Metric Induced Network Poset (MINP): A model of the network from an application point of view

Laurent Bobelin [1,2,3]
[1] CS Communications et Systèmes, 200 rue Pierre Duhem BP 389 13799 Aix-en-Provence Cedex 3 France
[3] CPPM - Centre de Physique des Particules de Marseille 163 avenue de Luminy - case 902 - 13288 Marseille Cedex 09 France
laurent.bobelin@c-s.fr

Traian Muntean [2]
[2] Mediterranee University, Parc Scientifique de Luminy, ESIL - F-13288 Marseille Cedex France
traian.MUNTEAN@univmed.fr

ABSTRACT

Nowadays grids connect up to thousands communicating resources that may interact in a partially or totally coordinated way. Consequently, applications running upon this kind of platform often involve massively concurrent bulk data transfers. In order to optimize overall completion times, those transfers have to be scheduled based on knowledge about network performances and topology.

Identifying and inferring performances of a network topology is a classic problem. Achieving this by using only end-to-end measurements at the application level is a method known as network tomography. When topology reflects capacities of sets of links with respect to a metric, the model used to represent the topology obtained is called a Metric-Induced Network Topology (MINT). Such a type of representation, obtained using statistical methods, has been widely used in order to represent performances of client/server communication protocols.

However, it is no longer accurate when dealing with grids. In this paper, we present a novel representation of the infered knowledge from multiple source and multiple destination measurements.

1. INTRODUCTION

Nowadays grid testbeds often aim to link together up to thousands of computing and data storage resources over the world. Connectivity is ensured using either the Internet, or *high bandwidth.delay* networks such as GEANT in Europe [3] or TeraGrid [5] in US. An example of the physical topology of such a network is given in figure 1.

Upon such a kind of testbed, applications usually deploy software and resources dedicated to bulk data transfer. For example, EGEE project [1] uses a notion of a hierarchy of *tiers*, as illustrated in figure 2. In such a hierarchy, each tier is a data storage center physically located in one of the project partner's lab. Level of each tier reflects the contribution of the project partner owning that resource. Tier-0 is located close to the experiment place (for EGEE, at CERN). Tier-0 communicates to tier-1, tier-1s can communicate with every tier-1s and to a subset of tier-2s and tier-2s communicate to a subset of tier-2s and a subset of tier-3s. Tier-1 are national or institutional centers, tier-2 are located close to large computing centers, tier-3 are located in labs. In such a case, the data transfer paradigm is no more a client/server one: each host is a source, a destination, or both, and each source communicates to a subset of destinations.

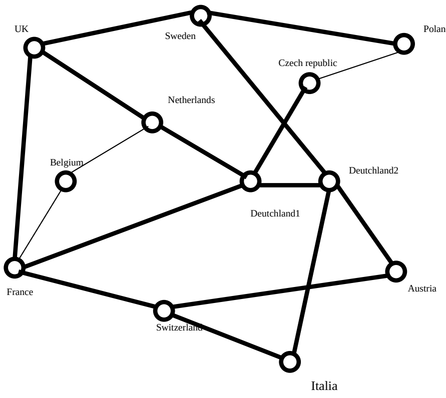

Figure 1: Overview of GEANT physical topology

This logical organization is mapped into the physical existing network as illustrated in figure 3. As we can see, this mapping can imply that logically separated links are physically the same. For example, links between Italian and French tier-1 and between French tier-1 and tier-0 are logically separated but have physically a common subpath.

Therefore, it is mandatory to know capacity and topology of the underlying network in order to optimize communications between tiers. If not, some logically independent transfers may compete for the same physical network resource while optimal performances would require transfers not to be scheduled simultaneously. Unfortunately, most of the time, physical topology is unknown. Moreover, existing monitoring tools like NWS [21] or WREN [17] allow

GridNets 2007 October 17-19, 2007, Lyon, France.
Copyright 2007 ICST ISBN 978-963-9799-07-3.
DOI 10.4108/gridnets.2007.2246

