ENERGY EFFICIENT COOPERATIVE VIDEO DISTRIBUTION

Enrollment Number: 101300

Name: Aseem Kukkar

Project Guide: Dr. Nitin

May 2014

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

Table of Content

Chapter – 1

Wireless Sensor Networks

Chapter - 2

Literature Survey

Chapter - 3

Testbed, Experimental Setup and Simulation Outputs

Chapter - 4

Future Directions

List of Figures

Certificate

This is to certify that the work titled **— "ENERGY EFFICENT COOPERATIVE VIDEO DISTRIBUTION**", submitted by **Aseem kukkar** partial fulfillment for the award of degree of Bachelor of Technology in Computer Science Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor

Name of Supervisor Dr. Nitin

Designation Associate Professor

Date

Acknowledgement

As we conclude our project with the God's grace, we have many people to thank; for all the help, guidance and support they lent us, throughout the course of our endeavor.

First and foremost, we are sincerely thankful to Dr.Nitin, our Project Guide, who has always encouraged us to put in our best efforts and deliver a quality and professional output. His methodology of making the system strong from inside has taught us that output is not the END of project. We really thank him for his time & efforts.

We are deeply indebted to all those who provided reviews and suggestions for improving the materials and topics covered in our project, and we extend our apologies to anyone we may have failed to mention.

Enrollment number: 101300
Name: Aseem Kukkar

Signature

ABSTRACT

For real-time video broadcast where multiple users are interested in the same content, mobile-to-mobile cooperation can be utilized to improve delivery efficiency and reduce network utilization. Under such cooperation, however, real-time video transmission requires end-to-end delay bounds. Due to the inherently stochastic nature of wireless fading channels, deterministic delay bounds are prohibitively difficult to guarantee. For a scalable video structure, an alternative is to provide statistical guarantees using the concept of effective capacity/bandwidth by deriving quality of service exponents for each video layer. Using this concept, we formulate the resource allocation problem for general multihop multicast network flows and derive the optimal solution that minimizes the total energy consumption while guaranteeing a statistical end-to-end delay bound on each network path. A method is described to compute the optimal resource allocation at each node in a distributed fashion. Furthermore, we propose low complexity approximation algorithms for energy-efficient flow selection from the set of directed acyclic graphs forming the candidate network flows. The flow selection and resource allocation process is adapted for each video frame according to the channel conditions on the network links. Considering different network topologies, results demonstrate that the proposed resource allocation and flow selection algorithms provide notable performance gains with small optimality gaps at a low computational cost.

CHAPTER 1

WIRELESS SENSOR NETWORKS

1.1 INTRODUCTION

Sensors integrated into structures, machinery, and the environment, coupled with the efficient delivery of sensed information, could provide tremendous benefits to society. Potential benefits include: fewer catastrophic failures, conservation of natural resources, improved manufacturing productivity, improved emergency response, and enhanced homeland security. However, barriers to the widespread use of sensors in structures and machines remain. Bundles of lead wires and fiber optic “tails” are subject to breakage and connector failures. Long wire bundles represent a significant installation and long term maintenance cost, limiting the number of sensors that may be deployed, and therefore reducing the overall quality of the data reported. Wireless sensing networks can eliminate these costs, easing installation and eliminating connectors.

The ideal wireless sensor is networked and scaleable, consumes very little power, is smart and software programmable, capable of fast data acquisition, reliable and accurate over the long term, costs little to purchase and install, and requires no real maintenance.

Selecting the optimum sensors and wireless communications link requires knowledge of the application and problem definition. Battery life, sensor update rates, and size are all major design considerations. Examples of low data rate sensors include temperature, humidity, and peak strain captured passively. Examples of high data rate sensors include strain, acceleration, and vibration.

Recent advances have resulted in the ability to integrate sensors, radio communications, and digital electronics into a single integrated circuit (IC) package. This capability is enabling networks of very low cost sensors that are able to communicate with each other using low power wireless data routing protocols. A wireless sensor network (WSN) generally consists of a base station (or "gateway") that can communicate with a number of wireless sensors via a radio link. Data is collected at the wireless sensor node, compressed, and transmitted to the gateway directly or, if required, uses other wireless sensor nodes to forward data to the gateway. The transmitted data is then presented to the system by the gateway connection. The purpose of this chapter is to provide a brief technical introduction to wireless sensor networks and present a few applications in which wireless sensor networks are enabling.

1.2 Individual Wireless Sensor Node Architecture

A functional block diagram of a versatile wireless sensing node is provided in Figure 1.1. A modular design approach provides a flexible and versatile platform to address the needs of a wide variety of applications. For example, depending on the sensors to be deployed, the signal-conditioning block can be re-programmed or replaced. This allows for a wide variety of different sensors to be used with the wireless sensing node. Similarly, the radio link may be swapped out as required for a given applications' wireless range requirement and the need for bidirectional communications. The use of flash memory allows the remote nodes to acquire data on command from a base station, or by an event sensed by one or more inputs to the node. Furthermore, the embedded firmware can be upgraded through the wireless network in the field.

The microprocessor has a number of functions including:

1) Managing data collection from the sensors

2) Performing power management functions

3) Interfacing the sensor data to the physical radio layer

4) Managing the radio network protocol

A key feature of any wireless sensing node is to minimize the power consumed by the system. Generally, the radio subsystem requires the largest amount of power. Therefore, it is advantageous to send data over the radio network only when required. This sensor event-driven data collection model requires an algorithm to be loaded into the node to determine when to send data based on the sensed event. Additionally, it is important to minimize the power consumed by the sensor itself. Therefore, the hardware should be designed to allow the microprocessor to judiciously control power to the radio, sensor, and sensor signal conditioner.

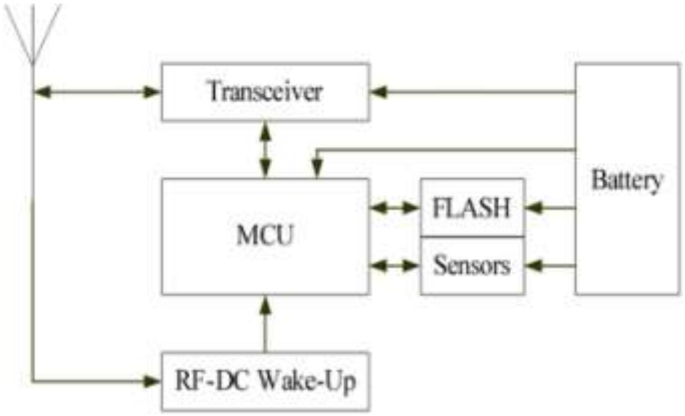


Figure 1.1: Wireless sensor node functional block diagram

1.3 Wireless Sensor Networks Architecture

There are a number of different topologies for radio communications networks. A brief discussion of the network topologies that apply to wireless sensor networks are outlined below.

1.3.1 Star Network (Single Point-to-Multipoint)

A star network (Figure 22.3.1) is a communications topology where a single base station can send and/or receive a message to a number of remote nodes. The remote nodes can only send or receive a message from the single base station, they are not permitted to send messages to each other. The advantage of this type of network for wireless sensor networks is in its simplicity and the ability to keep the remote node’s power consumption to a minimum. It also allows for low latency communications between the remote node and the basestation. The disadvantage of such a network is that the basestation must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network.

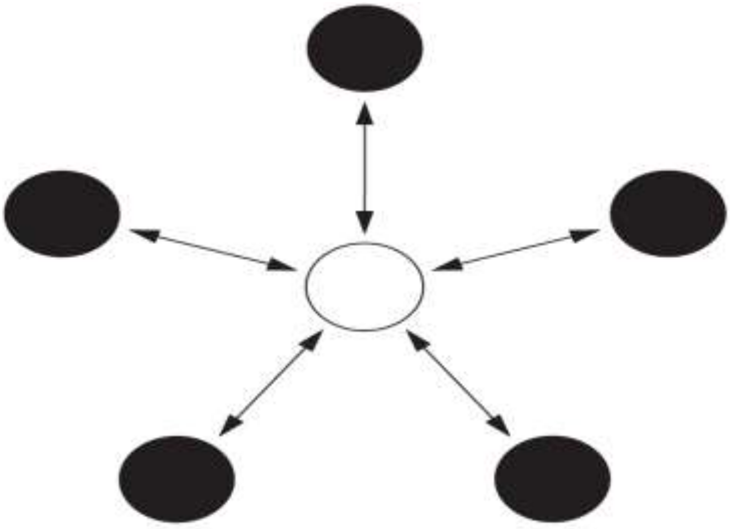


Figure 1.2: Star network topology

1.3.2 Mesh Network

A mesh network allows for any node in the network to transmit to any other node in the network that is within its radio transmission range. This allows for what is known as multihop communications; that is, if a node wants to send a message to another node that is out of radio communications range, it can use an intermediate node to forward the message to the desired node. This network topology has the advantage of redundancy and scalability. If an individual node fails, a remote node still can communicate to any other node in its range, which in turn, can forward the message to the desired location. In addition, the range of the network is not necessarily limited by the range in between single nodes, it can simply be extended by adding more nodes to the system. The disadvantage of this type of network is in power consumption for the nodes that implement the multihop communications are generally higher than for the nodes that don’t have this capability, often limiting the battery life. Additionally, as the number of communication hops to a destination increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

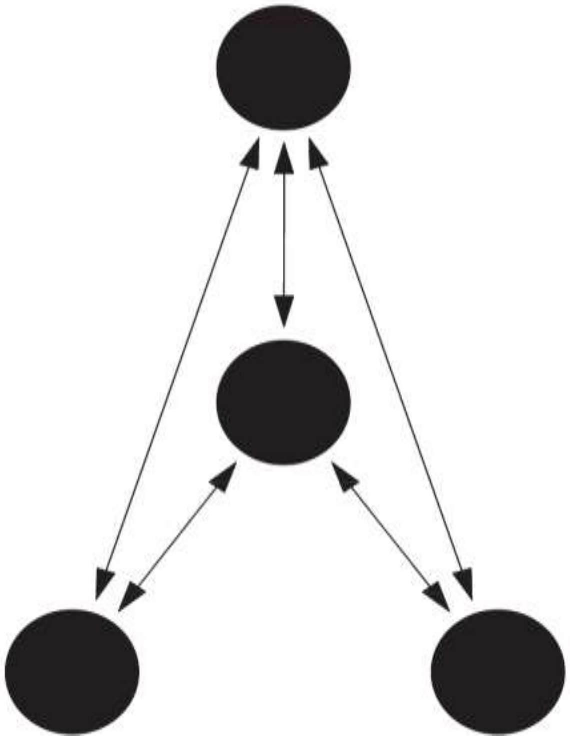


Figure 1.3: Mesh network topology

1.3.3 Hybrid Star – Mesh Network

A hybrid between the star and mesh network provides for a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum. In this network topology, the lowest power sensor nodes are not enabled with the ability to forward messages. This allows for minimal power consumption to be maintained. However, other nodes on the network are enabled with multihop capability, allowing them to forward messages from the low power nodes to other nodes on the network. Generally, the nodes with the multihop capability are higher power, and if possible, are often plugged into the electrical mains line. This is the topology implemented by the up and coming mesh networking standard known as ZigBee.

