Interval Routing Schemes allow Broadcasting with Linear Message-Complexity*

[Extended Abstract]

Pierre Fraigniaud† Cyril Gavoille‡ Bernard Mans§

ABSTRACT

The purpose of *compact routing* is to provide a labeling of the nodes of a network, and a way to encode the routing tables so that routing can be performed efficiently (e.g., on shortest paths) while keeping the memory-space required to store the routing tables as small as possible. In this paper, we answer a long-standing conjecture by showing that compact routing can also help to perform distributed computations. In particular, we show that a network supporting a shortest path *interval routing scheme* allows to broadcast with an $O(n)$ message-complexity, where n is the number of nodes of the network. As a consequence, we prove that $O(n)$ messages suffice to solve leader-election for any graph labeled by a shortest path interval routing scheme, improving therefore the $O(m+n)$ previous known bound.

Keywords: Compact routing, Interval routing, Broadcasting, Distributed computing.

1. INTRODUCTION

This paper addresses a problem originally formulated by D. Peleg, and that can be informally summarized as follows: "Does networks supporting shortest path compact routing schemes present specific ability in term of distributed computation? E.g., broadcasting, leader-election, etc." This paper answers by the affirmative, by showing that n-node networks supporting interval routing schemes [26, 27] (IRS for short), allow broadcasting with $O(n)$ message-complexity.

More formally, a network $G=(V,E)$ (in this paper, by *network*, we will always mean a connected undirected graph without loop and multiple edge) supports an IRS if the nodes of that network can be labeled from 1 to $n=|V|$ in such a way that the following is satisfied: given any node $x \in V$ of degree d, there is a set of d intervals $I_1,\dots,I_d$ of $\{1,\dots,n\}$, one for each edge $e_1,\dots,e_d$ incident to x, such that (1) $V \setminus \{x\} \subseteq \cup_{i=1}^{d} I_i$, (2) $I_i \cap I_j = \emptyset$ for every $i \neq j$, and (3) $y \in I_i$ implies that there is a shortest path from x to y passing through the edge e_i. So, IRS is a shortest path routing table having the property that the set of destination-addresses using a given link is a consecutive set of integers. IRS is a famous technique for the purpose of compact routing since a network of maximum degree Δ and supporting an IRS has its routing table of size $O(\Delta \log n)$ bits, to be compared with the $\Theta(n \log \Delta)$ bits of a routing table returning, for every destination, the output port corresponding to that destination. For more about IRS, we refer to [6, 10, 17, 19, 21, 22], and to [16] for a recent survey. For more about compact routing in general, we refer to [12, 13, 14, 18, 20, 25].

On the other hand, broadcasting is the information dissemination problem which consists, for an arbitrary node of a network, to send a same message to all the other nodes. The message-complexity of broadcasting lies between $\Omega(n)$ and $O(m)$, $m=|E|$, since, on one hand, the reception of the message by every node but the source requires at least $n-1$ messages, and, on the other hand, broadcasting can always be performed by flooding the network, that is, upon reception of the message, every node forwards that message through all its incident edges (apart the one on which it has received the message). Better upper and lower bounds can be derived as a function of the knowledge of the nodes of the network (e.g., see [1, 2, 4]), and of the maximal size of the message-headers (e.g., see [24]). In the following, the only knowledge of every node is its label in some IRS, and the intervals attached to its incident edges in the same IRS. The size of each message-header transmitted by our broadcasting protocol is $\lceil \log_2 n \rceil$ bits.

The relationship between IRS and broadcasting was previously investigated. For instances, van Leeuwen and Tan [27] proved that minimum spanning tree construction, and there-

*Part of this work has been done while the third author was visiting the Computer Science Department of University Paris-Sud at LRI, supported by the Australian-French ARC/CNRS cooperation #99N92/0523

†Laboratoire de Recherche en Informatique, Bât. 490, *Univ. Paris-Sud*, 91405 Orsay cedex, France. Additional support by the CNRS. http://www.lri.fr/~pierre

‡Laboratoire Bordelais de Recherche en Informatique, *Univ. Bordeaux I*, 33405 Talence cedex, France. Additional support by the Aquitaine Region project #98024002. http://dept-info.labri.u-bordeaux.fr/~gavoille

§Department of Computing, Division of ICS, *Macquarie Univ.*, Sydney, NSW 2109, Australia. http://www.comp.mq.edu.au/~bmans

fore broadcasting, and other related distributed problems such as leader-election can be solved by exchanging $O(n)$ messages in a ring labeled by an IRS, and $O(m+n)$ messages for arbitrary graphs labeled by an IRS. Recall that leader-election without any network knowledge requires $\Theta(n \log n)$ messages for a ring, and $\Theta(m+n \log n)$ messages for an arbitrary graph [15]. More generally, the question of how much a labeling can help in the solution of distributed problems was studied in [8, 9] in the framework of Sense of Direction [7]. It was shown in [5] that the message-complexity of the broadcast problem is $n-1$ in the restricted class of networks supporting *all-shortest-path* IRS (an IRS where all the shortest paths are represented) with additional restrictions on the intervals (strictness and linearity). Finally, de la Torre, Narayanan and Peleg [3] showed that the same result holds in IRS networks satisfying the so-called *ssr-tree* property. (Informally, this property states that, for any node x, the set of paths induced by the IRS originated at x, and ending at all the other nodes, is a tree.) In this paper, we improve these results by showing that a network supporting standard shortest path IRS supports a broadcast protocol of message-complexity $\Theta(n)$.

Our result has many consequences on other problems, such as *leader-election* or *distributed spanning tree*. For instance Korach et al. [23] have shown that the leader-election problem can be solved using $(b(n)+n)(\log n+1)$ messages where $b(n)$ is the message-complexity of broadcasting in an n-node networks. Therefore, n-node networks supporting shortest path IRS allow the leader-election problem to be solved with $O(n \log n)$ message-complexity (instead of $O(m+n \log n)$ for arbitrary networks). In fact, we prove that $O(n)$ messages suffice to solve leader-election for any graphs labeled by a shortest path IRS (see [11] for a proof), improving therefore the $O(m+n)$ previous bound of van Leeuwen and Tan [27].

The paper is organized according to several hypotheses on the IRS. These hypotheses will be relaxed while going further and further in the paper. We can indeed distinguish two types of intervals: an interval $[a,b]=\{a,\ldots,b\}$ with $b \geq a$ is said to be *linear*, whereas an interval $[a,b]$ with $b<a$ refers to the set $\{a,\ldots,n,1,\ldots,b\}$, and is said to be *cyclic*. The class of networks supporting linear IRS (LIRS for short), that is such that all intervals of the IRS are linear, is strictly included in the class of networks supporting an IRS which possibly includes cyclic intervals. We can also distinguish the case $V\setminus\{x\}=\cup_{i=1}^{d} I_i$ for every node x, from the case in which x appears in the interval of one of its incident edges, for at least one node x. In the former case, the IRS is said to be *strict*. Hence we get four types of interval routing schemes: IRS, LIRS, strict IRS, and strict LIRS. The following section presents some preliminary results. Section 3 is dedicated to networks supporting strict LIRS. Sections 4 and 5 then successively relax the strictness requirement and the linearity requirement, in order to present our main result given in Theorem 1.

2. PRELIMINARY RESULTS

In this section, we will present a distributed broadcast protocol for a network supporting an IRS. This protocol is very simple though very efficient since its message-complexity will be shown to be at most $O(n)$. It is called the up/down protocol. The second part of the section presents tools for the analysis of this protocol.

2.1 The up/down broadcast protocol

Let ν be the source of the broadcast, identified by its label in the IRS. The source initiates the broadcast by sending two copies of the message, one destinated to node $\nu-1$, and the other destinated to node $\nu+1$. The latter copy is called the "up" copy whereas the former is called the "down" copy. (Obviously, if $\nu=1$ or $\nu=n$, then the source sends only one copy.) There will be at most two copies of the message circulating in the network. Let us concentrate on the up copy destinated to $\nu+1$. The message will eventually reach $\nu+1$ by the shortest path set by the IRS. It may possibly cross intermediate nodes, but these nodes will just forward the message to its destination stored in the message-header, without taking care of its content. Once the node $\nu+1$ receives the message, it reads its content, modifies the header by replacing $\nu+1$ by $\nu+2$, and forwards it toward the node labeled $\nu+2$. More generally, once a node labeled x, $\nu<x<n$, receives the message destinated to x, it reads its content, modifies the header by replacing x by $x+1$, and forwards it toward the node labeled $x+1$. When the node labeled n receives the message, it reads its content, and removes it from the network. The same strategy is applied for the down copy, by replacing $x+1$ by $x-1$, $1<x<\nu$, until it reaches node labeled 1 which reads its content and removes it from the network. Again, at every given time, a copy of the message is destinated to only one specific node, called the *target*, and when a node different from that target receives the message, it does not take the opportunity to read its content, but just forwards the message to the target along the shortest path set by the IRS that leads to that target.

