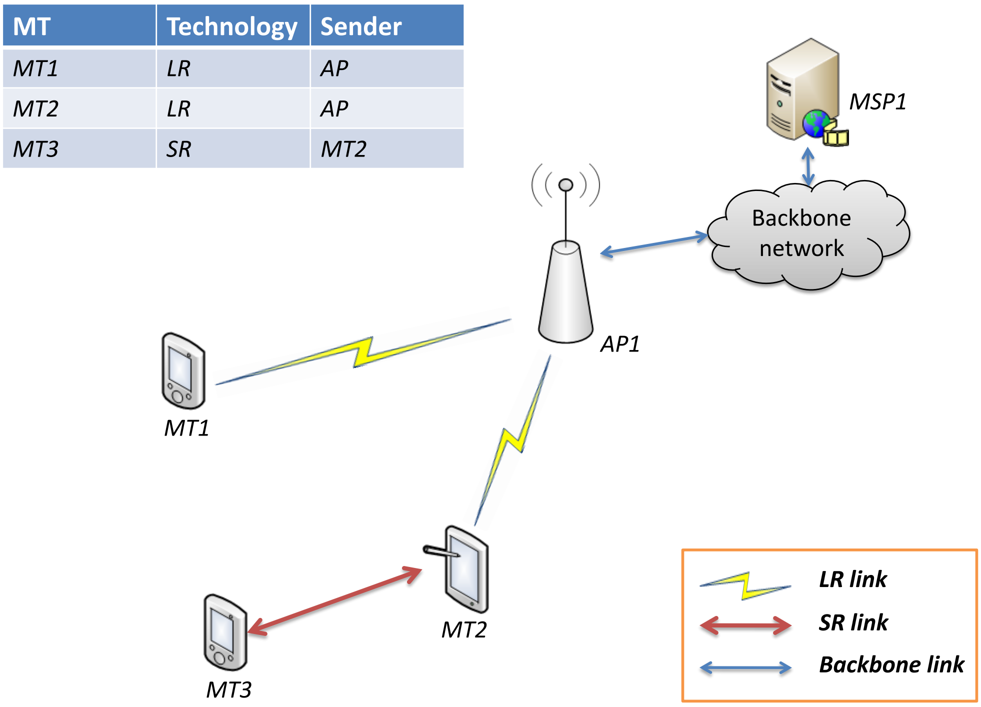


Figure 6.1: Sequence diagram of interaction between requesting MT and multimedia service provider.

The sequence diagram in Figure 6.1 shows the general interactions between a requesting MT and a multimedia service provider when cooperation is enabled. When a user A is interested in a service (file download, video streaming, online games, etc.), it communicates with the multimedia service provider MSP (Step

1) using its mobile terminal MT_A via an access point $AP1$ of one of the LR technologies (WIFI, 3G/4G technologies). If MT_A or MSP do not allow cooperation or there is no other MT in the vicinity of $AP1$ interested in the same service, MT_A will receive the requested service individually using a LR technology. However, if both MT_A and MSP enable cooperation, the service provider checks if other MTs in the vicinity of $AP1$ are interested in the same service (Step 2). Then MSP request all necessary information from MT_A and its neighbouring MTs (Step 3). MT_A collects the cooperation information (Step 4) and sends them back to MSP (Step 5). After receiving all the necessary cooperation information, MSP then utilises its dynamic cooperative content distribution framework to find an energy efficient solution to distribute the content among the requesting MTs (Step 6). Then MSP will provide the service to MT_A and other MTs following the retrieved solution (Step 7).



MT	Technology	Sender
MT1	*LR*	*AP*
MT2	*LR*	*AP*
MT3	*SR*	*MT2*

Figure 6.2: Example of the content distribution plan solved by the cooperative content distribution framework.

We mean by the solution of the dynamic cooperative content distribution framework, a content distribution plan among the cooperating MTs that determines which MTs will receive the content directly from the AP using a LR technology and which MTs will receive it from other MTs using a SR technology. Figure 6.2 shows a general content distribution plan solved by the cooperation framework for a snapshot network scenario consisting of three MTs $MT1$, $MT2$, and $MT3$.

In this example, $MT1$ and $MT2$ receive the content from the access point AP using a LR technology. $MT3$ receives the content from $MT2$ using a SR technology.

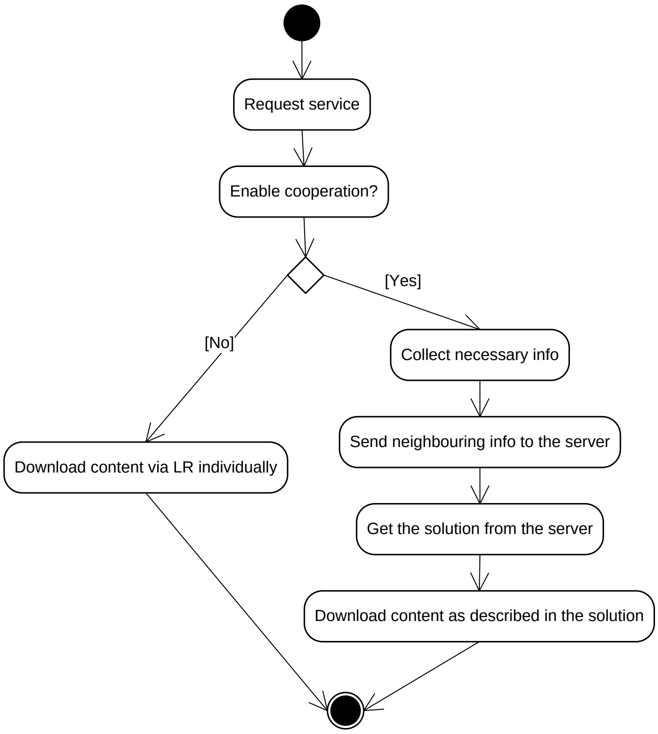


Figure 6.3: Activity diagram of requesting MT.

Figures 6.3 and 6.4 show general activity diagrams of the decision models of the requesting MT and the multimedia service provider respectively in a cooperative scenario. For a cooperative scenario to occur, three conditions must be fulfilled:

- The multimedia service provider allows cooperation.
- At least two MTs in the vicinity of the same AP are interested in the same service.
- Those MTs can communicate with each other using a SR technology and allow cooperation.

As shown in Figure 6.4, to solve the cooperative content distribution problem, the service provider, where the cooperative content distribution framework is installed, should first collect the necessary information to build the cooperative content distribution problem and then solve this problem using one of the algorithms described in Section 3. The cooperative content distribution framework requires collecting the following information:

- The bit rates of the direct LR links between the AP and each of the MTs that are in the vicinity of this AP and that are interested in the same content.

- The bit rates of the SR links among the MTs that are in the vicinity of the AP and are interested in the same content. The bit rate of the link between the MTs that could not be linked is considered zero (The weight of the corresponding edge will be infinity as defined in Section 3.2).

