Information exchange for routing protocols

Ángela I. Barbero[1] and Øyvind Ytrehus[2]
[1]Dept. of Applied Mathematics, Universidad de Valladolid, 47011 Valladolid, Spain
[2]Dept. of Informatics, University of Bergen, N-5020 Bergen, Norway
Email: angbar@wmatem.eis.uva.es, oyvind@ii.uib.no

***Abstract*—Distance vector routing is a classic distributed algorithm for obtaining routing tables in a communication network. The algorithm relies on message exchange between neighbor routers. This paper studies the amount of routing data that needs to be stored and exchanged. On a static network, a variation of the algorithm that exchanges routing trees or pseudotrees is slightly more information theoretically efficient than a traditional implementation that exchanges tables. Knowledge of an underlying graph model and proper estimation of parameters allow more efficient coding schemes, including schemes related to Slepian-Wolf coding. Further improvements can be obtained on a dynamic network.**

Keywords: Communication networks, distance vector algorithm, information theory, coding.

I. Introduction

Distance vector routing (DVR) is a classic technique [1], [2], [3] for obtaining minimum distance routing tables in a communication network in a distributed way. The algorithm, described in Section II-A, relies on message exchange between neighbor routers. DVR is used in the Internet in the form of the Routing Information Protocol (RIP) [4] and its extensions [5], [6].

Routing has traditionally received attention from graph theorists focusing on algorithms, with focus on the algorithmic complexity of in-node processing. However, there has not been much research on the amount of information that needs to be exchanged in routing protocols. Only relatively recently, information theorists have started to study routing algorithms such as *link state* routing [7]. And optimization approach is presented in [8].

On the other hand, the simple distance vector routing protocol seems to be overlooked in the information theory literature. This is a little surprising, since the simple protocol is shunned in certain applications precisely because of the amount of data that is exchanged in typical implementations of the protocol.

A. Contributions of this paper

In this paper we study the amount of information required for storage and exchange the basic DVR protocols. As far as we know, the DVR algorithm has not previously been studied from the information theoretic point of view. We study static and dynamic network models. In a static network model, we find that storage and exchange can be improved, but only marginally, unless we can make assumptions on the network model. Under some assumptions on the network model, one can represent the routing messages much more efficiently. In addition, for the more realistic case of dynamic networks, we suggest a variation of the algorithm that is more efficient in terms of message exchange.

B. Paper overview

Section II describes the network model we use, and introduces the basics of the distance vector routing algorithm. In Section III we discuss message sizes and storage requirements of the basic DVR algorithm, without assumptions on the network model. Section IV focuses on message sizes in a static network based on a Gilbert random graph; in some circumstances the message size can be considerably reduced. Then in Section V we go on to start an investigation in the case of dynamic networks. Finally, Section VI contains a discussion of the implication of the results as well as a list of new research directions.

II. Background, notation, and network model

A communication network is described in this paper* by a directed graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of directed edges. For convenience, we will use the terms *edge* and *link* as synonyms, and by *node* we will mean any router, station, or host computer. Each edge is assumed to have *unit capacity*.

A. Distance vector routing

Consider the situation in Fig. 1, which shows a connected network with $n + 1 = 7$ nodes. One node is called *me* and has a connection to node A, which we will sometimes call the *reporting node*.

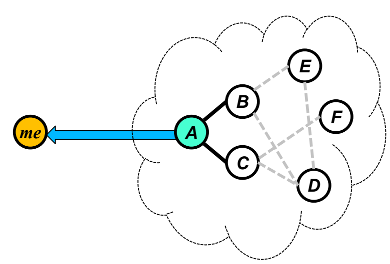

Fig. 1. Information exchange

The reporting node A has to inform *me* about the complete set of nodes (except *me*) that A knows that it is connected to, and its minimum distance to these nodes. In the example of

*The results may be adapted to more general models that also describe for example broadcast links or more diverse link capacities.

Destination node id:	$\mathbf{id}_1$	$\mathbf{id}_2$	$\cdots$	$\mathbf{id}_{n-1}$
Distance:	d_1	d_2	$\cdots$	d_{n-1}
Outgoing link:	L_{i_1}	L_{i_2}	$\cdots$	$L_{i_{n-1}}$

TABLE I
EXAMPLE OF A ROUTING TABLE.

Fig. 1, this would be the other $n-1=5$ nodes B, C, D, E, F. If there are other nodes connected to *me*, they also need to inform *me* in the same way, and it is up to *me* to decide which outgoing links to use for forwarding of data packets to a given destination node.

Distance-Vector routing (DVR) is an example of a distributed version of the Bellmann-Ford algorithm. The purpose of the algorithm is to determine the shortest path from each node to every other node. This path is represented in each node by a *routing table*. For the case of a sub-network with n nodes, this routing table at each node has $n-1$ columns and two or three rows, in the form of Table I.