Clearly, the message-complexity of the up/down protocol is equal to $\sum_{i=1}^{n-1} d(i,i+1)$ where $d(x,y)$ denotes the distance between the node labeled x and the node labeled y in G. Unfortunately, up to the knowledge of the authors, there is no good bounds on $d(i,i+1)$ for networks supporting IRS. Let us just point out that it is known [5] that $d(i,i+1)=1$ for networks supporting *all-shortest-path* strict LIRS that is for networks such that the strict LIRS encodes *all* shortest paths. The class of networks supporting all-shortest-path strict LIRS is very restricted (see [16]), and, as we will see, only weaker results can be proved for (single shortest path) LIRS or IRS.

Assume that the source is labeled 1, and let

$$W=w_1,w_2,\ldots,w_\tau$$

be the sequence of nodes visited by the message from its source w_1 (labeled 1) to its final destination w_τ (labeled n). The same node may appear several times in W. However, a node x can only appear once as a target node, and thus the possible other occurrences of node x correspond to steps of the protocol in which x is traversed by a message destinated to a target node $y \neq x$. Let us complement the sequence W by two virtual nodes, $w_0=n+1$, and $w_{\tau+1}=0$. These two nodes are not target nodes. More precisely, W can be written as:

$$W=Y_0,X_1,Y_1,X_2,Y_2,\ldots,X_{\varrho-1},Y_{\varrho-1},X_\varrho,Y_\varrho \qquad (1)$$

where X_i, $i=1,\ldots,\varrho$, is a maximal sequence of consecutive target nodes, and Y_i, $i = 0, \ldots, \varrho$, is the sequence of non target nodes between the last node of X_i and the first node of X_{i+1}. In particular, $Y_0 = w_0$, and $Y_\varrho = w_{\tau+1}$. The nodes of the Y_i's are called *intermediate* nodes.

Notation. For every i, throughout the paper, we will always make use of the following notation: $k_i = |X_i|$, $\ell_i = |Y_i|$, $x_i =$ the last node of X_i, $y_i =$ the last node of Y_i, and $z_i =$ the first node of Y_i.

2.2 Intermediate nodes sequences

The next lemmas give some properties satisfied by the intermediate nodes in the Y_i's. These properties will be shown to be helpful to compute the message-complexity of the up/down protocol.

LEMMA 1. *Let G be a network supporting a strict LIRS. Let $u_0 = x, u_1, \ldots, u_k = x+1$ (resp. $v_0 = x, v_1, \ldots, v_k = x-1$) be the shortest path from the node labeled x to the node labeled $x+1$ (resp. to node labeled $x-1$) in a strict LIRS. Then $u_1 > u_2 > \cdots > u_{k-1} > x+1$ (resp. $v_1 < v_2 < \cdots < v_{k-1} < x-1$).*

PROOF. Let I_i be the interval of edge (u_i, u_{i+1}) at u_i, $i = 0, \ldots, k-1$. We have $x+1 \in I_i$ by definition of the path $u_0 = x, u_1, \ldots, u_k = x+1$, and $u_{i+1} \in I_i$ because (u_i, u_{i+1}) is the unique shortest from u_i to u_{i+1}. On the other hand, $x \notin I_i$, $i > 0$, by definition of the path $u_0 = x, u_1, \ldots, u_k = x+1$, and $u_i \notin I_i$, $0 \leq i \leq k-1$, because the LIRS is strict. Therefore, since the intervals are all linear, $x+1 < u_{i+1} < u_i$ for $0 < i < k-1$. Similarly, one can show that $x-1 > v_{i+1} > v_i$ for $0 < i < k-1$. $\square$

Lemma 1 states that, in strict LIRS networks, once a target node x has received the message in the up (resp. down) protocol, no node of label smaller (resp. greater) or equal to x will be visited anymore.

COROLLARY 1. *In a network supporting a strict LIRS, $d(i, i+1) \leq \min\{i, n-i\}$.*

The next lemma analyzes the relationship between consecutive sequences of intermediate nodes. Its aim is to answer the following: given an intermediate node $w_i \in W$, where can we find another intermediate node $w_j \in W$, $j > i$, whose label is smaller than the label of w_i? (Lemma 1 answers this question if w_i is not the last node of an intermediate sequence Y_r.)

For any set of nodes $S \subseteq V$, and any node $u \in V$, we denote by $d(u, S)$ the distance between u and S, that is $d(u, S) = \min_{v \in S} d(u, v)$. Lemma 2 below says that if w_i is the last node of the sequence Y_{r-1} (i.e., $d(w_i, X_r) \leq 1$), and if $\ell_r > k_r$ (i.e., $|Y_r| > |X_r|$), then the (k_r+1)-th node of Y_r is smaller than w_i. In other words $w_i > w_j$ for $j = i + k_r + (k_r + 1)$. The look for such a w_j when $\ell_r \leq k_r$ is more complex. It depends mainly on the relative lengths of the Y_s's and X_s's for $s \geq r$.

LEMMA 2. *Let G be a network supporting a strict LIRS, and let u be a node of G. Let $r \in \{1, \ldots, \varrho\}$, and assume $d(u, X_r) \leq \delta$. Assume that there exists $s \geq r$ such that $\sum_{i=r}^{s}(\ell_i - k_i) > \delta - 1$, and $\sum_{i=r}^{s'}(\ell_i - k_i) \leq \delta - 1$ for every s', $r \leq s' < s$. Let $\kappa = k_s + \sum_{i=r}^{s-1}(k_i - \ell_i) + \delta$, and let $Y_s = v_1, \ldots, v_{\ell_s}$. If $u > x$ for every $x \in \cup_{i=r}^{s} X_i$, and if, for every intermediate node $w \in \cup_{i=r}^{s-1} Y_i$, $w \geq u$, then $u > v_\kappa$.*

PROOF. First note that $\sum_{i=r}^{s}(\ell_i - k_i) \geq \delta$ implies that $\kappa \leq \ell_s$. Note also that $\sum_{i=r}^{s-1}(\ell_i - k_i) \leq \delta - 1$ implies that $\kappa \geq k_s + 1 \geq 2$. Let us show by induction on i, that

$$d(u, y_i) \leq (\delta - 1) + \sum_{j=r}^{i}(k_j - \ell_j) \tag{2}$$

for every $i = r, \ldots, s-1$. Let $i = r$, and let $Y_r = u_1, \ldots, u_{\ell_r}$. Note that $y_r = u_{\ell_r}$. Let I_1 be the interval on the edge (x_r, u_1) at x_r, and, for $i > 1$, let I_i be the interval on the edge (u_{i-1}, u_i) at u_{i-1}. For every $i \geq 1$, we have $x_r + 1 \in I_i$, and $x_r \notin I_i$. From Lemma 1, we also have $u_1 > \cdots > u_{\ell_r} > x_r + 1$. By hypothesis, $u > x$ for every $x \in \cup_{i=r}^{s} X_i$, and $u \leq u_{\ell_r}$. Thus, since $u_i \in I_i$, we get that $u \in I_i$ for every $i = 1, \ldots, \ell_r$. Therefore, a shortest path from x_r to u goes through $u_1, \ldots, u_{\ell_r}$. As a consequence $\ell_r + d(y_r, u) \leq k_r + d(u, X_r) - 1$, and thus $d(u, y_r) \leq (\delta - 1) + (k_r - \ell_r)$.

Assume now that $d(u, y_i) \leq (\delta - 1) + \sum_{j=r}^{i}(k_j - \ell_j)$ for some i, $r \leq i < s-1$, and let us show that $d(u, y_{i+1}) \leq (\delta - 1) + \sum_{j=r}^{i+1}(k_j - \ell_j)$. For the same reasons as for the case $i = r$, we get that a shortest path from the last node x_{i+1} of X_{i+1} to u goes through all nodes of Y_{i+1}. We get that

$$\begin{aligned}
\ell_{i+1} + d(u, y_{i+1}) &\leq d(x_{i+1}, u) \\
&\leq k_{i+1} + d(u, y_i) \\
&\leq (\delta - 1) + k_{i+1} + \sum_{j=r}^{i}(k_j - \ell_j),
\end{aligned}$$

and thus $d(u, y_{i+1}) \leq (\delta - 1) + \sum_{j=r}^{i+1}(k_j - \ell_j)$. This completes the proof of Equation 2.

A consequence of Equation 2 is that

$$d(u, x_s) \leq (\delta - 1) + k_s + \sum_{j=r}^{s-1}(k_j - \ell_j) = \kappa - 1.$$

So now, let I_1 be the interval on edge (x_s, v_1) at x_s, and, for $i > 1$, let I_i be the interval on edge (v_{i-1}, v_i) at v_{i-1}. If $u < v_i$ for every $i \leq \kappa - 1$, then $u \in I_i$ for $i = 1, \ldots, \kappa - 1$, and thus, $d(u, x_s) > \kappa - 1$, a contradiction. Thus $u \geq v_i$ for some $i \leq \kappa - 1$, and therefore $u > v_\kappa$ by Lemma 1. This completes the proof of Lemma 2. $\square$

Now, we know enough to start the analysis of the up/down protocol.

