QoS routing: average complexity and hopcount in m dimensions

Kuipers, FA; Van Mieghem, PFA

DOI
10.1007/3-540-45412-8_9

Publication date
2001

Document Version
Accepted author manuscript

Published in
QoFIS 2001: proceedings

QoS routing: Average Complexity and Hopcount in m dimensions

F.A. Kuipers* and P. Van Mieghem **

Delft University of Technology
Information Technology and Systems
P.O Box 5031, 2600 GA Delft, The Netherlands

Abstract. QoS routing is expected to be an essential building block of a future, efficient and scalable QoS-aware network architecture. We present SAMCRA, an exact QoS routing algorithm that guarantees to find a feasible path if such a path exists. The complexity of SAMCRA is analyzed. Because SAMCRA is an exact algorithm, most findings can be applied to QoS routing in general.
The second part of this paper discusses how routing with multiple independent constraints affects the hopcount distribution. Both the complexity as the hopcount analysis indicate that for a special class of networks, QoS routing exhibits features similar to single-parameter routing.

1 Introduction: constrained-based routing

Delivering end-to-end Quality of Service (QoS) is widely believed to become an essential asset for the future Internet. Much research has been done (and is continuing) on this topic, which resulted in the proposal of several QoS architectural frameworks including IntServ/RSVP, DiffServ, MPLS and traffic engineering through constrained-based routing. Each of these proposals has some advantages over the others. A future QoS-aware network architecture for the Internet will therefore likely comprise of a combination of several architectural frameworks in order to provide end-to-end QoS in an efficient, stable and scalable manner. In this paper we argue that constrained-based routing is an essential QoS building block for providing efficient QoS solutions. Some examples motivate this statement. In the IntServ/RSVP framework, RSVP tries to reserve a (best-effort) path and then provides QoS through appropriate scheduling, queueing, dropping, etc.. However, because the constraints are ignored while setting up/reserving the path, the reserved path is possibly not the best choice, or even worse, a path may not be found at all (flow blocked), while there is a feasible path available. The same phenomenon can be observed in DiffServ. In DiffServ the local packet handling mechanisms (scheduling, etc.) provide the differentiated

* email: F.A.Kuipers@its.tudelft.nl
** email: P.VanMieghem@its.tudelft.nl

service, after which the packets are forwarded along a best-effort path. MPLS is a forwarding scheme where packets are given a label corresponding to a certain MPLS path over which they are forwarded. The combination of MPLS with constrained-based routing to set up a MPLS path seems an intuitively straightforward solution. In conclusion, a future QoS-aware network architecture will benefit from combining constrained-based routing with appropriate packet handling in order to provide efficient QoS solutions.

In section 2 we will present SAMCRA [17], a Self-Adaptive Multiple Constraints Routing Algorithm and analyze its complexity, which is verified through simulations. The second part (section 3) of this paper analyses the hopcount of the shortest path in multiple dimensions. This section is an extension to previous work on the hopcount in the Internet [16]. We conclude this paper with conclusions and speculations in section 4.

2 SAMCRA: a Self-Adaptive Multiple Constraints Routing Algorithm

SAMCRA is the successor of TAMCRA, a Tunable Accuracy Multiple Constraints Routing Algorithm [8], [7]. As opposed to TAMCRA, SAMCRA guarantees to find a path within the constraints, provided such a path exists. Furthermore, SAMCRA only allocates queue-space (= memory) when truly needed, whereas in TAMCRA the allocated queue-space is predefined. The major performance criterion for SAMCRA, namely the running-time/complexity, will be discussed in paragraph 2.2. Similar to TAMCRA, SAMCRA is based on three fundamental concepts: (1) a non-linear measure for the path length, (2) the k-shortest path approach [5] and (3) the principle of non-dominated paths [12]. Before we clarify these three concepts we will first introduce the notations used throughout this paper.

