On Data Transmission Scheduling considering Switching Penalty in Mobile Sensor Networks

Yu Zhou, Danielle Bragg, Mira Yun and Hyeong-Ah Choi
Department of Computer Science, The George Washington University, Washington, DC 20052
Email: {yuzhou, mirayun, hchoi}@gwu.edu, danielle.k.bragg@gmail.com

***Abstract*—This paper presents fundamental results for the problem of throughput optimization in heterogeneous mobile sensor networks by considering the penalty incurred when a mobile node switches to another receiver. We present online and offline algorithms determining which receiver each mobile node connects to at each time slot to maximize total throughput. We initially consider the network setup with a single mobile sensor, and expand our solutions for two models with multiple mobile sensors and limited capacity for stationary receivers. We analyze the difficulty of our problem, and also the lower bound of the throughput ratio of our online algorithms. Simulation results demonstrate the impact of the decision parameter used to determine the propensity to switch receivers on the throughput ratio.**

I. Introduction

A cyber-physical networking system (CPNS) usually consists of a network of distributed embedded sensors, actuators, and computers that are all equipped with computing and communication capabilities. These units interact with each other in real-time [1], with varying capabilities. These systems must sense a dynamically changing environment, observe and detect application-specific phenomenons characterized by the presence of certain criteria, and react to the phenomenons in real-time. Cyber-physical systems (CPSs) plays a crucial role in a broad spectrum of application domains [2][3][4]. Examples of such systems range from semi-autonomous spacecrafts to radio controlled battlebots, and from autonomous aerial systems to inter-connected radiation sensors for detecting the presence of radioactive materials. CPS also has applications in health services, where portable, wearable, or implantable sensors are used to collect health information [5][6][7].

Transmitting critical data efficiently is a major problem in CPS. The goal of most CPS applications is to transmit data on physical characteristics through cyber space in a timely fashion for appropriate analysis and decision making. When we consider a network of wireless, mobile sensors as the infrastructure of CPS, delivering large amounts of real-time data successfully to the stationary receivers is one of most important issues because of the capacity-constrained and highly dynamic nature of wireless networks. This problem of efficient data transmission is further complicated by increasingly diverse transmission capabilities. For example, new sensors with multiple radio access technologies are being developed and deployed with various emerging applications. For these reasons, efficient transmission of critical physical data in heterogeneous environments is a crucial and difficult problem in optimizing the network throughput and interference.

In heterogeneous mobile sensor networks, one technique employed to optimize network performance is to allow mobile sensors to switch between receivers. Generally, each mobile sensor node (MS) detects its nearby stationary receivers (SRs) by scanning all available wireless channels and connects to the one with the strongest received signal strength indicator (RSSI). However, the RSSI may vary during the connection so that the maximum throughput is not achieved if the mobile node maintains one connection without switching. On the other hand, if mobile nodes are constantly switching to receivers with stronger RSSIs, this switching will take a toll on the throughput because the overhead of switching between two receivers is not negligible. For example, establishing a connection to a 802.11 access point may take hundreds of milliseconds [8].

The rest of the paper is organized as follows: In Section II, we outline the system model and give our problem statement. In Section III, we present online and offline SR-switching algorithms for networks with multiple MSs and multiple SRs, and consider two cases: Case 1 where SRs have limited available bandwidth, and Case 2 where SRs are limited in the number of MSs they can simultaneously service. We also prove the NP-completeness of optimizing throughput in networks with multiple MSs and multiple SRs where SRs have limited bandwidth. In Section IV, we analyze the online algorithms presented in Section III to provide a lower bound of their throughput ratios. In Section V, our online algorithm is implemented with a varied parameter used in switching decisions. Results demonstrate the impact of the decision parameter used to determine whether switching is worthwhile on the throughput ratio. Finally, Section VI presents our conclusions and final thoughts.

II. Network Model and Problem Statement

A. Network Model

In our work, we consider wireless mobile sensor networks with m SRs and n MSs. Let $\mathbb{S} = \{S_i\}$ be the set of SRs and $\mathbb{M} = \{M_j\}$ be the set of MSs, where $1 \leq i \leq m$, $1 \leq j \leq n$. The total number of time slots is T.