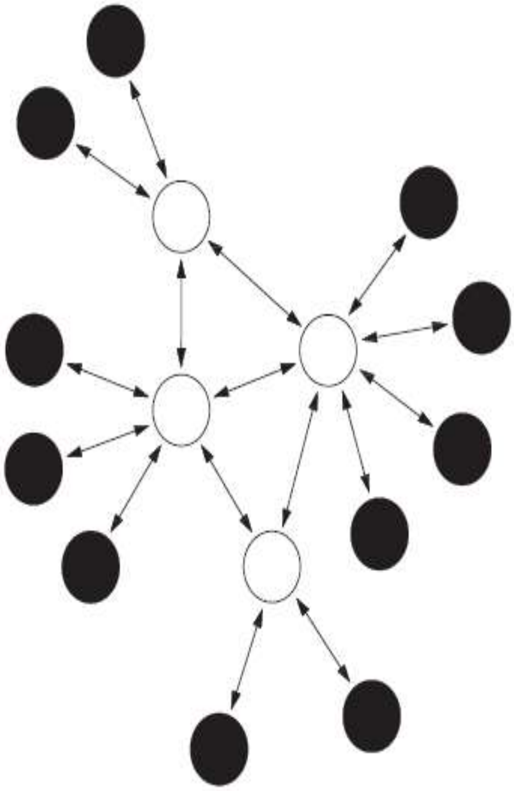


Figure 1.4: Hybrid star-mesh network topology

1.4 Challenges

In spite of the diverse applications, sensor networks pose a number of unique technical challenges due to the following factors:

>> **Ad hoc deployment:** Most sensor nodes are deployed in regions which have no infrastructure at all. A typical way of deployment in a forest would be tossing the sensor nodes from an aero plane. In such a situation, it is up to the nodes to identify its connectivity and distribution.

>>**Unattended operation:** In most cases, once deployed, sensor networks have no human intervention. Hence the nodes themselves are responsible for reconfiguration in case of any changes.

>>**Untethered:** The sensor nodes are not connected to any energy source. There is only a finite source of energy, which must be optimally used for processing and communication. An interesting fact is that communication dominates processing in energy consumption. Thus, in order to make optimal use of energy, communication should be minimized as much as possible.

>> **Dynamic changes:** It is required that a sensor network system be adaptable to changing connectivity (for e.g., due to addition of more nodes, failure of nodes etc.) as well as changing environmental stimuli. Thus, unlike traditional networks, where the focus is on maximizing channel throughput or minimizing node deployment, the major consideration in a sensor network is to extend the system lifetime as well as the system robustness.

1.5 Applications of Wireless Sensor Networks

Structural Health Monitoring – Smart Structures

Sensors embedded into machines and structures enable condition-based maintenance of these assets. Typically, structures or machines are inspected at regular time intervals, and components may be repaired or replaced based on their hours in service, rather than on their working conditions. This method is expensive if the components are in good working order, and in some cases, scheduled maintenance will not protect the asset if it was damaged in between the inspection intervals. Wireless sensing will allow assets to be inspected when the sensors indicate that there may be a problem, reducing the cost of maintenance and preventing catastrophic failure in the event that damage is detected.

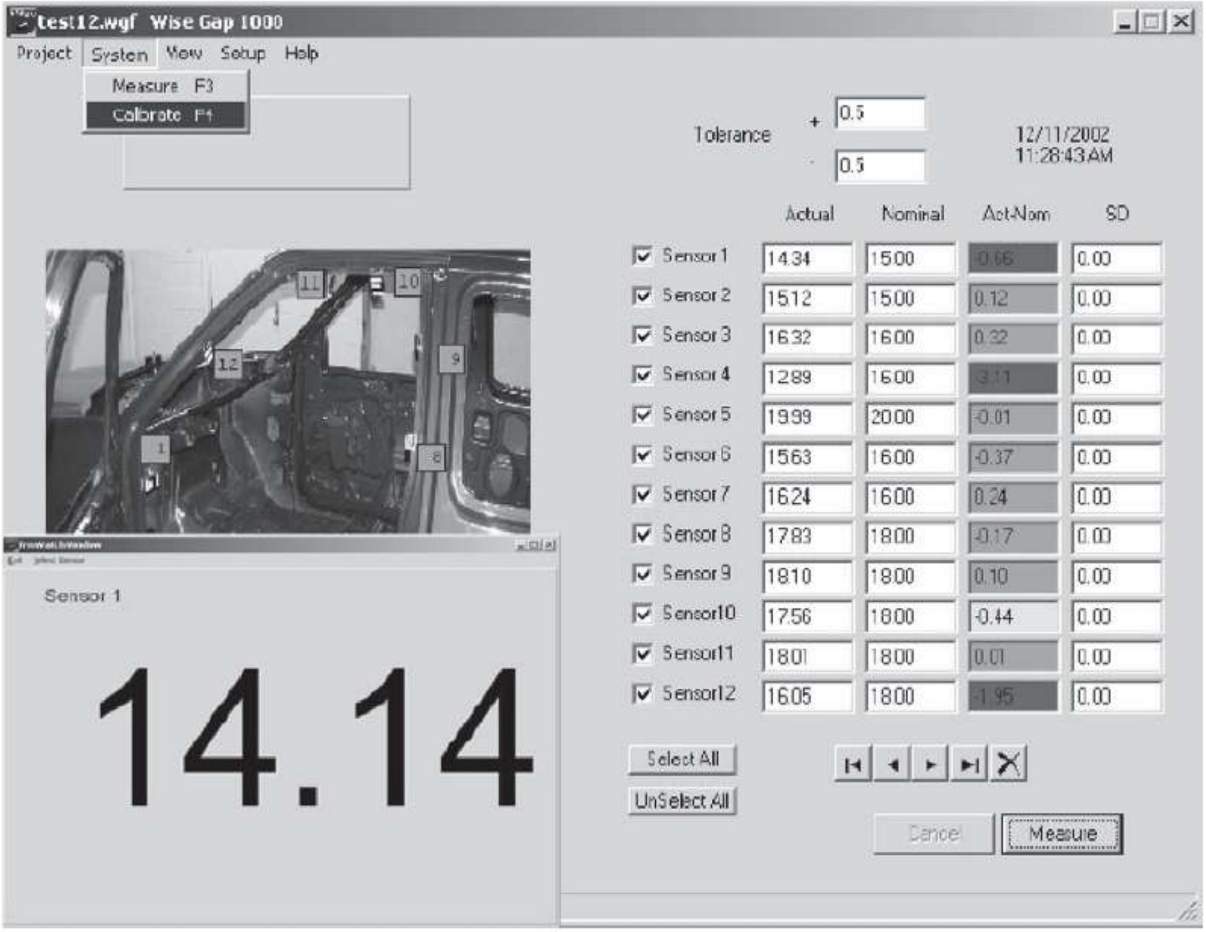
Additionally, the use of wireless reduces the initial deployment costs, as the cost of installing long cable runs is often prohibitive.

In some cases, wireless sensing applications demand the elimination of not only lead wires, but the elimination of batteries as well, due to the inherent nature of the machine, structure, or materials under

test. These applications include sensors mounted on continuously rotating parts , within concrete and composite materials, and within medical implants.

Industrial Automation

In addition to being expensive, lead wires can be constraining, especially when moving parts are involved. The use of wireless sensors allows for rapid installation of sensing equipment and allows access to locations that would not be practical if cables were attached. An example of such an application on a production line is shown in Figure 1.6. In this application, typically ten or more sensors are used to measure gaps where rubber seals are to be placed. Previously, the use of wired sensors was too cumbersome to be implemented in a production line environment. The use of wireless sensors in this application is enabling, allowing a measurement to be made that was not previously practical. Other applications include energy control systems, security, wind turbine health monitoring, environmental monitoring, location-based services for logistics, and health care.



Sensor 1

14.14

Tolerance + 0.5

Tolerance - 0.5

12/11/2002
11:28:43 AM

	Actual	Nominal	Act-Nom	SD
☑ Sensor 1	14.34	15.00	-0.66	0.00
☑ Sensor 2	15.12	15.00	0.12	0.00
☑ Sensor 3	16.32	16.00	0.32	0.00
☑ Sensor 4	12.89	16.00	-3.11	0.00
☑ Sensor 5	19.99	20.00	-0.01	0.00
☑ Sensor 6	15.63	16.00	-0.37	0.00
☑ Sensor 7	16.24	16.00	0.24	0.00
☑ Sensor 8	17.83	18.00	-0.17	0.00
☑ Sensor 9	18.10	18.00	0.10	0.00
☑ Sensor10	17.56	18.00	-0.44	0.00
☑ Sensor11	18.01	18.00	0.01	0.00
☑ Sensor12	16.05	18.00	-1.95	0.00

Figure 1.6: Industrial application of wireless sensors

Application Highlight – Civil Structure Monitoring

One of the most recent applications of today’s smarter, energy-aware sensor networks is structural health monitoring of large civil structures, such as the Ben Franklin Bridge (Figure 1.7), which spans the Delaware River, linking Philadelphia and Camden, N.J. The bridge carries automobile, train and pedestrian traffic. Bridge officials wanted to monitor the strains on the structure as high-speed commuter trains crossed over the bridge.

Figure 1.7: Ben Franklin Bridge

A star network of ten strain sensors were deployed on the tracks of the commuter rail train. The wireless sensing nodes were packaged in environmentally sealed NEMA rated enclosures. The strain gauges were also suitably sealed from the environment and were spot welded to the surface of the bridge steel support structure. Transmission range of the sensors on this star network was approximately 100 meters.

The sensors operate in a low-power sampling mode where they check for presence of a train by sampling the strain sensors at a low sampling rate of approximately 6 Hz. When a train is present the strain increases on the rail, which is detected by the sensors. Once detected, the system starts sampling at a much higher sample rate. The strain waveform is logged into local Flash memory on the wireless sensor nodes. Periodically, the waveforms are downloaded from the wireless sensors to the base station. The base station has a cell phone attached to it which allows for the collected data to be transferred via the cell network to the engineers’ office for data analysis. This low-power event-driven data collection method reduces the power required for continuous operation from 30 mA if the sensors were on all the time to less than 1 mA continuous. This enables a lithium battery to provide more than a year of continuous operation. Resolution of the collected strain data was typically less then 1 micro strain. A typical waveform downloaded from the node is shown in Figure 1.8. Other performance specifications for these wireless strain sensing nodes have been provided in an earlier work .

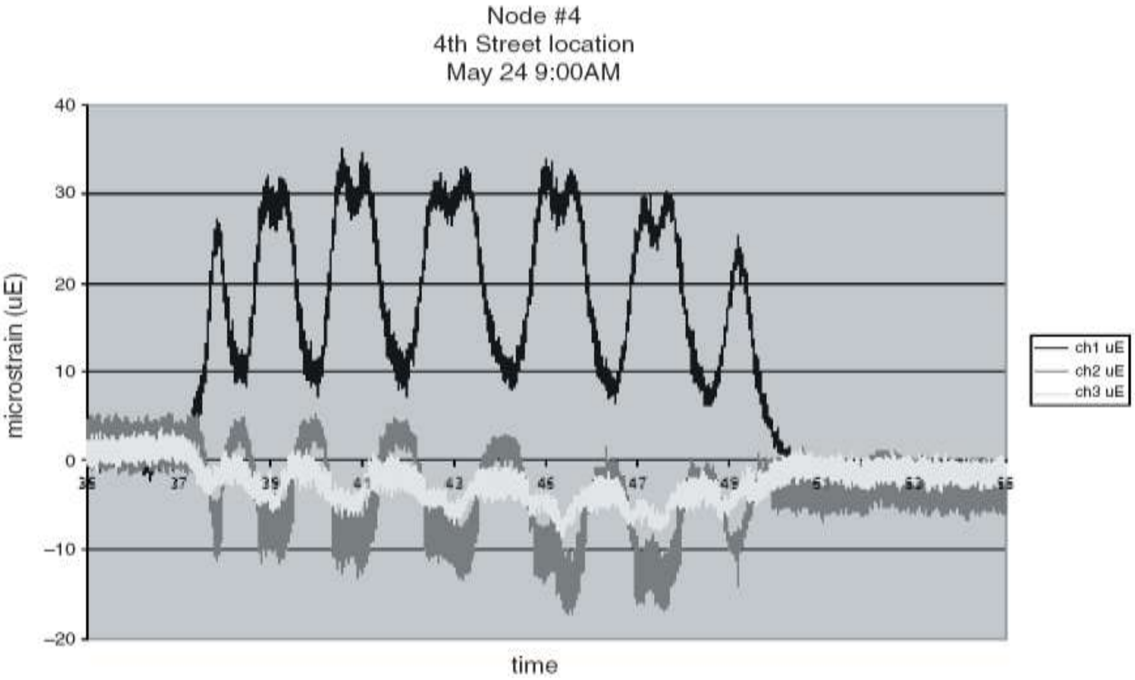


Figure 1.8: Bridge strain data

1.6 Future Developments

The most general and versatile deployments of wireless sensing networks demand that batteries be deployed. Future work is being performed on systems that exploit piezoelectric materials to harvest ambient strain energy for energy storage in capacitors and/or rechargeable batteries. By combining smart, energy saving electronics with advanced thin film battery chemistries that permit infinite recharge cycles, these systems could provide a long term, maintenance free, wireless monitoring solution .

