Backbone-based Scheduling for Data Dissemination in Wireless Sensor Networks with Mobile Sinks

Yu Chen, Eric Fleury

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Backbone-based Scheduling for Data Dissemination in Wireless Sensor Networks with Mobile Sinks

Yu Chen — Eric Fleury

N° ????

July 2007

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

Backbone-based Scheduling for Data Dissemination in Wireless Sensor Networks with Mobile Sinks

Yu Chen , Eric Fleury

Thème COM — Systèmes communicants
Projets ARES

Rapport de recherche n° ???? — July 2007 — 19 pages

Abstract: We consider sensor networks for monitoring tasks, where data of each stimulus is required to be collected by at least one of potentially multiple, mobile nodes, called sinks. We propose a backbone-based scheduling that achieves fast data dissemination. The proposed schedules can also be used for general communication that reles on backbone structures. Theoretical analyses are presented and we show that our scheme is asymptotically optimal for networks with certain properties. We also propose for loop-free routing protocols a scheme to update routing state, which handles sink mobility with a small overhead; the routing links updated are confined within those incident to sensor nodes in paths connecting the old and new positions of sinks.

Key-words: Algorithms, Theory, Data dissemination, Backbone

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Backbone-based Scheduling for Data Dissemination in Wireless Sensor Networks with Mobile Sinks

Résumé : Pas de résumé

Mots-clés : Pas de motclef

1 Introduction

A wireless sensor network [1] is a collection of sensor nodes, which are equipped with both sensory devices and wireless transceivers and usually play a dual role as both data generators and routers. Sensor networking has many practical applications such as environment monitoring, smart spaces and robotic exploration. In this work, we focus on sensor networks for monitoring tasks. When a sensor node, called a *source*, detects a stimulus, it generates data packets, which are required to be collected by at least one of special nodes, called *sinks*, in a timely fashion. Our goal is to design an efficient dissemination scheme to propagate data packets from sources to sinks. We consider the case where stimuli happen *continuously* almost *everywhere* in the sensor field. As many works on data dissemination (e.g.[19, 14]), we are interested in the scenarios where sensor nodes stay stationary at their initial locations after having been deployed, while sinks might move around in the sensor field.

One important issue that needs to be addressed in sensor networking is interferences. Recently, the availability of a large number of low-cost sensor nodes has enable a dense deployment of sensor nodes, which provides a good coverage of the sensor field and more potential communication paths. However, interferences are also very likely in such networks if transmissions are not well scheduled. Thus how to schedule sensor nodes' activities to benefic from a high density of them without being suffered from many interferences has become critical in network design. Most existing works on data dissemination focuses on data propagation from individual sensor nodes to sinks. For example, in TTDD[19], a source builds a grid structure throughout the sensor field, along which sink queries and data packets are forwarded. In SEAD[14], a dissemination tree is built to propagate data packets from a single source to multiple mobile sinks; if multiple sources are present, multiple trees will be constructed separately. These approaches work well in the presence of a single source. However, if many stimuli happen simultaneously at different locations, data forwarding from different sources might interfere with each other, which prevents data packets from being fast delivered. Note this problem cannot be trivially solved by using a single backbone for data propagation from multiple sources, if transmissions from non-backbone nodes and backbone nodes are not well scheduled.

In fact, interference avoidance in the presence of multiple sources requires a collaboration among sensor nodes. Since contention-based schemes are energy-consuming in high-traffic networks and efficient synchronization protocols have been proposed for sensor networks (e.g. [8, 17, 10]), in this work we assume time is structured into *slots* and consider interference avoidance by scheduling nodes' transmissions. It has been proved in [6] that minimum data dissemination time problem is NP-hard. An approach with bound on latency $(\Delta - 1)D$ is proposed in [6], where Δ is the network degree and D is the diameter of the network. Very recently, a data aggregation scheduling algorithm is presented in [12] for a single sink with an $O(\Delta + D)$ latency in unit disc graphs. In [12] a data aggregation tree is first constructed from a breadth first search (BFS) tree rooted at the single sink and then nodes' transmissions are scheduled based on this data aggregation tree. Note in both approaches, wireless sensor networks are modeled as *unit disk graphs*.

Different from [6, 12], we consider a more realistic model where nodes can have different transmission and interference ranges, and our focus is on scenarios with potential *mobile, multiple* sinks. Instead of a BFS tree rooted at a single sink as in [12], our scheduling is based on a general backbone structure. We present a clustering scheme which enables fast communication between non-backbone nodes and backbone nodes; this is different from most works that rely on backbone structures, where no specific constraint is required on clustering, provided that non-backbone nodes are adjacent or close to their cluster heads. We propose schedules for the communication between non-backbone nodes and backbone nodes and the schedule for the communication in the backbone. When these schedules are applied for data dissemination, we also achieve an $O(\Delta + D)$ latency in unit disc graphs, even in the presence of potential mobile multiple sinks. Furthermore, our schedules can be used for general communication that relies on backbone structures. Backbone structures have been considered in wireless networks for routing (e.g. [4, 5, 16, 11, 3, 2, 18]) and data dissemination (e.g.[19, 14]), but less work has been done on scheduling communication based on such a structure.

Our basic idea is to restrict the number of "high-cost" hops in the path along which packets are forwarded from a source to a sink. For this purpose, a "fast backbone" is constructed and data dissemination is achieved by the communication between non-backbone nodes and backbone nodes and the communication in the backbone. By saying a backbone is "fast", we mean transmissions in this backbone can be scheduled fast; this can be achieved if each node can interfere with only a small number of nodes in the backbone. Here a "backbone" refers to a subnetwork such that each node is either included in this subnetwork or is adjacent to some node in it; such a structure guarantees that, even when the communication between non-backbone nodes and backbone nodes is expensive, the number of such expensive hops in the route from each source to a sink is bounded by a constant. In this work, we first show the existence of such a structure in networks that satisfy certain properties and we discuss a schedule for the communication in the backbone. Then we propose schedules for the communication between non-backbone nodes and backbone nodes. Analyses on the performance of these schedules are given. We achieve a latency $O(\Delta_E \Delta_I + D)$ if the network satisfies certain properties, where Δ_E (Δ_I resp.) is the maximum number of nodes that a node can transmit to (interfere with resp.) and D is the network diameter. In particular, if nodes have bounded transmission range and a node can interfere with another if and only if it can transmit to that node, the latency is $O(\Delta_E + D)$, which is asymptotically optimal.

Another issue that needs to be addressed is sink mobility. Many existing works, e.g., Directed Diffusion [13], address sink mobility by letting each sink continuously propagate its location throughout the network. This approach enables data packets to be forwarded based on the latest location information, but it also consumes energy and introduces collisions. A backbone structure can help reduce message overhead; e.g., the grid structure built in TTDD [19] enables mobile sinks to continuously receive data on the move by flooding queries within a local cell only. In our work, we further discuss how to update the directions of routing links for loop-free routing protocols in the presence of sink mobility. Approaches that handle mobility by adjusting link directions have been proposed for general mobile networks; e.g. link reversal protocol[9] and TORA[15]. Compared to general cases, topology formed by sensor nodes is relatively stable; the main changes are caused by sink mobility. Although these changes can be treated as special cases in general mobile networks, an approach designed specifically for this type of changes can be more efficient. We propose a scheme to update the directions of routing links to handle sink mobility with a small overhead. In particular, links updated are confined within those incident to nodes in paths that connect the old and new positions of sinks. Note if a link reversal protocol for general topology changes (e.g.,[9, 15]) is applied for this case, global link reversal might be involved.

