THE INSTITUTE FOR SYSTEMS RESEARCH

ISR TECHNICAL REPORT 2009-8

Semiring Pruning for Information Dissemination in Mobile Ad Hoc Networks

Kiran K. Somasundaram, John S. Baras

Semiring Pruning for Information Dissemination in Mobile Ad Hoc Networks

Kiran K. Somasundaram, John S. Baras
University of Maryland, College Park
College Park, Maryland 20742
Email: kirans@umd.edu, baras@isr.umd.edu

Abstract

Link state routing mechanisms have shown good convergence behaviour with mobile hosts. Pruning methods such as those used in Optimized Link State Routing (OLSR) have showed good throughput behaviour when compared to traditional link-state approaches which suffered from broadcast storm problems. In this paper, we show that the pruning function is a fundamental component of link-state routing protocols for mobile networks. We also show that this component architecture aids in interpreting both distance vector and link-state approaches under a common framework. We develop a class of pruning methods for many of the most commonly used routing objectives and show that these pruning methods are a special case of the semiring distribution property.

1. Introduction

The plethora of new applications conceived for Mobile Ad-Hoc Networks (MANETs) has instigated significant research initiatives in the networking community in the recent past. In particular, routing in MANETs has been a prolific area in this regard. It is common to classify MANET routing protocols as *distance-vector* and *link-state* routing protocols. Though architecturally different, both the protocol mechanisms compute a route profile (a route might be a path or a set of paths to reach a destination set) by optimizing the same cost function. In this paper, we illustrate that this classification arises from

1) The type of information available for the minimization.
2) The role of minimizing/routing agents.

We relate these properties to the functional description of the Selector of Topology Information to Disseminate Component (STIDC) which was introduced as a part of the component architecture for MANET routing protocols ([13]). We show that the STIDC is closely related *neighbourhood computation* of the routing objective function. We develop instances of the STIDC which guarantee desired global properties for the routing agents. We show that the ability of our algorithms to preserve certain properties globally by localized pruning is a manifestation of a very general theory of semiring distribution. We trust that this generalization using semiring algebras would help develop architectures and protocols beyond the realm of routing.

This paper is organised as follows. In section 1, we introduce the mathematical notations. In sections 3 and 4, we introduce the notion of *neighbourhood computation*. In section 5, we briefly summarize the component architecture of link-state routing protocols. In section 6, we detail the functioning of the STIDC and present different realizations for STIDC. Finally in section 7, we generalize these realizations under an ordered semiring algebra.

2. Mathematical Notation

It is common in the field of mobile communications to model networks using dynamic graph models. Let $G(V, E[t]), \forall t \geq 0$ denote a dynamic graph where the vertex set V represents the mobile stations. $E[t]$ is the dynamic edge relation between a pair of stations. Stations $i, j \in V$ are adjacent at time t *iff* $(i, j) \in E[t]$. In this paper we consider only undirected links. The graph adjacencies are typically established using neighbour discovery mechanisms described in [3]. We denote the *one-hop* neighbourhood boundary of station i by $\mathbb{N}_1^b(i)[t]$. This corresponds to the nodes which have a direct adjacency to i. The nodes which share an adjacency with the nodes in $\mathbb{N}_1^b(i)[t]$ but not with i, form the *two-hop* neighbourhood boundary, $\mathbb{N}_2^b(i)[t]$. Similarly, the *r-hop* neighbourhood boundary, $\mathbb{N}_r^b(i)[t]$ is the set of nodes that share an adjacency wth $\mathbb{N}_{r-1}^b(i)[t]$ but not with $\mathbb{N}_j^b(i)[t], j < r - 1$ and

i. The neighbour discovery mechanism at station $i \in V$ typically makes visible, the *k-hop* neighbourhood $\mathcal{N}_k(i)[t] = \{i\} \cup_{j \leq k} \mathbb{N}_j^b(i)[t]$ along with their link metric weights at time t. This is illustrated in the forthcoming sections. We denote the network diameter at time t by $D_n[t]$.