In each time slot $1 \leq t \leq T$, each MS can connect and transfer data to a single SR, and it can not switch from one SR to another SR within one time slot. We represent the connection from MSs to SRs by a function $\delta(i, j)_t \in \{0, 1\}$, indicating whether or not M_j is connected to S_i at time t.

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

We also define $r(i,j)_t$ to be the selected data rate (Mbps) between S_i and M_j at time slot t. For example, in 802.11b, $r(i,j)_t \in \{1, 2, 5.5, 11\}$. We consider RSSI to be the only factor determining the available data rates, though additional factors may contribute in more complex scenarios.

One SR can simultaneously connect to multiple MSs at time t. This yields an aggregated throughput $B(i)_t = \sum_j \delta(i,j)_t r(i,j)_t$, for S_i serving all connected MSs. We use $C(i)_t$ to denote the maximum throughput capacity of S_i at t.

At the end of each time slot t, each MS has the chance to switch between SRs. If switching occurs, a penalty will be applied in the following time slot so that the throughout for that MS in the next time slot will be 0.

B. Problem Statement

Given: a 3-dimensional data rate matrix $r(i,j)_t$ on dimensions i, j and t.

Goal: find a solution on $\delta(i,j)_t$ for all $1 \leq i \leq m$, $1 \leq j \leq n$ and $1 \leq t \leq T$, such that $\max \sum_{1 \leq t \leq T} \sum_{1 \leq i \leq m} B(i)_t$.

III. Algorithms

In this section, we develop algorithms to help optimize throughput for networks with Multiple MSs and Multiple SRs (MMMS). We explore two possible models: a Bandwidth Constraint Model where each RS provides a limited amount of bandwidth, and a MS-Number Constraint Model where each RS can service a limited number of MSs simultaneously. To do this, we first explore algorithms for networks with a Single MS and Multiple SRs (SMMS).

A. Single MS Multiple SRs

In the Single MS Multiple SRs problem (SMMS), the network contains a single MS M, yielding a 2-dimensional data rate matrix on i and t, such that $r(i)_t$ denotes the data connection rate of M with S_i at time t, and $\delta(i)_t$ denotes the connection status of M to S_i at t. We assume each SR S_i can always accommodate M, that is, $C(i)_t \geq r(i)_t$. Since each MS can only connect to one SR at each time slot, this problem is equivalent to maximizing the overall throughput of M, i.e., $\max \sum_{1 \leq t \leq T, 1 \leq i \leq m} \delta(i)_t r(i)_t$.

1) Offline Algorithm:

Theorem 3.1: The time complexity of the offline algorithm is $O(mT)$.

Proof: The offline version of this problem can be solved by Dynamic Programming.

Let $TH(S_{i_0}, t_0)$ denote the maximum throughput of M in the optimal solution when M initially connects to S_{i_0} at t_0, so that we have $TH(S_{i_0}, t_0) = \max\{\sum_{t_0 \leq t \leq T, 1 \leq i \leq m} \delta(i)_t r(i)_t | \delta(i_0)_{t_0} = 1\}$.

Let $TH(t_0)$ denote the maximum throughput of M in the optimal solution starting from t_0. Then we have

$$\begin{cases} TH(t_0) = \max\{TH(S_i, t_0)\}, \forall 1 \leq i \leq m \\ TH(S_i, t_0) = r(i)_{t_0} + \max\{TH(S_i, t_0 + 1), TH(t_0 + 2)\} \end{cases} \tag{1}$$

■

2) Online Algorithm: The competitive ratio for any 0-lookahead online algorithm is unbounded because the switching penalty is one time slot. Suppose M_j connects to S_i at time slot $t-1$. No matter what algorithm is applied by the MS, if a switch is proposed at time slot $t-1$, no throughput will be gained at t. Alternatively, if the MS does not switch at time t and achieves throughput $r(i,j)_t$, we can always assume there is a much larger data rate available through another SR so that no competitive ratio can be achieved for M_j. However, 1-lookahead online algorithms can have bounded competitive ratios, as [9] proposed a 1-lookahead online algorithm with a competitive ratio of 4 that employs a Wait-Dominate strategy.

