Rescue - Report

(FREQUENCY AND TIME SLOT ASSIGNMENT ALGORITHM)

Guy Even* Moni Shahar† Alexander Zadorojniy‡

June 22, 2009

1 Frequency and Time Slot Assignment Algorithm

1.1 Project Main Blocks Overview

The frequency assignment problem is defined as follows. Given a set of clients and a set of streams between pairs of clients; we need to create a schedule and frequency allocation such that all the streams can be transmitted. Since transmitting all the streams may be infeasible, we define ρ_i to be the ratio of the bandwidth achieved divided by the bandwidth required for stream i. The problem we optimize here is to find a schedule for which we maximize $\rho \triangleq \min_i \rho_i$.

The project consists of four main blocks: scenario generator, multi-commodity flow algorithm, a scheduler and a simulator. The role of the scenario generator is to create instances of the problem, namely, a communication graph and requirements which are the inputs to the multi-commodity flow algorithm module. The algorithm module generates a conflict graph, flow constraints, defining a problem as a linear program and solving it using LP solver (CLP). The outputs of the algorithm module are flows and parameter ρ which defines the fraction of the requirements that is satisfied. Based on this data, a scheduler creates a frequency and time slots schedule table, which defines in what time slot and on which frequency each node receives or transmits. This table is input to the simulator module which models the IEEE 802.11b (we are planning to consider an extension to 802.11g and 802.11n as well) protocol. In this report we describe the multi-commodity flow algorithm and scheduler modules. Figure 1 shows the

*School of Electrical Engineering, Tel-Aviv Univ., Tel-Aviv 69978, Israel. guy@eng.tau.ac.il
†School of Electrical Engineering, Tel-Aviv Univ., Tel-Aviv 69978, Israel. moni@eng.tau.ac.il
‡School of Electrical Engineering, Tel-Aviv Univ., Tel-Aviv 69978, Israel. sasha@eng.tau.ac.il

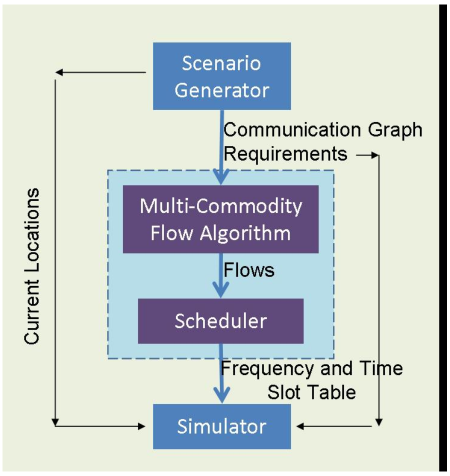

Figure 1: Project Block Diagram

structure of the project.

1.2 Multi-Commodity Flows

Single Frequency and Unicasting. The Communication Graph is an undirected graph $G' = (V', E')$, where each of its nodes $e \in E'$ has a capacity $c'(e)$. The i-th transmission requirement is defined by the triple (s_i, t_i, d_i) where s_i is the source node, t_i is the target node and d_i is the bandwidth demand. The flow graph is a directed graph $G = (V, E)$, where $V = V'$ and each undirected edge $e \in E'$ of the communication graph was split to two directed antisymmetric edges e and e^{rev}, where $|E| = 2 \cdot |E'|$.

Let $f_i(e)$ be a flow of commodity i in edge e. For a vertex $v \in V$ we define the neighborhood of v as $N(v) = \{u|(u, v) \in E\}$. The set of edges that can not be active while v is transmitting is defined by $conf(v) = \{e|e = (v, u)$ or $e = (u, w)$ where $u \in N(v)\}$.

Actually this conflict set defines the set of edges and their reversal edges with respect to v, which are in conflict with each other when v transmits. From these edges the only one can be active at each time.

The following constraints have to be satisfied.

$$\forall i, \forall v \neq s_i, t_i : \sum_{e \in in(v)} f_i(e) = \sum_{e \in out(v)} f_i(e). \; (Conservation) \tag{1}$$

$$\forall e : \sum_i f_i(e) + f_i(e^{rev}) \leq c'(e). \; (Capacity) \tag{2}$$

$$\forall i : \sum_{e \in out(s_i)} f_i(e) - \sum_{e \in in(s_i)} f_i(e) \geq \rho \cdot d_i. \; (Requirements) \tag{3}$$

$$\forall v : \sum_{e \in conf(v)} \sum_i \frac{f_i(e) + f_i(e^{rev})}{c'(e)} \leq 1. \; (Conflicts) \tag{4}$$

$$\forall i, e, f_i(e) \geq 0. \tag{5}$$

$$\rho \leq 1. \tag{6}$$

$$\max \rho. \; (Goal) \tag{7}$$

Multiple Frequencies and Unicasting. In the frequency assignment problem we assume that there is a fixed set J of frequencies common to all vertices. Let $f_{i,j}(e)$ be a flow of commodity i with frequency j on edge e.

