Conditions that impact the complexity of QoS routing

Kuipers, F.A.; Van Mieghem, P.F.A.

DOI
10.1109/tnet.2005.852882

Publication date
2005

Document Version
Accepted author manuscript

Published in
IEEE - ACM Transactions on Networking

Citation (APA)
Kuipers, F. A., & Van Mieghem, P. F. A. (2005). Conditions that impact the complexity of QoS routing. *IEEE - ACM Transactions on Networking*, *13*(4), 717-730. https://doi.org/10.1109/tnet.2005.852882

Conditions that Impact the Complexity of QoS Routing

Fernando A. Kuipers and Piet Van Mieghem

***Abstract*— Finding a path in a network based on multiple constraints (the MCP problem) is often considered an integral part of QoS routing. QoS routing with constraints on multiple additive measures has been proven to be NP-complete. This proof has dramatically influenced the research community, resulting into the common belief that exact QoS routing is intractable in practice. Hence, hardly any exact algorithms were proposed for this problem. However, to our knowledge, no one has ever examined which "worst-cases" lead to intractability. In fact, the MCP problem is not strong NP-complete, suggesting that in practice an exact QoS routing algorithm may work in polynomial time, making guaranteed QoS routing possible. The goal of this paper is to argue that in practice QoS routing may be tractable. We will provide properties, an approximate analysis, and simulation results to indicate that NP-completeness hinges on four conditions, namely (1) the topology, (2) the granularity of link weights, (3) the correlation between link weights, and (4) the constraints. We expect that, in practice, these conditions are unlikely to occur simultaneously and therefore believe that exact QoS routing is tractable in practice.**

I. Introduction

There is an increasing demand for using real-time multimedia applications over the Internet. In order for these applications to work properly, Quality of Service (QoS) measures like bandwidth, delay, jitter, packet loss, etc., need to be controlled. Currently, the Internet cannot guarantee that the QoS requirements of applications will be satisfied. This has triggered the research community to (en masse) investigate the QoS problem, resulting in proposals for QoS-based frameworks (e.g., IntServ, DiffServ, constraint-based MPLS), QoS routing protocols (e.g., Q-OSPF, PNNI), and many QoS routing algorithms (see [18]).

Routing in general consists of two entities, namely the routing protocol and the routing algorithm. The routing protocol has the task of capturing the state of the network and its available network resources and disseminating this information throughout the network. The routing algorithm uses this information to compute shortest paths. Best-effort routing performs these tasks based on a single measure, usually hopcount. QoS routing, however, must take into account multiple QoS measures and requirements. In this paper, we assume that the network-state information is temporarily static and that it has been distributed throughout the network and is accurately maintained at each node using QoS routing protocols. Once a node acquires the network-state information, it performs the second task in QoS routing, namely computing paths given multiple QoS constraints, also known as the multi-constrained path (MCP) problem. In this paper, we evaluate the complexity of exactly solving the MCP problem. Before giving the formal definition of the MCP problem, let us first describe the notation that is used.

Let $G(N,E)$ denote a network topology, where N is the set of nodes and E is the set of links. With a slight abuse of notation, we also use N and E to denote the number of nodes and the number of links, respectively. The number of QoS measures is denoted by m. Each link is characterized by an m-dimensional link weight vector, consisting of m non-negative QoS weights ($w_i(u,v)$, $i=1,...,m$, $(u,v)\in E$) as components. The QoS measure of a path can either be additive, multiplicative, or min/max. In the case of additive measures (e.g., delay, jitter), the path weight of that measure equals the sum of the QoS weights of the links defining the path. Multiplicative measures (e.g., 1 - packet loss probability) can be transformed into additive weights by using the logarithm. The path weight of min(max) QoS measures (e.g., available bandwidth and policy flags) refers to the minimum(maximum) of the QoS weights along the path. The QoS constraints of an application are expressed in the m-dimensional vector $\vec{L}$. Constraints on min(max) QoS measures can easily be treated by omitting all links (and possibly disconnected nodes), which do not satisfy the requested QoS constraint. In contrast, constraints on additive QoS measures cause more difficulties. Therefore, for our study on complexity, we assume all QoS measures to be additive.

Definition 1: Multi-Constrained Path (MCP) problem. Consider a network $G(N,E)$. Each link $(u,v)\in E$ is specified by m additive QoS weights $w_i(u,v)\geq 0$, $i=1,...,m$. Given m constraints L_i, $i=1,...,m$, the problem is to find a path P from a source node s to a destination node d such that

$$w_i(P)\overset{def}{=}\sum_{(u,v)\in P} w_i(u,v)\leq L_i \text{ for } i=1,...,m$$

There may exist multiple different paths in the graph $G(N,E)$ that satisfy all the constraints. Such paths are said to be feasible. According to Definition 1, any of these paths is a solution to the MCP problem. However, it might be desirable to retrieve the optimal path, according to some criterion, within the constraints. This more difficult problem is known as the *Multi-Constrained Optimal Path* (MCOP) problem.

The rest of this paper is organized as follows. Section II presents an overview of related work. Section III analyzes the worst-case NP complexity of the MCP problem. The proof that the MCP problem is NP-complete strongly depends on the size of the link weights and the level of correlation between those link weights. Section IV evaluates, mathematically and by simulation, the impact of correlation on the complexity of solving the MCP problem. Section V discusses the impact of the constraint values on the complexity and introduces the concept of phase transitions in the MCP problem. Finally, in Section VI, we will present our conclusions.

II. Related Work

The MCP problem is an NP-complete problem. Garey and Johnson [10] were the first to list the MCP problem with $m = 2$ as being NP-complete, but they did not provide a proof. Wang and Crowcroft have provided this proof for $m \geq 2$ in [30] and [31], which basically consisted in reducing the MCP problem for $m = 2$ to an instance of the *partition* problem, a well-known NP-complete problem [10]. The effect of this proof has been tremendous, because it suggests that the MCP problem is intractable, in which case heuristics should be used. Many simulations performed in [6], [7], [17], [19], [26], [28][1] suggest that exact QoS routing may not be intractable in practice. There are certain NP-complete problems, such as *partition*, which are considered by many practitioners to be tractable in practice. The reason for this is that, although no algorithms for solving them in time bounded by a polynomial in the input length (e.g., N, E) are known, there exist algorithms which solve those problems in time bounded by a polynomial in the input length and the magnitude of the largest number (e.g., largest QoS weight) in the given problem instance [11]. Such algorithms are called pseudo-polynomial-time algorithms. NP-complete problems for which no exact pseudo-polynomial-time algorithm exists, are called NP-complete in the strong sense. In the case of the *partition* problem, the NP-completeness strongly depends on the fact that arbitrarily large numbers are allowed. If any upper bound was imposed on these numbers in advance, even a bound which is a polynomial function of the input length, there would exist a polynomial-time algorithm for solving this (restricted) problem [11].

David Pisinger [25] has evaluated Knapsack problems, which are NP-complete problems (proved via reduction to the *partition* problem), and found that in practice these problems are tractable. For many more NP-complete problems, typical cases are "easy" to solve. A study of the phenomenon that typical cases are "easy," was performed by Cheeseman et al. [4], who introduced the concept of phase transitions in NP-complete problems. According to Cheeseman et al., NP-complete problems which are very under-constrained are soluble and it is usually easy to find one of the many solutions. NP-complete problems which are very over-constrained are insoluble. In the phase transition in between, as illustrated in Figure 1, problems are "critically constrained" and it is typically very hard to determine if they are soluble or insoluble [12]. For a more formal discussion of phase transitions we refer to [8]. Cheeseman et al. have conjectured that all NP-complete problems have at least one order parameter and that the hard to solve problems are around a critical value of this order parameter. Although this conjecture does not hold for all NP-complete problems [15], there seems to be a connection between complexity and phase transitions. The lack of a phase transition seems to have significant computational implications: such problems are either computationally tractable, or well-predicted by a single, trivial algorithm [15]. This alleged connection between complexity and phase transitions motivated us to investigate phase transitions in the MCP problem. Monasson et al. [23], report an analytic solution and experimental investigation of the phase transition in K-satisfiability (the first problem shown to be NP-complete). Gent and Walsh [12] show that phase transitions occur in the *partition* problem.

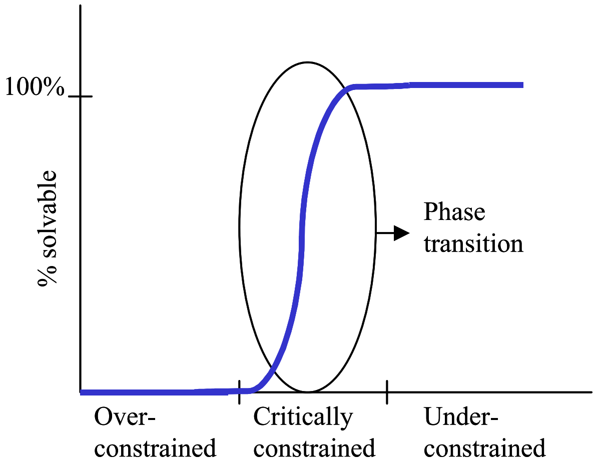

Fig. 1. The solubility of an NP-complete problem around a phase transition.

Levin [20] advocated a different study of NP-complete problems by introducing the concept of average-case complexity. He indicated that some NP-complete problems are "easy on average," while other (average-case NP-complete) problems may not be.

There exists also some work in the literature revealing important properties of the MCP problem. We will mention three of those properties, that all strengthen our belief that in practice exact QoS routing is tractable. First of all, the MCP problem is not strong NP-complete. Secondly, if all, but one, measures take bounded integer values, then the MCP problem is solvable in polynomial time [5]. Finally, if some specific dependencies exist between QoS measures, exact QoS routing can be performed in polynomial time [22]. The goal of our work is to provide more evidence that suggests the tractability of exact QoS routing, in practice.

III. Worst-case Complexity Analysis

In this section we will analyze the worst-case complexity of the MCP problem for $m = 2$. First, we will rewrite the proof that the MCP problem for $m = 2$ is NP-complete [30], [31], and refer to it as the NP-proof.

Theorem 1: The MCP problem is NP-complete.

Proof: First the proof for $m = 2$ is presented. Given a chain topology with $n + 1$ nodes and $2n$ links, each with a two-component weight vector $\vec{w}$ as depicted in Figure 2, and a set of numbers $a_i \in A$, $0 \leq a_i \leq S$, for $i = 1, ..., n$, where $S = \sum_{i=1}^{n} a_i$. The constraints are chosen as follows:

[1] Our paper [28] can be seen as a more simulative companion to this paper.

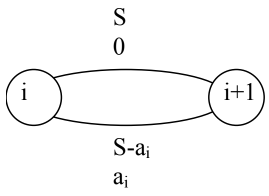

Fig. 2. The assignment of link weights to the links between nodes i and $i+1$, in a chain topology.

