A Dynamic Graph Algorithm for the Highly Dynamic Network Problem

Edwin Soedarmadji Robert J. McEliece

California Institute of Technology
Pasadena, CA 91125, USA
{*edwin,rjm*}@*systems.caltech.edu*

Abstract

A recent flooding algorithm [1] guaranteed correctness for networks with dynamic edges and fixed nodes. The algorithm provided a partial answer to the highly dynamic network (HDN) problem, defined as the problem of devising a reliable message-passing algorithm over a HDN, which is a network – or a network mobility model – where edges and nodes can enter and leave the network almost arbitrarily.

In this paper, we relax the flooding algorithms' assumptions by removing the requirement that the network stays connected at all time, and extend the algorithm to solve the HDN problem where dynamic nodes are also involved. The extended algorithm is reliable: it guarantees message-passing to all the destination nodes and terminates within a time bounded by a polynomial function of the maximum message transit time between adjacent nodes, and the maximum number of nodes in the network.

1 Introduction

Message-passing algorithms are among the most important algorithms in network communication. There are many different network message-passing algorithms for specific types of communication networks. This paper focuses on message-passing algorithms on mobile networks.

Mobile networks belong to a class of communication networks where nodes are "moving" with respect to the other nodes in a generalized parameter space that may include their physical location, antenna direction, transceiver power, and so on. In this general view, the notions of "mobility" and "motion" are not restricted to physical location.

These nodal motions affect the nodal edges, resulting in a *dynamic network*, broadly defined as a network with evolving topology and parameters. In practice, dynamic / mobile networks are most commonly implemented as wireless networks. Thus, in this paper, we interchange the terms dynamic network, wireless network, and mobile network.

When one talks about the space of all dynamic networks, one really talks about a vast and diverse spectrum of networks with different node and edge mobility models. On one extreme of the spectrum, we have the static network mobility model. In such a mobility model, communication nodes and edges move very slowly that they are assumed to be constant. On the other extreme of the spectrum, we have the stochastic network mobility models, where communication nodes and edges move almost unpredictably, and the only reasonable description of network motion is in terms of its stochastic parameters.

In this paper, we describe the *highly dynamic network* (HDN) mobility model that lies somewhere between these extremes. This mobility model is patterned after the model presented in [1]. In a dynamic network, nodes move in such a way that nodes and edges enter and leave the network as time evolves. We refer to these motions as the *node* and *edge mobility*, respectively. The HDN model is defined by a set of assumptions that distinguishes it from other types of dynamic networks. We describe them in detail in section 2.

The biggest question addressed in this paper is the HDN problem: *is there a reliable message-passing algorithm for a HDN with node and edge mobility?* A reliable algorithm guarantees that a message from the source reaches the destination nodes in a polynomially-bounded time period, i.e., it satisfies the correctness and termination criteria.

Previous message-passing algorithms in [2, 3, 5, 6, 8, 9] did not consider the HDNs and imposed stringent conditions on the network model. In [1], O'Dell and Wattenhofer designed flooding algorithms for dynamic networks with dynamic edges and fixed nodes. Among the assumptions used: the network stays connected at all time, and evolves slower than the message transit time. We found that the algorithm COUNTERFLOODING (CF) in [1] can be adapted to solve the HDN problem with dynamic nodes and edges.

The main contribution of this paper is in extending CF to a HDN model that supports node and edge mobility. First, we provide an alternative proof to CF. Our proof relaxes the various assumptions used in CF, most importantly the

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

stringent requirement that the network stays connected at all time. Next, we prove correctness and termination of the extended algorithm under the HDN assumptions.

The theoretical results presented in this paper describe the explicit set of assumptions required for guaranteed correctness and termination of flooding algorithms in a HDN. These results complement the results in [7, 10], which reported empirical success of the flooding algorithms called probabilistic routing and epidemic routing in the *intermittently connected networks* similar to the model in [1].

