Energy-Efficient Deterministic Routing Protocols in Radio Networks

Koji Nakano
Dept. of Electrical and Computer Engineering
Nagoya Institute of Technology
Showa-ku, Nagoya 466-8555, JAPAN
nakano@elcom.nitech.ac.jp

Stephan Olariu
Dept. of Computer Science
Old Dominion University
Norfolk, Virginia 23529, U.S.A.
olariu@cs.odu.edu

Albert Zomaya
Dept. of Electrical and Electronic Engineering
The University of Western Australia
Perth, WA 6970, AUSTRALIA

Abstract

A radio network (RN, for short) is a distributed system populated by small, bulk-produced, hand-held radio transceivers, running on batteries. Since recharging batteries may not be possible while on mission, it is important to design protocols that are highly energy-efficient. In this work we address the problem of energy-efficient routing in k-channel RNs. An important subproblem is that of permutation routing *an instance of which involves p stations each storing $\frac{n}{p}$ items. Since in the worst case each item must be transmitted at least once, every permutation routing protocol must take $\frac{n}{k}$ time slots. Similarly, each station must be awake for at least $\frac{n}{p}$ time slots.*

Our main contribution is to present an almost optimal energy-efficient permutation routing protocol on the k-channel, p-station RN that routes n items in at most $(2d+2b+1)\frac{n}{k}+k$ time slots, with no station being awake for more than $(4d+7b-1)\frac{n}{p}$ time slots, where $d = \lceil \frac{\log \frac{p}{k}}{\log \frac{n}{p}} \rceil$, $b = \lceil \frac{\log k}{\log \frac{n}{p}} \rceil$, and $k \leq \sqrt{\frac{p}{2}}$.

1 Introduction

Unlike the well-studied cellular systems that assume the existence of a robust infrastructure, *radio networks* (RN, for short) must be rapidly deployable, self-organizing, and capable of multimedia service support. Radio networks suit well the needs specific to disaster relief, search-and-rescue, law enforcement, collaborative computing, interactive mission planning, and other special-purpose applications. An RN is a distributed system with no central arbiter, consisting of p radio transceivers $S(1), S(2), \ldots, S(p)$, henceforth referred to as *stations*. We assume that each of the p stations has a unique ID in the range 1 to p. This is a reasonable assumption because the task of assigning unique ID numbers to the stations of an RN can be performed very efficiently [1].

The stations of the RN can only communicate through the radio frequency channels. As customary, time is assumed to be slotted and all the stations have a local clock that keeps synchronous time, perhaps by interfacing with a Global Positioning System (GPS, for short). We assume that the stations of the RN are simple hand-held devices running on batteries and, therefore, saving power is exceedingly important, as recharging batteries may not be possible while on mission. As it turns out, significant energy is being expended by a station while its transceiver is active that is, while transmitting or receiving a packet. Accordingly, we judge the goodness of a protocol by the following two yardsticks:

- the overall number of time slots required by the protocol to terminate, and
- for each individual station, the total number of awake time slots.

It is relatively straightforward to minimize the overall completion time at the expense of energy consumption. Similarly, one can minimize energy consumption at the expense of completion time [3, 4]. The challenge is to strike a balance between the two by designing protocols that take a small number of time slots to terminate while being, at the same time, as energy-efficient as possible.

Recently, several authors have looked at the power consumption characteristics of several medium access protocols in single-hop radio networks. Their conclusion is that protocols that require transmission contention resolution feature a higher energy consumption than reservation-based protocols. Accordingly, our routing protocols will involve a reservation stage that precludes collisions and, thus, saves energy in the long run.

Suppose that n items are stored by the p stations of an RN, such that every station stores $\frac{n}{p}$ items. Each item has a unique *destination* which is the identity of the station to which the item must be routed. The goal is to route all the items to their destinations. In *permutation routing*, every station is the destination of exactly $\frac{n}{p}$ items. We assume that each packet consists of an item along with a few additional bits such as the index of a station, the number of items, etc. For all practical purposes, this prohibits large aggregates from being sent in one packet. We refer the reader to Figure 1 for an illustration of the permutation routing problem with $n = 27$ and $p = 9$. For simplicity, for each item we only indicate its destination station.

Since each of n items must be transmitted at least once, permutation routing on the k-channel RN requires at least $\frac{n}{k}$ time slots. Concerning the energy efficiency, observe that in the worst case every station must be awake for at least $\frac{n}{p}$ time slots to transmit/receive items.