3. STRICT LINEAR INTERVAL ROUTING SCHEMES

In this section, we show that the message-complexity of the up/down protocol is at most $3n$ on a network of order n supporting a strict LIRS.

3.1 Partition: a sequence-decomposition algorithm

Assume first that the source is the node 1. We make use of the sequence W as displayed in Equation 1. Since the total number of target nodes is n, we have $\sum_{i=1}^{\varrho}|X_i| = n$. Therefore, the aim of the rest of the proof is to bound the total number $\sum_{i=0}^{\varrho}|Y_i|$ of intermediate nodes. For that purpose, we will partition the intermediate nodes of W in three types of (pairwise disjoint) subsequences. The result of this partition is called the *sequence-decomposition.* More precisely, the sequence-decomposition will be composed of an *active path*, of *dead-end paths*, and of *jumped paths.* The active path is built starting a walk from w_0 up to $w_{\tau+1}$. Along the construction of the sequence-decomposition, some parts of the active path become dead-end paths. At the end of the decomposition, the two extremities of the active path are w_0 and $w_{\tau+1}$, and the total number of nodes kept in the active path will be at most $O(n)$. Also, at the end of the construction, the sum of the lengths of the dead-end paths, and the sum of the lengths of the jumped paths will be both bounded by $O(n)$. Therefore, the total number of intermediate nodes will be $O(n)$, and $|W| = O(n)$. A careful analysis of the constants will actually show that $|W| \leq 3n$.

The sequence-decomposition is performed by visiting all intermediate nodes of the sequence W from w_0 to $w_{\tau+1}$, constructing in this way the active path. One may "jump" over some intermediate nodes if the length of the jump can be bounded by the number of jumped targets. The result of a jump is a jumped path. One may also backtrack along the active path for a bounded number of nodes. The result of a backtrack is a dead-end path. The decomposition requires two parameters: the *mark m*, and the *direction d*. The role of the mark is to keep in mind the progression along the sequence W. In general, the mark indicates the current position, or, more precisely, the index i of the current set Y_i. The mark is an important parameter because of the backtracks. When a backtrack occurs, the mark is set to remember the current maximum position ever reached. The direction indicates whether a backtrack recently occurred. In that case $d = -1$, and $d = +1$ otherwise.

Initially, the active path is reduced to w_0, and there is neither a dead-end path nor a jumped path. The mark m is set to 0, and the direction is set to $+1$. The construction of the sequence-decomposition is precisely described in Algorithm 1. The explanations of the several steps of the construction are given below.

Algorithm 1 One step of the construction of the sequence-decomposition for a strict LIRS

The current active path is $P = p_0, p_1, \dots, p_t$, with $p_0 = w_0$. Let r be such that $p_t \in Y_r$, and let $Y_r = u_1, \dots, u_{\ell_r}$.

Case 1: $p_t = u_i, i < \ell_r$ **and** $d = +1$. Then the active path is updated to $p_0, p_1, \dots, p_t, u_{i+1}$. There is neither a new dead-end path nor a jumped path. The mark and the direction are not modified.

Case 2: $p_t = u_{\ell_r}$ **or** $d = -1$. Then let $\delta = d(p_t, X_{m+1})$, and $s \geq m+1$ be the smallest index such that $\sum_{i=m+1}^{s}(\ell_i - k_i) > \delta - 1$. If s does not exist, then the active path is updated to $p_0, p_1, \dots, p_t, w_{\tau+1}$, there is no new dead-end path, all intermediate nodes between the last node of Y_m and $w_{\tau+1}$ form a new jumped path, and the construction stops. If s exists, then let $\kappa = \delta + k_s + \sum_{i=m+1}^{s-1}(k_i - \ell_i)$. Assume that $Y_s = v_1, \dots, v_{\ell_s}$, two cases are considered:

Case 2.1: $p_t \notin \cup_{i=m+1}^{s} X_i$. Then again two cases may occur.

Case 2.1.1: There is an intermediate node w in $\{v_1, \dots, v_{\kappa-1}\} \cup (\cup_{i=m+1}^{s-1} Y_i)$ such that $w < p_t$. Then pick the first node w of that type. The active path is updated to $p_0, p_1, \dots, p_t, w$, and all intermediate nodes between the last node of Y_m and w form a jumped path. Assume $w \in Y_\sigma$, then the mark m is set to σ.

Case 2.1.2: For every intermediate node w in $\{v_1, \dots, v_{\kappa-1}\} \cup (\cup_{i=m+1}^{s-1} Y_i)$, $w \geq p_t$. Then the active path is updated to $p_0, p_1, \dots, p_t, v_\kappa$, and all intermediate nodes between the last node of Y_m and v_κ forms a jumped path. The mark m is set to s.

In both cases, there is no new dead-end path, and the direction d is set to $+1$.

Case 2.2: $p_t \in \cup_{i=m+1}^{s} X_i$. Then the direction d is set to -1. Let $t' \in \{0, \dots, t-1\}$ be the largest index such that $p_{t'} \notin \cup_{i=m+1}^{s} X_i$. All nodes $p_{t'+1}, \dots, p_t$ forms a dead-end path. Assume $p_{t'+1} \in X_\sigma$, then if $\sigma > m+1$, all intermediate nodes of $\cup_{i=m+1}^{\sigma-1} Y_i$ form a jumped path. The mark m is updated to $\sigma - 1$.

In Case 1, the construction is currently visiting some Y_r. While the last node of this sequence of intermediate nodes is not reached, the active path is updated by adding all the forthcoming nodes of the current sequence. Informally, from Lemma 1, the size of the active path will not increase too much since the labels of the nodes are in a strictly decreasing order. Case 2 happens in particular when the last node of the current sequence Y_r is reached. The definition of s is motivated by Lemma 2. If s does not exist, the construction stops. If s does exist, then we make a jump in the sequence W as explained thereafter. (Note that, like in Lemma 2, $2 \leq \kappa \leq \ell_s$.) In Case 2.1.1, we simply jump at the next intermediate node w whose label is smaller than the label of the current node. Case 2.1.2 can be seen as an extremal case of Case 2.1.1. We know from Lemma 2 that jumping at v_κ is fine since the label of that node is smaller than the label of the current node p_t. The mark is updated to contain the index of the sequence Y_σ of intermediate nodes reached after the jump. Case 2.2 is a particular case because a backtrack occurs. This backtrack is motivated by the fact that one cannot find an intermediate node with a label smaller that the label of p_t. Informally, we backtrack along the active path $p_0, p_1, \dots, p_t$ until we reach a node $p_{t'}$, $t' < t$, for which Lemma 2 can be applied. Note that t' is well defined since $p_0 \notin \cup_{i=1}^{\varrho} X_i = \{1, \dots, n\}$.

LEMMA 3. *The construction of the sequence-decomposition given in Algorithm 1 produces a set of paths in which every intermediate node appears exactly once.*

The proof is based on the following claim.

CLAIM 1. *If* $d = +1$, *then* $p_t \in Y_m$.

PROOF. This claim initially holds. Assume the claim holds before some step i, and consider the several cases of

Algorithm 1. In Case 1, p_{t+1} will still be in Y_m, and d will still be $+1$. In Case 2.1, the direction is set to $+1$ and $p_{t+1} \in Y_m$ by definition of the setting of the mark m in both Cases 2.1.1, and 2.1.2. Case 2.2 sets d to -1. So the claim holds after step i too. $\square$

The proof of the lemma is then as follows:

PROOF. After every step i, let us define m_i as the resulting mark, and

$$q_i = \begin{cases} p_t & \text{if } p_t \in Y_{m_i}; \\ \text{last node of } Y_{m_i} & \text{otherwise.} \end{cases}$$

We claim that, at every step i, all intermediate nodes before q_i appears exactly once in the sequence-decomposition. This claim initially holds. Assume it holds *before* step i. In Case 1, from Claim 1, $q_{i-1} = p_t \in Y_{m_{i-1}}$. The active path is upgraded by adding the next node of the sequence, $Y_{m_i} = Y_{m_{i-1}}$, and thus the claim holds after step i. In Case 2.1, m_i is set such that $p_{t+1} \in Y_{m_i}$, and thus $q_i = p_{t+1}$. Every intermediate node between the last node of $Y_{m_{i-1}}$, that is q_{i-1}, and $q_i = p_{t+1}$ are put in a jumped path, so the claim holds after step i. In Case 2.2, some part of the active path becomes a dead-end path. By the setting of m_i, all the intermediate nodes not yet assigned to any type of paths, that is all intermediate nodes in $\cup_{j=m_{i-1}+1}^{m_i} Y_j$, are put in a jumped path. Thus every intermediate node between q_{i-1} and q_i are put in a jumped path. Therefore, *after* step i, all intermediate nodes before q_i appears exactly once in the sequence-decomposition.