1.7 Summary

Wireless sensor networks are enabling applications that previously were not practical. As new standards- based networks are released and low power systems are continually developed, we will start to see the widespread deployment of wireless sensor networks.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

The real-time nature of video broadcast demands quality-of-service (QoS) guarantees such as delay bounds for end-user satisfaction. Given the bit rate requirements of such services, delivery efficiency is another key objective. The H.264/AVC video coding standard [1] is designed to provide high coding efficiency that is suitable for wireless video transmission. To provide a network-friendly design, the scalable video coding (SVC) extension of H.264 [2] allows rate scalability at the bitstream level by generating embedded bit streams that are partially decodable at different bitrates with degrading quality. The basic level of quality is supported by the base layer and incremental improvements are provided by the enhancement layers. Video source rate scalability can be achieved with temporal, spatial, or quality scalability [2].

Deterministic delay bounds are prohibitively expensive to guarantee over wireless networks. Consequently, to provide a realistic and accurate model for quality of service, statistical guarantees are considered as a design guideline by defining constraints in terms of the delay-bound violation probability. The notion of statistical QoS is tied back to the well-developed theory of effective bandwidth [3], [4], [5] and its dual concept of effective capacity [6], [7], [8]. For scalable video transmission, a set of QoS exponents for each video layer are obtained by applying the effective bandwidth/capacity analyses on the incoming video stream to characterize the delay requirement [9], [10]. The problem of providing statistical delay bounds for layered video transmission over single hop unicast and multicast links was considered in [9]. For general multihop multicast network scenarios, it is inefficient to allocate resources independently among network links since the variation in the supported service rates among different links affects the end-to-end transport capability in the network.

Cooperation among mobile devices in wireless networks has the potential to provide notable performance gains in terms of increasing the network throughput [11], [12], [13], [14], [15], extending the network coverage [16], [17], [18], decreasing the end-user communication cost [19], and decreasing the energy consumption [20], [21], [22]. For example, the ICAM architecture presents an integrated cellular and ad hoc multicast scheme to increase the cellular multicast throughput through the use of mobile stations (MSs) as ad hoc relays [13]. In the UCAN architecture [12], the MSs use their WLAN interface to enhance the throughput and increase the coverage of a wireless wide area network. In [19], MSs are assumed to be connected to several wireless networks with different characteristics in terms of bandwidth, packet loss probability, and transmission cost. A near optimal solution is shown to reduce end-user cost while meeting distortion and delay constraints.

The advantages of cooperation among mobile devices in wireless networks have been also revealed for video streaming applications [23], [24], [25], [26], [27]. For example, the CHUM architecture assumes that all mobile devices are interested in the same video content that is divided into multiple descriptions [23]; each mobile device randomly selects and pulls a video description through a cellular link and multicasts it to all members in its cooperation group which is formed in an ad hoc manner. In [27], the authors propose distributed video scheduling schemes for multi-radio multihop wireless networks to minimize video distortion and ensure distortion-fairness sharing among multiple description video streams. The distortion model is constructed to provide a balance between the selfish motivation of minimizing video distortion and the global performance of minimizing network congestion.

Minimizing energy consumption in battery-operated mobile devices is essential for the development of next generation heterogeneous wireless communications systems. Enhancement schemes and communication architectures with cooperation among mobile devices to reduce energy consumption appear extensively in the literature, e.g., see [20], [22], [28], [29], [30]. In [22],a cooperative network architecture is presented and experimentally evaluated to reduce energy consumption in multiradio mobile devices for video streaming applications. In [30], a comprehensive experimental study is conducted where results presented demonstrate notable energy reduction gains by collaborative downloading. The problem of resource allocation with statistical QoS guarantees and optimized energy consumption over cooperative networks with general topologies has not been tackled yet in the literature. While the work in [27] addresses optimized rate allocation and routing over cooperative wireless networks, it is fundamentally different from our work since we consider minimizing energy consumption in mobile terminals as the central objective, providing statistical delay guarantees, and capturing layered video content with QoS requirements per layer.

In this work, we develop optimized flow selection and resource allocation schemes that can provide end-to-end statistical delay bounds and minimize energy consumption for video distribution over cooperative wireless networks. The network flow for video content distribution can be any sequential multihop multicast tree forming a directed acyclic graph that spans the network topology. We model the queuing behaviour of the cooperative network according to the effective capacity link layer model. Based on this model, we formulate and solve the flow resource allocation problem to minimize the total energy consumption subject to end-to-end delay bounds on each network path. More-over, we propose two approximation algorithms to solve the flow selection problem which involves selecting the optimal flow in terms of minimizing energy consumption. The first algorithm uses negated signal-to-noise ratios (SNR) as link weights on the complete network graph, finds the minimum spanning tree using those weights to maximize the sum rate, and performs optimal resource allocation on the flow corresponding to the obtained tree structure. The second algorithm maintains a set of dominant flows that are optimal for a potentially large percentage of channel states under a certain network topology and performs flow selection on that dominant set. By updating the optimal allocation and flow selection iteratively in each time frame according to the instantaneous channel states, the Algorithms effectively reselect the best network flow and reconfigure the service process on each link to provide optimized end-to-end transport.

The cooperative network model is presented in Section 2. The procedure for providing end-to-end delay bounds for each path in the network is described in Section 3. In Section 4, the problem of cooperative video distribution is formulated as a convex optimization problem and the optimal solution is derived. In Section 5, two approximation algorithms for flow selection are proposed and discussed in terms of complexity. In Section 6, results of the proposed resource allocation and flow selection schemes are pre-sented and analyzed for different network topologies. Finally, conclusions are drawn in Section 7.

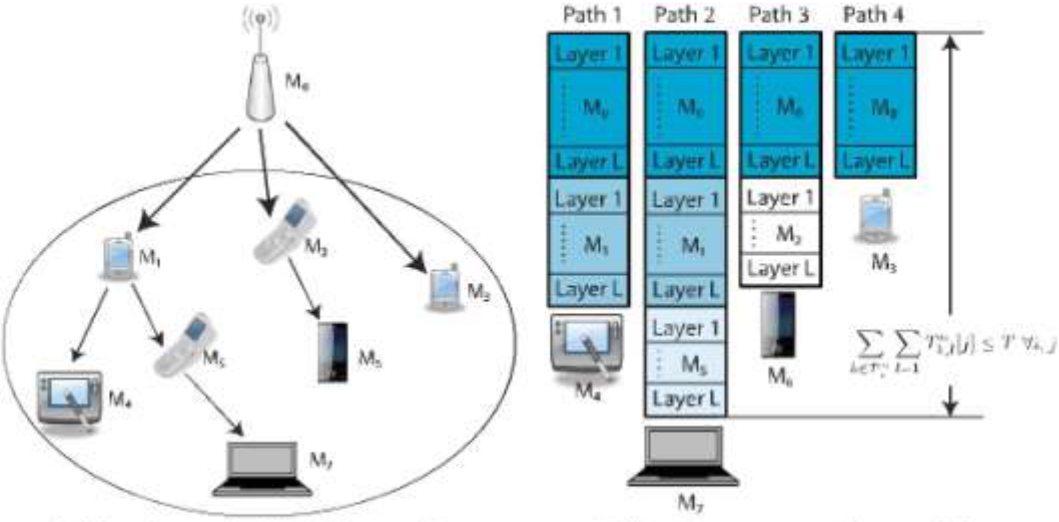
2.2 COOPERATIVE NETWORK MODEL

The proposed system model consists of a base station (BS), denoted by M_0, and K MSs M_1; . . . ; M_K which are capable of transmitting, receiving, or relaying a scalable video bitstream. The BS is responsible for distributing the same multilayer video stream to the MSs over wireless fading channels. We define a flow as a tree of adjacent links that represents consecutive unicast/multicast transmissions. We are given a set of N candidate flows where the nth flow is defined by a set of links F_n which form a directed acyclic tree (DAG).

Fig. 1 a shows an example network with seven MSs and a fixed network flow used to explain the system model. This network flow consists of four distinct paths leading to M_4, M_7, M_6, and M_3 and traversing all MSs. We define P^n_i as the set of nodes traversed by the ith path of the nth flow. For the first path in the given example network, $P^n_1=\{M_0,M_1,M_4\}$. We refer to p_n as the number of paths in flow F_n. Thus, $p_n=4$ for the fixed network flow n. The set of unicast/ multicast receivers for MS M_k in the nth flow is denoted M^n_k. For example, the set of multicast receivers for the BS transmission is $M^n_0=\{M_1; M_2; M_3\}$ and $|M^n_0j|=3$. Note that $|M^n_k| =1$ characterizes a unicast transmission by M_k.

The video stream generated by the scalable video codec consists of L video layers. Each layer maintains a separate queue at each node and has specific QoS requirements according to its relevance in the decoding process.

The time frame T is defined as the difference between the playback time of two video frames at the receiver, i.e., the reciprocal of the video frame rate. Within this duration T , the video frame contents corresponding to the L layers should be transmitted as per the construction of flow F_n to all K receivers to avoid playback buffer starvation. Fig. 1b shows the time frame structure corresponding to the fixed network flow in Fig. 1a for explanation purposes. We treat each path of the multicast tree separately by allowing the content to be streamed simultaneously (in parallel) on different paths of the network flow. This is based on the assumption that channels are readily available for all MSs in the network. Note that the number of channels required is upper bounded by the number of paths in the network. For example, the 4-path network flow in Fig. 1 requires only two channels to support the simultaneous transmission by M_1 and M_2. The number of paths is typically significantly



(a) Example network with a fixed network flow

(b) Time frame structure of the network flow

Fig. 1. System model.

Lower than the network size K, and in realistic scenarios, the network size is much less than the number of available channels in the wireless technology utilized for the short range transmissions. For instance, Bluetooth uses frequency hopping for multiple accesses with a carrier spacing of 1 MHz and a total bandwidth of 80 MHz [31].