We define the following denotations. Given a (directed) graph G with a set of vertices V and a set of edges E, we denote the directed link from node x to node y by $\langle x \rightarrow y \rangle$, the sets of v's outgoing and incoming neighbors by $N^+_G(x) \equiv \{y \in V | \exists \langle x \rightarrow y \rangle \in E(G)\}$, $N^-_G(x) \equiv \{y \in V | \exists \langle y \rightarrow x \rangle \in E(G)\}$. In particular, if $N^+_G(x) = N^-_G(x)$, we simplify the denotations by $N_G(x) \equiv N^+_G(x) = N^-_G(x)$. The outdegree and indegree of x are denoted by $\delta^+_G(x) \equiv |N^+_G(x)|$ and $\delta^-_G(x) \equiv |N^-_G(x)|$ respectively. Network degree is defined as $\max_{x \in V}\{\delta^+_G(x), \delta^-_G(x)\}$. Given a subset of nodes $X \subseteq V$, we denote by $N^-_G(X) \equiv \cup_{v \in X} N^-_G(v)$, $N^+_G(X) \equiv \cup_{v \in X} N^+_G(v)$ and $N_G(X) \equiv \cup_{v \in X} N_G(v)$.

2 System model & problem definition

Given a sensor network that consists of a set of sensor nodes V, we describe the network by $\mathcal{N}(V, E, I)$, where E and I are sets of links that model communication connectivity and interferences as follows:

- $\forall\ u, v \in V$, $\langle u \rightarrow v \rangle \in E$ if and only if the transmission from u can be successfully received by v when u is the only node that transmits in the network.
- $\forall\ u, v \in V$, $\langle u \rightarrow v \rangle \in I$ if and only if the transmission from u interferes with the reception at v.

The condition for a transmission to be interference-free can be formally described as follows:

Definition 1 *Given a network $\mathcal{N}(V, E, I)$, $\forall u, v \in V$, the transmission from u to v is interference-free if and only if (1) $\langle u \rightarrow v \rangle \in E$ and (2) $\forall$ node $w \neq u$ that also transmits at the same time, $\langle w \rightarrow v \rangle \notin I$.*

Here we assume $E \subseteq I$, since usually if a node is able to transmit to another node, it can also interfere with the reception at that node. Generally speaking, given two node u and v, $\langle u \rightarrow v \rangle \in E$ does not mean $\langle v \rightarrow u \rangle \in E$. Since most routing protocols require bidirectional communication of a link and only such links are considered in our work, for presentation simplification, we only include bidirectional links in E, that is, if $\langle u \rightarrow v \rangle \in E$, then $\langle v \rightarrow u \rangle \in E$. In this work, we consider general transmission and interference models. Discussions are also given for a specific model, called "disc graphs". In this model, each node v is associated with a transmission range $r(v)$, and for any two nodes $u, v \in V$, link

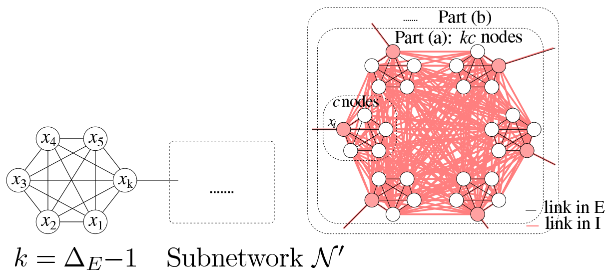
$\langle u \to v \rangle \in E$ if and only if $\|u, v\| \leq r(u)$, where $\|u, v\|$ denotes the distance between u and v. In particular, we say a network is modeled by *unit disc graph* if $r(v) = 1$ for all $v \in V$. In the sequel, we denote by G_E, called *the communication graph*, the graph with the set of vertices V and the set of edges E, and by G_I, called *the interference graph*, the graph with the set of vertices V and the set of edges I. We denote the diameter of G_E by D_E, and the degree of G_E and G_I by Δ_E and Δ_I. We call a node's degree in the communication graph (interference graph resp.) its *communication degree* (*interference degree* resp.). For presentation simplicity, we omit "G" form the subscripts in the denotations defined for G_E and G_I; for example, $N_I^-(u) \equiv N_{G_I}^-(u)$.

An important task in sensor networking is to schedule sensor node's transmissions to guarantee interference-free communication. We assume time is structured into discrete units called *slots* and consider schedules that are organized into *frames*. A frame is a minimal sequence of consecutive slots in which each link under consideration is scheduled to be interference-free, where "being interference-free" is defined in Definition 1. In this work, we consider transmissions along three different sets of links and present a schedule for each set.

Our forcus is on data dissemination in sensor networks for monitoring tasks, where sensor nodes stay static after having been deployed, while potentially multiple sinks move around in the sensor field. When a sensor node, called *source*, detects a stimulus, it generates data packets, which are required to be gathered by *at least one* of the sinks; since sinks are usually more powerful and controllable devices, information gathered by them can be aggregated later. An important performance metric is the time it takes for data dissemination. Note in the scenarios where stimuli occur continuously everywhere in the sensor field, any sensor node can be a source. Since we allow sinks to move to any location in the field, any subset of sensor nodes can be the set of nodes that connect to at least one sink. So, as sinks have to communicate with some sensor nodes to gather data packets, a data dissemination scheme should guarantee a small latency from *any sensor node* to *at least one node* in *any subset* $S \subseteq V$. Thus given $S \subseteq V$, $v \in V - S$, letting $latency(v, S) \equiv \min_{s \in S}\{$latency for v to forward packets to $s\}$, we define *dissemination latency* as $\max_{\forall S \subseteq V}\{\max_{\forall v \in V-S} latency(v, S)\}$.

In this work, we consider scenarios where, instead of gathering data continuously, applications specify *a desired data refresh interval*, that is, the interval between two times data propagation starts. We assume the desired data refresh interval is relatively large compared to the dissemination latency. To simplify the presentation, we assume it is very rare for two sinks to stay close to each other; here by saying "not being close", we mean interference does not occur when the only communication in the network is that between each sink and one of its adjacent sensor nodes; if interferences do happen, a contention-based scheme can be used to resolve the interferences.

Here we give a lower bound on the dissemination latency. Most works on scheduling problem assume $E = I$. In this case, it is easy to see the existence of a class of networks in which any data dissemination scheme has latency $\Omega(\Delta_E + D_E)$ given any values for Δ_E and D_E; an example is given in (a) of Fig. 1, where subnetwork $\mathcal{N}'$ can be any subnetwork provided that the communication graph of the whole network has degree Δ_E and diameter D_E. This lower bound is due to the competition of links in a clique. While it is obvious the lower bound $\Omega(\Delta_E + D_E)$ still holds for general case $E \subseteq I$, the lower bound $\Omega(\Delta_I + D_E)$ is not so obvious, since the size of a clique in the communication graph of a network is at most Δ_E. Below we present an example where latency is $\Omega(\Delta_I + D_E)$.