3. Link State Routing Protocols

Link state routing protocols have a significant importance in the history of routing in data networks. A particularly notable significance of link-state mechanism was in the stabilization of ARPANET routing protocol. The original routing protocol proposed for wired ARPANET was an ambitious adaptive routing scheme [5]. This ARPANET routing was based on shortest path routing. At every router $i \in V$ the length of the link to router $j \in V$ was chosen to be the delay $d(i, j)$ seen at the interface to j. The shortest path delay computations were based on *Bellman-Ford* equation [5]. Every station computed the shortest path to reach a destination set by message passing algorithms with their neighbours $\mathcal{N}_1(i)$. The same mechanism is employed in IP distance-vector protocols such as RIP [7]. However this routing mechanism was not able to cope up the delay dynamics at the interface queues and the routing paths exhibited oscillations. After a decade of modifications and improvements, a new routing paradigm for the ARPANET was introduced in [9]. The modified new routing protocol is similar to modern link-state protocols such as OSPF ([11]). The protocol involved local delay averaging for every 10 seconds and network-wide broadcasting of the delay states every 60 seconds [5]. In this case, since the link state (delay) information is available at every router $i \in V$, the routers can locally compute the Bellman-Ford equations. This modification showed better stability properties for delay-aware routing in ARPANET [10]. This property of link-state routing protocols made them attractive for dynamic networks. This encompasses the mobile networks too . For instance, preliminary studies by *Johnson* [4] show that link-state routing protocols exhibit better convergence properties for networks with mobile hosts. This is due to the ability of link-state algorithms to process local information. This is elaborated in the forthcoming section.

4. Network Neighbourhood Computation

In the context of routing, every station can be considered as a routing agent attempting to minimize a global cost function. In the case of distance vector mechanisms the agents perform a local minimization and exchange this *processed information*. On the contrary, in pure link state mechanism the nodes broadcast *raw* (unprocessed information). This raw information creates a complete *global view* of the network information for each routing agent to autonomously minimize the global cost. We observe that there are two fundamental network operations involved in these mechanisms :

1) *Neighbourhood processing* - Processing the local raw information to prune the search space for the global minimization.
2) *Network broadcasting* - Broadcasting the processed information to all the routing agents to perform the global optimization.

This is illustrated in figure 1. While pure link state mechanisms have no neighbourhood processing, the distance-vector methods perform only neighbourhood processing and no broadcasting. This suggests that it is meaningful to classify routing protocols based on the neighbourhood over which the network processing is carried out. This classification is illustrated in figure 2. It shows that pure link-state algorithms which perform no neighbourhood computation (*network processing*) broadcast a lot of information. The pure distance vector approach performs D_n (network diameter) wide network processing and hence does not broadcast any raw information. The figure also shows algorithms such as OLSR ([17], [14]) which have access to a local view of $\mathcal{N}_2(i), \forall i \in V$ perform local network processing to reduce the broadcast information. In a similar manner, mechanisms which have access to $\mathcal{N}_k(i), \forall i \in V$ can significantly reduce the broadcast information at the cost of local network processing. In the coming sections we visit the component architecture proposed in [13] and relate the neighbourhood computation to the functional description of the STIDC.

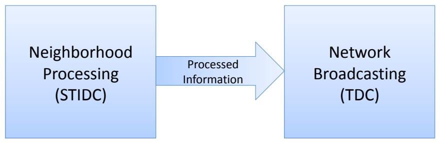


Figure 1: Fundamental components for information dissemination in routing protocols.

5. Component Architecture for Link-State Routing algorithms

The fundamental idea of our previous work was to identify and partition the primary functionalities of

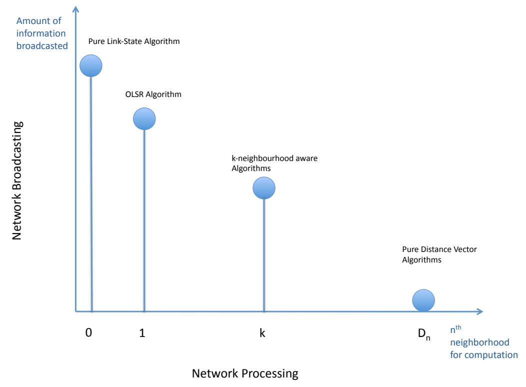


Figure 2: Relative contribution of the fundamental components.