A network topology is denoted by $G(N, E)$, where N is the set of nodes and E is the set of links. With a slight abuse of notation we will also denote by N and E respectively the number of nodes and the number of links. A network supporting QoS consists of link weight vectors with m non-negative QoS measures ($w_i(e)$, $i = 1, ..., m$, $e \in E$) as components. The QoS measure of a path can either be additive in which case it is the sum of the QoS measures along the path (such as delay, jitter, the logarithm of packet loss, cost of a link, etc.) or it can be the minimum(maximum) of the QoS measures along the path (typically, available bandwidth and policy flags). Min(max) QoS measures are treated by omitting all links (and possibly disconnected nodes) which do not satisfy the requested min(max) QoS constraints. We call this topology filtering. Additive QoS measures cause more difficulties: *the multiple constrained path (MCP) problem, defined as finding a path subject to more than one additive constraint (L_i), is known to be NP-complete* [10], [20] and hence considered as intractable for large networks.

We continue by explaining SAMCRA's three basic concepts:

1. Motivated by the geometry of the constraints surface in m-dimensional space, we defined the length of a path P as follows [8]:

$$l(P) = \max_{1 \leq i \leq m} \left(\frac{w_i(P)}{L_i} \right) \tag{1}$$

where $w_i(P) = \sum_{e \in P} w_i(e)$.
The definition of the path length has to be non-linear in order to guarantee that a retrieved path is within the constraints, i.e. $l(P) \leq 1$. A solution to the MCP problem is a path whose weights are all within the constraints. SAMCRA can also be applied to solve multiple constrained optimization problems, e.g. delay-constrained least-cost routing. Depending on the specifics of a constrained optimization problem, SAMCRA can be used with different length functions, provided they obey the criteria for length in vector algebra. Example length functions are given in [17]. In [11] and [13] TAMCRA-based algorithms with specific length functions are proposed and evaluated. By using length function (1), all QoS measures are considered as equally important. An important corollary of a non-linear path length as (1) is that *the subsections of shortest paths in multiple dimensions are not necessarily shortest paths.* This suggests to consider in the computation more paths than only the shortest one, leading us naturally to the k-shortest path approach.
2. The k-shortest path algorithm [5] is essentially Dijkstra's algorithm that does not stop when the destination is reached, but continues until the destination has been reached k times. This concept is applied to the intermediate nodes i on the path from source node s to destination node d, where we keep track of multiple sub-paths from s to i. Not all sub-paths are stored, but an efficient distinction based on non-dominance is made.
3. The third concept in SAMCRA is that of dominance. A (sub)-path Q is dominated by a (sub)-path P if $w_i(P) \leq w_i(Q)$ for $i = 1, .., m$, with an inequality for at least one i. SAMCRA only considers non-dominated (sub)-paths. This property allows us to efficiently reduce our search-space (all paths between the source and destination) without compromising the solution.

2.1 SAMCRA meta-code

SAMCRA(G, s, d, L)
G: graph, s: source node, d: destination node, L: constraints

```
1. counter = 0 for all nodes
2. endvalue = 1.0
3. path(s[1]) = NULL and length(s[1]) = 0
4. put s[1] in queue
5. while(queue ≠ empty)
6.     extract_min(queue) -> u[i]
7.     u[i] = marked grey
```

```
8.     if(u = d)
9.         stop
10.     else
11.         for each v ∈ adjacency_list(u)
12.             if(v ≠ previous node of u[i])
13.                 PATH = path(u[i]) + (u,v)
14.                 LENGTH = length(PATH)
15.                 check all non-black paths at v and PATH
           for dominancy & endvalue → mark obsolete paths black
16.                 if(LENGTH ≤ endvalue and PATH non-dominated)
17.                     if(no black paths)
18.                         counter(v) = counter(v)+1
19.                         j = counter(v)
20.                         path(v[j]) = PATH
21.                         length(v[j]) = LENGTH
22.                         put v[j] in queue
23.                     else
24.                         replace a black path with PATH
25.                     if(v = d and LENGTH < endvalue)
26.                         endvalue = LENGTH
```

For a detailed explanation of this meta-code, we refer to [17].