We complete the proof of the lemma by noticing that, after every step, either the active path is increased, or a part of the active path is put in a dead-end path. By the setting of the mark, an intermediate-node put in a dead-end path or in a jumped path will not be considered anymore in the further steps. Therefore the construction of the sequence-decomposition of Algorithm 1 ends after a finite number of steps. $\square$

3.2 Message-complexity of the up/down protocol

From the sequence-decomposition algorithm, let us compute the message-complexity of the up/down protocol.

LEMMA 4. *The number of intermediate nodes in the active path is at most n, excluding w_0 and $w_{\tau+1}$. More precisely, the labels of the nodes of the active path, including w_0 and $w_{\tau+1}$, form a decreasing sequence in $n+1, \dots, 0$.*

PROOF. To prove that lemma, let us go again through the several cases of a step of the decomposition. Let p_t be the last node of the current active path. In Case 1, Lemma 1 insures that $p_{t+1} < p_t$. In Case 2, if s does not exist, then the next node of the active path is $w_{\tau+1} = 0$ which is smaller than every other nodes of the current active path. So, in the remaining of this proof, we assume that s does exist. Since Case 2.2 does not add new node to the active path, we focus on case 2.1. In Case 2.1.1, the new node added to the active path is, by definition, smaller than p_t. In Case 2.1.2, $v_\kappa < p_t$ by application of Lemma 2, which completes the proof of that lemma. $\square$

LEMMA 5. *The number of nodes in the dead-end paths is at most n.*

PROOF. A dead-end path is composed of a sequence of nodes that were formerly in the active path, and through which a backtrack occurred. Every backtrack is driven by a set of target nodes $\cup_{i=m+1}^{\sigma} X_i$ where $p_{t'+1} \in X_\sigma$. Let $X_\sigma = x+1, \dots, x+k$, and $p_{t'+1} = x+i$, $1 \le i \le k$. Similarly, let $p_t \in X_{m'+1} = x'+1, \dots, x'+k'$, $p_t = x'+i'$, $1 \le i' \le k'$.

From Lemma 4, the number of nodes of the dead-end path corresponding to $X_{m+1}, \dots, X_\sigma$ is bounded by $(k'-i') + \sum_{j=m'+2}^{\sigma-1} |X_j| + i$ because the active path, traversed in the reverse direction, produced a sequence of nodes of increasing labels. Now, m is updated to $\sigma - 1$ after the backtrack, and thus $X_{m+1}, \dots, X_{\sigma-1}$ will not be considered anymore for the counting of the number of nodes in the dead-end paths. X_σ may be considered again in another dead-end path since m' can be equal to m. However, only the $k-i$ last nodes of X_σ will be involved. As a consequence, the total number of nodes in dead-end paths is bounded by $\sum_{i=1}^{\varrho} |X_i| = n$. $\square$

CLAIM 2. *Assume that two jumps J' and J successively occurred at x (that is J' occurred, and then later a backtrack led back to x, and J occurred). Let $y' \in Y_{\sigma'} = u_1, \dots, u_{\ell_{\sigma'}}$ and $y \in Y_\sigma$ be the respective extremities of J' and J, and assume $y' = u_i$. Let m and δ (resp. m' and δ') be the setting of the mark and the distance when J (resp. J') occurred. Then $\delta \le (\delta'-1) + \sum_{j=m'+1}^{\sigma'-1}(k_j - \ell_j) + (k_{\sigma'} - i)$.*

PROOF. By definition $\delta = d(x, X_{m+1})$. Also, $y' \in X_{m+1}$ by the setting of m when backtracking through J' as $p_{t'+1} = y'$. Thus $\delta \le d(x, y')$. On the other hand, by the same arguments as for Equation 2, $d(y_{m'+1}, x) + \ell_{m'+1} \le k_{m'+1} - 1 + \delta'$, and thus $d(y_{m'+1}, x) \le (\delta'-1) + (k_{m'+1} - \ell_{m'+1})$. More generally, $d(y_{\sigma'-1}, x) \le (\delta'-1) + \sum_{j=m'+1}^{\sigma'-1}(k_j - \ell_j)$. Since $i + d(y', x) \le k_{\sigma'} + d(y_{\sigma'-1}, x)$, we get

$$d(x, y') \le (\delta'-1) + \sum_{j=m'+1}^{\sigma'-1} (k_j - \ell_j) + (k_{\sigma'} - i)$$

which completes the proof of the claim. $\square$

LEMMA 6. *The number of nodes in the jumped paths is at most n.*

PROOF. As dead-end paths, the jumped paths are characterized by disjoint sets of target nodes of the form $\cup_{i=m+1}^{s} X_i$. Let us first consider the jumped paths created by application of Case 2.1 of Algorithm 1. Assume that a jump J occurred between $x \in Y_r$, and $y \in Y_\sigma$, $r < \sigma \le s$. If $\sigma < s$, then the number of nodes of the jumped path is at most $\sum_{i=m+1}^{\sigma} \ell_i - 1 < (\delta - 1) + \sum_{i=m+1}^{\sigma} |X_i|$. If $\sigma = s$, then assume that $Y_s = v_1, \ldots, v_{\ell_s}$, $y = v_i$ with $i \leq \kappa = \delta + k_s + \sum_{i=m+1}^{s-1}(k_i - \ell_i)$. Thus, the number of nodes of the corresponding jumped path is at most $\sum_{i=m+1}^{s-1} \ell_i + \kappa - 1 \leq (\delta - 1) + \sum_{i=m+1}^{s} |X_i|$. So, in any case, the number of nodes of the jumped path is at most $(\delta - 1) + \sum_{i=m+1}^{\sigma} |X_i|$. If $\delta > 1$, then this setting corresponds to another jump J' occurred at x, say between x and $y' \in Y_{\sigma'} = u_1, \ldots, u_{\ell'}$, $y' = u_i$. Let m' and δ' be respectively the value of the mark and of the distance when J' occurred. From Claim 2, we have

$$\delta \leq (\delta' - 1) + \sum_{j=m'+1}^{\sigma'-1} (k_j - \ell_j) + (k_{\sigma'} - i).$$

On the other hand, the size of the jump at y' is $i - 1 + \sum_{j=m'+1}^{\sigma'-1} \ell_j$. It yields that the number of node for the two jumps is at most $(\delta' - 1) + \sum_{i=m'+1}^{\sigma'} |X_i| + \sum_{i=m+1}^{\sigma} |X_i|$. We can repeat for δ' what we did for δ, until we have considered all jumps that successively occurred at x. It yields that the total number of nodes of the set of jumped paths occurring at the same node x is at most $\sum_{i=r+1}^{\sigma} |X_i|$ where Y_σ contains the other extremity of the last jump occurred at x.

The analysis is the same for the jumped paths created in Case 2.2 by proceeding as if the jump occurred between $p_{t'}$ and the last node of $Y_{\sigma-1}$ (recall that $p_{t'+1} \in X_\sigma$).

We conclude the proof by noticing that due to the setting of the mark m, no two jumps occur above the same set of target nodes, and therefore the total number of nodes is at most $\sum_{i=1}^{\varrho} |X_i| = n$. $\square$

Combining Lemmas 4, 5, and 6 with Lemma 3 allows to conclude that the total number of intermediate nodes is at most $3n$. We can actually improve this upper bound to $2n$. Let x be the extremity of the active path when s cannot be defined, that is when $\sum_{i=m+1}^{s-1}(\ell_i - k_i) \leq \delta - 1$ for every s, $m + 1 \leq s \leq \varrho$. Since $x \in \cup_{i=m+1}^{\varrho} X_i$, all nodes of the active path belong to $\cup_{i=m+1}^{\varrho} X_i$ by application of Lemma 4. By application of the same lemma, the total number of nodes in the active path is at most $\sum_{i=m+1}^{\varrho} |X_i|$. Since the sets $X_{m+1}, \ldots, X_\varrho$ were not used to bound the total number of dead-end paths, we can conclude that the sum of the number of nodes in the active path plus the number of nodes in the dead-end paths is at most n. From all what precede, the total number of nodes in the sequence-decomposition, and therefore the total number of intermediate nodes, is at most $2n$. Therefore, the total number of nodes in the sequence W is at most $3n$, and hence the message-complexity of the up/down protocol is at most $3n$.

If the source node is not the node labeled 1, then let $\nu > 1$ be the label of the source. From Lemma 1, the message complexity of the copy going upward is at most $3(n - \nu)$ whereas the message-complexity of the copy going downward is at most 3ν.