- The transmission power (in Joules/seconds) of all the cooperating MTs using the LR technology.

- The transmission and receiving powers (in Joules/seconds) using the SR technology.

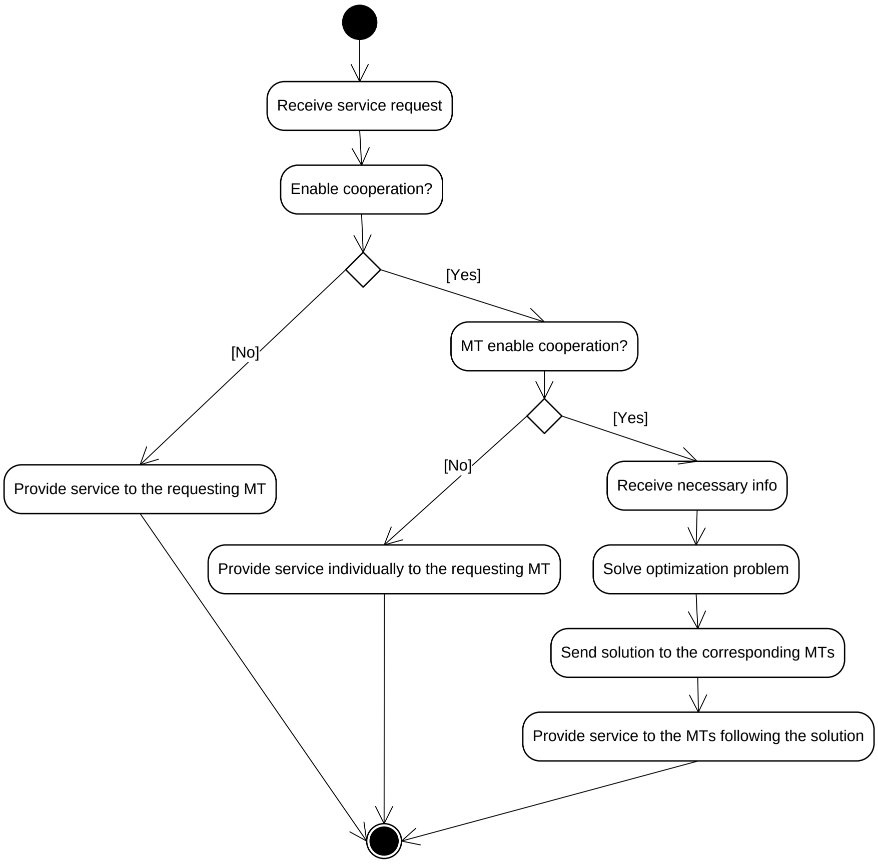


Figure 6.4: Activity diagram of service provider.

The way the bit rates are collected and transmitted to the cooperative content distribution server is technology dependent. For most of the Short-Range (SR) and Long-Range(LR) technologies, control packets or beacons (for example, RTS and CTS packets for IEEE 802.11) are transmitted between the sender and receiver of a link to determine the bit rate to be used for each link. Usually for each technology, there is a list of possible bit rates used. For IEEE 802.11b, the possible bite rates that can used are 1, 2, 5.5, 11 Mbit/s.

The transmission and receiving powers of the wireless interfaces depend on the wireless technology used. These powers should be either available by the manufacturers of the MTs or should be estimated using an energy consumption model installed on each MT. The optimal case is to have an energy consumption application that runs this model to estimate the real energy consumption of each mobile service and application taking into consideration the processing energy consumption in addition of the transmitting and receiving energy consumptions. In the literature, many energy consumption models are suggested to estimate application energy consumption in addition to transmission and reception energy consumption [51] [52]. In this work, we will assume that each MT can transmit its transmission and receiving powers for both LR and SR wireless interfaces.

A MT can easily detect the bit rate of the LR link with the AP. However, on the SR, each MT has to initiate P2P connection with all its neighbours to detect the bit rates with all neighbouring MTs. For instance, if the clients are using one of the IEEE 802.11 technologies as a SR technologies, the MTs can either activate the "802.11 ad hoc mode" or use the "WI-FI Direct technology" defined in Section 2.4. These two options allow the MTs to connect directly to each other without the need to have a centralized 802.11 AP. Moreover, in IEEE 802.11, MTs can automatically setup an ad hoc network without a controlling AP. The resulting, as shown in Section 2.4, is called an independent basic service set (IBSS). The MTs in the same IBSS share the same basic service set identification (BSSID). For an IBSS, the BSSID is a locally administered MAC address generated from a 46-bit random number. The BSSID is used to identify the MTs that cooperate to distribute the same content in an ad-hoc network.

6.2 Network Formation and Selection Mechanisms

In this section, we present how MTs communicate with each other and with the multimedia service provider in a cooperative scenario.