Let $T_{k,l}^n[j]$ denote the time slot duration allocated by node k to layer l in the jth frame if flow Fn is used. We need to guarantee that $\sum_{k\in\mathcal{P}_i^n}\sum_{l=1}^{L} T_{k,l}^n[j] \le T\ \forall$ path iVframe j where k=0 refers to the BS transmission and k > 0 refers to MS Mk’s transmission. The time proportion $t_{k,l}^n[j] = T_{k,l}^n[j]/T$ represents the normalized resource allocation, thus, the resource allocation bound reduces to

$$\frac{1}{T}\sum_{k\in\mathcal{P}_i^n}\sum_{l=1}^{L}T_{k,l}^{n}[j]=\sum_{k\in\mathcal{P}_i^n}\sum_{l=1}^{L}t_{k,l}^{n}[j]\le 1\ \forall i,j \qquad (1)$$

In general, different links may be employing different technologies, and thus having different bandwidths. The signal bandwidth for the transmission of node k is denoted Bk Hz. The wireless channels are assumed to be flat fading. Then, the instantaneous signal to noise ratios are used to characterize the channel state information (CSI) under the assumption that the SNR is perfectly estimated by all MSs and reliably fed back to the BS without delay. The SNR vector $\gamma[j]\triangleq\{\gamma_{k,k'}[j]\}_{k=0,k'=1}^{K\ \ K}$ represents a fading state of the network for frame j where $\gamma_{k,k'}[j]$ is the instantaneous SNR on the link between nodes Mk and Mk’ . Moreover, ϒis modeled as an ergodic and stationary block-fading process invariant within the duration of the time frame and uncorrelated among consecutive frames.

We consider a network with no loss tolerance, that is, multicast is limited by the supported rate on the link with the lowest SNR [9]. For this purpose, we define the effective SNR for the transmission of node k as follows:

$$\gamma_k[j]=\min_{k'\in\mathcal{M}_k^n}\gamma_{k,k'}[j] \qquad (2)$$

Assuming capacity-achieving codes are used to provide the capability of operating at the Shannon capacity, the transmis-sion rate of node k $R_k[j]=B_k\log(1+\gamma_k[j])$. The Service process $C_{k;l}[j]$ (bits/frame) of node k for the lth video layer and jth frame can be expressed as follows:

$$C^n_{k,l}[j] = T^n_{k,l}[j] R_k[j] = B_k T t^n_{k,l}[j] \log_2(1 + \gamma_k[j]) \quad (3)$$

In the sequel, we drop the frame index j for convenience.

2.3 STATISTICAL END-TO-END DELAY BOUNDS FOR GENERAL NETWORK FLOWS

In this section, we describe the procedure for providing end-to-end delay guarantees by characterizing link-level QoS metrics according to the effective capacity link layer model.

2.3.1 Queuing Network Model for Multihop Layered Video Transmission

A separate queue is maintained for each video layer at each node. The arrival process at the BS is denoted $\{A_{0,l}\}$ and is determined by the scalable codec parameters and the video content. Given this arrival process and the instantaneous SNR $\{\gamma_{k,k'}\}_{k=0,k'=1}^{K\quad K}$, we are interested in adaptively configuring the service process such that the following condition is satisfied.

$$\Pr\left\{\sum_{k\in\mathcal{P}^n_i} D^n_{k,l} > D_{\text{th}}\right\} \le P_{\text{th}}\ \forall l, i, \quad (4)$$

where $D^n_{k;l}$ is the queuing delay at node k for layer l if flow F_n is used, D_{th} is the end-to-end statistical delay-bound, P_{th} is the target delay-bound violation probability, P^n_i is the set of nodes along the ith path of flow F_n.

The behavior of the queue-length process in queuing-based communication networks is extensively treated in [32]. In an acyclic network, the queue length at time t of each queue can be bounded exponentially as t→∞

$$\Pr\{Q_{k,l}(t) > Q_{\text{th}}\} \doteq e^{-\theta_l Q_{\text{th}}}; \qquad 0 \le k \le K, 0 \le l \le L,$$

(5)

where Qk,l is the queuing delay at MS Mk for video layer l and Qth is the queue-length threshold. The parameter Ɵl ,termed the QoS exponent, is used to characterize delay. More stringent QoS requirements are characterized by larger Ɵl while looser QoS requirements require smaller Ɵl

2.3.2 Effective Bandwidth/Capacity Model

The effective capacity channel model captures a generalized link-level capacity notion of the fading channel by characterizing wireless channels in terms of functions that can be easily mapped to link-level QoS metrics, such as delay-bound violation probability. Thus, it is a convenient tool for designing QoS provisioning mechanisms [6], [9].

We denote by $C^{n}_{k;l}$ (ϴl) and $A^{n}_{k;l}$(ϴl) the effective capacity and effective bandwidth functions, respectively, for the kth link and lth layer when flow F_n is used. Given an arrival process its effective bandwidth, denoted by An k;l(ϴl) (bits/frame), is defined as the minimum constant service rate required to guarantee a specified QoS exponent ϴl. In contrast, for a given service process $\{C_{k;l}^{n}\}$, its effective capacity, denoted by $C^{n}_{k;l}$(ϴl) (bits/frame), is defined as the maximum constant arrival rate which can be supported by $\{C_{k;l}^{n}\}$ subject to the specified QoS exponent $_l$. Moreover, for a stationary and ergodic service process $\{C_{k;l}^{n}\}$ that is uncorrelated across time frames, the effective capacity can be expressed as follows [33]:

$$C^{n}_{k,l}(\theta_l) = -\frac{1}{\theta_l}\log\left(\mathbb{E}\left\{e^{-\theta_l C^{n}_{k,l}}\right\}\right)$$

$$= -\frac{1}{\theta_l}\log\left(\mathbb{E}\left\{e^{-\theta_l B_k T(t^{n}_{k,l}\log(1+\gamma_k))}\right\}\right)$$

(6)&(7)

The target QoS exponent ϴ$_l$ for a given layer l should be the same for all transmissions to guarantee the same level of quality over the entire path. To provide the QoS guarantee ϴ$_l$ for the lth video layer, the effective capacity on the kth link should be equal to the effective bandwidth [34], i.e.,

$$C^{n}_{k,l}(\theta_l) = \mathcal{A}^{n}_{k,l}(\theta_l)\ \forall l = 1,\ldots,L;\ k = 0,\ldots,K$$

(8)

Moreover, we assume that, at the link layer, the service process of the kth queue is input instantaneously as the arrival processes to all its multicast nodes k€M^{n}k ,i.e.,

$\{A^n_{k,l}\} = \{C^n_{k',l}\} \; \forall k' \in \mathcal{M}^n_k$. Thus, $\mathcal{A}^n_{k,l}(\theta_l) = \mathcal{C}^n_{k',l}(\theta_l) \; \forall l; k = 0, \dots, K; k' \in \mathcal{M}^n_k$ and each end-to-end path of the network flow can be treated separately. By induction, we deduce that the service processes should be designed to guarantee that the effective capacity is the same for the multicast/unicast transmissions by each node in the network, i.e.,

$$\mathcal{C}^n_{0,l}(\theta_l) = \mathcal{C}^n_{k,l}(\theta_l) \; \forall k = 1, \dots, K \tag{9}$$

Since all links satisfy the same QoS requirement Θl and using (7), it is sufficient to ensure that the service processes on all links are equal. We deduce that the following condition on the resource allocation t^n k,l will implicitly satisfy (9)

$$B_0\left(t^n_{0,l}\log(1+\gamma_0)\right) = B_k\left(t^n_{k,l}\log(1+\gamma_k)\right) \; \forall k = 1, \dots, K. \tag{10}$$

Intuitively, (10) can be described as an adaptive allocation strategy that guarantees that the video frame data (in bits/frame) are delivered reliably over all multihop paths. Since individual links support different rates according to the instantaneous fading state, the required time slot allocations are weighted inversely by the supported rates.

2.3.3 End-to-End Delay Bounds on Network Paths

The arrival process $\{A_{0;l}\}^{L}_{l¼1}$ at the BS has an average arrival rate $\mu_l = \mathbb{E}[A_{0,l}/T]$ & for each video layer l. The per-layer arrival rate is determined by the parameters of the scalable codec and by the video content. Given this arrival process and the instantaneous SNRs, we are interested in adaptively configuring the service processes $\{C^n_{k,l}\}^{L,K}_{l=1,k=0}$ such that the end-to-end delay bounds are satisfied for every path. To model the end-to-end delay bound for each path in the network, we consider a fluid model of traffic such that the service at each node is cut through. Considering constant rate fluid traffic, the total delay can be bounded as follows [7], [8].

$$\Pr\left\{\sum_{k\in\mathcal{P}_i^n} D_{k,l}^n > D_{\mathrm{th}}\right\} \le P_{\mathrm{th}} \doteq e^{-\theta_l C_{i,l}^n(\theta_l) D_{\mathrm{th}}}\ \forall l, i, \tag{11}$$

Where Dnk; is the queuing delay at node k for layer l if flow Fn is used land P_{th} is the delay-bound violation probability, and $C^n_{i;l}(\Theta l)$ is the effective capacity function of the equivalent channel for path P^n_i. Also, $C^n_{i;l}(\Theta l)$ can be conveniently upper bounded by the effective capacity of the worst link along
the path, i.e., $C^n_{i;l}(\Theta l)$ $\min_{k\in P^n_i} \bar{C}_{k;l}(\Theta l)$.

Thus, we can guarantee the statistical delay constraint by requiring $\mathcal{C}_{i,l}^n(\theta_l) = \mathcal{C}_{k,l}^n(\theta_l) \ge \mathcal{C}_l \forall k \in \mathcal{P}_i^n$ where C_l is selected to satisfy the delay constraint with equality, that is, $\overline{\mathcal{C}}_l = \mathcal{A}_l = \mu_l \cdot$ T for each video layer. The service processes $\{C_{k;l}^n\}$ are then designed such that the end-to-end statistical delay bound is satisfied. Since $C^n_{k;l}(\Theta l)$is a function of $_l$, for a given D_{th} and P_{th}, we use (11) to find the required QoS exponent $_l$ for each layer. For the constant-rate arrival process $\{A_l\}$, the expression is

$$\theta_l = -\frac{\log P_{\mathrm{th}}}{T\mu_l D_{\mathrm{th}}}. \tag{12}$$

2.4.1 Minimum Spanning Tree Flow Selection

To find a suitable flow without using brute force, we should deal with network variables that are independent of the flow structure so that the flow choice is done independently and prior to resource allocation. While it is tempting to construct the spanning tree using link weights that take into account the energy consumption required to transmit on the link, this is not possible because the energy consumption requirement is a function of the resource allocation strategy and the set of multicast receivers on that link, which are both specific to the choice of the network flow. The network variable that can be readily used is the instantaneous link SNR. For video frame j, given the fading state $\gamma[j] \triangleq \{\gamma_{k,k'}[j]\}_{k=0,k'=1}^{K\ \ K}$, we construct the complete network graph with edge weights $-\gamma_{k,k'}[j]$ on the link between nodes Mk and Mk0 .We then use Prim’s algorithm [37] to obtain the minimum spanning tree. The spanning tree is mapped to the corresponding directed a cyclic graph representing the network flow, and wireless resources are allocated on that flow according to the convex problem in (22) through (25) to minimize total energy consumption. The chosen flow under this strategy maximizes the sum SNR over the network links, or equivalently the sum rate because the shannon rate is a concave function of the SNR. Note that the chosen flow is not necessarily throughput optimal because the actual rate on the link between nodes Mk and Mk0 is not determined by $\gamma_{k,k'}[j]$. Instead, due to multicast, it is determined by the effective SNR on the worst link $\gamma_k = \min_{k' \in \mathcal{M}_k^n} \gamma_{k,k'}$ which is not known beforehand because it depends on the choice of n.

Using this approach, the flow selection problem is separated from the resource allocation problem. The flow selection process using Prim’s minimum spanning tree algorithm costs O(E+ Vlog(V))=O(K(K+1)+(K+1)log(K+1))= O(K^2) where E=K+1 is the number of edges in the complete graph and V=K+1 is the number of vertices. Thus, the total time complexity of the flow selection and resource allocation is O(K^2)+ O(K^2+KL)=O(K^2+KL) as opposed to O((K^2+KL).$(K+1)^{K-1}$)the optimal flow selection. Results presented in Section 6 demonstrate that this significant decrease in complexity costs a limited increase in the average energy consumption in various network scenarios.

