Fall 2010

Minimize end-to-end delay through cross-layer optimization in multi-hop wireless sensor networks

Yibo Xu

MINIMIZE END-TO-END DELAY THROUGH CROSS-LAYER OPTIMIZATON IN MULTI-HOP WIRELESS SENSOR NETWORKS

by

YIBO XU

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER SCIENCE

2010

Approved by

Maggie Cheng, Advisor
Wei Jiang
Sahra Sedighsarvestani

ABSTRACT

End-to-end delay plays a very important role in wireless sensor networks. It refers to the total time taken for a single packet to be transmitted across a network from source to destination. There are many factors could affect the end-to-end delay, among them the routing path and the interference level along the path are the two basic elements that could have significant influence on the result of the end-to-end delay. This thesis presents a transmission scheduling scheme that minimizes the end-to-end delay when the node topology is given. The transmission scheduling scheme is designed based on integer linear programming and the interference modeling is involved. By using this scheme, we can guarantee that no conflicting transmission will appear at any time during the transmission. A method of assigning the time slot based on the given routing is presented. The simulation results show that the link scheduling scheme can significantly reduce the end-to-end delay. Further, this article also shows two methods which could directly addresses routing and slot assignment, one is MI+MinDelay algorithm and the other is called One-Phase algorithm. A comparison was made between the two and the simulation result shows the latter one leads to smaller latency while it takes much more time to be solved. Besides, due to the different routing policy, we also demonstrate that the shortest path routing does not necessarily result in minimum end-to-end delay.

ACKNOWLEDGMENTS

The author would like to express his gratitude to his advisor, Dr. Maggie Cheng, for her guidance, valuable advice, and encouragement. Without her kind help and understanding, the author could not have reached this goal. Also, thanks go to thesis committee members Dr. Jiang, Wei and Dr. Sedighsarvestani, Sahra for their advice and their participation in the defense. Thanks to all the students in the Network Research Lab at Missouri University Science and Technology for numerous instructive discussions and their kind support of this research. The author would also like to thank Dr. Maggie Cheng for her illuminating explanations of wireless networks in CS385 and CS485 classes.

Finally, the author would like to thank his family for their help and encouragement. With their help, the author has studied hard and lived happily in the peaceful town of Rolla, Missouri.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

1.1. SENSOR NETWORKS

A sensor is such a device that could respond to an input quantity by generating a functionally related output usually in the form of an electrical or optical form. In general, it’s a cheap low-power device that measures a physical quantity (such as heat, light, sound, pressure, magnetism, or a particular motion) and converts it into a signal which can be read by an observer or by an instrument. Due to the small size, the sensors can be easily carried and deployed, but it also makes them have limited processing speed and storage capacity.

During the past two decades, there has been an unprecedented growth in the number of products and services, which utilize information gained by monitoring and measuring using different types of sensors. In many scenarios sensors need to communicate with each other for the purposes of exchanging or sharing data, and such set of sensors performing coordination actions build a Wireless Sensor Network (WSN).

WSN is a wireless network that consists of spatially distributed autonomous sensors which could either have a fixed location or randomly deployed. The development of WSN was originally motivated by military applications such as battlefield surveillance. Now they are widely used in many industrial and civilian application areas, including environmental observation, building monitoring, and healthcare and so on. Sensors usually communicate with each other using a multi-hop approach. The flowing of data ends at special nodes which are called base stations (sometimes they are also referred to as sinks). A base station connects the sensor network to another network (like a gateway) to disseminate the data sensed for further storing or processing. Base stations usually have enhanced capabilities over simple sensor nodes since they must do complex data processing; this justifies the fact that bases stations have workstation/laptop class processors, and of course enough memory, energy, storage and computational power to perform their tasks well. Usually, the communication between base stations is initiated over high bandwidth links.

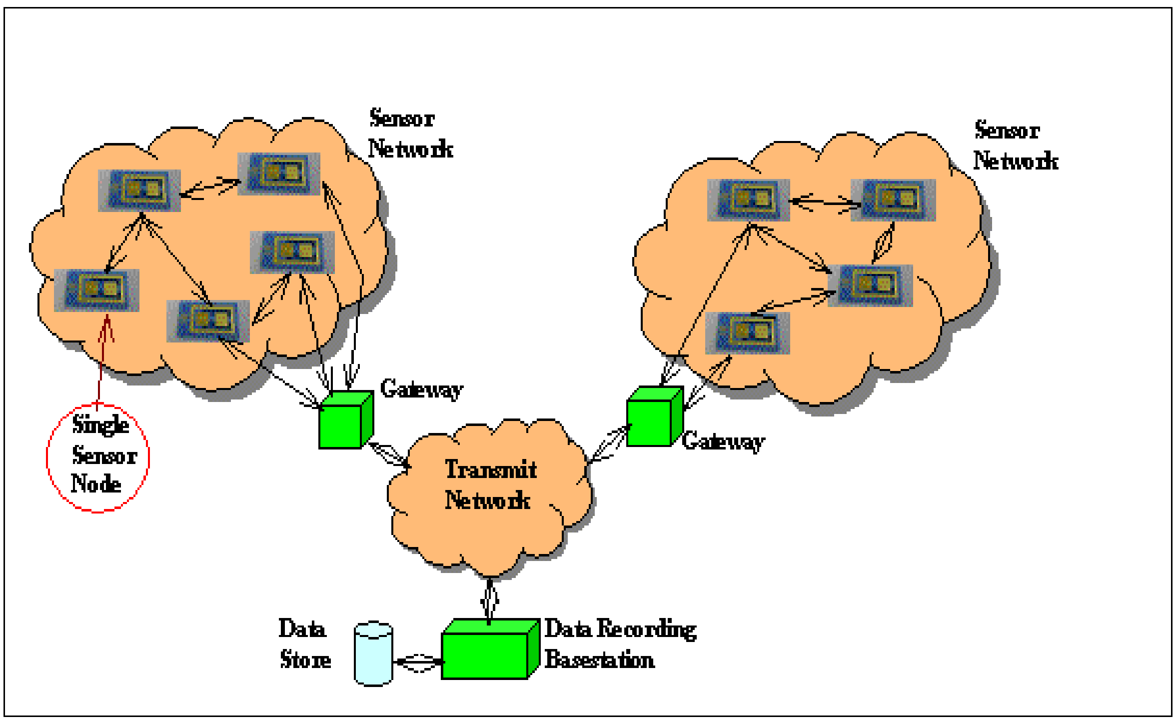


Figure 1.1. Typical structure of wireless sensor network

1.2 THROUGHPUT

In wireless sensor networks, throughput is defined as the amount of data transferred from one sensor node to another in a specified amount of time. Typically, throughput is usually measured in bits per second (bits/s or bps), and sometimes in data packets per second or data packets per time slot. In this article, the throughput is measured in data packets per time slot.

Maximum throughput is the largest amount of data volume that can be generated by the entire network. Maximum throughput routing is the routing path which could lead to maximum throughput for a sensor network. Mathematically, it can be formulated as a linear programming (LP) problem (See Chapter 2) within which the objective function is defined as the sum of rates over all the nodes in the network, and the constraints for this LP problem are: (1) flow reservation is preserved at each node, and (2) the bandwidth constraint at each node can be satisfied.

1.3 END-TO-END DELAY

The end-to-end delay refers to the total time taken for a single packet to be transmitted across a network from source to destination. It is one of the most important and fundamental issue for wireless sensor networks. Many applications of sensor networks require an end-to-end latency guarantee for time sensitive data. However, it is very difficult to bound the end-to end delay for event-driven sensor networks, where nodes produce and deliver data only when an event of interest occurs, thus generate unpredictable traffic load.

Many wireless applications require an end-to-end delay guarantee for the time-sensitive data. For example, in wireless sensor networks, it is required that sensors should collect and deliver data in a timely manner so that sensors can take timely actions. Another example is a target tracking system may require sensors to collect and propagate target information to destinations before the target leaves the surveillance area. However, the end-to-end latency is difficult to bound for event-driven sensor networks due to their unpredictable traffic pattern.

1.4 OVERVIEW OF MAIN CONTRIBUTION

This thesis concerns the optimal solution to the latency problem in multi-hop wireless sensor networks, with an objective of achieving minimum end-to-end delay through cross-layer optimization. Besides, it also shows that the shortest path routing does not necessarily lead to minimum delay due to the interferences along the path.

2. MINIMUM DELAY SCHEDULING

2.1 INTRODUCTION

With the increasing application of wireless mesh networks and sensor networks, multi-hop wireless networking technology is expected to not only provide multi-hop connectivity in locations where wired networks cannot reach, but also to support user data traffic with certain service guarantees. Throughput and delay are the two major factors of quality of service (QoS). The user-perceived data transfer speed is a combined effect of both data rate and end-to-end latency. The former of the two becomes a dominating factor for transferring a small file, and the latter one dominates the data transfer speed for transferring a large file. In a typical wireless sensor network, where small packets generated by sensors need to be periodically transmitted to the base station, delay plays a more important role.

In the past, many works have been done regarding how to maximize network throughput in multi-hop wireless networks. However, most of their solutions neglects the delay factor and leads to poor performance in the end-to-end delay. See an example in Fig 2.1, in this network topology, for a maximum throughput routing algorithm, it would choose (a) to deliver the packets since the two paths do not interfere with each other; but for a minimum delay routing algorithm, it would choose (b) for the transmission since it is the shortest path and there is no interference from other data flows.