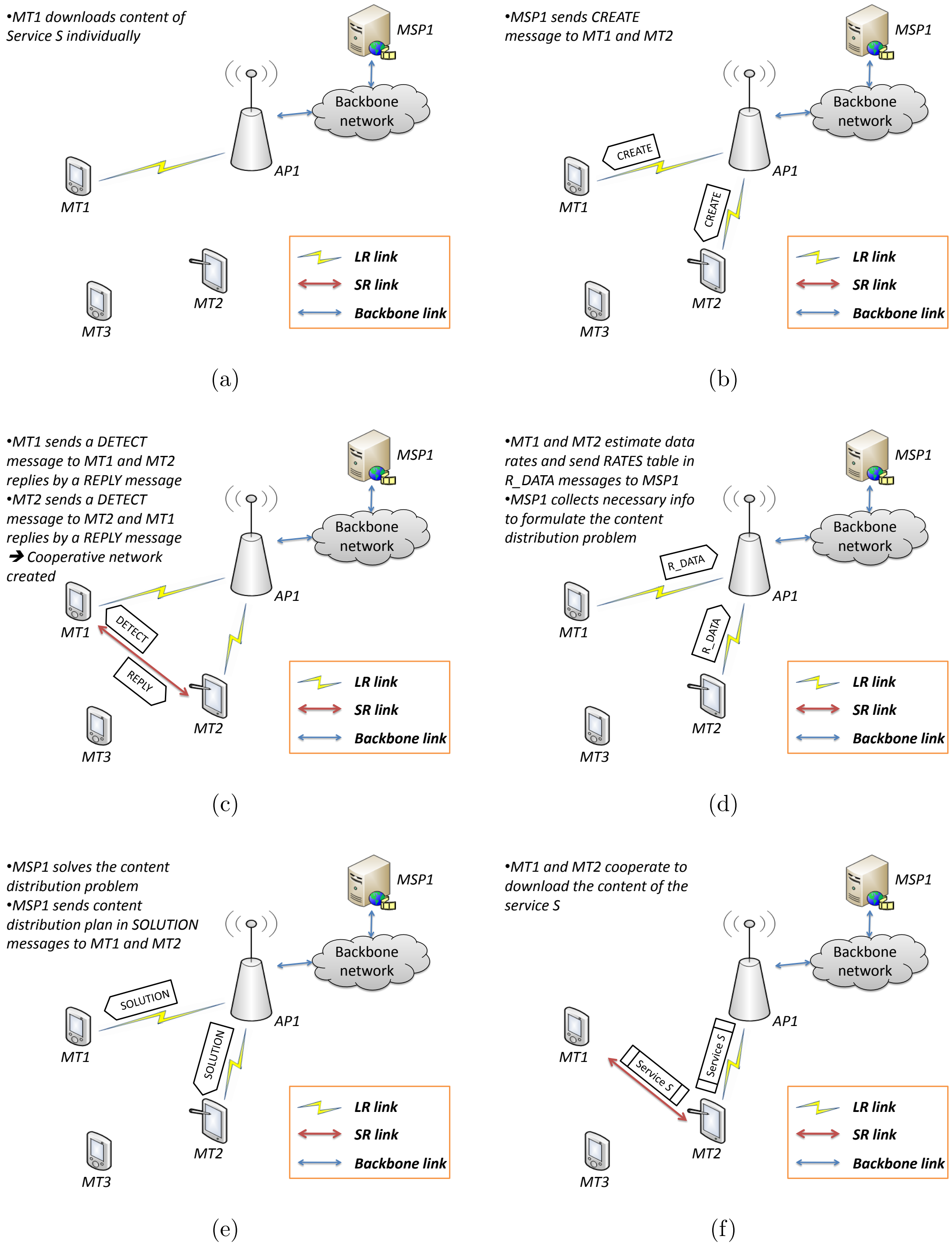
The same example shown in Figure 6.2 is used throughout this section to illustrate the proposed mechanisms. In Figure 6.2, three MTs $MT1$, $MT2$, and $MT3$ are located in the vicinity of the same access point $AP1$ of a LR technology. $MT1$, $MT2$, and $MT3$ can connect to each other using a SR technology.

6.2.1 Initialization Phase

When $MT1$ requests a multimedia service S from the multimedia service provider $MSP1$, it is asked if it is willing to cooperative while using this service or not (if the option to cooperate is not configured in the settings of the application or the OS of $MT1$). Since $MT1$ is the only MT interested in service S in the vicinity of $AP1$, $MSP1$ provides the service to $MT1$ via LR link as shown in Figure 6.5(a). $MSP1$ maintains in its database that $MT1$ is requesting service S in the vicinity of $AP1$ and it enables cooperation. In this example all the MTs are configured to cooperate while using the sevice S.

When another MT ($MT2$) sends a request for service S to $MSP1$, $MSP1$ checks if there are other MTs in the vicinity of $AP1$ that request the same service. Since $MT1$ already requests the service S, $MSP1$ sends $CREATE$ messages to $MT1$ and $MT2$ as shown in Figure 6.5(b). This message contains the list of other MTs interested in the same service in the vicinity of $AP1$ and the name of the service provider ($MSP1$) and the requested servide (S). Therefore, the $CREATE$ message sent to $MT2$ contains the MAC address of $MT1$ and similarly the $CREATE$ message sent to $MT1$ contains the MAC address of $MT2$. The $CREATE$ message contains also the service id (such as BSSID for IEEE 802.11 technology) of the ad hoc network established among the MTs interested in service S. Since the ad hoc network is not establish yet, the field of service id in the $CREATE$ message sent to $MT1$ and $MT2$ is empty. When receiving the $CREATE$ message, since there is no ad hoc network already created, $MT1$ and $MT2$ initialize an ad hoc network using SR technology and generate a service id that will characterise all the MTs that join this network later.

Each MT sends a $DETECT$ message to each of its neighbouring MTs. All reachable MTs reply by $REPLY$ messages (Figure 6.5(c)). Each MT uses the $DETECT$ and $REPLY$ messages to estimate the bit rates of sending and receiving on the SR with all neighbouring MTs. Each MT maintains a $RATES$ table containing the list of neighbouring MTs interested in the same service (this list were sent initially by the multimedia service provider in the $CREATE$ message) and the bit rate to communicate with each of these MTs using SR technology. The $RATES$ table of $MT1$ contains a record for $MT2$ and the estimated bit rate to communicate with $MT2$ using SR technology. Similarly, $MT2$ maintains a $RATES$ table that contains a record for $MT1$ and the estimated bit rate to communicate with it using SR technology. The MTs $MT1$ and $MT2$ then sends a message R_DATA to $MSP1$ (Figure 6.5(d)). R_DATA message includes all the necessary information (described in Section 4.1) used in the formulation of the content distribution problem. Moreover, R_DATA messages include the service id generated while initializing the ad hoc network. The multimedia service provider $MSP1$ records the service id in its database.



(a)
- MT1 downloads content of Service S individually

(b)
- MSP1 sends CREATE message to MT1 and MT2

(c)
- MT1 sends a DETECT message to MT1 and MT2 replies by a REPLY message
- MT2 sends a DETECT message to MT2 and MT1 replies by a REPLY message

➔ Cooperative network created

(d)
- MT1 and MT2 estimate data rates and send RATES table in R_DATA messages to MSP1
- MSP1 collects necessary info to formulate the content distribution problem

(e)
- MSP1 solves the content distribution problem
- MSP1 sends content distribution plan in SOLUTION messages to MT1 and MT2