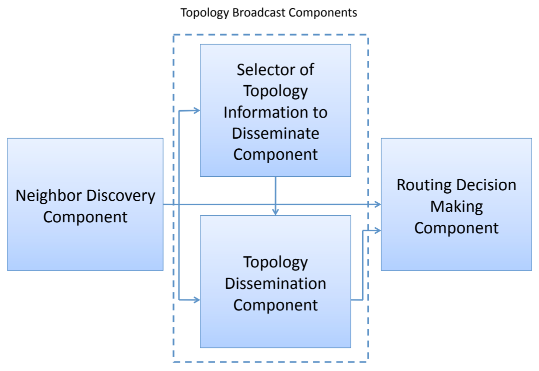

Topology Broadcast Components
Selector of Topology Information to Disseminate Component
Neighbor Discovery Component
Routing Decision Making Component
Topology Dissemination Component

Figure 3: Components of Link-State Routing Protocol

routing protocols into components. To illustrate the idea let us consider the different functionalities of the OLSR protocol. OLSR neighbour discovery mechanism enables every station to be aware of $\mathcal{N}_2(i), \forall i \in V$. This is captured by the Neighbour Discovery Component (NDC). OLSR's MPR selection, which is based on a local vertex covering problem [2] serves two purposes: choosing the subset of topology information that must be broadcasted; nominating the stations to relay this information. These two functionalities can be logically partitioned into into *Selector of Topology Information to Disseminate Component* (STIDC) and *Topology Dissemination Component* (TDC) respectively. These components feed into the *Route Selection Component*(RSC) which builds the routing/forwarding tables. These components are shown in figure 3. In this paper, we define the functional requirements of the STIDC and provide a design mechanism to meet these requirements.

6. Selector of Topology Information to Disseminate Component

As the name signifies, the STIDC is responsible for selecting the information that creates a global view for every routing agent. This information could represent coarse details such as link's ON-OFF state or more precise details such as the interface delay. For instance, in ARPANET the STIDC chooses the average link delays in the network. In OSPF the cost (also called metric) of an interface is an indication of the overhead required to send packets across a certain interface (i.e. the cost of an interface is inversely proportional to the bandwidth of that interface [1]). The NDC mechanism for both APRNET and OSPF expose only $\mathcal{N}_1(i), \forall i \in V$. This limited information limits the STIDC to a naive functionality of selecting all information exposed by the NDC. In other words, the STIDC does not have sufficient information to perform local pruning and hence there is significant flooding of raw information (figure 2). On the contrary, the NDC for OLSR (and many other MANET protocols) exposes $\mathcal{N}_2(i)$. This enables their STIDC to prune the topology information that will be broadcasted by TDC. However the limitation with the OLSR protocol is that the STIDC and TDC are fused into a single component. The local vertex covering guarantees network-wide broadcasting [14] and indeed satisfies the requirements of TDC. However it does not serve well for the functionality of the STIDC. OLSR's STIDC guarantees that the shortest path in terms of hop count is preserved in the global view. However for mobile networks it is very natural to associate a cost to the wireless link, based on its stability, capacity or metrics of reliability. In these cases, an OLSR-like STIDC based on the covering condition is handicapped to meet the functional requirements.

We observe that the STIDC has great practical significance especially for mobile routing protocols. This is because the STIDC serves as the interface between the local and global views of the dynamic graph. While NDC quickly exposes a dynamic $\mathcal{N}_k(i)[t], \forall i \in V$(local graph) to the STIDC, the later has the responsibility of cleverly choosing information to create a global view. Ideally one would expect to flood all the local information. But this results in significant overhead which consumes the already limited bandwidth of a wireless medium. Instead if the STIDC can summarize the sufficient information for routing, it can help improve the throughput.