B. MMMS Case 1: Bandwidth Constraint Model

In the Multiple MSs Multiple SRs problem (MMMS), more than one MS can connect to the same SR in the same time slot. In reality, multiple MSs connecting to the same SR can cause collisions, but for simplicity we neglect the effect of these collisions on the throughput. In Case 1, where the summation of the data rates of all MSs connected to a single SR cannot exceed the bandwidth capacity of that SR, our problem can be described as:

Given: a 3-dimension data rate matrix $r(i,j)_t$ on dimensions i, j and t.

Goal: find a solution on $\delta(i,j)_t$ for all $1 \leq i \leq m$, $1 \leq j \leq n$ and $1 \leq t \leq T$, such that $\max \sum_{1 \leq t \leq T} \sum_{1 \leq i \leq m} B(i)_t$ under the condition that $B(i)_t \leq C(i)_t \, \forall i$.

1) Offline Algorithm:

Theorem 3.2: Case 1 of our problem is NP-Complete even if $T = 1$.

Proof: Transformation from the 3-Partition Problem.

Let $Q = \{q_1, \cdots q_{3m}\}$ be an instance of $3m$ elements to the 3-Partition Problem where q_i is a positive integer. Note that for each $1 \leq k \leq 3m$, $\frac{C}{4} < q_i < \frac{C}{2}$ where $C = \frac{\sum_{k=1}^{3m} q_k}{m}$. We then construct an instance as an input to Case 1 as follows.

Let $n = 3m$, i.e., $\mathbb{M} = \{M_1, M_2, \ldots, M_{3m}\}$. $r(i,j)$ denotes the data rate from M_j to S_i, and we let $q_j = r(1,j) = r(2,j) = \cdots = r(m,j)$ $\forall 1 \leq j \leq 3m$. We do not consider time, since we are handling the case of a single time slot.

We now claim that there exists a 3-partition of Q if and only if Case 1 can be solved with total throughput mC.

Suppose there exists a 3-partition of Q such that each partition is of form $\{3(i-1)+1, 3(i-1)+2, 3(i-1)+3\}, \forall 1 \leq i \leq m$. We then define the connection from MS to SR as follows:

$$\begin{cases} \delta(i,j)_1 = 1, \text{if } j \in \{3(i-1)+1, 3(i-1)+2, 3(i-1)+3\}. \\ \delta(i,j)_1 = 0, \text{otherwise.} \end{cases} \tag{2}$$

We observe that the bandwidth requirement for each MS as well as the bandwidth constraint for each SR are satisfied and the total throughput mC is achieved.

On the other hand, assume that there is a feasible solution to Case 1 with total throughput mC. Suppose there is an SR, say S_{i_0}, that is simultaneously assigned more than three MSs.

We then note that the total bandwidth required by the MSs assigned to S_{i_0} is larger than C due to the condition that $\frac{C}{4} < q_i < \frac{C}{2}$ for each $1 \leq i \leq 3m$. However, each SR can handle a maximum bandwidth of C, which means that at most three MSs are assigned to each SR. As there are $3m$ MSs that need to be assigned among m SRs, this implies that each SR must be assigned exactly three MSs. A solution to the 3-Partition Problem can then be easily obtained, and this completes the proof. ∎

2) Online Algorithm:

Theorem 3.3: The Wait-Dominate algorithm has a competitive ratio of $\frac{4}{\beta}$ for our problem when each SR has a limited total bandwidth available to connect to MSs.

Proof: $0-1$ multiple Knapsack problem [8]: Suppose we are given n items, each having a profit p_j and a weight w_j, where $1 \leq j \leq m$, and m knapsacks. Assign the items to the knapsacks so that the total profit of the assigned items is maximal, the total weight assigned to each knapsack does not exceed its capacity, and each item is either assigned to one of the knapsacks or rejected. Formally:

maximize: $\sum_{i=1}^{m}\sum_{j=1}^{n} p_j x_{i,j}$

subject to:

1. $\sum_{j=1}^{n} w_j x_{i,j} \leq k_i$, for $1 \leq i \leq m$;
2. $\sum_{i=1}^{m} x_{i,j} \leq 1$, for $1 \leq j \leq n$;
3. $x_{i,j} = 0$ or 1, for $1 \leq i \leq m$, $1 \leq j \leq n$

The SR switching problem with $T = 1$ is equivalent to the $0-1$ multiple Knapsack problem if we assume the SR capacity C_i in our problem is equal to k_i in the $0-1$ multiple Knapsack problem, and the data rate of each MS is equal to weight w_j. Let $x_j = w_j$ for each item.