The flow conservation, the capacity and the requirements constraints are similar to the single frequency case.

$$\forall i, \forall v \neq s_i, t_i : \sum_{j, e \in in(v)} f_{i,j}(e) = \sum_{j, e \in out(v)} f_{i,j}(e). \; (Conservation) \tag{8}$$

$$\forall e : \sum_{i,j} (f_{i,j}(e) + f_{i,j}(e^{rev})) \leq c'(e). \; (Capacity) \tag{9}$$

$$\forall i : \sum_{j, e \in out(s_i)} f_{i,j}(e) - \sum_{j, e \in in(s_i)} f_{i,j}(e) \geq \rho \cdot d_i. \; (Requirements) \tag{10}$$

$$\forall i, j, e, f_{i,j}(e) \geq 0. \tag{11}$$

$$\rho \leq 1. \tag{12}$$

However, the conflicts constraints for multiple frequencies are more complex than for a single frequency. Every node must satisfy two constraints. The first constraint prevents a simultaneous transmissions and receptions on the node. The second constraint prevents mutual interference as in the single frequency case.

$$\forall v : \sum_{u \in N(v)} \sum_i \sum_j \frac{f_{i,j}(e) + f_{i,j}(e^{rev})}{c'(e)} \leq 1, \text{where } e = (v,u). \tag{13}$$

$$\forall v, \forall j : \sum_{e \in conf(v)} \sum_i \frac{f_{i,j}(e) + f_{i,j}(e^{rev})}{c'(e)} \leq 1. \tag{14}$$

Now we can define the LP to maximize the transmission rate.

$$max\ \rho \tag{15}$$

$$s.t. \tag{16}$$

$$eq.\ (8-14) \text{ hold.} \tag{17}$$

1.3 Scheduling

The inputs to the scheduling algorithm are the flows $f_{i,j}(e)$ and a parameter ρ. The output of the algorithm is a table of time slots and frequency assignments. Our scheduling algorithm is based on the greedy algorithm for vertex coloring. An edge e is assigned $n_{e,i,j}$ time slots in frequency j for stream i where:

$$n_{e,i,j} = \lceil \frac{f_{i,j}(e)}{c'(e)} \cdot T \rceil, \text{ where T is the number of time slots.} \tag{18}$$

The format of the time slot assignment table is given in Table 3.

1.4 Implementation Specifications

Reader. Reader is a module which reads the data from the file and stores the extracted parameters in a defined data structure. We have two readers: Communication Graph reader and Stream List reader. They read edge list of Communication Graph file (Table 1) and Stream List file (Table 2), parse them to extract and store the parameters.

$edge\ number$	i	j	$capacity$	$reception\ power$	$modulation$	PER

Table 1: Edge List of Communication Graph

$stream$	$source$	$required\ bandwidth$	$number\ of\ destinations$	$list\ of\ dest.$

Table 2: Stream List Table

Conflicts Creator. The conflicts creator is a module which generates the conflicts on the nodes of the communication graph.

CLP Constraints Writer. A writer is a module which uses data generated by the readers and the conflicts creator and generates constraints for LP Solver (CLP).

Solver. The solver is an IBM freeware solver (CLP) that computes the solution of the linear program.

Scheduler. The inputs to the scheduler are the flows $f_{i,j}(e)$ and a parameter ρ (obtained by CLP), and the output of the algorithm is a time slot assignment table (Table 3).

Files and Tables Definitions. Communication Edge List File (see Table 1)

The first line in the table indicates the number of nodes in the communication graph, the rest of the table's format illustrated on Table 1.

Stream List Requirements File (see Table 2).

The first line in the table indicates the number of requests, the rest of the table's format illustrated on Table 2

Time Slot Assignment Table (Scheduling) (see Table 3).

slot - The slot number.

freq - The frequency used.

transm. - The transmitting node.

stream - The stream number.

#rec - The number of receivers.

rec.list - The list of receivers.

$slot$	$freq$	$transm.$	$stream$	$\#rec.$	$rec.\,list$	$trans.\,rate$	$rec.\,power$	$modul.$	$avr.\#packet$

Table 3: Time Slot Assignment Table

trans. rate - The transmission rate.

receiving power - The power at which a signal is received at the destination node.

modul. - The modulation technique used.

avr.#packet - The average number of packets to be transmitted at the slot on the edge.

Global Parameters List.

T - The number of time slots.

N - The maximum number of nodes.

F - The number of different frequencies.

TWI - The type of wireless interface (e.g., 802.11b or 802.11g).