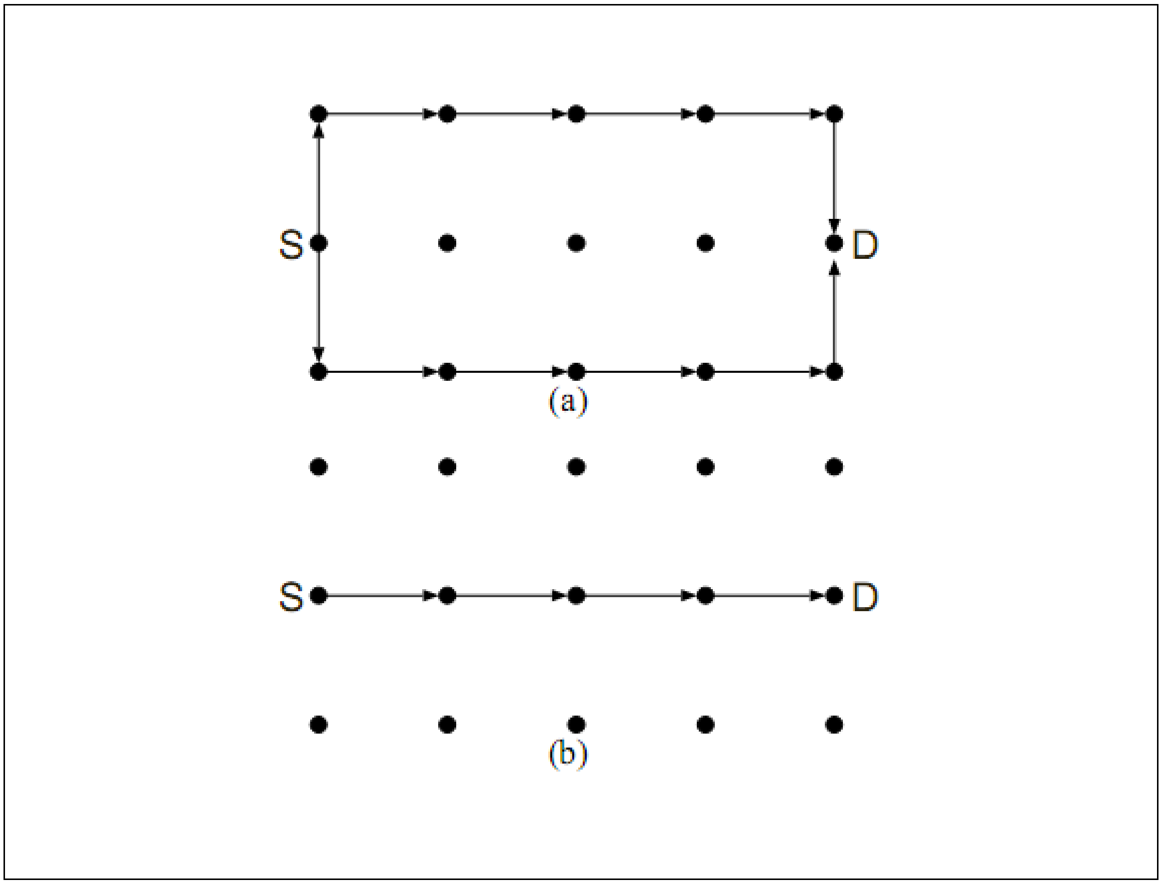
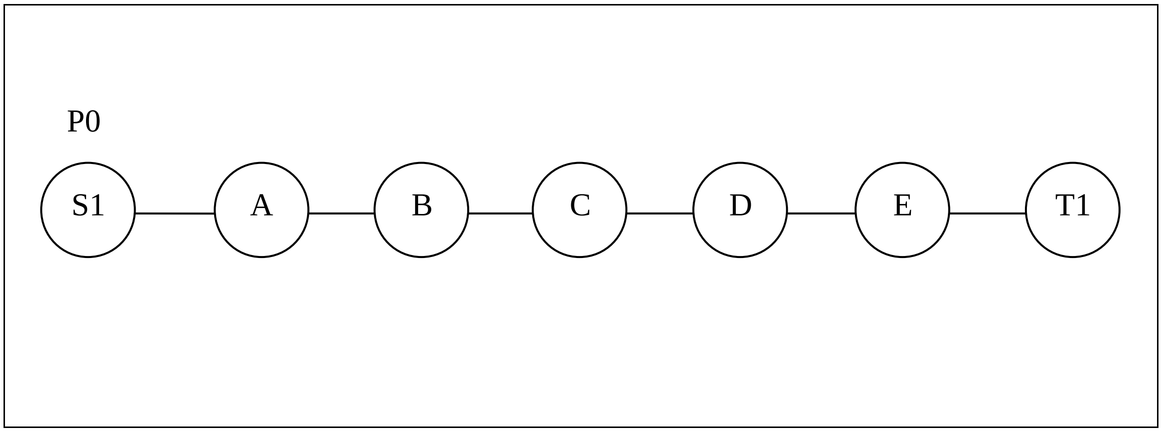


Figure 2.1. (a) With maximum throughput routing, latency is 6 slot-time; (b) With minimum delay routing, latency is 4 slot-time.

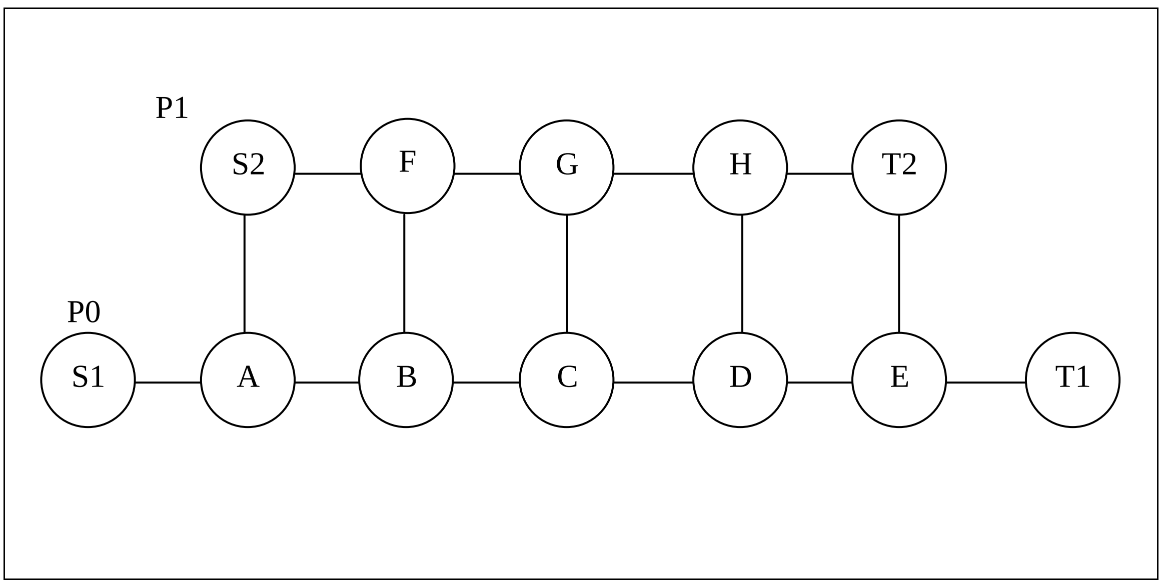
In the example shown in Figure 2.1, the shortest path happens to have the minimum delay. Actually it is a misbelieve that the shortest path always leads to the minimum delay. In fact, end-to-end delay is a result of both the number of hops on the path, and the interference level along the path. Shortest path leads to the minimum delay only if the shortest path is the least interfered path.

Interference works adversely for delay the same way it does for throughput. Figure 2.2 shows that if there is only one data flow from source S1 to destination T1, end-to-end delay is 6 slots, assuming each slot is used to transmit one packet. However, if there are other transmissions nearby, the end-to-end delay of the same flow can be jumped up to 10 slots if we do not use optimization techniques and a packet is scheduled to use the next available slot as soon as it arrives.

Table 2.1 uses the global slot table to show the packets sent by source(s) based on the topology given in Figure 2.2. In the global slot table, for each slot, only one node can be scheduled for sending packets. In Figure 2.2(a), since it's has only one flow of packets, the time sequence of the packets sent by source S1 is consecutive. The relay node forwards the packet as soon as it receives it. So, the end-to-end latency is "perfect", which is 6 slot time. But for (b), another source S2 starts sending packets just after the first S1 sends its packet to the first relay node. This relay node detects there exists another node, which is S2 is now sending the packet. Since transferring at the same time will cause collision, the relay node will store the packet for a slot time and rearrange the sending time to be the next slot time. That's why S1's packet begins to be sent at slot 3 instead of slot 2. Similarly, after S2 sends its packet to the neighbor(relay node), the relay node detects that S1's packet is being transferred at slot 3, so it also keeps the packet for one slot time and tends to send it at next slot—slot 4. Finally, it takes 6 slot times for S2 to finish the transferring and for S1, it takes 10 slot times.



(a)



(b)

Figure 2.2. (a) With a single data flow, latency is 6 slot-time; (b) When other transmitters are active, the latency becomes 10 slot-time. Numbers on links are slot numbers. There are 5 distinct slot numbers

Table 2.1. (a) Global slot table of Figure 2.2 (b) Global slot table of Figure 2.2

(a)

Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
S1→A(P0)	A→B(P0)	B→C(P0)	C→D(P0)	D→E(P0)	E→T1(P0)

(b)

Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
S1→A(P0)	S2→F(P1)	A→B(P0)	F→G(P1)	B→C(P0)	G→H(P1)

Slot 7	Slot 8	Slot 9	Slot 10
C→D(P0)	H→T2(P1)	D→E(P0)	E→T1(P0)

When there are multiple data flows in the network, it is not straight forward to find the optimal transmission schedule that leads to the minimum delay. This thesis propose a linear programming-based link scheduling scheme that computes time slot assignment such that the end-to-end delay is minimum and there are no conflicting transmissions at any time. This link scheduling scheme can work with any routing scheme.

The main contribution of this thesis is that a linear optimization model is designed to capture the impact of wireless interference on network delay in multi-hop wireless sensor networks. Compared to previous linear models, this linear model is more accurate; and compared to the exact solution, which is an NP-hard to compute, this solution is more efficient.

2.2 OPTIMIZATION MODEL AND ALGORITHM

2.2.1 Scheduling Delay. Given the routing information, the end-to-end latency can be further reduced by optimization on link scheduling delay. When a relay node forwards a packet, there is a mandatory store-and-forward delay and a link scheduling delay that is dependent on scheduling policy. Link scheduling delay is introduced when the outgoing link use s a time slot that is not immediately after the slot used by the incoming link. In Figure.2.3, if the outgoing link uses slot number v , and incoming link uses slot number u , the total delay introduced at relay node r is $d_r = v - u$ if $v > u$, or $d_r = v - u + F$ if $v < u$, where F is the total number of distinct slots in a superframe. If the schedule is conflict-free, it is guaranteed that $u \neq v$. The end-to-end delay for a path is $\sum_r d_r$. From this formula one can see that end-to-end delay is related to both the total number of hops, and the scheduling delay at each relay node. When routing information is given, the only factor that can be optimized is the scheduling delay.