(a) $E = I$ (b) $E \subseteq I$

Figure 1: Lower bound on dissemination latency

Theorem 1 *Given any values for Δ_I, Δ_E and D_E, there exists a class of networks in which the dissemination latency for any scheme is $\Omega(\Delta_I + D_E)$.*

Proof. We construct a network as follows ((b) of Fig. 1), which has two parts. Part (a) consists of kc sensor nodes, where $c = \Delta_E - 1$ and $k = \left\lfloor \frac{\Delta_I - 1}{\Delta_E - 1} \right\rfloor$. The interference graph of part (a) is a clique of kc nodes. The communication graph of part (a) consists of k cliques, denoted by $C_1, \ldots, C_k$, each of which has c nodes; in each clique C_i, only one node, say x_i, connects to part (b). In part (a), the communication degree of a node is at most $c + 1 \leq \Delta_E$ and the interference degree is at most $kc + 1 \leq \Delta_I$. Part (b) can be any network provided that (1) the communication graph of the network has degree Δ_E and diameter D_E and the interference graph has degree Δ_I, and (2) for any $i \in [1, k]$, there is a node, say y_i, that is $D_E - 1$ hops from x_i. The existence of such a network is obvious.

It is easy to see in a slot at most one of the transmissions along links in $\{\langle v \to x_i \rangle, \forall v \in (C_i - \{x_i\}), \forall i \in [1, k]\}$ can be scheduled to be interference-free. Thus there is a node in part (a), say in clique C_i, that needs to wait for at least $k(c-1)$ slots to transmit to x_i, which needs at least $D_E - 1$ slots to reach y_i. Thus the latency is at least $k(c-1) + D_E - 1$ $= \left\lfloor \frac{\Delta_I - 1}{\Delta_E - 1} \right\rfloor (\Delta_E - 2) + D_E - 1 = \Omega(\Delta_I + D_E)$. ■

It is worth pointing out that the constructed network can be unit disc graphs. Thus this lower bound still holds for this special type of networks. Note the latency depends on the interference degree of each node, which can be much larger than its communication degree.

3 Framework of our approach

Our scheme needs to be initialized before data dissemination starts; this can be done when the network is initialized. The initialization of our scheme consists of two parts: backbone construction and computation of nodes' schedules; routing protocols used to forward packets in the backbone can also be initialized in this phase. In section 4, we present our requirements on the backbone structure, and investigate an existing backbone construction. We show that the constructed structure satisfies our requirements if the network satisfies certain properties. Then we propose schedules for three types of communication, called NB-schedules, BN-schedules and BB-schedules respectively: (1) transmission from non-backbone nodes to backbone nodes, (2) transmission from backbone nodes to non-backbone nodes, and (3) communication between backbone nodes. Our construction is based on vertex-colorings that satisfy certain constraints. The frame length of the constructed schedule is euqal to the number of colors and each slot in a frame corresponds to one color. In each slot, only nodes assigned the corresponding color are allowed to transmit. The coloring constraints are given for each type of schedule in section 4.

Data dissemination can be scheduled according to applications' desired data refresh interval. As sinks have to communicate with some sensor nodes to gather data, each sink designates an adjacent sensor node as *its representative node*, which is responsible to forward it packets. Thus given a backbone and schedules for the three types of communication, data dissemination can be accomplished in three phases. In the first phase, each non-backbone node transmits data packets to a backbone node, and, depending on applications' requirement, backbone nodes might acknowledge the reception of data packets. In this phase, an NB-schedule is used for the data transmission and one frame is sufficient. The BN-schedule can be used for sending acknowledgements, which also takes one frame. In the second phase, data packets are forwarded in the backbone by some routing protocol to at least one of the representative nodes. The routing protocol decides the path to forward the packets and a BB-schedule is used to forward packets along links in the selected path. The number of frames in this phase depends on the routing protocol and the distance from sources to sinks. Technically, any routing protocol can be used. The last phase is packet forwarding from representative nodes to sinks. Sink mobility is handled in this phase. If a sink is still adjacent to its representative node, packets received by the representative node are forwarded to the sink. Otherwise, the sink picks a new representative node from its adjacent sensor nodes, and sends a notice to its old representative node. Packets received by the old representative node are forwarded to the new representative node, which will then be forwarded to the sink. In order to handle sink mobility, we propose in section 5 a scheme to update routing state for loop-free routing protocols, which guarantees loop-free property and packet delivery in the presence of sink mobility. When the next data dissemination starts, routing paths have been updated and data packets can be sent directly to the new representative node.

Routing protocols might need to propagate its own control packets in the backbone. In order to guarantee a small dissemination latency, we separate the propagation of control packets and that of data packets, and schedule the control packets to be sent by a BB-type schedule between two consecutive data dissemination. In the sequel, we discuss backbone construction and three type of schedules in section 4. Then we present a scheme that updates routing state to handle sink mobility in section 5.

4 Backbone-based Scheduling

We represent a backbone by a set of nodes $V^b \subseteq V$ and a set of links $E^b \subseteq E$. Interferences among backbone nodes are decided by I. We define $I^b \equiv \{\langle u \to v\rangle \in I, u, v \in V^b\}$ and denote by $\mathcal{N}^b(V^b, E^b, I^b)$ the subnetwork formed by backbone nodes, by G_E^b (G_I^b resp.) the graph that consists of nodes V^b and links E^b (I^b resp.) and by Δ_E^b (Δ_I^b resp.) the degree of G_E^b (G_I^b resp.). We consider backbones with the following properties. First, in order to reduce the number of interferences, a small Δ_I^b is required. At the same time, the sparseness of a backbone should not compromise too much on the length of available communication paths, so we consider backbones that are *spanners*, that is, subgraphs where the distance between any two nodes is no more than a constant factor of that in the original network. To enable communication in the backbone, G_E^b should be connected. We also require connectivity between backbone and non-backbone nodes, that is, $\forall v \in V - V^b$, $\exists b \in V^b$, such that $\langle v \to b\rangle \in E \wedge \langle b \to v\rangle \in E$.

In this section, we present an NB-schedule for the transmission from non-backbone nodes to backbone nodes with frame length at most $\Delta_E \max_{v\in V}(\delta_I^+(v) - \delta_E(v)) + \Delta_I$, a BN-schedule for the transmission from backbone nodes to non-backbone nodes with frame length at most Δ_I, and a BB-schedule for the communication in the backbone with frame length at most $O(\Delta_E^b \Delta_I^b)$. Thus dissemination latency under our scheme is $O(\Delta_E\Delta_I + \Delta_E^b\Delta_I^b D_E)$ for general cases. As we show in disc graphs in which nodes have bounded transmission and interference ranges, Δ_E^b and Δ_E^b can be bounded by constants, the latency is $O(\Delta_E\Delta_I + D_E)$ in such networks. In particular, if each nodes has the same interference and transmission range, the latency is no more than $O(\Delta_E + D_E)$, which is asymptotically optimal.