$L_1 = nS - \frac{S}{2}$, and $L_2 = \frac{S}{2}$. To solve the MCP problem, we need to find a path from node 1 to node $n+1$, that obeys the constraints. Since, for all link weight vectors, the sum of the components equals S, we have that $w_1(P) + w_2(P) = nS$. Accordingly, a solution satisfying the constraints is only found if $w_1(P) = nS - \frac{S}{2}$ and $w_2(P) = \frac{S}{2}$. The problem has now become an instance of the well-known NP-complete *partition* problem [10] and can only be solved by finding the set $A' \subseteq A$, for which $\sum_{a_i \in A'} a_i = \frac{S}{2}$. A feasible path exists if the set A' exists, in which case it is retrieved by choosing the lower link if $a_i \in A'$ and the upper link if $a_i \notin A'$.

We have proved that the MCP problem with $m = 2$ is NP-complete. The proof that MCP in general is NP-complete inductively follows. We assume that the MCP problem with m measures is NP-complete. If we extend the number of measures with 1 to $m+1$ and choose $L_{m+1} = \sum_{(u,v)\in E} w_{m+1}(u,v)$, then all paths between source and destination obey this constraint. The MCP problem with $m+1$ measures is then only solved if the MCP problem with m measures is solved. This concludes the proof. ■

If the constraints are chosen such that only one feasible path exists, then the MCP problem is equal to the MCOP problem, and hence the MCOP problem is also NP-complete.

Corollary 2: The MCP problem is not NP-complete in the strong sense.

Proof: The MCP problem is not strong NP-complete, because there exist pseudo-polynomial algorithms that exactly solve this problem (e.g., see [16], [21]). ■

The proof that a problem is NP-complete or not is entirely based on a worst-case argument. A problem is called polynomially solvable if it can be solved by an algorithm that terminates after a number of steps (instructions) that is bounded by a polynomial in the input length. A problem is called NP-complete if there is even one instance that is not polynomially solvable (unless P = NP). It may occur that in some instances the running time required to solve the MCP problem is polynomial. We call those polynomially solvable instances tractable and we will use the term intractable when instances require a non-polynomial running time (i.e., they are not polynomially solvable).

We desire to distinguish the instances of the MCP problem that are tractable and those that are intractable. If we look at the graph on which the MCP problem should be solved, we could delineate the class of polynomially solvable graphs, i.e. the class of graphs in which the number of paths between two nodes increases as a polynomial function of N (e.g., tree-, circle-, and star-topologies). This class of graphs is most likely very small and therefore most graphs potentially can lead to intractability. Fortunately, the underlying topology *alone* is not sufficient to lead to intractability: we also need a specific link weight structure. For instance, if all link weights are assigned the value 1, then the MCP problem is polynomially solvable regardless of the underlying topology. We will proceed by defining a link weight structure that leads to intractability in the chain topology. We will use the chain topology as depicted in Figure 3 and ascertain that all paths from source s to destination d are non-dominated.

Definition 2: Dominance. A path P *dominates* a path P' if $w_i(P) \leq w_i(P')$ for all link weight components i except for at least one j for which $w_j(P') < w_j(P)$. A path P is called *non-dominated* if there[2] does not exist a path P' for which $w_i(P') \leq w_i(P)$ for all link weight components i except for at least one j for which $w_j(P') < w_j(P)$.

In general, there are two important properties that can reduce the search space when solving the MCP problem without losing exactness, namely non-dominance and the constraints themselves. If a sub-path P from source node s to node i exceeds one or more constraints, it can never become a feasible path[3], because the path weight vector from i to destination node d consists of non-negative weights. Similarly, if for two paths P_1, P_2 from s to i it holds that P_1 dominates P_2, then all weights of P_1 are smaller than (or equal to) those of P_2 and hence we can omit P_2 from our search space and continue with P_1 [7], because the paths extended from P_2 will always be dominated by the paths extended from P_1. According to [28], the maximum number of non-dominated paths that obey the constraints is upper bounded by $\frac{\prod_{i=1}^{m} L_i}{\max_j(L_j)}$, where the constraints L_i are expressed as an integer number of the smallest granularity. This value provides a worst-case estimate of the size of our search space. According to Levin [20] some NP-complete problems are "easy on average," while other (average-case NP-complete) problems may not be. The average-case complexity therefore also gives some indication whether an NP-complete problem could be tractable in practice. In [28] we have shown that if the path weights are independently distributed in the solution space, then the MCP problem can be solved in polynomial time on average.

Without loss of generality, we assume that the link weights in Figure 3 are chosen such that $a_i > c_i$ and $b_i < d_i$, for $i = 1$, ..., N ($c_i > a_i$ and $d_i < b_i$ would also have been possible). It can be verified that if $a_i \geq c_i$ and $b_i \geq d_i$ or $c_i \geq a_i$ and $d_i \geq b_i$ were allowed, this would lead to dominance.

[2] If there are two or more different paths between the same pair of nodes that have an identical weight vector, only one of these paths suffices. In the sequel, we will therefore assume one path out of the set of equal-weight vector paths as being non-dominated and regard the others as dominated paths.

[3] This also holds for the lower-bound estimation of the end-to-end path weight vector $\vec{w}(P) + \vec{b}$, where $\vec{b}$ denotes a lower-bounds vector consisting of the m one-dimensional shortest path weights from i to d.

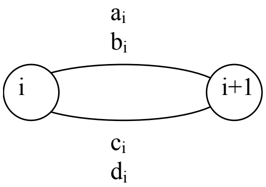

Fig. 3. A chain topology with two QoS weights per link and N nodes in total.

Property 1: If, in the chain topology in Figure 3, it holds that

$$\left\{ \begin{array}{l} a_i - c_i > \sum_{j=0}^{i-1} (a_j - c_j) \\ b_i - d_i < \sum_{j=0}^{i-1} (b_j - d_j) \end{array} \right. \tag{1}$$

for $i = 1, ..., N-1$, where $a_0 = b_0 = c_0 = d_0 = 0$, then all 2^{N-1} paths from node 1 to node N are non-dominated.

Proof: We will give a proof by induction.
$i = 1$: There are two paths from node 1 to node 2, namely $P_1(1 \to 2) = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$ and $P_2(1 \to 2) = \begin{pmatrix} c_1 \\ d_1 \end{pmatrix}$. According to formula (1): $a_1 > c_1$ and $b_1 < d_1$, which shows that both paths from node 1 to node 2 are non-dominated.
The inductive step is to assume the correctness of formula (1) for a certain i. It remains to prove that it also holds for $i+1$. There are 2^{i-1} paths from node 1 to i. From i there are two possible links to i+1, resulting in a total of 2^i paths from node 1 to node i+1. 2^{i-1} paths will follow the upper link from i to i+1, while the remaining 2^{i-1} paths will follow the lower link. Since all paths at i are non-dominated (inductive assumption), the paths following the upper link are also non-dominated, because the same vector is added to each of the path vectors. The same property applies to the paths that follow the lower link. It remains to show that if (1) holds, then the paths following the upper link and the paths following the lower link do not dominate each other.
If (1) is satisfied, then all paths following the upper link possess a first path weight larger than the first weights of the paths following the lower link. Similarly, the paths following the lower link have a second weight, which is larger than the second weights of the paths following the upper link. Hence, the paths following different links are non-dominated. ■

The *partition* problem is NP-complete, because the values involved in an instance of the *partition* problem may be arbitrarily large (or have an infinite granularity). The same phenomenon is observed in formula (1), where the difference between a_i and c_i (and correspondingly d_i and b_i) grows exponentially:

$$\begin{aligned} a_{i+1} - c_{i+1} &> \sum_{j=0}^{i} (a_j - c_j) = (a_i - c_i) + \sum_{j=0}^{i-1} (a_j - c_j) \\ &> 2\sum_{j=0}^{i-1} (a_j - c_j) > ... > 2^{i-1}(a_1 - c_1) \end{aligned}$$

If a_i in the NP-proof are not chosen according to formula (1), but if they take bounded integer values, then the problem becomes polynomially solvable.

A second important phenomenon that we observe from formula (1) is that the link weights display a perfect *negative correlation*. If the link weights would have had a positive correlation, then if $a_i > c_i$ most likely also $b_i > d_i$, leading to dominance.

Lemma 3: Property 1 is a sufficient but also necessary condition for all paths in the chain topology to be non-dominated.

Proof: We need to show that if formula (1) does not hold, then at least one path from node 1 to node $i+1$ is dominated. If (1) does not hold, we have

$$\left\{ \begin{array}{l} \sum_{j=0}^{i-1} c_j + a_i \leq \sum_{j=0}^{i-1} a_j + c_i \\ \sum_{j=0}^{i-1} d_j + b_i \geq \sum_{j=0}^{i-1} b_j + d_i \end{array} \right. \tag{2}$$

or

$$\left\{ \begin{array}{l} \sum_{j=0}^{i-1} c_j + a_i > \sum_{j=0}^{i-1} a_j + c_i \\ \sum_{j=0}^{i-1} d_j + b_i \geq \sum_{j=0}^{i-1} b_j + d_i \end{array} \right. \tag{3}$$

or

$$\left\{ \begin{array}{l} \sum_{j=0}^{i-1} c_j + a_i \leq \sum_{j=0}^{i-1} a_j + c_i \\ \sum_{j=0}^{i-1} d_j + b_i < \sum_{j=0}^{i-1} b_j + d_i \end{array} \right. \tag{4}$$

We have written these formulas slightly differently from (1) to illustrate that they correspond to two paths, namely the path that followed all the lower links up to node i and took the upper link from node i to node $i+1$ and the path that took all the upper links towards node i and the lower link from node i to node i+1. Formula (2), without the equalities, is exactly the same as (1), but a is called c and b is called d. If the equality sign applies, then the path that followed all the lower links up to node i and took the upper link from node i to node $i+1$ is the same as the path that took all the upper links towards node i and the lower link from node i to node $i+1$. According to Definition 1 only one of these two paths is non-dominated. When formula (3) applies, the path that followed all the lower links up to node i and took the upper link from node i to node $i+1$ is dominated by (or dominates in the case of formula (4)) the path that took all the upper links towards node i and the lower link from node i to node $i+1$. ■

Property 1 and Lemma 3 seem very restrictive, because they are solely based on the chain topology and we require all paths to be non-dominated. If only a subset of all paths (that increases non-polynomially in N) were non-dominated, then the problem would still be intractable. However, if only such a subset of all paths would be non-dominated, then property 1 must hold for a subset of the links/subpaths. Otherwise, all link weights would be bounded and the problem would be polynomially solvable.

Also the chain topology can be put into perspective. Links in the chain topology can be seen as sub-paths.

Lemma 4: If there are more than two links (all with two weights) between two nodes in the chain topology, formula (1) should hold for all possible pairs of links, in order for all paths from node 1 to node N to be non-dominated.

In practice we do not expect links/sub-paths to satisfy formula (1). If formula (1) is not satisfied, Lemma 4 suggests that when there are many sub-paths to a node, the probability that all these paths are non-dominated decreases and consequently also the search space decreases.