2 Formulation

Consider a set V containing all the network nodes under consideration. In practice, the number of nodes $|V|$ might only be known up to an upper bound $\bar{V}$. For wireless networks, these nodes represent wireless devices characterized by a generalized coordinate that specifies the devices' identity, physical location, antenna directivity, transceiver power, protocol, etc. Dynamic nodes "move" in their generalized coordinate and alter the network itself.

Denote by $V(t)$ or V_t the subset of V containing all non-isolated nodes in V with degree greater than zero at a given time t. The nodes v_1 and $v_2 \in V(t)$ are *connected* at time t if a hypothetical message with infinite speed launched from either node can reach the other. Disconnection may be due to incompatible protocol, path obstruction, insufficient power, internal filtering, or masking. Denote by v_0 the source node, and by V_d the set of all destination nodes d.

Connectivity between v_1 and v_2 at time t is represented by an edge $e(1,2,t)$, also denoted by $e_{12}(t)$, $e(t)$, or simply e. Denote by E, E_t, or $E(t)$ the set of all edges e at time t, and by $N(v,t)$ the set of all neighboring nodes to v at time t. At any given time t, the network is represented by $V(t)$ and $E(t)$, forming a graph G, also denoted by G_t, $G(t)$, and $G(V_t, E_t)$. Fig. 1 shows the sets V, $G(t)$, $V(t)$, and $E(t)$.

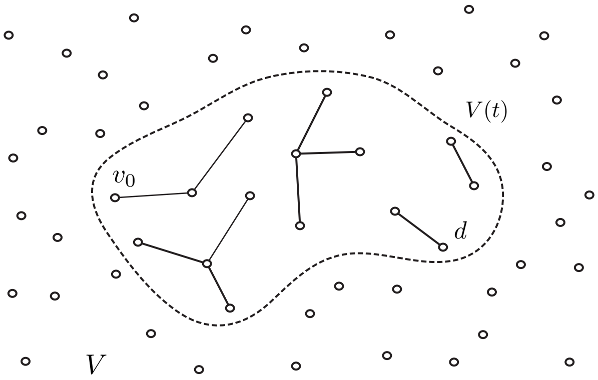

Figure 1: The sets V, $G(t)$, $V(t)$, and $E(t)$

The graph $G(t)$ is a dynamic system with discrete events that originates from the network's moving parts. From its two types of moving parts, $G(t)$ has two types of events.

(a) The *connectivity-driven* events $c = c(v,t)$ are generated (and received) by the nodes v – the network's main moving parts – whenever they gain or lose edges. For now, assume instantaneous detection of connectivity change. The case with delay is discussed later.

Denote by $C_v = \{\ t \mid N(v, t-\epsilon) \neq N(v, t+\epsilon)\ \}$ the set of all events $c(v,t)$ at a given v. Thus, C_v contains v's local vertex events. Denote by $C = \bigcup_v C_v$ the network global events. The events $c(v,0)$ mark the "creation" of $G(t)$.

(b) The *message-driven* events $m(v,t)$ are triggered at v when a new message m from $v' \neq v$ is received. Beside nodes, messages are also network moving parts as they move between the nodes in G. Denote by M_v the set of $m(v,t)$, and M the global counterpart.

We discussed how $G(t)$ consists of its dynamic components $V(t)$ and $E(t)$. The set V consists of two complementary sets of all nodes visited by (and thus stored) the message, and those yet to be visited, and still waiting for the message denoted by $V_\bullet(t)$ and $V_\circ(t)$, as shown in Fig. 2. At $t = 0$, $V_\bullet(0) = \{v_0\}$, and $V_\circ(0) = V \backslash \{v_0\}$.