If energy efficiency is not an issue, routing on the RN is rather straightforward and has been discussed in the literature [2]. Indeed, consider an instance of the routing problem involving n items stored, in some order, by the p stations of an RN. Taking turns, each station transmits its items, one by one, and every station tunes to the channel. The station whose identity matches the destination of the item being transmitted copies the item in its local memory. It should be clear that this simple protocol terminates in exactly n time slots and is, therefore, optimal. However, the protocol is not energy-efficient since every station has to monitor the channel for n time slots.

Recently, Nakano *et al.* [2] presented permutation routing protocols running in $2\frac{n}{k} + k - 1$ time slots subject to k satisfying $k \leq \sqrt{\frac{p}{2}}$. The protocols in [2] trade total number of slots for energy efficiency since. Indeed these protocols run extremely fast at the expense of a high energy consumption since every station must be awake for the duration of the protocol.

Our main contribution is to present an almost optimal energy-efficient permutation routing protocol for the RN(p, k) that routes n packets in at most $(2d + 2b + 1)\frac{n}{k} + k$ time slots, with no station being awake for more than $(4d + 7b - 1)\frac{n}{p}$ time slots, where $d = \lceil \frac{\log \frac{p}{k}}{\log \frac{n}{p}} \rceil$, $b = \lceil \frac{\log k}{\log \frac{n}{p}} \rceil$, and $k \leq \sqrt{\frac{p}{2}}$. Our protocol is optimal if for some constant ϵ, $(0 < \epsilon < 1)$, $p \leq n^{1-\epsilon}$, which is the case in the vast majority of practical applications. This protocol is similarly optimal whenever $p \leq n^{1-\epsilon}$ for every fixed $\epsilon > 0$.

2 Reservation protocols

The purpose of this section is to discuss the details of a simple reservation protocol that will allow the stations to implicitly reserve the channel for an appropriate number of time slots to complete their transmissions without any collisions.

Consider a single-channel, p-station RN populated by stations $S(1), S(2), \ldots, S(p)$ and assume that for every i, $(1 \leq i \leq p)$, station $S(i)$ has n_i items to transmit. At the beginning of the reservation protocol only station $S(i)$ knows n_i.

The details of the reservation stage follow. To begin, station S(1) transmits the number n_1 of items it has to transmit. At this time all the stations have their transceiver powered off (i.e. asleep) except for S(1) that is transmitting and S(2) that is picking up n_1. Now, S(2) computes $n_1 + n_2$ and transmits the result to S(3). In this time slot all the stations are asleep except for S(2)

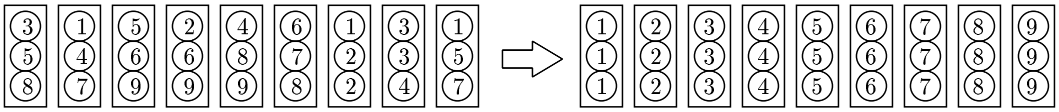

Figure 1: *Illustrating permutation routing on a 9-station RN*

and S(3). Upon receiving $n_1 + n_2$, S(3) computes $n_1 + n_2 + n_3$ and transmits the result to S(4). This is then continued until, at the end of $p - 1$ time slots, S(p) receives $n_1 + n_2 + n_3 + \cdots + n_{p-1}$. Thus, the reservation stage runs for $p - 1$ time slots. Importantly, during this stage no station is awake for more than two time slots.

Notice that at the end of the reservation stage each station can determine the exact number of transmissions that will be performed by stations preceding it and, consequently, will power itself off until the exact time slot when it has to transmit. More precisely, after the reservation stage, each station S(i), $(1 \leq i \leq p)$, knows $n_1 + n_2 + \ldots + n_{i-1}$. Thus, station S($i$) will wake up in time slot $n_1 + n_2 + \ldots + n_{i-1} + 1$ and will start transmitting its n_i items after which it will go back to sleep.

It is worth noting that this reservation protocol is an instance of the well-known *prefix sums* protocol used extensively in distributed and parallel systems. To summarize, we state the following result.