Depending on the precise implementation, the first line may be implied by the order of id's of the $n-1$ neighbours, and can be omitted for actual representation†. On the other hand, if the id's of the neighbours are not implicitly known, the id's (e. g., IP addresses) must be explicitly represented.

When a node has a data packet to forward to a destination node with id $\mathbf{id}_k$, the sending node sends it to the corresponding outgoing link L_k.

The distance can in principle be any non-negative number, representing for example cost of using the line, or delay. In the version most commonly used in the Internet (because it is simple) and which we also adopt here (because it is simple), each link has a unit cost, and the distance metric is the number of hops.

In this distributed version of the BF algorithm, the nodes exchange information (only) with its nearest neighbours. This information, corresponding to the arrow from node A to *me* in Fig. 1, is conveyed in messages that contain the sending node's current routing table (I) *except for the last row*, which contains information which is irrelevant to *me* in the BF algorithm. In the standardized protocols used in the Internet, like RIP in different versions, this routing table exchange takes place at regular intervals.

In addition, each node will monitor the distance $d(\mathbf{id})$ to each of its neighbors **id** on the corresponding link, (which, as noted above, in our case is 1). Each time an updated routing table arrives, the current node will update each column (i. e. each destination node $\mathbf{id}_k$) of its routing table according to the following rules:

1) Let $d_k(old)$ be the distance to node $\mathbf{id}_k$ according to the old table, let $d_k(new)$ be the distance according to the incoming table, and let $d(neighbor)=1$ be the distance of the link.
2) If the incoming routing table message arrives on a link L that is *not* used for node $\mathbf{id}_k$, then update the routing table if $d(neighbor)+d_k(new) < d_k(old)$ with the new distance $d(neighbor)+d_k(new)$ and the new link.
3) If the incoming routing table message arrives on a link L that *is* currently used for node $\mathbf{id}_k$, then update the routing table with the new distance $d(neighbor)+d_k(new)$.

Thus, in distance vector routing, each node executes the following algorithm:

Algorithm 1 The DVR algorithm

Determine who are the neighbors
Initialize routing table: For each neighbor node, set the distance to 1 and the outgoing link to the obvious value
while *network is active* **do**
  Determine who are the neighbors now
  At regular intervals, exchange routing tables with the neighbors
  Calculate new minimum distances to each destination node, and update routing tables accordingly
end while

B. Advantages & disadvantages

The DVR algorithm is distributed and very simple for small networks. However it has a few drawbacks:

- Scalability: The amount of routing information *seems* (in typical implementations) to grow more than linearly in the number of nodes in the network. The practical approach to solving this is to apply the algorithm in a hierarchical way. The network is divided into clusters, with dedicated contact nodes
- Count-to-infinity: The algorithm reacts fast to good news (new edges, nodes, and paths becoming available), but slowly to bad news (edges, nodes, and paths disappearing.) There exist heuristic ways to deal with this.
- *The amount of exchanged routing protocol data appears to be prohibitive for some applications,* such as mobile ad-hoc networks. This paper is dedicated to the study of how much information really needs to be exchanged.

III. MESSAGE SIZE AND STORAGE SIZE IN THE BASIC DVR ALGORITHM

In this section we study the amount of information that needs to be exchanged between neighbor nodes, and the amount of information that needs to be collected by and stored at each node in the DV routing algorithm. The exact set of node identities which is known to the reporting node is independent of the network structure among those nodes. Thus, in a static network, the first row of (I) can be represented independently of the rest. So when we talk about table sizes and message sizes in Sections III and IV, *we assume that the number and identity of known nodes are otherwise represented*‡.

†In this case the table also need to be able to represent a distance ∞ for nodes that cannot be reached.

‡Strictly speaking, we are cheating here: The set of identities is independent of the structure, but the number of known nodes is not.

A. Message exchange

In this section we investigate the actual amount of information contained in each exchanged message.

If messages are implemented directly as in Table I, then if a reporting node (A in Fig. 1) knows $n-1$ nodes apart from *me*, and since the distance is nonzero and at most $n-1$ and hence can be represented by $\log_2(n-1)$ bits, the message size is

$$M_{\mathrm{TAB}} = (n-1)\log_2(n-1). \tag{1}$$

1) Information in terms of routing trees: We will need the definition of entropy:

Definition 1: Consider a discrete stochastic variable X with set of possible outcomes $\mathcal{X}$ and probability mass function $p(x)$. The *entropy* of X, measured in bits, is

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x). \tag{2}$$

In particular, in the (worst) case where $p(x)$ is the uniform distribution on $\mathcal{X}$, it holds that $H(X) = \log_2(|\mathcal{X}|)$.