2.2.2 Interference Modeling. In order to find a conflict-free schedule, it is important that all active links in the same collision domain use different slots. In another word, no two links can use the same slot if they interfere with each other. The collision domain is defined as a group of links that are mutually conflicting with each other. To list all the collision domains in a network needs to build a conflict graph first and then to find all cliques in the conflict graph. The conflict graph is built as follows: vertices are used to represent wireless links, and then add an edge between two vertices if the wireless links they represent interfere with each other.

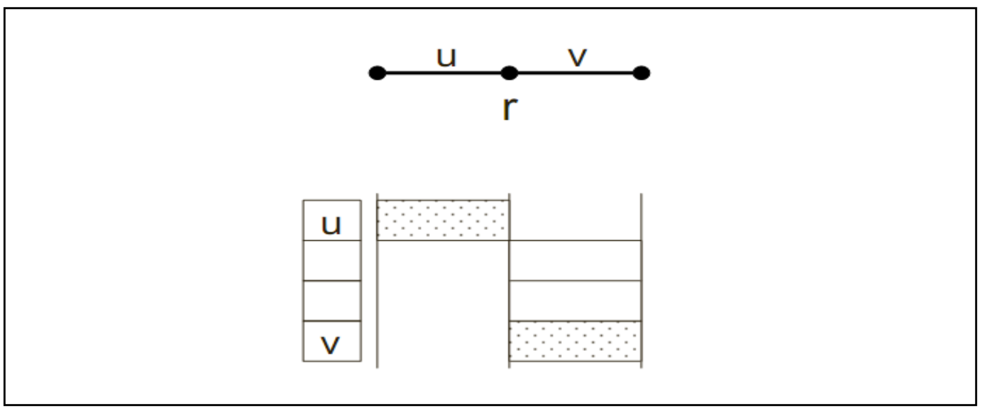


Figure 2.3. Scheduling delay at relay Node

In Figure 2.4, (a) is the example of a simple link topology. Link i, j, k are connected with each other and they are interfered with each other during the transmission. In (b), the three links are treated as nodes or vertex, i, j, k respectively. Since they are collide with each other as graph (a) shows, edges should be draw between the corresponding nodes. To build the conflict graph can be done in polynomial time, however to find all cliques in the graph is an NP-hard problem. To avoid solving an NP-hard problem, it will be better to find a sufficient set of links that includes all links in a clique and approximates the clique as closely as possible.

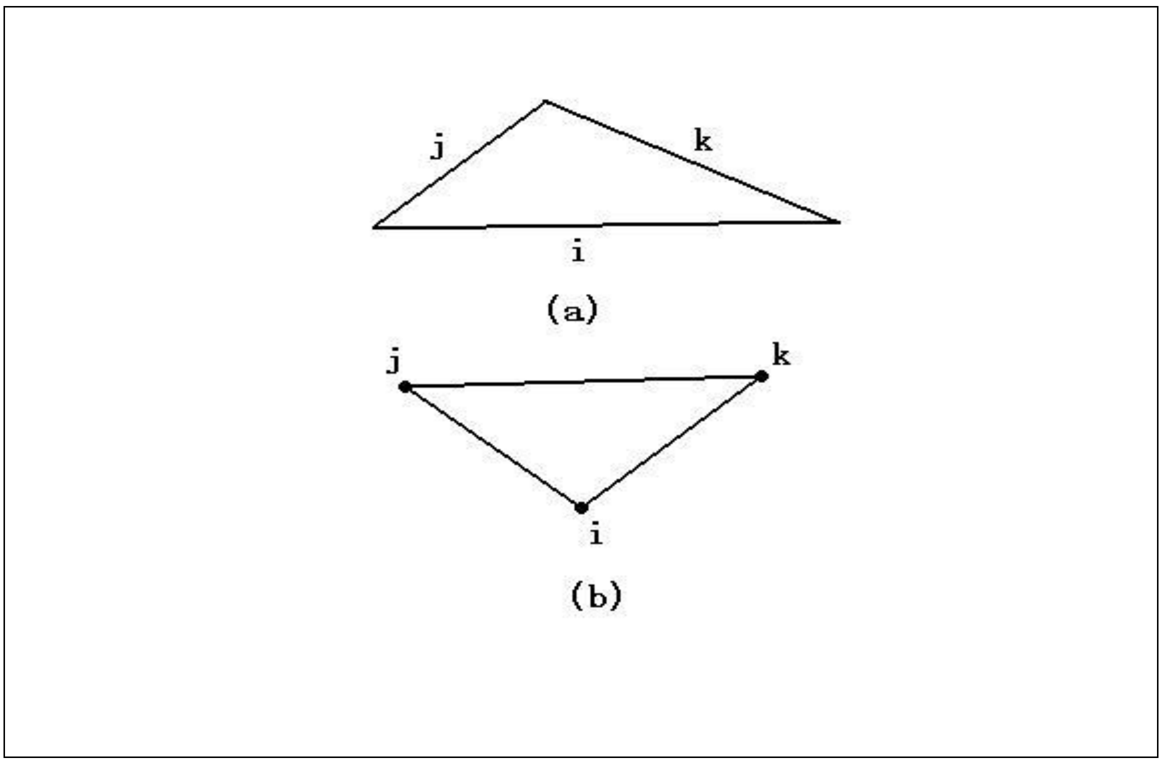


Figure 2.4. (a) Simple link topology (b) Conflict graph

Let $N2_{i,j}$ denote the group of links $\{(k,l)\}$ that satisfy: $N2_{i,j}$ = {(k, l)| link (k, l) is a two-hop neighbor of link (i, j), and its one end is one hop away from (i,j), the other end is at most 3 hops away from (i,j) via a different path} (See Table 2.2). If there is no other path, the distance is counted as ∞.

For example, in Figure 2.5. (a), link (k1, l1) and (k2, l2) belong to $N2_{i,j}$, but (k2, l1) does not, because (k2, l1) is not a 2-hop neighbor of link (i, j); in Figure 2.5. (b), link (k, l) does not belong to N2ij , since there is only one path to reach link (i, j) from k and l; the distance from k to (i, j) is 1 and the distance from l to (i, j) is ∞. In this case, the mutual conflicting relation among (i, j), (j, k), and (k, l) is captured when we apply the capacity constraint on link (j, k): we make sure the data rate satisfy $r_{j,k} + r_{i,j} + r_{k,l} \leq B$.

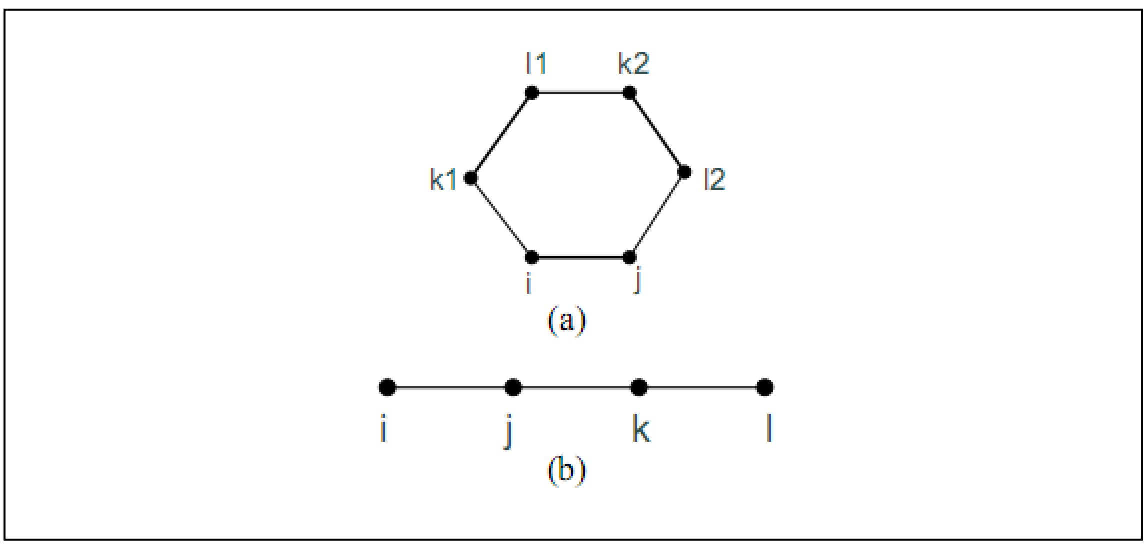


Figure 2.5. Capacity constraint

Table 2.2. Definition of $N2_{i,j}$

$d(j,k)=1, \ 1\le d(i,l)\le 3, \ d(j,k)+d(i,l)\le 4,$
or
$d(i,k)=1, \ 1\le d(j,l)\le 3, \ d(i,k)+d(j,l)\le 4$

In Table 2.2, where $k \ne i, j$, and $l \ne i, j$; $d(u,v)$ is the number of hops between node u and node v.

The collision domain CD_{ij} of link (i, j) includes:

1) Link (i, j) itself, and
2) All adjacent links of link (i, j), and
3) All two-hop links of (i, j) defined in $N2_{i,j}$.

In the simulation, the mathematical model of collision domain of link (i, j) is defined in this way, see Table 2.3.

Table 2.3. Mathematical model of collision domain of link (i, j)

$$L_{i,j} + \sum_{l \in N_i, l \neq j} L_{i,j} + \sum_{k \in N_j, k \neq i} L_{j,k} + \sum_{(k,l) \in N2_{i,j}} L_{k,l} \leq B;$$

$L_{i,j}$ is defined as: $L_{i,j} = \sum_s l_{i,j,s} + \sum_s l_{j,i,s}$

2.2.3. An ILP Model for Minimum Delay Link Scheduling. To achieve minimum scheduling delay, I first formulate it as an optimization problem. Since the routing information is given, I use $link_{l,s} = 1$ to indicate link l is on the path for flow s. What needed to be solved is the slot assignment for links. Here a 0-1 variable is introduced-- $sl_{l,f}$ for slot assignment. $sl_{l,f} = 1$ indicates link l uses slot f. If a link l is shared by multiple data flows, only one flow can use the slot f on the same link. $sl_{l,s,f} = 1$ indicates link l uses slot f for sending data which generated by source s.