At the beginning of this section we mentioned that there are two important properties to reduce the search space, namely non-dominance and the values of the constraints. If the constraints are chosen very large, then it will be easy to find a path that obeys these constraints. On the other hand, if the constraints are very strict, there may not be a path available that can obey these constraints. For the chain topology, besides formula (1), the constraints must lie in the range:

$$\begin{cases} \sum_{j=0}^{N-1} c_j \leq L_1 \leq \sum_{j=0}^{N-1} a_j \\ \sum_{j=0}^{N-1} d_j \geq L_2 \geq \sum_{j=0}^{N-1} b_j \end{cases}$$

to induce intractability (i.e., then the MCP problem reduces to the *partition* problem as illustrated in the NP-proof). Since $c_i < a_i$, the shortest path for measure 1 from node 1 to node N, equals $\sum_{j=0}^{N-1} c_j$. If $L_1 < \sum_{j=0}^{N-1} c_j$, then no feasible path exists. If $L_1 > \sum_{j=0}^{N-1} a_j$, then all possible (loop-free) paths can obey this constraint. The same reasoning applies to L_2 and is further motivated in Section V.

In this section we have used the chain topology to create an intractable instance of the MCP problem. This instance provided us with some hints on the underlying causes of intractability. In Section IV we will further evaluate the impact of correlation on the complexity of QoS routing.

IV. The impact of link correlation on complexity

Section III hinted at a connection between link correlation and complexity. In this section we will discuss the impact of link correlation on the complexity of QoS routing by giving some properties and presenting simulation results.

A. Theory

Ma and Steenkiste [22] have shown that when specific dependencies (correlation) exist between QoS measures, due to Weighted Fair Queueing scheduling, QoS routing can be performed in polynomial time. However, it is a misconception that if all QoS measures are a function of a common measure, then by just minimizing this common measure, we will have minimized all measures. We will illustrate that this is not always the case and provide some conditions when this statement holds. We will denote by $f(.)$ a convex function, by $\varphi(.)$ a concave function, by $\psi(.)$ a linear function, and by $g(.)$ a monotone increasing function.

Consider Figure 4: if $f(x)$ is a convex function, then the shortest path based on x is not necessarily the shortest path for $f(x)$. For example, suppose that $f(x) = e^x$ and $x_1 = 2$, $x_2 = 2$, $x_3 = 3$. Then the shortest path from a to c is $a - c$ for x, but $a - b - c$ for $f(x)$.

Likewise, if $\varphi(x)$ is a concave function, the shortest path based on x is not necessarily the shortest path for $\varphi(x)$, e.g. $\varphi(x) = \log(x)$ and $x_1 = 1.2$, $x_2 = 1.2$, $x_3 = 2.2$. Then the shortest path from a to c is $a - c$ for x, but $a - b - c$ for $\varphi(x)$.

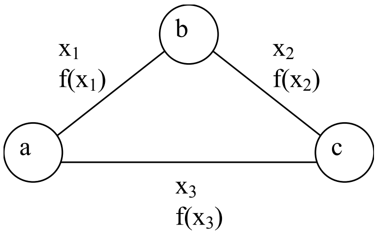

Fig. 4. An example topology.

In case of a linear function $\psi(x) = ax + b$, then the shortest path based on x will also be the shortest path for $\psi(x)$ if $a > 0$ and $b = 0$.

In the rest of this subsection we consider graphs, for which all link weights are a function of a common link weight. Each link i has a weight vector $\vec{w} = \begin{bmatrix} f_1(x_i) \\ \vdots \\ f_m(x_i) \end{bmatrix}$, where x_i is the common link parameter (links may have different x_i and different f_j). In the sequel we will refer to this graph as G_w. We also introduce the graph G_x, which is identical in structure to G_w, but for which the links only have weight x_i.

Let P_x be the shortest path from source s to destination d in G_x, then

$$w(P_x) = \sum_{i \in P_x} x_i \leq w(P) = \sum_{i \in P} x_i$$

where P is any other path ($\neq P_x$) from s to d in G_x. Let $\varphi(x)$ be a concave function, then

$$\varphi(\frac{1}{h}\sum_{i=1}^{h} x_i) \geq \frac{1}{h}\sum_{i=1}^{h} \varphi(x_i)$$

where h is the hopcount of a path P.

Property 4: If the weight vector of a link, $\vec{w} = \begin{bmatrix} \varphi_1(x_i) \\ \vdots \\ \varphi_m(x_i) \end{bmatrix}$ with $\varphi_j(x_i)$ concave functions, is a function of a single parameter x_i and if P is the shortest path from s to d in G_x with length $X = \sum_{i=1}^{h} x_i$ and hopcount h, then P in G_w satisfies the constraint vector $\vec{L}$ if

$$X \leq h\varphi_j^{-1}\left(\frac{L_j}{h}\right), \quad 1 \leq j \leq m \tag{5}$$

Proof: The constraints are satisfied if $\sum_{i \in P} \varphi_j(x_i) \leq L_j$. Since φ_j are concave functions:

$$\sum_{i=1}^{h} \varphi_j(x_i) \leq h\varphi_j\left(\frac{1}{h}\sum_{i=1}^{h} x_i\right) \leq L_j$$

or,

$$\varphi_j\left(\frac{1}{h}\sum_{i=1}^{h} x_i\right) \leq \frac{L_j}{h}$$

Hence,

$$X = \sum_{i=1}^{h} x_i \leq h\varphi_j^{-1}\left(\frac{L_j}{h}\right)$$

■

Note that although P is the shortest path in G_x, this does not mean that P is also the shortest path in G_w (there may be another path P' for which $\sum_{i\in P'} \varphi(x_i) < \sum_{i\in P} \varphi(x_i)$). Equation (5) is a sufficient, but not a necessary condition, because there may be a path that does not obey (5), but still satisfies the constraints.

Property 5: If the weight vector of a link, $\vec{w} = \begin{bmatrix} f_1(x_i) \\ \vdots \\ f_m(x_i) \end{bmatrix}$ with $f_j(x)$ convex functions, is a function of a single parameter x_i and if P is the shortest path from s to d in G_x with length $X = \sum_{i=1}^{h} x_i$ and hopcount h, then P (and therefore all paths) violates the constraints in G_w if

$$X > hf_j^{-1}\left(\frac{L_j}{h}\right) \tag{6}$$

for at least one j.

Proof: By convexity,

$$hf_j\left(\frac{1}{h}\sum_{i=1}^{h} x_i\right) = hf_j\left(\frac{X}{h}\right) \leq \sum_{i=1}^{h} f_j(x_i)$$

The j-th constraint is violated if $\sum_{i=1}^{h} f_j(x_i) > L_j$, which is the case if $hf_j\left(\frac{X}{h}\right) > L_j$, which is equivalent to (6). ■

Property 6: If the weight vector of a link $\vec{w} = \begin{bmatrix} g_1(x_i) \\ \vdots \\ g_m(x_i) \end{bmatrix}$ with $g_j(x_i)$ monotone increasing and P is the shortest minimum-hop path from s to d in G_x and $x_i \leq x_i'$, where x_i' is the i-th ordered common link weight of another path P' from s to d in G_x, then P is also the shortest path in G_w.

Proof: The property is a corollary from Theorem 107 from [13]: Suppose that the sets (a) and (a') are arranged in descending order of magnitude. Then a necessary and sufficient condition that $g(a_1') + ... + g(a_n') \leq g(a_1) + ... + g(a_n)$ should be true for all continuous and increasing g is that $a_v' \leq a_v$ $(v = 1, 2, ..., n)$. ■

B. Simulation results

In this section we will evaluate the complexity of QoS routing through simulations. We will present simulation results for several classes of graphs, namely the class of random graphs, the class of two-dimensional lattices, the class of power-law graphs, and the chain topology. The class of random graphs is of the type $G_p(N)$ [3], where p is the expected link-density[4] ($p = 0.2$). We only consider square two-dimensional lattices (an example can be found in Figure 14). For the class of Internet-like power-law graphs [9], we have chosen the power $\alpha = -2.4$ in the nodal degree distribution $\Pr[d = k] \sim k^{\alpha}$. The chain topologies were of a triangular shape (as depicted in Figure 4). We have simulated with three different distributions for the $m = 2$ link weights, namely the uniform, exponential, and Gaussian distributions. We only present the simulation results for correlated uniformly distributed link weights $\in [0, 1]$ with correlation coefficient[5] ρ [24], because they led to a higher complexity than the exponential and Gaussian distributions. We have previously also simulated with $m > 2$. The results are scattered over several papers (e.g., [28] and [19]). The results (assuming independence among the m weights) do not show a more than linear increase in complexity as a function of m. We have confined to $m = 2$ for the correlation study, because for $m = 2$ the correlation coefficient can span the entire range $[-1, 1]$, while if m grows, the links cannot all be correlated with $\rho = -1$ and the "mutual" correlation range tends to $[0, 1]$.

All simulations consisted of generating 10^5 different graphs and in each graph a path has been computed via the SAMCRA algorithm [29]. SAMCRA incorporates four concepts: (1) a nonlinear measure for the path length $l(P) = \max_{j=1,...m}\left(\frac{w_j(P)}{L_j}\right)$, (2) a k-shortest path approach[6] to examine multiple subpaths per node, (3) the principle of non-dominated paths to reduce the search space and (4) the "look-ahead" concept. The look-ahead concept precomputes one or multiple shortest path trees rooted at the destination and then uses this information to reduce the search space. In TAMCRA [7], the polynomial-time predecessor of SAMCRA, k is fixed (giving its polynomial complexity), but with SAMCRA this k can grow exponentially in the worst case. SAMCRA does not only exactly solve the MCP problem, but also exactly solves the MCOP problem by finding the optimal path within the constraints. Since the MCOP problem is more difficult than the MCP problem, the simulation results presented here should be interpreted as an upper bound. We have simulated a worst-case scenario by choosing the constraints so large that all paths can satisfy the constraints. Therefore, SAMCRA must search in the largest search space possible (all non-dominated paths between the source and destination), for the optimal path. If SAMCRA was only solving the MCP problem, choosing such large constraints would make the MCP problem "easy," because then any path is a solution to the MCP problem. During all simulations, we kept track of the minimum queue-size ($k_{\min}$: the minimum number of paths that needs to stored at a node) needed to find a feasible path. If TAMCRA [7] had used this particular $k_{\min}$ under the same conditions, it would have found the same optimal path as SAMCRA did. If a smaller queue-size had been used, TAMCRA would not have been able to find the optimal path. This minimum queue-size $k_{\min}$ can grow as a factorial in the worst case and presents our measure for the complexity of QoS routing.

As illustrated in Figures 5-7, the results for the class of random graphs, do not display any intractability. We can see

[4] We have used a fixed link density, because according to [27] the dependency on the link density becomes vanishingly small as N grows.

[5] We have verified that the correlation coefficient ρ' of the generated random variables equals the desired ρ.

[6] A k-shortest path algorithm does not stop when the destination has been reached for the first time, but continues until it has been reached through k different paths succeeding each other in length.

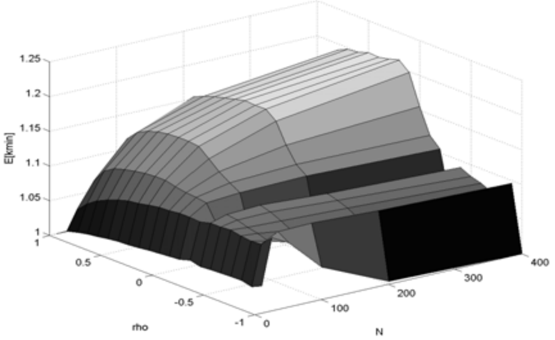

Fig. 5. Expected queue-size for the class $G_p(N)$, with $m = 2$ uniformly distributed correlated link weights, as a function of the number of nodes N and the correlation coefficient ρ.