By applying a β-approximation algorithm at each time slot, we can obtain an array X of the upper bound throughput in the whole network for our SMMS-Online Algorithm. We can then apply the Wait-Dominate online algorithm to the SMMS problem on X. Since the β-approximation algorithm yields a competitive ratio of $\frac{1}{\beta}$, [1] and the Wait-Dominate strategy yields a competitive ratio of 4 as proved in [9], we obtain the overall competitive ratio $\frac{4}{\beta}$. ∎

Finally, we present our α-Online Algorithm with Bandwidth Constraint (α-OABC). For a MS M_j connected to SR S_i at time slot $t-1$, if there exists an SR $S_{i'}$ such that $\alpha(r(i,j)_t + r(i,j)_{t+1}) < r(i',j)_{t+1}$, then M_j switches to $S_{i'}$ at time slot t, and otherwise it maintains its connection with the current SR S_i. $\alpha > 1$ is called the decision factor. If multiple SRs are satisfied by the condition, choose the one having the largest $r(i',j)_{t+1}$. For each SR S_i at time slot t, if the overall throughput from the connected MSs is larger than the throughput capacity of S_i, one MS with the minimum data rate will be chosen from those connected to S_i, and its throughput will be reset to 0 at t. The process repeats until the the remaining overall throughput of S_i is no more than its throughput capacity $C(i)_t$. α-OABC Algorithm is shown in Algorithm 1.

Algorithm 1 α-Online Algorithm with Bandwidth Constraint (α-OABC)

```
for j = 1 to n do
  find i with the maximum r(i,j)_1 + r(i,j)_2;
  δ(i,j)_1 ← 1;
end for
for i = 1 to m do
  while B(i)_1 > C(i)_1 do
    find the minimum r(i,j)_1 such that r(i,j)_1 > 0 and
    δ(i,j)_1 = 1;
    B(i) ← B(i) − r(i,j)_1;
    r(i,j)_1 ← 0;
  end while
end for
for t = 2 to T do
  for j = 1 to n do
    find i such that δ(i,j)_{t−1} = 1;
    find i' with the maximum r(i,j')_{t+1} such that α(r(i,j)_t +
    r(i,j)_{t+1}) < r(i',j)_{t+1};
    δ(i,j)_t ← 0; δ(i',j)_t ← 1;
    r(i',j)_t ← 0;
  end for
  for i = 1 to m do
    while B(i)_t > C(i)_t do
      find i with the minimum r(i,j)_t such that r(i,j)_t > 0
      and δ(i,j)_t = 1;
      B(i) ← B(i) − r(i,j)_t;
      r(i,j)_t ← 0;
    end while
  end for
end for
```

C. MMMS Case 2: MS-Number Constraint Model

In Case 2, we consider networks where each SR can connect to a limited number of MSs at any time t, and this limit is expressed as $N(i)_t$, for all $1 \leq i \leq m$. We neglect the effect of collisions on the throughput when $K(i)_t$, the total number of connections to S_i at time t, is no more than $N(i)_t$, so that our problem can be described as:

Given: a 3-dimensional data rate matrix $r(i,j)_t$ on dimensions i, j and t.

Goal: find a solution on $\delta(i,j)_t$ for all $1 \leq i \leq m$, $1 \leq j \leq n$ and $1 \leq t \leq T$, such that $\max \sum_{1 \leq t \leq T} \sum_{1 \leq i \leq m} B(i)_t$ under the condition that $K(i)_t \leq N(i)_t$. We assume $n \leq \sum N(i)_t$ for all t.

1) Offline Algorithm:

Theorem 3.4: The Case 2 of our problem is a P problem if $T = 1$.