(f)
- MT1 and MT2 cooperate to download the content of the service S

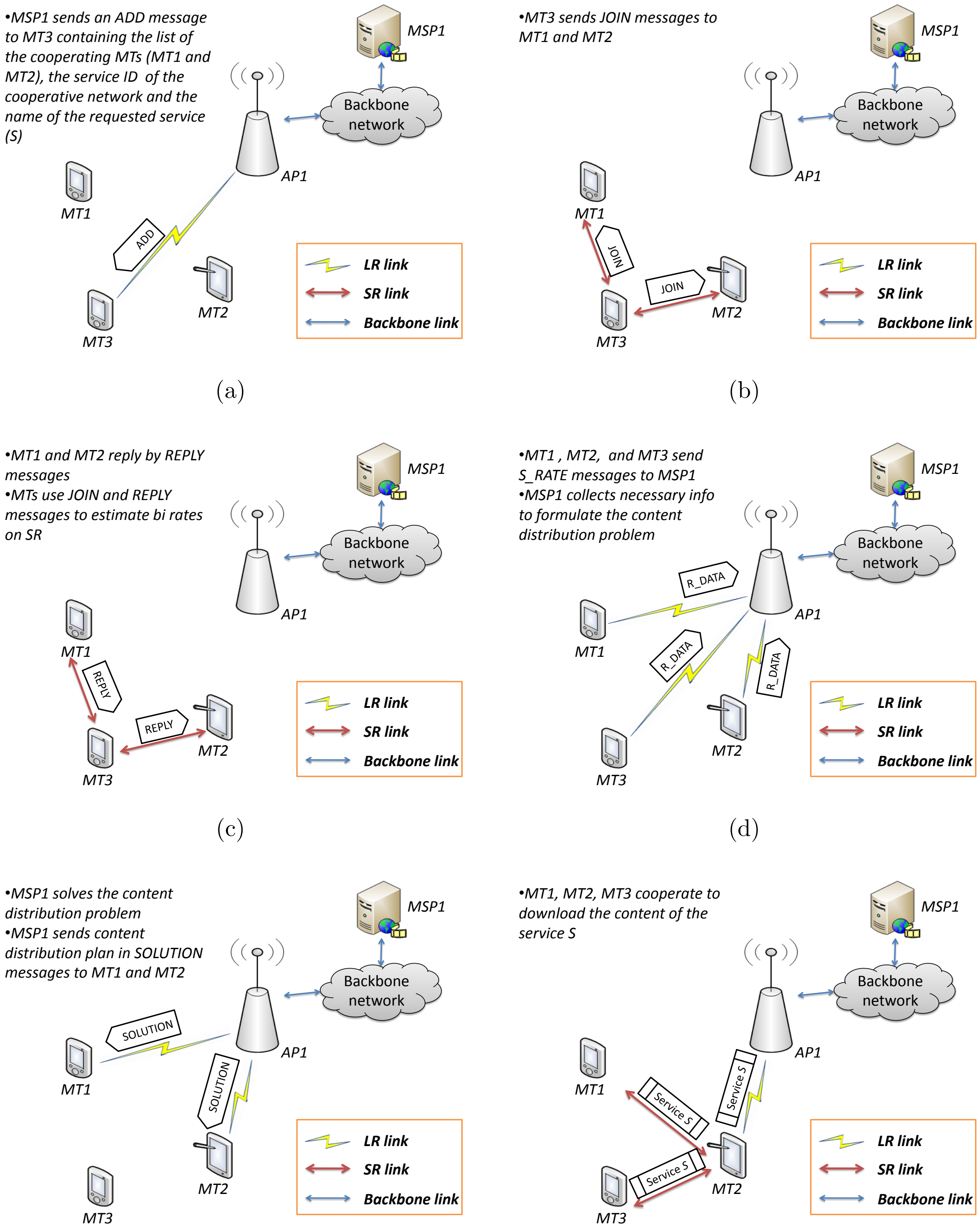
Figure 6.5: Initialization of the cooperative content distribution network.

When the $MSP1$ receives all the necessary information (receives R_DATA messages from all MTs interested in the same service), it formulates the cooperative content distribution problem using one of the algorithms defined in Section 3 and solved it. It then sends a $SOLUTION$ message to each of the cooperating MTs ($MT1$ and $MT2$) as shown in 6.5(e). The $SOLUTION$ message describes how the cooperating MTs will download the content (download it on the LR or receive from other MT using SR technology) as shown in Figure 6.5(f).

6.2.2 Node Addition Phase

If a new MT $MT3$, already exiting in the vicinity of $AP1$ (as shown in 6.5(f)), requests the same service S from the multimedia service provider $MSP1$, an addition mechanism must defined to allow this MT to join the cooperating MTs ($MT1$ and $MT2$). $MSP1$ checks if there are other MTs in the vicinity of the AP interested in the same service. It finds that $MT1$ and $MT2$ are effectively cooperating in downloading the content of the service S. Therefore, a cooperative content distribution network is already established and the MT $MT3$ must be added to this network to cooperate with other MTs. This is equivalent to the node addition (node entering the network) described in Chapter 4.

To add $MT3$ to the cooperative network in the vicinity of $AP1$, $MSP1$ sends an ADD message to $MT3$ containing the list of MTs cooperating in downloading the content of the service S along with service id of the ad hoc network established among these MTs, the name of service provider and the requested service (Figure 6.6(a)). As shown in 6.6(b), $MT3$ then sends $JOIN$ messages to all the cooperating MTs ($MT1$ and $MT2$). When receiving the $JOIN$ messages from $MT3$ and verifying that $MT3$ is trying to join the cooperative network, $MT1$ and $MT2$ sends back $REPLY$ messages (Figure 6.6(c)). The $JOIN$ and $REPLY$ messages are used by the MTs to estimate the bit rates among MTs using the SR technology. Then, as in the initialization of the cooperative network, $MT3$ creates a $RATES$ table and sends a R_DATA message to $MSP1$. Similarly, $MT1$ and $MT2$ update their $RATES$ tables by adding a record for the new added MT ($MT3$) and send R_DATA message to $MSP1$ (Figure 6.6(d)). $MSP1$ then collects all the necessary info to formulate the cooperative content distribution problem. $MSP1$ can use one of the node addition pseudo-optimal algorithms defined in Chapter 5 to solve the content distribution problem. After solving the problem, $MSP1$ sends back $SOLUTION$ messages (Figure 6.6(e)) to all cooperating nodes ($MT1$, $MT2$, and $MT3$) as in the initialization of the cooperative network. The MTs then cooperate to download the content of the service S as shown in Figure 6.6(f).



- MSP1 sends an ADD message to MT3 containing the list of the cooperating MTs (MT1 and MT2), the service ID of the cooperative network and the name of the requested service (S)

(a)

- MT3 sends JOIN messages to MT1 and MT2

(b)

- MT1 and MT2 reply by REPLY messages
- MTs use JOIN and REPLY messages to estimate bi rates on SR

(c)

- MT1 , MT2, and MT3 send S_RATE messages to MSP1
- MSP1 collects necessary info to formulate the content distribution problem

(d)

- MSP1 solves the content distribution problem
- MSP1 sends content distribution plan in SOLUTION messages to MT1 and MT2

(e)

- MT1, MT2, MT3 cooperate to download the content of the service S

(f)

Figure 6.6: Addition to the cooperative content distribution network.