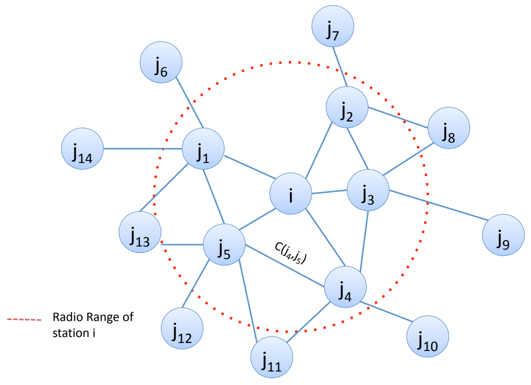


Figure 4: Local view at station i

In [12] *Wu et al.* introduce the concept of local view at every station. This captures the time-varying adjacency at local neighbourhood at every station $i \in V$. We extend this notion to also capture the various metric weights.

Definition The $LocalView_i[t], \forall t \geq 0$ at station $i \in V$ is the subgraph of the dynamic graph $G[t]$ along with the metric weights $c(i,j)[t], \forall (i,j)$ in the subgraph that is exposed at i by its NDC.

For instance, figure 4 shows the local view at station i for a NDC mechanism that exposes $\mathcal{N}_2(i) \forall i \in V$. It is assumed that in the neighbour discovery phase, along with the station identifiers, the interface cost $c(i,j)[t]$ is also exchanged. In this paper, we assume that the $LocalView_i[t]$ is composed of $\mathcal{N}_k(i)[t]$ along with the metric weights for each link in it. Given this $LocalView_i[t]$, the *functional requirement* of the STIDC is *to summarize $LocalView_i[t], \forall t \geq 0$, such that the union of this processed information at every $i \in V$ along with the information exposed by the NDC is sufficient for every routing agent to perform a global minimization to compute the optimal route profile.*

Definition The $GlobalView_i[t], \forall t \geq 0$ at station $i \in V$ is the subgraph of the dynamic graph $G[t]$ along with the metric weights $c(i,j)[t], \forall (i,j)$ in this subgraph that is made available at station i by the $LocalView_i[t]$ created by the NDC and the broadcasted STIDC information.

6.1. STIDC algorithms

In the forthcoming subsection we introduce STIDC instances that satisfy the functional description for a set of commonly used routing objective functions. In all the algorithms that follow, every station $i \in V$ runs the STIDC pruning algorithm to summarize its $LocalView_i[t]$. For each station, the algorithms return a subset of the links incident to that station along with their link costs ($\mathcal{L}_i$). This information is fed into the TDC to be broadcasted across the network. The broadcasted subgraph corresponds to $G_{broadcast}[t] = \cup_{i \in V} \mathcal{L}_i[t]$. Then the corresponding global view at every station i is $GlobalView_i[t] = LocalView_i[t] \cup G_{broadcast}[t]$. We show that for a very general class of routing objectives, this $GlobalView_i[t]$ contains sufficient information for the routing agents to build their routing tables.

6.2. Preserving Shortest Path

One of the most common routing metric is the shortest path metric. The length of any path p at time t is given by

$$l(p)[t] = \sum_{(i,j) \in p} c(i,j)[t]$$

Then the shortest pair between an source-destination pair (S,T) is

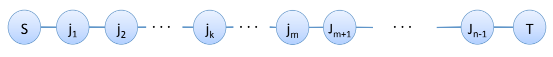
$$p^{SP}(S,T)[t] = arg \min_{p \in \mathcal{P}_{S,T}[t]} l(p)[t]$$

where $\mathcal{P}_{S,T}[t]$ is the set of all paths from S to T. The functionality of the STIDC is to preserve the shortest paths in $GlobalView_i[t], \forall i \in V$. Running algorithm 1 at every station i creates a $GlobalView_i[t], \forall i \in V$ in manner described in subsection 6.1.

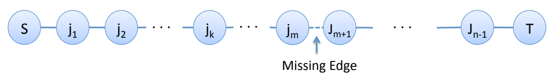
Algorithm 1 Pruning algorithm for shortest path at station i

```
INPUT: $LocalView_i[t]$
$\mathcal{L}_i[t] \leftarrow \emptyset$
$Tree_{SP}(i) \leftarrow$ Shortest Path Tree Rooted at $i$ for $LocalView_i[t]$
for all $j \in \mathbb{N}^1(i)[t]$ do
    if $(i,j) \in Tree_{SP}(i)$ then
        $\mathcal{L}_i[t] = \mathcal{L}_i[t] \cup \{(i,j,c(i,j)[t])\}$
    end if
end for
return $\mathcal{L}_i[t]$
```

Theorem 6.1: At every station $i \in V$, the $GlobalView_i[t]$ generated by algorithm 1 preserves all pair source-target shortest paths in $G[t]$.