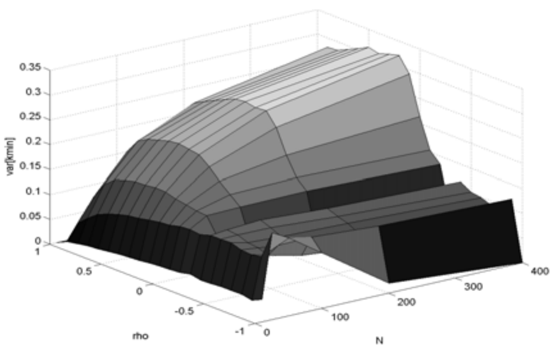

Fig. 6. Variance in queue-size for the class $G_p(N)$, with $m = 2$ uniformly distributed correlated link weights, as a function of the number of nodes N and the correlation coefficient ρ.

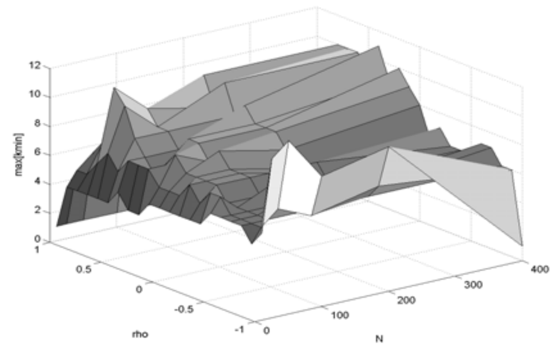

Fig. 7. The maximum observed queue-size in the class $G_p(N)$, with $m = 2$ uniformly distributed correlated link weights, as a function of the number of nodes N and the correlation coefficient ρ.

that a positive correlation leads to a slightly higher $E[k_{\min}]$ than with a negative correlation. This peculiar phenomenon has only been observed in the class of random graphs, with correlated uniformly distributed link weights. An explanation can be found if we look at Figure 8. Figure 8 shows that a positive correlation between the link weights may induce a higher expected hopcount. When the link weights become more positively correlated, the weights become similar, and the problem approaches the $m = 1$ case. Since, the expected hopcount of the m-dimensional shortest paths approaches the minimum hopcount if m grows to infinity [26], the $m = 1$ case is expected to have the largest hopcount. A negative correlation between the link weights also leads to shorter hopcount paths. A low hopcount is possible because there are sufficiently many paths in $G_p(N)$, which can be viewed as a thinning of a complete graph provided $p > \frac{\ln N}{N}$. For negative correlated link weights, a small link weight component is likely accompanied with a large one. For perfect negatively correlated link weight components ($\rho = -1$), SAMCRA's shortest-length path (15) compensates outliers in the link weight components with the result that (one or two) links with weight components close to $\frac{1}{2}$ are selected which leads to the observed minimum-hop paths.

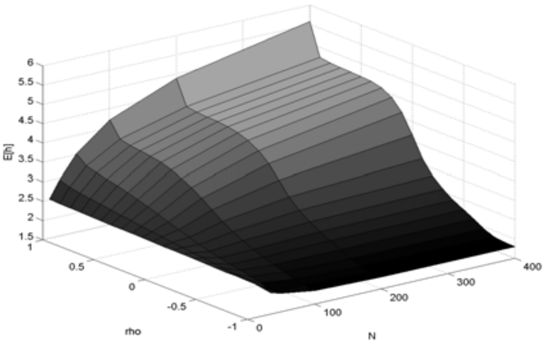

Fig. 8. The expected hopcount for the class $G_p(N)$, with $m = 2$ uniformly distributed correlated link weights, as a function of the number of nodes N and the correlation coefficient ρ.

In general, the more hops we must traverse to find the shortest path, the more (sub)-paths we must evaluate and the more complex the computation becomes. We believe that one of the measures for the "computational complexity" of a class of topologies is the expected (minimum) hopcount of an arbitrary path in that topology. The expected hopcount (for $m = 1$) scales as $O(\log N)$ in a random graph, while as $O(\sqrt{N})$ in a two-dimensional lattice and $O(N)$ in the chain topology. Besides the expected hopcount in a graph, also the number of paths between a source and destination can provide a measure for the "computational complexity" of a class of topologies. The class of random graphs with $p = 0.2$ and N increasing, has an increasing number of paths and an increasing average nodal degree, giving the graph a small diameter (i.e., the source and destination are directly linked or a few hops apart). This can be interpreted from Figure 8. Figure 9 gives the expected queue-size for three different classes of graphs, namely the random graphs ($p = 0.2$), the two-dimensional lattices, and the Internet-like power-law graphs (with power $\alpha = -2.4$). For all three classes of graphs, the source and destination nodes were chosen randomly. Only for the class of two-dimensional lattices "Lattice2," we have chosen the source and destination nodes in opposite corners, to attain the largest minimum hopcount. In the class of random graphs $G_p(N)$, although the number of paths is large, the expected hopcount is small, leading to a small complexity. For the extreme regular class Lattice2 of two-dimensional lattices, the number of paths and the expected hopcount are large, which leads to a large complexity. The class of power-law graphs may be considered, in terms of randomness, to lie between the random graphs and the two-dimensional lattices. The power-law graphs with $\alpha = -2.4$ have a moderate expected hopcount and a small number of paths, and lie, in terms of complexity, closer to the class of random graphs than to the class of two-dimensional lattices. We have also simulated with different link weight distributions, namely Gaussian and exponentially distributed correlated link weights. If we use exponentially distributed correlated link weights, the first weight has a higher probability of being small, than with a uniform distribution. With a uniform distribution, each value for the first weight is equiprobable. Therefore, with exponentially (and also Gaussian) distributed correlated link weights, there is a higher probability that the link weight vectors are similar. For uniformly distributed link weights there is a larger variability, leading to a somewhat worse performance than in the exponential (or Gaussian) case. However, in all cases the expected queue-size in the class of random graphs was close to one, leading to a complexity similar to that of Dijkstra's algorithm. These simulation results therefore suggest that, irrespective of the link weight structure, QoS routing in the class of random graphs (and according to [27] also Waxman graphs) is possible in polynomial time. In contrast, the regularity and large expected hopcount in the class of two-dimensional lattices, may provide ground for intractability. Indeed, we can observe a tendency towards intractability in Figure 10 and true non-polynomial behavior in Figure 11.

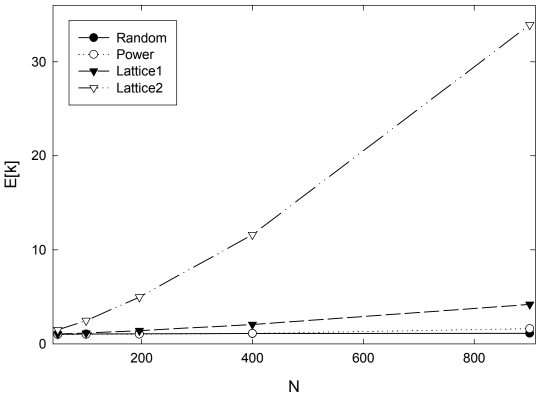

Fig. 9. The expected queue-size for different topology classes as a function of the number of nodes N, with $m = 2$ independent ($\rho = 0$) uniformly distributed link weights.

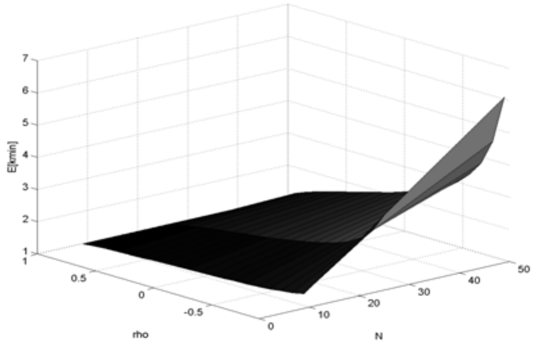

Fig. 10. The expected queue-size in the class of two-dimensional lattices as a function of the number of nodes N and correlation coefficient ρ. The $m = 2$ link weights were uniformly distributed and the source and destination nodes were chosen in opposite corners.

Because the chain topology was used in the proof that the MCP problem is NP-complete, we have also evaluated the performance of SAMCRA in chain topologies. The results are plotted in Figures 11 and 12.

Our simulation results[7] indicate that in the class of two-dimensional lattices and chain topologies, the MCP problem seems tractable for nearly the entire range of correlation coefficient ρ, except for extreme negative values. Recall that the NP-proof is based on an extreme negative link correlation. We doubt that in practice link weights will display such a negative correlation, suggesting that exact QoS routing in

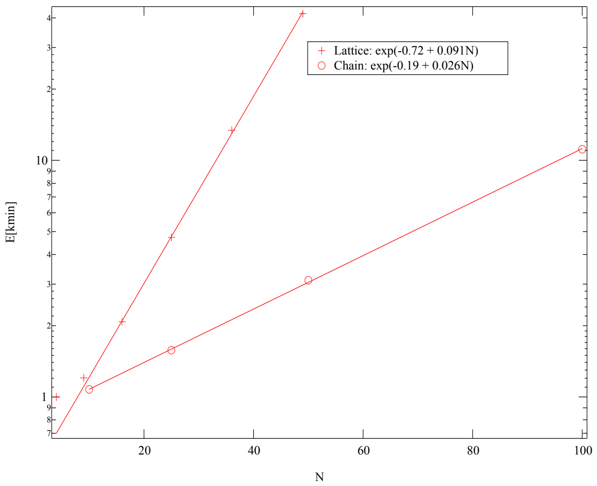

Fig. 11. The expected queue-size (on a logarithmic scale) in the class of two-dimensional lattices and chains, as a function of the number of nodes N, with correlation coefficient $\rho = -1$. The $m = 2$ link weights were uniformly distributed and the source and destination nodes were chosen in a way that the minimum hopcount was largest. We have fitted with exponentials, which perfectly match the results in the simulated range. Simulating with larger N may consume months of CPU time and therefore can only be done by reducing the number of iterations or via parallel procesing.

[7] Recall that the simulation results reflect the complexity of the much more difficult MCOP problem.

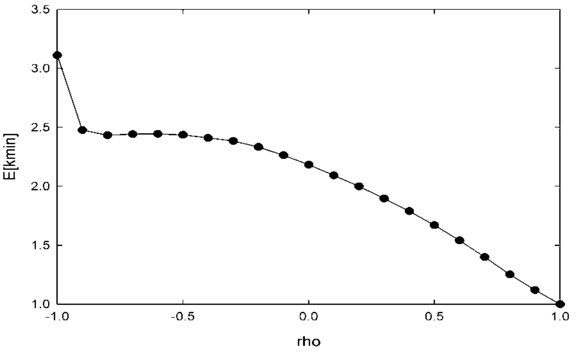

Fig. 12. The expected queue-size in the chain topology, with $m = 2$ correlated uniformly distributed link weights for $N = 50$, as a function of the correlation coefficient ρ.

practice, irrespective of the underlying topology, is possible in polynomial time.