Lemma 2.1 *The reservation protocol involving the p stations of the single-channel* RN *terminates in $p - 1$ time slots, with no station being awake for more than two time slots.*

To set the stage for our first energy-efficient permutation routing protocol, let $\alpha(i, j)$, $(1 \leq i, j \leq p)$, be the set of items stored by station S(i) whose final destination is station S(j). Write

$$\begin{aligned} \alpha(*, j) &= \alpha(1, j) \cup \alpha(2, j) \cup \cdots \cup \alpha(p, j), \text{ and} \\ \alpha(i, *) &= \alpha(i, 1) \cup \alpha(i, 2) \cup \cdots \cup \alpha(i, p). \end{aligned}$$

Clearly, our job is to route for every j, $(1 \leq j \leq p)$, the items in $\alpha(*, j)$ to station S(j).

Fix an arbitrary j, $(1 \leq j \leq p)$. The protocol involves two stages. The first stage is a *reservation stage* where each station $S(i)$, $(1 \leq i \leq p)$, will implicitly reserve the channel for an appropriate number of time slots that will allow it, in the second stage, to transmit all the items destined for $S(j)$. Interestingly, outside of this timeframe, station $S(i)$ will be asleep, thus saving energy.

In the first stage, we use the reservation protocol for Lemma 2.1. Notice that at the end of the reservation stage each station can determine the exact number of transmissions that will be performed by stations preceding it and, consequently, will power itself off until the exact time slot when it has to transmit.

In the second stage, taking turns, the stations will wake up at the right time and will transmit their items to $S(j)$. Thus, for a given value of j, the task of routing all the items whose destination is $S(j)$ takes $\frac{n}{p} + p - 1$ time slots; station $S(j)$ is awake for $\frac{n}{p} + 2$ time slots while every station $S(i)$, $(i \neq j)$, is awake for $|\alpha(i, j)| + 2$ time slots.

Since the above has to be repeated for every j, $(1 \leq j \leq p)$, the total number of time slots needed by the routing protocol is $p(\frac{n}{p} + p - 1) = n + p^2 - p$. Each station S($i$), $(1 \leq i \leq p)$, is awake for at most $\frac{n}{p} + 2 + \sum_{i \neq j}(|\alpha(i, j)| + 2) \leq 2(\frac{n}{p} + p)$ time slots. The following result summarizes our findings.

Lemma 2.2 *The task of permutation routing involving n items can be performed on the single-channel p-station* RN *in $n + p^2 - p$ time slots, with no station being awake for more than $2\frac{n}{p} + 2p$ time slots.*

The protocol for Lemma 2.2 is energy-efficient only when $p \leq \frac{n}{p}$. Clearly, the bottleneck is the reservation stage that takes $p^2 - p$ time. In case

$p > \frac{n}{p}$, the performance of the protocol is suboptimal. Since every conflict-free permutation routing protocol must involve a reservation stage, any improvement will have to come from completing the reservation stage more efficiently. One idea that we shall exploit, is to avoid running a reservation protocol for every station separately. Instead, intuition suggests partitioning the stations into several groups and running a single reservation stage per group.

To implement this idea, partition the p stations into $\sqrt{p}$ equal-sized groups $G(1), G(2), \ldots, G(\sqrt{p})$ such that group $G(i)$, $(1 \leq i \leq \sqrt{p})$, contains the $\sqrt{p}$ stations $\mathrm{S}((i-1)\sqrt{p}+1), \mathrm{S}((i-1)\sqrt{p}+2), \ldots, \mathrm{S}(i\sqrt{p})$. Let $\beta(i,j)$, $(1 \leq i \leq p; 1 \leq j \leq \sqrt{p})$, denote the set of items stored by station $\mathrm{S}(i)$ whose final destination is some station in group $G(j)$. In other words,

$$\begin{aligned}\beta(i,j) &= \alpha(i,(j-1)\sqrt{p}+1) \cup \\ &\quad \alpha(i,(j-1)\sqrt{p}+2) \cup \cdots \cup \alpha(i, j\sqrt{p}).\end{aligned}$$

Further, let

$$\beta(*,j) = \beta(1,j) \cup \beta(2,j) \cup \cdots \cup \beta(p,j)$$

denote the set of items whose final destination is some station in group $G(j)$. The details of the protocol are spelled out as follows.

Protocol `Square-root`

Phase 0 Partition the p stations into $\sqrt{p}$ groups $G(1), G(2), \ldots, G(\sqrt{p})$ of $\sqrt{p}$ stations each;

Phase 1 Route all the items in $\beta(*,j)$, $(1 \leq j \leq \sqrt{p})$, to the stations in group $G(j)$;

Phase 2 Route the items locally in each group $G(j)$, $(1 \leq j \leq \sqrt{p})$, to their final destination.