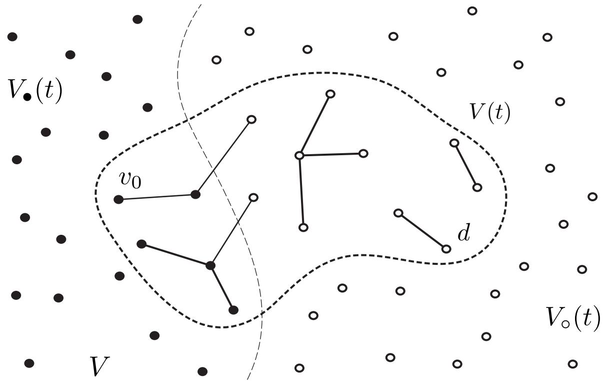

Figure 2: The sets $V_\bullet(t)$ and $V_\circ(t)$

Depending on whether a node v is in $V_\bullet(t)$ or $V_\circ(t)$, it processes the incoming events using the following schemes. If triggered by $c(v,t)$, a node $v \in V_\bullet(t)$ retransmits the message m in its storage to its neighbors $N(v,t)$. If triggered by $m(v,t)$, v retransmits and stores m. When $v \in V_\bullet(t)$ leaves $V(t' > t)$, it remains in $V_\bullet(t')$ and retains all its data.

With this scheme, a message m from the source node v_0 floods other nodes, including the destination nodes $d \in V_d$. Arrival of m at V_d depends on: (a) how far the nodes $d \in V_d$ are from v_0, (b) how rapidly $E(t)$ and $V(t)$ change, and (c) how fast m travels from $v' \in V(t)$ to $N(v',t)$.

Assumptions need to be made on $G(t)$ to ensure arrival of m at V_d within a finite amount of time. This requirement is effectively the correctness and termination criteria for a reliable message-passing algorithm on a dynamic network.

We begin by reviewing the assumptions used by O'Dell and Wattenhoffer in the definition of the CF algorithm [1] before discussing the assumptions used here:

1. Edge mobility, but not node mobility, i.e., $V(t) = V$,
2. $E(t)$ is such that $G(t)$ stays connected at all time,
3. Any change $\Delta E(t)$ in $E(t)$ instantaneously triggers connectivity-driven events at all the nodes affected,
4. Adjacent nodes can pass messages in less than τ,
5. Consecutive events in C_v are separated by at least τ,
6. Nodes store the messages they receive, and
7. Nodes know the value of $\bar{V}$

In the HDN assumptions used in this paper, assumptions 4, 5, and 7 are the same as the ones made in [1], but the other assumptions are relaxed, and assumption 6 is made explicit:

1. Node and edge mobility,
2. $E(t)$ is such that each node d in V_d stays connected to at least one node v in $V_\bullet(t)$ at all time,
3. Events in C_v may be delayed from $\Delta E(t)$,
4. Adjacent nodes can pass messages in less than τ,
5. Consecutive events in C_v are separated by at least τ,
6. Nodes store the messages they receive for τ_m, and
7. Nodes know the value of $\bar{V}$

In this paper, assumption 3 is generalized by introducing a delay between the actual connectivity change $\Delta E(t)$ and its corresponding event $c(v, t)$. Without this delay, a node needs to continuously broadcast its presence, resulting in poor energy efficiency. We assume that the nodes beacon their presence periodically every τ_c. This signal can be used to detect disconnections. In section 3, we derive the minimum τ_m required for reliable message-passing.

The receiving node masks the incoming beacon signal if it comes from a neighboring node that already sent the same message. Due to delay, assumption 5 should really say: adjacent events in C_v are separated by at least $\max(\tau, \tau_c)$.

3 Algorithm

In this section, we prove that CF is a reliable algorithm that solves the HDN problem. To prove this, we first prove the following claim that the number of nodes visited by the message at time $t + 2\tau$ is always strictly larger (by at least one node) than the number of nodes with message at t, as long as there is at least one destination node without the message. First, let us review the CF algorithm listed below.

```
if message $m(v,t)$ is received for the first time then
    Broadcast message $m$
    $k \leftarrow 0$
end if
if event $c(v,t)$ is received then
    if $k < 2\bar{V}$ then
        Broadcast message $m$
        $k \leftarrow k + 1$
    end if
end if
```

Lemma 1 *$|V_\bullet(t + 2\tau)| > |V_\bullet(t)|$ when $V_d \cap V_\circ(t) \neq \varnothing$.*

Consider one of the destination nodes $d \in V_d$. Denote by $V_\bullet(d, t)$ all nodes in $V_\bullet(t)$ connected to d. By assumption 2, $V_\bullet(d, t) \neq \varnothing$ at all time t. Refer to Fig. 3. Assumption 2 also implies that as long as $d \in V_\circ(t)$, there is at least one node $u \in V_\bullet(d, t)$ that is adjacent to some nodes $v \in V_\circ(t)$ (to see why this is true, consider the case where $v = d$). Denote by $V_\circ(d, t)$ all such adjacent nodes in $V_\circ(t)$.