2.4.2 Dominant Set Flow Selection

The second approximation algorithm is based on the observation that most of the network flows can only be optimal for a small percentage of the fading states

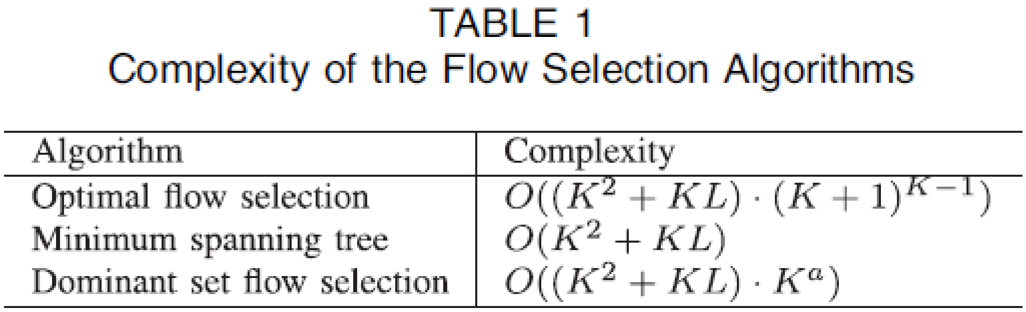


TABLE 1
Complexity of the Flow Selection Algorithms

Algorithm	Complexity
Optimal flow selection	$O((K^2 + KL) \cdot (K+1)^{K-1})$
Minimum spanning tree	$O(K^2 + KL)$
Dominant set flow selection	$O((K^2 + KL) \cdot K^a)$

corresponding to extreme instantaneous SNRs on the network links. For instance, if MS M1 lies between the BS and MS M2, it is very unlikely that the transmission to M1 through M2 is more energy efficient than the transmission toM2 throughM1.We thus attempt to reduce the number of candidates by taking into account the network flows that collectively correspond to a large percentage of the fading states. We refer to this set of network flows as the dominant set. Since we employ a block-fading model, the flows that are optimal in a percentage p of the fading states are also optimal in a percentage p of the video frames for an asymptotically large number of video frames. Thus, we can estimate the dominant set using an offline simulation of the brute force algorithm. After running the offline brute force simulation for a large number of frames, we obtain statistics on the flow usage. We sort the flows by percentage of usage and select the sorted flows in descending order such that he total usage of the selected set is greater than a threshold. In our simulations, we use a threshold of 90%, and call this the 90th percentile dominant set. As the threshold increases, the dominant set solution approaches the optimal solution but the complexity of flow selection increases. On the other hand, a smaller threshold corresponds to a looser approximation, but reduced complexity. Using this approximation scheme, we can achieve solutions asymptotically close to the optimal solution. In fact, we will show that this algorithm exhibits better performance than the first one at a higher computational cost. Despite the asymptotic performance, one major limitation in this approach is that it is topology dependent, that is, we have different dominant sets for different instances of the MS locations. This makes the algorithm suitable for low mobility scenarios. However, it is worth noting that the variation of the dominant set as a function of the MS locations is smooth, so that the same set can be used for a well-defined set of network topologies. A mobility model can also be used in the process of populating the dominant flow set to obtain more accurate statistics on flow use and better approximations. In general, the more clustered the network topology and the lower the mobility, the smaller the size of the dominant set. Finally, we should note that although the dominant set construction is not a real-time process, this offline simulation might not always be computationally feasible if K is somewhat large. In this case, we may consider the following alternative approaches to make the dominant set construction feasible:

1.Populate the dominant set using the minimum spanning tree algorithm: After running the minimum spanning tree algorithm for a sufficiently long training period, all the flows selected at least once will be used to form the dominant set. This training phase is computationally very efficient and it might end up with a superset of the 90th percentile set due to the frequent reuse of the “good” flows, thus it may perform better.

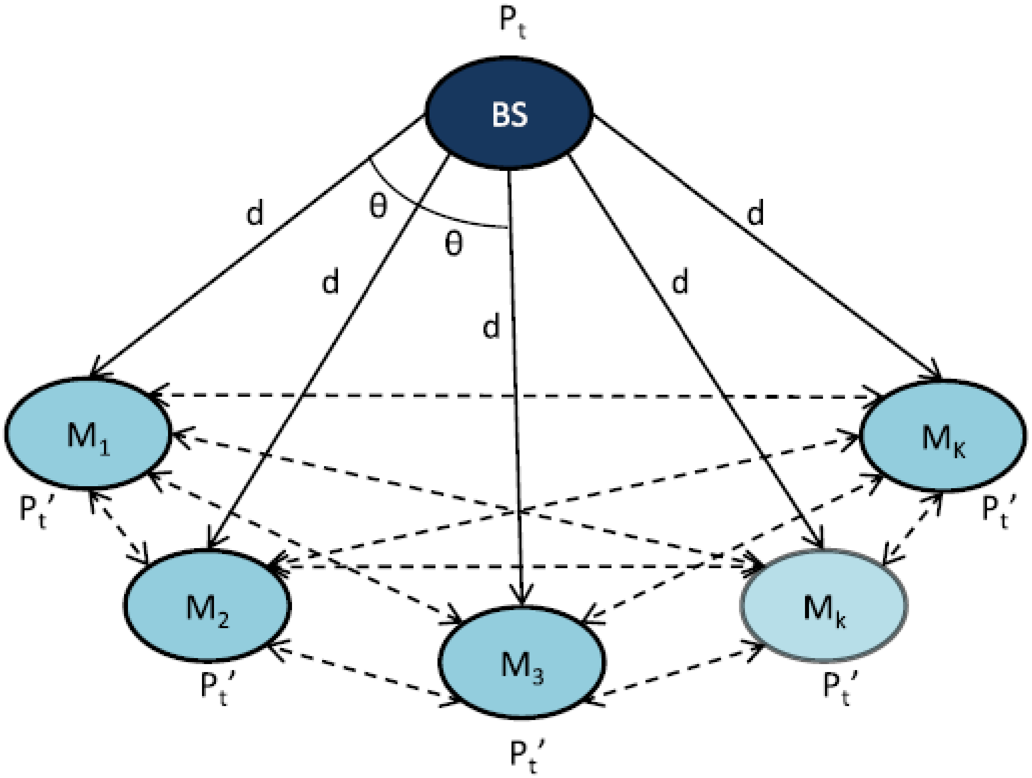


Fig. 2. Simulation model with one BS and K MSs.

2. Use a clustering algorithm to group the mobile stations according to the physical MS locations: After grouping the mobile stations, choose the clusterheads as the closest MS in each cluster to the BS, perform an optimal flow selection on each group separately using the clusterhead as the source, use the BS to multicast to all clusterheads and use the optimal flow within each cluster/group.

3. Place restrictions on the number of flows: Limit the search space for dominant set construction, e.g., by limiting the maximum number of hops of the mulicast group size.

We will show that the number of dominant flows is polynomial in K unlike the total number of flows because the percentage of dominant flows decays exponentially as K increases (See Fig. 8b). For general topologies, the dominant set has size $O(K^a)$ where the exponent a is a constant that depends on the network topology and mobility dynamics. A good fit typically yields a between two and three. Since the resource allocation problem is solved for each of the $O(k^a)$ dominant flow to obtain the best dominant flow, the time complexity of the flow selection and resource allocation is $O((K^2.+KL).K^a\ )$The complexity of the different flow selection algorithms is summarized in Table 1.

2.6 Summary

These were certain important papers which helped us a lot in completing our project. Information provided by these papers was very helpful for us.

CHAPTER 3

Testbed, Experimental Setup and Simulation Outputs

3.1 Introduction:

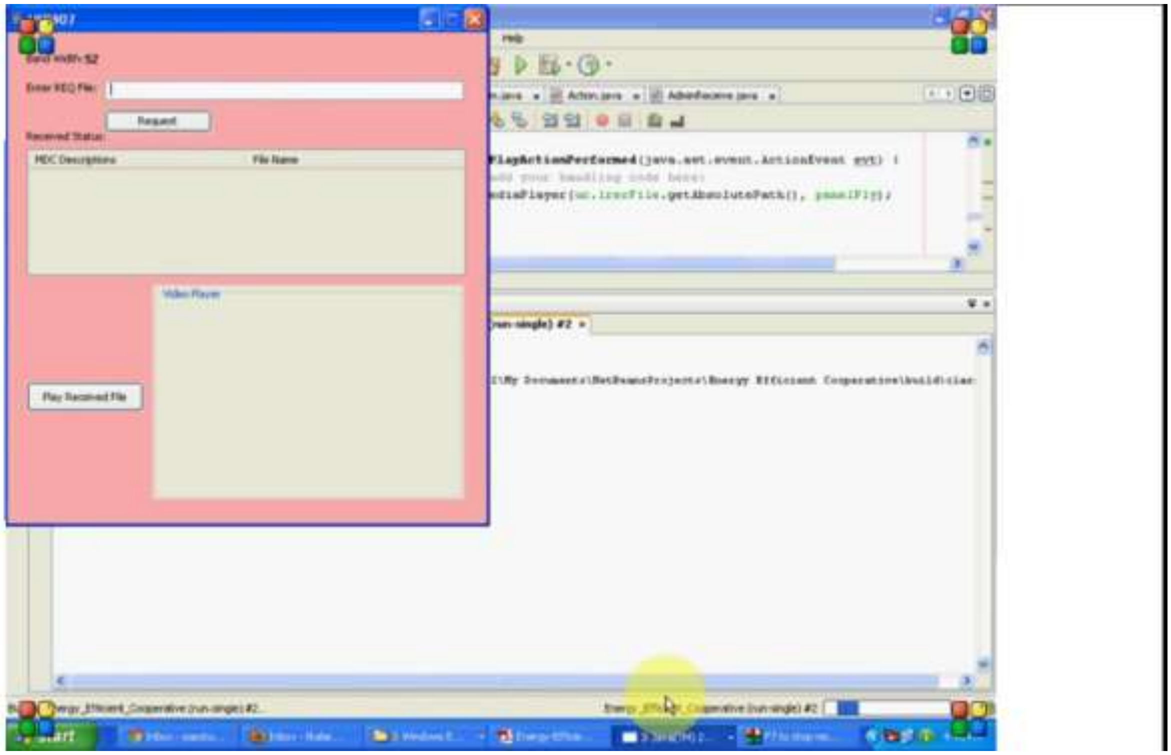
I have used Java Technology (i.e. JDK 1.7) for simulating and this software is running on top of the Vaio System, running with Windows 7 Home Basic operating system. I have made use of Java Applets and all the panels have been incorporated on the same applet.

