Route Recovery in Vertex-Disjoint Multipath Routing for Many-To-One Sensor Networks

Cheng et. al

MobiHoc08

04.17.09

Outline

1. Introduction
2. Path Existence
   - Flip Algorithm
   - Omvdp Algorithm
3. Route Recovery
4. Bibliography

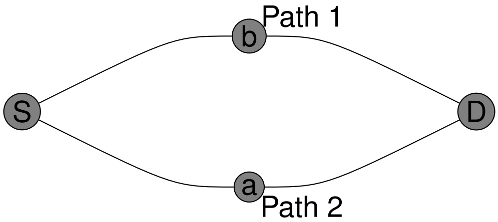
Outline

1. Introduction
2. Path Existence
   - Flip Algorithm
   - Omvdp Algorithm
3. Route Recovery
4. Bibliography

Definitions

Vertex Disjoint

Two paths are point/node/vertex disjoint if they have common endpoints but no common midpoints.



Path 1 and Path 2 are vertex disjoint

Relevance

Multi-Path routing

A routing scheme in which a given source has two or more paths to a destination is a *Multi-path* scheme. If those paths are strictly vertex disjoint, there are benefits in load balancing, QoS, and redundancy.

Prior Work

The authors maintain that prior work does not consider their specific challenges. In particular, the authors present a novel algorithm that is guaranteed to find a Vertex-Disjoint Path if one exists.

Research Questions

1. When one of a set of N Vertex-Disjoint Paths is broken due to a node failure or linkage failure, is it possible to create a new set of N Vertex-Disjoint Paths?
2. If a set of N **VDPs** cannot be created, is it possible to determine the correct position for some nodes to take in order to create the set?
3. What information is needed to compute a set of N **VDPs**, and how can this be accomplished using only local information?

Goal

The authors goal is to implement a distributed algorithm that allows a network to maintain a minimum number of vertex disjoint paths from any source to a specific destination.

Claims

The authors make three basic claims.

1. They identify the conditions in which a vertex disjoint path exists
2. They provide a maintenance framework for cases in which a minimum number of **VDPs** are required but some may be lost over time
3. They argue that this algorithm is a good complement for traditional max-flow algorithms in cases where **VDPs** are required.

Outline

1. Introduction

2. Path Existence
   - Flip Algorithm
   - Omvdp Algorithm

3. Route Recovery

4. Bibliography

Notation

S	S is the source node of the basic set
P	P is the source node of the reference set.
T	T is the destination node.
$\mathcal{C}_S$	$\mathcal{C}_S = \{S_{T_1}, S_{T_2}, \ldots, S_{T_i}, \ldots, S_{T_{N_B}}\}$, the basic set, where N_B is the number of paths and $1 \leq i \leq N_B$.
$\mathcal{C}_P$	$\mathcal{C}_P = \{P_{T_1}, P_{T_2}, \ldots, P_{T_j}, \ldots, P_{T_{N_R}}\}$, the reference set, where N_R is the number of paths and $1 \leq j \leq N_R$.
S_{T_i}	$S_{T_i} = \{s_{i_0}, s_{i_1}, \ldots, s_{i_{B_i}}\}$, the ith path in the basic set, where $s_{i_0} = S$, $s_{i_{B_i}} = T$ and B_i is the number of vertexes.
P_{T_j}	$P_T = \{p_{j_0}, p_{j_1}, \ldots, p_{j_{R_j}}\}$, the jth path in the reference set, where $p_{j_0} = P$, $p_{j_{R_j}} = T$ and R_j is the number of vertexes.

First Level Induced Path

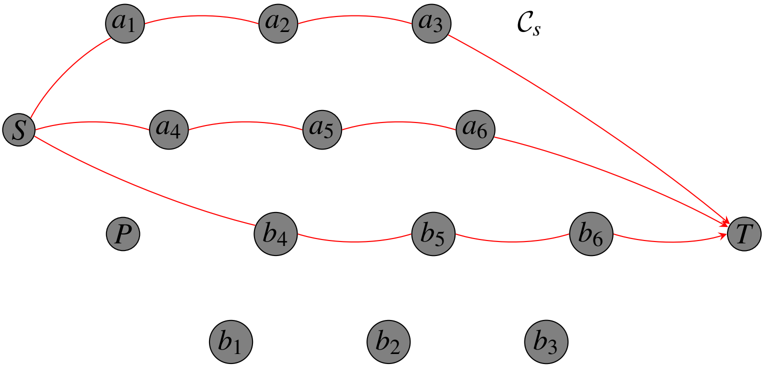
Paired Vertex Disjoint Set

Given a source node and a destination node (S, D), a set of Vertex-Disjoint Paths between them is a paired vertex disjoint path set *(My Notation)

