Complexity Analysis of Availability Models for Underlay Aware Overlay Networks

Madhu Kumar S. D
Department of Computer Science and Engineering
Indian Institute of Technology Bombay, India

Umesh Bellur
Department of Computer Science and Engineering
Indian Institute of Technology Bombay, India

Abstract

Availability of an overlay network is a necessary condition for event delivery in event based systems. The availability of the overlay links depends on the underlying physical network. Overlay networks have to be underlay aware in order to provide assured levels of availability. We propose two models of availability for overlay networks, Manifest and Latent Availability. In the Manifest availability model, distinct paths at the overlay level are also node disjoint at the underlay and hence the alternate paths viewed by the overlay are independent in the underlay also. In the latent availability model, it is only guaranteed that any two overlay nodes have a guaranteed number of node disjoint paths between them in the underlay. We analyze both the models for complexity of formation and maintenance, and prove that in the general case, both are NP-complete. Then we identify a set of practical constraints applicable to large scale networks. We demonstrate that under these constraints, latent availability constraint becomes a polynomial time problem. We also introduce the concept of reduced underlays, and further reduce the complexity of the problem of determining latent availability overlays.

1 Formalization of underlay aware overlay

An overlay network is a logical abstraction of the communication network needed by the distributed application or the event based middleware, which allows nodes running the application to communicate using a transparent broker network of event brokers. The event broker nodes, the client nodes and the logical links between these form the overlay network. The communication required is provided through a physical network of computers connected via physical communication links. The network and transport layers of the network provide the facilities for routing and transfer of information. We hereby refer to the entire physical network as the underlying network or the *underlay* of the overlay network.

The physical network is represented by a the underlay graph $G_u =< V_u, E_u >$ where V_u is a set of nodes each corresponding to a physical node in the computer network and E_u is a set of edges corresponding to each physical link existing between a pair of computers in the network.

$E_u = \{(p, q) | p, q \in V_u$, *and there exists a physical link in the computer network between nodes corresponding to* p *and* $q\}$.

Some computers in the network are broker nodes and hence a part of the overlay network. A link in the overlay network between two nodes is conceptual and implies the existence of a physical path in the computer network between the two nodes, which may contain other nodes in the computer network.

Thus an overlay network is represented as the overlay graph $G_o =< V_o, E_o >$, where V_o is the set of overlay nodes, and $E_o = \{(p, q) | p, q \in V_o$, *and there exists an overlay link in the overlay network between nodes* p *and* $q\}$.

Each overlay node has a unique underlay node associated with it. Thus,
$VERTEXMAP : V_o \rightarrow Vu$ is a function that maps every overlay node to a unique underlay node. Moreover,
$(p, q) \in E_o \Rightarrow \exists x(0 \leq x \leq |V_u|) \wedge \forall i(0 \leq i \leq x)\exists n_i(n_i \in G_u) :$

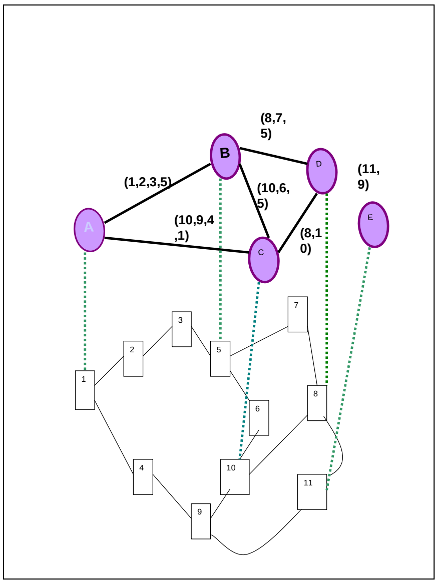
$((VERTEXMAP(p), n_0, n_1 \ldots n_x, VERTEXMAP(q))$ *is a simple path in* G_u).
Let $PATHS_u$ be the set of all simple paths in G_u.
An overlay graph, the underlay graph and $VERTEXMAP$ also define a set ($EDGEMAPS$) of functions (MAP).
A MAP is defined as a function which has E_o as its domain and $PATHS_u$ as its range and every link in E_o is mapped to a simple path in the underlay graph, between the corresponding $VERTEXMAP$s of its end nodes.
If M is a member of $EDGEMAPS$, then for any overlay link $l = (b, c)$, $M(l) = p$ where p is a simple path in G_u with end points $VERTEXMAP(b)$ and $VERTEXMAP(c)$.
For any simple path $path \in PATHS_u$, we define $INT\_NODES(path)$ to be the set of nodes in $path$ which are not the end points of $path$.
For example, if $path = (p, a, b, c, q)$, then $INT\_NODES(path) = \{a, b, c\}$.
We also define $END\_NODES(path)$ to be the set of end points of path.
In the previous example, $END\_NODES(path) = \{p, q\}$.