V. The Impact of Constraints on Complexity

In this section we analyze the influence of the constraints on the complexity of the MCP problem. For this purpose, we will initiate an evaluation of a phase transition [4], [14] in the MCP problem.

A. Theory

Property 7: Let $P_{s-d;i}$ denote the one-dimensional shortest path from source s to destination d, for which $w_i(P_{s-d;i}) \leq w_i(P^*)$, $\forall P^*$. Then, the MCP and MCOP problems are not NP-complete when

$$L_i < w_i(P_{s-d;i}) \tag{7}$$

for at least one constraint.

Proof: $P_{s-d;i}$ is the path with the shortest i-th weight $w_i(P_{s-d;i})$. Therefore $w_i(P_{s-d;i})$ is a lower bound on the i-th weight $w_i(P_{s-d})$ that any path P_{s-d} between s and d can attain. Therefore, if for any constraint i it holds that $L_i < w_i(P_{s-d;i})$, then no path P_{s-d} can obey L_i. Since $P_{s-d;i}$ can be found in polynomial time (e.g., via the Dijkstra algorithm), the MCP problem is solvable (i.e., it is verified that no solution exists) in polynomial time if any constraint obeys (7). ■

Property 8: Let $P_{s-d;i}$ denote the one-dimensional shortest path from source s to destination d for which $w_i(P_{s-d;i}) \leq w_i(P^*)$, $\forall P^*$. Then, the MCP problem is not NP-complete when

$$L_i \geq \max_{j=1,\ldots,m} \left(w_i\left(P_{s-d;j}\right)\right) \tag{8}$$

for at least $m-1$ constraints.

Proof: If $L_i \geq \max_{j=1,\ldots,m}\left(w_i\left(P_{s-d;j}\right)\right)$ for all m constraints, then all m one-dimensional shortest paths $P_{s-d;i}$, (for $i = 1, \ldots, m$) obey the constraints. Hence, any path $P_{s-d;i}$ can be chosen as a feasible path.

If $L_i \geq \max_{j=1,\ldots,m}\left(w_i\left(P_{s-d;j}\right)\right)$ for $m-1$ constraints (say $i = 1, \ldots, m-1$) and $L_i < \max_{j=1,\ldots,m}\left(w_i\left(P_{s-d;j}\right)\right)$ for one constraint ($i = m$), then if $L_m \geq w_m(P_{s-d;m})$ path $P_{s-d;m}$ obeys all m constraints. If $L_m < w_m(P_{s-d;m})$, then by property 7 we know that no feasible path exists. Since the paths $P_{s-d;i}$ can be found in polynomial time (e.g., via the Dijkstra algorithm), the MCP problem is solvable in polynomial time if at least $m-1$ constraints obey (8). ■

For $m = 2$, properties 7 and 8 constitute a closed NP-complete range

$$w_i(P_{s-d;i}) < L_i < \max_{j=1,\ldots,m} \left(w_i\left(P_{s-d;j}\right)\right) \tag{9}$$

The MCP problem with $m = 2$ is only NP-complete if both constraints lie in the NP-complete range (9). When the link weights are positively correlated, the NP-complete range (9) will be smaller than when the link weights are negatively correlated. This is illustrated in Figure 13 for $m = 2$. At the cost of increased (polynomial-time) complexity, we can further reduce the NP-complete range by using property[8] 9.

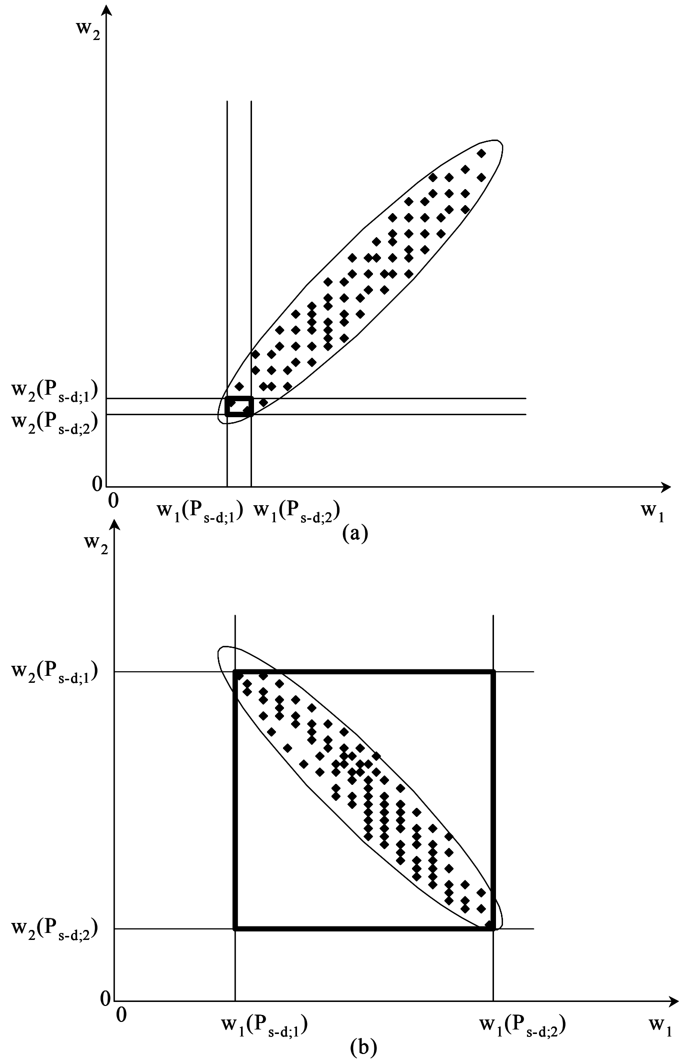

Fig. 13. The constraints range (bold square) for (a) positive correlation and (b) negative correlation. The dots in the figure denote paths in the two-dimensional space ($m = 2$).

[8] We have not programmed property 9 in our simulations.

Property 9: Let P_{s-d} denote the path from source s to destination d for which $\sum_{i=1}^{m} \alpha_i w_i(P_{s-d}) \leq \sum_{i=1}^{m} \alpha_i w_i(P_{s-d}^*)$, $\forall P_{s-d}^*$. Then, if

$$\sum_{i=1}^{m} \alpha_i L_i < \sum_{i=1}^{m} \alpha_i w_i(P_{s-d})$$

where $\alpha_i \geq 0$ with an inequality for at least one i, then there is no feasible path present that can solve the MCP or MCOP problem.

Proof: A proof by contradiction. Assume that P_{s-d} denotes the path from source s to destination d for which $\sum_{i=1}^{m} \alpha_i w_i(P_{s-d}) \leq \sum_{i=1}^{m} \alpha_i w_i(P_{s-d}^*)$, $\forall P_{s-d}^*$ and that $\sum_{i=1}^{m} \alpha_i L_i < \sum_{i=1}^{m} \alpha_i w_i(P_{s-d})$. If a path P_{s-d}^* would exist that obeys the constraints, then $\sum_{i=1}^{m} \alpha_i w_i(P_{s-d}^*) \leq \sum_{i=1}^{m} \alpha_i L_i$, for $i = 1, ..., m$ and consequently $\sum_{i=1}^{m} \alpha_i w_i(P_{s-d}^*) \leq \sum_{i=1}^{m} \alpha_i L_i < \sum_{i=1}^{m} \alpha_i w_i(P_{s-d})$, which contradicts our assumption that $\sum_{i=1}^{m} \alpha_i w_i(P_{s-d}) \leq \sum_{i=1}^{m} \alpha_i w_i(P_{s-d}^*)$, $\forall P_{s-d}^*$. Since the path P_{s-d} can be found in polynomial time (e.g., via the Jaffe algorithm [16]), the MCP problem is solvable in polynomial time if $\sum_{i=1}^{m} \alpha_i L_i < \sum_{i=1}^{m} \alpha_i w_i(P_{s-d})$. ■

The work presented in Section II suggested that there is a connection between worst-case complexity and phase transitions. Using the terminology of Gent and Walsh [12], if problems are very under-constrained, then it is usually easy to find one of the many solutions. When problems are very over-constrained, it is usually easy to determine that they are insoluble. In the phase transition in between, problems are "critically constrained" and it is typically very hard to determine if they are soluble or insoluble. Applied to the MCP problem, we can distinct a phase transition based on the values of the constraints. If one of the constraints obeys (7), the probability of finding a path obeying the constraints is zero. Moreover, it can be verified in polynomial time, that there exists no path in the graph that obeys the constraints (property 7). On the other hand, if the values of the constraints are very large (under-constrained), such that all constraints follow (8), then a path satisfying these large constraints can be found in polynomial time. A phase transition is therefore expected to occur if the constraints do not obey (7) and (8). For small values of $L_i = w_i(P_{s-d,i}) + \epsilon$ (with $\epsilon > 0$) the MCP problem may still be insoluble, however the effort (complexity) needed to verify that indeed no feasible path is present in the graph has increased. In contrast to the case where the constraints $L_i < w_i(P_{s-d,i})$, only computing the m Dijkstra shortest paths is not sufficient to determine that the problem is insoluble. The SAMCRA [29] algorithm (or another exact MCP routing algorithm) must be invoked and will eventually observe that no path can obey the constraints. The larger the constraints become, the longer it will take to determine that no feasible path exists. Hence, increasing the constraints until a feasible path emerges augments the complexity of its solution. On the other hand, when decreasing the constraints starting from the upper boundary (8), first many paths will obey the constraints $L_i = \max_j(w_i(P_{s-d,j})) - \epsilon$ leading to a high probability that a feasible path will be found fast. If the values of the constraints decrease, the probability of finding a feasible path fast will also decrease. It is therefore expected that a phase transition occurs if there are only a few (if any) feasible paths present. In this case MCP ≈ MCOP. The steepness of the phase transition depends on the range between (7) and (8), which is heavily influenced by the correlation coefficient ρ as illustrated in Figure 13 (and by the computations in the Appendix). As discussed in Section IV, the correlation coefficient also impacts the level of complexity, which decreases if ρ increases.

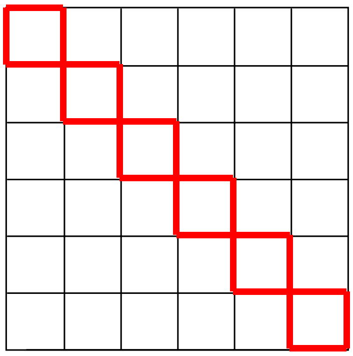

Fig. 14. Two-dimensional lattice with 49 nodes.

B. *Simulation results*

To be able to observe a phase transition, we must choose an intractable configuration. The simulation results in the previous section suggest that the graphs should contain many paths, have a large expected hopcount, and the link weights should have a negative correlation. All these properties are present in the class of two-dimensional lattices, which in terms of structure and complexity can be seen as a counterpart of the class of random graphs. In the remainder of this paper we confine attention to this class of lattices and try to distinguish a phase transition via simulations and an approximate analysis. For our simulations, we have chosen to use a single two-dimensional lattice with $N = 49$ nodes and correlated uniformly distributed link weights in the range [0,1]. Figure 14 illustrates the two-dimensional lattice that we have used.