3.2 Software Used

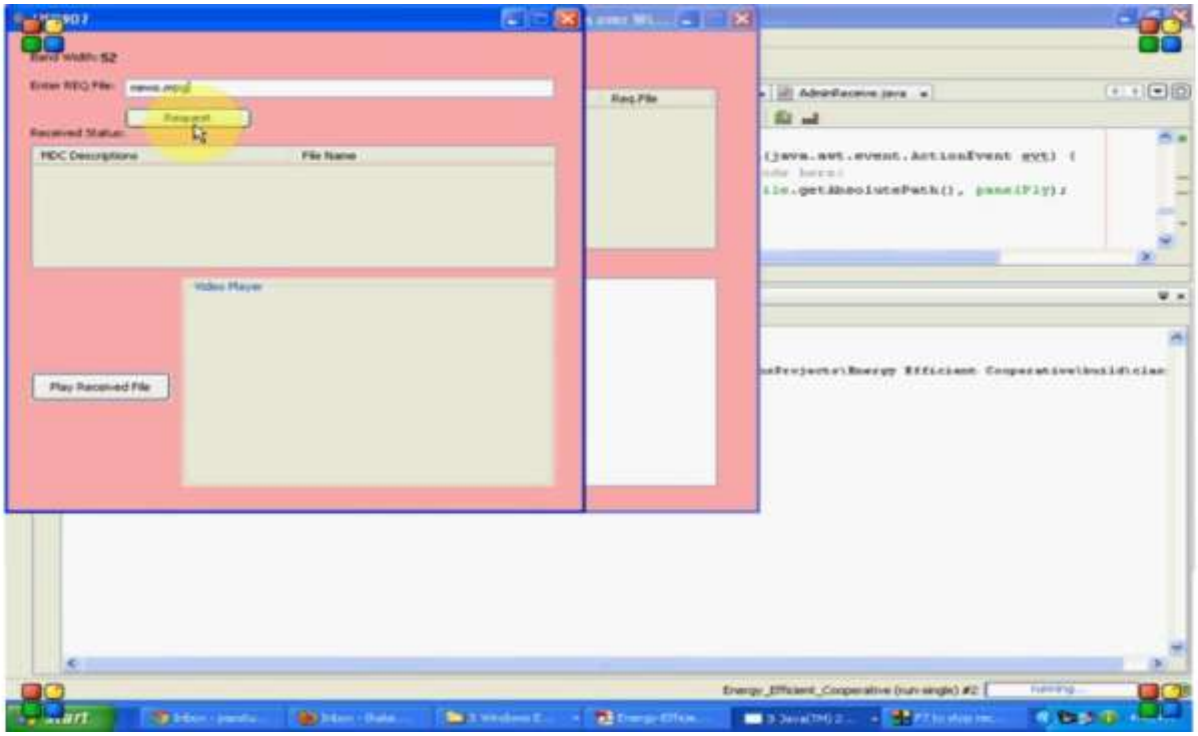
I have used NetBeansIDE 6.9 version for running my software. **NetBeans** is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5.It is also an application platform framework for Java desktop applications and others. The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by third party developers.The NetBeans Team actively support the product and seek future suggestions from the wider community. Every release is preceded by a time for Community testing and feedback

3.3 Requesting for video

This tool is designed for users to download the videos which are available at server end for downloading. user just have to request the file for downloading it but user should know the name of the video as if the requested video is not available at server end then user cannot download it

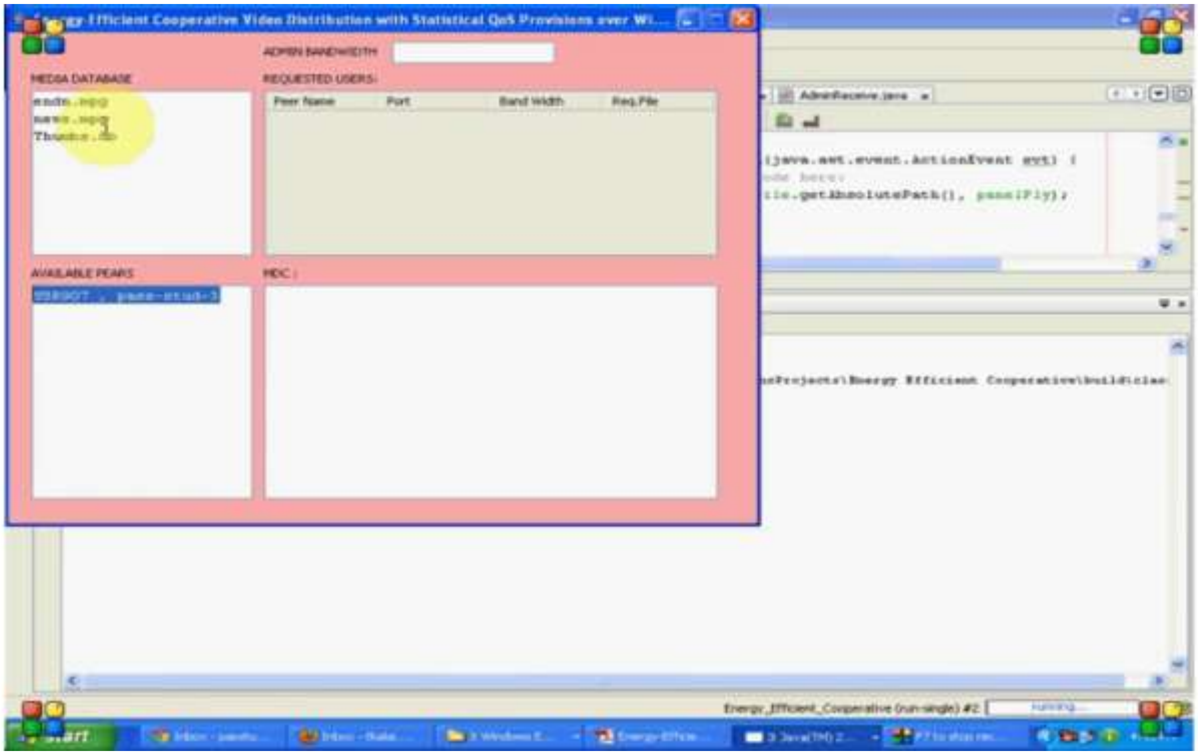


As you can see in below fig that a video is requested by user named news.mpg which is already stored in server database

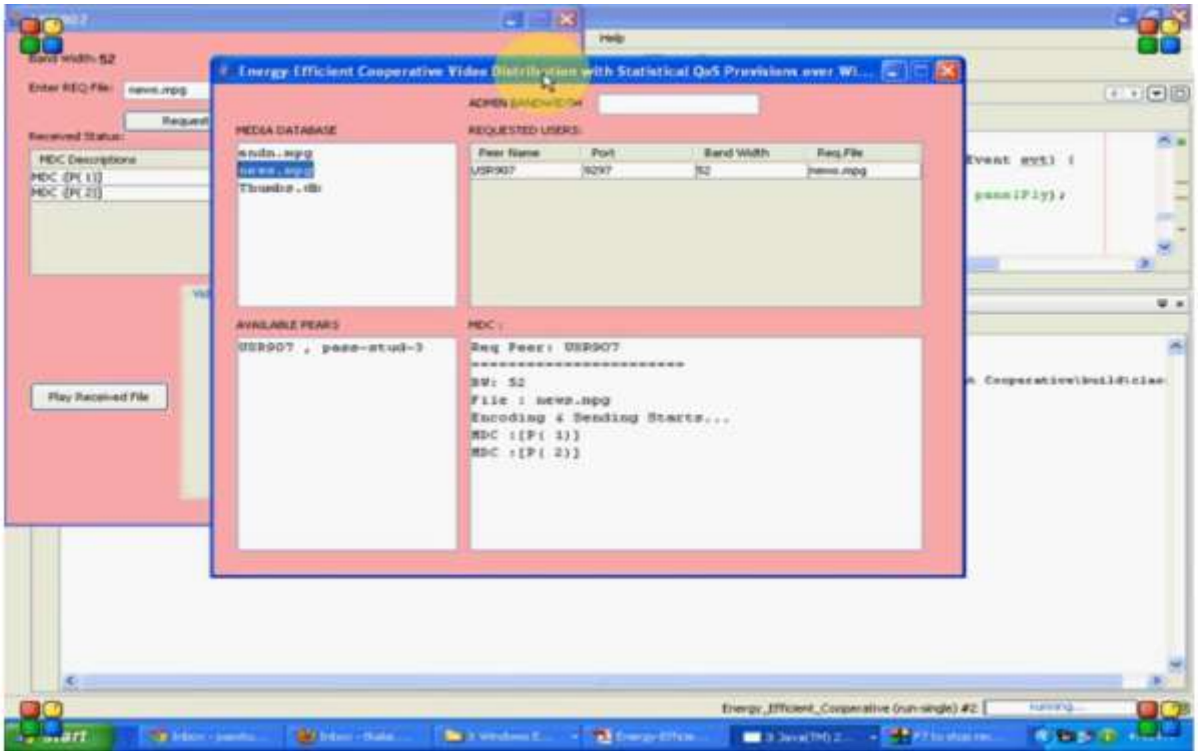

Enter REQ File: news.mpg

3.4 Sending and Receiving a video

As soon as we run the user file it will automatically pick a random number and assign it to user as it will be user id which is given to him by the sever and you can see the name and user id of user in server API under available peers



After requesting for the video it will be send to user in form of packets and each packet is transferred one by one as you can see in the fig.



Band width: 52

Enter REQ File: news.mpg

Received Status:

MDC Descriptions
MDC :[P(1)]
MDC :[P(2)]

Energy Efficient Cooperative Video Distribution with Statistical QoS Provisions over Wi...

ADMIN BANDWIDTH

MEDIA DATABASE:

[illegible].mpg
news.mpg
Thumbs.db

REQUESTED USERS:

Peer Name	Port	Band Width	Req.File
USR907	9297	52	news.mpg

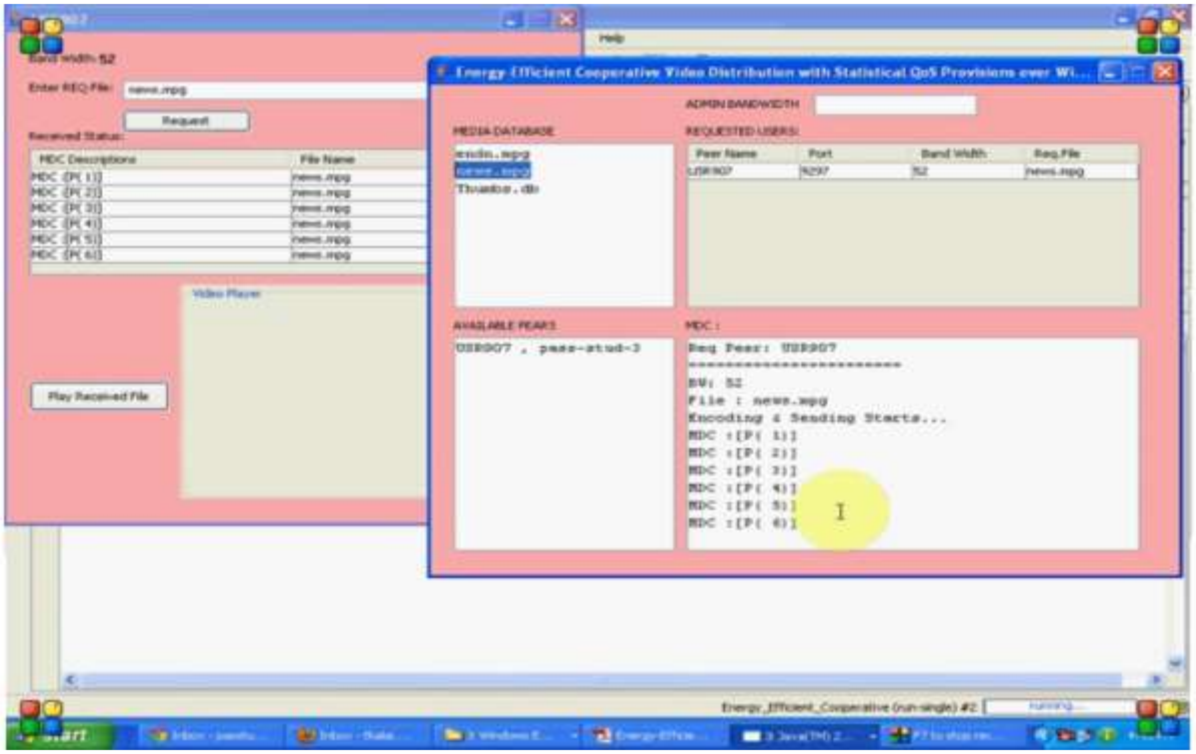
AVAILABLE PEARS:

USR907 , pass-stud-3

MDC :

```
Req Peer: USR907
========================
BW: 52
File : news.mpg
Encoding & Sending Starts...
MDC :[P( 1)]
MDC :[P( 2)]
```

Simultaneously the packets are received at the user end as you can see



Band width: 52

Enter REQ File: news.mpg

Received Status:

MDC Descriptions	File Name
MDC :[P(1)]	news.mpg
MDC :[P(2)]	news.mpg
MDC :[P(3)]	news.mpg
MDC :[P(4)]	news.mpg
MDC :[P(5)]	news.mpg
MDC :[P(6)]	news.mpg

Energy Efficient Cooperative Video Distribution with Statistical QoS Provisions over Wi...