4.1 Backbone construction and scheduling

Here we consider a *connected dominating set* for the backbone structure; note that our schedules do not rely on the specific structure. *A dominating set* is a subset of nodes such that each node is either in this set or it is adjacent to some node in this set; a dominating set is a *connected dominating set* if the subgraph induced by it is connected. The construction of such a structure has been widely studied, and approaches that construct a connected dominating set that satisfies our requirements have been proposed for *unit disc graphs* (e.g., [3, 2]). Here we briefly review such a construction and then show that our requirements are still guaranteed when it is applied in more general networks.

Typically, a connected dominating set is constructed in two phases. In the first phase, a maximal independent set V^d is selected; *an independent set* is a subset of nodes where no edge exists between any two nodes. Note a maximal independent set is also a dominating set and we call nodes in V^d dominators. In the second phase, a set of connectors, denoted by V^c, are selected to connect dominators, thus the set $V^b = V^d \cup V^c$ is a connected dominating set. The selection of V^c is usually based on the connectedness of graph $VirtG$ which consists of nodes V^d and links that connects all pairs of $u, v \in V^d$ if there is a path in G_E that connects them with at most three hops; for example, in [3], for any two adjacent nodes in $VirtG$, a path of at most three hops between them is chosen and the internal nodes in the chosen path are added to V^c.

The focus of [3] is on the approximation ratio of the minimal connected dominating set so constructed in unit disc graphs. Here we discuss this approach when it is applied in general graphs. It is easy to see G_E^b is connected and each non-backbone node is connected to a backbone node. The spanner property can be proved similarly to [2]. Here we consider the interference degree. We define the following denotations for a subset of nodes $X \subseteq V$:

- $\mathcal{U}^k(X) \equiv$ the set of nodes within k hops in G_E to some node in X. It can be defined recursively: $\mathcal{U}^0(X) \equiv X$ and $\mathcal{U}^k(X) \equiv N_E(\mathcal{U}^{k-1}(X)) \cup \mathcal{U}^{k-1}(X)$.
- $\alpha^k(X) \equiv$ the maximum size of an independent set in the subgraph of G_E that is induced by $\mathcal{U}^k(X)$.

Here we show the number of backbone nodes in X, $\left|X \cap V^b\right|$, is at most

$$\alpha^0(X) + \alpha^2(X) \max_{u \in \mathcal{U}^2(X)} \left\{\alpha^3(\{u\})\right\}$$

Note the number of backbone nodes in X is $|X \cap V^b| = \left|X \cap V^d\right| + |X \cap V^c|$. By the definition of $\alpha^k(X)$, we have $|X \cap V^d| \le \alpha^0(X)$. Now we consider $|X \cap V^c|$. For each node $c \in X \cap V^c$, there exists a pair of adjacent backbone nodes in $VirtG$ such that c is one internal node in the path that connects them; since $c \in X$ and such a path has at most 3 hops, these backbone nodes are in $\mathcal{U}^2(X)$. Since the number of backbone nodes in $\mathcal{U}^2(X)$ is at most $\alpha^2(X)$ and each backbone node b is connected to at most $\alpha^3(\{b\})$ backbone nodes in $VirtG$, the number of such pairs is at most $\frac{\alpha^2(X) \max_{u \in \mathcal{U}^2(X)} \left\{\alpha^3(\{u\})\right\}}{2}$. Since each of such pair introduces at most two nodes to V^c, so we have $|X \cap V^c| \le \alpha^2(X) \max_{u \in \mathcal{U}^2(X)} \left\{\alpha^3(\{u\})\right\}$. Thus the number of backbone nodes in X, that is, $\left|X \cap V^b\right|$, is no more than $\alpha^0(X) + \alpha^2(X) \max_{u \in \mathcal{U}^2(X)} \left\{\alpha^3(\{u\})\right\}$.

So given a backbone node $b \in V^b$, by letting X be $N_I^-(b)$ and $N_I^+(b)$ ($N_E(b)$ resp.), we can get an upper bound on the interference degree (communication degree resp.) in the backbone. Note $N_E(b) \subseteq N_I^-(b) \cap N_I^+(b)$. Thus the problem is reduced to the maximum size of an independent set in a node's neighborhood defined by G_I.

While this problem is still open to an arbitrary graph, it has been well investigated for unit disc graphs. It is proved in [2] that for every node v, the size of a maximal independent set inside the disc centered at v with radius k-units is bounded by a constant $l_k \le (2k+1)^2$. This result can be extended to general disc graphs. We show below that $l_k \le \left(\frac{2k + r_{max}}{r_{min}}\right)^2$, where r_{max} (r_{min} resp.) is the maximum (minimum resp.) transmission range of nodes in the disc with radius k-units centered at v. Note for any two nodes u and v, $(\langle u \to v\rangle \in E \wedge \langle v \to u\rangle \in E)$ if and only if we have $\|u, v\| \ge \min\{r(u), r(v)\}$.

Lemma 2 *Given a network $\mathcal{N}(V, E, I)$ modeled by a disc graph and any maximal independent set S, for any node $v \in V$, the number of nodes in S that are inside the disc centered at*

v with radius k-units is at most $l_k \leq \left(\frac{2k+r_{max}}{r_{min}}\right)^2$, where r_{max} (r_{min} resp.) is the maximum (minimum resp.) transmission range of nodes in this disc.

Proof. For any $u \in S$, we consider the disc, denoted by $D(u)$, centered at u with radius $\frac{1}{2}\min\{r(u') : \|u', u\| \leq r(u), u' \in S\}$, which is at most $\frac{r(u)}{2}$ since $\|u, u\| = 0$. For any two dominator $u, v \in S$, we now show $D(u)$ and $D(v)$ are disjoint. Without loss of generality, we assume $r(u) \geq r(v)$. If $D(u)$ and $D(v)$ overlap, then $\|u, v\| \leq \frac{1}{2}\min\{r(u') : \|u', u\| \leq r(u)\} + \frac{1}{2}\min\{r(v') : \|v', v\| \leq r(v)\} \leq \frac{r(u)+r(v)}{2} \leq r(u)$. Since $\|u, v\| \leq r(u)$, we have $\frac{1}{2}\min\{r(u') : \|u', u\| \leq r(u)\} \leq \frac{1}{2}r(v)$. So we have $\|u, v\| \leq \frac{r(v)}{2} + \frac{r(v)}{2} = r(v)$. Thus we have $\|u, v\| \leq r(v) \leq r(u)$, which contradicts to the condition for two nodes to be in an independent set. Given any disc with radius k-unit, the disc $D(u)$ of all the dominators in this area is within a disc with radius $k + \frac{r_{max}}{2}$. So the the number of dominators is at most $\frac{\pi\left(k+\frac{r_{max}}{2}\right)^2}{\pi\left(\frac{r_{min}}{2}\right)^2} = \left(\frac{2k+r_{max}}{r_{min}}\right)^2$ ■

Thus if each node v can only interfere with nodes within certain range, say $i(v)$ units, its interference degree in the backbone is determined by the transmission ranges of nodes within $i(v) + 5$ units, which are only related to individual nodes and are independent of the size and density of the network. Thus the interference degree of the backbone is bounded by a constant if nodes have bounded interference and transmission ranges. Even in networks where nodes have widely varied transmission ranges, as long as nodes close to each other have similar properties, the interference degree can still be small.