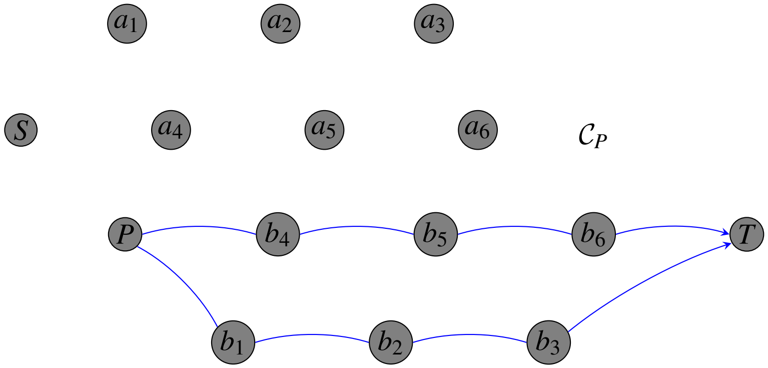
Induced Path

Given two sources (S, P) and one destination T, where one source S is designated as the basic source and one source P, as the reference source, an induced path is a path that starts at S, overlaps **exactly** one path in the basic set $\mathcal{C}_S$, overlaps *at most* one path in the reference set $\mathcal{C}_P$, and ends at T.

Example of an Induced Path

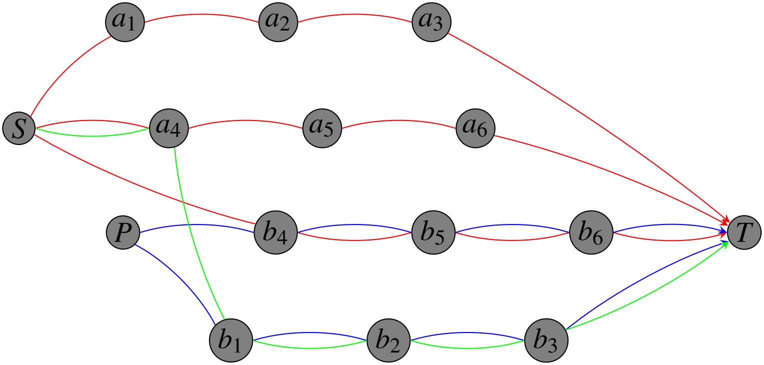


Example of an Induced Path



Example of an Induced Path

An induced path on Basic Set $\mathcal{C}_S$, and Reference Set $\mathcal{C}_P$



The Ordered Intersection Set

Definition

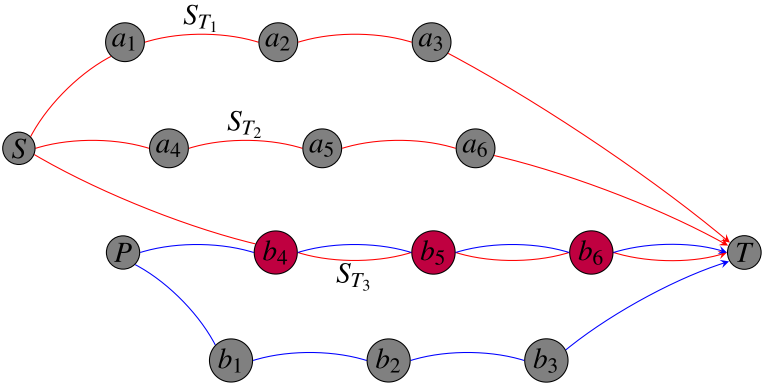
$X_i = \{S_{i_k} | \exists j, S_T.s_{i_k} \in P_{T_j}, 0 \leq k \leq B_i\}$ is the ordered intersection set of S_{T_i}. A vertex is a member of X_i if it is on a specific path S_{T_i} and a path P_{T_j} in cC_P.