Theorem 1: (When the number n and the identity of all nodes are known,) the size of routing messages can be reduced to

$$S_{\mathrm{Tree}} = \lceil (n-2)\log n \rceil. \tag{3}$$

Proof: The collection of all routing tables induces a routing tree for each outgoing link. This tree is not precisely known at the router, but the *set* of possible trees can be deduced. Any tree in this set can be picked arbitrarily to represent the set. The number of trees on n nodes rooted in a specific of them is n^{n-2} [9], [10]. Assuming a worst case uniform distribution on all trees and then applying the entropy function yields (3). ■

The tree representation is asymptotically of the same size as the table representation, but marginally smaller for finite network sizes. The tree representation contains more information about the network structure, but this information is of no use in the basic DVR algorithm. It may, however, be useful in other algorithms.

2) Information in terms of psedudotrees:

Definition 2: Let $T(0) = 1$ and

$$T(n) = \sum_{j=1}^{n} \binom{n}{j} T(n-j) \tag{4}$$

for $n \geq 1$. $T(n)$ is a well known sequence that gives the "Number of preferential arrangements of n labeled elements; or number of weak orders on n labeled elements. " [11]. In this paper, for convenience we will call this kind of object a *pseudotree* because it almost captures the structure of a tree. More precisely, it captures the essence of the second row of the routing table in the form of (I), with the extra condition that it *corresponds to an underlying tree structure,* where the precise tree may not be completely determined: Given a pseudotree/table, one can deduce the precise underlying tree structure up to but not beyond the first fork in the tree.

Example 1: Consider a network of the type illustrated in Fig. 1, but where the reporting node A knows that there are paths to $n-1=2$ other nodes except *me*: B and C. See also Fig. 2. The three ($= T(2)$) pseudotrees can be represented as $[(B,1),(C,1)]$, $[(B,1),(C,2)]$, and $[(C,1),(B,2)]$, respectively, where each pseudotree is listed in short notation as $[(\mathbf{id}_1,d_1),\ldots,(\mathbf{id}_{n-1},d_{n-1})]$. For convenience and uniqueness, we apply the convention that this list is sorted first in ascending order of the distances d_j and then secondly in some arbitrary order on $\mathbf{id}_j$. In this case number of pseudotrees coincides with the number of trees, and the tree structure is uniquely determined from a given pseudotree, but for $n > 3$ this is no longer always the case. Please also observe that an unrestricted table implementation will allow a fourth case: $[(C,2),(B,2)]$; however the latter case does not correspond to any tree.

3) Exchange of pseudotrees:

Theorem 2: The message size can be reduced to $\lceil \log_2 T(n-1) \rceil$.

Proof: $T(n-1)$ is the number of pseudotrees for a connected graph with n nodes including the reporting node. Hence, the number of bits required is at most $\log_2 T(n-1)$. ■

Large values of n. Since

$$\begin{aligned}
T(n) &= \sum_{k=1}^{n}\binom{n}{k}T(n-k)\\
&= nT(n-1) + \sum_{k=2}^{n}\binom{n}{k}T(n-k)\\
&= nT(n-1) + \sum_{k=1}^{n-1}\binom{n}{k+1}T(n-k-1)\\
&= nT(n-1) + \sum_{k=1}^{n-1}\frac{n}{k+1}\binom{n-1}{k}T(n-k-1)\\
&\leq nT(n-1) + \frac{n}{2}T(n-1),
\end{aligned}$$

it holds that $n \leq \frac{T(n)}{T(n-1)} \leq \frac{3}{2}n$ (while, for example, the growth rate of the number of trees is asymptotically en.) Thus, the improvement using pseudotrees as opposed to trees, in terms of bits in the representation, grows linearly with n. However, *relatively*, the difference goes to zero: All the schemes considered so far grow like $O(n\log n)$.

B. Message storage size

This differs from the amount of information exhanged in the sense that it is also necessary to represent the output link, i. e. the last row in I). The node needs to know the exact set of reachable nodes, the first link on the way to get to each node, and the distance. Thus the table consists of $n-1$ columns, $\log_2(n-1)$ bits per column for encoding distances, and $\log_2(n-1)$ bits per column link for encoding the outgoing link, assuming no restrictions on the number of neighbors. Since there is no coding in the table, each table value is