Assume for source s, relay node r is on the routing path P_s. Relay node r receives flow from link m and forwards it to link n, the total delay at relay node r is $d_{r,s} = f_n - f_m + x \cdot F$, where f_n is the slot number for link n and f_m is the slot number for link m. Each slot time is equivalent to one standard packet transmission time. x is a Boolean variable, $x = 1$ when $f_n < f_m$.

The integer linear programming model is now formulated as follows:

Table 2.4. Mathematical model for minimum delay link scheduling

Objective:

To minimize maximum delay:

Minimize $\max_{s} \sum_{r \in P_s} d_{r,s}$ (1)

Or

To minimize total delay:

Minimize $\sum_{s} \sum_{r \in P_s} d_{r,s}$ (2)

Subject to:

$$\sum_{l' \in CD_l} sl_{l',f} \leq 1, \quad \forall l, f \tag{3a}$$

$$sl_{l,f} = \sum_{s} sl_{l,s,f}, \quad \forall l, f \tag{3b}$$

$$\sum_{f=1}^{F} sl_{l,s,f} = link_{l,s} \cdot R_s, \ \forall l, s \tag{3c}$$

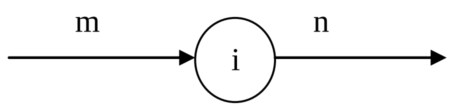
$$d_{r,s} = \sum_{f=1}^{F} sl_{n,s,f} \cdot f - \sum_{f=1}^{F} sl_{m,s,f} \cdot f + x_{r,s} \cdot F, \quad \forall r, s \tag{3d}$$

$$0 < d_{r,s} < F, \quad x_{r,s} = \{0,1\}, \ \forall r, s, \ sl_{l,f} = \{0,1\} \tag{3e}$$

$$sl_{l,f} = \{0,1\}, \quad sl_{l,s,f} = \{0,1\} \tag{3f}$$

The objective function of the above link scheduling scheme is to minimize the total delay at each relay node r. Formula (3d) shows the way to calculate the latency for each relay node. Figure 2.6 shows an example of computing the total delay for the data flow when passing relay node i. Suppose there are three data flows pass through the relay node i and the three data flows are sent from source S1, S2, S3 respectively. m and n are the only incoming and outgoing links carrying the entire data flows. For incoming link m, assume the slots assigned to each source is slot 10 for source S1, slot 15 for source S2, slot 22 for source S3; for outgoing link n, the slots assigned to each source is slot 9 for source S1, slot 18 for source S2, slot 20 for S3. By using formula (3d) in Table 2.4 one is able to calculate the delay for each data flow at relay node i. For the data flow sent from source S1, its delay at node i $d_{i,S1} = 9 - 10 + 23 = 22$; similarly, for the data flow sent from source S2, $d_{i,S2} = 18 - 15 = 3$; for the one sent from source S3, $d_{i,S3} = 20 - 22 + 23 = 21$. So, the total delay of the entire data flows at relay node i is $d_i = \sum_S d_{i,S} = 22 + 3 + 21 = 46$.

In Table 2.4 (3c), R_s is the data rate of source s, given as input. Although our purpose is only to minimize the end-to-end delay of a single packet regardless of the source data rate, the model is general enough to consider sources with different data rates. In simulation, I set $R_s = 1$ for all sources.



Data flow	In	Out
S1	10	9
S2	15	18
S3	22	20

Figure 2.6. Computing total delay at relay node i

2.2.4. Computing the Slot Assignment. To solve the above integer linear programming problem is NP-hard. In order to avoiding solving a NP-hard problem, it should be first relaxed to a linear programming problem, then use maximum likelihood rounding to map real numbers to integer slot numbers.

Find the optimal solution for the LP problem with slot numbers relaxed to real numbers. Table 2.5. below shows the steps for slot round up.

Table 2.5. Slot round up

(1) Sort slot $sl_{l,f}$ in non-increasing order, set $Th = 0.5$.
(2) For each non-zero variable $sl_{l,f}$, if $sl_{l,f} \geq Th$, assign $sl_{l,f} = 1$. Assign $sl_{l',f} = 0$ for other links l' that are conflicting with l. Assign remaining values appropriately to satisfy flow conservation; if Th > the largest $sl_{l,f}$, set Th = the largest $sl_{l,f}$.
(3) Repeat step (2) until all variables are rounded to integers.

2.3. PERFORMANCE EVALUATION

In this section, minimum-delay link scheduling algorithm (MinDelay) will be compared to First-Come-First-Serve (FCFS) scheduling policy when routing information is given. Choices for routing are shortest path routing and maximum throughput routing. The simulation result shows that the proposed time slot assignment algorithm can significantly reduce scheduling delay. Besides, it also shows that the shortest path does not always lead to the least latency.

In the simulation study, I use 50 nodes deployed on a 150×150 square region, with node transmission range 30. 10 out of the 50 nodes are randomly selected as the source nodes, and all source nodes transmit to a common receiver (sink node).

It assumes that routing information is given and I compare the end-to-end latency achieved by using FCFS with the one achieved by MinDelay. Each source node generates a packet and I observe the end-to-end latency of the single packet.

2.3.1. First-Come-First-Serve (FCFS) Algorithm. In FCFS, the packet arrival order is random. A relay node schedules a packet as soon as it arrives; when deciding which slot to use, a relay node chooses the next available slot to transmit the packet if it does not conflict with other transmissions. FCFS is one of the most commonly used scheduling policies in practice.

Figure 2.8 is an example for a network topology and Table 2.8 gives a global slot table for the topology by using FCFS algorithm.

In this example, 5 nodes $s1$, $s2$, $s3$, $s4$, $s5$ are going to send their packets $p1$, $p2$, $p3$, $p4$, $p5$ to one sink node t respectively. $R1$, $R2$, $R3$, $R4$, $R5$, $R6$ are the 6 relay nodes along the path. It assumes that the initial order for sending packets is $p1$, $p2$, $p3$, $p4$, $p5$. At beginning, source node $s1$ tends to transmit packet $p1$ to relay node $R1$. It checks the global slot table, since node $s1$ is the first one to send packet, no collision will occur. So, node $s1$ uses slot 1 to transmit packet $p1$ to $R1$. For source node $s2$, it also checks the slot table before start sending and detects that node $s1$ has used slot 1, since node $s1$ and node $s2$ share the same relay node $R1$, they will collide with each other if transmitting packets at same slot.

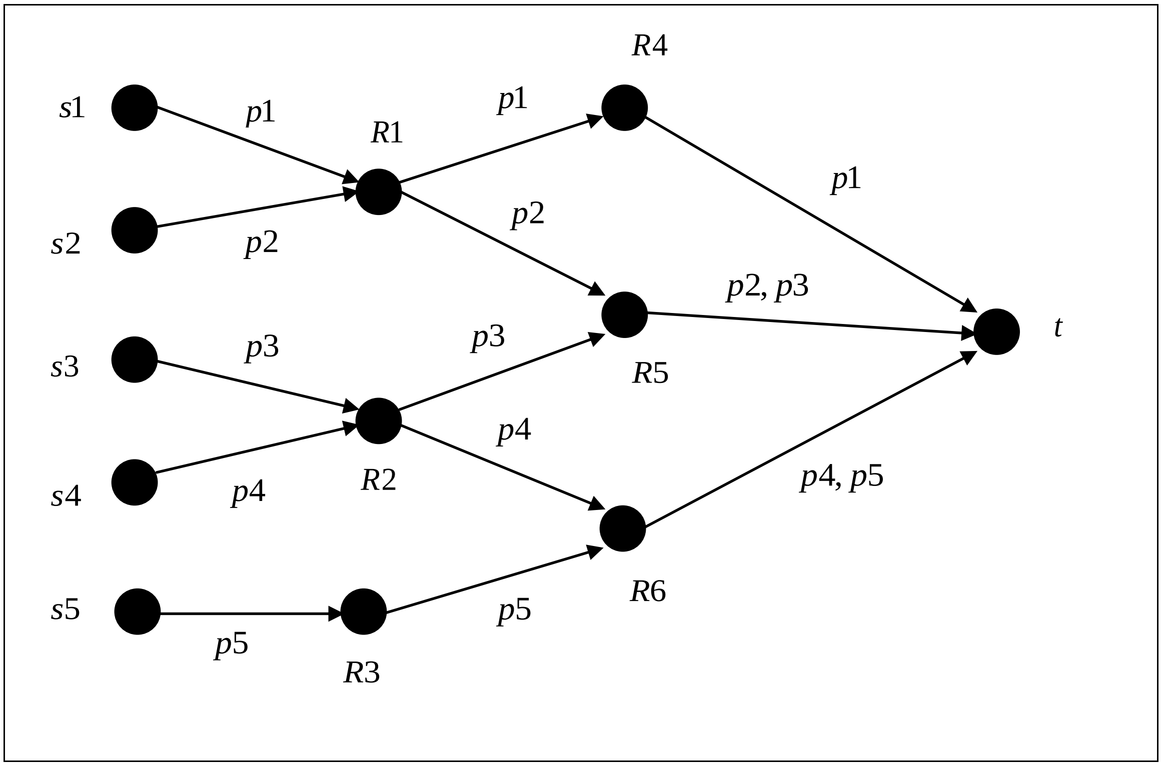


Figure 2.7. An example of sensor network topology

Table 2.6. Global slot table of the example by using FCFS algorithm

Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7	Slot 8	Slot 9
s1→R1 (p1)	s2→R1 (p2)	R1→R4 (p1)	R1→R5 (p2)	R2→R5 (p3)	R5→t (p2)	R5→t (p3)	R6→t (p4)	R6→t (p5)
s3→R2 (p3)	s4→R2 (p4)	R2→R6 (p4)		R4→t (p1)				
s5→R3 (p5)	R3→R6 (p5)							

In this case, source node $s2$ will store the packet $p2$ for a slot time and tends to send it again at next slot time. Then node $s3$ has the turn for sending packet, although slot 1 has been occupied by node $s1$, node $s3$ does not collide with $s1$ since they are in the different collision domain. Thus node $s3$ can also use slot 1 for sending packet. Similarly, source node $s4$ will not send packet since it collide with $s3$ and $s5$ can also send at slot 1. After node $s5$ finishes sending, node $s2$ can use slot 2 for transmission. For node $s4$, since it does not collide with node $s2$, it can also use slot 2. Finally, the arriving order for all packets at sink node t is $p1$, $p2$, $p3$, $p4$, $p5$.