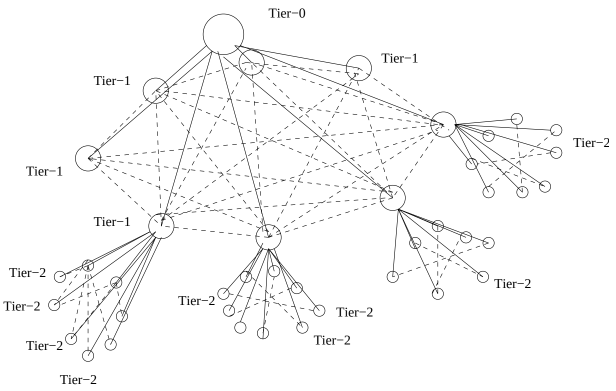

Figure 2: Overview of a n-Tier organization similar to EGEE

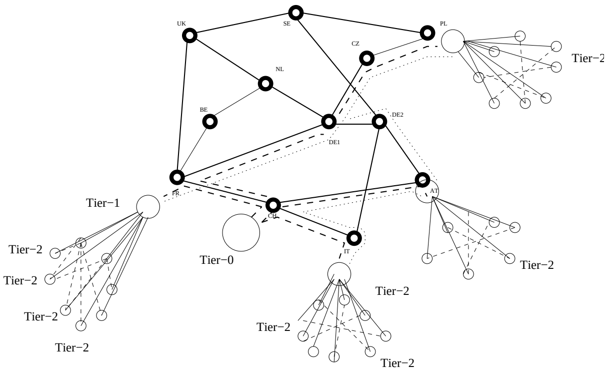

Figure 3: Tier organization plunged into physical topology

to model only basic interactions between transfers. In their model, either transfers occur between hosts belonging to the same group (called *clique*) and then share a common link, or not. If not, transfers are considered as not interfering with each other.

Most of the time, the topology discovery can be done using tools like traceroute [16]. The resulting topology is unlabeled. It is formed by matching IP address of network equipments belonging to the different observed paths. Moreover, these tools use information that can only be obtained if network administrators allow doing so. As a grid application runs on hosts owned by organizations applying different security policies, using such tools is most of the time not realistic. In order to infer a topology one must use only application level measurements. Such a method is known in the literature as network tomography [22].

Since a decade, network tomography has been widely studied. Different approaches have been used, depending both on the needs expressed and on targeted network (see [12] for a state of the art). Most of the time, topology is inferred using values of links for a given metric. This metric can be for example the maximum achievable bandwidth or the delay. Such a topology is an oriented graph where each edge is labeled with the capacity of the set of physical objects it represents. In client/server case, this topology is a tree. The root is the server, the leaves are client and inner nodes are disjunction point of paths between the server and clients. Vertices are labeled with the capacity (in respect to the metric considered) of routers and wires belonging to the subpath considered. Such inferred topologies are known as Metric Induced Network Topologies (MINT).

This kind of topology inference is an inverse problem. Most of the time, it is solved using statistical techniques that aim to estimate likelihood. Roughly speaking, it consists to collapse inferred points into one when capacities of the paths leading to those inferred points are similar. These methods have drawbacks. Most of all, it relies on the assertion that the resulting topology is a tree. But as mentioned in [10], a tree cannot characterize the network when multiple sources and multiple destinations are involved.

The remainder of this paper is organized as follows. First, we give an overview of existing work addressing the problem of modelling the networks in section 2. We motivate why we define a subproblem of the MINT problem based on multiple source/multiple destination end-to-end measurement in section 3. We define the terms and notations used in this article in section 4. Our model, the *Metric Induced Network Poset* is described in section 5. We exhibit the relationships between our model and existing knowledge representation in section 6. Finally, we conclude in section 7.

2. RELATED WORK

As stated before, the first formalization of necessary conditions that a metric must satisfy to allow a metric-induced topology reconstruction has been given in [8]. Nevertheless, those definitions are no longer valid when the problem shifts from client/server to a multiple sources, multiple destinations.

For the classical case of a single source communicating to a set of destination, the problem has been widely studied. Different approaches have been tested. Both passive [18] and active [8] measurements have been used. It has been applied to cases such as one source communicating to many destination or many sources communicating to a single destination. Reconstruction techniques are most of the time similar : they are based on statistical methods (see [12] for a state of the art). The main differences occur in the measurements procedure. Measurements are mainly realized using packet train techniques but can also be based for example on multicast trees [10].

Up to now, some studies have focused on finding a topology for the multiple source/ multiple destination, but only a few tried to characterize the topology produced. In [19], authors use the existing MINT model to induce tree topologies. Then, they infer subpaths common to two trees. And by this mean, they infer conjunction points between trees. The main drawback relies in the fact that "having a common subpath" is not transitive. Indeed, if a path a has a common subpath with a path b, and if b has a common subpath with a path c, that does not mean that a has a common subpath with c. Even if a has a common subpath with c, it does not mean that there is a subpath common to all paths a,b and c. Therefore, conjunction point exists only between two trees. The method used is close to the one used in [10] where identification of common subpath is done on edges belonging to multicasts trees.

In [15], authors formalize a problem close to our. The idea is to reconstruct a topology by detecting the subpaths common to flows by using a metric related to bandwidth without labeling the edges. The notion of interference used there is close to the notion of having common detectable subpath. Moreover, the metric used avoids any labeling.