With a backbone in place, we consider a BB-schedule to guarantee interference-free transmissions in E^b. Constraints on a coloring for a BB-schedule are given below.

Definition 2 *Given a network $\mathcal{N}(V, E, I)$ and a backbone structure $\mathcal{N}^b(V^b, E^b, I^b)$, a coloring for a* BB*-type schedule is a non-negative integral function color() on V^b that satisfies: $\forall\langle u \to v\rangle \in E^b$, (1) color$(u) \neq$ color(v) and (2) color$(u) \neq$ color(w) for all $w \neq u$ such that $\langle w \to v\rangle \in I^b$.*

By comparing this constraint to the definition of an $L_S(1,1)$-coloring on graph G in [7], we can find this coloring is essentially an $L_{G_E^b}(1,1)$-coloring on graph G_I^b. A heuristics for an $L_S(1,1)$-coloring on graph G has been given in [7], which can be proved to use $O(\Delta_S \Delta_G)$ colors. Thus the frame length of a BB-schedule constructed based on this heuristics is $O(\Delta_I^b \Delta_E^b)$. Note $O(\Delta_I^b \Delta_E^b)$ is bounded by a constant in networks modeled by disc graphs where transmission and interference range are bounded by constants.

4.2 Schedules for communication between backbone nodes and non-backbone nodes

In this section, we aim to schedule sensor nodes' activities to guarantee communication between backbone nodes and non-backbone nodes. In order to achieve this goal, we first cluster the network by allocating each backbone node a set of adjacent non-backbone nodes. Then we propose schedules that guarantee interference-free communication between a backbone node and the non-backbone nodes allocated to it.

We represent the clustering by a function $\chi()$:$V^b \to \{X : X \subseteq V - V^b\}$, such that, $\forall b \in V^b$, $\chi(b) \subseteq V - V^b$ is the set of non-backbone nodes that are allocated to b; we call b the cluster head of the non-backbone nodes in $\chi(b)$. The requirements on $\chi()$ are (1) $\cup_{b \in V^b} \chi(b) = V - V^b$, (2) $\forall b, b' \in V^b$, $b \neq b'$, $\chi(b) \cap \chi(b') = \emptyset$, and (3) $\forall b \in V^b$, $\forall u \in \chi(b)$, $\langle u \to b \rangle \in E \wedge \langle b \to u \rangle \in E$. A computation of $\chi()$ will be presented in our scheduling scheme.

Before we present our scheduling scheme, we give the constraints on the colorings based on which our schedules are constructed. Given $\chi()$, the constraints on the coloring for an NB-schedule are given below, which guarantees interference-free transmissions from each non-backbone node $u \in \chi(b)$ to its cluster head b. Intuitively, when u transmits, no node that is able to interfere with the reception at b is allowed to transmit. Note this is a schedule on the transmissions of *non-backbone nodes*.

Definition 3 *Given a network $\mathcal{N}(V, E, I)$, a backbone $\mathcal{N}^b(V^b, E^b, I^b)$ and a clustering $\chi()$, a coloring for an* NB*-schedule is a non-negative integral function color() on $V - V^b$ that satisfies: $\forall b \in V^b$, $\forall u \in \chi(b)$, $\forall\ w \in V - V^b$ such that $w \neq u$ and $\langle w \to b \rangle \in I^b$, we have $color(u) \neq color(w)$.*

We give below the constraints on the coloring for a BN-type schedule, which guarantees interference-free transmissions from backbone node b to each non-backbone node in $\chi(b)$. Intuitively, two backbone nodes cannot transmit simultaneously if one of them can interfere with the reception at some non-backbone node allocated to the other. Note this is a schedule on the transmissions of *backbone nodes*.

Definition 4 *Given a network $\mathcal{N}(V, E, I)$, a backbone $\mathcal{N}^b(V^b, E^b, I^b)$ and a clustering $\chi()$, a coloring for a* BN*-schedule is a non-negative integral function color() on V^b that satisfies: $\forall b, b' \in V^b$, $color(b) \neq color(b')$ if $\exists u \in \chi(b)$ such that $\langle b' \to u \rangle \in I^b$.*

A straightforward solution for both types of coloring is a two-hop vertex coloring on G_I; the schedule so constructed is an entirely interference-free schedule. However, due to the best known upper bound on the number of colors used by a two-hop vertex coloring, the frame length of such a schedule is $O\left(\Delta_I^2\right)$. We observe that, compared to general scheduling problems, the constraints on such schedules are much simpler. Here we present coloring schemes specifically for such schedules and construct an NB-schedule with $\Delta_E \max_{v \in V}(\delta_I(v) - \delta_E(v)) + \Delta_I$ frame length and a BN-schedule with Δ_I frame length. Note both have Δ_E frame length in the special case $E = I$, Our scheduling involves a computation of $\chi()$. This is different from most works that rely on backbone structures, where no specific constraint is required on clustering, provided that non-backbone nodes are adjacent or close to their cluster heads. Here we show that latency can be reduced if clustering is well designed.

We assume each backbone node b is assigned a globally unique priority, $prior(b)$; such a priority can be computed according to some properties, e.g., node identification. Our scheduling scheme is presented below. For presentation simplification, we use some abstract primitives without going into details of message exchanges. For example, the if-condition in line I-1 can be implemented by node u sending b the priorities of all its backbone neighbors; in line II-8, when we say a backbone node b assigns a color, say c, to a non-backbone u node by statement $color_{\mathrm{NB}}(u) = c$, we mean b sends u a message that includes the assigned color, and u sets its local variable $color_{\mathrm{NB}}(u)$ accordingly. Note here we use the denotation $N_E(u)$ instead of $N_E^-(u)$ or $N_E^+(u)$, due to the assumption that if $\langle u \to v \rangle \in E$ then $\langle v \to u \rangle \in E$. There are three parts in this scheme, each of which will be explained in the sequel of this section.