2 Different availability models for underlay aware overlay

The question addressed at this point is *given a physical topology and a set of broker nodes can an algorithm be found to determine whether the set of brokers can form an overlay which is fault tolerant in the face of k node/link failures ?*

2.1 Manifest Availability model

A *manifest availability overlay network of degree k* on an underlay network G_o is defined as an overlay network which satisfies two conditions
(i) *Connectivity constraint:* The overlay graph G_o has a node connectivity of k, which means every pair of nodes in G_o have k pairwise node disjoint paths between them in G_o, and
(ii) *Manifest Availability constraint:* A pair of simple paths that are node disjoint in the overlay graph G_o have corresponding node disjoint simple paths in the underlay graph G_u.
Condition (ii) would be satisfied if and only if every link in the overlay graph has a mapping to a simple path in the underlay graph which is node disjoint with the simple path that is mapped to by any other overlay link. This can be formally stated as
$\exists MAP|((MAP \in EDGEMAPS) \wedge ((\forall p \forall q(p \in E_o \wedge q \in E_o \wedge (p \neq q)) \Rightarrow (INT\_NODES(map(p)) \cap INT\_NODES(map(q)) = \Phi))$.



1

Figure 1. A Manifest Availability Overlay

This would imply that there exists a mapping from all overlay links to underlay paths such that any node n in the underlay graph belongs to at the most one underlay path in the mapping.

2.2 Latent Availability Model

The *latent availability overlay network of degree k* is a general overlay network which has the property that there are at least k distinct vertex disjoint paths in the underlay network between every pair of nodes in the underlay network corresponding to the nodes in the overlay. Formally, a latent overlay graph $G_o =< V_o, E_o >$ of availability degree k on an underlay graph $G_u =< V_u, E_u >$ is a graph such that

1. $\exists VERTEXMAP : (\forall p \forall q((p \in V_o \wedge (q \in V_o) \wedge (p \neq q)) \Rightarrow (\exists a \exists b : ((a \in V_u) \wedge (b \in V_u) \wedge (a \neq b) \wedge (a = VERTEXMAP(p)) \wedge (b = VERTEXMAP(q))))))$

   which means that there exists a mapping function $VERTEXMAP$ which maps every node in the overlay graph to a distinct underlay node.

2. $\forall p \forall q((p, q \in V_o) \Rightarrow (\exists_{i=1}^{k} path_i : ((path_i \in PATHS_u) \wedge (END\_NODES(path_i) = \{VERTEXMAP(p), VERTEXMAP(q)\})) \wedge$

$(\forall j \forall l (j \neq l) \Rightarrow (INT\_NODES(path_j) \cap INT\_NODES(path_l) = \Phi))))$

which means that between every pair of overlay nodes there are k node disjoint paths in the corresponding underlay.

We next analyze the complexity of overlay formation for the two availability models.

3 NP Completeness of Manifest Availability Constraint

Consider the problem, *Given a physical network and an overlay network, is it possible to construct a manifest availability overlay on the physical network.*

This problem is at least as difficult as the problem of determining whether an overlay network satisfying the connectivity constraint also satisfies the manifest availability constraint on the given underlay network. Hence we address the problem of finding whether a given overlay and underlay graph satisfy the manifest availability constraint. The formal statement is: Given an underlay graph G_u and the overlay graph G_o, find a $VERTEXMAP$ and $map \in EDGEMAPS$ such that $\forall p \forall q, p, q \in E_o \wedge p \neq q \Rightarrow$ $INT\_NODES(map(p)) \bigcap INT\_NODES(map(q)) = \Phi$. In the next section, we demonstrate that the problem is NP-complete.