A worst-case scenario is obtained if the source node is positioned in the upper left corner and the destination node in the lower right corner, causing the largest minimum hopcount. For each constraint L_1 and L_2, 100 different values were chosen in the NP-complete range (9) as discussed above, leading to a total of 10^4 iterations, all in the same lattice. Figure 15 displays the maximum queue-size[9] k used by SAMCRA, for $N = 49$ and $\rho = -1$. The corresponding contour plot is given in Figure 16.

Different constraints can lead to different m-dimensional shortest paths. For instance, if L_1 is small (e.g., 5.0 in Figure 15) and L_2 is large (e.g., 7.0 in Figure 15), then a path

[9] k is different from the previously used $k_{\min}$, since k denotes the maximum queue-size in SAMCRA whereas $k_{\min}$ is the queue-size that TAMCRA would have needed to attain the same solution as SAMCRA. We have used this larger value here, because $k_{\min} = 0$ if there is no path present.

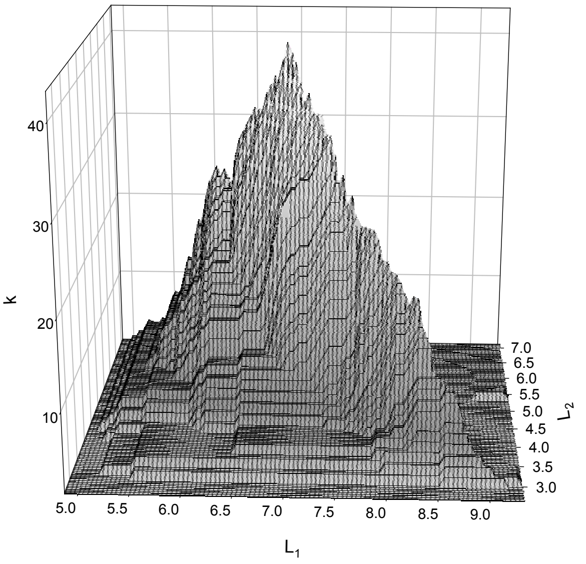

Fig. 15. The queue-size in a two-dimensional lattice, with correlated uniformly distributed link weights, $N = 49$, $\rho = -1$, and 10^4 different constraint vectors.

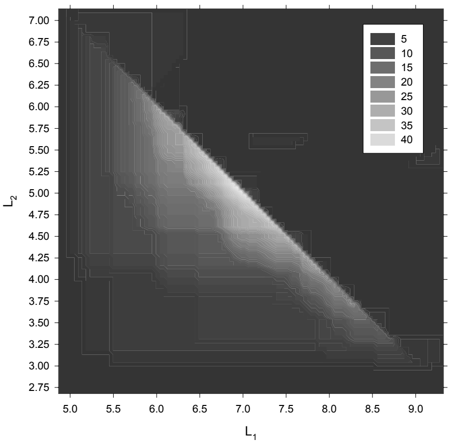

Fig. 16. Contour plot of the queue-size in a two-dimensional lattice, with correlated uniformly distributed link weights, $N = 49$, $\rho = -1$ and 10^4 different constraint vectors.

P obeying these constraints must also have a small weight $w_1(P) \leq L_1$ and the second weight may be large as long as $w_2(P) \leq L_2$. Since L_1 is slightly larger than the weight $w_1(P_{s-d;1})$ of the shortest Dijkstra path for measure 1, the path P may closely approximate $P_{s-d;1}$, which may be easy to find as indicated by small k values in Figure 15. Similarly, if L_1 is large (e.g., 9.0 in Figure 15) and L_2 is small (e.g., 3.0 in Figure 15), then a path P obeying these constraints may closely approximate the Dijkstra shortest path for measure 2 $P_{s-d;2}$, which may also be easy to find (as verified in Figure 15). We observe from Figure 15 that the complexity is largest when $L_1 = 6.94$ and $L_2 = 5.06$. These values are situated near the center of the rectangle (Figure 13) spanned by the NP-complete range (9) at $L_1^* = 7.09$ and $L_2^* = 4.91$. These observations seem to suggest that the complexity is largest when the constraints closely approximate the weights of the m-dimensional shortest path P, which equal $\sqrt{N} - 1$ on average (see Appendix, Eq. (19)). For two-dimensional lattices of $N = 49$ nodes, we therefore expect the highest complexity for $L_1 = L_2 = 6$. The deviation in our case is caused by only examining one single lattice, instead of the many required for statistical results.

The sharp edge/line in Figure 16, constituted by the different shortest paths, can be attributed to the extreme negative correlation ($\rho = -1$) as explained in Figure 13b and the Appendix. Since the link weights are chosen in the range [0,1], we have that for $\rho = -1$, $w_1(u,v) = 1 - w_2(u,v)$, $\forall(u,v) \in E$. Hence the path weights of any path P obey $w_1(P) = h - w_2(P)$, where $w_i(P) = \sum_{(u,v)\in P} w_i(u,v)$ and h equals the hopcount of path P. If we again look at Figure 16, we may observe that the straight line, once continued, intersects both axes L_1 and L_2 at 12, which is precisely the minimum hopcount of the two-dimensional lattice with 49 nodes. Moreover, since $w_1(P) = h - w_2(P)$, we know (see property 8) that when $L_1 + L_2 < h$, then no feasible path exists. This means that for the class of two-dimensional lattices with correlated ($\rho = -1$) uniformly distributed link weights, the constraints must obey $L_1 + L_2 \geq h$, for a feasible path to be possible. This condition for the constraints can be checked in polynomial time and it is therefore possible to obtain a much steeper phase transition than observed in Figures 15 and 16. Finally, we have also simulated with independent uniformly distributed link weights ($\rho = 0$) in the range [0,1]. As discussed in section IV, the complexity of solving the MCP and MCOP problems under independent link weights is smaller than with negatively correlated link weights. To observe a phase transition, we had to simulate with a lattice larger than $N = 49$. Figure 17 gives the contour plot for $N = 400$ and $\rho = 0$. The complexity is largest for $L_1 = 12.58$ and $L_2 = 15.11$.

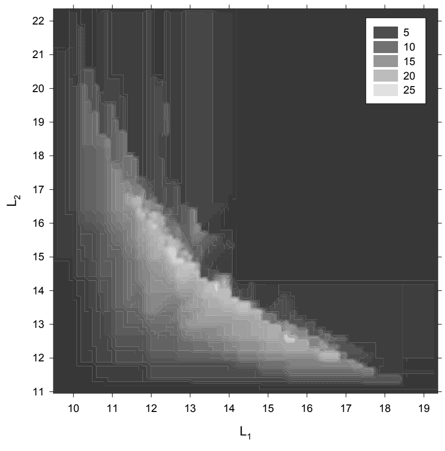

Fig. 17. Contour plot of the queue-size in a two-dimensional lattice, with uniformly distributed link weights, $N = 400$, $\rho = 0$ and 10^4 different constraint vectors.

It would be desirable to obtain an estimation of the size of the constraints that make the MCP problem critically constrained. Such an estimation would allow us to predict the location of the phase transition and hence give us an indication of the "critically constrained" region. In the next subsection we will attempt to provide an approximate analysis of the weights of the m-dimensional shortest path, because as we have seen above, choosing the constraints close to these weights may lead to a non-polynomial running time.

C. Estimation of the length of the shortest path in a lattice.

This last subsection discusses the approximate computation of the length of the m-dimensional shortest path between two corner points in a rectangular two-dimensional (2d) lattice with z_1 links vertically and z_2 links horizontally. The link weights are independent uniformly distributed in the range $(0, 1]$. The approximate analysis of the formulas presented in this subsection and some of the notation that is used, can be found in the Appendix. The asymptotic average weight of a $h = z_1 + z_2$ hop path in one dimension for a 2d-lattice is given by (13) as $E[W_1] \simeq \frac{h}{2e} \approx \frac{\sqrt{N}}{e}$. This estimate agrees reasonably well with simulations in the range $N \in [100, 1600]$, which accurately follow $E[W_{sim}] \approx 0.6N^{0.48}$.

The extension to m dimensions with independent link weight components ($\rho = 0$) for the average length $W_m = L_{eq}l_h$ is the approximation (17),

$$E[W_m] \simeq \frac{h}{e2^{\frac{1}{m}}}$$

The scaling $2^{-\frac{1}{m}}$ as a function of m has been observed in simulations, even for $N = 49$. This approximate analysis (16) shows that there is no shortest path obeying the constraints if the length, as defined in (15), $l_h(P) > 1$. This event has probability

$$\Pr[l_h > 1] \approx \exp\left(-\frac{h!}{z_1!z_2!}\left(\frac{L_{eq}^h}{h!}\right)^m\right)$$

Clearly, if the lattice (i.e., z_1, z_2 and $h = z_1 + z_2$) is fixed and the constraints decrease (increase), all (no) paths violate the constraints. The fact that there exists a path within the constraints depends on the product of the constraints or equivalent constraint L_{eq}. If $\frac{L_{eq}^h}{h!} > 1$ or $L_{eq} > (h!)^{\frac{1}{h}} \approx \frac{h}{e}$ (for large h), nearly all paths obey the constraints. If $\frac{L_{eq}^h}{h!} < 1$ or $L_{eq} < (h!)^{\frac{1}{h}} \approx \frac{h}{e}$, for a large number m of constraints, no path obeys the constraints. Hence, for large m and large h, there seems to be a critical value of the equivalent constraint $L_{eq} > (h!)^{\frac{1}{h}} \approx \frac{h}{e}$ for which $E[l_h] = \left(\frac{z_1!z_2!}{h!}\right)^{\frac{1}{mh}} < 1$ and specifically for the square lattice $E[l_h] \approx 2^{-\frac{1}{m}}$. Below that value the shortest path behavior is clearly different than above that value, which points to a phase transition.

The result (18) in two dimensions ($m = 2$), with perfectly negative correlation ($\rho = -1$), even points to a more confining situation, as was readily observed by comparing Figures 16 and 17. Since $E[L_{eq}l_h] \approx \frac{h}{2}$ (see (19)) and any random variable $L_{eq}l_h \geq \frac{h}{2}$, the average weight of the shortest path lies very close to the boundary $\frac{h}{2}$.

In summary, we have estimated the average length or weights of the shortest path for large values of h or, equivalently, the number of nodes N in the 2d-lattice. As common for extremal distributions, the variance is small, which implies a fast transition from 0 to 1 of $\Pr[L_{eq}l_h \leq y]$ around the average. The knowledge of the shortest path is important to set the constraints: if the constraints are close to $E[L_{eq}l_h]$, the problem is critically constrained and more computations are needed to determine whether there exists a path obeying the constraints or not. For constraints larger or smaller than $E[L_{eq}l_h]$, the problem is either under- or over-constrained and the verdict that there exists a path within the constraints is usually simple to draw with high probability. In the analysis presented in the Appendix, we have assumed that a possible overlap of h-hop paths is sufficiently weak to allow the application of the limit laws for independent random variables. Only relatively few paths will share a large number of links. We have used a heuristic argument to validate this assumption and have observed a good agreement with our simulation results. The second assumption is that the shortest path in the 2d-lattice has h hops or that $\Pr[\text{hops} > h]$ is negligibly small. This approximation is reasonable since simulations show that $\Pr[\text{hops} = h + 2k]$ is rapidly decaying in k with decay rate dependent on the size of the graph. The larger the graph, the slower the decay rate. However, for increasing m, simulations show that the shortest path tends to have h hops. Also for very negative correlation coefficients, the probability that shortest paths have h hops increases. Finally, although computed for uniformly distributed link weights, the same results hold for any distribution whose h-fold convolved distribution also behaves as $\frac{x^h}{h!}$ for small x. Any distribution in the same sphere of minimal attraction (such as exponentially distributed link weights with mean 1) yields the same results.