2.3.2. An ILP Model for Maximal Throughput Routing. In the simulation, the maximal throughput routing is used in order to make comparison with the shortest path routing. The integer linear programming model is now formulated as follows:

In Table 2.7, Formula (3a) first defines $l_{i,j,s} = \{0,1\}$ is a 0, 1 variable which indicates this link has been chosen or not. In this formula, l is short for $link\ \{i, j\}$ indicates for a link or two nodes i, j which are connected with each other. And s is the notation of source node. Formula (3b) means that there is only one flow from the source node to its neighbors, in other words, for each source s, there will be only one data flow routed to the sink node or destination. The objective of formula (3b) is to eliminate the condition of splitting flow. In formula (3c), t represents sink node, this formula means for each source s, there must be one data flow routed to the sink t. Formula (3d) defines that for each relay node (which is neither source nor sink node) i, the total amount of all the incoming data flows must be equal to the total amount of all the outgoing data flows, as shown in Figure 2.8 (a). Similarly, for those source nodes who maybe acted as relay nodes, as presented in formula (3e), the amount of its incoming data flows must equals to the amount of its outgoing data flows minus its own outgoing flows since node i not only has the responsibility of delivering the packets which come from other sources but also itself is act as a source node, see Figure 2.8 (b).

Table 2.7. Mathematical model for maximum throughput routing

Objective:

To maximize source rate R_s:

$$\max : R_s \qquad (1)$$

Or to minimize q:

$$\min : q \qquad (2)$$

Subject to:

i. Flow conservation define in link $l_{i,j,s}$:

$$l_{i,j,s} = \{0,1\}, \quad \forall r, s \qquad (3a)$$

Table 2.7. (Continued)

$$\sum_{j\in N_s} l_{s,j,s} = 1, \quad \forall s \qquad (3b)$$

$$\sum_{i\in N_t} l_{i,t,s} = 1, \quad \forall s \qquad (3c)$$

$$\sum_{j\in N_i} (l_{i,j,s} - l_{j,i,s}) = 0 \quad \forall s, \quad \forall i \notin \{s,t\} \qquad (3d)$$

$$\sum_{j\in N_i} (l_{i,j,s} - l_{j,i,s}) = 0 \quad \forall s, \quad \forall i \in \{s,t\}, \ i \neq s \qquad (3e)$$

$$l_{j,s,s} = 0, \quad j \in N_s, \quad \forall s \qquad (3f)$$

$$0 \le l_{i,j,s} + l_{j,i,s} \le 1, \quad j \in N_s, \quad \forall s \qquad (3g)$$

ii. Bandwidth constraint:

$$\sum_{C.D.} R_{i,j} \le F \Rightarrow$$

$$\sum_{C.D.} R_{i,j,s} \le F \Rightarrow$$

$$\sum_{C.D.} \sum_{s} l_{i,j,s} \cdot R_s \le F \Rightarrow$$

$$\sum_{C.D.} \sum_{s} l_{i,j,s} \le \frac{F}{R_s} = q \cdot F \text{, let } q = \frac{1}{R_s} \qquad (4)$$

In Table 2.7, (ii) is the mathematical model for bandwidth constraint. *C.D.* represents collision domain, and R_s is data rate of source s, also assume that R_s are the same for all the sources. The bandwidth constraints confirms that the bandwidth of all links which belong to the same collision domain should be bounded by $\frac{F}{R_s}$. F is the total slot number for each node, which is defined in this way: $F = \frac{bandwidth}{packet\_size}$, the bandwidth we choose is 802.11b standard which equals to 11Mpbs, and the size of each packet is assumed to be 500 Kb. So slot number $F = \frac{11Mbps \cdot 1024}{500Kb} \approx 23$.

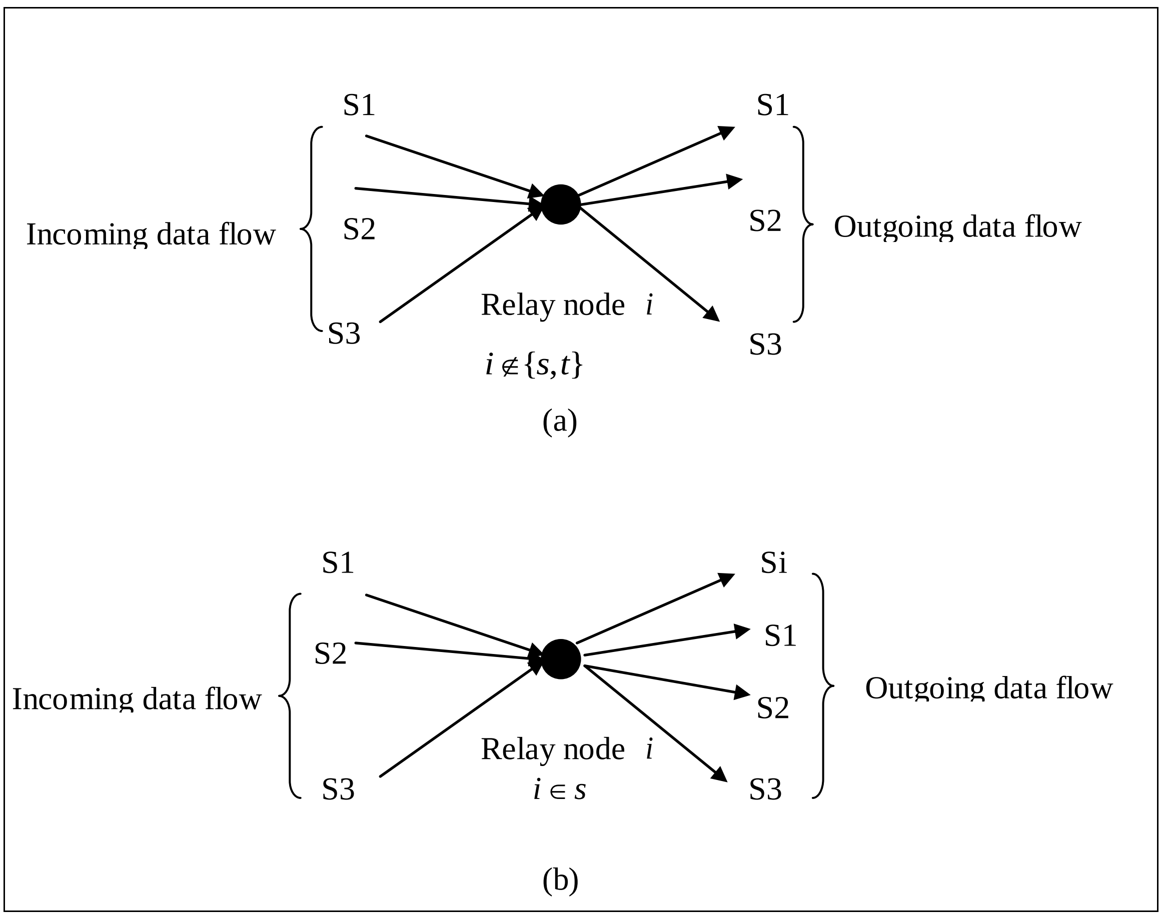


Figure 2.8. (a) Incoming and outgoing data flows of relay node i, $i \notin \{s,t\}$

(b) Incoming and outgoing data flows of relay node i, $i \in s$

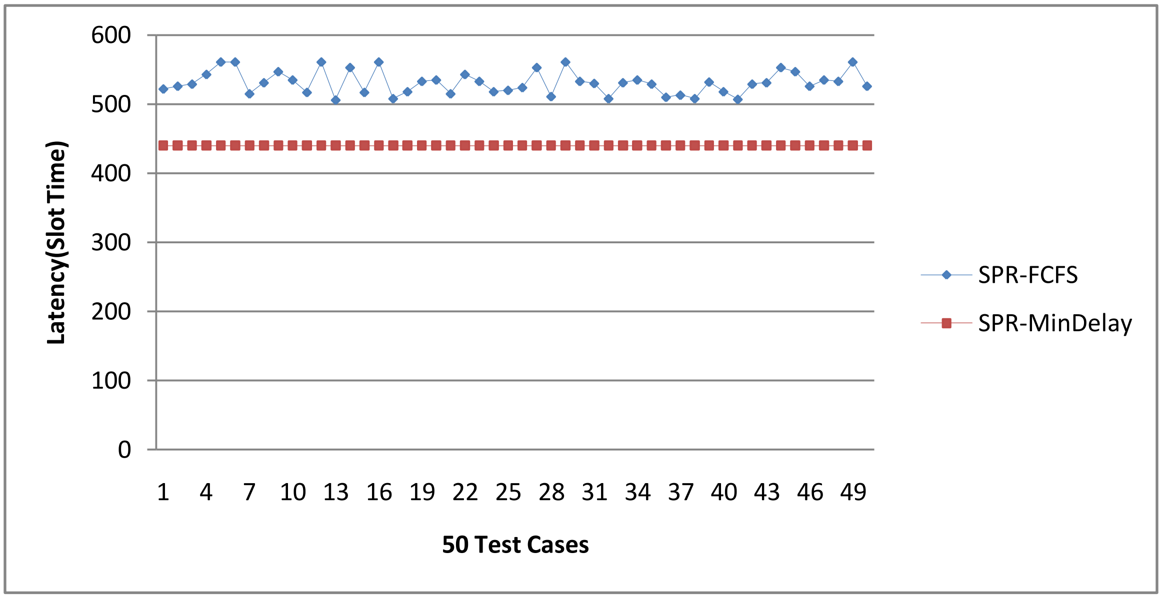
The results of the ILP formulation may not be integer, which means the data flow which belongs to specific source s could be split to multiple data flows. In order to solve this problem, the ILP results must be rounded up to integers. Table 2.8 shows the way to round up link $l_{i,j,s}$:

Table 2.8. Link round up

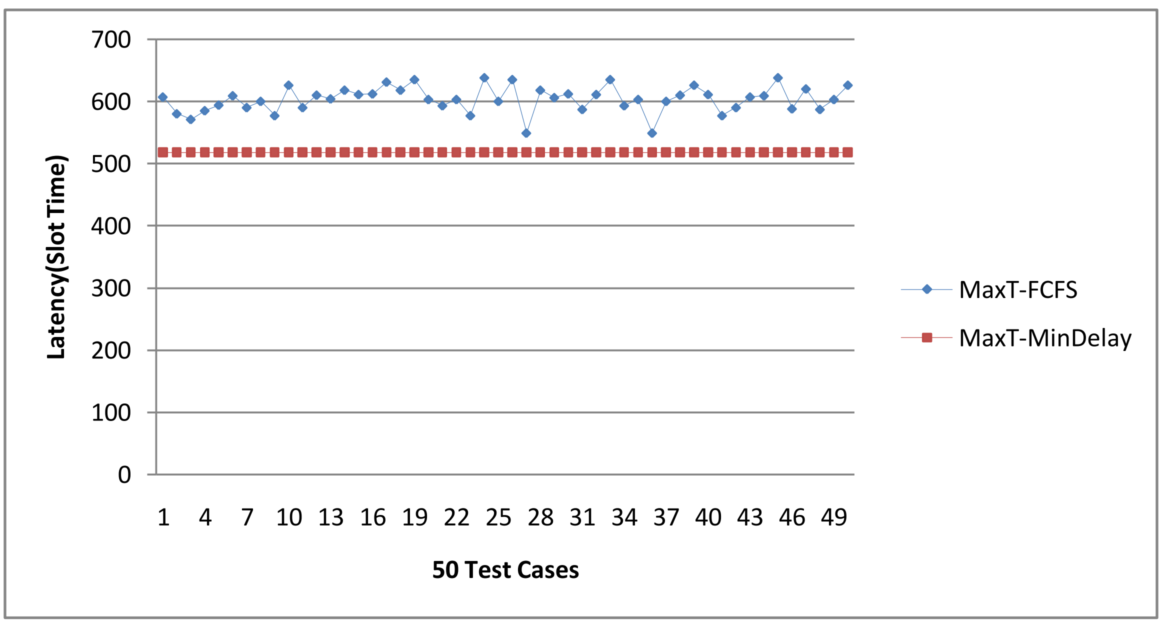
(1) We start from every source node s_i, choose the link which has maximum value/result. (2) Then we choose the link which has the maximum value from all adjacent links of the link we get from step 1. At the same time, we must also check all the links which have been selected must satisfy the bandwidth constraints, as stated in Table 2.7. If the constraints are voided after one link has been chosen, then we drop this link and pick up another link which has the second greatest value. (3) Repeat step (2) until the links belongs to one source can be connected to become a complete data flow. In other words, from source s_i, we have a path to get sink node t. (4) Then repeat step (1) until all the sources have a complete path that can be routed to the sink t.

2.3.3. Simulation Result. Back to the simulation, since the packet arrival order is an important factor to FCFS, 50 cases have been tested on 50 random arrival orders for each given network topology. The collected data are for minimizing total delay, given by optimization objective (2) (See Table 2.4).

I compared MinDelay with FCFS when they are used with shortest path routing (SPR) and maximum throughput routing (MaxT, presented in Table 2.7). Simulation results show that when using SPR, MinDelay outperforms FCFS by 15% to 27.5% in total delay; and when using the MaxT routing, MinDelay outperforms FCFS by 11% to 23.5%. From this simulation I observed MinDelay has shorter latency than FCFS in all scenarios regardless of what routing algorithm is used, but the shortest path routing does not necessarily always have smaller latency than other routing algorithms. This is because the shortest path routing may lead to too much collision, thus results in large end-to end delay; but for the maximal throughput routing, the collision might be reduces since paths are longer, thus data flows from different sources could go more "smoothly" instead of being blocked by other flows. In Figure 2.9, the delay for shortest path is smaller that the delay for maximum throughput routing. Figure 2.10 shows that for a different network topology, delay for shortest path routing may be worse than delay for maximum throughput routing.



(a)



(b)

Figure 2.9. (a) MinDelay vs FCFS by using shortest path routing (b) MinDelay vs FCFS by using maximum throughput routing

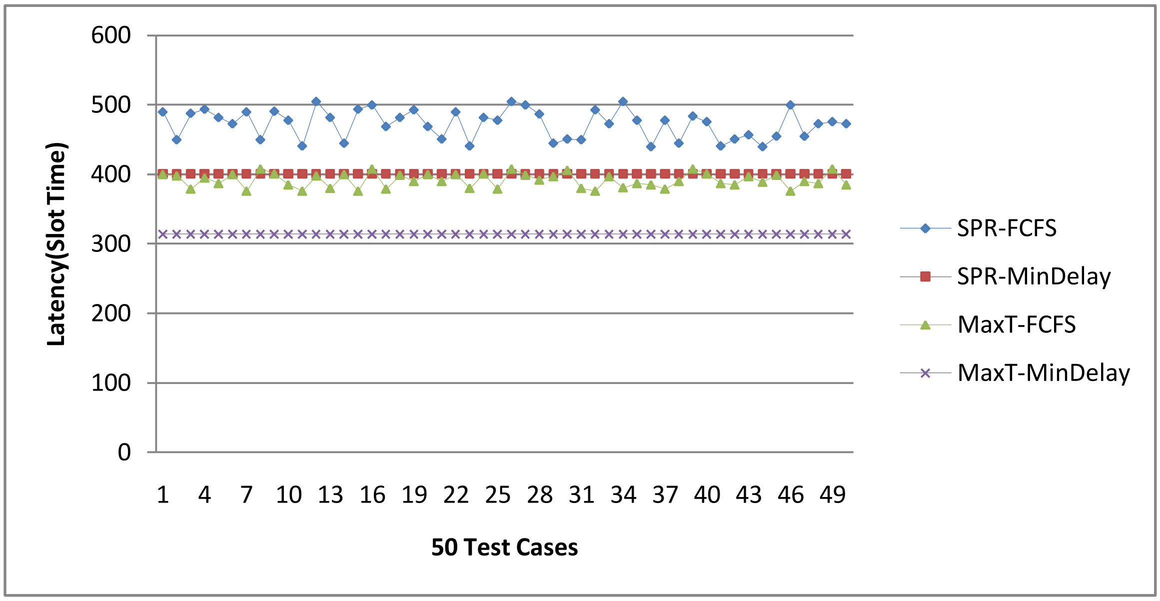


Figure 2.10. In a different network setup, shortest path routing has larger latency.

3. CROSS-LAYER OPTIMIZATION FOR MINIMUM DELAY

The minimum delay link scheduling described in chapter 2 is operated at MAC layer. It computes the slot assignment based on the giving data flows. In this chapter I'll introduce a method which could directly addresses routing (Network layer) and slot assignment (MAC layer) just based on the node topology without knowing the data flows. Although we could also use the MinDelay algorithm to compute the minimum end-to-end delay after deciding the data flow at first step, the defect is it will use 'roundup' two times (one is link roundup after sovling ILP model for routing; the other is slot roundup after solving ILP model for slot), and leads the final result more distracted from the optimal solution. Because of this flaw I improved the model and modified it by combining the two steps into one, I call it One-Phase (OP) algorithm, which is defined as for a giving node topology, it can determine the data flow and slot assignment at the same time. The good point is it uses roundup only once and makes the result much closer to the optimal solution, the trade off is it costs more time to be solved. In section 3.1 I'll show the mathematical model of Minimum Interference (MI) routing + MinDelay algorithm. In section 3.2 I'll present the ILP model of One-Phase algorithm and the comparison result of One-Phase algorithm with Minimum Interference (MI) routing + MinDelay algorithm will be shown in section 3.3.

3.1. TWO-PHASE ALGORITHM: MINIMUM INTERFERENCE ROUTING WITH MINIMUM DELAY SCHEDULING

The minimum interference (MI) routing is such a routing scheme that tends to choose the path which has the least total interference. In this chapter I'll use the minimum delay scheduling to solve the slot assignment when MI routing is used. I referred this as two-phase algorithm since it can be divided by two basic steps: fist step is link assignment and second is slot assignment. Below is the ILP formula for minimum interference routing.

Table 3.1. Mathematical model for minimum interference routing

Objective:

To minimize total interference:

$$\min : \sum_{i,j} M_{i,j} \qquad (1)$$

Subject to:

i. Flow conservation:

$$R_{i,j,s} = \{0,1\}, \ \forall i, j, s \qquad (2a)$$

$$\sum_{j \in N_s} R_{s,j,s} = R_s , \quad \forall s \qquad (2b)$$

$$\sum_{i \in N_t} R_{i,t,s} = R_s , \quad \forall s \qquad (2c)$$

Table 3.1. (Continued)

$$\sum_{j \in N_i} (R_{i,j,s} - R_{j,i,s}) = 0 \quad \forall s, \quad \forall i \notin \{s,t\} \qquad (2d)$$

$$R_{j,s,s} = 0, \quad j \in N_s, \qquad \forall s \qquad (2e)$$

$$0 \le R_{i,j,s} + R_{j,i,s} \le 1, \quad j \in N_i, \quad \forall i,s \qquad (2f)$$

ii. Bandwidth constraint:

$$r_{i,j} + \sum_{l \in N_i, l \neq j} r_{i,j} + \sum_{k \in N_j, k \neq i} r_{j,k} + \sum_{k,l \in N2_{i,j}} r_{k,l} = M_{i,j}, \quad \forall link(i,j) \qquad (3a)$$

$$r_{i,j} = \sum_s R_{i,j,s} + \sum_s R_{j,i,s} \qquad (3b)$$

In Table 3.1, it is assumed that the throughput from each source is 1 ($R_s = 1$). In order to calculate the minimum delay for minimum interference, the following steps should be followed:

(1) Round up the link $l_{i,j,s}$ which solved by the LP program.(See Table 2.5)

(2) Use the ILP model for minimum delay scheduling to assign the slot based on the topology which got from step (1). (See Table 2.6)

(3) Compute the slot assignment based the result which got from step (2). (See table 2.7)

By using minimum interference routing we are able to find the path which has least collision, and guides the packets transferring with minimum end-to-end latency. Figure 3.1 shows a example which uses MI routing scheme and MaxT routing scheme. Table 3.2 shows the comparison of the two.