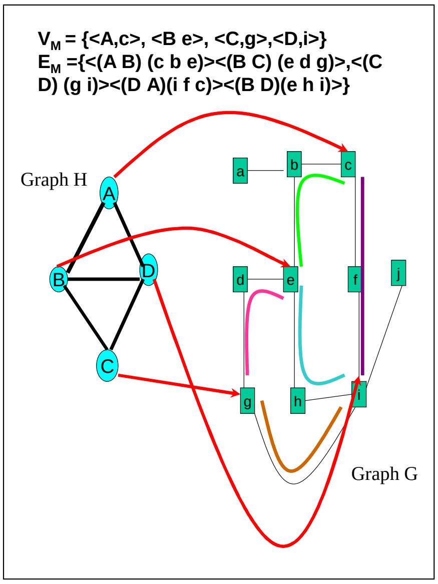
The manifest availability constraint problem, stated in Section 2 is NP-Complete. We prove that by showing that it is
(i)NP
(ii)NP Hard

3.1 Manifest Availability constraint is NP

In order to prove this we have to prove that problem is polynomial time verifiable, which means that given an underlay graph G_u, an overlay graph G_o, $VERTREXMAP$ and $M \in EDGEMAP$, we can verify whether M satisfies the manifest availability criterion in time polynomial in terms of the size of graphs and the map. Each vertex can be checked for inclusion in two paths in time $O(|map| * (V_0))$, using simple linear search. Each overlay link's map can be tested for validity in time $O(|map|)$, by looking up the underlay graph. Hence *Manifest availability constraint is NP.*

3.2 Manifest Availability constraint is NP-Hard

We prove that *Manifest Availability Constraint* is NP hard by showing that Node Disjoint Subgraph



1

Figure 2. A Node disjoint Homeomorphism

Homeomorphism, a known NP Complete problem is polynomially reducible to Manifest Availability Constraint. The **Node disjoint Homeomorphism problem** as defined in [1]is

> "Let H and G be two graphs both directed or both undirected. A subgraph homeomorphism is defined as a pair of one-to-one mappings (v,a), the first from nodes of H to nodes of G; the second from edges of H to simple paths in G. We require that a path in G which corresponds to edge (x,y) of H go from v(x) in G to v(y) in G. the graph H is called a pattern graph. If the image of edges of H is a set of paths which are node disjoint upto end points the homeomorphism is a node disjoint homeomorphism. We say that H is node disjoint homeomorphic to a subgraph of G. The most general node disjoint homeomorphism problem is NP-Complete."

. Node disjoint homeomorphism has been proved polynomial only for simple patterns like triangles and two independent edges [1].

We demonstrate that the general *node disjoint homeomorphism problem* is polynomially reducible to general *manifest availability constraint*, in fact manifest availability constraint is a straight reduction from the general node disjoint homeomorphism.

Reduction

Let G_o be the pattern graph $H =< P_v, P_e >$ and G_u be the input graph $G =< I_v, I_e >$.

Lemma: $< G_u, G_o >$ satisfies the manifest availability criterion if and only if H is a node disjoint homeomorphism of G

Proof:

1. $< G_u, G_o >$ satisfies the manifest availability constraint

   $\Rightarrow \exists VERTEXMAP, \exists map \in EDGEMAPS | \forall p \forall q : ((p, q \in E_o \wedge p \neq q) \Rightarrow INT\_NODES(map(p)) \cap INT\_NODES(map(q)) = \Phi)$

   $\Rightarrow \forall h : (h \in P_v \Rightarrow VERTEXMAP(h) \in I_v) \wedge \forall h \forall i : (h \in P_e \wedge i \in P_e) \Rightarrow (map(h)$ and $map(i)$ are paths in G and node disjoint)

   $\Rightarrow$ H is a node disjoint homeomorphic on G

2. H is a node disjoint homeomorphic on G

   $\Rightarrow \exists v \exists a | \forall h : (h \in P_v \Rightarrow v(h) \in I_v) \wedge \forall p \forall i | ((p \in P_e) \wedge (i \in P_e)) \Rightarrow (a(h)$ and $a(i)$ are paths in G and node disjoint)

   $\Rightarrow \exists VERTEXMAP(= v), (\exists map(= a) \in EDGEMAPS) : (\forall p : p \in V_o \Rightarrow VERTEXMAP(p) \in V_u) \wedge (\forall p \forall q ((p \in E_o) \wedge (q \in E_o) \wedge p \neq q)) \Rightarrow (INT\_NODES(map(p)) \cap INT\_NODES(map(q)) = \Phi)$

   $< G_u, G_o >$ satisfies the manifest availability constraint

Theorem: Manifest Availability is NP-Complete

The second model for availability is based on the underlying connectivity between overlay nodes provided in the underlaying network. It does not call for the rigid condition that any path that is node disjoint in the underlay should be node disjoint at the overlay level also.