Backbone-based Scheduling for Data Dissemination

Local variable at each backbone node b:

- $\chi(b)$: a set of non-backbone nodes, initially $\chi(b) = \emptyset$.
- $wait_{\mathrm{NB}}(b)$: the set of backbone nodes such that, b starts coloring nodes in $\chi(b)$ for an NB-schedule only after all the nodes in $wait_{\mathrm{NB}}(b)$ have done with this coloring.
- $wait_{\mathrm{BN}}(b)$: the set of backbone nodes such that, b starts coloring for a BN-schedule only after all the nodes in $wait_{\mathrm{BN}}(b)$ have done with this coloring.
- $color_{\mathrm{BN}}(b)$: coloring for a BN-schedule

Local variable at each non-backbone node u:

- $color_{\mathrm{NB}}(u)$: coloring for an NB-schedule

Code on a backbone node b:

// Part I: Compute $\chi(b)$ for each backbone node b.
I-0 **for** each of non-backbone node $u \in N_E(b)$ **do**
I-1   **if** $(prior(b) == \max\{prior(b') : b' \in N_E(u)\})$ **then**
I-2   add u to $\chi(b)$;
I-3   **endif**
I-4 **endfor**

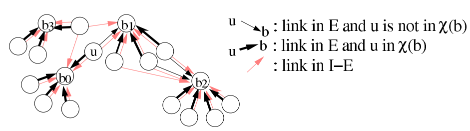
// Part II: Coloring for NB-type scheduling
II-0 $wait_{\mathrm{NB}}(b) = \{\ b' \in V^b\!: ((N_I^-(b') \cap N_E(b) \neq \emptyset) ||$
$(N_E(b') \cap N_I^-(b) \neq \emptyset))$
$\wedge (prior(b') > prior(b))\}$;
II-1 **while** $(wait_{\mathrm{NB}}(b) \neq \emptyset)$ **do**
II-2   **Upon receipt of** a message indicating $b' \in V^b$ is done with NB-coloring
II-3   **if** $(b' \in wait_{\mathrm{NB}}(b))$ **then** remove b' from $wait_{\mathrm{NB}}(b)$;
II-4 **endwhile**
II-5 **if** $(wait_{\mathrm{NB}}(b) == \emptyset)$ **then**
II-6   **for** each node $u \in \chi(b)$ **do**
II-7 $ib(u)=\{b' \in V^b :$
$(b' \in (N_I^+(u) - N_E(u)) \wedge (prior(b') > prior(b))\}$;
II-8 $color_{\mathrm{NB}}(u)$ = a color that has not been assigned to the non-backbone nodes in $\cup_{b' \in ib(b)} \chi(b') \cup N_I^-(b)$;
II-9   **endfor**

II-10 broadcast in $N_E(N_I^-(b)) \cup N_I^+(N_E(b))$ that b is done with NB-coloring.
II-11 **endif**

// Part III: Coloring for BN-type scheduling
III-0 $f(b) = \{b' \in V^b : \exists u \in \chi(b'), u \in N_I^+(b)\}$
III-1 $wait_{\mathrm{BN}} = \{b' \in V^b : b' \in f(b) \wedge prior(b') > prior(b)\}$
III-2 **while** $(wait_{\mathrm{BN}} \neq \emptyset)$
III-3 **Upon receipt of** a message indicating $b' \in V^b$ is done with BN-coloring
III-4 **if** $(b' \in wait_{\mathrm{BN}}(b))$ **then** remove b' from $wait_{\mathrm{BN}}(b)$;
III-5 **endwhile**
III-6 **if** $(wait_{\mathrm{BN}}(b) == \emptyset)$ **then**
III-8 $color_{\mathrm{BN}}(b) =$ a color that has not been assigned to nodes in $f(b)$;
III-9 broadcast in $f(b)$ that b is done with BN-coloring;
III-7 **endif**

4.2.1 Computation of $\chi()$

In Part I of our scheme, each backbone node b computes the set of non-backbone nodes, $\chi(b)$, that will be scheduled to transmit to and receive from b. Intuitively, each non-backbone node picks the backbone node that has the highest priority among those in its neighborhood to be its cluster head. For example, in Figure 2, node u has two backbone neighbors b_0 and b_1; the one with higher priority, b_0, is selected as its cluster head and $u \in \chi(b_0)$. It is easy to verify the $\chi()$ so computed satisfied the three properties we require on clustering.



$prior(b_0) > prior(b_1) > prior(b_2) > prior(b_3)$

Figure 2: Part I of our scheduling

4.2.2 Construction of nb-schedules

Part II computes a coloring for a NB-schedule. In such a schedule, all the backbone nodes stay in receive mode and transmissions of non-backbone nodes are scheduled based on the coloring function $color_{\mathrm{NB}}()$ defined on $V - V^b$. Given $\chi()$ and any non-backbone node $u \in V - V^b$, $color_{\mathrm{NB}}(u)$ is assigned by its cluster head, that is, $b \in V^b$ such that $u \in \chi(b)$. Each backbone node b colors non-backbone nodes in $\chi(b)$ after all the backbone nodes in $wait_{\mathrm{NB}}(b)$ are done with this coloring. Intuitively, $wait_{\mathrm{NB}}(b)$ (line II-0) is the set of backbone nodes $\{b'\}$ that has a higher priority and satisfies that (1) transmissions from a neighbor of

b can interfere with the reception at b', that is, $N_I^-(b') \cap N_E(b) \neq \emptyset$, or (2) transmissions from a neighbor of b' can interfere with the reception at b, that is, $N_E(b') \cap N_I^-(b) \neq \emptyset$.

Intuitively, each time a backbone node b colors a non-backbone node $u \in \chi(b)$ (lines II-6 to II-9), it guarantees the following two properties. First, for each backbone node b' that has a higher priority, the receptions at b' from each non-backbone node in $\chi(b')$ is interference-free. This can be achieved by assigning u a color that is not used by the non-backbone nodes in $\chi(b')$ for all the higher-priority backbone nodes with which u can interfere, that is, $\{b' \in V^b : \big(b' \in N_I^+(u)\big) \wedge (prior(b') > prior(b))\}$ (see Figure 3, where $prior(b_i) > prior(b)$, $i = 1, \ldots, k$); as we shown in theorem 3, this set is exactly the set $ib(u)$ computed in line II-7. The second property is that the receptions at b from the non-backbone nodes in $\chi(b)$ that have been colored are guaranteed to be interference-free; this is achieved by assigning u a color that are not used by those in $N_I^-(b)$.

Theorem 3 *Part II of our scheduling scheme constructs an* NB*-schedule with frame length at most* $\Delta_E \max_{v\in V}(\delta_I^+(v) - \delta_E(v)) + \Delta_I$.

Proof. Note $\forall b \in V^b$, $\forall u \in \chi(b)$, $\{b' \in V^b : \big(b' \in N_I^+(u)\big) \wedge (prior(b') > prior(b))\} = \{b' \in V^b : \big(b' \in N_I^+(u) - N_E(u)\big) \wedge (prior(b') > prior(b))\}$, because $\{b' \in V^b : (b' \in N_E(u)) \wedge (prior(b') > prior(b))\} = \emptyset$ as b is the backbone node with the highest priority in $N_E(u)$ (lines I-1 to I-3). Now we consider $\forall b \in V^b$, $\forall u \in \chi(b)$, $\forall w \in V - V^b$ such that $w \in N_I^-(b)$. Let $b' \in V^b$ be the backbone node such that $w \in \chi(b')$. There are three cases.