n	Table (1)	Tree n^{n-2}	Tree $\lceil \log_2(\leftarrow) \rceil$	P.tree $T(n-1)$	P.tree $\lceil \log_2(\leftarrow) \rceil$	Storage (5)	Storage eq. (7)	Storage $\lceil \log_2(eq.(7)) \rceil$
3	**2**	3	**2**	3	**2**	**4**	3	**2**
4	**5**	16	**4**	13	**4**	**10**	16	**4**
5	**8**	125	**7**	75	**7**	**16**	113	**7**
6	**12**	1296	**11**	541	**10**	**24**	986	**10**
7	**16**	16807	**15**	4683	**13**	**32**	10237	**14**
8	**20**	262144	**19**	47293	**16**	**40**	123096	**17**
9	**25**	4782969	**23**	545835	**20**	**49**	1680737	**21**
10	**29**	100000000	**27**	7087261	**23**	**58**	25668766	**25**
20	**81**	-	**78**	-	**67**	**162**	-	**71**
30	**141**	-	**138**	-	**118**	**282**	-	**124**
40	**207**	-	**203**	-	**174**	**413**	-	**181**
50	**276**	-	**271**	-	**235**	**551**	-	**243**
60	**348**	-	**343**	-	**297**	**695**	-	**307**
70	**422**	-	**417**	-	**363**	**843**	-	**373**
80	**498**	-	**494**	-	**430**	**996**	-	**442**
90	**577**	-	**572**	-	**500**	**1153**	-	**512**
100	**657**	-	**652**	-	**570**	**1313**	-	**584**

TABLE II
MESSAGE AND STORAGE SIZES FOR THE DIFFERENT ALTERNATIVES.

represented by an integer number of bits, and the size of the table is therefore

$$S_{\mathrm{TAB}} = 2(n-1)\lceil \log(n-1) \rceil. \tag{5}$$

Alternatively, a tree representation contains all the information that is in the table, and can in principle (if not in practice) be used directly. A pseudotree representation requires a little more refinement, since there is a pseudotree attached to each outgoing link; thus from a routing table one can deduce a little more about the network structure than from a pseudotree alone. We therefore need the following notation.

Definition 3: Let k and n be integers with $k \leq n$ and let $\underline{\ell} = (\ell_1, \ldots, \ell_k) \in \mathbb{N}^k$, where $\mathbb{N}$ is the set of positive integers, be an integer partition of the integer n, i. e. $\ell_1 + \cdots + \ell_k = n$, where each $\ell_i \geq 1$. For convenience we also impose the condition that $\ell_i \geq \ell_{i-1}, i = 2, \ldots, k$. Further let m be the number of different integers in $\underline{\ell}$, and let $r_1, r_2, \ldots, r_m$ be the number of copies of each of these m integers in $\underline{\ell}$. Then define

$$\gamma(\underline{\ell}) = \frac{\prod_{j=1}^{k} \binom{n - \sum_{i=1}^{j} \ell_{i-1}}{\ell_j}}{\prod_{i=1}^{m} (r_i)!} \tag{6}$$

In words, $\gamma(\underline{\ell})$ is the number of ways to choose k nonempty subsets of a set of n elements with subset sizes $\ell_1, \ldots, \ell_k$.

Lemma 1: Let k be the number of outgoing links in a node. The message size can be reduced to $\log_2(S_{\mathrm{CT}}(n+1))$, where

$$S_{\mathrm{CT}}(n+1) = \sum_{k=1}^{n} \sum_{\underline{\ell}} \gamma(\underline{\ell}) \prod_{i=1}^{k} \ell_i \cdot T(\ell_i - 1) \tag{7}$$

where the second summation runs over all $\underline{\ell} \in \mathbb{N}^k$ such that $\ell_1 + \cdots + \ell_k = n$ and $\ell_1 \geq 1$ and $\ell_i \geq \ell_{i-1}, i = 2, \ldots, k$.

Proof: Enumeration of pseudotrees conditioned on the distribution of nodes on the k outgoing links, then applying entropy under an assumption of uniform probability. ■

Remark. The tables need to be available for the nodes in a data structure that is convenient for use in packet forwarding and also that can be easily updated. The basic table implementation is likely to be the most convenient data structure for this. However, the results in this section are included for theoretical reasons, and also to show that the representation of (I) is actually close to optimum. This is especially true when considering that also the first line of (I) needs to be represented.

IV. EXCHANGE OF PSEUDOTREES IN A GILBERT RANDOM GRAPH WITH KNOWN EDGE PROBABILITY

The proof of the previous results assumes enumerative encoding over the trees. This makes sense if all pseudotrees are equally likely, but this is not necessarily reasonable in a typical network (whatever that means.) In order to consider probability distributions on families of networks, we will expand the rough network model introduced in Section II.

Definition 4: A Gilbert random graph[12] is a graph where there is a link between every pair of nodes with probability P.[§]

Figure 2 shows all eight graphs on $n = 3$ nodes, together with the associated probability $P^w(1-P)^{3-w}$ that each graph with exactly w links occur as a Gilbert random graph.

Please observe that with respect to the routing messages/pseudotrees for node A, graphs $\boxed{0}$ and $\boxed{7}$ correspond to the same tree, $\boxed{0}$, and hence also the same pseudotree. Similarly, graphs $\boxed{2}$ and $\boxed{5}$ correspond to the same tree, $\boxed{2}$, and the same pseudotree. That is, with respect to the pseudotree, edges can be added to or deleted from a graph without changing the pseudotree as long as these edges are not present in the pseudotree. Such edges are those that are between pairs of nodes on the same distance from the reporting node A, or those that are between nodes at distance i and nodes at distance $i+1$, as long as there remains at least one such edge for each node at distance $i+1$.