4 NP Completeness of the Latent Availability Overlay

We prove that the Latent availability problem is NP-Complete. The first step towards this is to show that Latent availability is NP-Hard.

4.1 The latent availability problem is NP Hard

The problem of latent availability overlay can be stated as: Given an underlay graph G_u, and an overlay graph vertex set V_o, find $VERTEXMAP$ such that every overlay node is mapped to a distinct underlay node and for every node pair in the underlay which are mapped to from overlay nodes by $VERTEXMAP$, there are at least k pairwise node disjoint paths between the node pairs in the underlay. In other words find a $VERTEXMAP$ such that the conditions 1 and 2 above, hold.

The latent availability criterion is NP complete for a general underlying graph and overlay node set cardinality $|V_o|$. We demonstrate the NP Hardness by showing that the k-clique problem [2], a well known NP complete problem, is polynomially reducible to the latent availability problem.

Reduction: The k-clique problem is: Given a graph G and a positive integer k, find whether a k-clique exists as a subgraph of G, and return the k-clique if it exists, and otherwise indicate that a k-clique does not exist.

Let $< G, k >$ be the input to the k-clique problem, such that $|V| > 2$. We polynomially construct an instance of the latent availability problem such that the solution to the latent availability problem answers the k-clique problem,

Construct a graph G_u in the following manner. A sample graph G and the constructed G_u are illustrated in Figure 3.

- STEP I: Corresponding to every node i in G add a node v_i in G_u. Let D be the set of nodes added in this step.
- STEP II: Let $n = |V_G|$. Corresponding to every edge (i, j) in G, add n vertices v_{ij}^1 to v_{ij}^n in G_u, and add edges $(v_i v_{ij}^k)$ and $(v_{ij}^k v_j)$ for all k from 1 to n. Thus $2 * |V_G| * |E_G|$ edges and $|V_G| * |E_G|$ nodes are added. Let I be the set of nodes added in this step.

Find a latent availability overlay of degree n and overlay node cardinality k in G_u.

Lemma:

A k-clique exists in G if and only if a latent availability overlay of degree n with k nodes exists in the underlay graph G_u.

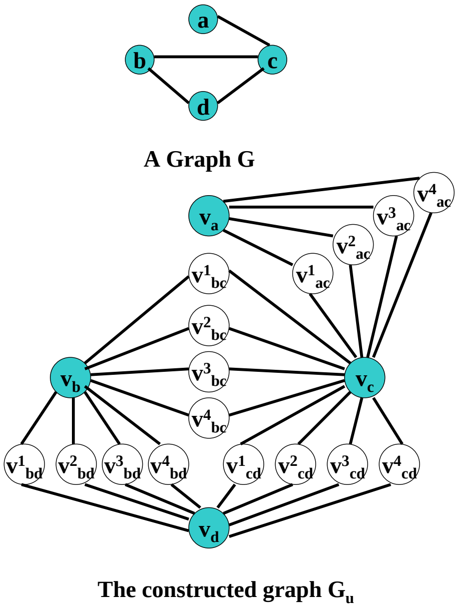


Figure 3. Reduction from k-clique to Latent Availability

Proof:

1. If k-clique exists in G then latent availability overlay of degree n with k nodes exists in G_u

   Let K be the set of nodes in the clique in G.

   $\forall p \forall q : (p, q \in K)$

   $\Rightarrow ((p, q) \in E_G)$

   $\Rightarrow$ v_p and v_q have n node disjoint paths $\forall_{i=1}^{n}(v_p, v_{pq}^{i}, v_q)$between them in G_u

   $\Rightarrow$ If K be the range set for $VERTEXMAP$ of an overlay G_o of size k, then G_o is a latent availability overlay of degree n

   $\Rightarrow$ a latent availability overlay of k nodes of degree n exists in the underlay graph G_u

2. If a latent availability overlay of k nodes of degree n exists in the underlay graph G_u then a $k-clique$ exists in G

   Let $VERTEXMAP$ be the vertexmap obtained in G_u for the overlay network. Let $Q = \{x | \exists y, y \in V_o : VERTEXMAP(y) = x\}$ i.e let Q be the set of nodes in G_u that are mapped to by the overlay nodes.

   Then, $\forall x \forall y (x, y \in Q \Rightarrow \forall_{i=1}^{n} \exists path_i | (END\_NODES(path_i) = \{x, y\} \wedge (\forall j \forall i (j \neq i) \Rightarrow ((INT\_NODES(path_i) \cap INT\_NODES(path_j)) = \phi)))$

   Claim I: For any x $x \in Q \Rightarrow x \in D$, i.e., it is a node added by STEP I of the construction We prove the claim using proof by contradiction.