VI. Conclusions

In this paper we have evaluated the complexity of Quality of Service (QoS) routing. Finding a path based on multiple QoS constraints is proven to be an NP-complete problem. However, this Multi-Constrained Path (MCP) selection problem is not NP-complete in the strong sense, meaning that a pseudo-polynomial algorithm can exactly solve the problem. The NP-completeness of the MCP problem hinges on four factors, namely (1) the underlying topology, (2) link weights that can grow arbitrarily large or have an infinite granularity, (3) a very negative correlation among the link weights, and (4) the values of the constraints. If the values of the constraints are very large then it is easy to find a path within the constraints. On the contrary, if the values of the constraints are very small, then it is easy to verify that there is no path within the constraints. This indicates that there will be a phase transition if the constraints are around the weights of the m-dimensional shortest path in the network. In this case, it is expected to be difficult to establish whether a feasible path exists. If the four above-mentioned conditions are all necessary to induce intractability, they will allow network and service providers to properly dimension their network and to avoid intractable scenarios. Moreover, if the theory of phase transition holds for the MCP problem, then we know that QoS requirements close to the m-dimensional shortest path will, if admitted, provide the highest possible level of QoS, but also the highest computational cost. Such information is invaluable for pricing and billing mechanisms and admission control algorithms. Finally, a proper understanding and use of the four conditions, will allow for efficient QoS routing at controlled computational costs.

APPENDIX

In this appendix we will present an approximate analysis of the length of the m-dimensional shortest path in a two-dimensional lattice.

A. Analysis for a single link weight ($m = 1$)

Consider a rectangular 2d-lattice with size z_1 and z_2 and with independent uniformly distributed link weights on $(0, 1]$. The shortest hop path between two diagonal corner points consists of $h = z_1 + z_2$ hops. The weight W_h of such a h-hop path is the sum of h independent uniform random variables u_j and $W_h = \sum_{j=1}^{h} u_j$ has distribution,

$$F(x) = \Pr[W_h \le x] = \frac{1}{h!} \sum_{j=0}^{h} \binom{h}{j} (-1)^j (x-j)^h 1_{j \le x} \tag{10}$$

In particular, $\Pr[W_h \le h] = 1$ and for small $x < 1$ holds that $F(x) = \frac{x^h}{h!}$. We assume that the number[10] $l = \binom{z_1+z_2}{z_1} = \frac{h!}{z_1!z_2!}$ of those h-hop paths is large. Although these paths can possibly overlap, we ignore this dependence for the moment and assume that the minimum weight among all h-hop paths is well approximated by the limit law (of extremal types [2]) for the minimum of a set of independent random variables X_k with identical distribution F. In particular, if $\lim_{l\to\infty} l(F(x_l)) = \zeta$,

$$\lim_{l\to\infty} \Pr\left[\min_{1\le k\le l} X_k > x_l\right] = e^{-\zeta} \tag{11}$$

The limit sequence must obey $l(F(x_l)) \to \zeta$ for sufficiently large l, which implies that $F(x_l)$ must be small or, equivalently, x_l must be small. Hence, $l\frac{x_l^h}{h!} = \zeta$ or $x_l = \left(\frac{h!\zeta}{l}\right)^{\frac{1}{h}}$. The limit law (11) for the minimum weight $W = \min_{1\le k\le l} W_{h,k}$ of the shortest hop path between two corner points in a rectangular 2d-lattice is

$$\lim_{l\to\infty} \Pr\left[\min_{1\le k\le l} W_{h,k} > \left(\frac{h!x}{l}\right)^{\frac{1}{h}}\right] = e^{-x}$$

In other words, the random variable $\frac{lW^h}{h!}$ tends to an exponential random variable with mean 1 for large $l = \frac{h!}{z_1!z_2!}$ or

$$\Pr[W \le y] \approx 1 - \exp\left(-\frac{y^h}{z_1!z_2!}\right)$$

The mean shortest weight of a h-hop path equals

$$\begin{aligned} E[W] &= \int_0^\infty (1 - F_W(x))\,dx \approx \int_0^\infty \exp\left(-\frac{x^h}{z_1!z_2!}\right)dx \\ &= \Gamma\left(1 + \frac{1}{h}\right)(z_1!z_2!)^{\frac{1}{h}} \end{aligned} \tag{12}$$

For a square 2d-lattice where $z_1 = z_2 = \frac{h}{2}$, we have

$$E[W] = \Gamma\left(1 + \frac{1}{h}\right)\left(\left(\frac{h}{2}\right)!\right)^{\frac{2}{h}}$$

Using Stirling's formula [1, 6.1.38] for the factorial $h! = \sqrt{2\pi}h^{h+\frac{1}{2}}e^{-h+\frac{\theta}{12h}}$ where $0 < \theta < 1$, we finally arrive for large h at

$$E[W] \simeq \left(\frac{h}{2e}\right)\left(\sqrt{\pi h}e^{\frac{\theta}{6h}}\right)^{\frac{2}{h}} \approx \frac{h}{2e} \tag{13}$$

We now provide a heuristic argument why, for large h, the neglect of the dependence between h-hop paths is justified. Denote by Γ_h the set of all h-hop paths in the 2d-lattice between corner points, with the number of those paths $|\Gamma_h| = \binom{h}{z_1}$. A particular path of the set Γ_h is denoted by γ_h. We denote the weight of γ by $w(\gamma)$. Let w_N be the (random) weight of the shortest path between corner points in the 2d-lattice with independent uniformly distributed link weights. The event $\{h_N = h, w_N \le z\}$ implies that there is a h-hop path γ_h with weight $w(\gamma_h) \le z$ and, therefore,

$$\begin{aligned} \Pr[h_N &= h, w_N \le z] \le \Pr[\cup_{\gamma\in\Gamma_h}\{w(\gamma) \le z\}] \\ &\le \sum_{all\ \gamma} \Pr[\gamma \in \Gamma_h, w(\gamma) \le z], \end{aligned} \tag{14}$$

where the second inequality follows from Boole's inequality ($\Pr[\cup A_j] \le \sum \Pr[A_j]$). Using the independence of the link and the link weights,

$$\begin{aligned} \Pr[h_N &= h, w_N \le z] \le \sum_{all\ \gamma} \Pr[\gamma \in \Gamma_h] \Pr[w(\gamma) \le z] \\ &= E[|\Gamma_h|] \Pr[w(\gamma_h) \le z] \end{aligned}$$

or since $\Pr[w(\gamma_h) \le z] = \Pr[W_h \le z]$ given by (10)

$$\Pr[h_N = h, w_N \le z] \le \binom{h}{z_1} F(z)$$

From this rigorous inequality we infer the heuristic argument $\Pr[h_N = h, w_N \le z] \simeq \binom{h}{z_1} F(z)$. For a typical value of z, the probabilities should sum to 1, yielding,

$$1 = \sum_{j=0}^{\infty} \Pr[h_N = h + 2j, w_N \le z] \simeq F(z)\binom{h}{z_1}$$

where the assumption is that $\sum_{j=1}^{\infty} \Pr[h_N = h + 2j, w_N \le z] << \Pr[h_N = h, w_N \le z]$. Hence a *typical* value for the weight of the shortest path is the solution of $F(z) = \frac{1}{\binom{h}{z_1}}$. For small z, we have $F(z) = \frac{z^h}{h!}$ such that

$$z \sim (z_1!z_2!)^{\frac{1}{h}}$$

which agrees with $E[W]$ in (12).

[10] Any path in a rectangular lattice can be represented by a sequence of r(ight), l(eft), u(p), and d(own). A shortest hop path between diagonal corner points consists of x r's (or l's) and y d's (or u's). The total number of these paths equals $\binom{z_1+z_2}{z_1}$.

B. Analysis for multiple link weights ($m > 1$)

Let us now consider a 2d-lattice where each link is specified by a link weight vector $\vec{w} = (w_1, w_2, \ldots, w_m)$. We further confine to the case where all link weight components are independent and uniformly distributed. Using the non-linear length of SAMCRA [29], the length of a h-hop path is computed as

$$l_h(P) = \max_{1 \leq j \leq m} \left[\frac{W_{h,j}}{L_j} \right] \tag{15}$$

where each weight per component j is $W_{h,j} = \sum_{n=1}^{h} u_{n,j}$ with distribution F given in (10). Since all link weight components are independent,

$$\Pr\left[l_h(P) \leq x\right] = \prod_{j=1}^{m} F\left(L_j x\right)$$

For small x, $\prod_{j=1}^{m} F\left(L_j x\right) \approx \prod_{j=1}^{m} \frac{(L_j x)^h}{h!} = \left(\frac{x^h}{h!}\right)^m \prod_{j=1}^{m} L_j^h$. We define an equivalent constraint $L_{eq} = \left(\prod_{j=1}^{m} L_j\right)^{\frac{1}{m}}$. Neglecting the dependence of h-hop paths due to possible overlap as above and applying the limit law for the minimum length with $\lim_{l\to\infty} l\left(\prod_{j=1}^{m} F\left(L_j x_l\right)\right) = \zeta$ results in

$$\lim_{l\to\infty} \Pr\left[\min_{1\leq k\leq l} l_{h,k}(P) > \left(\frac{x(h!)^m}{l\left(L_{eq}\right)^{mh}}\right)^{\frac{1}{mh}}\right] = e^{-x}$$

For large $l = \frac{h!}{z_1! z_2!}$, we obtain the approximate distribution of the minimum length, $l_h = \lim_{l\to\infty} \min_{1\leq k\leq l} l_{h,k}(P)$, of a h-hop path,

$$\Pr\left[l_h \leq y\right] = 1 - \exp\left(-\frac{h!}{z_1! z_2!}\left(\frac{\left(L_{eq} y\right)^h}{h!}\right)^m\right) \tag{16}$$

The average length of the shortest h path is with $(h!)^{\frac{1}{h}} \approx \frac{h}{e}\left(2\pi h\right)^{\frac{1}{2h}} \approx \frac{h}{e}$,

$$\begin{aligned} E\left[l_h\right] &= \int_0^\infty \Pr\left[l_h > y\right] dy \\ &= \Gamma\left(1 + \frac{1}{mh}\right) \frac{(h!)^{\frac{1}{h}}\left(\frac{z_1! z_2!}{h!}\right)^{\frac{1}{mh}}}{L_{eq}} \\ &\approx \frac{h}{eL_{eq}}\left(\frac{z_1! z_2!}{h!}\right)^{\frac{1}{mh}} \end{aligned}$$

Since all link weight components are independent and equal in distribution, we can interpret $E\left[L_{eq} l_h\right]$ as the weight of the shortest path in m dimensions. For a square 2d-lattice, using [1, 6.1.49] $\binom{2z}{z} \approx \frac{2^{2z}}{\sqrt{\pi z}}$, the formula

$$E\left[L_{eq} l_h\right] \approx \frac{h}{e 2^{\frac{1}{m}}} \tag{17}$$

shows that the weight of the shortest path very slowly increases with m as $2^{-\frac{1}{m}}$ and that for any dimension m, $\frac{h}{e2} \leq E\left[L_{eq} l_h\right] \leq \frac{h}{e}$.