Other authors rely on active but ”stealth” measurements (i.e. without requiring the collaboration of destinations) in order to reconstruct unlabeled topologies [20]. They use *Round Trip Time* in order to infer common links between flows. Anyhow, their method cannot infer labeled topologies, and is thus useless in our case.

An interesting work on finding how a subset of flows interfere with each other passively has been done in [7]. Author is using passive measurements (i.e. traces from TCP flows from various sources to various destination). They correlate flows that have interacted with each other in order to detect potential common bottlenecks using time-based statistical methods. Their network knowledge is represented in a model where TCP flows are grouped in classes where each flow shares the same bottlenecks. This model can be viewed as a subset of the Metric Induced Network Poset we present here.

3. MOTIVATION

As stated before, most grid projects deploy software and resources dedicated to bulk data transfer. For example, EGEE project has a dedicated set of resources managing the *File Transfer Service* (*FTS* [2]), provided by the *gLite* [4] project. This service aims to reliably copy persistent sets of files from a site to another. It uses a 3rd party copy (e.g. gridftp [6]) to achieve this. This middleware component offer clients a web service interface to which they can submit a request to copy a file from one grid storage resource to another. Once a request is submitted, it is inserted in a Transfer Job Database containing all transfer requests. Regularly, transfer agents check for new transfer requests. Each transfer agent finally schedules these transfers according to *Virtual Organization* [13] own internal policies, while trying to optimize network usage (see figure 4 for an overview). *FTS* has its inner logical organization between hosts. It defines sets of *channels*, which are directed links between hosts. Only those *channels* are used to transfer data.

For this kind of service, it is mandatory to have a both accurate and adequate vision of the network and its capacities. *FTS* for example focuses on bandwith and hence does not have a real need for data like a realistic vision of the network, picturing each equipment deployed along the paths used to transfer data. *Metric Induced Network Topology* provides a much more adequate model for such a service. Because *MINT* only models capacities of paths and common subpaths to sets of paths, it is much more shorter than a complete description of network topology labeled by its capacity. In an earlier paper [9], we have formally redefined what is a *MINT* representation of a network when it is induced by a multiple sources and multiple destination communication paradigm. However, even this kind of network topology representation contains additional useless information.

Let consider figure 5 (1). The two graphs on the left depicts two simple possible topologies ; white circle indicates sources and black ones destinations. Suppose that the logical organization of flows only allows to use flows coming from a to a', b to b' and so on. Suppose the capacity of the link e equals those of e' link as capacity of f equals those of f'. As nowadays networks protocols are end-to-end protocols and equipments deployed are most of the time only able to constitute *”dumb networks”*, they behave like black boxes where sources only inject packets into and destinations hope to receive it ([14]). Congestion control is mainly done at end hosts. Because of this, those two different topologies will behave similarly. The precedence relation between edges e and f depicted by figure 5 is useless. This enlight the fact that it would be interesting to have a simpler way to model network performances in order to give to a service such as *FTS* only the significant informations.

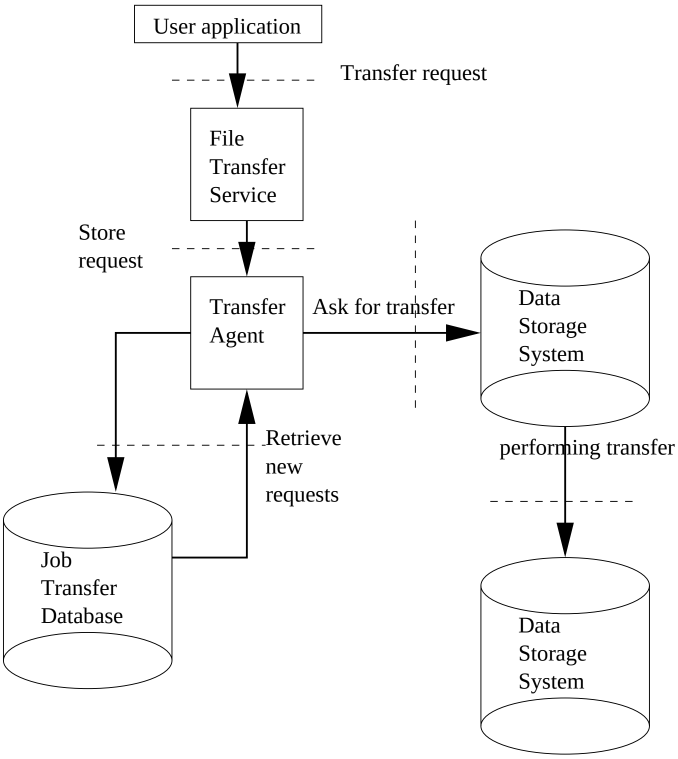

Figure 4: Overview of File Transfer Service

Moreover, reconstructing a Multiple Source Multiple Destination *MINT* (*MSMDMINT*) is a tricky ill-posed, ill-defined inverse problem, as multiple solutions can be found for a single statement of this problem. By specifying a new model to reconstruct instead of the whole *MSMDMINT* in the next section, we define a new subproblem of the general *MSMDMINT* which is *a priori* easier to solve, because it is well-defined.