ADMIN BANDWIDTH

MEDIA DATABASE:

[illegible].mpg
news.mpg
Thumbs.db

REQUESTED USERS:

Peer Name	Port	Band Width	Req.File
USR907	9297	52	news.mpg

AVAILABLE PEARS:

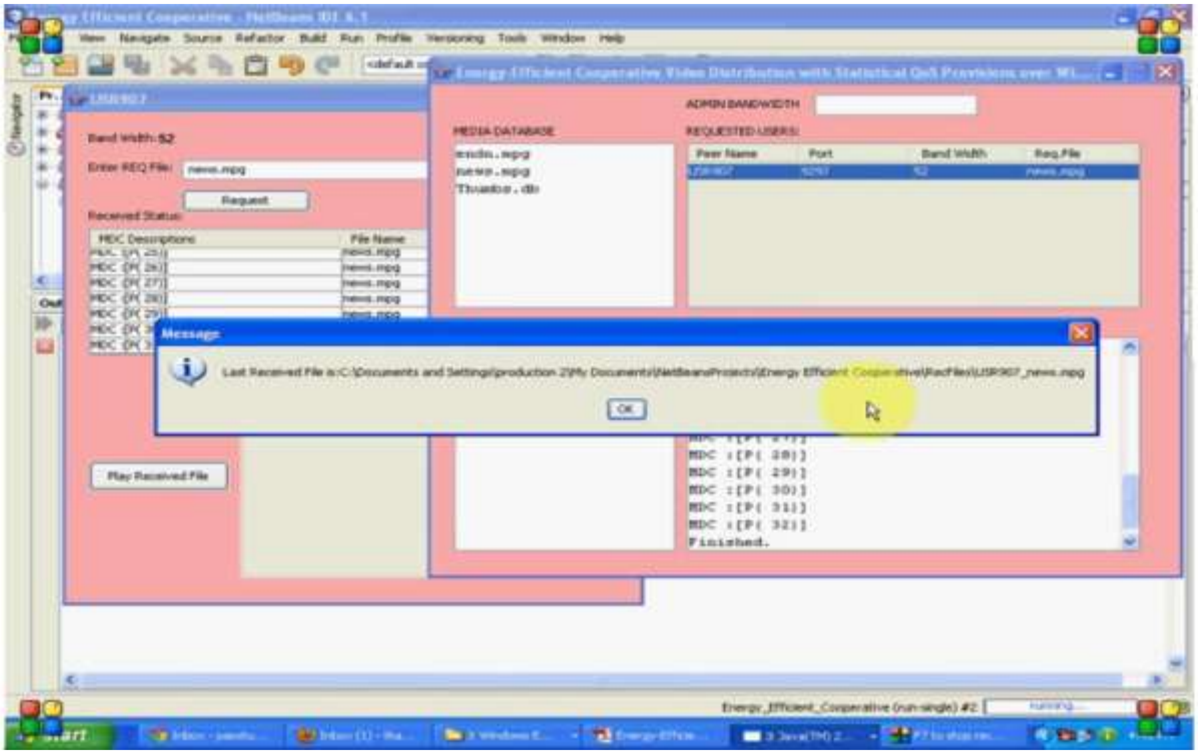
USR907 , pass-stud-3

MDC :

```
Req Peer: USR907
========================
BW: 52
File : news.mpg
Encoding & Sending Starts...
MDC :[P( 1)]
MDC :[P( 2)]
MDC :[P( 3)]
MDC :[P( 4)]
MDC :[P( 5)]
MDC :[P( 6)]
```

3.5 Saving and playing of the video

After the video has been received by user it will be save to the folder RECfiles which is saved in the project folder.



Band Width: 52

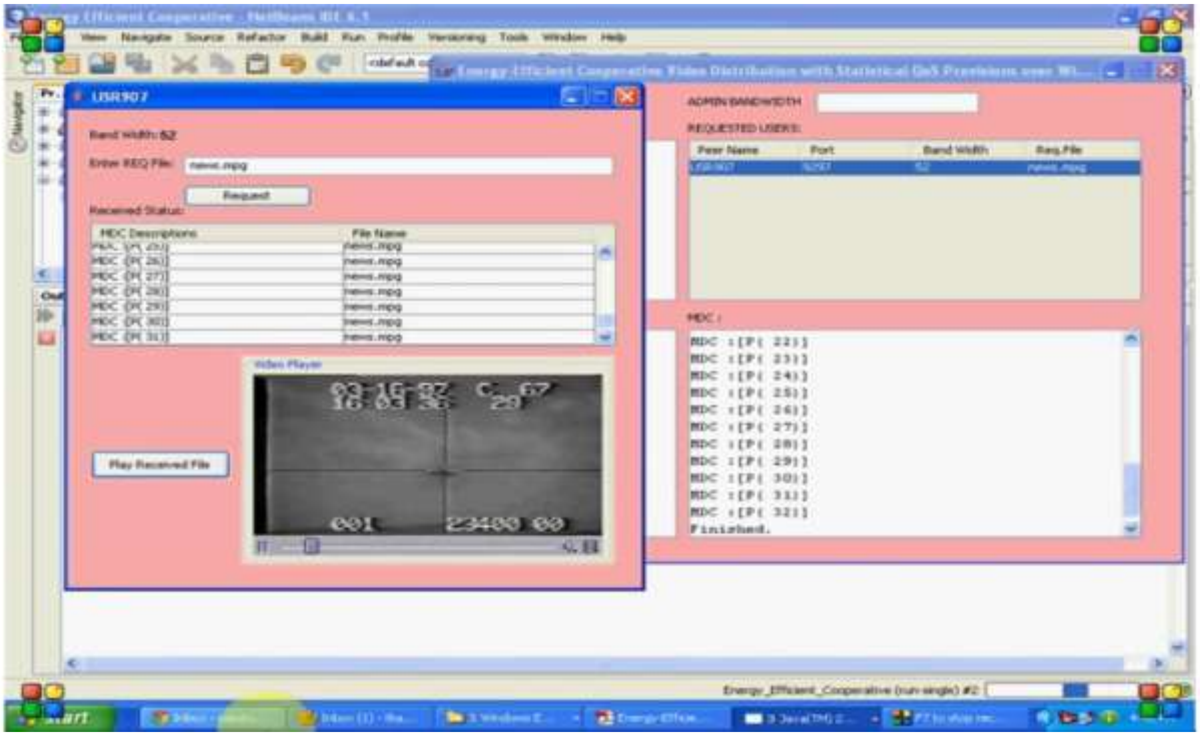
Enter REQ File: news.mpg

Message

Last Received File is:C:\Documents and Settings\production 2\My Documents\NetBeansProjects\Energy Efficient Cooperative\RecFiles\USR907_news.mpg

```
MDC :[P( 28)]
MDC :[P( 29)]
MDC :[P( 30)]
MDC :[P( 31)]
MDC :[P( 32)]
Finished.
```

After the file has been saved you can play the video as shon in below fig. you file will be save under name of your user id and the name of the video requested by you like if the name of the user is USR907 and video name is news.mpg then it will be saved under the name of USR907_news.mpg.



Band Width: 52

Enter REQ File: news.mpg

Received Status:

MDC Descriptions	File Name
MDC :[P(25)]	news.mpg
MDC :[P(26)]	news.mpg
MDC :[P(27)]	news.mpg
MDC :[P(28)]	news.mpg
MDC :[P(29)]	news.mpg
MDC :[P(30)]	news.mpg
MDC :[P(31)]	news.mpg

ADMIN BANDWIDTH

REQUESTED USERS:

Peer Name	Port	Band Width	Req.File
USR907	5297	52	news.mpg

MDC :

```
MDC :[P( 22)]
MDC :[P( 23)]
MDC :[P( 24)]
MDC :[P( 25)]
MDC :[P( 26)]
MDC :[P( 27)]
MDC :[P( 28)]
MDC :[P( 29)]
MDC :[P( 30)]
MDC :[P( 31)]
MDC :[P( 32)]
Finished.
```

Energy_Efficient_Cooperative (run-single) #2

CHAPTER 4

FUTURE DIRECTIONS

4.1 Future Scope

The future vision of WSNs is to embed numerous distributed devices to monitor and interact with physical world phenomena, and to exploit spatially and temporally dense sensing and actuation capabilities of those sensing devices. These nodes coordinate among themselves to create a network that performs higher-level tasks.

Although extensive efforts have been exerted so far on the routing problem in WSNs, there are still some challenges that confront effective solutions of the routing problem. First, there is a tight coupling between sensor nodes and the physical world. Sensors are embedded in unattended places or systems. This is different from traditional Internet, PDA, and mobility applications that interface primarily and directly 32 with human users. Second, sensors are characterized by a small foot print, and as such nodes present stringent energy constraints since they are equipped with small, finite, energy source. This is also different from traditional fixed but reusable resources. Third, communications is primary consumer of energy in this environment where sending a bit over 10 or 100 meters consumes as much energy as thousands-to-millions of operations (known as R4 signal energy drop-off) [36].

Although the performance of these protocols is promising in terms of energy efficiency, further research would be needed to address issues such as Quality of Service (QoS) posed by video and imaging sensors and real-time applications. Energy-aware QoS routing in sensor networks will ensure guaranteed bandwidth (or delay) through the duration of connection as well as providing the use of most energy efficient path. Another interesting issue for routing protocols is the consideration of node mobility. Most of the current protocols assume that the sensor nodes and the BS are stationary. However, there might be situations such as battle environments where the BS and possibly the sensors need to be mobile. In such cases, the frequent update of the position of the command node and the sensor nodes and the propagation of that information through the network may excessively drain the energy of nodes. New routing algorithms are needed in order to handle the overhead of mobility and topology changes in such energy constrained environment. Future trends in routing techniques in WSNs focus on different directions, all share the common objective of prolonging the network lifetime. We summarize some of these directions and give some pertinent references as follows:

- Exploit redundancy: typically a large number of sensor nodes are implanted inside or beside the phenomenon. Since sensor nodes are prone to failure, fault tolerance techniques come in picture to keep the network operating and performing its tasks. Routing techniques that explicitly employ fault tolerance techniques in an efficient manner are still under investigation.

- Tiered architectures (mix of form/energy factors): Hierarchical routing is an old technique to enhance scalability and efficiency of the routing protocol. However, novel techniques to network clustering which maximize the network lifetime are also a hot area of research in WSNs.

- Exploit spatial diversity and density of sensor/actuator nodes: Nodes will span a network area that might be large enough to provide spatial communication between sensor nodes. Achieving energy efficient communication in this densely populated environment deserves further investigation. The dense deployment of sensor nodes should allow the network to adapt to unpredictable environment.

- Achieve desired global behavior with adaptive localized algorithms (i.e., do not rely on global inter- action or information). However, in a dynamic environment, this is hard to model.