Y_j is the ordered intersection set of P_{T_j}.

The set is ordered because for each s_{i_n}, S_{i_m} in X_i, $n < m$ if s_{i_n} can reach S in fewer hops than s_{i_m}

Example of an Ordered Intersection Set

The set $X_3 = \{b_4, b_5, b_6\}$



More Definitions

lth level Subpath

If a node n is the lth element of X_i, then the subpath from S to n on ST_i is the lth level subpath.

Effective Partition Node

A node n is an effective partition node of $P_{T_j} \iff n \in X_i \wedge$ the subpath n, T is vertex disjoint to all other paths S_{T_i}

Effective Residual Path

The subpath on P_{T_j} from an effective partition node to T.

Illustration of Effective Partitioning

1^{st} and 2^{nd} level subpaths of S_{T_3}

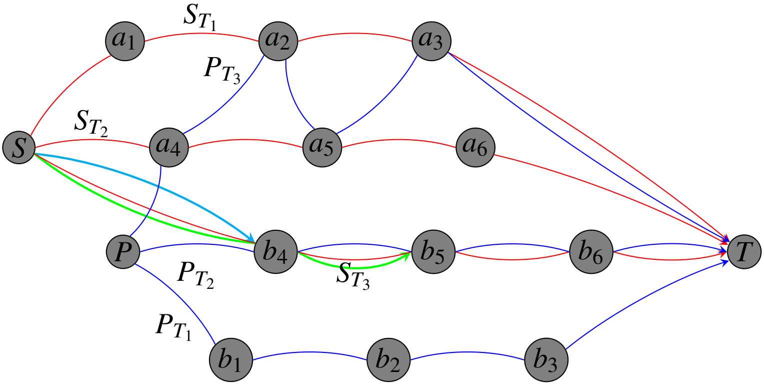