4. NOTATIONS

4.1 Vocabulary

We will call a *probe* the atomic action of injecting messages into the network in order to determine its properties. The complete process of injecting probes in order to discover the entire targeted network will be called *measurement procedure*. Except when explicitely stated, we will assume that there is no cross traffic. Hereafter in this article, we will similarly assume that routing is *consistent* and *stable*. By the former, we suppose that routing function does not allow routing paths to join, fork, and join again. By the latter, we suppose that routing paths will not change during the whole probing process.

We consider the network as an oriented graph $G = (V \cup S \cup R, E)$ where vertices V are network equipments such as routers, hub, etc., S the set of hosts which will behave like senders, R the set of hosts which will behave as receivers and E physical links between them ($E \subset V \cup S \times V \cup R$). We will note l_{ij} a directed edge from i to j. A host that is both a source and a destination will be considered as two different hosts, one source and one destination.

Upon this graph, routing function defines a set of paths. If routing is *consistent*, there is a unique path between each source a and destination b. Indeed if two paths exists between a and b, that means that they have joined in a, then fork, and join again in b. We will note p_{ab} the path from $a \in S$ to $b \in R$. This path is an ordered sequence $p_{ab} = \{l_{ai}, l_{ij}, l_{jk}, ..., l_{qb}\}$ of directed edges $l_{ij} \in E$. We will use either link or edge in order to name such l_{ij}. Each directed edge of any path starts from the destination of the edge preceding it (if such an edge exists). A subpath of p_{ab} is a subsequence of this sequence that satisfies the path definition between a source $a' \in S \cup V$ and a destination $b' \in V \cup R$. We will say that this subpath is *contained* by p_{ab}. We will call *length* of a path the number of directed edges in the sequence. The set of all paths defined by the routing function between each source $s \in S$ and each destination $r \in R$ will be noted P_{e2e}. It is the set of end-to-end paths. The set resulting of the union of P_{e2e} and the set of subpaths of each of its elements without repetition will be noted P. We will call *flow* probes packets going through an element of P.

We will call *common subpath* to a set of paths P_s a subpath *contained* by each element of P_s. We will call *common maximum subpath* of a set of paths P_s the longest *common subpath* of P_s. If *consistency* holds, the longets common maximum subpath is unique for a given P_s. This subpath will be noted $p^{P_s}_{maximum}$. We will say that paths contained in P_s *admit* a common maximum subpath. The set of *common maximum subpath admitted* by at least one subset of P will be noted Max^P.

4.2 Metric

A *metric* is a function whose initial domain is the set of flows and whose range is reals. As flows are defined over paths, the value obtained for a flow can label a path. We will note c^p_m the *capacity* of a path for the metric. For example, if the metric m is the delay, the *capacity* c^p_m of a path will be equal to the sum of the delay induced by each directed edge composing it.

A capacity of a path p will be *detectable* if there exists a set of paths containing p such that probing over those paths can exhibit capacity of p. For example, if the metric is the throughput achievable by TCP flows on steady-state, then the capacity of a subpath can be detected only if it is feasible to saturate this path. An *undetectable* capacity of a path can be for example a path inducing no delay for the delay metric, or a path with infinite capacity if the metric is the bandwidth.

A metric will be *constant* with respect to *measurement* if the capacity c^p_m does not depend on the paths followed by *probes* that *detect* it. For example, if the metric is the delay induced by a path, the capacity of a subpath common to a set of path will be the same for each of these paths. The ratio of achievable bandwidth between two cooccurring TCP flows on a same subpath is a *non-constant* metric. Indeed, two TCP concurrent flows will share bandwidth according to their respective *round trip time*. Therefore, two pairs of TCP flows admitting the same common subpath will not share the achievable bandwidth the same way, and will exhibit a different ratio.

5. METRIC INDUCED NETWORK POSET

This section is devoted to the definition of the model used to describe the network for services such as FTS.

5.1 Definition

A metric induced network poset is a poset $P^m = (X, \prec)$ formed from $Max^{P_{e2e}}$.

- X is defined by the relation $\forall i \in Max^{P_{e2e}}$, i *detectable* for the metric $m \iff i \in X$,
- $\prec$ is defined by the relation $\forall i, j \in X, i \subset j \iff j \prec i$,
- Every element of $p \in X$ is labeled by its capacity c^p_m.