[§]Gilbert random graphs are for many purposes similar to Erdős-Rényi random graphs [13], but in this context not, since the latter random graph will have fixed node degrees.

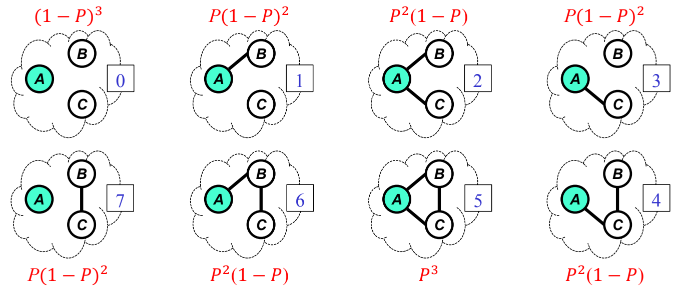

Fig. 2. The eight different networks on three nodes, labelled $\boxed{0}$,...,$\boxed{7}$.

Definition 5: The *distance distribution* of a pseudotree

$$\mathcal{T} = [(\mathbf{id}_1, d_1), \ldots, (\mathbf{id}_{n-1}, d_{n-1})]$$

is the sequence

$$\begin{aligned} D(\mathcal{T}) &= [D_1, \ldots, D_m] \\ &= [\sharp \textit{ nodes at dist. 1 from the reporting node}, \ldots, \\ &\quad \ldots, \sharp \textit{ nodes at dist. } m \textit{ from the reporting node }]. \end{aligned}$$

where m is the maximum distance between the reporting node and any node that it knows about.

For a given pseudotree $\mathcal{T}$ on n nodes, let $g_{D(\mathcal{T}),w}$ be the number of graphs with exactly w edges that after a suitable distance vector processing causes A to report to *me* exactly the pseudotree $\mathcal{T}$. Further, let

$$G_{D(\mathcal{T})}(x) = \sum_{w=n-1}^{\binom{n}{2}} g_{D(\mathcal{T}),w} x^w$$

be a generating function for $g_{D(\mathcal{T}),w}$. The lower and upper bounds of the summation correspond to the cases where the subtree that is being reported are, respectively, a tree or a complete graph.

Lemma 2: The generating function is given by

$$\begin{aligned} G_{D(\mathcal{T})}(x) &= G_{[D_1,\ldots,D_m]}(x) \\ &= (x+1)^{\sum_{i=1}^m \binom{D_i}{2}} \prod_{j=1}^{m} ((x+1)^{D_{j-1}} - 1)^{D_j}, \end{aligned} \quad (8)$$

where by convention $D_0 = 1$.

Proof: Start with a pseudotree with the given distance distribution, and form one arbitrary of the corresponding trees without any unnecessary edges. Considering the set of edges that can be added without altering the pseudotree, it follows that (8) counts the number of distinct graphs that each will result in the same pseudotree T: The first term corresponds to the number of distinct graphs of each weight that can be created by toggling edges between pairs of nodes at the same distance from the reporting node A. The product term counts the number of distinct graphs that can be created by toggling edges between nodes at distance $j - 1$ and j, while always keeping at least one edge to each node at distance j. ∎

Theorem 3: The probability that a random Gilbert graph with edge probability P produces the pseudotree $\mathcal{T}$ at the reporting node A, conditioned on the fact that the graph has exactly n nodes is

$$Pr(\mathcal{T}) = \frac{G_{D(\mathcal{T})}(\frac{P}{1-P})}{\sum_{|\mathcal{T}'|=n} G_{D(\mathcal{T}')}(\frac{P}{1-P})}. \quad (9)$$

Proof: The numerator is the sum of probabilities over all graphs that reduce to the pseudotree $\mathcal{T}$. Using Bayes' formula, divide by the denominator, which is the probability that the pseudotree has n nodes. ∎

Applying the entropy function to the probability distribution $Pr(\mathcal{T})$ as a function of P, we get the entropy in Figure 3. Note that for small values of P, with high probability the random graphs will usually not be connected. The critical value of P where the probability of a *giant component* becomes significant [14] is indicated in the figure; normally a network should operate with P higher than this value in order to be useful.

V. Information Exchange in Dynamic Networks

In a static network, after a while the tables of routing converge and remain unchanged. Still the standard protocol mandates regular message exchanges. This is required to keep up with changes in a dynamic network. The real potential for savings in information exchange is of course offered in a dynamic network. In order to analyze this we need to introduce a formalism for dynamic networks.