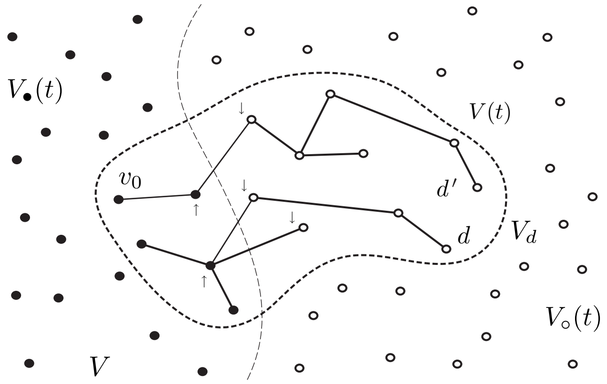

Figure 3: $V_\bullet(d, t)(\uparrow)$, $V_\circ(d, t)(\downarrow)$, $E(d, t)$, and V_d.

Denote by $E(d, t)$ the set of all edges connecting the nodes $u \in V_\bullet(d, t)$ to the nodes $v \in V_\circ(d, t)$. Consider an edge $e \in E(d, t)$ connecting u and v.

Two types of events trigger a message transmission from u to v: (a) when u detects the presence of v at time t_c, the event $c(u, t_c) \in C_u$ is generated, and (b) when a new message m reaches u at time t_m and converts u into the set $V_\bullet(d, t_m)$, the event $m(u, t_m)$ is generated. If a message has been previously received, no event is generated.

Now, we define the relationship between t, t_m, and t_c. We know that u and v are already connected at time t, so $t_c < t$. Further, since $u \in V_\bullet(d, t)$ at time t, then $t_m < t$. However, either $t_m < t_c$ or $t_c < t_m$, as shown in Fig. 4.

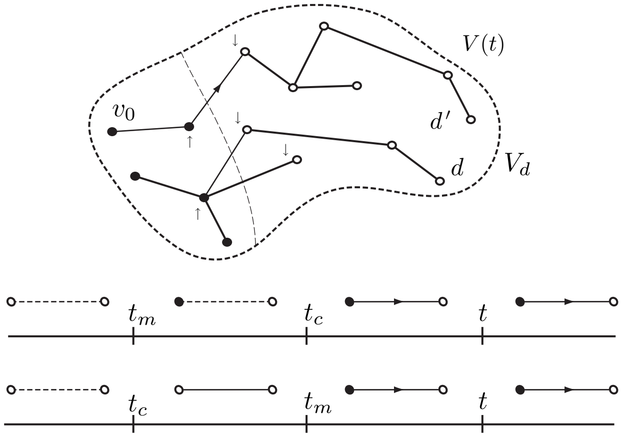

Figure 4: Two possible orders between t, t_c, and t_m

0-7695-2520-2/06 $20.00 © 2006 IEEE

In the first case of $t_m < t_c$, the connection is established after u already receives and stores the message, while in the second case of $t_c < t_m$, the message arrives after connection between u and v is established. In both cases, the message m is transmitted from u to v at time $\max(t_m, t_c)$. Let us consider these two mutually exclusive cases:

(a) If the message is transmitted at t_c, then by assumption 4 of HDN, it reaches v within τ. By assumption 5, up to time $t_c+\tau$: (i) the connection e established at time t_c cannot be broken, and (ii) the node v cannot leave the network. Otherwise, either one of these will produce an event in C_u less than τ apart from the previous event.