   Assume $\neg(x \in D)$.

   Then, $x \in I$

   $\Rightarrow$ $degree(x) = 2$ because in STEP II nodes are added with just two edges

   $\Rightarrow$ x can have at most two disjoint paths to any node, as all its paths go through two adjacent nodes

   $\Rightarrow \neg x \in Q$ which is a contradiction. Hence, $x \in D$

   Claim II: There is at least one path between x and y consisting of just a single node of degree two, added in STEP II. Formally,

   $x, y \in Q \Rightarrow \exists path_q \exists p : END\_NODES(path_q) = \{x, y\} \wedge INT\_NODES(path_q) = \{p\} \wedge p \in I$

   To prove by contradiction, let us assume the opposite, i.e. all paths between x and y have one or more nodes added in STEP I, or have more than one node added in STEP II without any nodes added in STEP I.
   Formally,
   $\forall path \in Paths_u(END\_NODES(path) = \{x, y\} \Rightarrow ((\exists a | (a \in INT\_NODES(path) \wedge a \in D)) \vee ((\exists a, b \in INT\_NODES(path) \wedge \{a, b\} \subset I \wedge (a \neq b)) \wedge ((INT\_NODES(path) \cap D) = \phi))))$

   The second clause on the right hand side is not true as a and b have a degree of just 2 and each link of a (and of b) is to a node in D. Hence any path between two nodes belonging to I contains a node belonging to D. Hence the second clause cannot be true. Hence the assumption can be changed to contain the first clause alone, i.e., all paths between x and y have one or more nodes added in STEP I.
   Formally

   $\forall path \in Path_u$
   $(END\_NODES(path) = \{x, y\} \Rightarrow$
   $((\exists a | (a \in INT\_NODES(path)$
   $\wedge a \in D))$

   As there are n node disjoint paths between x and y, each has a different node belonging to $D - \{x, y\}$. But $|D - \{x, y\}| = n - 2$ by STEP I. Hence by the pigeonhole principle, there should be at least two paths without a node from $D - \{x, y\}$. Hence, it must have been true that there is at least one path between x and y which contains a single node p which belongs to I, i.e added in STEP II as $(2 > 1)$.

By construction p would have been added in STEP II only if there was an edge between nodes corresponding to x and y respectively in G.

Hence the nodes corresponding in G to the nodes in the latent availability overlay of size k in G_u form a clique, as an edge exists between each pair of those nodes in G.

Thus G has a clique of size k.

4.2 Latent availability overlay is NP

The problem of Latent Availability overlay is NP can be stated as: Given graphs G_u, G_o, $VERTEXMAP$ which maps every overlay node to a node in G_u, and a positive integer k, it can be verified whether the overlay forms a latent availability overlay in G_u of degree k.

The problem now is to verify whether $\forall x \forall y((x \in V_o) \wedge (y \in V_o)) \Rightarrow \exists_{i=1}^{k} path_i | ((END\_NODES(path_i) = x, y) \wedge \forall_{i=1}^{k} \forall_{j=1}^{k} (i \neq j \Rightarrow ((INT\_NODES(path_i) \cap INT\_NODES(path_j)) = \phi))$

The problem can be polynomially reduced to the k vertex disjoint path problem for directed graphs which is polynomially reducible to the maximum flow problem [2]. The maximum flow problem has an $O(|V|^3)$ solution, using the Relabel-to-Front algorithm [2]. The reduction is explained in the algorithm Latent_Availability and the constructed graph is illustrated in Figure 4.

Algorithm: **Latent_Availability** (G_u, G_o, k)

1. Construct G'_u, a directed graph from G_u, by replacing each node x in G_u with two nodes x and x', adding a directed edge $< xx' >$, and for every undirected edge (x, y) in G_u, add two directed edges $< x'y >$ and $< y'x >$ in G'_u and assign a capacity of 1 to each edge in G'_u.

2. For every node pair $< x'y >$ in G'_u such that $\exists\ v_x \exists v_y | v_x, v_y \in V_o \wedge\ VERTEXMAP(v_x) = x \wedge VERTEXMAP(v_y) = y$, do step 3

3. Run the Relabel to front [2] algorithm on (G'_u, x', y) and determine M, the maximum set of pairwise edge disjoint paths between x' and y. Any node p and its image p' occur in only one path in the set, as the only outgoing edge from p is to p', and the only incoming edge to p' is from p, and $< pp' >$ is included in only one path in M. Hence each path corresponds to a path in G_u which is node disjoint from every other path obtained.