(a) Original Path



(b) Broken Path

Figure 5: Path $(S \rightarrow T)$

Proof: Let us consider any source-target pair (S, T). Let the shortest path from S to T be $p^{SP}(S,T)[t] = S \rightarrow j_1 \rightarrow j_2 \rightarrow \cdots \rightarrow j_{n-1} \rightarrow T$. This is shown in figure 5a. Let us suppose that the this shortest path is not preserved in $GlobalView_S[t]$. Let us consider the intersection of $p^{SP}(S,T)[t]$ and $GlobalView_S[t]$ shown in the figure 5b. Since the shortest path is not preserved, this corresponds to a broken path. Let us choose one missing link (j_m, j_{m+1}).
$\Rightarrow$ Edge (j_m, j_{m+1}) is not a part of the shortest path from j_m to j_{m+1}. (By algorithm 1)
$\Rightarrow \exists$ a shortest path $(j_m \rightarrow j_l \rightarrow \cdots \rightarrow j_{m+1})$, where $j_l \neq j_{m+1}$ in the $LocalView_{j_m}[t]$.
Let us denote $p^R(S,T)$ be a path obtained by replacing the edge (j_m, j_{m+1}) in $p^{SP}(S,T)[t]$ with this lesser cost sub-path. Then cost $l(p^{SP}(S,T)[t]) > l(p^R(S,T)[t])$. This a contradiction. So edge (j_m, j_{m+1}) is indeed preserved. We can extend the proof to every missing edge to prove that $p^{SP}(S,T)[t]$ is preserved in $GlobalView_i[t]$. □

6.3. Preserving max-min paths

Another routing metric is the bottleneck metric. It is typically used to route traffic through the maximum capacity path. For any path p the bottleneck metric is given by

$$b(p)[t] = \min_{(i,j) \in p} c(i,j)[t]$$

Then the *max-min* path between the source-target pair (S, T) is given by

$$p^{MM}(S,T)[t] = arg \max_{p \in \mathcal{P}_{S,T}[t]} b(p)[t]$$

Algorithm 2 runs at every station i and creates a $GlobalView_i[t]$ at i.

Theorem 6.2: At every station $i \in V$, the $GlobalView_i[t]$ generated by algorithm 2 preserves all pair source-target max-min paths in $G[t]$.

Algorithm 2 Pruning algorithm for max-min path at station i

```
INPUT: $LocalView_i[t]$
$\mathcal{L}_i[t] \leftarrow \emptyset$
$Tree_{MM}(i) \leftarrow$ Max-Min Tree rooted at $i$
for $LocalView_i[t]$
for all $j \in \mathbb{N}^1(i)[t]$ do
  if $(i,j) \in Tree_{MM}(i)$ then
    $\mathcal{L}_i[t] = \mathcal{L}_i[t] \cup \{(i,j)[t]\}$
  end if
end for
return $\mathcal{L}_i[t]$
```

Proof: Let us consider any source-target pair (S, T). Let the max-min path from S to T be $p^{MM}(S,T)[t] = S \rightarrow j_1 \rightarrow j_2 \rightarrow \cdots \rightarrow j_{n-1} \rightarrow T$. This is shown in figure 5a. Let us suppose that the this max-min path is not preserved in $GlobalView_S[t]$. Let us consider the intersection of $p^{MM}(S,T)[t]$ and $GlobalView_S[t]$ shown in the figure 5b. Let us choose one missing link (j_m, j_{m+1}).
$\Rightarrow$ Edge (j_m, j_{m+1}) is not a part of the max-min path from j_m to j_{m+1}. (By algorithm 2)
$\Rightarrow \exists$ a max-min path $(j_m \rightarrow j_l \rightarrow \cdots \rightarrow j_{m+1})$, where $j_l \neq j_{m+1}$ in the $LocalView_{j_m}[t]$.
Let us denote $p^R(S,T)[t]$ be a path obtained by replacing the edge (j_m, j_{m+1}) in $p^{MM}(S,T)[t]$ with better max-min sub-path. Then the bottleneck metric $b(p^{MM}(S,T)[t]) < b(p^R(S,T)[t])$. This a contradiction. So edge (j_m, j_{m+1}) is indeed preserved. We can extend the proof to every missing edge to prove that $p^{MM}(S,T)[t]$ is preserved in $GlobalView_i[t]$. □