In this Figure 3.1, suppose node A and node G both want to send one packet P0, P1 respectively to node F. By using minimum interference routing (dashed line), the total interference we get is 3 + 3 + 4 + 2 = 12, after applying the MinDelay algorithm we can get the end-to-end delay is 41; if using the maximum throughput routing (the paths in solid line), the total interference is 2 + 2 + 3 + 3 + 3 + 3 = 16 and total end-to-end delay we get is 71. So, we can see there is the big benefit for using MI+MinDelay when compared with MaxT+MinDelay. Table 3.2 shows the comparison of the two.

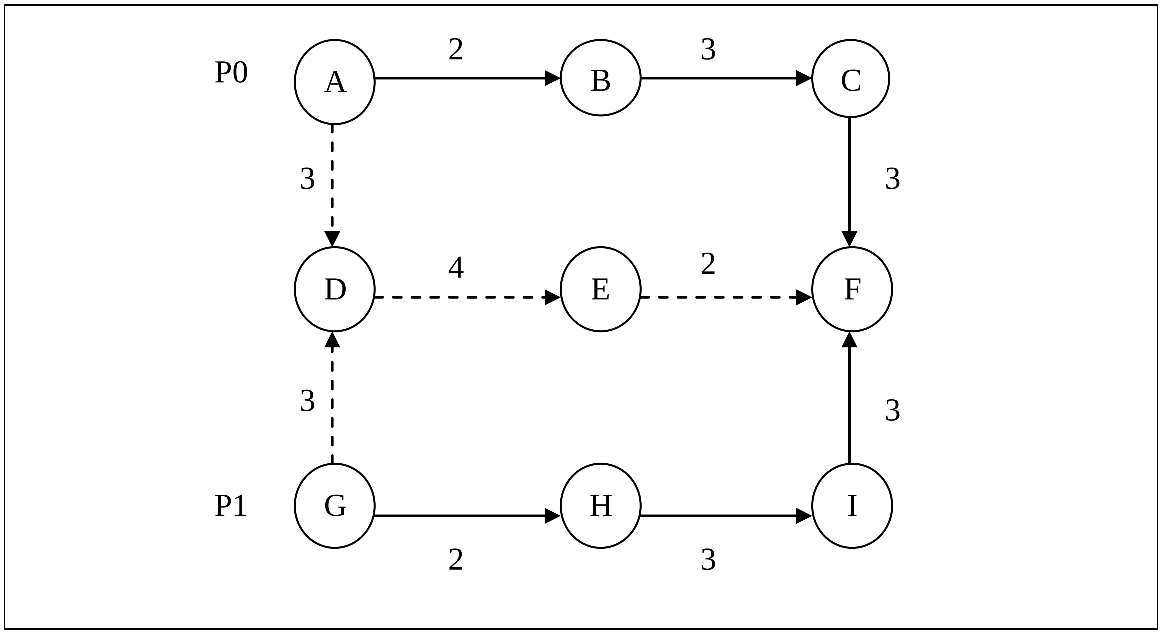


Figure 3.1. Example: MI routing vs MaxT routing

Table 3.2. MI routing vs MaxT routing

	Total interference	End-to-end delay
MI routing	12	41
MaxT routing	16	71

3.2. JOINT ROUTING AND SCHEDULING

Table 3.3 is the mathematical model for One-Phase algorithm. It is assumed that the throughput from each source is 1 ($R_s = 1$). The objective function of the above ILP model is to minimize parameter α to make sure the delay of each routing path is within α -factor of the minimum delay path. In inequality (4e), $D_{s,\min}$ is the minimum end-to-end delay of flow s . It is the total time it takes for one unit of data to travel from source to the sink.

Table 3.3. Mathematical model for One-Phase algorithm

Objective:

To minimize α :

$$\min : \alpha \qquad (1)$$

Subject to:

iii. Flow conservation:

$$R_{i,j,s} = \{0,1\}, \ \forall i, j, s \qquad (2a)$$

$$\sum_{j \in N_s} R_{s,j,s} = R_s , \quad \forall s \qquad (2b)$$

$$\sum_{i \in N_t} R_{i,t,s} = R_s , \quad \forall s \qquad (2c)$$

$$\sum_{j \in N_i} (R_{i,j,s} - R_{j,i,s}) = 0 \quad \forall s , \quad \forall i \notin \{s,t\} \qquad (2d)$$

Table 3.3. (Continued)

$$\sum_{j \in N_i} (R_{i,j,s} - R_{j,i,s}) = 0 \quad \forall s\,, \quad \forall i \in \{s,t\}\,, \quad i \neq s \qquad (2e)$$

$$R_{j,s,s} = 0\,, \quad j \in N_s\,, \qquad \forall s \qquad (2f)$$

$$0 \leq R_{i,j,s} + R_{j,i,s} \leq 1\,, \quad j \in N_i\,, \quad \forall s \qquad (2g)$$

iv. Bandwidth constraint:

$$r_{i,j} + \sum_{l \in N_i, l \neq j} r_{i,j} + \sum_{k \in N_j, k \neq i} r_{j,k} + \sum_{k,l \in N2_{i,j}} r_{k,l} \leq B\,, \quad \forall link(i,j) \qquad (3a)$$

$$r_{i,j} = \sum_s R_{i,j,s} + \sum_s R_{j,i,s} \qquad (3b)$$

v. Delay constraint:

$$\sum_{l' \in CD_l} sl_{l',f} \leq 1, \qquad \forall l, f \qquad (4a)$$

$$sl_{l,f} = \sum_s sl_{l,s,f}\,, \quad \forall l, f \qquad (4b)$$

$$\sum_{f=1}^{F} sl_{l,s,f} = R_{l,s}\,, \quad \forall l, s \qquad (4c)$$

$$d_{r,s} = \sum_{f=1}^{F} sl_{n,s,f} \cdot f - \sum_{f=1}^{F} sl_{m,s,f} \cdot f + x_{r,s} \cdot F, \quad \forall r \notin \{s,t\}, s \qquad (4d)$$

$$\sum_r d_{r,s} \leq \alpha \cdot D_{s,\min} \cdot R_s, \quad \forall s \qquad (4e)$$

$$0 < d_{r,s} < F, \quad x_{r,s} = \{0,1\}, \quad \forall r, s\,, \quad sl_{l,f} = \{0,1\} \qquad (4f)$$

$$sl_{l,f} = \{0,1\}, \quad sl_{l,s,f} = \{0,1\} \qquad (4g)$$

The model above has integer variables, the ‘slot roundup’ method (See Table 2.7) is used to find the integer solutions.

3.3. PERFORMANCE EVALUATION

I compare the delay of minimum interference routing with the delay of one-phase algorithm. Since the OP algorithm works very slow in large network topology, I shrink the topology into a much smaller one: 10 nodes randomly deployed in a 65×65 area, 3 sources have been chosen to transfer packet to one common sink. Each source is guaranteed to have at least three hops from the sink node. The simulation result shows the one-phase algorithm outperforms the MI+MinDelay by 4% to 14%. (See Figure 3.2). I will also show that the time for solving OP is much longer than the time for solving MI+MinDelay. Figure 3.3 shows that the time for solving OP algorithm exceeds the time for solving MI+MinDelay by 7.8 to 10.4 times.