- If $b' == b$, then by lines II-6 to II-9, $color_{\mathrm{NB}}(u) \neq color_{\mathrm{NB}}(w)$ since $\{w, u\} \subseteq \chi(b) \subseteq N_I^-(b)$.
- If $prior(b') > prior(b)$, then $b' \in wait_{\mathrm{NB}}(b)$, since $w \in N_I^-(b)$ and $w \in \chi(b') \subseteq N_E(b')$. So when u is colored by b, w has been colored by b'. By line II-8, $color_{\mathrm{NB}}(u)$ is assigned a color that is not used by nodes in $N_I^-(b)$, including $color_{\mathrm{NB}}(w)$. So $color_{\mathrm{NB}}(u) \neq color_{\mathrm{NB}}(w)$.
- If $prior(b') < prior(b)$, then $b \in wait_{\mathrm{NB}}(b')$, since $w \in N_I^-(b)$ and $w \in \chi(b') \subseteq N_E(b')$. So when w is colored by b', u has been colored by b. As $w \in N_I^-(b)$, we have $b \in \{b'' \in V^b : (b'' \in N_I^+(w)) \wedge prior(b'') > prior(b')\} = ib(w)$. By line II-8, $color_{\mathrm{NB}}(w)$ is assigned a colored that is not used by nodes in $\cup_{b'' \in ib(w)} \chi(b'')$, including $color_{\mathrm{NB}}(u)$ since $b \in ib(w)$ and $u \in \chi(b)$. So $color_{\mathrm{NB}}(u) \neq color_{\mathrm{NB}}(w)$.

Now we consider the number of used colors. For any backbone node b and non-backbone node $u \in \chi(b)$, we have $|\cup_{b' \in ib(u)} \chi(b')| \leq \Delta_E |ib(u)| \leq \Delta_E \max_{v \in V}(\delta_I^+(v) - \delta_E(v))$. When node b colors u in line II-7, the number of colors that are taken by nodes in $\cup_{b' \in ib(u)} \chi(b')$ and those in $N_I^-(b)$ that have been colored is at most $\Delta_E \max_{v \in V}(\delta_I^+(v) - \delta_E(v)) + \Delta_I - 1$. Thus there exists a color in $[0, \Delta_E \max_{v \in V}(\delta_I^+(v) - \delta_E(v)) + \Delta_I - 1]$ that can be assigned to u. ■

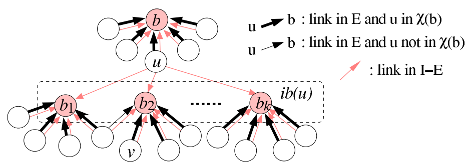


Figure 3: Part II of our scheduling

4.2.3 Construction of bn-schedules

In Part III, a coloring for BN-type scheduling is proposed. The idea is very straightforward. Each backbone node b computes the set of backbone nodes $f(b) = \{b'\}$ such that transmissions from b might interfere with receptions of nodes in $\chi(b')$. Then a coloring is applied to guarantee b and b' are not assigned the same color. Since $|f(b)| = |\{b' \in V^b : \exists u \in \chi(b'), u \in N_I^+(b)\}| \leq |N_I^+(b)|$, Δ_I colors are sufficient for such a coloring.

Theorem 4 *Part III of our scheduling constructs a* BN*-schedule with frame length at most* Δ_I*.*

5 Sink mobility

In our approach, each sink is assigned an adjacent sensor node as its representative node, which is responsible to forward it data packets. Routing protocols used to propagate packets in the backbone should guarantee that each packet is sent to at least one of the representative nodes. Technically, any protocol that satisfies this condition can be used. Here we propose a link update scheme to handle sink mobility with a small overhead for *loop-free* protocols, that is, protocols under which routing links form a *directed acyclic graph* (DAG). For presentation simplicity, we describe the routing state of such a protocol by a subset of neighbors, $to\_hops(v) \subseteq N_E^b(v)$, at each backbone node $v \in V^b$, indicating that, upon receipt a packet, v will forward the packet to nodes in $to\_hops(v)$. Thus the set of routing links can be represented as $\bigcup_{v \in V^b}(\langle v \to u\rangle : u \in to\_hops(v))$.

Given a network $N(V, E, I)$ with a set of sinks S and a backbone $N^b(V^b, E^b, I^b)$, for each sink $s \in S$, we denote by $rep(s, T) \in V^b$ the representative node of sink s at a given time T and by $R(T) = \{rep(s, T) : s \in S\}$ the set of all the representative nodes at time T. Our focus is on how to update the directions of the routing links to guarantee that packets sent by each backbone node are forwarded to at least one of the representative nodes without violating the loop-free condition in the presence of sinks mobility. We formally present this condition below:

Definition 5 *Given a network $N(V, E, I)$ with a set of sinks S and a backbone $N^b(V^b, E^b, I^b)$, we define the following two conditions for a given time T:*

- LOOP-FREE*: the subgraph of G_E^b induced by the routing links is a DAG.*

- Connectivity-to-Sinks: *$\forall v \in V^b - R(T)$, $\exists r \in R(T)$ such that there is a directed path from v to r in the subgraph of G_E^b induced by the routing links.*

As sinks move around, these conditions might be invalidated due to the changes in the composition of set $R(T)$. Assuming these conditions are true at time T, we present below a scheme to update routing state when the condition is invalidated, say, at time T'. We denote the set of sinks that have changed their representative nodes since T by $S^\star = \{s \in S : rep(s, T') \neq rep(s, T)\}$. For each sink $s \in S^\star$, let $Path(s)$ be a path in the backbone that connects the old representative node, $rep(s, T)$, and the new representative node, $rep(s, T')$. Here we assume each sink s has a unique identification, $id(s)$. Nodes in the paths $\{Path(s), s \in S^\star\}$ or those adjacent to some node in these paths update their routing state as follows, where complications arise when the paths $Path(s)$ for different sink s intersect. Intuitively, a representative node will not relay any data packet (lines I-1 to I-3 and line II-1). For each non-representative node u in these path, if u is adjacent to some representative node, it forwards packets to the adjacent representative nodes (line I-4 to I-6); otherwise it forwards packets along the path associated with the sink that has the largest identification among those in $\{s \in S^\star : u \in Path(s)\}$ (lines I-7 to I-8). For each non-representative node that is adjacent to some node in these paths, it forwards packets to the adjacent nodes in these paths (line II-2).

---

for $\forall u \in \bigcup_{s \in S^\star} Path(s)$,
I-1 **if** $(u \in R(T'))$ **then**
I-2 $\quad to\_hops(u) = \emptyset$;
I-3 **else**
I-4 $\quad$ **if** $((N_E^b(u) \cap R(T')) \neq \emptyset)$ **then**
I-5 $\quad\quad to\_hops(u) = N_E^b(u) \cap R(T')$;
I-6 $\quad$ **else**
I-7 $\quad\quad$ let u' be a node such that $\langle u \rightarrow u' \rangle \in Path(s')$,
$\quad\quad\quad$ where $s' = \max\{id(s) : s \in S^\star, u \in Path(s)\}$;
I-8 $\quad\quad to\_hops(u) = \{u'\}$;
I-9 $\quad$ **endif**
I-10 **endif**

for $\forall u \in N_E^b \left(\bigcup_{s \in S^\star} Path(s)\right) - \bigcup_{s \in S^\star} Path(s)$,
II-1 **if** $(u \in R(T'))$ **then** $to\_hops(u) = \emptyset$;
II-2 **else** add $N_E^b(u) \cap \bigcup_{s \in S^\star} Path(s)$ to $to\_hops(u)$.