To summarize, we get:

PROPERTY 1. *The message-complexity of the up/down broadcast protocol is at most $3n$ in a network of order n supporting any strict LIRS.*

Note that $K_{2,n-2}$, the complete bipartite graph with one partition of size 2, and another partition of size $n - 2$, supports a strict LIRS for which the up/down protocol uses $2n - 4$ messages.

4. LINEAR INTERVAL ROUTING SCHEMES

The next lemma is a variant of Lemma 1 adapted to (non strict) LIRS networks.

LEMMA 7. *Let G be a network supporting an LIRS. Let $u_0 = x, u_1, \ldots, u_k = x + 1$ (resp. $v_0 = x, v_1, \ldots, v_k = x - 1$) be the shortest path from the node labeled x to the node labeled $x + 1$ (resp. to node labeled $x - 1$) in an LIRS. Then, for every i, $k > i > 1$, we have $u_i > x + 1$, and $u_i < u_j$ for every j, $1 < j \leq i - 2$ (resp., for every i, $1 < i < k$, $v_i < x - 1$, and $v_i > v_j$ for every j, $1 < j \leq i - 2$).*

PROOF. Let I_i be the interval of edge (u_i, u_{i+1}) at u_i, $i = 0, \ldots, k-1$. By definition, we have $x + 1 \in I_i$, and $u_{i+1} \in I_i$. For $i \geq 1$, we also have $x \notin I_i$ (otherwise there would exist a shorter path from x to $x + 1$). Thus $u_i > x + 1$ for $i > 1$. More generally, if $i \geq 1$, then $u_j \notin I_i$ for every $j \leq i - 1$. Therefore, since all the intervals are linear, $u_{i+1} < u_j$ for every j, $1 < j \leq i - 1$. The result for the v_i's is obtained in a similar way. $\square$

COROLLARY 2. *In a network supporting an LIRS, $d(i, i + 1) \leq 1 + \min\{i, n - i\}$.*

The difference between Lemma 1 and Lemma 7 motivates the following adaptation of Lemma 2 for LIRS networks that are non necessarily strict.

LEMMA 8. *Let G be a network supporting an LIRS, and let u be a node of G. Let $r \in \{1, \ldots, \varrho\}$, and assume that $d(u, X_r) \leq \delta$. Assume that there exists $s \geq r$ such that $\sum_{i=r}^{s}(\ell_i - k_i) \geq 2(s - r + 1) + \delta$, and $\sum_{i=r}^{s'}(\ell_i - k_i) < 2(s' - r + 1) + \delta$ for every s', $r \leq s' < s$. Let $\kappa = k_s + \sum_{i=r}^{s-1}(k_i - \ell_i) + 2(s - r) + \delta$, and let $Y_s = v_1, \ldots, v_{\ell_s}$. Assume that $u > x$ for every $x \in \cup_{i=r+1}^{s} X_i$, and that, for every intermediate node $w \in \cup_{i=r}^{s-1}(Y_i \setminus \{z_i\})$, $w > u$. Then $\min\{v_{\kappa+1}, v_{\kappa+2}\} \leq u$.*

PROOF. Note first that, $\sum_{i=r}^{s-1}(k_i - \ell_i) > 2(r - s) - \delta$ implies $\kappa > k_s \geq 1$, and $\sum_{i=r}^{s}(k_i - \ell_i) \leq 2(r - s - 1) - \delta$ implies $\kappa \leq \ell_s - 2$. We proceed is a way similar to the proof of Lemma 2. We show that, for every $i \in \{r, \ldots, s - 1\}$,

$$d(u, y_i) \leq \sum_{j=r}^{i}(k_j - \ell_j) + 2(i - r) + (\delta + 1). \qquad (3)$$

Let $i = r$. The shortest path from z_r to u set by the IRS goes through all nodes of Y_r. Thus $\ell_r - 1 + d(y_r, u) \leq k_r + d(u, X_r)$, and hence $d(y_r, u) \leq (k_r - \ell_r) + (\delta + 1)$. Therefore, Equation 3 holds for $i = r$. Assume Equation 3 holds for i, and let us show that it holds for $i + 1$. Again, the shortest path from z_{i+1} to u set by the IRS goes through all nodes of Y_{i+1}, and thus $\ell_{i+1}-1+d(y_{i+1},u) \leq k_{i+1}+1+d(u,y_i)$, and hence $d(y_{i+1},u) \leq \sum_{j=r}^{i+1}(k_j-\ell_j)+(\delta+1)+2(i+1-r)$, and Equation 3 holds for $i+1$.

A consequence of Equation 3 is that

$$d(z_s,u) \leq k_s+1+\sum_{i=r}^{s-1}(k_i-\ell_i)+(\delta+1)+2(s-r-1)=\kappa.$$

Thus there is a node in $\{v_2,\ldots,v_{\kappa+1}\}$ such that $v_i \leq u$. Indeed, otherwise, the path set by the IRS from z_s to u would go through $v_2,\ldots,v_{\kappa+1}$, and thus would not be a shortest path. Let i be the smallest index in $\{2,\ldots,\kappa+1\}$ such that $v_i \leq u$. If $i<\kappa+1$, then $v_{\kappa+2}<u$ from Lemma 7. If $i=\kappa+1$, then $v_{\kappa+1} \leq u$ (actually, $v_{\kappa+1}=u$), and the result holds. $\square$

In order to analyze the up/down protocol in networks supporting LIRS, we partition the sequence W in slightly different manner than in the previous section. The formal decomposition is given in Algorithm 2. The decomposition for LIRS actually looks very similar to the decomposition for strict LIRS. A fourth type of path is introduced: the *auxiliary path*. This path is motivated by the fact the active path will include one node every two, according to Lemma 7. Every node in between two nodes of the active path is dropped in the auxiliary path. Let us just mention some differences appearing at every step of the sequence-decomposition. Case 1 is roughly the same as Case 1 in Algorithm 1, that is the decomposition uses the current intermediate sequence to construct the active path. The only modification is that one node every two is dropped in the auxiliary path, and that one may stop at u_{ℓ_r-1} or u_{ℓ_r}. Therefore, Case 2 considers also the case $p_t=u_{\ell_r-1}$, which, if true, implies that u_{ℓ_r} is put in the auxiliary path. Case 2 also differs from Case 2 in Algorithm 1 by the definition of both s and κ. The new settings are adapted from the statement of Lemma 8. Otherwise, the general structure of the decomposition for LIRS is the same as Algorithm 1.

We will show the following results. Firstly, the labels of the nodes of the active path form a non increasing sequence $p_0,\ldots,p_t$ from $p_0=n+1$ to $p_t=0$ such that the number of times that $p_i=p_{i-1}$ is at most ϱ. Thus the number of intermediate nodes in the active path is at most $n+\varrho$. This result can be refined by showing that the number of nodes in dead-end paths or in the active path is at most $n+\varrho$. A direct consequence is that the number of nodes in the auxiliary path is at most $n+\varrho$ since the number of nodes in the auxiliary path does not exceed the number of nodes in the active path or in the dead-end path. The contribution of the jumped paths is a bit larger since the number of nodes in the jumped paths is at most $n+3\varrho$. Therefore, $\sum_{i=1}^{\varrho-1}|Y_i| \leq 3n+5\varrho \leq 8n$, and thus $|W| \leq 9n$.

LEMMA 9. *The number of intermediate nodes in the active path is at most $n+\varrho$, excluding w_0 and $w_{\tau+1}$. More precisely, the labels of the nodes of the active path form a non increasing sequence $p_0,\ldots,p_t$ from $p_0=n+1$ to $p_t=0$ such that the number of times that $p_i=p_{i-1}$ is at most ϱ.*

PROOF. From lemma 7, Case 1 insures that $p_{t+1}=u_{i+2}<u_i=p_t$. In Case 2, if s does not exist, then $p_{t+1}=w_{\tau+1}=0<p_t$. So assume that s exists. There is no setting of the active path in Case 2.2. In case 2.1.1, a jump occurs and $p_{t+1} \leq p_t$ by definition. In case 2.1.2, a jump occurs too and $p_{t+1} \leq p_t$ from Lemma 8. The number of jumps is at most ϱ. $\square$

LEMMA 10. *The number of nodes in dead-end paths or in the active path is at most $n+\varrho$.*

PROOF. By similar arguments as in the proof of Lemma 5, and using the same notation, the number of nodes of a dead-end path corresponding to the sets $X_{m+1},\ldots,X_\sigma$ is at most $(k'-i')+\sum_{j=m'+1}^{\sigma-1}|X_j|+i$ plus the number of jumps occurring in the portion $p_{t'+1},\ldots,p_t$ of W. It yields a total number of nodes in dead-end paths of at most $\sum_{i=1}^{\varrho}|X_i|+\varrho=n+\varrho$.