Proof: This problem is identical to the Maximum Weighted Matching Problem. In graph G, we create n vertices M_j representing n MSs, where $1 \leq j \leq n$. We create $N(i)_t$ vertices S_i^k for each S_i, where $1 \leq i \leq m$ and $1 \leq k \leq N(i)_t$. Between each vertex pair S_i^k and M_j, we create an edge

[1]In paper [10], the author provides a $\frac{1}{2}$-approximation algorithm for the $0-1$ multiple Knapsack problem.

weighed by $r(i,j)_t$. The problem of finding the maximum summation of throughput for all SRs is equivalent to finding a maximum weighted matching in the bipartite graph G. Finding the maximum weighted matching in a bipartite graph can be done in $O((n + \sum N(i))^{2.5})$ time. The details can be found in [11], [12].

Figure 1 gives an example of our problem with 2 SRs and 4 MSs, which can be transformed into a bipartite graph G, as in Figure 2. The maximum weighted matching is displayed in blue, and the maximum weight is 41. ■

		M1	M2	M3	M4
N1 = 2	S1	12	18	8	7
N2 = 2	S2	6	16	4	5

Fig. 1. An Example with 2 SRs and 4 MSs

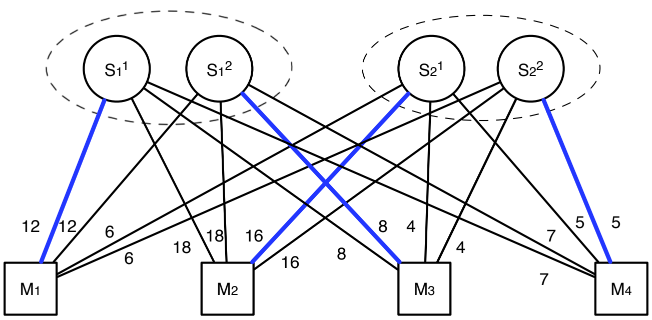

Fig. 2. Maximum Weighted Matching

2) *Online Algorithm:*

Theorem 3.5: The Wait-Dominate algorithm has a competitive ratio of 4 for our problem when each SR has a maximum number of MSs it can service.

Proof: Applying the offline algorithm above, we can obtain an array W of the upper bound throughput in the network at each time slot. By using the Wait-Dominate online algorithm for SMMA on X, we conclude that the competitive ratio of our solution is 4 for Case 2, when the number of MSs an SR can service simultaneously is constrained. ■

Our α-Online Algorithm with MS-Number Constraint (α-OAMNC) is exactly the same as the Case 1 algorithm presented in the previous section except that the number of connected MSs is used as a constraint instead of bandwidth.

IV. Theoretical Analysis on General α-Algorithm

Definition 4.1: α-**Algorithm**: A MS M_j performs a switch from S_i to $S_{i'}$ if and only if $\alpha(r(i,j)_t + r(i,j)_{t+1}) < r(i',j)_{t+1}$, where α is the decision factor and $\alpha > 1$. If more than one SR is satisfied, M_j switches to the one which has the maximum $r(i',j)_{t+1}$.

Algorithm 2 α-Online Algorithm with MS-Number Constraint (α-OAMNC)

for $j = 1$ to n **do**
  find i with the maximum $r(i,j)_1 + r(i,j)_2$;
  $\delta(i,j)_1 \leftarrow 1$;
end for
for $i = 1$ to m **do**
  while $K(i)_1 > N(i)_1$ **do**
    find the minimum $r(i,j)_1$ such that $r(i,j)_1 > 0$ and $\delta(i,j)_1 = 1$;
    $K(i) \leftarrow K(i) - 1$;
    $r(i,j)_1 \leftarrow 0$;
  end while
end for
for $t = 2$ to T **do**
  for $j = 1$ to n **do**
    find i such that $\delta(i,j)_{t-1} = 1$;
    find i' with the maximum $r(i,j')_{t+1}$ such that $\alpha(r(i,j)_t + r(i,j)_{t+1}) < r(i',j)_{t+1}$;
    $\delta(i,j)_t \leftarrow 0$; $\delta(i',j)_t \leftarrow 1$;
    $r(i',j)_t \leftarrow 0$;
  end for
  for $i = 1$ to m **do**
    while $K(i)_t > N(i)_t$ **do**
      find i with the minimum $r(i,j)_t$ such that $r(i,j)_t > 0$ and $\delta(i,j)_t = 1$;
      $K(i) \leftarrow K(i) - 1$;
      $r(i,j)_t \leftarrow 0$;
    end while
  end for
end for

Adversarial strategies can be categorized into two major types. In Type 1, the adversary always generates a rate r for the SRs that MS M is not connected to below the switching threshold, so that M persistently stays connected to S_1 without performing any switching during the duration of the pattern.