We refer the reader to Figure 2 for an illustration of protocol `Square-root`. Phase 0 does not take any inter-station communications since each station can determine the group to which it belongs by performing a simple algebraic computation on its ID. Phase 1 involves a reservation stage followed by a data transfer stage. For an arbitrary j, $(1 \leq j \leq \sqrt{p})$, we run the reservation protocol discussed in Section 2 that terminates in $p-1$ time slots with no station being awake for more than two time slots. At the end of the reservation stage, every station in group $G(i)$, $(1 \leq i \leq \sqrt{p})$, knows the exact time slot when it should start transmitting its $|\beta(i,j)|$ items for the stations in $G(j)$. Every station in group $G(j)$ receives exactly $\frac{n}{p}$ items from $\beta(*,j)$. Observe that the task of routing all items in $\beta(*,j)$ to the stations in group $G(j)$ takes $|\beta(*,j)| = \frac{n}{\sqrt{p}}$ time slots. Also, each station in $\mathrm{S}(i)$ in $G(j)$ is awake for $\frac{n}{p}+|\beta(i,j)|$ time slots. Each station $\mathrm{S}(i)$ $(\notin G(j))$ is awake for $|\beta(i,j)|$ time slots.

To complete Phase 1 of the protocol, the above has to be performed for all j, $1 \leq j \leq \sqrt{p}$. Thus, altogether, Phase 1 requires at most

$$\sqrt{p} \cdot \left(\frac{n}{\sqrt{p}} + p - 1\right) = n + p^{\frac{3}{2}} - \sqrt{p}$$

time slots. Further each station $\mathrm{S}(i)$ is awake for at most

$$\frac{n}{p} + \sum_{j=1}^{\sqrt{p}} (|\beta(i,j)|) + 2p = \frac{2n}{p} + 2\sqrt{p}$$

time slots.

In Phase 2, we need to route the items locally within each group $G(j)$, $(1 \leq j \leq \sqrt{p})$. Recall that group $G(j)$ contain $\sqrt{p}$ station and each of them is equally loaded with $\frac{n}{p}$ items. By Lemma 2.2, the local routing in $G(j)$ can be performed in at most $\frac{n}{\sqrt{p}} + p$ time slots with no station being awake for more than $2(\frac{n}{p} + \sqrt{p})$ time slots. For all groups combined, Phase 2 can be performed in at most $(\frac{n}{\sqrt{p}} + p) \cdot \sqrt{p} = n + p^{\frac{3}{2}}$ time slots. Thus, we have proved the following result.

Lemma 2.3 *The task of permutation routing involving n items can be performed on the single-channel p-station* RN *in at most $2(n + p^{\frac{3}{2}})$ time slots with no station being awake for more than $4(\frac{n}{p} + \sqrt{p})$ time slots.*

We can further extend and generalize the ideas above. Instead of partitioning the set $\mathcal{S}$ of stations into $\sqrt{p}$ groups of $\sqrt{p}$ stations each, we partition

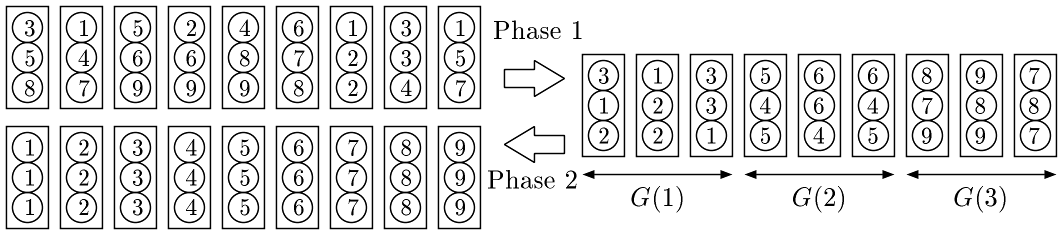

Figure 2: Illustrating the behavior of protocol `Square-root`

$\mathcal{S}$ into $p^{\frac{1}{d}}$ groups $G(1), G(2), \ldots, G(p^{\frac{1}{d}})$ of $p^{1-\frac{1}{d}}$ stations each such that for $1 \leq j \leq p^{\frac{1}{d}}$, group $G(j)$ involves the stations

$$S((j-1)p^{1-\frac{1}{d}}+1), S((j-1)p^{1-\frac{1}{d}}+2), \ldots, S(jp^{1-\frac{1}{d}}).$$

Notice that for $d = 1$ we get the naive protocol for Lemma 2.1, while for $d = 2$ we rediscover protocol `Square-root`. The new protocol is recursive as shown below.