The variance equals

$$\begin{aligned} var\left[l_h\right] &= \int_0^\infty (y - E\left[l_h\right])^2 d\Pr\left[l_h \leq y\right] \\ &= \frac{(h!)^{\frac{2}{h}}\left(\frac{z_1! z_2!}{h!}\right)^{\frac{2}{mh}}}{\left(L_{eq}\right)^2}\left(\Gamma\left(1 + \frac{2}{mh}\right) - \Gamma^2\left(1 + \frac{1}{mh}\right)\right) \end{aligned}$$

For large h, we see that

$$\Gamma\left(1 + \frac{2}{mh}\right) - \Gamma^2\left(1 + \frac{1}{mh}\right) = \frac{\pi^2}{6}\frac{1}{\left(mh\right)^2} + O\left(\frac{1}{\left(mh\right)^3}\right)$$

Hence,

$$var\left[l_h\right] \approx \frac{\pi^2}{6}\frac{\left(E\left[l_h(P)\right]\right)^2}{\left(mh\right)^2} \to \frac{\pi^2}{6}\frac{1}{em^2 L_{eq}}$$

which is rather small and independent of h as is common for extremal distributions.

C. Perfect negative correlation ($m = 2$)

In case of $m = 2$ and perfect negative correlation, the first path weight is $W_{h,1} = \sum_{j=1}^{h} u_j$ and the second is $W_{h,2} = h - \sum_{j=0}^{h} u_j = h - W_{h,1}$. Then,

$$l_h(P) = \max\left[\frac{W_{h,1}}{L_1}, \frac{W_{h,2}}{L_2}\right] = \max\left[\frac{W_{h,1}}{L_1}, \frac{h - W_{h,1}}{L_2}\right]$$

If $L_1 = L_2 = L_{eq}$, then $L_{eq} l_h(P) \geq \frac{h}{2}$ and if $W_{h,1} \leq x \leq \frac{h}{2}$, then $L_{eq} l_h(P) \geq h - x$ else $\frac{h}{2} \leq L_{eq} l_h(P) \leq x$. Thus, $\Pr\left[\frac{h}{2} \leq L_{eq} l_h(P) \leq z\right]$ equals

$$\begin{aligned} &\Pr\left[\frac{h}{2} \leq W_{h,1} \leq z\right] + \Pr\left[h - z \leq W_{h,1} \leq \frac{h}{2}\right] \\ &= F(z) - F\left(\frac{h}{2}\right) + F\left(\frac{h}{2}\right) - F(h - z) \\ &= F(z) - F(h - z) \end{aligned}$$

Assuming as before independence of paths, then for the minimum length path holds,

$$\Pr\left[\frac{h}{2} \leq \min_{1\leq k\leq l} L_{eq} l_{h,k}(P) \leq z\right] = 1 - \prod_l \Pr\left[L_{eq} l_h(P) > z\right]$$

With $\Pr[L_{eq} l_h] = \Pr\left[\frac{h}{2} \leq \min_{1\leq k\leq l} L_{eq} l_{h,k}(P) \leq z_l\right]$,

$$\begin{aligned} 1 - \Pr\left[L_{eq} l_h\right] &= \exp\left[l \log\left(1 - \left[F(z_l) - F(h - z_l)\right]\right)\right] \\ &= \exp\left[-l\left[F(z_l) - F(h - z_h)\right]\right] \\ &\quad \times\left(1 + o\left[hF(z_l) - F(h - z_l)\right]\right) \end{aligned}$$

If $\lim_{l\to\infty} l\left[F(z_l) - F(h - z_l)\right] = \xi$, then $1 - \Pr\left[\frac{h}{2} \leq L_{eq} l_h \leq z_l\right] = e^{-\xi}$. It remains to find z_l in terms of ξ. We rewrite $z_l = \frac{h}{2} + x_l$. For small x_l and with $f(x) = \frac{dF(x)}{dx}$,

$$\begin{aligned} \xi &= l\left[F\left(\frac{h}{2} + x_l\right) - F\left(\frac{h}{2} - x_l\right)\right] \\ &= l\left[F\left(\frac{h}{2}\right) + f\left(\frac{h}{2}\right) x_l + \right. \\ &\quad \left. - F\left(\frac{h}{2}\right) + f\left(\frac{h}{2}\right) x_l + O\left(x_l^3\right)\right] \\ &= 2lf\left(\frac{h}{2}\right) x_l + o\left(x_l^3\right) \end{aligned}$$

such that, with the Gaussian approximation for $f\left(\frac{h}{2}\right) \simeq \frac{1}{\sqrt{\frac{\pi h}{6}}}$ and $l = \frac{h!}{z_1!z_2!}$

$$x_h = \frac{\xi}{2lf\left(\frac{h}{2}\right)} = \frac{\xi(z_1!z_2!)}{2h!}\sqrt{\frac{\pi h}{6}}$$

Finally,

$$\Pr\left[\frac{h}{2} \leq L_{eq}l_h \leq \frac{h}{2} + y\right] = 1 - \exp\left(-2\frac{h!}{z_1!z_2!\sqrt{\frac{\pi h}{6}}}y\right) \tag{18}$$

from which

$$\begin{aligned} E\left[L_{eq}l_h\right] &= \frac{h}{2} + \int_0^\infty \exp\left(-2\frac{h!}{z_1!z_2!\sqrt{\frac{\pi h}{6}}}y\right) dy \\ &= \frac{h}{2} + \frac{(z_1!z_2!)}{2h!}\sqrt{\frac{\pi h}{6}} \end{aligned} \tag{19}$$

Hence, for large h, the average $E\left[L_{eq}l_h\right]$ rapidly tends to $\frac{h}{2}$, as has been verified through simulations.

Acknowledgment

We would like to thank Selma Begtasevic for providing us with many simulation results.

References

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, Inc., N.Y., 1968.

[2] M.A. Berger, *An Introduction to Probability and Stochastic Processes*, Springer Texts in Statistics, Springer-Verlag, New York, 1993.

[3] B. Bollobas, *Random Graphs*, Cambridge University Press, second edition, 2001.

[4] P. Cheeseman, B. Kanefsky and W.M. Taylor, "Where the Really Hard Problems Are," Proc. of IJCAI-91, pp. 331-337, San Mateo, CA, 1991.

[5] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," Proc. of the ICC'98 conference, vol. 2, pp. 874-879, 1998.

[6] H. De Neve and P. Van Mieghem, "A multiple quality of service routing algorithm for PNNI," Proc. of IEEE ATM Workshop, pp. 324-328, Fairfax, May 26-29, 1998.

[7] H. De Neve and P. Van Mieghem, "TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm," Computer Communications, 2000, vol. 23, pp. 667-679.

[8] P.E. Dunne, A. Gibbons and M. Zito, "Complexity-theoretic models of phase transitions in search problems," Theoretical Computer Science, no. 249, pp. 243-263, 2000.

[9] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On power-law relationships of the Internet topology," Proc. of ACM SIGCOMM'99, Cambridge, Massachusetts, pp. 251-262, 1999.

[10] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.

[11] M.R. Garey and D.S. Johnson, "'Strong' NP-completeness Results: Motivation, Examples, and Implications," Journal of the ACM, vol. 25, no. 3, July 1978, pp. 499-508.

[12] I.P. Gent and T. Walsh, "Analysis of heuristic for number partitioning," Computational Intelligence, vol. 14, no. 3, pp. 430-452, 1998.

[13] G.H. Hardy, J.E. Littlewood and G. Polya, *Inequalities*, Cambridge University Press, 2nd edition, 1973.

[14] B. Hayes, "Can't Get No Satisfaction," American Scientist, vol. 85, no. 2, pp. 108-112, March-April, 1997.

[15] G. Istrate, "Computational Complexity and Phase Transitions," Proc. of the 15th anual Conference on Computational Complexity, Florence, Italy, pp. 104-115, July 4-7, 2000.

[16] J.M. Jaffe, "Algorithms for finding paths with multiple constraints," Networks, vol. 14, pp. 95-116, 1984.

[17] F.A. Kuipers and P. Van Mieghem, "QoS routing: Average Complexity and Hopcount in m Dimensions," Proc. of Second COST 263 International Workshop, QofIS 2001, Coimbra, Portugal, pp. 110-126, September 24-26, 2001.

[18] F.A. Kuipers, T. Korkmaz, M. Krunz and P. Van Mieghem, "Overview of Constraint-Based Path Selection Algorithms for QoS Routing," IEEE Communications Magazine, vol. 40, no. 12, December 2002.

[19] F.A. Kuipers, T. Korkmaz, M. Krunz and P. Van Mieghem, "Performance Evaluation of Constraint-Based Path Selection Algorithms," to appear in IEEE Network.

[20] L.A. Levin, "Average Case Complete Problems," SIAM J. Comput., 15(1):285-286, 1986.

[21] D.H. Lorenz and A. Orda, "QoS Routing in Networks with Uncertain Parameters," IEEE/ACM Transactions on Networking, vol. 6, no. 6, December 1998.

[22] Q. Ma and P. Steenkiste, "Quality-of-Service Routing with Performance Guarantees," Proc. of IWQoS, New York, May 1997.

[23] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman and L. Troyansky, "Determining computational complexity from characteristic 'phase transitions'," Nature, vol. 400, pp. 133-137, July 8, 1999.

[24] E.F. Mykytka and C-Y Cheng, "Generating Correlated Random Variates Based on an Analogy Between Correlation and Force," Proc. of the 1994 Winter Simulation Conference, pp. 1413-1416, 1994.

[25] D. Pisinger, *Algorithms for Knapsack Problems*, Ph.D. thesis, Dept. of Computer Science, University of Copenhagen, Denmark, February 1995.

[26] P. Van Mieghem, H. De Neve and F.A. Kuipers, "Hop-by-hop Quality of Service Routing," Computer Networks, vol. 37/3-4, pp. 407-423, 2001.

[27] P. Van Mieghem, "Paths in the simple Random Graph and the Waxman Graph," Probability in the Engineering and Informational Sciences (PEIS), vol. 15, pp. 535-555, 2001.

[28] P. Van Mieghem and F.A. Kuipers, "On the Complexity of QoS Routing," Computer Communications, special issue on QofIS'01, vol. 26, no. 4, pp. 376-387, March 2003.

[29] P. Van Mieghem and F.A. Kuipers, "Concepts of Exact Quality of Service Algorithms," to appear in IEEE/ACM Transaction on Networking.

[30] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications," IEEE JSAC, vol. 14, no. 7, pp. 1228-1234, September, 1996.

[31] Z. Wang, "On the complexity of quality of service routing," Information Processing Letters, vol. 69, pp. 111-114, 1999.