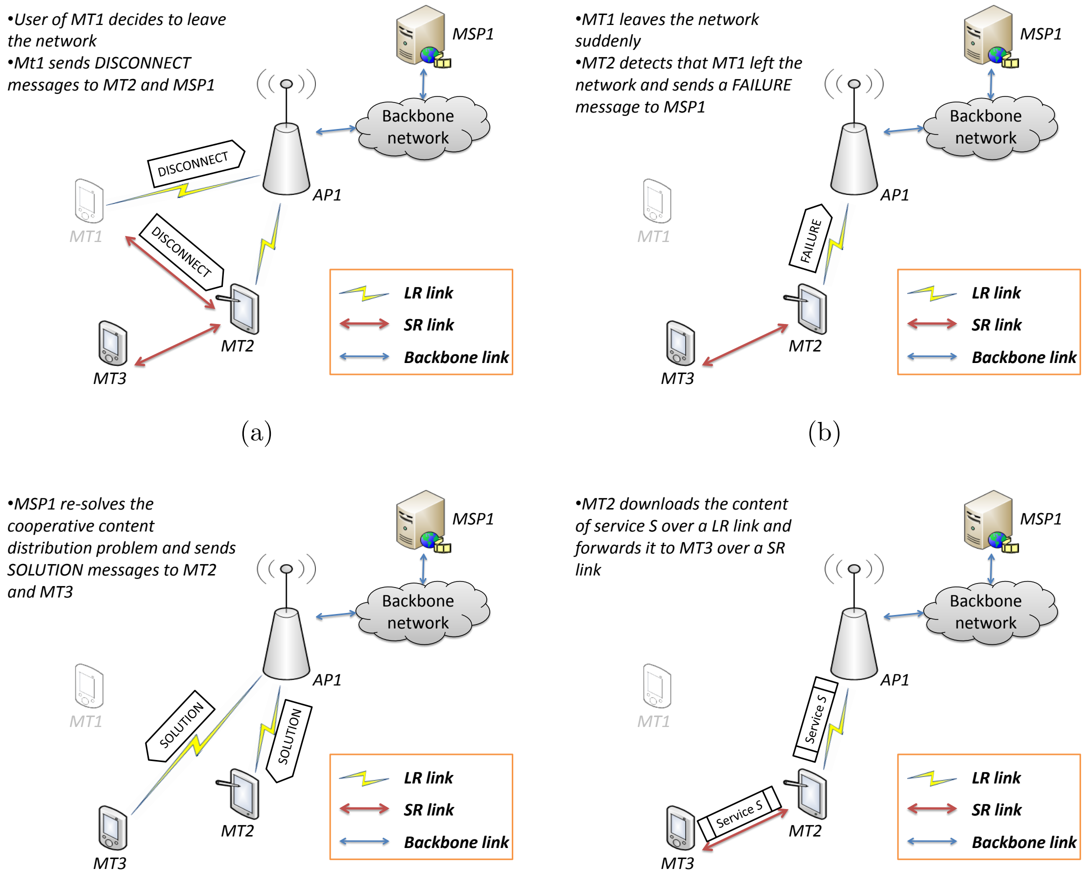
6.3 Failure Detection and Recovery Mechanisms

In a mobile network, the MTs are moving and the channel conditions change over time. MTs can leave the network at any point in time; the MT user may decide to stop downloading the content or it moves and leaves the range of the AP. Moreover, while cooperating with other MTs, a MT can move and become unreachable from other MTs.
To account for the mobility in the network, a failure detection and recovery mechanism is defined. This mechanism takes into consideration the different aspects of mobility in the network. The two main causes of failure in a content distribution network are: (1) When a MT leaves the network for any reason and (2) when a MT moves in the network and becomes unreachable from some cooperating MTs but it is still in the range of the AP and/or some other MTs. These two causes are related to the node deletion and edge weight modification aspects algorithms presented in Chapter 4.

6.3.1 Failure Mechanism for Node Deletion

In a cooperative content distribution network, a set of MTs cooperates in downloading a content of a service provided by a multimedia service provider. A MT can leave the cooperative network by either moving outside the range of AP or by stop using the common service. If a MT in a cooperative network decides to stop downloading the content of a given service, it sends a $DISCONNECT$ message to the service provider and to the MTs it was cooperating with. The $DISCONNECT$ message includes the MAC address of the MT that left the network. Then the service provider deletes the information related to this MT and reformulates the cooperative content distribution problem and sends the distribution plan using $SOLUTION$ messages to other MTs. Alternatively, the cooperative content distribution framework can use one of the node deletion algorithms presented in Chapter 5 to find a quick and pseudo-optimal energy efficient solution for the content distribution plan among the MTs.

If a MT leaves suddenly the network (user turns off the MT, or becomes outside the range of AP), the neighbouring MTs that were exchanging (sending or receiving) the common content with this MT detect that this MT left the network and send $FAILURE$ message to the service provider. The $FAILURE$ message includes the MAC address of the MT that left the network. Accordingly, the service provider then reformulate ans re-solve the cooperative content distribution problem using one the algorithms presented in Chapter 5. The service provider then sends $SOLUTION$ messages to all the remaining MTs in the network. The $SOLUTION$ messages include the content distribution plan of the cooperating MTs.



- User of MT1 decides to leave the network
- Mt1 sends DISCONNECT messages to MT2 and MSP1

(a)

- MT1 leaves the network suddenly
- MT2 detects that MT1 left the network and sends a FAILURE message to MSP1

(b)

- MSP1 re-solves the cooperative content distribution problem and sends SOLUTION messages to MT2 and MT3

(c)

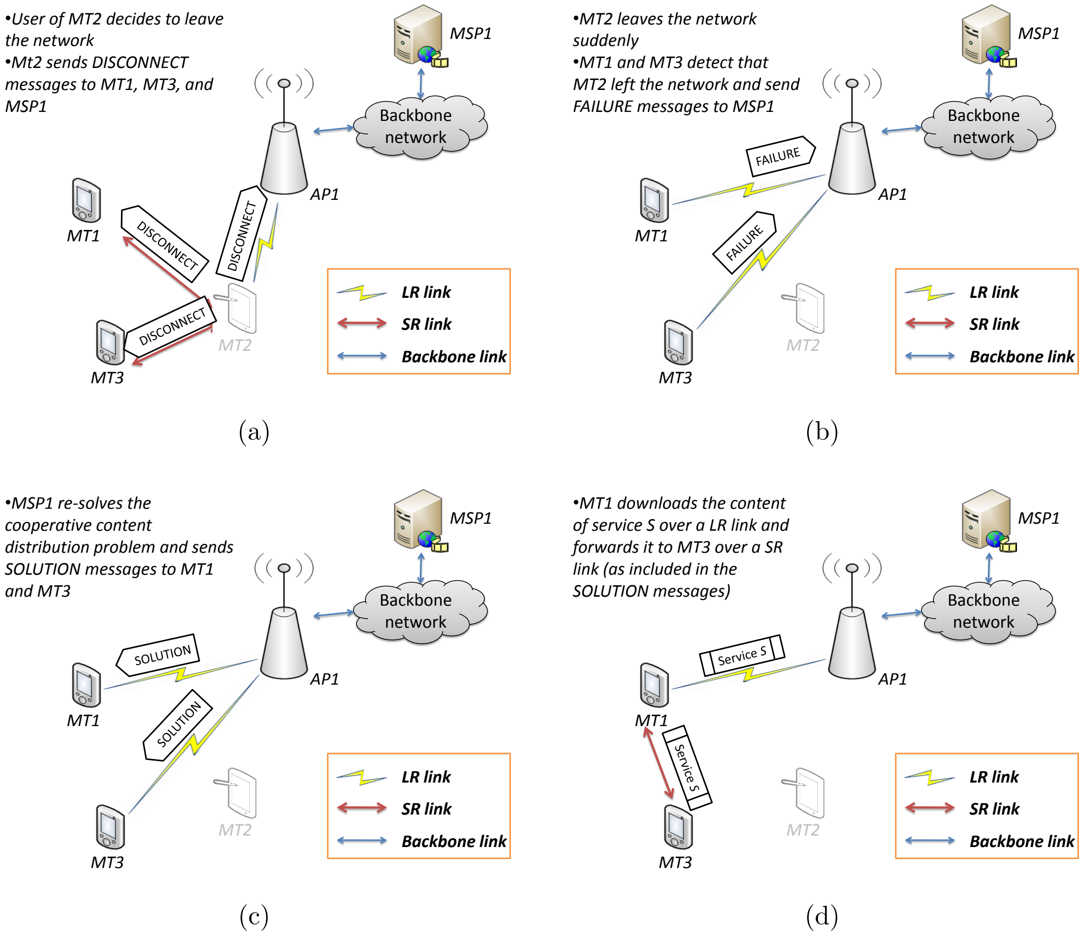
- MT2 downloads the content of service S over a LR link and forwards it to MT3 over a SR link