6.4. Preserving K-shortest path

Another routing objective is the K-shortest paths used for reliability, security and load-balancing. For any source-target pair (S, T) the K-shortest paths are the first K paths of the set $\mathcal{P}_{S,T}[t]$ ranked in increasing path lengths. Again the functionality of the STIDC is to preserve these paths for every source-target pair. The STIDC runs algorithm 3 to prune for this set of paths.

Theorem 6.3: At every station $i \in V$, the $GlobalView_i[t]$ generated by algorithm 3 preserves all *K-shortest* path sets in $G[t]$.

Proof: Let us consider any source-target pair (S, T). Let the K-shortest path set be $\mathcal{P}^{KSP}_{S,T}[t]$. Let us suppose this set of paths is not preserved in $GlobalView_S[t]$. Let us consider the intersection of $\mathcal{P}^{KSP}_{S,T}[t]$ with the $GlobalView_i[t]$. This creates a broken set of paths shown in figure 6. Let us consider a missing link (j_m, j_{m+1}).

Algorithm 3 Pruning algorithm for K-shortest paths at station i

```
INPUT: $LocalView_i[t]$
$\mathcal{L}_i[t] \leftarrow \emptyset$
$Tree_{KSP}(i)$ ← K-Shortest Path Tree Rooted at $i$ for $LocalView_i[t]$
for all $j \in \mathbb{N}^1(i)[t]$ do
  if $(i,j) \in Tree_{KSP}(i)$ then
    $\mathcal{L}_i[t] = \mathcal{L} \cup \{(i,j)c(i,j)[t]\}$
  end if
end for
return $\mathcal{L}_i[t]$
```

$\Rightarrow$ Edge (j_m, j_{m+1}) is not a part of the *K-Shortest path* set from j_m to j_{m+1}. (By algorithm 3)
$\Rightarrow$ $\exists$ a path set $\mathcal{P}^{KSP}_{j_m,j_{m+1}}[t]$ in the $LocalView_i[t]$ such that none of the paths in the set use the edge (j_m, j_{m+1}).
Again we have a better replacement path set between the (S,T) pair using the path set $\mathcal{P}^{KSP}_{S,T}[t]$. We can extend the proof to every missing edge to prove that $\mathcal{P}^{KSP}_{S,T}[t]$ is indeed preserved in $GlobalView_i[t]$. □

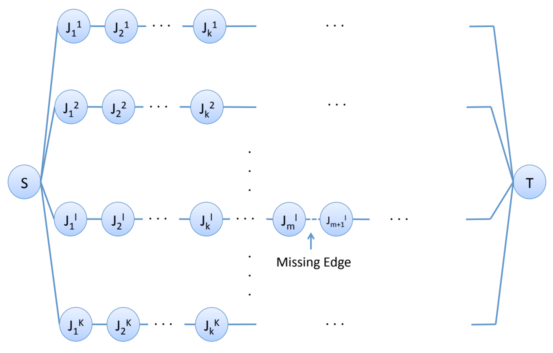


Figure 6: Broken Path Set between (S,T)

7. Generalized Semiring Pruning Methods

The pruning methods introduced in the previous sections suggest that there is an underlying algebra to these pruning methods. The algorithms suggest that by preserving a *property* in the local neighbourhood, we are able to preserve the *property* globally. We show that this algebra is a semiring algebra. For a detailed survey of applications of semirings we refer the reader to [8], [15], [6] and [16].