Protocol `Route`(A, d);
if $|A| < p^{\frac{1}{d}}$ exit the protocol;
partition the $|A|$ stations in A into $p^{\frac{1}{d}}$ groups
  $G(1), G(2), \ldots, G(p^{\frac{1}{d}})$ of $\frac{|A|}{p^{\frac{1}{d}}}$ stations each;
for $j = 1$ **to** $j = p^{\frac{1}{d}}$ **do**
  route to $G(j)$ all the items whose final
    destination is some station in group $G(j)$;
endfor;
for $j = 1$ **to** $j = p^{\frac{1}{d}}$ **do**
  `Route`$(G(j), d)$
endfor

It is easy to confirm that the entire task of permutation routing amounts to the call `Route`$(\mathcal{S}, d)$. By analyzing the performance of this protocol, we have the following theorem:

Theorem 2.4 *For every integer d, $(1 \leq d \leq \lceil \log p \rceil)$, the task of permutation routing involving n items can be performed on the single-channel RN in fewer than $d(n + p^{1+\frac{1}{d}}) - 2(p-1)$ time slots, with no station being awake for more than $2d(\frac{n}{p} + p^{\frac{1}{d}})$ time slots.*

Due to stringent page limitations the proof is omitted.

Recall that we assumed $\frac{n}{p} \geq 2$. In particular, writing $d = \lceil \frac{\log p}{\log \frac{n}{p}} \rceil$, we have $1 \leq d \leq \lceil \log p \rceil$ and, consequently, $p^{\frac{1}{d}} \leq p^{\frac{\log n - \log p}{\log p}} = \frac{n}{p}$. Thus, we obtain the following corollary of Theorem 2.4.

Corollary 2.5 *The task of permutation routing involving n items can be performed on the single-channel RN in fewer than $2dn - 2(p-1)$ time slots, with no station being awake for more than $\frac{4dn}{p}$ time slots, where $d = \lceil \frac{\log p}{\log \frac{n}{p}} \rceil$.*

3 Permutation routing on the k-channel RN

We are now in a position to generalize our results to the case where k channels $\mathrm{C}(1), \mathrm{C}(2), \ldots, \mathrm{C}(k)$ are available. In practice, the number of channels allocated to RN is very limited. We model this situation by insisting that

$$k \leq \sqrt{\frac{p}{2}}. \qquad (1)$$

Consider an k-channel RN populated by p stations $S(1), S(2), \ldots, S(p)$, each storing $\frac{n}{p}$ items. We partition the p stations into k equal-sized groups $G(1), G(2), \ldots, G(k)$ such that group $G(i)$, $(1 \leq i \leq k)$, contains the $\frac{p}{k}$ stations $\mathrm{S}((i-1)\frac{p}{k}+1), \mathrm{S}((i-1)\frac{p}{k}+2), \ldots, \mathrm{S}(i\frac{p}{k})$. Note that since $k \leq \sqrt{\frac{p}{2}}$, each group contains at least $\frac{p}{k} \geq 2k$ stations. Let $\gamma(i, j)$, $(1 \leq i, j \leq k)$, denote the set of items stored by the stations in group $G(i)$ and whose destination is some station in group $G(j)$. Our permutation routing protocol consists of two phases as described next:

Protocol k-`channel-routing`

Phase 1 For every i, $(1 \le i \le k)$, assign channel $C(i)$ to group $G(i)$ and route locally the $\frac{n}{k}$ items in $G(i)$ such that:

(1.1) each station in $G(i)$ stores at most $\frac{2n}{p}$ items, and

(1.2) each station in $G(i)$ stores items from one set $\gamma(i,j)$ only.

Phase 2 Let $H(i)$, $(1 \le i \le k)$, denote the set of stations storing items whose final destination is some station in group $G(i)$.

For an illustration of the various phases of protocol k-`channel-routing` with parameters $k = 3$, $p = 18$, and $n = 54$ we refer the reader to Figure 3. A detailed discussion of the three phases of the protocol follows.