A. Models for dynamic networks

Networks are dynamic due to changes in link capacity, changes in the absence or presence of nodes and arguably, depending on definition, also changes in traffic patterns. Dynamism may occur for immobile networks (e. g. plugging/unplugging of cables, turning on or off computers) as well as in wireless networks of mobile nodes moving into and out of each others wireless communication range. Clearly the physical characteristics of this dynamism heavily influences the experienced network characteristics.

We will introduce some simple models of dynamism. In order to deal with combinatorial issues we will rely on a preliminary lemma on random walks in Hamming space.

Lemma 3: Let $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)}) \in \mathbb{F}_2^N$ be a binary N-dimensional vector observed at (discrete) time t, where t is an integer. Assume that each bit $x_i^{(t)}$, $1 \le i \le N$, at time t can change into $x_i^{(t+1)}$ at time $t+1$ independently from the others and according to the conditional probability distributions

$$Pr(x_i^{(t+1)} = u | x_i^{(t)} = v) = \begin{cases} 1-p & \text{if } u = 0, v = 0 \\ p & \text{if } u = 1, v = 0 \\ q & \text{if } u = 0, v = 1 \\ 1-q & \text{if } u = 1, v = 1 \end{cases} \quad (10)$$

Now consider the N-dimensional Markov chain represented by the sequence

$$\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}, \ldots$$

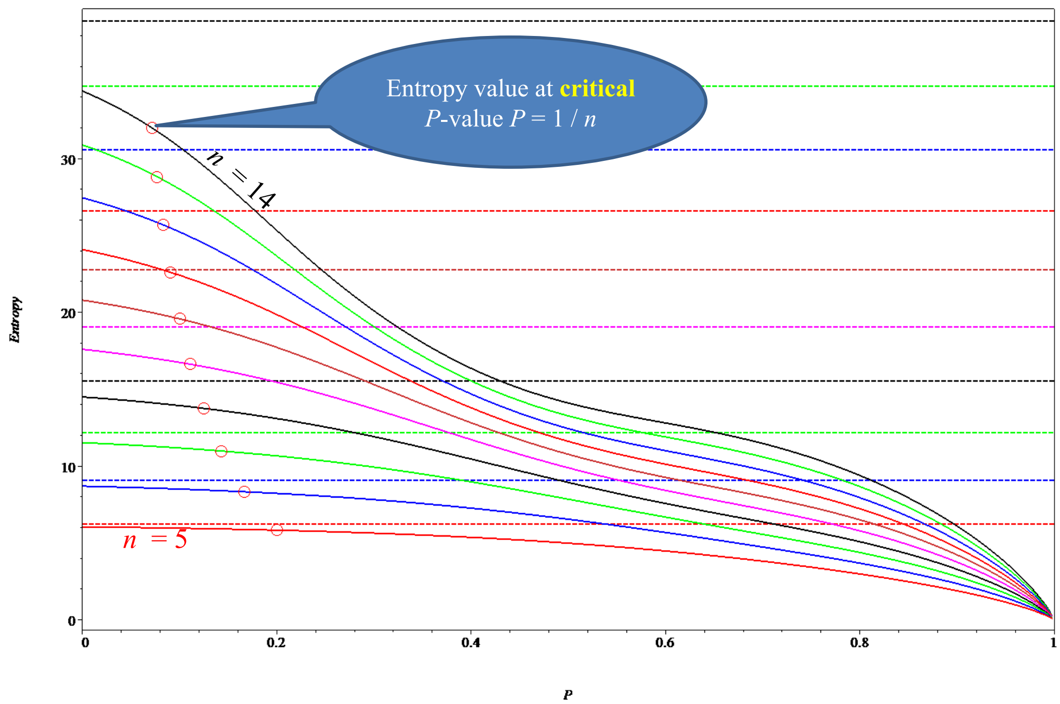

Fig. 3. Entropy of routing information in a static network with an underlying Gilbert graph of edge probability P. It is assumed that the reporting node can report about a known network of n nodes, including itself, to a node *me* which is not among these n. The solid line curves show the entropy functions for $n = 5, \ldots, 14$. For comparison, the corresponding horizontal dotted lines show the (unrounded) values $\log_2(T(n-1))$.

Then the steady state probability distribution over $\mathbb{F}^N$ of this Markov chain is given by

$$\lim_{t\to\infty} Pr(\mathbf{x}^{(t)} = \mathbf{x}) = \frac{p^{w(\mathbf{x})} q^{N-w(\mathbf{x})}}{(p+q)^N}, \tag{11}$$

where $w(\mathbf{x})$ is the Hamming weight of $\mathbf{x}$.