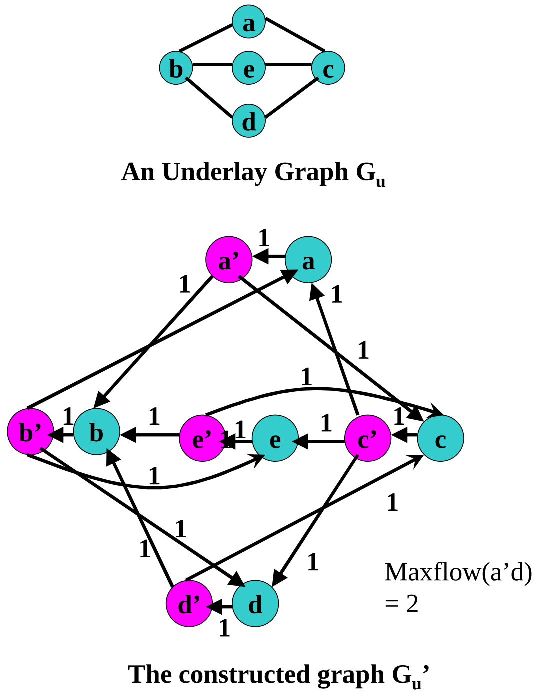


Figure 4. The reduction to maximum flow determination

4. If for every such pair the flow is equal to k or more then G_o forms a latent availability overlay of degree k on the underlay G_u.

Complexity analysis

Step 1 is $O(|V_u| + |E_u|)$

Step 2 is $O|V_o^2|$

Step 3 is $O|V_u|^3$: Thus $O(|V_o|^2 |V_u|^3)$ for the loop execution

Step 4 is $O(|V_o|^2)$

Hence the algorithm is polynomial on the input size. Hence latent availability is NP. As it is also NP Hard it is NP complete.

Theorem: Latent Availability is NP-Complete

5 Pragmatics induced constraints

The complexity for the availability models discussed makes it infeasible to have efficient solutions to the problems unless P=NP. Hence, in order to have methods of generating available overlays, we need to look at pragmatic issues and use the information to restrict the problem to a set of cases and thereby reduce the complexity involved in solving the problem.

An important point is that in a practical scenario on the internet, or a similar large network, every node in the network cannot be a potential broker or run the application software. It is decided by proprietary issues, node capabilities, application requirements and similar external factors. Hence instead of considering the issue of whether a particular availability model can be supported by the given underlay, we can consider the mapping of the overlay nodes to underlay nodes, i.e., the $VERTEXMAP$ as fixed, and analyse whether the availability model can be supported by the set of overlay nodes on the given underlay. We name this as the *spatial invariability constraint.*

Secondly, the application software or broker software runs on an asynchronous distributed system. The information about a particular nodes joining or aspiring to join the broker network may be generated at different points of time, while the broker software or application is running on an existing broker network. We call this as the *temporal variability constraint.*

Thirdly, the whole underlay is not static, and at internet scales, it is also too large for the whole information to be present at every node. Hence only a partial knowledge of the underlay graph could be available at any node. This is known as the *Local Knowledge constraint.*

In the next section we study the effect of the spatial invariability criterion on the two availability models, and the complexity improvement it produces on them.

6 Spatial Invariability

When the spatial availability constraint is satisfied, the nodes to become broker nodes are fixed, i.e the $VERTEXMAP$ is invariable. In this situation the manifest availability criterion can be modified. the revised constraint is:

Given an underlay graph G_u and the set of overlay nodes V_o, and $VERTEXMAP$ which maps nodes in V_o to nodes in V_u find whether there exists E_o and $map \in EDGEMAPS$ such that it satisfies the conditions for the manifest availability, i.e.,

(i) Every overlay node ($v \in V_o$) has d node disjoint paths to every other overlay node in G_o.

(ii) All edges $p, q \in E_o$ are mapped by map to paths in G_u such that

$INT\_NODES(p) \cap INT\_NODES(q) = \phi$

As with the given number of nodes, many overlay graphs of node connectivity d can be formed, and this number is combinatorial, it is given a lower bound by ${}^nC_d$, where $n = |V_o|$. The testing of a given overlay for node disjointedness in the underlay can be done using the technique of flow determination in polynomial time. But the total time for testing all the possibilities would be exponential as the number of possible overlay graphs is exponential.