Illustration of Effective Partitioning

$$X_3 = \{b_4, b_5, b_6\}$$

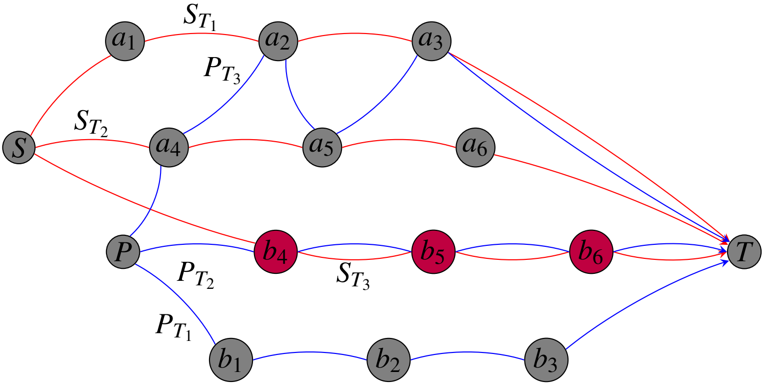


Illustration of Effective Partitioning

$$Y_2 = \{b_4, b_5, b_6\}$$

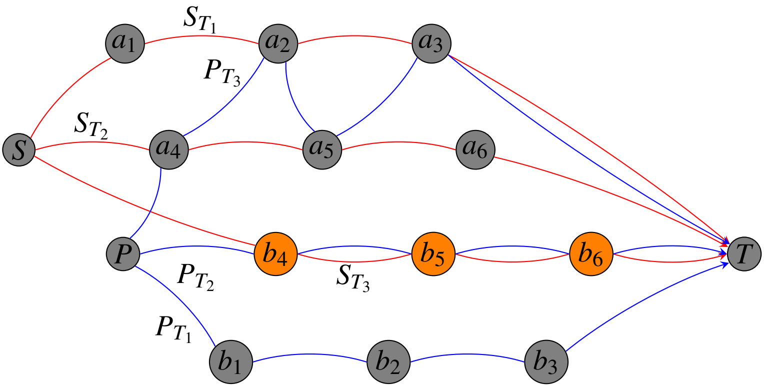


Illustration of Effective Partitioning

Effective Partition Nodes of $P_{T_2} = \{b_4, b_5, b_6\}$

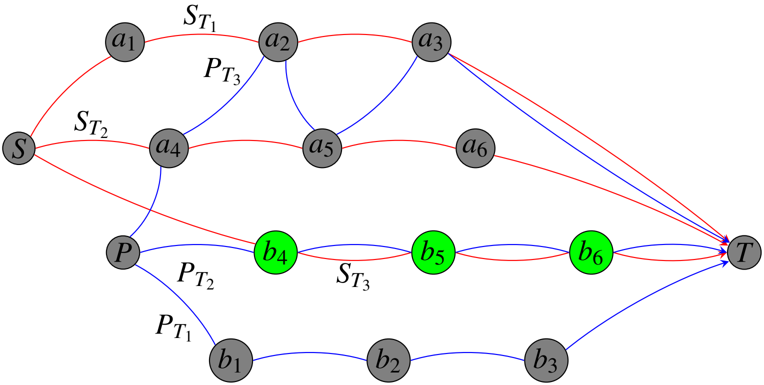


Illustration of Effective Partitioning

$$X_1 = \{a_2, a_3\}$$

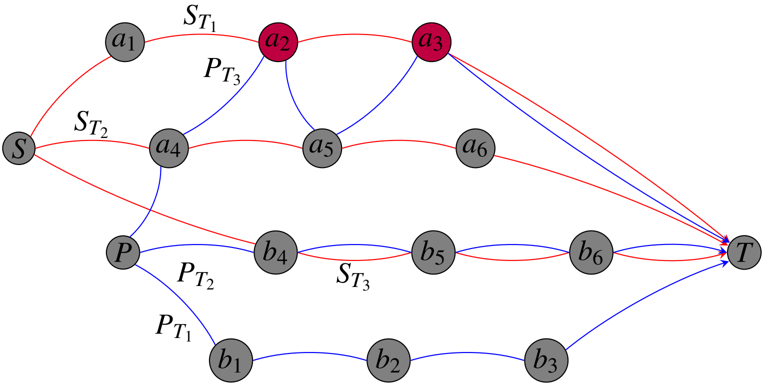


Illustration of Effective Partitioning

$$Y_3 = \{a_4, a_2, a_5, a_3\}$$

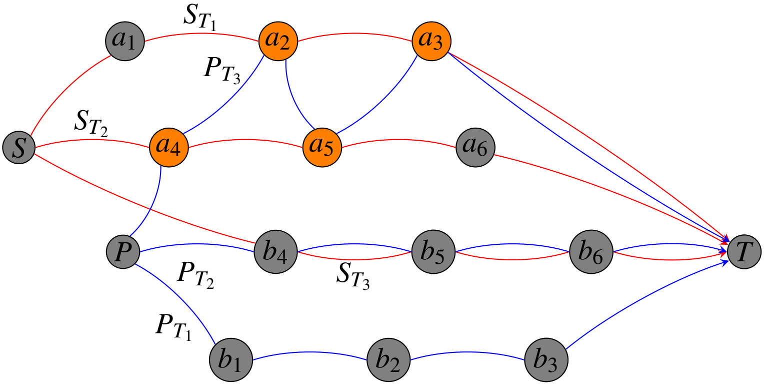


Illustration of Effective Partitioning

Effective Partition Nodes of $P_{T_3} = \{a_3\}$

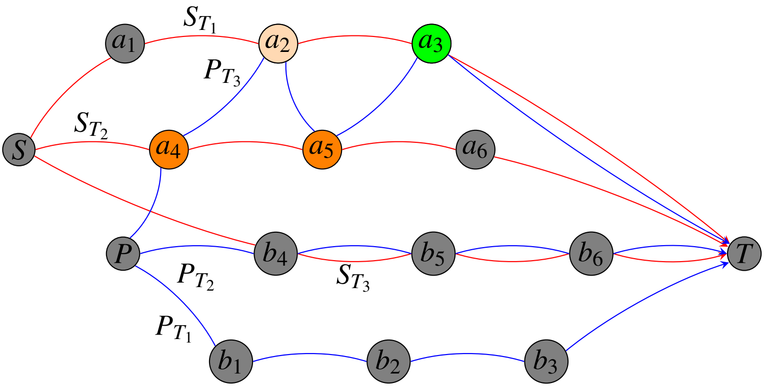
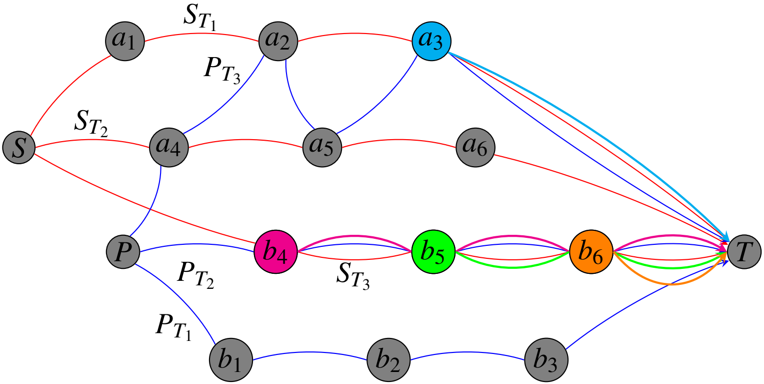