Now, as we observed in the strict LIRS case, if $x \in \cup_{i=m+1}^{\varrho}X_i$ is the last node of the active path different from $w_{\tau+1}$, then the total number of nodes in the active path is at most $(k'-i')+\sum_{i=m+2}^{\varrho}|X_i|+j$ where j is the number of jumps of the final active path. The sets $X_{m+2},\ldots,X_\varrho$ were not used to enumerate the nodes in the dead-end paths, and the total number of jumps cannot exceed ϱ. Therefore, the total number of nodes in the active path or in dead-end paths is at most $n+\varrho$. $\square$

LEMMA 11. *The number of nodes in the auxiliary path is at most $n+\varrho$.*

PROOF. This is a direct consequence of Lemma 10 since the number of nodes in the auxiliary path does not exceed the number of nodes in the active path or in the dead-end path. Indeed, at most one node is dropped in the auxiliary path for every node entering the active path. Some nodes formerly in the active path become member of a dead-end path. $\square$

LEMMA 12. *The number of nodes in the jumped paths is at most $n+3\varrho$.*

PROOF. Let us first consider a jump created by application of Case 2.1. Let J be a jump corresponding to the sets $X_{m+1},\ldots,X_s$. Assume that J occurred between $x \in Y_r$ and $y \in Y_\sigma$, $r<\sigma \leq s$. If $\sigma<s$, then the jump was over at most $\sum_{i=m+1}^{\sigma}\ell_i-1$ intermediate nodes, with $\sum_{i=m+1}^{\sigma}\ell_i<\sum_{i=m+1}^{\sigma}k_i+2(\sigma-m)+\delta$. If $\sigma=s$, let $Y_s=v_1,\ldots,v_{\ell_s}$, $y=v_i$. Then $i \leq \kappa+2=\delta+k_s+\sum_{i=m+1}^{s-1}(k_i-\ell_i)+2(s-m)$. Therefore, in any case, the number of jumped intermediate nodes is at most $(\delta-1)+\sum_{i=m+1}^{\sigma}|X_i|+2(\sigma-m)$. If $\delta>2$, then this setting corresponds to another jump J' occurred at x, say between x and $y' \in Y_{\sigma'}=u_1,\ldots,u_{\ell'}$, $y'=u_i$. Let m' and δ' be the value of the mark, and the value of the distance, respectively, when this jump occurred. We have

$$\delta \leq (\delta'-1)+\sum_{j=m'+1}^{\sigma'-1}(k_j-\ell_j)+2(\sigma'-m')+(k_{\sigma'}-i).$$

Indeed, $\delta = d(x, X_{m+1})$. Moreover, $y' \in X_{m+1}$ by the setting of m when backtracking through J' as $y' = p_{t'+1}$. Therefore, $\delta \leq d(x, y')$, and, by the same arguments as in the proof of Claim 2,

$$d(x, y') \leq (\delta' - 1) + \sum_{j=m'+1}^{\sigma'-1} (k_j - \ell_j) + 2(\sigma' - m') + (k_{\sigma'} - i).$$

The size of the jump J' is $i - 1 + \sum_{j=m'+1}^{\sigma'-1} (k_j - \ell_j)$, and thus the size of the two jumps is at most

$$(\delta' - 1) + \sum_{i=m+1}^{\sigma} |X_i| + 2(\sigma - m) + \sum_{i=m'+1}^{\sigma'} |X_i| + 2(\sigma' - m').$$

If $\delta' > 2$, then one can apply on δ' the same arguments. We get that the number of nodes of the set of jumped paths occurring at the same node $x \in Y_r$ is at most $1 + \sum_{i=r+1}^{\sigma} |X_i| + 2(\sigma - r)$ where Y_σ contains the extremity of the last jump occurred at x.

The analysis is the same for the jumped paths created in Case 2.2 by proceeding as if the jump occurred between $p_{t'}$ and the last node of $Y_{\sigma-1}$ (recall that $p_{t'+1} \in X_\sigma$).

The worst case is reached when every jump is over a single X_i, whose cost in term of nodes is $1 + |X_i| + 2 = |X_i| + 3$, for a total number of nodes at most $\sum_{i=1}^{\varrho} |X_i| + 3\varrho$. □

Therefore, we get:

PROPERTY 2. *The message-complexity of the up/down broadcast protocol is at most $9n$ in a network of order n supporting any LIRS.*

5. INTERVAL ROUTING SCHEMES

As opposed to linear IRS, the labels of the nodes of an IRS play all the same role. Therefore, we slightly modify the up/down protocol to adapt it to IRS networks. If the source is labeled ν, $1 < \nu < n$, then there is only one copy of the message, going upward from ν to n, then from n to 1, and finally from 1 to $\nu - 1$, rather than two copies, one going downward from ν to 1, and the other going upward from ν to n. This protocol is denoted by up/1/up. So, as far as the up/1/up protocol is concerned, one can assume, w.l.g., that the source is labeled 1.

Again, in order to analyze the up/1/up protocol, we make use of the sequence W defined in Equation 1. For every $x \in \{0, \ldots, n-1\}$, we define

$$L_x : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$$

by

$$L_x(u) = 1 + [(n + u - x - 1) \bmod n].$$

We have $L_x(x) = n$ and $L_x(x+1) = 1$. By using the same techniques as Lemmas 1 and 7, the reader can check that:

LEMMA 13. *Let G be a network supporting an IRS. Let $u_0 = x, u_1, \ldots, u_k = x + 1$ (resp. $v_0 = x, v_1, \ldots, v_k = x - 1$) be the shortest path from the node labeled x to the node labeled $x + 1$ (resp. to node labeled $x - 1$) in this IRS.*

- *If the IRS is strict, then $L_x(x) = n > L_x(u_1) > L_x(u_2) > \cdots > L_x(u_{k-1}) > L_x(x+1) = 1$ (resp. $L_{x-1}(x) = 1 < L_{x-1}(v_1) < \cdots < L_{x-1}(v_{k-1}) < L_{x-1}(x-1) = n$).*
- *Otherwise, for every i, $k > i > 1$, we have $L_x(u_i) > 1$, and $L_x(u_i) < L_x(u_j)$ for every j, $1 < j \leq i - 2$ (resp., for every i, $1 < i < k$, $L_{x-1}(v_i) < n$, and $L_{x-1}(v_i) > L_{x-1}(v_j)$ for every j, $1 < j \leq i - 2$).*

By using the same techniques as Lemmas 2 and 8, the reader can also check that:

LEMMA 14. *Let G be a network supporting an IRS, and let u be any node of G. Let $r \in \{1, \ldots, \varrho\}$, and assume that $d(u, X_r) \leq \delta$.*

- *If the IRS is strict, then assume that there exists $s \geq r$ such that $\sum_{i=r}^{s} (\ell_i - k_i) > \delta - 1$, and $\sum_{i=r}^{s'} (\ell_i - k_i) \leq \delta - 1$ for every s', $r \leq s' < s$. Let $\kappa = k_s + \sum_{i=r}^{s-1} (k_i - \ell_i) + \delta$, and let $Y_s = v_1, \ldots, v_{\ell_s}$. If, for every $i \in \{r, \ldots, s\}$, and every $x \in X_i$, $L_{x_i}(u) > L_{x_i}(x)$, and if, for every $i \in \{r, \ldots, s-1\}$ and every intermediate node $w \in Y_i$, $L_{x_i}(w) \geq L_{x_i}(u)$, then $L_{x_s}(v_\kappa) < L_{x_s}(u)$.*
- *Otherwise, assume that there exists $s \geq r$ such that $\sum_{i=r}^{s} (\ell_i - k_i) \geq 2(s - r + 1) + \delta$, and $\sum_{i=r}^{s'} (\ell_i - k_i) < 2(s' - r + 1) + \delta$ for every s', $r \leq s' < s$. Let $\kappa = k_s + \sum_{i=r}^{s-1} (k_i - \ell_i) + 2(s - r) + \delta$, and let $Y_s = v_1, \ldots, v_{\ell_s}$. If, for every $i \in \{r, \ldots, s\}$, and every $x \in X_i$, $L_{x_i}(u) > L_{x_i}(x)$, and if, for every $i \in \{r, \ldots, s-1\}$ and every intermediate node $w \in Y_i \setminus \{z_i\}$, $L_{x_i}(w) > L_{x_i}(u)$, then $\min\{L_{x_s}(v_{\kappa+1}), L_{x_s}(v_{\kappa+2})\} \leq L_{x_s}(u)$.*

Therefore, the sequence-decomposition of Algorithm 1, and its adaptation to LIRS networks described in Section 4 can be applied by introducing the relabeling L_x's every time that a comparison is performed between two labels. The resulting algorithms are given in Algorithms 3 and 4.