Therefore, at $t_c + \tau$, the node v joins $V_\bullet$. How about at $t + 2\tau$? If we set τ_m in assumption 6 to a very large number for now, then at $t + 2\tau$, the node v remains in $V_\bullet$. If the above is true for an edge e, then at $t + 2\tau$, the set $V_\bullet$ grows by one, and $V_\circ$ shrinks by one from their size at t.

(b) If the message is transmitted at t_m, then the situation is more difficult. Within the transit time τ for m to reach v from u, e might disconnect, or v might leave the network. In fact, these events might take place even before t_m!

We invoke assumption 2 to prove that such a failure cannot happen on *all* the edges in $E(d, t)$. If failure does not occur on the edge e, then assuming a large τ_m, a successful message transmission on e increases the size of $V_\bullet$ by one, and decreases the size of $V_\circ$ by one at $t_m + 2\tau$.

To prove this claim, consider separate timelines for each edge in $E(d, t)$, aligned with marks placed at time t. A message traveling along an edge e is represented by an interval $[t_m, t_m + \tau]$, placed on the timeline such that $t_m \le t \le t_m + \tau$. Since several edges may originate from the same node and thus have the same t_m, some intervals occupy identical locations on their timelines. Refer to Fig. 5.

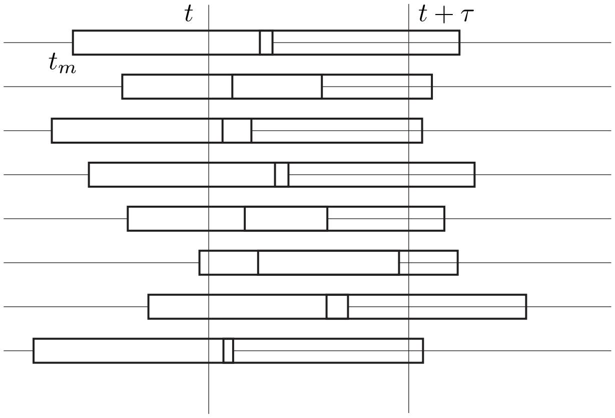

Figure 5: The timelines for $E(t)$ showing transit and dead regions

For a particular edge $e \in E(d, t)$, let t_d denote the time when either e disconnects, or v leaves the network. Since e exists at t, then $t_d > t$. If $t \le t_d < t_m+\tau$, then the message fails to reach v. For $\forall e \in E(d, t)$, call the region $[t_d, t_d + \tau]$ the dead region $d(e, t_d)$. Again, since several edges may head toward the same node and have the same t_d, some of these dead regions occupy identical time regions.

Select two of the timelines with the maximum and minimum values of t_m – denoted by t_m^* and t_{m*} – with their associated intervals. They must overlap because they both contain t. If these two extreme intervals overlap, then so must the other intervals with $t_{m*} < t_m < t_m^*$.

Suppose dead regions exist in *all* the intervals. Using the above argument, the dead regions of the maximum and minimum intervals must overlap because they contain $t+\tau$.

If all the intervals contain dead region, then they all overlap at $t+\tau$, i.e, the nodes in $V_\bullet(d, t)$ are completely disconnected from those in $V_\circ(d, t)$, and consequently from the destination node d, violating assumption 2.

Therefore, to preserve connectivity, there must be at least one interval without a dead region. A message successfully transmitted along the edge corresponding to this interval converts one node v from $V_\circ$ to $V_\bullet$ sometime between the edge's t_m and $t_m+\tau$. Assuming a large τ_m, then at $t+2\tau$, v stays in $V_\bullet$, implying that the set $V_\bullet$ grows by one node, and $V_\circ$ shrinks by one node from their sizes at t. So we proved cases (a) and (b) and continue with the rest of the proof.