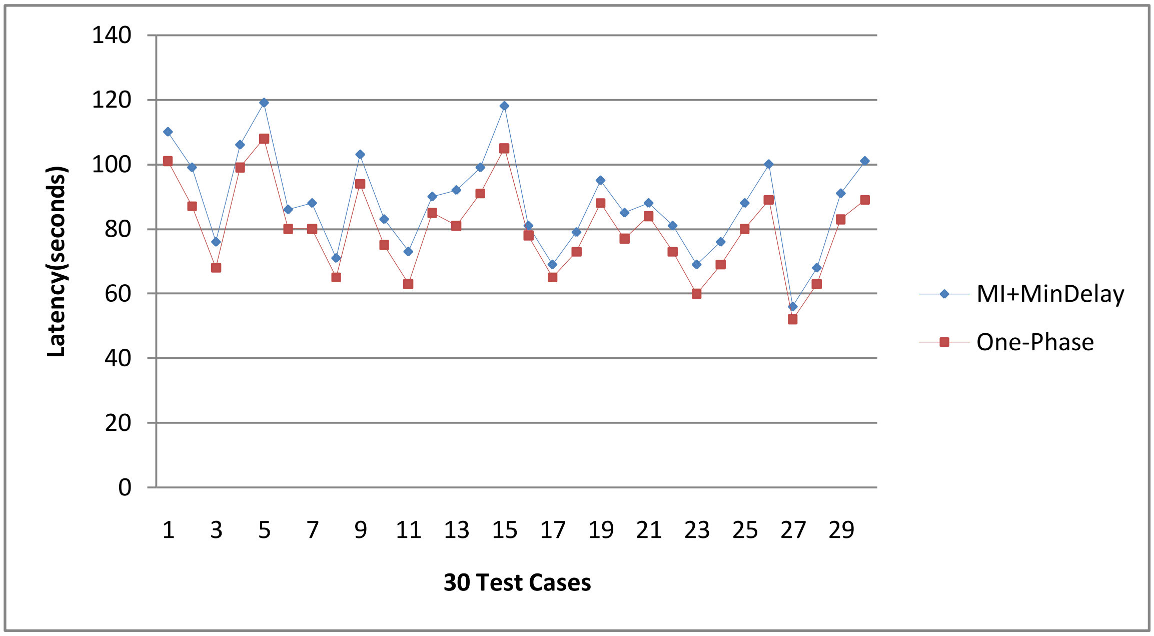


Figure 3.2. MI+MinDelay vs OP

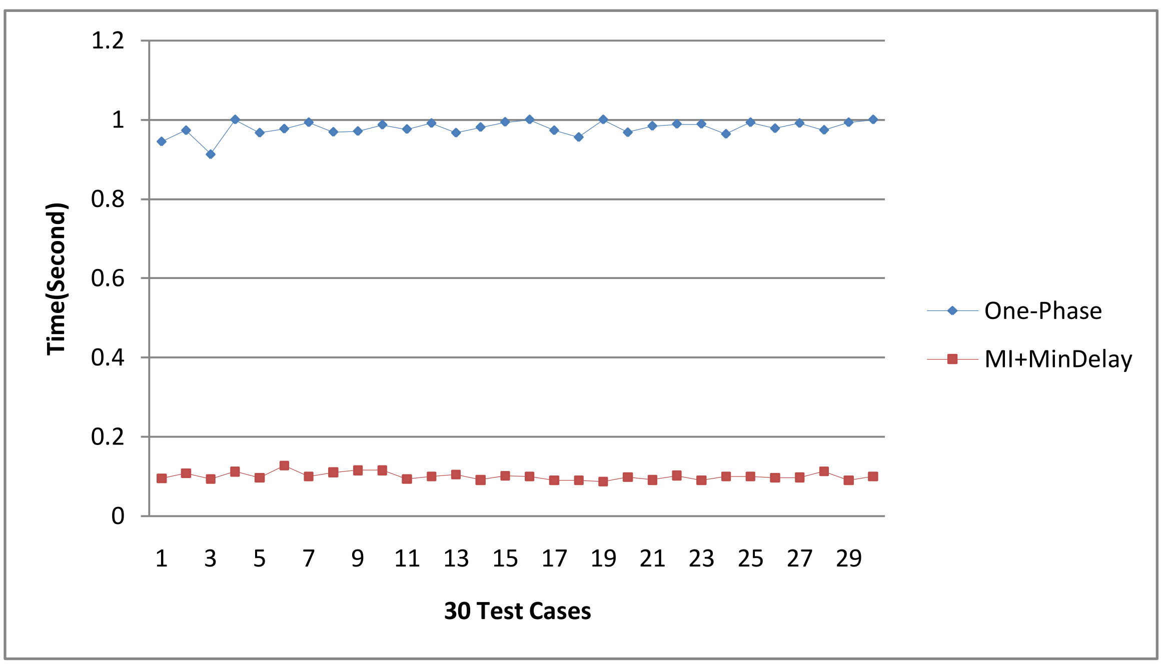


Figure 3.3. Time cost for solving MI+MinDelay and OP

4. RELATED WORK

For interference modeling, the most related work includes [1]–[5]. [1] first used conflict graphs to model the effect of wireless interference under a simplified protocol model; [3] continued to use conflict graphs to model interference under IEEE 802.11interference model; [6] focused on estimation of interference and studied the effect of interference on aggregated network throughput based on IEEE802.11 model; [4] proposed a physical interference model which is based on measured interference rather than distance between nodes. [9] did joint routing and link rate control based on a different interference model that is based on directed graphs.

To find the exact solution for maximum network throughput, the wireless link bandwidth must be considered. To deal with the bandwidth constraint, some scholar extended the capacity constraint of flow networks to wireless networks without considering the interference from other links [7, 8]; Some attempted to model interference but used global information such as cliques on a conflict graph ([3]). Since finding all cliques in a graph is an NP-hard problem, there is no known solution that is both efficient (in polynomial time) and accurate. Our interference model uses the sufficient condition on bandwidth constraint, and the algorithm is polynomial time. It can be efficiently applied in practice since it only uses local information.

Delay optimization, often very important in sensor networks, has been approached from routing, MAC layer scheduling, or both. [10] presented in sensor networks when the routing tree is given, how to determine the time slot of each node such that the maximum latency to send a packet from a node to the sink is minimized. [11] presented an algorithm to find optimal routing paths between sensor and sink node pairs with the objective of minimizing the total end-to-end delay. [12] presented approximation algorithms for minimum latency aggregation in sensor networks, which computes an aggregation tree as well as time slot assignment for links so that the make span of the schedule is minimum.

5. CONCLUSION AND FUTURE WORK

In this article, an important problem in practice has been addressed: Given a multi-hop wireless sensor network with multiple sources and sinks, how to achieve the minimum end-to-end delay? This article presented a cross-layer linear programming-based link scheduling scheme, in which wireless interference is sufficiently addressed. By using this scheme, the conflicting transmissions can be avoided at any time. Through the simulation, I show that the proposed link scheduling scheme can significantly reduce the end-to-end latency no matter what routing algorithm is used. Besides, this article also shows that the shortest path does not always lead to the least end-to-end latency. In this article, it is assumed that the underlying MAC is IEEE 802.11. Other MAC schemes will need minor modification to our model.

The optimization model is useful for feasibility analysis given a set of QoS constraints, and it is also useful for predicting the achievable performance of the network and improving delay when routing information is given. The optimization framework can also be used for admission control as part of QoS provisioning in wireless sensor networks. I will address this issue in the future work.

.

BIBLIOGRAPHY

[1] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, “Impact of interference on multi-hop wireless network performance,” in Mobi Com’ 3: Proceedings of the 9th annual international conference on Mobile computing and networking. New York, NY, USA: ACM, 2003, pp. 66–80.

[2] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, “Interference – aware fair rate control in wireless sensor networks,” in SIGCOMM’ 06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications. New York, NY, USA: ACM, 2006, pp.63–74.

[3] Y. Li, L. Qiu, Y. Zhang, R. Mahajan, Z. Zhong, G. Deshpande, and E.Rozner, “Effects of interference on wireless mesh networks: Pathologies and a preliminary solution,” in HotNets 2007, Nov. 2007.

[4] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, “A general model of wireless interference,” in MobiCom ’07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking. New York, NY, USA: ACM, 2007, pp. 171–182.

[5] Y. Li, L. Qiu, Y. Zhang, R. Mahajan, and E. Rozner, “Predictable performance optimization for wireless networks,” in SIGCOMM ’08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication. New York, NY, USA: ACM,2008,pp.413–426.

[6] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, “Estimation of link interference in static multi-hop wireless networks,” in IMC ’05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. Berkeley, CA, USA: US ENIX Association, 2005, pp. 28–28.

[7] M. Kodialam and T. Nandagopal, “Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem,” in MobiCom ’03: Proceedings of the 9th annual international conference on Mobile computing and networking. New York, NY, USA: ACM, 2003, pp. 42–54.

[8] M. Alicherry, R. Bhatia, and L. Li, “Joint channel assignment and routing for throughput optimization in multi radio wireless mesh networks,” in ACM MobiCom, 2005.

[9] M. Cheng, X. Gong, and L. Cai, “Link rate allocation under bandwidth and energy constraints in sensor networks,” in Global Telecommunications Conference, 2008. IEEE GLOBE-COM 2008. IEEE, 302008-Dec.42008, pp. 1–5.

[10] P. Chatterjee and N.Das, “A cross-layer distributed tdma scheduling for data gathering with minimum latency in wireless sensor networks,” in Wireless VITAE 2009, may 2009, pp. 813 – 817.

[11] F. Sivrikaya and B. Yener, “Minimum delay routing for wireless networks with stdma,” Wireless Networks, vol. 15, no. 6, pp. 755–772, October 2007.

[12] P. - J. Wan, S.C.- H. Huang, L. Wang, Z. Wan, and X. Jia, “Minimum-latency aggregation scheduling in multihop wireless networks,” in MobiHoc 09, 2009, pp. 185–194.

[13] M. X. Cheng and X. Gong, “Interference-aware multipath routing and link rate control in multihop wireless networks,” 2010, manuscript.

[14] Gamal, A.El. and Mammen, J. and Prabhakar, B. and Shah, D. “Throughput-delay trade-off in wireless networks,” in INFOCOM 2004, pp. 464–475.

[15] Xiaoxia Huang and Yuguang Fang, “End-to-end delay differentiation by prioritized multipath routing in wireless sensor networks,” in MILCOM 2005, pp. 1277–1283.

[16] Peng-Jun Wan and Xiaohua Xu and Lixin Wang and Xiaohua Jia and Park, E.K. “Minimum-Latency Beaconing Schedule in Multihop Wireless Networks,” in INFCOM.2009, pp. 2340–2346.

[17] Qin Xin, Yan Zhang and Jie Xiang. “Practical Routing in Delay-Tolerant Networks,” in IEEE International Conference.2009, pp. 1–5.

[18] Jones, E.P.C. and Li, L. and Schmidtke, J.K. and Ward, P.A.S. “Minimum-Latency Gossiping in Multi-Hop Wireless Mesh Networks,” pp. 943 –959.

[19] Jain, Sushant and Fall, Kevin and Patra, Rabin. “Routing in a delay tolerant network,” in SIGCOMM.2004, pp. 145 –158.

[20] Lei Yin and Hui-mei Lu and Yuan-da Cao. “A Novel Single Copy Replication Routing Strategy for Delay Tolerant Networks,” in WICOM.2004, pp. 1–5.

[21] Lee, U. and Soon Young Oh and Kang-Won Lee and Gerla, M. “RelayCast: Scalable multicast routing in Delay Tolerant Networks,” pp. 218–227.

[22] Nayebi, A. and Sarbazi-Azad, H. and Karlsson, G. “Routing, data gathering, and neighbor discovery in delay-tolerant wireless sensor networks,” in IPDPS. 2009, pp. 1–6.

[23] Boyer, J. and Falconer, D.D. and Yanikomeroglu, H. “Multihop diversity in wireless relaying channels,” in TCOMM. 2004, pp. 1820–1830.

[24] Li, H. and Lott, M. and Weckerle, M. and Zirwas, W. and Schulz, E. “Multihop communications in future mobile radio networks,” in Personal, Indoor and Mobile Radio Communications, 2002., pp. 54–58.

VITA

Yibo Xu was born on November 24 , 1984, in WuHan, P.R.China. He earned the Bachelor of Science degree at ShangHai Jiao Tong University in 2007. He was awarded the Master of Science degree in Computer Science from the Missouri University of Science and Technology in December 2010.