Illustration of Effective Partitioning

Effective Residual Paths



Flip Algorithm

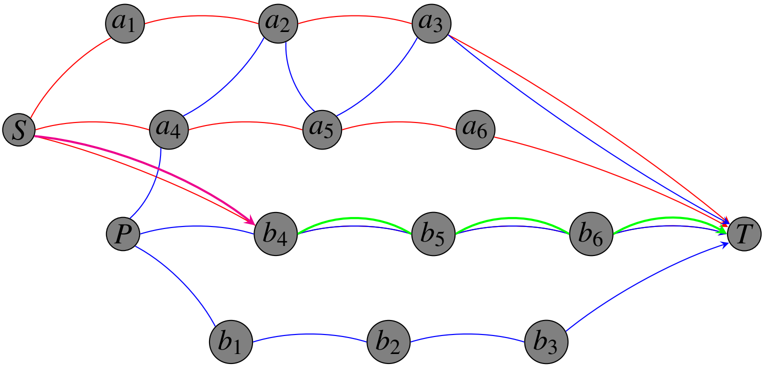
Construct the ordered intersection set for each path in C_S
if $\exists i, s.t. X_i = \emptyset$ **then**
  S_{T_i} is the induced path
  STOP
end if
Find the effective partition nodes for each path in C_P
Set $l = 1$
loop
  if $\exists i, s.t. S_{T_i}$'s lth level subpath joints an effective residual path of P_{T_j} **then**
    The induced path is the concatenation of these two subpaths
    STOP
  end if
  $l = l + 1$
end loop

Concept

- If there is a path in $\mathcal{C}_s$ that is vertex disjoint to all $\mathcal{C}_P$, output that path.
- If there is not, construct an induced path from $\mathcal{C}_s, \mathcal{C}_p$ such that the shortest lth level subpath is matched to the longest possible effective residual path.

Illustration of concatenation

An induced path the matches the 1st level subpath of S_{T_3} with the longest effective residual path of P_{T_2}



Notation and Definitions

$N - credit$

If there are N Vertex-Disjoint Pathsfrom a node n to the sink node, then n is an $N - credit$ node

A	$A = \{S_{T_1}, S_{T_2}, \ldots, S_{T_i}, \ldots, S_{T_{N-1}}\}$, the set of vertex-disjoint paths for the $(N-1)$-credit node S is the source node of the basic set
B	$B = \{P_{T_1}, P_{T_2}, \ldots, P_{T_j}, \ldots, P_{T_N}\}$, the set of vertex-disjoint paths for the N-credit node P
New_A	$New_A = NewA = \{new_1, new_2, \ldots, new_N\}$,, the set of vertex-disjoint paths constructed by Omvdp for S