So far, we assume that $d \in V_d$ is part of $V_\circ(t)$, separated from the nodes $u \in V_\bullet(d, t)$ by zero or more nodes $v \in V_\circ(d, t)$. For now, assume that $d \notin V_\circ(d, t)$. At least one of the nodes in $V_\circ(d, t)$ must remain connected to $V_\bullet(d, t)$ to satisfy assumption 2. Consequently, one of the nodes in $V_\circ(d, t)$ then joins $V_\bullet(d, t + 2\tau)$ at a later time. This proves that $|V_\bullet(d, t + 2\tau)| > |V_\bullet(d, t)|$ when $d \in V_\circ(d, t)$.

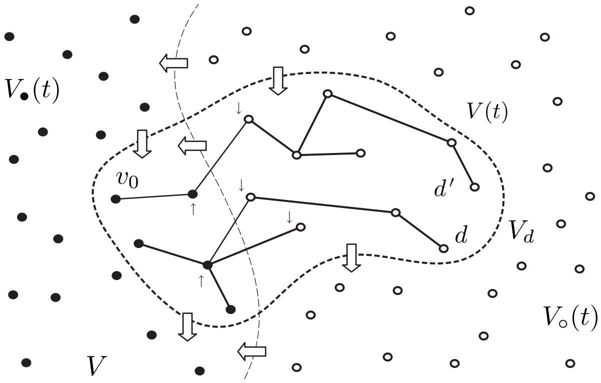

Figure 6: The arrows indicate allowable transitions of the network nodes in the sets $V_\circ(t)$, $V_\bullet(t)$, $V(t)$, and $V \backslash V(t)$ for HDN.

This process is repeated until $d \in V_\circ(d, t)$ (or equivalently, until $N(d, t) \cap V_\bullet(d, t) \neq \varnothing$). When $d \in V_\circ(d, t)$, then it will have received a message from a node in $V_\bullet(d, t)$ at $t + 2\tau$. Once $d \in V_\bullet(t)$, then one of its neighbors $n \in N(d, t)$ are also in $V_\bullet(t)$ and assumption 2 is automatically met. Consequently, the remaining nodes in $V_\circ(t)$ may or may not be in contact with the other nodes in $V_\bullet(t)$. This proves that $|V_\bullet(d, t+2\tau)| \geq |V_\bullet(d,t)|$ when $d \in V_\bullet(t)$.

Imagine the same process happening in parallel to all the destination nodes in V_d as shown in Fig. 6. Nodes in $V_\bullet(t)$ may leave $V(t)$ and become isolated, but when they re-enter the network at $t' > t$, they will be in $V_\bullet(t')$. The nodes in $V_\circ(t)$ is converted to $V_\bullet(t+2\tau)$ one by one.

The arrows in Fig. 6 indicate these transitions. Since $|V|$ is finite, the entire V_d is guaranteed to ultimately join $V_\bullet$ at some time t^*, at which point $V_\bullet(t > t^*)$ may or may not grow further. Thus we have finally proved lemma 1. □

Having proven the preceding lemma 1, we now prove that the CF algorithm is a solution to the HDN problem.

The algorithm requires one input $\bar{V}$ and intercepts two types of events: the connectivity-driven events $c(v,t)$ and the message-driven events $m(v,t)$. When the message is first received at v, the counter k is reset to 0, and the message is retransmitted to $N(v,t)$. If v is notified of a neighborhood change with the arrival of $c(v,t)$, it increments the counter k by one until it reaches the maximum $2\bar{V}$.

Theorem 1 *The algorithm CF solves the HDN problem and terminates in less than* $2\,\tau\,\bar{V}$.

To prove that CF is a solution to the HDN problem, we have to prove two things. First, we have to prove that a message sent from the source using CF under the HDN assumption is guaranteed to reach the destination nodes. However, this was already proved in lemma 1. The counter k is used to ensure that CF does not broadcast after termination.

Second, we have to prove that CF ends in finite time. Consider one destination node d. In the worst case scenario, $|V_\bullet(t)|$ is equal to 1 at $t = 0$ (where $V_\bullet(0)$ contains only the source v_0), is equal to 2 at $t = 1$, is equal to 3 at $t = 2$ and so on, up to $t = 2\tau\,(\bar{V} - 1)$ where $|V_\bullet(t)| = |V| \leq \bar{V}$.