(d)

Figure 6.7: Failure mechanism example for leaf MT deletion.

When a failure occurred in the cooperative network, a recovery mechanism to retransmit the content packets is needed. If the MT that left the network is a leaf MT, packet retransmission is not needed. However, if the MT that left had children MTs in the cooperative network, the provider server, before re-solving the content distribution problem, retransmits the lost content packets to all the children of this MT over LR links.

To illustrate the failure mechanisms for MT deletion in a cooperative scenario, we start from the network snapshot in Figure 6.6(f). In this cooperative content distribution example, $MT2$ receives the content packets on a LR link and forwards each packet to $MT1$ and $MT2$ using SR links. If the user of $MT1$ decides to leave the cooperative network, $MT1$ sends $DISCONNECT$ messages to the multimedia service provider $MSP1$ and to $MT2$ (Figure 6.7(a)). However, if $MT1$ leaves suddenly the network, $MT2$ sends a $FAILURE$ message to $MSP1$ notifying that $MT1$ left the network (Figure 6.7(b)). For both cases,



(a) (b)

(c) (d)

Figure 6.8: Failure mechanism example for parent MT deletion.

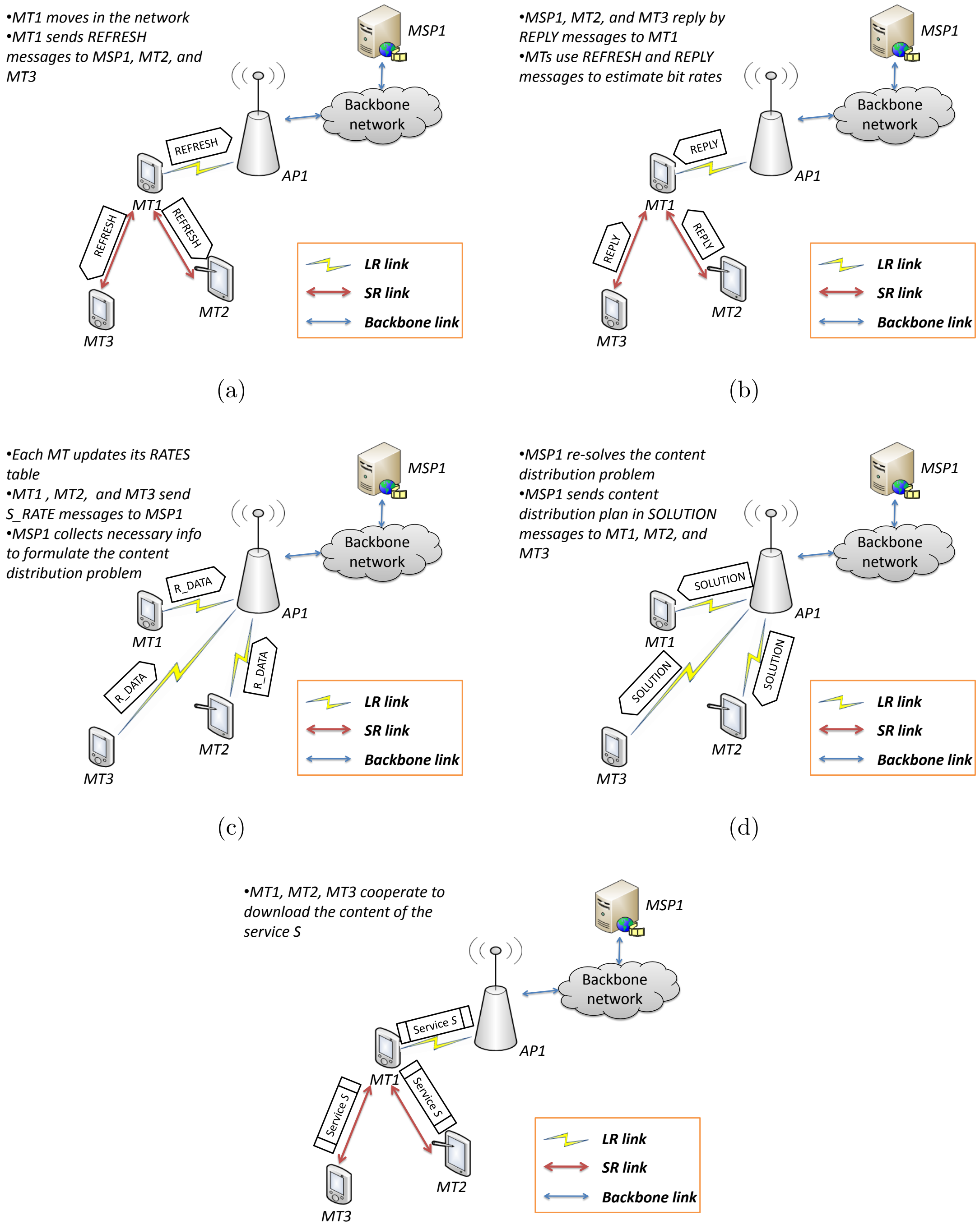
$MSP1$ reformulates the cooperative content distribution problem and re-solves it. Then it sends $SOLUTION$ messages (as shown in Figure 6.7(c) that include the content distribution plan to the remaining MTs ($MT2$ and $MT3$). However, if $MSP1$ is using one of the node deletion re-optimization algorithms ($MST - DEL$, $2HOP - DEL1$, or $2HOP - DEL2$) presented in Chapter 5, there is no need to reformulate and re-solve the cooperation content distribution problem since $MT2$ was a leaf MT. Finally $MT2$ and $MT3$ continue downloading the content of service S as described in the $SOLUTION$ messages (Figure 6.7(d)).