For practical issues, we add an upper join node and lower node to the poset. The upper node is an element of X that we will note Ω which is a detectable path of length 0 and which is linked to other elements of X by the relation $\forall i \in X, \Omega \prec i$. For the bandwidth metric, the label of Ωcould be $c^{\Omega}_m = \infty$. Because of the definition of the Metric Induced Network Poset, the resulting poset is a join-semi-lattice, as Ω is by definition the supremum of the MINP.

We add a lower node p_∞ in order to have a lattice structure, which is easier to represent and work with. This lower node can be defined as the set E containing all networks links. We usually do not depict it.

Roughly speaking, this model does not anymore represent the topology, but *detectable* common subpaths for any of the subpaths of P_e2e and the set of longer (sub)paths in which they are included.

5.2 Drawing

We will use a Hasse diagram graphical rendering of partially ordered sets. We display the poset via the cover relation (the transitive reflexive reduction of the partial order) of the partially ordered set with an implied upward orientation. In a Hasse Diagram a point is drawn for each element of the poset, and arcs are drawn between these points according to the following two rules:

- If $x \prec y$ in the poset, then the point corresponding to x appears lower in the drawing than the point corresponding to y.
- The line segment between the points corresponding to any two elements x and y of the poset is included in the drawing iff x covers y or y covers x.

We will display P_{e2e} elements at the bottom of the drawing. When displayed, the upper node is the conceptual subpath Ω.

In figures below, white circles depicts sources and black ones destinations. Squares are physical router or hubs that are conjunction/disjunction points for paths of P_{e2e}. Hereafter parameters $c^e_m, c^{e'}_m, c^f_m, c^{f'}_m, c^g_m, c^h_m$ and c^i_m depicts capacities of edges. Dotted lines depicts various routes of flows between sources and destinations. We consider that the logical organization of transfers only allows to communicate from a to a', b to b' and c to c'. As we focus mainly on achievable bandwidth, we will consider hereafter this metric.

5.3 MINP and available bandwidth

Figure 5 (1) represents two 3 sources 3 destinations topologies. We will consider that all other links in the picture have higher capacity than e and f. As stated before, these topologies will have similar impact on communications performances if $c^{e}_{Bandwidth} = c^{e'}_{Bandwidth}$ and $c^{f}_{Bandwidth} = c^{f'}_{Bandwidth}$. This is explained by the fact that path $a \rightarrow a'$ and path $b \rightarrow b'$ will share in both case a narrow link of available bandwidth $c^{e}_{Bandwidth}$ and that the common narrow link to all paths will have a capacity of $c^{f}_{Bandwidth}$.

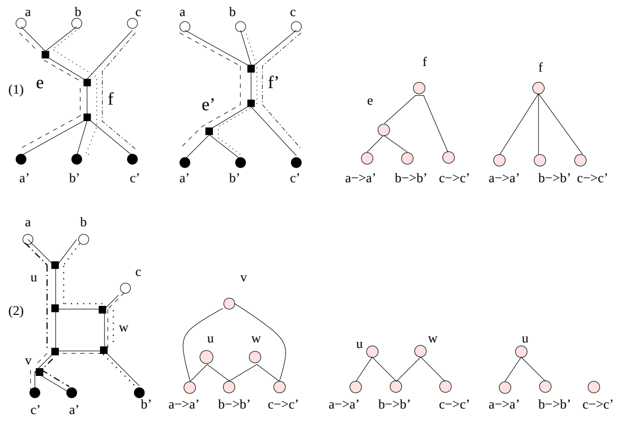

Figure 5: Simple topologies and their representation in the metric induced network poset

The two possibles MINP depicted on the right correspond to two different relation between the values of $c^{e}_{Bandwidth}$ and $c^{f}_{Bandwidth}$. The MINP on the left depicts the case where $c^{e}_{Bandwidth} < c^{f}_{Bandwidth}$. The upper node represents the subpath f, the middle one the subpath formed by the links e and f, and finally the lower nodes represents, from left to right paths $a \rightarrow a'$, $b \rightarrow b'$ and $c \rightarrow c'$. We do not depict neither the infimum and supremum here. The MINP on the right depicts the case where $c^{e}_{Bandwidth} \geq c^{f}_{Bandwidth}$. In such a case, the subpath e is not *detectable* as no subset of P_{e2e} can saturate this link while probing simultaneously. The difference between the two possible cases in real life can be caused by cross-traffic, when dealing with available bandwidth. If we consider that the wire capacity of path e is constant, the former stands for a case when cross traffic make the available bandwidth decrease on e so that capacity of e appears *detectable.* This is important, as it means that a MINP representation of a network depends not only on network topology and paths included in P_{e2e} but also on cross-traffic.