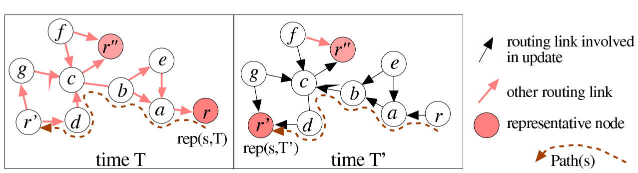


Figure 4: Link update on backbone

An example is given in Figure 4. As for nodes in $Path(s)$, we have $to\_hops(r') = \emptyset$ (line I-2) and $to\_hops(d) = \{r'\}$ $to\_hops(c) = \{r"\}$ (line I-5), $to\_hops(b) = \{c\}$, $to\_hops(a) = \{b\}$, $to\_hops(r) = \{a\}$ (line I-8). As for nodes adjacent to nodes in $Path(s)$, we have $to\_hops(r") = \emptyset$ (line II-1) and $to\_hops(g) = \{r', c\}$, $to\_hops(f) = \{c, r"\}$, $to\_hops(e) = \{a, b\}$ (line II-2). It is obvious that only nodes in $\bigcup_{s \in S^\star} Path(s)$ and $N_E^b\left(\bigcup_{s \in S^\star} Path(s)\right)$ update their routing state. We state below that the resulting routing state satisfies the LOOP-FREE and CONNECTIVITY-TO-SINKS conditions.

Theorem 5 *Given a network $N(V, E, I)$ with a set of sinks S and a backbone $N^b(V^b, E^b, I^b)$, if the routing state satisfies the conditions in Definition 5 at time T, then after the routing state is updated at time T', these conditions are still guaranteed.*

Proof. First we prove no cycle is formed by the routing links in the resulting state. We assume in contradiction that a cycle exists, $cir = \{c_0, \ldots, c_{k-1}\}$, where $c_{(i+1)\%k} \in to\_hops(c_i), i = 0, \ldots, k-1$. Note no node in cir is a representative node because of $to\_hops(r) = \emptyset$ for any representative node r. Now we show $cir \subseteq \bigcup_{s \in S^\star} Path(s)$ or $cir \subseteq V^b - \bigcup_{s \in S^\star} Path(s)$: if it is not true, without loss of generality, we assume $c_i \in \bigcup_{s \in S^\star} Path(s)$ and $c_{i+1} \in V^b - \bigcup_{s \in S^\star} Path(s)$; note $to\_hops(c_i) \subseteq R(T')$ (line I-5) or $to\_hops(c_i) \subseteq \bigcup_{s \in S^\star} Path(s)$ (lines I-7 to I-8), so we have $c_{i+1} \notin to\_hops(c_i)$, which is a contradiction. Since the LOOP-FREE condition was true at time T and no link that connects two nodes in $V^b - \bigcup_{s \in S^\star} Path(s)$ is added, we have $cir \subseteq \bigcup_{s \in S^\star} Path(s)$. Note cir cannot be a circle if there is $i \in [0, k-1]$ such that c_i is adjacent to a representative node, since otherwise c_{i+1} is a representative node (line I-5). Thus $to\_hops(c_i)$ is computed by lines I-7 to I-8 . Given c_i, $i \in [0, k-1]$, let s_i be the sink such that $id(s_i) = \max\{s \in S^\star : c_i \in Path(s)\}$. We have $\langle c_i \to c_{(i+1)\%k}\rangle \in Path(s_i)$. Since c_{i+1} is a non-representative node, $Path(s_i)$ does not end at node c_{i+1} and there is a link $\langle c_{i+1} \to c'\rangle \in Path(s_i)$ for some neighbor c' of c_{i+1}. By the computation of $to\_hop(c_{(i+1)\%k})$, we have $id(s_i) \leq id(s_{i+1})$. Thus we have $id(s_0) \leq id(s_1) \leq \ldots \leq id(s_{i-1}) \leq id(s_0)$, that is, $Path(s_0)$ is a circle. Contradiction! So we proved no cycle is formed by the routing links in the resulting state.

Now we prove that, given any non-representative node, there is a path from it to some representative node. Note each non-representative node has at least one outgoing routing link. Since there is no cycle and the number of nodes is finite, by following these outgoing links, we can eventually reach a representative node. ∎

6 Conclusion

We propose backbone-based schedules that achieve fast data dissemination. The proposed schedules can also be used for general communication that rely on a backbone structure. Theoretical analyses are given and we show that the proposed scheme is asymptotically optimal for disc graphs where nodes have bounded transmission range and nodes can interfere with each other if and only if they can communicate with each other. We further propose an approach to update routing state to handle sink mobility with a small overhead for loop-free protocols.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[2] K. Alzoubi, X. Li, Y. Wang, P. Wan, and O. Frieder. Geometric spanners for wireless ad hoc networks. *IEEE Trans. Para. Distrib. Syst.*, 14(4):408–421, 2003.

[3] K. Alzoubi, P. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *MobiHoc '02: Proc. 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002.

[4] D. J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Trans. Commun.*, 29(11):1694– 1701, 1981.

[5] P. Bose, P. Morin, I. Stojmenovicacute, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[6] X. Chen, X. Hu, and J. Zhu. Minimum data aggregation time problem in wireless networks. In *Proc. 1st Int'l Conference on Mobile Ad hoc and Sensor Networks*, 2005.

[7] Y. Chen and E. Fleury. A distributed policy scheduling for wireless sensor networks. In *Proc. IEEE INFOCOM*, 2007.

[8] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proc. 5th symposium on Operating systems design and implementation (OSDI)*, 2002.

[9] E. Gafni and D. Bertsekas. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. *IEEE Trans. Commun.*, COM(29):11–18, 1981.

[10] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. Srivastava. Estimating clock uncertainty for efficient duty-cycling in sensor networks. In *SenSys '05: Proc. 3rd international conference on Embedded networked sensor systems*, 2005.

[11] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *MobiHoc '01: Proc. 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001.

[12] S. Huang, P. Wan, C.Vu, Y. Li, and F. Yao. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In *Proc. IEEE INFOCOM*, 2007.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom'00: Proc. 6th annual international conference on Mobile computing and networking*, 2000.

[14] H. Kim, T. Abdelzaher, and W. Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *SenSys '03: Proc. 1st international conference on Embedded networked sensor systems*, 2003.

[15] V. Park and M. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. IEEE INFOCOM*, 1997.

[16] M. Seddigh, J. Gonzalez, and I. Stojmenovic. RNG and internal node based broadcasting algorithms for wireless one-to-one networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(2):37–44, 2001.

[17] W. Su and I. F. Akyildiz. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 13(2):384–397, 2005.

[18] Y. Wang, W. Wang, and X. Li. Distributed low-cost backbone formation for wireless ad hoc networks. In *MobiHoc '05: Proc. 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005.

[19] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *MobiCom '02: Proc. 8th annual international conference on Mobile computing and networking*, 2002.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399