Therefore, we can say that by $t = 2\tau\,\bar{V}$, the message has reached the destination nodes. In other words, CF terminates in less than $2\,\tau\,\bar{V}$. This completes our proof. □

Algorithm CF can also be extended to support multiple messages simultaneously. From the listing, we can see that if the simple counter k is converted into an array k_m indexed by the message, then CF terminates after all $k_m = \bar{V}$.

So far, we assume a large value of message retention time τ_m to ensure that when a node in $V_\bullet(t)$ leaves $V(t)$, it returns into $V_\bullet(t')$ some time $t' > t$ later. However, while these nodes need to pass many messages over its lifetime, realistically, most HDN nodes only have a limited amount of storage which also stores $N(v,t)$ used to detect $c(v,t)$.

Therefore, the nodes cannot keep their messages indefinitely — a large value of τ_m. From theorem 1, if the nodes have access to a synchronized network clock, then they need to keep only messages $2\,\tau\,\bar{V}$ and younger (relative to the origination timestamps). Otherwise, messages $4\,\tau\,\bar{V}$ and younger (relative to the local *receipt* time) are needed.

There are other flooding and routing algorithms in [1] that use slightly modified assumptions from those used in CF. For example, the algorithm LISTFLOODING (LF) does not assume that $\bar{V}$ is available. Instead, it assumes that the nodes have unique identifiers used to estimate $\bar{V}$.

Since these algorithms also use CF as their main ingredient, they too can be easily extended using lemma 1 to solve the modified HDN problems. Thus, we omit this derivation.

Finally, we observe that the connectivity assumption of a HDN can be relaxed even further without affecting the reliability of CF! Assumption 2 can be restated as: $E(t)$ is such that at any time t, there is at least one connection between one node in $V_\circ(t)$ and another node in $V_\bullet(t)$.

The proof of this assertion is quite simple and sketched as follows. By requiring that at least one node in $V_\circ(t)$ is connected to a node in $V_\bullet(t)$ at all time, we can use the same argument as in lemma 1 and state that $|V_\bullet(d, t+2\tau)| \geq |V_\bullet(d,t)|$. Since the number of nodes in V is finite, eventually d will be connected to a node in $V_\bullet$. In the worst case scenario, the nodes in V_d are the last nodes to receive the message, implying an identical upper bound.

References

[1] R. O'Dell and R. Wattenhofer, "Information dissemination in highly dynamic graphs," *DIALM-POMC 2005*, Cologne, Germany 2005.

[2] H. Dubois-Ferriere *et al.*, "Age matters: efficient route discovery in mobile ad-hoc networks using encounter ages," *Proceedings of MobiHoc 2004*.

[3] E. Gafni and D. Bertsekas. "Distributed algorithms for generating loop-free routes in networks with frequently changing topology." *IEEE Trans. Comm.*, 29(1), 1981.

[4] Z.J. Haas, M.R. Pearlman, and P. Samar. "ZRP: A hybrid framework for routing in ad-hoc networks," *Ad Hoc Networking*, Addison-Wesley, 2001.

[5] D.B. Johnson *et al.*. "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks", *Ad Hoc Networking*. Addison-Wesley, 2001.

[6] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. "Geometric ad-hoc routing: Of theory and practice," *Proceedings of PODC, 2003*.

[7] A. Lindgren, A. Doria, and O. Schelen. "Probabilistic Routing in Intermittently Connected Network," *Proceedings of SAPIR 2004*, Fortaleza, Brazil, 2004.

[8] C.E. Perkins and E.M. Royer. "Ad-hoc on-demand distance vector routing," *Proceedings of WMCSA 1999.*

[9] P. Samar, *et al.*. "Independent Zone Routing: An adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM TON*, 12(4), 2004.

[10] A. Vahdat and D. Becker. "Epidemic routing for partially connected ad-hoc networks," *Technical Report CS-200006*, Duke University, April 2000.