Figure 5 (2) represents an interesting situation : each pair of flows shares a common subpath, but there is no common subpath to all flows. This implies that a tree-based representation cannot be constructed for this network configuration. If u, v and w are narrow links with similar capacities ($c^{u}_{m} = c^{v}_{m} = c^{w}_{m}$), then the corresponding MINP is the one on the left, because each narrow link is *detectable.* The MINP on the middle represent the case similar to $c^{v}_{m} > c^{w}_{m}$ and $c^{u}_{m} = c^{w}_{m}$. In such a case, no injection of flow will saturate this link, and we will only have 2 *detectable common subpaths.* Finally the MINP on the right represent the case where c^{v}_{m} and c^{w}_{m} are not detectable, because $c^{v}_{m} > c^{w}_{m} > c^{u}_{m}$ for example.

5.4 Existence

A Metric Induced Network Poset can always be constructed from a network under the following assumptions. Paths need to be *stable* in order to be decomposed in *detectable* subpaths which can be labeled only if the metric is *constant.* If not, paths decomposition into subset of links is not feasible. Authors in [11] states that paths over the internet is highly stable. If we consider internet has properties similar to our target network, it implies that paths are quite *stable.* Under this assumption, one can prove *the existence of a MINP for any network.*

PROOF. Let us consider the network as an oriented graph $V(G, E)$ and an associated *stable* routing function. The routing function is a mapping between the set of end-to-end paths P_{e2e} and sets of consecutive links in E. Each image of each element of P_{e2e} by the routing function in E can be decomposed into subset of adjacent links. Such sets of elements of E are either *detectable* or not. *Detectable* sets that belong to the $Max^{P_{e2e}}$ set are by definition contained by the power set of E. Power set of E exists because of the axiom of the power set ; so does its elements. The poset formed by the power set $P(E)$ and the $\subset$ relation is trivially a lattice, with empty set as supremum and empty set as infimum. The subset of $P(E)$ formed by *detectable* subpaths included in $Max^{P_{e2e}}$ plus the empty set and E itself forms by definition, a MINP. The resulting poset is still a lattice, as it has still a supremum and an infimum (respectively the empty set and E). So a MINP exists for any network. □

5.5 Uniqueness

As we have stated before, several MINP can be infered for the same network, depending on cross-traffic. So, by stating that unique MINP exists for a given network and its *stable* routing function, one must also make some assumption about cross-traffic. One must make the assumption that cross-traffic does not change significantly during the measurement process that leads to establish which edges $e \in E$ are *detectable* and can be labeled only if the metric is *constant* and which are not. Under these assumptions, one can prove that *it exists unique MINP for a given network.*

PROOF. Because of the cross-traffic that does not change, the *detectability* relation is a injection from the set $P(E)$ to the set of *detectable* subpaths. Hence the MINP definition is based on a injection from $Max^{P_{e2e}}$ poset to MINP and one from the $\subset$ to the $\prec$, a MINP is unique for a given $Max^{P_{e2e}}$ and a $\subset$ relation. Hence the point is to prove the uniqueness of the $Max^{P_{e2e}}$ set and $\subset$ for any $G(V, E)$ and any associated *stable* routing function. As $Max^{P_{e2e}}$ contains (by definition) all *common maximum subpaths* for the set P_{e2e} one *cannot* find two different $Max^{P_{e2e}}$ for a given P_{e2e}. As routing paths are supposed to be stable, there is only an unique P_{e2e} for a given $G(V, E)$ and its associated *stable* routing function. As the $\subset$ relation does not depend on the network state, but on set property, one can state that there is a unique MINP for a network. □

5.6 Well-definedness

Because of the existence of unique MINP for every possible network with the constraints given previously, one can state that the problem of discovering a MINP representation is well-defined. This is important, as we are dealing with an inverse problem.

Stability is mandatory. If paths are not stable enough to infer at least a MINP snapshot of the network, then multiple MINP can be infered for the same network. As stated before, internet paths are stable enough, so it is a quite realistic assumption.

So does the *non-changing cross-traffic* assumption, as significant changes in cross-traffic can lead to a change of *detectability*. The significance of a change, however, should differ according to the measurement process, but this is out of the scope of this paper. However, the traffic needs to remain relatively stable only during the measurement process. After infering an initial MINP, if paths are stable enough, we think that it would be quite efficient to update this representation instead of regularly reconstruct this network from measurements. We believe, as reconstruction involves costly measurements (in terms of time), that the better way to use MINP is first to infer it from active measurements, then *update* it from passive ones.

The changing cross-traffic also enlighten why the inverse problem of finding a MINP from end-to-end measurements is ill-posed. The solution lacks of stability, as small changes in cross-traffic can lead to really different solutions.