Illustration of Omvdp

Inputs to the One more vertex disjoint path algorithm

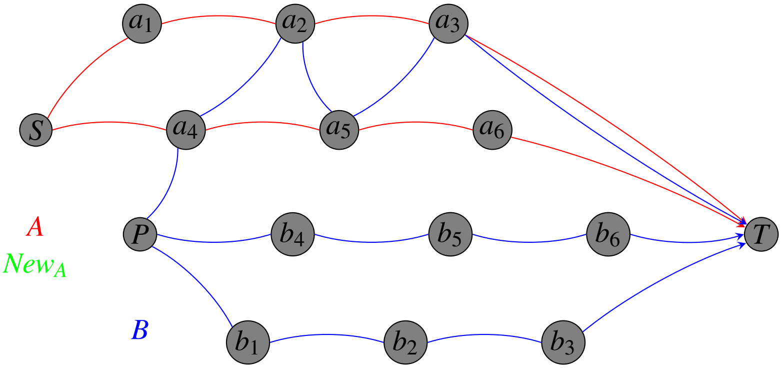


Illustration of Omvdp

Case 1: there is a path in B that is vertex disjoint to A

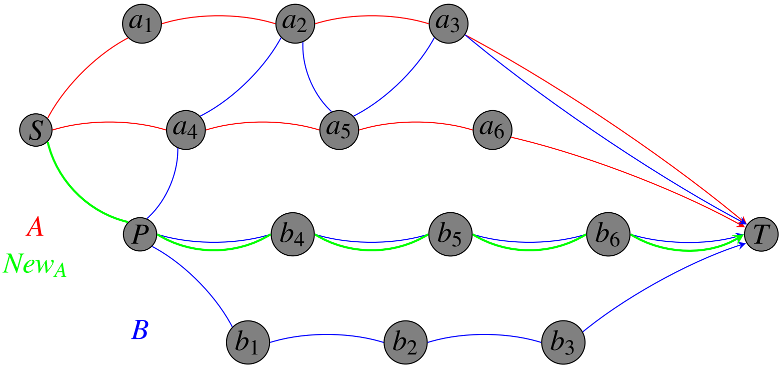


Illustration of Omvdp

Use P_{FLIP} to induce a path

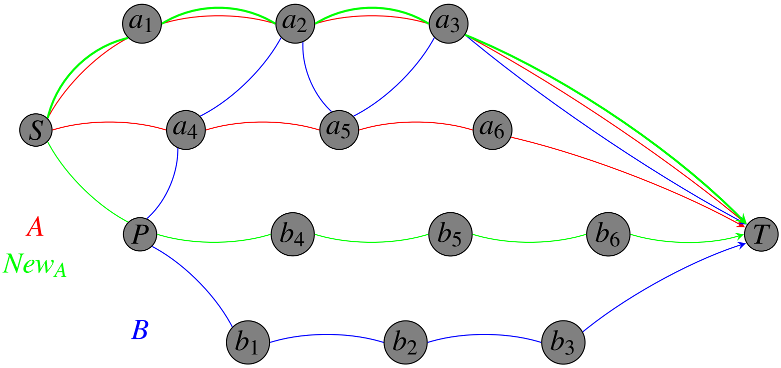


Illustration of Omvdp

A path in A is vertex disjoint to every path in B

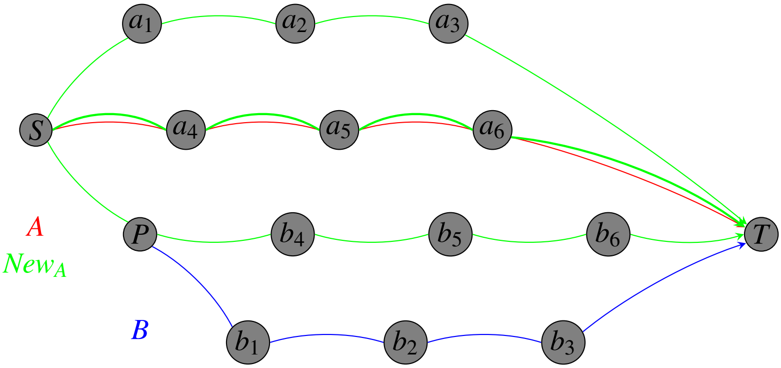
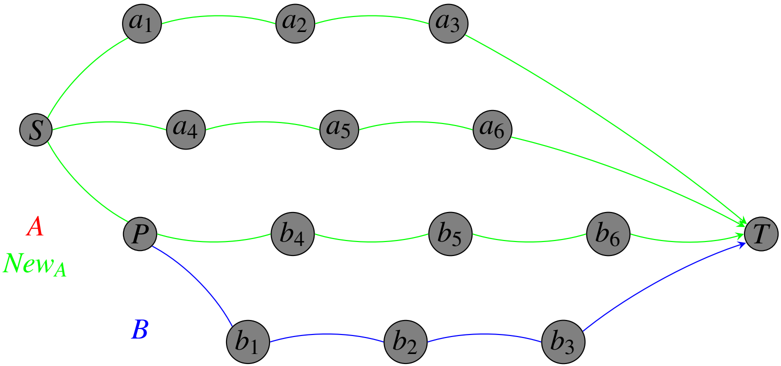