Starting from the same example in Figure 6.6(f), if the user of a parent MT such as $MT2$ decides to leave the network, $MT2$ sends $DISCONNECT$ messages to $MT1$, $MT3$, and $MSP1$ (Figure 6.8(a)). However, if $MT2$ suddenly leaves the network, $MT1$ and $MT3$ send $FAILURE$ messages to $MSP1$ that include the MAC address of $MT2$ (Figure 6.8(b)). In both cases, $MSP1$ stops sending the content packets and re-solve the optimization problem. If $MT1$ and $MT3$ did not receive the old content packets that were sent to $MT2$, $MSP1$ sends the missing packets over LR links. Then, it sends the $SOLUTION$ messages to the remaining MTs as shown in Figure 6.8(c). Finally, $MT1$ and $MT3$ re-establish the cooperative network and start downloading the content of service S following the content distribution plan included in the $SOLUTION$ messages (Figure 6.8(d)).

6.3.2 Failure Mechanism for Node Movement

In a dynamic network, MTs are moving and channel conditions are varying over time. The variations of the distances and of the channel conditions change the bit rates selected by each of the MTs to communicate with each other and with the AP. In this cooperative content distribution architecture, the MTs uses the exchanged content data packets to estimate the bit rates of SR links. Moreover, MTs estimate the bit rates of the LR links with the AP using the control and acknowledgement packets. If a MT detects a variation in the bit rates of the links with other MTs or with AP, it sends $REFRESH$ messages of all existent MTs in the cooperative network. After receiving the $REFRESH$ message, each MT replies by a $REPLY$ message. The $REFRESH$ and $REPLY$ messages are used to estimate the bit rates among the MTs. Then each MT updates its $RATES$ table that includes the bit rates of the SR links with other MTs and of the LR link with AP. Then, it sends a R_DATA message to the service provider which reformulate the cooperative content distribution problem accordingly and re-solve the problem. Alternatively, the service provider can use one of the edge weight modification re-optimization algorithms presented in Chapter 5. After solving the content distribution plan, the service provider sends $SOLUTION$ messages to the cooperating MTs that include the updated content distribution plan.

To illustrate the failure mechanism for MT movement, we consider the cooperative content distribution scenario presented in Figure 6.6(f). If $MT1$ moves in the network, it sends $REFRESH$ messages to $MSP1$, $MT2$, and $MT3$ (Figure 6.9(a)). $MSP1$, $MT2$, and $MT3$ then reply by $REPLY$ messages (Figure 6.9(b)). The MTs in the cooperative network use the $REFRESH$ ans $REPLY$ messages to re-estimate the bit rates of the SR links among each other. Each MT then updates its $RATES$ table accordingly. Then each MT sends its table $RATES$ to $MSP1$ using a R_DATA message (Figure 6.9(c)). $MSP1$ collects all necessary information and re-solves the content distribution problem. Then, it sends $SOLUTION$ messages to the cooperating MTs (Figure 6.9(d)). $MT1$, $MT2$ and $MT3$ cooperate in downloading the content of service S following the distribution plan included in the $SOLUTION$ messages (Figure 6.9(e)).



(a)
- MT1 moves in the network
- MT1 sends REFRESH messages to MSP1, MT2, and MT3

(b)
- MSP1, MT2, and MT3 reply by REPLY messages to MT1
- MTs use REFRESH and REPLY messages to estimate bit rates

(c)
- Each MT updates its RATES table
- MT1 , MT2, and MT3 send S_RATE messages to MSP1
- MSP1 collects necessary info to formulate the content distribution problem

(d)
- MSP1 re-solves the content distribution problem
- MSP1 sends content distribution plan in SOLUTION messages to MT1, MT2, and MT3

(e)
- MT1, MT2, MT3 cooperate to download the content of the service S

Figure 6.9: Failure mechanism for MT movement.

Chapter 7

Conclusions

In this thesis, the topic of cooperative content distribution in a dynamic network is tackled. This problem is formulated using a Minimum Spanning Tree (MST) formulation and a two-hop IP formulation. Starting from an optimal static solution for both the MST formulation and the two-hop formulation, re-optimization algorithms are proposed for each of the list the mobility aspects that were considered in a dynamic network. These algorithms were able to return close-to-optimal solutions with low computational complexity compared to re-solving the optimal optimization problem from scratch for each variation in the network. Moreover, a generic application layer architecture of the dynamic content distribution model is presented.

To test the proposed algorithms, we implemented a customized simulator where nodes are entering, leaving and moving in the network randomly. An interesting extension of the thesis work would be to test the performance of the proposed algorithms using a network simulator with realistic mobility models, e.g., ns-2 [53] or OMNeT++ [54]. Moreover, the real test bed can be developed using Android mobile applications to test the proposed algorithms under realistic operational conditions.

A specific MAC protocol that accounts for both the content distribution and the mobility in the network needs to be designed. This MAC protocol should focus on the recovery and acknowledgement mechanisms that guarantee the reception of the content in a cooperative content distribution scenario. Finally, an important future research direction is to consider the security and privacy aspects of mobile-to-mobile cooperation in order to avoid possible types of attacks.

Bibliography

[1] F. Fitzek and M. Katz, "Cooperation in wireless networks: Principles and applications," in *Springer*, 2006.

[2] P. Havinga and G. Smit, "Energy-efficient wireless networking for multimedia applications," in *Journal of Wireless Applications and Mobile Computing, Wiley*, vol. 1, pp. 165–184, June 2001.

[3] A. Vosoughi and Y. Jia, "Maximizing throughput in cooperative networks via cross-layer adaptive designs," in *Proceedings of IEEE Sarnoff Symposium*, April 2010.

[4] L. Wang and K. Liu, "An throughput-optimized cooperative routing protocol in ad hoc network," in *Proceedings of IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, October 2009.

[5] J. Chen, S. Li, S.-H. Chan, and J. He, "Wiani: wireless infrastructure and adhoc network integration," in *Proceedings of IEEE International Conference on Communications*, May 2005.

[6] J. Chen, S.-H. Chan, and S.-C. Liew, "Mixed-mode wlan: the integration of ad hoc mode with wireless lan infrastructure," in *Proceedings of IEEE Global Telecommunications Conference*, December 2003.

[7] H. Im, Y. Lee, and S. Bahk, "Incentive-driven content distribution in wireless multimedia service networks," in *Proceedings of IEEE Global Telecommunications Conference*, December 2010.

[8] M. Leung and S. G. Chan, "broadcast-based peer-to-peer collaborative video streaming among mobiles," *IEEE Transactions on Broadcasting*, vol. 53, pp. 350–361, March 2007.

[9] M. Katz and F. Fitzek, "Cooperative techniques and principles enabling future 4g wireless networks," in *Proceedings of the International Conference on "Computer as a Tool"*, November 2005.