Under the spatial invariability constraint, the latent availability criterion changes to

Given an underlay graph G_u and the set of overlay nodes V_o, and $VERTEXMAP$ which maps nodes in V_o to nodes in V_u, and availability degree k, find whether there exist k node disjoint paths between every node pair.

We can observe that this problem is the same as the verification problem for Latent availability overlays, described in the previous section. We have already proved it to be polynomial time solvable.

Thus the fixed spatiality constraint advocates the use of the latent availability model for overlay networks.

7 The temporal variability criterion

The temporal variability constraint states that the size of the overlay graph is not constant over time. Nodes may join or leave (stop being broker nodes) at any point of time. Whenever a new broker node is to be added to the overlay, the overlay has to be reconstructed and its availability requirements have to be assured. The impact this has on the two availability models is analyzed below.

7.1 The Manifest Availability Model

If an existing overlay G_u satisfies the manifest availability requirements, then a new broker node has to be added such that the overlay has a node connectivity of d, as well as every edge in the overlay is node disjoint in the overlay. This problem could be stated in two ways.

I. Given an overlay graph G_o, an underlay graph G_u, $VERTEXMAP$ and $EDGEMAPS$ such that the Manifest Availability Constraint are satisfied, and an underlay node b' determine a new overlay graph G'_o such that $V'_o = V_o \cup x$, with $VERTEXMAP' = VERTEXMAP \cup < x, b >$, and G'_o satisfies the manifest availability criterion on G_u.

II. Given an overlay graph G_o, an underlay graph G_u, $VERTEXMAP$ and $EDGEMAPS$ such that the Manifest Availability Constraint are satisfied, and an underlay node b' determine a new overlay graph G'_o such that $V'_o = V_o \cup x$, $E_o \subseteq E'_o$ with $VERTEXMAP' = VERTEXMAP \cup < x, b >$ such that the manifest availability criterion is satisfied by G'_o on G_u.

In the first case new manifest overlay graphs of connectivity k having one more node than before have to be found. Hence all possible such overlay topologies have to be generated and tested for the manifest availability criterion, calling for an exponential time computation.

In the second case, the existing overlay and $EDGEMAP$ elements are retained, and new edges and paths are to be added to $EDGEMAP$ such that it has no nodes in common with existing paths in $EDGEMAP$. The possible new overlay edge sets have to be tried out, thus ${}^{V_{[o|-1}}C_d$ number of combinations have to be evaluated for the manifest availability criterion, again giving an exponential time solution.

7.2 The Latent Availability Model

In the latent availability model, the temporal variability constraint problem says that the new node to join the network should also have k node disjoint paths to each existing overlay node. The problem can be formally stated as he problem of latent availability overlay can be stated as:

Given an underlay graph G_u, and an overlay graph vertex set V_o, and $VERTEXMAP$ such that every overlay node is mapped to a distinct underlay node and for every node pair in the underlay which is mapped to from an overlay node by $VERTEXMAP$, there are at least k pairwise node disjoint paths between the node pairs in the underlay, an underlay node b, determine G'_o, such that $V'_o = V_o \cup v_x$, $VERTEXMAP = VERTEXMAP \cup v_x, x$ and every node pair $p, q \in V'_o$ is such that there are k node disjoint paths in G_u between $VERTEXMAP(p)$ and $VERTEXMAP(q)$.

This problem is polynomial time solvable, as it amounts to determining whether the x has k node disjoint paths in overlay to every node which is mapped to by $VERTEXMAP$ from a node in V_o. Thus the technique outlined in Step 3 of Section 4.2 can be repeatedly applied to the $|V_o|$ node pairs each taking polynomial time, to ascertain the latent availability constraint satisfaction.

8 The Local Knowledge constraint

The Local Knowledge constraint states that the full knowledge of the underlay graph is practically very large for large scale networks, and hence a full knowledge of the underlay for individual nodes is impossible to store. For the manifest availability constraint, for every new node joining the broker network, the full knowledge about the overlay graph and its underlay map is required to ascertain the satisfaction of availability constraints. For the latent availability overlay, at a time only the knowledge about the paths to a particular overlay node are needed, hence the knowledge requirement is "local" and more manageable.

In this manner, all the constraints indicate that latent availability model is more practical, economic and easy to use. Hence we concentrate on the latent availability model availability model for underlay aware highly available overlays.