• Leverage data processing inside the network and exploit computation near data sources to reduce communication, i.e., perform in-network distributed processing. WSNs are organized around naming data, not nodes identities. Since we have a large collections of distributed elements, localized algo- rithms that achieve system-wide properties in terms of local processing of data before being sent to the destination are still needed. Nodes in the network will store named data and make it available for processing. There is a high need to create efficient processing points in the network, e.g., duplicate suppression, aggregation, correlation of data. How to efficiently and optimally find those points is still an open research issue.

• Time and location synchronization: energy-efficient techniques for associating time and spatial coor- dinates with data to support collaborative processing are also required [20].

• Localization: sensor nodes are randomly deployed into an unplanned infrastructure. The problem of estimating spatial-coordinates of the node is referred to as localization. Global Positioning System (GPS) cannot be used in WSNs as GPS can work only outdoors and cannot work in the presence of any obstruction. Moreover, GPS receivers are expensive and not suitable in the construction of small cheap sensor nodes. Hence, there is a need to develop other means of establishing a coordinate system without relying on an existing infrastructure. Most of the proposed localization techniques today, depend on recursive trilateration/multilateration techniques (e.g., [38]) which would not provide enough accuracy in WSNs.

• Self-configuration and reconfiguration is essential to lifetime of unattended systems in dynamic, and constrained energy environment. This is important for keeping the network up and running. As nodes die and leave the network, update and reconfiguration mechanisms should take place. A feature that is important in every routing protocol is to adapt to topology changes very quickly and to maintain the network functions.

• Secure Routing: Current routing protocols optimize for the limited capabilities of the nodes and the application specific nature of the networks, but do not consider security. Although these protocols have not been designed with security as a goal, it is important to analyze their security properties. One aspect of sensor networks that complicates the design of a secure routing protocol is in-network aggregation. In WSNs, in-network processing makes end-to-end security mechanisms harder to deploy because intermediate nodes need direct access to the contents of the messages.

- Other possible future research for routing protocols includes the integration of sensor networks with wired networks (i.e. Internet). Most of the applications in security and environmental monitoring require the data collected from the sensor nodes to be transmitted to a server so that further analysis can be done. On the other hand, the requests from the user should be made to the BS through Inter- net. Since the routing requirements of each environment are different, further research is necessary for handling these kinds of situations.

APPENDIX

Code Of GUI for Admin

```
package com.design;
import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;
import com.actions.Action;
import com.actions.AdminReceive;
import com.multicast.MulticastRx;
import java.awt.Color;

public class Admin extends javax.swing.JFrame
{

private static final long serialVersionUID = 1L;
Action action;

/** Creates new form Admin */
```

```
public Admin()
{
initComponents();
init();
}

private void init() {
// TODO Auto-generated method stub
action = new Action();
action.loadFiles(jtaDB);
new AdminReceive(this);
new MulticastRx(this);
}

/**
* This method is called from within the constructor to initialize the form.
private void initComponents() {

jLabel1 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jtaDB = new javax.swing.JTextArea();
jLabel2 = new javax.swing.JLabel();
txtBW = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
tbl = new javax.swing.JTable();
jScrollPane3 = new javax.swing.JScrollPane();
jtaMDC = new javax.swing.JTextArea();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jScrollPane4 = new javax.swing.JScrollPane();
```

```
jtaPeers = new javax.swing.JTextArea();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(null);

jLabel1.setText("MEDIA DATABASE");
getContentPane().add(jLabel1);
jLabel1.setBounds(20, 40, 130, 16);

jtaDB.setColumns(20);
jtaDB.setRows(5);
jScrollPane1.setViewportView(jtaDB);

getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(20, 60, 190, 170);

jLabel2.setText("ADMIN BANDWIDTH");
getContentPane().add(jLabel2);
jLabel2.setBounds(220, 10, 110, 20);
getContentPane().add(txtBW);
txtBW.setBounds(330, 10, 140, 22);

jLabel3.setText("REQUESTED USERS:");
getContentPane().add(jLabel3);
jLabel3.setBounds(220, 40, 120, 16);

dft = new DefaultTableModel();
tbl.setModel(dft);
jScrollPane2.setViewportView(tbl);
dft.addColumn("Peer Name");
dft.addColumn("Port");
```

```
dft.addColumn("Band Width");
dft.addColumn("Req.File");

getContentPane().add(jScrollPane2);
jScrollPane2.setBounds(220, 60, 390, 170);

jtaMDC.setColumns(20);
jtaMDC.setRows(5);
jScrollPane3.setViewportView(jtaMDC);

getContentPane().add(jScrollPane3);
jScrollPane3.setBounds(220, 260, 390, 220);

jLabel4.setText("MDC :");
getContentPane().add(jLabel4);
jLabel4.setBounds(220, 240, 190, 16);

jLabel5.setText("AVAILABLE PEARS");
getContentPane().add(jLabel5);
jLabel5.setBounds(20, 240, 150, 16);

jtaPeers.setColumns(20);
jtaPeers.setRows(5);
jScrollPane4.setViewportView(jtaPeers);

getContentPane().add(jScrollPane4);
jScrollPane4.setBounds(20, 260, 190, 220);
getContentPane().setBackground(Color.PINK);
setSize(650, 537);
setTitle("Energy-Efficient Cooperative Video Distribution with Statistical QoS Provisions
over Wireless Networks -Admin:");
```

```
setResizable(false);
setVisible(true);

}

public static void main(String args[]) {
try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
} catch (Exception e) {
e.printStackTrace();
}
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new Admin().setVisible(true);
}
});
}

// GEN-BEGIN:variables
// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
public javax.swing.JTextArea jtaDB;
public javax.swing.JTextArea jtaMDC;
```

```
public javax.swing.JTextArea jtaPeers;
private javax.swing.JTable tbl;
private javax.swing.JTextField txtBW;
public DefaultTableModel dft;
// End of variables declaration//GEN-END:variables

}
```

Code of GUI for user

```
package com.design;

import javax.swing.UIManager;
import javax.swing.table.DefaultTableModel;

import com.actions.*;

import com.multicast.MulticastTx;
import java.awt.Color;


public class User extends javax.swing.JFrame {


private static final long serialVersionUID = 1L;
Action action;
public String source;
int port;
int bw;
UserReceive ur;
```

```
/** Creates new form User */
public User() {
initComponents();
action = new Action();
init();
}

private void init() {
source = action.getSource();
setTitle(source);
bw = action.getBW();
jLabel2.setText("" + bw);
port = action.getPort();
action.setProperty("SCPorts.properties", source, "" + port);
ur = new UserReceive(this, port);
new MulticastTx(source);

}



// This method is called from within the constructor to initialize the form.



private void initComponents() {

jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
txtReqFile = new javax.swing.JTextField();
btnREQ = new javax.swing.JButton();
```

```
jLabel4 = new javax.swing.JLabel();
btnPlay = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
panelPly = new javax.swing.JPanel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(null);

jLabel1.setText("Band Width:");
getContentPane().add(jLabel1);
jLabel1.setBounds(20, 20, 60, 14);

jLabel2.setFont(new java.awt.Font("Arial", 1, 12));
getContentPane().add(jLabel2);
jLabel2.setBounds(80, 20, 110, 14);

jLabel3.setText("Enter REQ File:");
getContentPane().add(jLabel3);
jLabel3.setBounds(20, 50, 80, 14);
getContentPane().add(txtReqFile);
txtReqFile.setBounds(100, 50, 370, 20);

btnREQ.setText("Request");
btnREQ.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
btnREQActionPerformed(evt);
}
});
getContentPane().add(btnREQ);
btnREQ.setBounds(100, 80, 110, 23);
```

```
jLabel4.setText("Received Status:");
getContentPane().add(jLabel4);
jLabel4.setBounds(20, 100, 190, 14);

btnPlay.setText("Play Received File");
btnPlay.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
btnPlayActionPerformed(evt);
}
});
getContentPane().add(btnPlay);
btnPlay.setBounds(20, 360, 120, 30);

dft = new DefaultTableModel();
jTable1.setModel(dft);
jScrollPane1.setViewportView(jTable1);
dft.addColumn("MDC Descriptions");
dft.addColumn("File Name");
getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(20, 120, 452, 130);
getContentPane().setBackground(Color.PINK);
panelPly.setBorder(javax.swing.BorderFactory
.createTitledBorder("Video Player"));
panelPly.setLayout(null);
getContentPane().add(panelPly);
panelPly.setBounds(150, 260, 320, 220);

setSize(500, 537);
setResizable(false);
setVisible(true);
```

```
}

private void btnPlayActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
action.mediaPlayer(ur.lrecFile.getAbsolutePath(), panelPly);

}

private void btnREQActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
action.sendREQ(source, jLabel2.getText(), port, txtReqFile.getText());
}


public static void main(String args[]) {
try {
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
} catch (Exception e) {
e.printStackTrace();
}
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new User().setVisible(true);
}
});
}

// GEN-BEGIN:variables
// Variables declaration - do not modify
private javax.swing.JButton btnPlay;
private javax.swing.JButton btnREQ;
```

```
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JPanel panelPly;
private javax.swing.JTextField txtReqFile;
public DefaultTableModel dft;
// End of variables declaration//GEN-END:variables
```

REFRENCES

[1] Amin Abdel Khalek and Zaher Dawy“Energy Efficient Cooperative Video Distibution with statistical qos provision over wsn

[2] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,” IEEE Trans. Circuits and Systems for Video Technology, vol. 17, no. 9,

pp. 1103-1120, Sept. 2007

[3] A. Kumar, D. Manjunath, and J. Kuri, Communication Networking: An Analytical Approach. Morgan Kaufmann, 2004.

[4] C.-S. Chang, Performance Guarantees in Communication Networks. Springer-Verlag, 2000.

[5] C.-S. Chang and J. Thomas, “Effective Bandwidth in High Speed Digital Networks,” IEEE J. Selected Areas in Comm., vol. 13, no. 6, pp. 1091-1100, Aug. 1995.

[6] D. Wu and R. Negi, “Effective Capacity: A Wireless Link Model for Support of Quality of Service,” IEEE Trans. Wireless Comm., vol. 2, no. 4, pp. 630-643, July 2003.

[7] D. Wu and R. Negi, “Effective Capacity-Based Quality of Service Measures for Wireless Networks,” Mobile Networks and Applications, vol. 11, no. 1, pp. 91-99, June 2006.

[8] D. Wu, “Providing Quality-of-Service Guarantees in Wireless Networks,” PhD thesis, Carnegie Mellon Univ., 2003.

[9] Q. Du and X. Zhang, “Statistical QoS Provisionings for Wireless Unicast/Multicast of Layered Video Streams,” Proc. IEEE INFOCOM, Apr. 2009.

[10] X. Zhang and Q. Du, “Cross-Layer Modeling for QoS-Driven Multimedia Multicast/Broadcast over Fading Channels in Mobile Wireless Networks,” IEEE Comm. Magazine, vol. 45, no. 8, pp. 62-

70, Aug. 2007. [11] H. Wu, C. Qiao, S. De, and O. Tonguz, “Integrated Cellular and

Ad Hoc Relaying Systems: iCAR,” IEEE J. Selected Areas in Comm.,

vol. 19, no. 10, pp. 2105-2115, Oct. 2001.

[12] H. Luo, R. Ramjeey, P. Sinha, L.E. Li, and S. Lu, “UCAN: A Unified Cellular and Ad Hoc Network Architecture,” Proc. ACM MobiCom, Sept. 2003.

[13] R. Bhatia, L.E. Li, H. Luo, and R. Ramjee, “ICAM: Integrated Cellular and Ad Hoc Multicast,” IEEE Trans. Mobile Computing, vol. 5, no. 8, pp. 1004-1015, Aug. 2006.

[14] M. Peng and W. Wang, “Investigation of Cooperative Relay Node Selection in Heterogeneous Wireless Communication Systems,” Proc. IEEE Int’l Conf. Comm., May 2008.