Proof: Start by assuming $N = 1$. Then it can readily be verified that the steady state distribution is given by

$$\lim_{t\to\infty} Pr(x^{(t)} = 1) = \frac{p}{p+q} \triangleq P,$$
$$\lim_{t\to\infty} Pr(x^{(t)} = 0) = \frac{q}{p+q} \triangleq 1 - P.$$

Next apply the probabilities P and $1 - P$ to the binomial probability distribution, and (11) follows. ■

Example 2: Consider networks on three nodes, and now with a very simple model of a dynamic network. Suppose that if there is no edge between two nodes at time t, then with probability p there is an edge at time $t+1$. Similarly, if there is an edge between two nodes at time t, then with probability q there is no edge at time $t+1$. We form an N-dimensional binary vector, with $N = \binom{3}{2}$ = the number of possible edges in this graph, and let each of the N elements of this vector indicate whether an edge is up or down. When we apply Lemma 3 to the vector, we obtain the steady state probability distribution for the set of graphs. Observing this process at an arbitrary point in discrete time, we actually get a model which is consistent with the static network models discussed in Section IV. The upper half of Figure 4 shows the set of possible networks at time t, the unique routing corresponding to each graph, and the pseudotree corresponding to each tree. The lower half shows the same at time $t+1$. The arrows from the upper to the lower half shows possible transitions. Here we have made the simplifying but reasonable[¶] assumption that there is at most one edge change per message exchange. This enables an efficient coding of routing messages. If, during a brief period, the number of changes exceeds the assumed average values, the message passing procedure will still catch up after a while.

This example illustrates some issues that apply regardless of the specific model of dynamism in the network:

1) There is an (entropy-reducing) dependency between *graphs* at time t and at time $t+1$.
2) For dynamic networks, also *node identities* depend on the state of the system at the previous time instant, and should no longer be encoded separately.
3) If nodes know about and exchange table information or pseudotrees, but not the underlying graphs, this dependency is diffused, but not lost entirely. This may still be an argument to derive a new algorithm that exchanges

[¶]The reasonable assumption is that changes *on average* are few, otherwise there is reason to believe that the protocol will collapse regardless of how routing information is represented.

routing trees instead of routing tables. After all the size of messages is similar, but such a protocol may offer a better encoding scheme than DVR in a dynamic network since it is "closer to" the physical changes in the network.

Similar examples can be made about other models of dynamic networks, such as nodes becoming active or passive, or mobile nodes in a wireless network. The details are beyond the scope of this paper.

VI. Conclusions, remarks, and future work

We observe that there are many open problems in this area. Space/time limitations prohibit us from going into detail, but we briefly point out a few:

- What are the limits in terms of dynamism before a routing protocol breaks down?
- Is there a tradeoff between message exchange interval and message size, given a set of dynamics parameters, in order to minimize convergence time and/or routing protocol bandwidth?
- What is the best way to estimate P in a Gilbert graph?
- Do there exist efficient encoding and decoding algorithms for the routing messages and storage methods discussed in this paper?
- Are there connections between iterative decoding and this class of routing protocols? One heuristic ("split horizon") used in newer versions of RIP, in order to reduce the count-to-infinity problem, is to limit the flow of information: I. e, the reporting node A should not report to *me* an optimum route that passes through *me* [6]. This is akin to a belief propagation principle: A should not report to *me* a piece of information which has been received from *me*, but may pass information received from other sources. The latter principle is slightly more permissive. It is not clear if and how this can be used to improve performance.
- There are multiple objectives in routing, for example fast routing convergence, efficient use of bandwidth, and efficient computation in the nodes. Are these objectives independent, or in conflict?
- The entropy of a given network structure is studied in [15].
- Routing messages to *me* from different neighbors contain information about the same underlying network structure, and each message will potentially reduce the entropy (for *me*) about this network structure. Thus these messages are correlated, and hence one can use the Slepian-Wolf theorem [16] to reduce the message size, even with independent encoders. However, in practice it may be difficult to obtain the underlying network parameters that can be used to quantify this correlation, and hence it can be difficult to implement the basic Slepian-Wolf encoding scheme. Fortunately, due to iterated exchange of information, one can use dependent routing messages to achieve the same message compression. How to achieve this in practice is currently being investigated and will be reported in future papers.

Conclusion: For finite-size networks, there is a small saving in storage and message size by choosing other data structures than tables, but not much, and in the asymptotic case the *factor* of relative saving it is zero. If a network model can be assumed, and one can estimate some simple network parameters (as in the case of the Gilbert graphs), encoding of routing information can be made much more efficient.

A tree representation of messages is smaller than the table representation, and provides more information of network structure. This may be useful in terms of resilience against network changes. More importantly, a tree representation preserves state transition information better than tables/pseudotrees do. As mentioned in Example 2, this suggests that a "routing tree" algorithm can be designed more to be information theoretically more efficient than the distance vector algorithm.

References

[1] L. R. Ford Jr., "Network Flow Theory," Paper P-923. Santa Monica, California: RAND Corporation, August 14, 1956.