Consider a generic group $G(i)$, $(1 \le i \le k)$. Note that $G(i)$ has $\frac{p}{k}$ stations that store, collectively, $\frac{n}{k}$ items. The main task to be accomplished in Phase 1 is to route the items in

$$\gamma(i,*) = \gamma(i,1) \cup \gamma(i,2) \cup \cdots \cup \gamma(i,k)$$

so that the items in $\gamma(i,j)$ are stored by $\lceil |\gamma(i,j)| \frac{p}{2n} \rceil$ stations in such a way that conditions (1.1) and (1.2) above are satisfied. The stations that store at the end of Phase 1 items whose final destination is group $G(j)$ are said to be *agents* for group $G(j)$. In other words, $H(j)$ is the set of agents for group $G(j)$.

We now evaluate the number of stations needed to implement the scheme outlined above.

$$\begin{aligned}\sum_{j=1}^{k} \lceil |\gamma(i,j)| \frac{p}{2n} \rceil &< \sum_{j=1}^{k} (|\gamma(i,j)| \frac{p}{2n} + 1) \\ &= \frac{p}{2k} + k \le \frac{p}{k}\end{aligned}$$

Thus, since $G(i)$ contains $\frac{p}{k}$ stations, the scheme is feasible. To implement our plan, we need to run k reservation protocols similar to the one discussed in Section 2. The j-th reservation protocol has for goal to (1) determine the total number of items in $\gamma(i,j)$ as well as the number of stations in group $G(i)$ that will act as agents for group $G(j)$, $(1 \le j \le k)$; (2) for each station in $G(i)$ that stores items from $\gamma(i,j)$, the exact time slot when it should start transmitting its items.

The j-th such reservation takes $\frac{p}{k} - 1$ time slots, with no station being awake for more than two time slots. In one additional time slot the last station group in $G(i)$ will transmit on the channel the value $|\gamma(i,j)|$. Consequently, during the j-th reservation no station is awake for more than 3 time slots.

Thus, at the end of k reservation protocols every station knows the identity of the group for which it will act as an agent, as well as the exact time slot when it should wake up to pick up its share of items. Clearly, in Phase 1 each station transmits at most $\frac{n}{p}$ items and receives at most $\frac{2n}{p}$ items. In addition, it will be awake for at most $3k$ time slots during the k reservation stages. Thus, each station is awake for at most $\frac{3n}{p} + 3k$ time slots. As discussed, Phase 1 requires a total of

$$\sum_{j=1}^{k} \left(\frac{p}{k} - 1 + |\gamma(i,j)| \right) \quad = \quad \frac{n}{k} + p - k$$

time slots.

The main task to be handled in Phase 2 is to route the items stored by the stations in $H(j)$, $(1 \le j \le k)$, to their final destinations in group $G(j)$ such that each station will store exactly $\frac{n}{p}$ items. To see how this is done, recall that for every i, $(1 \le i \le k)$, (1) every station in $H(j) \cap G(i)$ knows $|\gamma(i,j)|$; (2) every station in $H(j) \cap G(i)$ knows its relative rank in $H(j) \cap G(i)$; (3) with at most one exception, every station in $H(j) \cap G(i)$ stores exactly $\frac{2n}{p}$ items destined to $G(j)$,

The last station in $H(j) \cap G(i)$ (which may have fewer than $\frac{2n}{p}$ items) will play an important role in the first stage of Phase 2. Specifically, the subgroup of $H(j)$ consisting of the last station in each $H(j) \cap G(i)$, $(1 \le i \le k)$, will run a