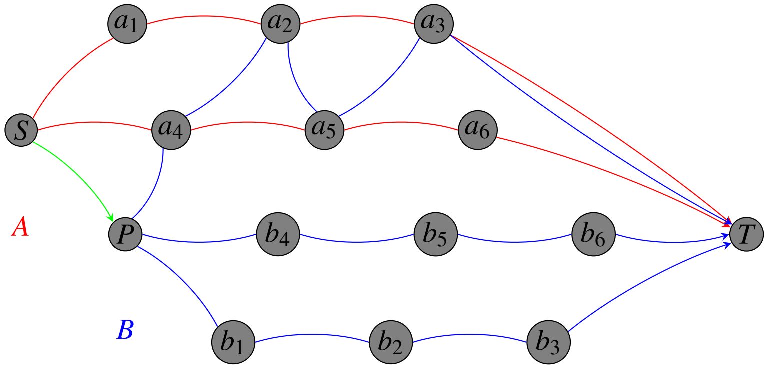
Illustration of Omvdp

S is now $N - credit$



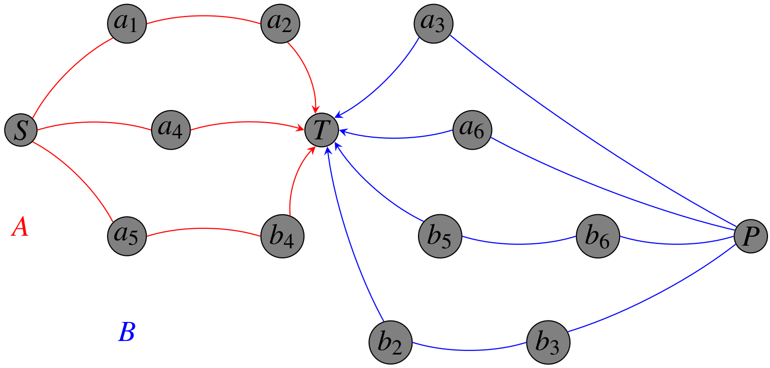
When does a new path exist?

If S_P is disjoint to A and B, S is $N - credit$



When does a new path exist?

If all paths S_p pass through T, S is not $N - credit$



Outline

1. Introduction
2. Path Existence
   - Flip Algorithm
   - Omvdp Algorithm
3. Route Recovery
4. Bibliography

Local Use

Theorem

If a node n has N neighbors that are $N+1-credit$, n is $N-credit$

Corollary

For newly deployed nodes, if there are N neighbors that are N credit, the new node is $N-credit$.

Route recovery is greatly simplified by this finding. Essentially, a newly started node can learn immediately from its neighbors whether it will be able to provide the required route.

More Local Use

Corollary

For an existing node n with some number M of existing **VDPs**, if there are $N - M$ neighbors that are $N + 1 - credit$, n is $N - credit$

These corollaries arise from the induction method proof. That is, at some point in the proof the node will have used M neighbors to construct M **VDPs**, so the corollary follows.

Discussion

- One consequence of this finding is philosophical. It demonstrates that each node gains **VDPs** when it maintains **VDPs**. This indicates that cooperative strategies are useful in sensor or ad hoc network routing.
- Vertex-Disjoint Paths are very important for QoS and stability in mobile networks, so this is a significant finding.
- Another consequence relates to my own research. The number of Vertex-Disjoint Pathsis equivalent to the minimum cut of the graph, that is, if there are N **VDPs** from all to all in the graph, the graph has a min cut of N.
- If N **VDPs** can be found in $O(N)$ time, as claimed, then for each node, ensuring a min cut for itself to the network is $O(n^2)$, which is not bad.

Outline

1. Introduction
2. Path Existence
   - Flip Algorithm
   - Omvdp Algorithm
3. Route Recovery
4. Bibliography

Bibliography

W. Cheng, K. Xing, X. Cheng, X. Lu, Z. Lu, J. Su, B. Wang, and Y. Liu.
Route recovery in vertex-disjoint multipath routing for many-to-one sensor networks.
In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 209–220, New York, NY, USA, 2008. ACM.