[2] E. M. Moore, "The shortest path through a maze," *Proc. Internat. Sympos. Switching Theory 1957,* Part II. Cambridge, Mass.: Harvard Univ. Press. pp. 285–292, 1959.

[3] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics,* 16:87–90, 1958.

[4] "RFC1058 - Routing Information Protocol", *Internet Engineering Task Force,* June 1988.

[5] "RFC1723 - RIP Version 2 Carrying Additional Information", *Internet Engineering Task Force,* November, 1994.

[6] "RFC2453 - RIP Version 2", *Internet Engineering Task Force,* November, 1998.

[7] G. Cheng and N. Ansari, "An Information Theory Based Framework for Optimal Link State Update", *IEEE Comm. Letters,* Vol. 8, no. 11, November 2004, pp. 692–694.

[8] Fang Zhao, Muriel Médard, Asuman Ozdaglar, Desmond Lun, Member, "Convergence Study of Decentralized Min-Cost Subgraph Algorithms for Multicast in Coded Networks", IEEE Transactions on Information Theory, vol. 60, no. 1, pp. 410–421, January 2014.

[9] C. W. Borchardt, "Über eine Interpolationsformel für eine Art Symmetrischer Functionen und über Deren Anwendung," *Math. Abh. der Akademie der Wissenschaften zu Berlin,* 1–20, 1860.

[10] A. Cayley, "A theorem on trees," *Quart. J. Math,* 23: 376–378, 1889.

[11] The On-Line Encyclopedia of Integer Sequences, OEIS®, http://oeis.org/ .

[12] E. N. Gilbert, "Random Graphs," *Annals of Mathematical Statistics,* 30 (4), pp. 1141–1144, 1959.

[13] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5, pp. 17–61, 1960.

[14] Béla Bollobás, *Random Graphs,* 2nd Edition, Cambridge University Press, 2001.

[15] Eun Jee Lee, Michael Langberg, and Michelle Effros, "Outer bounds and a functional study of the edge removal problem," *Proc. Information Theory Workshop (ITW)*, Sevilla, Spain, September 2013.

[16] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," IEEE Transactions on Information Theory 19, pp. 471–480, 1973.

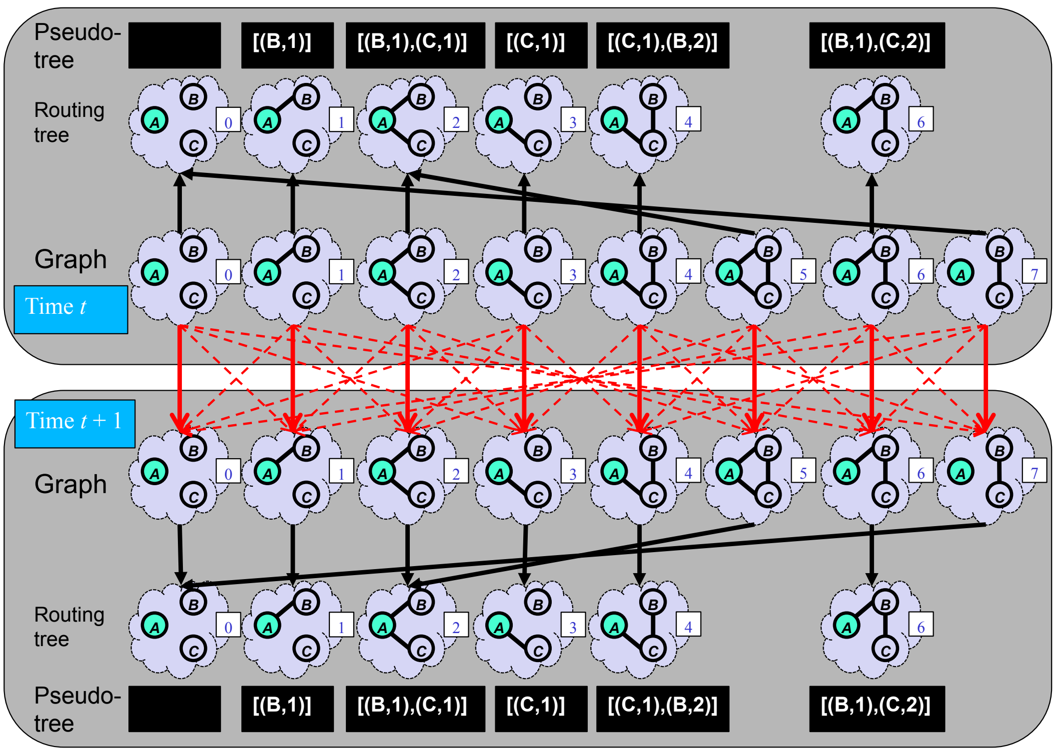

Fig. 4. The relationship between graphs, routing trees and pseudotrees for networks on three nodes.