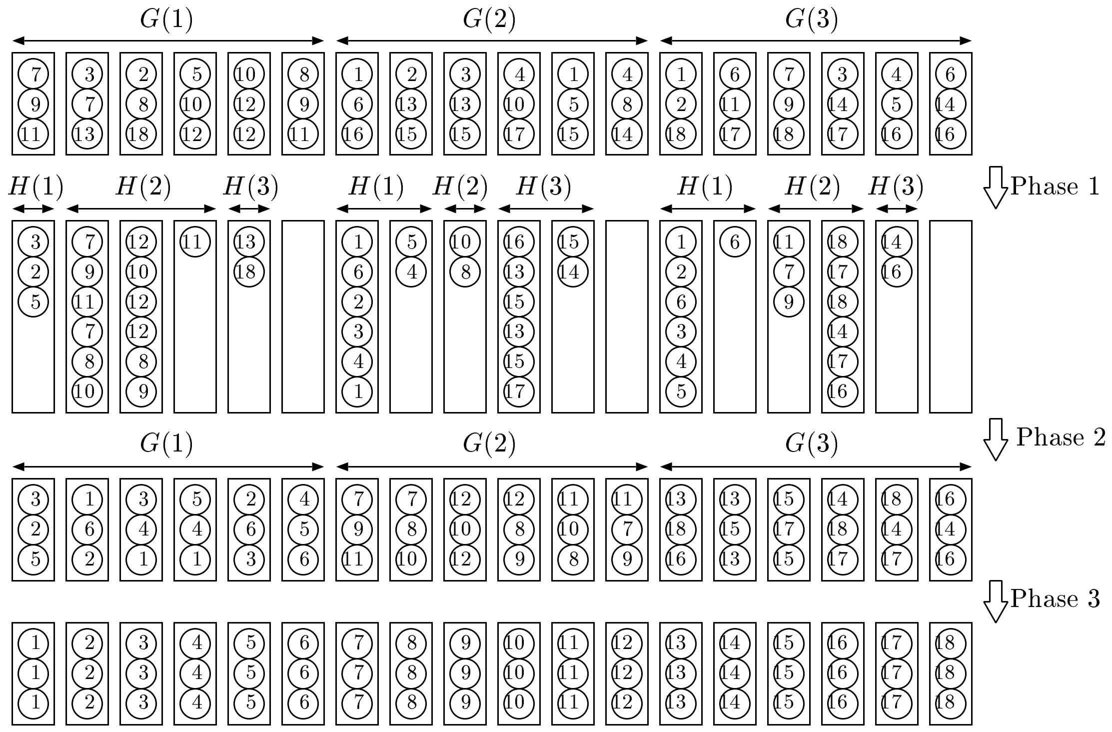

Figure 3: *Illustrating the behavior of protocol* k-`channel-routing`.

"global" reservation protocol to establish the exact time slot when the stations in $H(j) \cap G(i)$ will start transmitting on the channel. The motivation for this global reservation protocol is that at the end of Phase 1 no station in $H(j) \cap G(i)$ knows how many items were stored by the stations in $H(j) \cap G(1), H(j) \cap G(2), \ldots, H(j) \cap G(i-1)$ and, therefore, the stations in $H(j) \cap G(i)$ don't know when their time has come to transmit. Consequently, the main goal of this global reservation protocol is to allow each group $H(j) \cap G(i)$ know the number of items from $H(j)$ stored by the stations in $H(j) \cap G(t)$, with $1 \leq t \leq i-1$. Observe that once this information is available, each station knows exactly when its turn has come: this is because, as we already pointed out, within each $H(j) \cap G(i)$ there is a natural ordering of the stations. Moreover, all these stations, except for the last one have exactly $\frac{2n}{p}$ items to send. Consequently, they can easily determine the right time when they have to wake up and transmit their items.

First, it is clear that the global reservation protocol only needs to be run once and that it can be completed in $k-1$ time slots with the last station in each subgroup $H(j) \cap G(i)$ being awake for at most two time slots.

At this time we note that the number of stations in $H(j)$, $1 \leq j \leq k$, satisfies the strict inequality

$$\begin{aligned} \sum_{i=1}^{k} \lceil |\gamma(i,j)| \frac{p}{2n} \rceil &< \sum_{i=1}^{k} (|\gamma(i,j)| \frac{p}{2n} + 1) \\ &= \frac{p}{2k} + k, \end{aligned}$$

implying that, in fact, the number of stations in $H(j)$ is at most $\frac{p}{2k} + k - 1$.

To avoid cumbersome notation we let $Z(i)$, $(1 \leq i \leq \frac{p}{k})$, denote the i-th station in group $G(j)$, that is, station $\mathrm{S}((j-1)\frac{p}{k} + i)$. In order to route in an energy-efficient fashion the items whose final destination is station $Z(i)$, the stations in $H(j)$ need to run a preliminary reservation protocol. By the previous observation, the reservation protocol for station $Z(i)$, $(1 \leq i \leq \frac{p}{k})$ terminates in at most $\frac{p}{2k} + k - 2$ time slots, with no station in $H(j)$ being awake for more than two time slots. The reservation protocol is then followed by data transfer to station $Z(i)$. Of course, this takes at most $\frac{n}{p}$ time slots. The above has to be repeated $\frac{p}{k}$ times. Thus, Phase 2 requires: (1) $k-1$ time slots for the global reservation protocol (2) at most $\frac{p}{k}\left(k + \frac{p}{2k} - 2\right)$ time slots spent in the $\frac{p}{k}$ reservation protocols for the stations in group $G(j)$, and (3) at most $\frac{n}{k}$ time slots spent in transferring data items to the stations in group $G(j)$. As a consequence, Phase 2 of the protocol takes at most

$$\frac{n}{k} + p + \frac{p^2}{2k^2} - \frac{2p}{k} + k - 1 < \frac{n}{k} + 2p - 2k - 1$$

time slots with no station being awake for more than $\frac{3n}{p} + \frac{2p}{k}$ time slots. Thus, we have,