Theorem 4.1: The lower bound of the throughput ratio ψ of Type 1 is $\frac{1}{2\alpha}$.

t	1	2	3	4	...	k-1	k	k+1
S1:	$r(1)_1$	$r(1)_2$	$r(1)_3$	$r(1)_4$	...	$r(1)_{k-1}$	$r(1)_k$	$r(1)_{k+1}$
S2:	$r(2)_1$	$r(2)_2$	$r(2)_3$	$r(2)_4$	...	$r(2)_{k-1}$	$r(2)_k$	$r(2)_{k+1}$

Fig. 3. Type 1

Proof: Suppose the arbitrary r values in Figure 3 are given, where there are two SRs S_1 and S_2. The arrow indicates the SR choice by the MS. The shadow means the throughput at time slot t is larger than 0. From the definition of Type 1, we have the following inequalities.

$$\begin{cases} \alpha(r(1)_1 + r(1)_2) \geq r(2)_2, \\ \alpha(r(1)_2 + r(1)_3) \geq r(2)_3, \\ \alpha(r(1)_3 + r(1)_4) \geq r(2)_4, \\ \dots, \\ \alpha(r(1)_{k-1} + r(1)_k) \geq r(2)_k, \\ \alpha(r(1)_k + r(1)_{k+1}) \geq r(2)_{k+1}, \end{cases} \tag{3}$$

which is

$$\alpha(\sum_{i=1}^{k} r(1)_i + \sum_{i=2}^{k-1} r(1)_i) \geq \sum_{i=2}^{k} r(2)_i \tag{4}$$

And we also know

$$Th(ON) = \sum_{i=1}^{k} r(1)_i \tag{5}$$

and

$$Th(OPT) = \sum_{i=2}^{k} r(2)_i \tag{6}$$

From Inequality (4), Equation (5) and (6), we complete the proof that

$$\psi = \frac{Th(ON)}{Th(OPT)} \geq \frac{1}{2\alpha} \tag{7}$$

The proof can be easily extended from two SRs to more than two SRs. All SRs except S_1 have the same x values as S_2 such that M still constantly connects to S_1 without switching. ∎

In Type 2, the second type of adversarial strategy, the adversary continuously generates a rate r above the switching threshold. Consequently, M switches back and forth between S_1 and S_2 in order to obtain a larger throughput in the next time slot, but instead constantly incurs the switching penalty and obtains throughput 0, until the last time slot k. M does not connect to the same SR in any two adjacent time slots except during the last two.

Theorem 4.2: The lower bound of the throughput ratio ψ of Type 2 is $\frac{\alpha-1}{\alpha}$.

t	1	2	3	4	...	k-1	k	k+1
S_1:	$r(1)_1$	$r(1)_2$	$r(1)_3$	$r(1)_4$	...	$r(1)_{k-1}$	$r(1)_k$	$r(1)_{k+1}$
S_2:	$r(2)_1$	$r(2)_2$	$r(2)_3$	$r(2)_4$	...	$r(2)_{k-1}$	$r(2)_k$	$r(2)_{k+1}$

Fig. 4. Type 2

Proof: Figure 4 shows the SR choices of Type 2 where we suppose k is odd without loss of generality.

$$\begin{cases} \alpha(r(1)_1 + r(1)_2) < r(2)_2, \\ \alpha(r(2)_2 + r(2)_3) < r(1)_3, \\ \alpha(r(1)_3 + r(1)_4) < r(2)_4, \\ \ldots, \\ \alpha(r(2)_{k-1} + r(2)_k) < r(1)_k, \\ \alpha(r(1)_k + r(1)_{k+1}) \geq r(2)_{k+1}, \end{cases} \tag{8}$$

which is

$$\alpha(\sum_{i=1}^{k-1} r(1)_i + \sum_{i=2}^{k} r(2)_i) < \sum_{i=1}^{\frac{k-3}{2}} r(1)_{2i+1} + \sum_{i=1}^{\frac{k-1}{2}} r(2)_{2i} + r(1)_k$$

$$\begin{aligned} (\alpha-1)(\sum_{i=1}^{k-1} r(1)_i + \sum_{i=2}^{k} r(2)_i) &< r(1)_k \\ (\alpha-1)(\sum_{i=1}^{k} r(1)_i + \sum_{i=2}^{k} r(2)_i) &< \alpha r(1)_k \end{aligned} \tag{9}$$