The previous lemmas of this section show that there is no major difference between the sequence-decomposition for LIRS networks and the sequence-decomposition for IRS networks. In fact, the up/1/up protocol for IRS and strict IRS networks satisfy the same properties as the up/down protocol for LIRS and strict LIRS networks, respectively. The key of the proof is the following result:

LEMMA 15. *Assume $p_i \in Y_r$, and $p_{i+1} \in Y_\sigma$, $r < \sigma$. Then $L_{x_\sigma}(p_{i+1}) < L_{x_r}(p_i)$ in the sequence-decomposition for strict IRS networks, and $L_{x_\sigma}(p_{i+1}) \leq L_{x_r}(p_i)$, where the maximum number of equality is at most ϱ, in the sequence-decomposition for IRS networks.*

PROOF. According to the statement of the lemma, we are considering Case 2.1 of the sequence decomposition. Hence $p_i \notin \cup_{j=m+1}^{\sigma} X_j$. More precisely $p_i \notin \cup_{j=r+1}^{\sigma} X_j$ from the possible setting of the mark after several backtracks ending at p_i. Let us first consider the strict IRS decomposition.

We have $p_{i+1} \in Y_\sigma$, and $L_{x_\sigma}(p_{i+1}) < L_{x_\sigma}(p_i)$. Let us assume that $L_{x_r}(x_\sigma) > L_{x_r}(p_i)$. Then, since from Lemma 13, $L_{x_r}(p_i) > L_{x_r}(x_r+1)$, we get that p_i is equal to a target node in $\cup_{i=r+1}^{\sigma} X_i$, a contradiction with the hypothesis of Case 2.1. Therefore $L_{x_r}(x_\sigma) < L_{x_r}(p_i)$, and hence $L_{x_\sigma}(p_i) < L_{x_r}(p_i)$. Therefore, $L_{x_\sigma}(p_{i+1}) < L_{x_r}(p_i)$. Similar arguments allow to show that $L_{x_\sigma}(p_{i+1}) \leq L_{x_r}(p_i)$ in the sequence-decomposition for IRS networks. $\square$

Therefore, if $f(i)$ denotes the index such that $p_i \in X_{f(i)}$, then the active path $p_0, p_1, \ldots, p_t, p_{t+1}$ resulting from the sequence-decomposition for strict IRS networks satisfies $p_0 = n+1$, $p_{t+1} = 0$, and $n+1 > L_{x_{f(i)}}(p_i) > L_{x_{f(j)}}(p_j) > 0$ for any pair (i,j), $1 \leq i < j \leq t$. In IRS networks, there might be up to ϱ equalities in the sequence. As a consequence:

THEOREM 1. *The message-complexity of the up/1/up broadcast protocol is respectively at most $3n$ in a network of order n supporting any strict IRS, and at most $9n$ in a network of order n supporting any IRS. As a consequence, networks supporting a shortest path Interval Routing Scheme allows broadcasting with $O(n)$ message-complexity.*

COROLLARY 3. *In a network supporting any strict IRS, the average distance between two nodes labeled by two consecutive integers is at most $3+O(1/n)$. In a network supporting any IRS, the average distance between two nodes labeled by two consecutive integers is at most $9+O(1/n)$.*

6. REFERENCES

[1] B. Awerbuch, I. Cidon, S. Kutten, Y. Mansour, and D. Peleg. Optimal broadcast with partial knowledge. *SIAM Journal on Computing*, 28(2):511–524, 1998.

[2] B. Awerbuch, O. Goldreich, D. Peleg, and R. Vainish. A tradeoff between information and communication in broadcast protocols. *Journal of the ACM*, 37(2):238–256, 1990.

[3] P. de la Torre, L. Naranayan, and D. Peleg. Thy neighbor's interval is greener: A proposal for exploiting interval routing schemes. In *5th International Colloquium on Structural Information and Communication Complexity (SIROCCO '98)*, pages 214-228. Carleton Scientific, 1998.

[4] K. Diks, E. Kranakis, D. Krizanc, and A. Pelc. The impact of knowledge on broadcasting time in radio networks. In 7^{th} *Annual European Symposium on Algorithms (ESA)*, volume 1643 of Lectures Notes in Computer Science, pages 41–52. Springer-Verlag, 1999.

[5] T. Eilam, D. Peleg, R. Tan, and S. Zaks. Broadcast in linear messages in IRS representing all shortest paths. Manuscript, 1997.

[6] M. Flammini, J. van Leeuwen, and A. Marchetti-Spaccamela. The complexity of interval routing on random graphs. In 20^{th} *International Symposium on Mathematical Foundations of Computer Sciences (MFCS)*, volume 969 of Lecture Notes in Computer Science, pages 37–49. Springer-Verlag, 1995.

[7] P. Flocchini, B. Mans, and N. Santoro. Sense of direction: Definitions, properties and classes. *Networks*, 32:165–180, 1998.

[8] P. Flocchini, B. Mans, and N. Santoro. On the impact of Sense of Direction on Message Complexity. *Information Processing Letters*, 63(1):23–31, 1997.

[9] P. Flocchini, B. Mans, and N. Santoro. Sense of direction in distributed computing. In *12th International Symposium on Distributed Computing (DISC)*, volume 1499 of Lecture Notes in Computer Science, pages 1–15. Springer-Verlag, 1998.

[10] P. Fraigniaud and C. Gavoille. Interval routing schemes. *Algorithmica*, 21:155–182, 1998.

[11] P. Fraigniaud, C. Gavoille, and B. Mans. Interval Routing Schemes allow Broadcasting with Linear Message-Complexity. Technical Report LRI-1241, Laboratoire de Recherche en Informatique, Univ. Paris-Sud, 91405 Orsay, France. Jan. 2000.

[12] G. Frederickson. Searching among intervals and compact routing tables. In 20^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 700 of Lecture Notes in Computer Science, pages 28–39. Springer-Verlag, 1993.

[13] G. Frederickson and R. Janardan. Separator-based strategies for efficient message routing. In 27^{th} *Symposium on Foundations of Computer Science (FOCS)*, pages 428–437, 1986.

[14] G. Frederickson and R. Janardan. Designing networks with compact routing tables. *Algorithmica*, 3:171–190, 1988.

[15] G. Gallager, P. Humblet, and P. Spira. A distributed algorithm for minimal spanning tree. *ACM Trans. on Programming Languages Systems*, 5(1):66–77, 1983.

[16] C. Gavoille. A survey on interval routing. *Theoretical Computer Science*. To appear in Vol.245.

[17] C. Gavoille and E. Guévremont. Worst case bounds for shortest path interval routing. *Journal of Algorithms*, 27:1–25, 1998.

[18] C. Gavoille and N. Hanusse. Compact routing tables for graphs of bounded genus. In 26^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1644 of Lecture Notes in Computer Science, pages 351–360. Springer-Verlag, 1999.

[19] C. Gavoille and D. Peleg. The compactness of interval routing. *SIAM Journal of Discrete Mathematics*, 12(4):459–473, 1999.

[20] C. Gavoille and S. Pérennès. Memory requirement for routing in distributed networks. In 15^{th} *Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 125–133, 1996.

[21] E. Kranakis and D. Krizanc. Lower bounds for compact routing. In 13^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1046 of Lecture Notes in Computer Science, pages 529–540. Springer-Verlag, 1996.

[22] E. Kranakis, D. Krizanc, and S. Ravi. On multi-label linear interval routing schemes. In 19^{th} *International Workshop on Graph Theoretic Concepts in Computer Science (WG)*, volume 790 of Lecture Notes in Computer Science, pages 338–349. Springer-Verlag, 1993.

[23] E. Korach, S. Kutten and S. Moran. A modular technique for the design of efficient distributed leader finding algorithms, *ACM Trans. on Programming Languages Systems*, 12(1):84–101, 1990.

[24] D. Peleg and E. Upfal. Efficient message passing using succinct routing tables. In 20^{th} *Annual ACM Symposium on Theory of Computing (STOC)*, pages 43–52, 1988.

[25] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *Journal of the ACM*, 36(3):510–530, 1989.

[26] N. Santoro and R. Khatib. Labelling and implicit routing in networks. *The Computer Journal*, 28(1):5–8, 1985.

[27] J. van Leeuwen and R. Tan. Interval routing. *The Computer Journal*, 30(4):298–307, 1987.

Algorithm 2 One step of the construction of the sequence-decomposition for an LIRS

The current active path is $P = p_0, p_1, \dots, p_t$, with $p_0 = w_0$. Let r be such that $p_t \in Y_r$, and let $Y_r = u_1, \dots, u_{\ell_r}$.

Case 1: $p_t = u_i, i < \ell_r - 1$ **and** $d = +1$. Then the active path is updated to $p_0, p_1, \dots, p_t, u_{i+2}$, and node u_{i+1} is put in the auxiliary path. There is neither a new dead-end path nor a jumped path. The mark and the direction are not modified.