9 Reduced Underlay Graph

We define a reduced underlay graph of an underlay graph G_u as a multigraph $G_{ru} =< V_{ru}, E_{ru} >$

where $V_{ru} = V_u - \{x|(degree(x) = 1) \vee (\neg(\exists u(VERTEXMAP(u) = x)) \wedge (degree(x) \leq 2))\}$

and $E_{ru} = \{(x, y)|(x, y \in V_{ru} \wedge ((x, y) \in E_u) \vee \exists path \in PATHS_U(END\_NODES(path) = (x, y) \wedge (INT\_NODES(path) \cap V_{ru} = \phi))\}$

Thus in the reduced underlay graph the nodes of degree 1 (pendant) are not included. Nodes of degree two which are not broker nodes (connector nodes) are simply "joined across" by their adjacent nodes and removed from the picture.

> **Lemma:** An overlay graph G_o forms a latent (or manifest) availability overlay of degree k on an underlay graph G_u if and only if G_o forms a latent (or manifest) availability overlay of degree k on the reduced underlay graph G_{ru}

Proof:
1. *If G_o forms a latent (or manifest) availability overlay of degree k on the reduced underlay graph G_{ru} then G_o forms a latent (or manifest) availability overlay of degree k on an underlay graph G_u.*

We claim that any two paths in G_{ru} that are internally node disjoint in G_{ru} have mappings to distinct node disjoint paths in G_u. Let $(x, p_1, p_2..p_n, y)$ and $(x, q_1, q_2..q_m, y)$ be the two node disjoint paths in G_{ru}. Then $p_1, p_2..p_n$ and $q_1, q_2..q_m$ are all expander nodes or overlay nodes. Consider any edge $(p_i p_j)$ or $(q_l q_k)$ in the path. It implies that either there was an edge in G_u between the nodes or that there was a path between them in G_u consisting of only connector nodes (degree 2). Hence the paths $(x, p_1, p_2..p_n, y)$ and $(x, q_1, q_2..q_m, y)$ have corresponding paths $(x, p_{01}, ..p_1, p_{11}..p_2...p_{n-11}..p_n, p_{n1}..y)$ and $(x, q_{01}, ..q_1, q_{11}..q_2...q_{n-11}..q_n, q_{n1}..y)$ in G_u, where all the nodes between p_i and p_{i+1} or q_i and q_{i+1} are connector nodes. It can be proved that the same node

x cannot be p_{ij} as well as q_{kl} which means that the connector nodes in the first path are distinct from the connector nodes in the second path. If there was such a node x, it is connected to only two nodes as its degree is two. Hence the adjacent nodes would also have to be common in both paths. following this line of reasoning for all the succeeding two degree nodes, the expander/overlay nodes p_j and q_l will have to be the same, thus implying that the paths were not internally node disjoint in G_{ru}.

2. *If G_o forms a latent (or manifest) availability overlay of degree k on an underlay graph G_u then G_o forms a latent (or manifest) availability overlay of degree k on the reduced underlay graph G_{ru}*

We claim that any two paths in G_u that are internally node disjoint in G_u have mappings to distinct node disjoint paths in G_{ru}. Let $(x, p_1, p_2..p_n, y)$ and $(x, q_1, q_2..q_m, y)$ be the two node disjoint paths in G_u. Each node in this is either a connector node or an expander/overlay node. Consider two nodes a and b in such a path that are expander/overlay nodes. Then a and b have an edge between them in G_{ru}. Thus there is path in G_{ru} between x and y consisting of expander/overlay nodes in the path in G_u. As the paths are internally node disjoint in G_u, the set of internal nodes in the path in G_{ru} being a subset of the nodes in G_{ru}, the paths in G_{ru} are also internally node disjoint.

Hence, we further restrict our problem to that of finding overlays for the given set of overlay nodes on the reduced underlay graph V_{ru}, satisfying the latent availability constraint.

10 Conclusion

In this paper we present two models of availability for overlay networks, Manifest and Latent Availability. We have analyzed both the models for complexity of formation and maintenance, and proved that in the general case, both are NP-complete. Then we identified a set of practical constraints applicable to large scale networks and demonstrated that under these constraints, latent availability constraint becomes a polynomial time problem. The concept of reduced underlays, which further reduces the complexity of the problem of determining latent availability overlays is also introduced.

References

[1] Andrea S Lalpaugh and Ronald L Rivest. The Subgraph Homeomorphism Problem. In *Proceedings of the tenth annual ACM symposium on Theory of Computing*, pages 40–50, November 1978.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* Prentice Hall of India, 2002.