And we also know

$$Th(ON) = r(1)_k \tag{10}$$

and

$$Th(OPT) < \sum_{i=1}^{k} r(1)_i + \sum_{i=2}^{k} r(2)_i \tag{11}$$

From Inequality (9), Equation (10) and (11), we complete the proof that

$$\psi = \frac{Th(ON)}{Th(OPT)} > \frac{\alpha-1}{\alpha} \tag{12}$$

If there are more than two SRs in the network, the same results can be derived.

∎

V. Simulation Results

In our simulations, we create an area of size 100×100. 20 SRs and 100 MSs are randomly deployed in this area. Each mobile station moves in a straight line with constant speed v between random points on the boarder of the area. The overall simulation time is 1000 time slots. We run 100 trials of each simulation setting and take the average of these trials.

Since our switching protocols are designed for networks limited to single-hop communication capabilities, we model our simulations on the settings of IEEE 802.11 wireless networks. Accordingly, we consider data rates 5.6, 10.5, 18.6, 25.2, 30.5 and 32.5 (Mbps), which have ranges 141, 113, 76, 61, 45 and 41 (meters) respectively. From these data rates, we know that the simulation results will not be effected if $\alpha > \frac{32.5}{5.6+5.6} = 2.902$ because for larger values of alpha, no switching can occur. We run simulations for both Case 1, where SRs are bandwidth-constrained, and Case 2, where SRs can service a limited number of MSs simultaneously.

From the results in Fig 5 and 6, we notice that when the decision factor α is small, each MS is more sensitive to data rate changes and SR switching is more prone to happen. This sensitivity produces a high throughput ψ [2] for MSs that move slowly, since the variability in their available data rates is

[2] ψ is defined as the throughput of online algorithm over the summation of the throughput of each MS using Dynamic programming in SMMS-Online Algorithm.

relatively low, which allows them to take full advantage of each switch. Conversely, this sensitivity produces a relatively low throughput psi for MSs that move quickly. When MSs move very quickly, by the time they switch to an SR with a higher capacity, they have moved so much that the available data rates have changed, and they must switch again. This constant switching incurs too many switching penalties, which waste throughput.

When α is large, each MS is less sensitive to data rate changes. Thus, slow-moving MSs, which are presented a low variability of available data rates, miss out on more opportunities to gain the benefits of switching and thus yield lower throughput. Conversely, fast-moving MSs, which have a high variability of available data rates, avoid the excessive switching penalties incurred when they switched more freely. As a result, these fast-moving nodes enjoy a boost in throughput for larger values of α.