A semiring is an algebraic structure $(S, \oplus, \otimes)$ which satisfies the following axioms:
(A1) (S,$\oplus$) is a commutative semigroup with a neutral element ⓪

$$\begin{aligned} a \oplus b &= b \oplus a \\ a \oplus (b \oplus c) &= (a \oplus b) \oplus c \\ a \oplus ⓪ &= a \end{aligned}$$

(A2) (S,$\otimes$) is a semigroup with a neutral element ① *and* ⓪ *as an absorbing element*

$$\begin{aligned} a \otimes (b \otimes c) &= (a \otimes b) \otimes c \\ a \otimes ① &= a \\ a \otimes ⓪ &= ⓪ \end{aligned}$$

(A3) $\otimes$ distributes over $\oplus$

$$\begin{aligned} a \otimes (b \oplus c) &= (a \otimes b) \oplus (a \otimes c) \\ (a \oplus b) \otimes c &= (a \otimes c) \oplus (b \otimes c) \end{aligned}$$

It should be noted that the functions which have this semiring structure lend themselves to distributed computation/evaluation by the virtue of the distributivity property (A3). This property of semiring structures have been used in many path problems in graphs [8]. One particularly useful semiring for optimization is the *Ordered Semiring*. Here the $\oplus$ is the supremum or infimum operator and $(S, \otimes, \preceq)$ is an ordered semigroup. An ordered semigroup is a semigroup with an order relation which is monotone with respect to $\otimes$. i.e. $a, b, a', b' \in S$ we have

$$a \preceq b \quad and \quad a' \preceq b' \quad \Rightarrow \quad a \otimes a' \preceq b \otimes b'$$

In this paper we consider only ordered semirings. Without loss of generality, we assume that the $\oplus$ operator is the infimum operator. In the context of mobile networks with metrics on the links, we associate with every edge (i,j) of the dynamic graph a semiring element $c(i,j)[t] \in S$.

Definition A general semiring path problem on a dynamic graph corresponds to computing

$$p^*(S,T)[t] = arg \oplus_{p \in \mathcal{P}_{S,T}[t]} \otimes_{(i,j) \in p} c(i,j)[t] \qquad (1)$$

The equivalence of this definition with shortest path, max-min path and k-shortest paths problem is well illustrated in [8]. Let us consider an abstract pruning algorithm at every $i \in V$ in algorithm 4. The procedure `Semiring Pruned Tree Rooted at` i computes the optimal paths from i to $j \in LocalView_i[t]$ based on Equation 1. The algorithm creates a $GlobalView_i[t] \forall i \in V$ by the procedure illustarted in subsection 6.1.

Algorithm 4 Semiring pruning algorithm i

```
INPUT: $LocalView_i[t]$
$\mathcal{L}_i[t] \leftarrow \emptyset$
$Tree_{Semiring}(i) \leftarrow$ Semiring Pruned Tree Rooted at $i$ for $LocalView_i[t]$
for all $j \in \mathbb{N}^1(i)[t]$ do
    if $(i, j) \in Tree_{Semiring}(i)$ then
        $\mathcal{L}_i[t] = \mathcal{L}_i[t] \cup \{(i, j, c(i, j)[t])\}$
    end if
end for
return $\mathcal{L}_i[t]$
```

Theorem 7.1: At every station $i \in V$, the $GlobalView_i[t]$ generated by algorithm 4 preserves all pair optimal paths (optimality with respect to Equation 1) in $G[t]$.

Proof: Let us consider any source-target pair (S, T). Let the optimal path from S to T be $p^*(S, T)[t] = S \rightarrow j_1 \rightarrow j_2 \rightarrow \cdots \rightarrow j_{n-1} \rightarrow T$. Let us suppose that this optimal path is not preserved in $GlobalView_S[t]$. Let us consider the intersection of $p^*(S, T)$ and $GlobalView_S[t]$.Let us choose one missing link (j_m, j_{m+1}).
$\Rightarrow$ Edge (j_m, j_{m+1}) is not a part of the optimal path from j_m to j_{m+1}. (By algorithm 4)
$\Rightarrow \exists$ a optimal path $(j_m \rightarrow j_l \rightarrow \ldots j_{l'} \rightarrow j_{m+1})$, where $j_l \neq j_{m+1}$ in the $LocalView_i[t]$.
$\Rightarrow c(j_m, j_{m+1}) \succ c(j_m, j_l) \otimes \cdots \otimes c(j_{l'}, j_{m+1})$