5.7 k-detectability

Detectability as we have defined in section 4 is a property of a subpath p. p is *detectable* if there is at least a subset P of P_{e2e} such that a probe applied to P can exhibit the capacity of p.

Hereafter we need a more restrictive definition of *detectability* in order to enlighten possibilities of reconstruction of a MINP representation from end-to-end measurements. We enhance this notion by defining *k-detectability*.

K-DETECTABILITY 1. *A subpath p is k-detectable if there is at least one subset of $P' \subseteq P$, $\quad |P'| \leq k$ such that a probe applied to P can exhibit the capacity of p.*

We illustrate this notion in figure 6. As usual, we depicts sources as white circles and destination as black ones and consider only flows going from a to a', b to b' and so on. Let the metric m be the achievable bandwith in steady state of TCP flows. Depicted values on the figure correspond to the c^p_m of the links. Let suppose that we use a measurement procedure defined for each P^{probe}, where $P^{probe} \subseteq P_{e2e}$, an $|P^{probe}| = k$. One can notice that links that has a c^p_m of 1 are *1-detectable*, as they represents bottlenecks for each depicted flows. The e link is *2-detectable* as it becomes a bottleneck only when simultaneous flows are injected other the two first paths. Finally, link f is *4-detectable* as we need to establish a flow between each pair of source and destination in order to saturate this link.

This is important, as practical algorithms cannot rely on a probe involving all paths in P_{e2e}.

This naturally leads to a taxonomy of measurement procedures : a measurement procedure allowing the detection of *any k-detectable* subpath will be named a *k-measurement procedure*. For the example given above, a measurement procedure that will repeat the upper measurement procedure for each pair of flow is a *2-measurement procedure*.

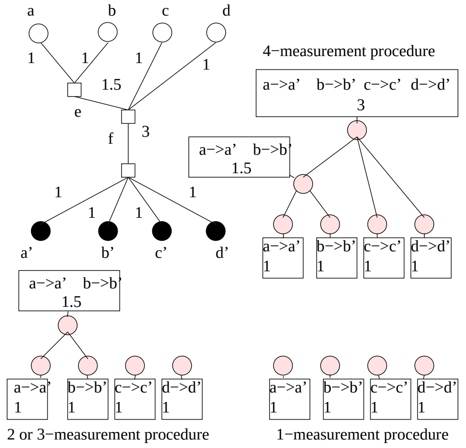

Figure 6: Sample topology

This basic definition has a direct impact on MINP reconstructed from a measurement procedure involving tests for n paths each time a probe is run. For example, using achievable bandwidth as the metric allows to state that any element of X will have a value such that $c^i_{Bandwith} \leq \sum_{j \in S} c^j_{Bandwith}$ where S is the set of elements of X such that $j \in S \iff i \preceq j$ and i covers j and $|S| \leq n$. It also means that the link f will be *undetectable* by using 1, 2 or *3-measurement procedure*, as depicted at figure 6.

An interesting issue is that *k-detectability* is still an injection from the elements of $P(E)$ to the set of paths. By applying the same reasoning that in previous section, one can prove that the MINP reconstruction is still well-defined, even with *k-detectability* instead of *detectability*.

6. RELATIONSHIP WITH EXISTING MODELS

6.1 MINT

Tree representation is the most frequently used way to model interaction between flows in a one-to-many paradigm. Those trees can be either deduced from probes from one source to many destinations or from many sources to one destination. Basically, one can transform the trees inferred from a one-to-many paradigm easily in reconstructing the semi-lattice. Let's note $T(V, E)$ the tree inferred from one server to n clients. Tree-based logical topologies are labelled on edges, each label representing the capacity c in respect to the metric m of the link between each preceeding/following conjunction/disjunction of flow. The label of the edges connecting inner nodes to the leaves are the maximum capacity that can obtain a flow with itself. So, the transform from one logical tree inferred from a conjunction/disjunction probe into a MINP is straigthforward by using the line graph of the initial graph, and its representation is still a tree.

On the other hand, and as stated before, general topologies can exhibit common subpath that are not in the set of edges and vertices of any tree-based representation, and hence, no graph operation without losses of information can be found between the two model. However in order to transform a MINP into a set of tree-based topology, one can do as describe hereafter in order to find a graph mapping from the MINP model into the tree based representation. For each source s, First, drop all vertices and edges corresponding to the fact that, for each vertice e, there is outgoing path from e to a flow which is not incoming from s. Then, just transform the graph into its line graph.

6.2 Traceroute-based model

As traceroute-based representations of a network topology is not based on a metric, no operation can be rigourously defined between those two representation, except by statistical inference, where a similarity between two nodes of traceroute-based model can be infered by assertions upon their output/input degree. However, as it target a different need, we will no longer argue here about this kind of probes.