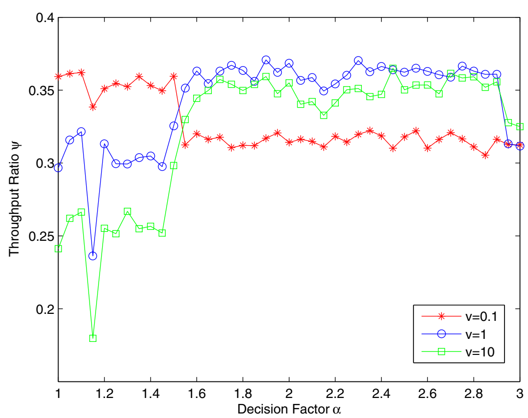

Fig. 5. Case 1 on α

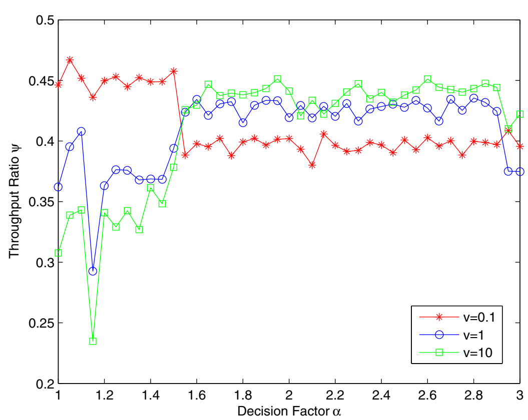

Fig. 6. Case 2 on α

VI. Conclusion

This paper presents online and offline SR-switching algorithms for networks with multiple MSs and multiple SRs in order to maximize total throughput. In such networks, MSs may achieve better throughput by connecting to an SR with higher service capabilities. However, the act of switching deducts from the throughput. By considering this switching penalty, we proposed algorithms for various network setups, including networks with a single MS and multiple SRs, as well as networks with multiple MSs and multiple SRs. For networks with multiple MSs and multiple SRs having limited bandwidth, we also prove the NP-completeness of optimizing throughput. We analyze our online algorithms to provide a lower bound of their throughput ratios. Finally, our simulation results demonstrate the impact of the switching decision parameter on the throughput ratio.

In the course of this work, we focus on networks limited to single-hop capabilities. We present protocols for MSs to communicate directly with SRs, instead of communicating with each other. In the future we would like to expand these protocols to take advantage of multi-hop capabilities in networks. Since many wireless sensor networks allow sensor nodes to communicate with each other, such an expansion would have powerful applications for wireless sensor networks.

References

[1] E.A. Lee, “Cyber physical systems: Design challenges,” in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, May 2008, pp. 363 –369.

[2] L.J. Hoffman, T. Rosenberg, R. Dodge, and D. Ragsdale, “Exploring a national cybersecurity exercise for universities,” *Security Privacy, IEEE*, vol. 3, no. 5, pp. 27 – 33, 2005.

[3] A.M.K. Cheng, “Cyber-physical medical and medication systems,” in *Distributed Computing Systems Workshops, 2008. ICDCS ’08. 28th International Conference on*, 2008, pp. 529 –532.

[4] K. Gill, Shuang-Hua Yang, Fang Yao, and Xin Lu, “A zigbee-based home automation system,” *Consumer Electronics, IEEE Transactions on*, vol. 55, no. 2, pp. 422 –430, May 2009.

[5] Yonglin Ren, R.W.N. Pazzi, and A. Boukerche, “Monitoring patients via a secure and mobile healthcare system,” *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 59 –65, 2010.

[6] A. Karargyris and N. Bourbakis, “Wireless capsule endoscopy and endoscopic imaging: A survey on various methodologies presented,” *Engineering in Medicine and Biology Magazine, IEEE*, vol. 29, no. 1, pp. 72 –83, 2010.

[7] J. Abouei, D. Brown, K. Plataniotis, and S. Pasupathy, “Energy efficiency and reliability in wireless biomedical implant systems,” *Information Technology in Biomedicine, IEEE Transactions on*, 2011.

[8] Silvano Martello and Paolo Toth, “Algorithm 632: A program for the 0–1 multiple knapsack problem,” *ACM Trans. Math. Softw.*, vol. 11, pp. 135–140, June 1985.

[9] Amrinder Arora, Fanchun Jin, Gokhan Sahin, Hosam Mahmoud, and Hyeong-Ah Choi, “Throughput analysis in wireless networks with multiple users and multiple channels,” *Acta Inf.*, vol. 43, pp. 147–164, September 2006.

[10] M. Dawande, J. Kalagnanam, P. Keskinocak, F.S. Salman, and R. Ravi, “Approximation algorithms for the multiple knapsack problem with assignment restrictions,” *Journal of Combinatorial Optimization*, vol. 4, pp. 171–186, 2000, 10.1023/A:1009894503716.

[11] Andrew V. Goldberg and Robert E. Tarjan, “A new approach to the maximum flow problem,” *Journal of the ACM*, vol. 35, pp. 921–940, 1988.

[12] John E Hopcroft and Richard M Karp, “An n5/2 algorithm for maximum matchings in bipartite graphs,” *SIAM Journal on Computing*, vol. 2, no. 4, pp. 225–231, 1973.