Case 2: $p_t = u_{\ell_r}$ **or** $p_t = u_{\ell_r - 1}$ **or** $d = -1$. Then let $\delta = d(p_t, X_{m+1})$, and $s \geq m + 1$ be the smallest index such that $\sum_{i=m+1}^{s}(\ell_i - k_i) \geq 2(s - m) + \delta$. If s does not exist, then the active path is updated to $p_0, p_1, \dots, p_t, w_{\tau+1}$, there is no new dead-end path, all intermediate nodes between the last node of Y_m and $w_{\tau+1}$ form a new jumped path, and the construction stops. If s exists, then let $\kappa = \delta + k_s + \sum_{i=m+1}^{s-1}(k_i - \ell_i) + 2(s - m - 1)$. If $p_t = y_{\ell_r - 1}$ and $d = +1$, then u_{ℓ_r} is put in the auxiliary path. Assume that $Y_s = v_1, \dots, v_{\ell_s}$, two cases are considered:

- **Case 2.1:** $p_t \notin \cup_{i=m+1}^{s} X_i$. Then again two cases may occur.
  - **Case 2.1.1:** There is an intermediate node w in $\{v_1, \dots, v_\kappa\} \cup (\cup_{i=m+1}^{s-1}(Y_i \setminus \{z_i\}))$ such that $w \leq p_t$. Then pick the first node w of that type. The active path is updated to $p_0, p_1, \dots, p_t, w$, and all intermediate nodes between the last node of Y_m and w form a jumped path. Assume $w \in Y_\sigma$, the mark m is updated to σ.
  - **Case 2.1.2:** For every intermediate node w in $\{v_1, \dots, v_\kappa\} \cup (\cup_{i=m+1}^{s-1}(Y_i \setminus \{z_i\}))$, $w > \min\{p_t, p_{t-1}\}$. Then the active path is updated by setting $p_{t+1} = \min\{v_{\kappa+1}, v_{\kappa+2}\}$. All intermediate nodes between the last node of Y_m and p_{t+1} forms a jumped path. The mark m is set to s.

  In both cases, there is no new dead-end path, and the direction d is set to $+1$.
- **Case 2.2:** $p_t \in \cup_{i=m+1}^{s} X_i$. The direction d is set to -1. Let $t' \in \{0, \dots, t-1\}$ be the largest index such that $p_{t'} \notin \cup_{i=m+1}^{s} X_i$. All nodes $p_{t'+1}, \dots, p_t$ forms a dead-end path. Assume that $p_{t'+1} \in X_\sigma$, then if $\sigma > m + 1$, all intermediate nodes of $\cup_{i=m+1}^{\sigma-1} Y_i$ form a jumped path. The mark m is updated to $\sigma - 1$.

Algorithm 3 One step of the construction of the sequence-decomposition for a strict IRS

The current active path is $P = p_0, p_1, \dots, p_t$, with $p_0 = w_0$. Let r be such that $p_t \in Y_r$, and let $Y_r = u_1, \dots, u_{\ell_r}$.

Case 1: $p_t = u_i, i < \ell_r$ **and** $d = +1$. Then the active path is updated to $p_0, p_1, \dots, p_t, u_{i+1}$. There is neither a new dead-end path nor a jumped path. The mark and the direction are not modified.

Case 2: $p_t = u_{\ell_r}$ **or** $d = -1$. Then let $\delta = d(p_t, X_{m+1})$, and $s \geq m + 1$ be the smallest index such that $\sum_{i=m+1}^{s}(\ell_i - k_i) > \delta - 1$. If s does not exist, then the active path is updated to $p_0, p_1, \dots, p_t, w_{\tau+1}$, there is no new dead-end path, all intermediate nodes between the last node of Y_m and $w_{\tau+1}$ form a new jumped path, and the construction stops. If s exists, then let $\kappa = \delta + k_s + \sum_{i=m+1}^{s-1}(k_i - \ell_i)$. Assume that $Y_s = v_1, \dots, v_{\ell_s}$, two cases are considered:

- **Case 2.1:** $p_t \notin \cup_{i=m+1}^{s} X_i$. Then again two cases may occur:
  - **Case 2.1.1:** There is an intermediate node w in $\{v_1, \dots, v_{\kappa-1}\} \cup (\cup_{i=m+1}^{s-1} Y_i)$, such that $w \in Y_\sigma$ and $L_{x_\sigma}(w) < L_{x_\sigma}(p_t)$. Then pick the first node w of that type. The active path is updated to $p_0, p_1, \dots, p_t, w$, and all intermediate nodes between the last node of Y_m and w form a jumped path. The mark m is updated to σ.
  - **Case 2.1.2:** For every intermediate node w in $\{v_1, \dots, v_{\kappa-1}\} \cup (\cup_{i=m+1}^{s-1} Y_i)$,
    $$w \in Y_\sigma \Rightarrow L_{x_\sigma}(w) \geq L_{x_\sigma}(p_t).$$
    Then the active path is updated to $p_0, p_1, \dots, p_t, v_\kappa$, and all intermediate nodes between the last node of Y_m and v_κ forms a jumped path. The mark m is set to s.

  In any case, there is no new dead-end path, and the direction d is set to $+1$.
- **Case 2.2:** $p_t \in \cup_{i=m+1}^{s} X_i$. The direction d is set to -1. Let $t' \in \{0, \dots, t-1\}$ be the largest index such that $p_{t'} \notin \cup_{i=m+1}^{s} X_i$. All nodes $p_{t'+1}, \dots, p_t$ forms a dead-end path. Assume $p_{t'+1} \in X_\sigma$, then if $\sigma > m + 1$, all intermediate nodes of $\cup_{i=m+1}^{\sigma-1} Y_i$ form a jumped path. The mark m is updated to $\sigma - 1$.

Algorithm 4 One step of the construction of the sequence-decomposition for an IRS

The current active path is $P = p_0, p_1, \dots, p_t$, with $p_0 = w_0$. Let r be such that $p_t \in Y_r$, and let $Y_r = u_1, \dots, u_{\ell_r}$.

Case 1: $p_t = u_i, i < \ell_{r-1}$ **and** $d = +1$. Then the active path is updated to $p_0, p_1, \dots, p_t, u_{i+2}$, and node y_{i+1} is put in the auxiliary path. There is neither a new dead-end path nor a jumped path. The mark and the direction are not modified.

Case 2: $p_t = u_{\ell_r}$ **or** $p_t = u_{\ell_r - 1}$ **or** $d = -1$. Then let $\delta = d(p_t, X_{m+1})$, and $s \geq m + 1$ be the smallest index such that $\sum_{i=m+1}^{s}(\ell_i - k_i) \geq 2(s - m) + \delta$. If s does not exist, then the active path is updated to $p_0, p_1, \dots, p_t, w_{\tau+1}$, there is no new dead-end path, all intermediate nodes between the last node of Y_m and $w_{\tau+1}$ form a new jumped path, and the construction stops. If s exists, then let $\kappa = \delta + k_s + \sum_{i=m+1}^{s-1}(k_i - \ell_i) + 2(s - m - 1)$. If $p_t = u_{\ell_r - 1}$ and $d = +1$, then y_{ℓ_r} is put in the auxiliary path. Assume that $Y_s = v_1, \dots, v_{\ell_s}$, two cases are considered:

- **Case 2.1:** $p_t \notin \cup_{i=m+1}^{s} X_i$. Then again two cases may occur:
  - **Case 2.1.1:** There is an intermediate node w in $\{v_1, \dots, v_\kappa\} \cup (\cup_{i=m+1}^{s-1}(Y_i \setminus \{z_i\}))$ such that $w \in Y_\sigma$ and $L_{x_\sigma}(w) \leq L_{x_\sigma}(p_t)$. Then pick the first node w of that type. The active path is updated to $p_0, p_1, \dots, p_t, w$, and all intermediate nodes between the last node of Y_m and w form a jumped path. The mark m is updated to σ.
  - **Case 2.1.2:** For every intermediate node w in $\{v_1, \dots, v_\kappa\} \cup (\cup_{i=m+1}^{s-1}(Y_i \setminus \{z_i\}))$,
    $$w \in Y_\sigma \Rightarrow L_{x_\sigma}(w) > L_{x_\sigma}(p_t).$$
    Then the active path is updated by setting $p_{t+1} = \min\{v_{\kappa+1}, v_{\kappa+2}\}$. All intermediate nodes between the last node of Y_m and p_{t+1} forms a jumped path. The mark m is set to s.

  In both cases, there is no new dead-end path, and the direction d is set to $+1$.
- **Case 2.2:** $p_t \in \cup_{i=m+1}^{s} X_i$. Then the direction d is set to -1. Let $t' \in \{0, \dots, t-1\}$ be the largest index such that $p_{t'} \notin \cup_{i=m+1}^{s} X_i$. All nodes $p_{t'+1}, \dots, p_t$ forms a dead-end path. Assume that $p_{t'+1} \in X_\sigma$, then if $\sigma > m + 1$, all intermediate nodes of $\cup_{i=m+1}^{\sigma-1} Y_i$ form a jumped path. The mark m is updated to $\sigma - 1$.