6.3 Interference graphs

An important point about interference graphs (a routed graph representing mainly commmon subpath of flows) [15] is that it identifies narrow links between subset and flow and infer a total order along a routing path between them. As the example given in figure 5 (1), MINP does not allow such total orders. So, we can trivially state that the MINP model does not give as much information as an interference graph does and hence that there is no bijective operations between both representations.

However, as each narrow link between set of flows can be represented in both models, there is a graph operation from the routing based interference graph into the MINP. Moreover, this projection only removes edges that represent interactions that are useless when using end-to-end congestion protocols. Namely, those edges represent the total order between common subpaths of a routing path. So, we can consider that MINP is a simplification of the routed graph model which preserves only the significant information.

7. CONCLUSION AND DISCUSSION

The problem of modeling network performances of an unknown platform upon which actors transfer bulk data in a many-to-many paradigm and on a coordinated scheme has arose from the offspring of network intensive grid applications. So, infering, modeling and representing kwnoledge about performances of a network is a challenging new problem.

In this article, we presented a new model for representing such data, and enlighten the relationship between previous models and our. We have also demonstrated that the use of such a model as a target for reconstruction does provide a way to have an *a-priori* easier inverse problem that the usual ones, as it is well-defined. It is an important shift in metrology for the grid because even the way we represent the network topology when dealing with classic distributed applications has to be revisited.

Our ongoing work is to build a prototype of a tool that can reconstruct and depict such a kind of knowledge, in order to bring more precise model to optimization processes.

8. REFERENCES

[1] Egee project, 2007. http://www.eu-egee.org/.

[2] Fts : File transfer service, 2007. https://twiki.cern.ch/twiki/bin/view/EGEE/FTS.

[3] Geant website, 2007. http://www.geant.net/.

[4] glite project, 2007. http://glite.web.cern.ch/glite/.

[5] Teragrid project, 2007. http://www.teragrid.org/.

[6] W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link. The globus striped gridftp framework and server. In *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 54, Washington, DC, USA, 2005. IEEE Computer Society.

[7] D. Arifler. *Network Tomography Based on Flow Level Measurements*. PhD thesis, University of Texas at Austin, USA, 2004.

[8] A. Bestavros, J. Byers, and K. Harfoush. Inference and labeling of metric-induced network topologies. Technical Report BUCS-TR-2001-010, Boston University, Computer Science Department, June 2001.

[9] L. Bobelin and T. Muntean. Multiple sources, multiple destinations metric induced network topology discovery: a graph theory approach. In *IEEE 2nd International Conference on Intelligent Computer Communication and Processing (ICCP 2006).*

[10] T. Bu, N. Duffield, F. Presti, and D. Towsley. Network tomography on general topologies. T. Bu, N.G. Duffield, F. Lo Presti, and D. Towsley. Network Tomography on General Topologies. UMass CMPSCI Technique Report.

[11] K. Butler, P. McDaniel, and W. Aiello. Optimizing bgp security by exploiting path stability. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 298–310, New York, NY, USA, 2006. ACM Press.

[12] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: Recent developments. *Statistical Science*, 19, no. 3.

[13] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150:1–??, 2001.

[14] F. Kelly. Fairness and stability of end-to-end congestion control. *European Journal of Control 9*, pages 159–176, 2003.

[15] A. Legrand, F. Mazoit, and M. Quinson. An application-level network mapper. Technical Report 2002-09, LIP, feb 2002.

[16] C. Logg, L. Cottrell, and J. Navratil. Experiences in traceroute and available bandwidth change analysis. Presented at SIGCOMM 2004 Workshops, Portland, Oregon, 30 Aug - 3 Sep 2004.

[17] B. B. Lowekamp, N. Miller, R. Karrer, T. Gross, and P. Steenkiste. Design, implementation, and evaluation of the Remos network monitoring system. *Journal of Grid Computing*, 1(1):75–93, 2003.

[18] V. Padmanabhan and L. Qiu. Network tomography using passive end-to-end measurements, 2002.

[19] M. Rabbat, R. D. Nowak, and M. Coates. Multiple source, multiple destination network tomography. In *INFOCOM*, 2004.

[20] Y. Tsang, M. C. Yildiz, P. Barford, and R. D. Nowak. Network radar: tomography from round trip time measurements. In *Internet Measurement Conference*, pages 175–180, 2004.

[21] R. Wolski, N. T. Spring, and J. Hayes. The network weather service: a distributed resource performance forecasting service for metacomputing. *Future*

Generation Computer Systems, 15(5–6):757–768, 1999.

[22] Y.Vardi. Network tomography : estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91:365–377, 1996.