[10] L. Cao, J. Zhang, and N. Kanno, "Relay-coded multi-user cooperative communications for uplink lte-advanced 4g systems," in *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, September 2009.

[11] S. Kang and M. Mutka, "A mobile peer-to-peer approach for multimedia content sharing using 3g/wlan dual mode channels," *Journal of Wireless Communications and Mobile Computing*, vol. 5, pp. 633–645, September 2005.

[12] J.-W. Yoo and K. H. Park, "A cooperative clustering protocol for energy saving of mobile devices with wlan and bluetooth interfaces," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 491–504, April 2011.

[13] L. Al-Kanj and Z. Dawy, "Optimized energy efficient content distribution over wireless networks with mobile-to-mobile cooperation," in *Proceedings of International Conference on Telecommunications*, April 2010.

[14] S.-S. Kang and M. W. Mutka, "Mobile peer membership management to support multimedia streaming," in *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pp. 770–775, May 2003.

[15] M. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 2915–2934, July 2006.

[16] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions ON Information Theory*, vol. 43, pp. 1067–1073, May 1997.

[17] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1936–1948, December 1992.

[18] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, vol. 39, pp. 150–154, February 2001.

[19] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 89–103, January 2005.

[20] W. Ye and F. Ordonez, "Robust optimization models for energy-limited wireless sensor networks under distance uncertainty," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 2161–2169, june 2008.

[21] K. Yang, Y. Wu, J. Huang, X. Wang, and S. Verdu, “Distributed robust optimization for communication networks,” in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, April 2008.

[22] T. Liu, W. Yang, J. Liao, and J. Huang, “Robust optimization for designing reliable telecommunication networks with node failure scenarios,” in *Proceedings of IEEE International Conference on Emergency Management and Management Sciences (ICEMMS)*, August 2010.

[23] L. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in *Proceedings of Joint Conference of the IEEE Computer and Communications Societies*, 2001.

[24] D. Zhu and M. Mutka, “Cooperation among peers in an ad hoc network to support an energy efficient im service,” in *Journal of Pervasive and Mobile Computing*, vol. 4, pp. 335–359, June 2008.

[25] H.-C. Liu, J. Min, and H. Samueli, “A low-power baseband receiver ic for frequency-hopped spread spectrum communications,” *IEEE Journal on Solid-State Circuits*, vol. 31, pp. 384–394, March 1996.

[26] S. Cui, A. Goldsmith, and A. Bahai, “Energy-constrained modulation optimization,” *IEEE Transactions on Wireless Communications*, vol. 4, pp. 2349–2360, September 2005.

[27] G. Miao, N. Himayat, Y. Li, and A. Swami, “Cross-layer optimization for energy-efficient wireless communications: A survey,” *Journal on Wireless Communications and Mobile Computing*, vol. 9, pp. 529–542, April 2009.

[28] J. Wu and H. Li, “On calculating connected dominating set for efficient routing in ad hoc wireless networks,” in *Proceedings of ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999.

[29] Y. P. Chen and A. L. Liestman, “Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks.,” in *Proceedings of MobiHoc’02.*

[30] J. Yu and P. Chong, “3hbac (3-hop between adjacent clusterheads): a novel non-overlapping clustering algorithm for mobile ad hoc networks,” in *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, August 2003.

[31] T. J. Kwon, M. Gerla, V. Varma, M. Barton, and T. Hsing, “Efficient flooding with passive clustering-an overhead-free selective forward mechanism for

ad hoc/sensor networks," *Proceedings of the IEEE*, vol. 91, pp. 1210–1220, August 2003.

[32] P. Basu, N. Khan, and T. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proceedings of International Conference on Distributed Computing Systems*, April 2001.

[33] A. McDonald and T. Znati, "Design and performance of a distributed dynamic clustering algorithm for ad-hoc networks," in *Proceedings of Simulation Symposium*, 2001.

[34] J. Wu, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," in *Proceedings of International Conference on Parallel Processing*, September 2001.

[35] J.-H. Ryu, S. Song, and D.-H. Cho, "New clustering schemes for energy conservation in two-tiered mobile ad-hoc networks," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2001.

[36] T. Ohta, S. Inoue, and Y. Kakuda, "An adaptive multihop clustering scheme for highly mobile ad hoc networks," in *Proceedings of International Symposium on Autonomous Decentralized Systems*, April 2003.

[37] M. Chatterjee, S. Sas, and D. Turgut, "An on-demand weighted clustering algorithm (wca) for ad hoc networks," in *Proceedings of IEEE Global Telecommunications Conference*, 2000.

[38] C.-H. Lung and C. Zhou, "Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach," in *Ad Hoc Networks*, vol. 8, pp. 328–344, 2010.

[39] Y. Zhou, M. Hart, S. Vadgama, and A. Rouz, "A hierarchical clustering method in wireless ad hoc sensor networks," in *Proceedings of IEEE International Conference on Communications(ICC)*, June 2007.

[40] "Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE 802.11-2007 Standard*, 2007.

[41] "Wi-fi peer-to-peer (p2p) technical specification v1.0," *Wi-Fi Alliance, P2P Technical Group*, December 2009.

[42] "Wi-fi protected setup specification v1.0h," *Wi-Fi Alliance*, December 2006.

[43] A. Goldsmith, *Wireless Communications.* New York, NY, USA: Cambridge Univ. Press, 2005.

[44] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” in *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, February 1956.

[45] R. C. Prim, “Shortest connection networks and some generalizations,” in *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[46] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, ch. The algorithms of Kruskal and Prim. MIT Press and McGraw-Hill, 2001.

[47] K. M. et al., *Multivariate Analysis*. Academic Press, 1979.

[48] V. T. P. Nicolas Boria, “Fast reoptimization for the minimum spanning tree problem,” in *Laboratoire d’Analyse et Modlisation de Systmes pour l’Aide la Dcision*, Novembre 2008.

[49] H. M. M. H. K. Mahmud, M. Inoue and H. Morikawa, “Energy consumption measurement of wireless interfaces in multi-service user terminals for heterogeneous wireless networks,” in *IEICE Trans. Commun.*, vol. E88-B, p. 10971110, March 2005.

[50] R. Doe, “Hp ipaq pocket pc h6300 series quickspecs,” 2005.

[51] R. C. Radhika Mittal, Aman Kansal, “Empowering developers to estimate app energy consumption,” in *MobiCom 2012, Istanbul, Turkey*, August 2012.

[52] A. V. Niranjan Balasubramanian, Aruna Balasubramanian, “Energy consumption in mobile phones: A measurement study and implications for network applications,” in *IMC 2009, Chicago, Illinois, USA*, November 2009.

[53] “The Network Simulator NS-2.” http://www.isi.edu/nsnam/ns/.

[54] “The Network Simulator OMNeT++.” http://www.omnetpp.org/.