Lemma 3.1 *The task of permutation routing involving n items can be performed on the k-channel, p-station RN in* $\frac{2n}{k} + 3p - 2k - 2$ *time slots with no station being awake for more than* $\frac{6n}{p} + \frac{2p}{k} + 3k$ *time slots, provided* $k \leq \sqrt{\frac{p}{2}}$.

To improve the performance of the protocol we take a look at Phase 1 and will rely on ideas similar to those used in Section 2. Instead of partitioning $\gamma(i,*)$ into k groups we proceed to partition $\gamma(i,*)$ into $k^{\frac{1}{b}}$ groups $\delta(i,1), \delta(i,2), \ldots, \delta(i,k^{\frac{1}{b}})$ for some $b \geq 2$ and route the items such that (1) the items in $\delta(i,j)$ will be stored by a group of $\lfloor |\delta(i,j)| \frac{p}{2n} \rfloor + k^{1-b}$ stations, (2) no station stores more than $\frac{2n}{p}$ items, and (3) no station stores items from more than one $\delta(i,j)$.

Using this idea, we have the following result.

Theorem 3.2 *Given an arbitrary integer d,* $(1 \leq d \leq \log \frac{p}{k})$*, and an integer b,* $(1 \leq b \leq \log k)$*, the instance of permutation routing involving n items can be performed on the k-channel, p-station RN in* $(d+b+1)\frac{n}{k} + d(\frac{p}{k})^{\frac{d+1}{d}} + \frac{bp}{k^{1-\frac{1}{b}}} + k$ *time slots, with no station being awake for more than* $(2d + 4b - 1)\frac{n}{p} + 2d(\frac{p}{k})^{\frac{1}{d}} + 3bk^{\frac{1}{b}}$ *time slots, provided that* $k \leq \sqrt{\frac{p}{2}}$.

Due to page limitations the proof is omitted.

By choosing $d = \lceil \frac{\log \frac{p}{k}}{\log \frac{n}{p}} \rceil$ we have $(\frac{p}{k})^{\frac{1}{d}} \leq \frac{n}{p}$ and, consequently, $(\frac{p}{k})^{\frac{1+1}{d}} \leq \frac{n}{k}$. Likewise, by choosing $b = \lceil \frac{\log k}{\log \frac{n}{p}} \rceil$ we have $k^{\frac{1}{b}} \leq \frac{n}{p}$ and $\frac{p}{k^{1-\frac{1}{b}}} \leq \frac{n}{k}$. Thus, we obtain the following corollary of Theorem 3.2,

Corollary 3.3 *The task of permutation routing involving n items can be performed on the k-channel, p-station RN in at most* $(2d + 2b + 1)\frac{n}{k} + k$ *time slots, with no station being awake for more than* $(4d + 7b - 1)\frac{n}{p}$ *time slots, provided that* $k \leq \sqrt{\frac{p}{2}}$*,*

Acknowledgements

This work supported in part by NSF grant CCR-9522093, by ONR grant N00014-97-1-0526 and by Grant-in-Aid for Encouragement of Young Scientists (12780213) from Ministry of Education, Science, Sports, and Culture of Japan.

References

[1] J. L. Bordim, J. Cui, T. Hayashi, K. Nakano, and S. Olariu, Energy-efficient initialization protocols for ad-hoc radio networks, *Proc. of International Symposium on Algorithms and Computation*, Chennai, India, December 1999, LNCS 1741, Springer-Verlag, 1999, 215–224.

[2] K. Nakano, S. Olariu, and J. L. Schwing, Broadcast-Efficient Protocols for Mobile Radio Networks, *IEEE Transactions on Parallel and Distributed Systems*, 10, 12, pp.1276–1289, December 1999.

[3] M. Stemm, P. Gauthier, and D. Harada, Reducing power consumption on network interfaces in hand-held devices, *Proc. 3-rd International Workshop on Multimedia Communications*, Princeton, 1996.

[4] J. E. Wieselthier, G. D. Nguyen, and A. Ephemerides, Multicasting in energy-limited ad-hoc wireless networks, *Proc. MILCOM'98*, 1998.