$$\begin{aligned} Cost(p^*(S,T)[t]) &= \otimes_{(i,j)\in p^*(S,T)[t]} c(i,j)[t] \\ &= c(S,j_1)[t] \otimes c(j_1,j_2) \otimes \ldots \\ &\quad \otimes c(j_m,j_{m+1}) \otimes \ldots c(j_{n-1},T) \\ &\succ c(S,j_1)[t] \otimes c(j_1,j_2) \otimes \cdots \otimes \\ &\quad c(j_m,j_l) \otimes \cdots \otimes c(j_{l'},j_{m+1}) \otimes \\ &\quad \cdots \otimes c(j_{n-1},T) \textit{ (By A3)}. \end{aligned}$$

This means there is a better path from S to T. This is a contradiction. Edge (j_m, j_{m+1}) is indeed preserved. We can extend the proof to every missing edge to prove that $p^*(S, T)$ is preserved in $GlobalView_i[t]$. □

We mention that this generalization using semiring distribution is not necessarily limited to routing objectives. The same architectural abstractions can be to extended other applications such has sensor fusion, estimation and tracking (many of these algorithms are message passing algorithms which can be abstracted as semirings).

8. Conclusion

In this paper we define the functional requirements of STIDC. We detail the importance of the STIDC pruning for routing in MANETs. We then present instances of the STIDC which aid the routing agents to correctly configure their routing tables. We show that these instances can preserve important properties such as shortest paths, min-max paths and K-shortest paths by local pruning. We also generalize this property and show it as a special case of the semiring distribution property.

References

[1] Ospf design guide. www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094e9e.shtml. Last accessed 05/30/2009.

[2] Qayyum A., Viennot L., and Laouiti A. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, volume 9, page 298, 2002.

[3] Ben Khedher D., Glitho R., and Dssouli R. A novel overlay-based failure detection architecture for manet applications. In *IEEE International Conference on Networks*, pages 130–135, 2007.

[4] Johnson D.B. Routing in ad hoc networks of mobile hosts. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 158–163, 1994.

[5] Bertsekas D.P. and Gallager R. *Data Networks*. Prentice Hall, 1992.

[6] Kschischang F.R., Frey B.J., and Loeliger H. Factor graphs and sum-product algorithm. *IEEE Transactions on Information Theory*, 46:489–519, 2001.

[7] Malkin G. Rip version 2. RFC 2453, November 1998. Network Working Group.

[8] Rote G. Path problems in graphs, 1989.

[9] McQuillan J., Richer I., and Rosen E. The new routing algorithm for the arpanet. *IEEE Transactions of Communication*, 28:711– 719, 1980.

[10] McQuillan J., Richer I., and Rosen E. An overview of the new routing algorithm for the arpanet. In *ACM SIGCOMM Computer Communication Review*, volume 25, pages 54–60, 1995.

[11] Moy J. Ospf version 2. RFC 2328.

[12] Wu J. and Dai F. A generic distirbuted broadcast scheme in ad hoc wireless networks. *IEEE Transactions of Computers*, 53(10):1343–1354, 2004.

[13] Baras J.S., Tabatabaee V., Purkayastha P., and Somasundaram K. Component based performance modeling of the wireless routing protocols. In *IEEE ICC Ad Hoc and Sensor Networking Symposium*, June 2009.

[14] Jacquet P., Laouiti A., Minet P., and Viennot L. *Performance of multipoint relaying in ad hoc mobile routing protocols*. Springer, February 2004.

[15] McEliece R.J. and Aji S.M. The generalized distributive law. *IEEE Transactions on Information Theory*, 46(2):325–343, 2000.

[16] Verdu S. and Poor V. Abstract dynamic programming models under commutativiy conditions. *SIAM Journal on Control and Optimization*, 25(4):990–1006, 1987.

[17] Clausen T. and Jacquet P. Optimized link state routing protocol (olsr). RFC, Oct 2003.