2.2 Complexity of SAMCRA

If N and E are the number of nodes and of links respectively in the graph $G(N, E)$, the queue in SAMCRA can never contain more than kN path lengths, where k denotes the number of feasible (i.e. within the constraints) non-dominated paths that are stored in the queue of a node. Because SAMCRA self-adaptively allocates the queue-size at each node, k may differ per node. When using a Fibonacci (or relaxed) heap to structure the queues [6], selecting the minimum path length among kN different path lengths takes at most a calculation time of the order of $log(kN)$. As each node can at most be selected k times from the queue, the **extract_min** function (explained in [6]) in line 6 of SAMCRA's meta-code takes $O(kNlog(kN))$ at most. The for-loop starting on line 11 is invoked at most k times from each side of each link in the graph. Calculating the length takes $O(m)$ when there are m metrics in the graph while verifying path dominance takes $O(km)$ at most. Adding or replacing a path length in the queue takes $O(1)$. Adding the contributions yields a worst-case complexity with $k = k_{\max}$ of

$$O(kN \log(kN) + k^2 mE) \quad (2)$$

where $k_{\max}$ is an upper-bound on the number of non-dominated paths in $G(N, E)$. A precise expression for k is difficult to find. However knowledge about k is crucial to the complexity of SAMCRA, because a large k could make the algorithm useless. As an upper-bound for k, we will use $k_{\max} = \lfloor e(N-2)! \rfloor$, which is an upper-bound on the total number of paths between a source and destination in $G(N, E)$ [18]. If the constraints/metrics have a finite granularity, another upper-bound applies [8]:

$$k_{\max} = \frac{\prod_{i=1}^{m} L_i}{\max_j (L_j)} \quad (3)$$

where the constraints L_i are expressed as an integer number of a basic metric unit. For instance, if the finest granularity is 1 msec, then the constraint value is expressed in an integer number times 1 msec.

Clearly, for a single constraint ($m = 1$ and $k = 1$), the complexity (2) reduces to that of Dijkstra's algorithm. For multiple metrics the worst-case complexity of SAMCRA is NP-complete.

We will now discuss the complexity of SAMCRA/MCP in more depth. First we will illustrate by an example that an exponential-time algorithm may be better in practice than a polynomial time algorithm: exponential $O(1,001^N)$ versus polynomial $O(N^{1000})$. For any reasonable network size (up to $N = 1,6 \cdot 10^7$) the exponential time algorithm outperforms the polynomial time algorithm. Although this example is exaggerated, there exist some (pseudo)-polynomial ε-approximation algorithms whose running-time is too large for practical purposes and which are therefore only of theoretical interest. Secondly, although there exist many problems that are NP-complete, their average-case complexity might not be intractable, meaning that such an algorithm could have a good performance in practice. The average level of "intractability" differs per NP-complete problem. The theory of average-case complexity was first advocated by Levin [14]. We will now give a calculation that suggests that *the average and even amortized*[1] *complexity of SAMCRA is polynomial in time* for fixed m and all weights w_i independent random variables.

Lemma 1: The expected number of non-dominated vectors in a set of T i.i.d. vectors in m dimensions is upper bounded by $(\ln T)^{m-1}$.

A proof of Lemma 1 can be found by adopting a similar approach as presented in [3] or [2]. Moreover, the results are the same. To gain some insight into the number of non-dominated paths in a graph, we will assume that the path-vectors are i.i.d. vectors[2]. When we apply lemma 1 to the complexity of SAMCRA, we see that in the worst-case SAMCRA examines $T = \lfloor e(N-2)! \rfloor$ paths, leading us to an expected number of non-dominated paths between the source and destination in the worst-case of

$$(\ln T)^{m-1} = (\ln(\lfloor e(N-2)! \rfloor))^{m-1} \leq (1 + (N-2)\ln(N-2))^{m-1}$$

[1] Amortized analysis differs from average-case analysis in that probability is not involved; an amortized analysis guarantees the average performance of each operation in the worst-case [6].

[2] In reality the m weights of the path-vectors will not be i.i.d..

which is polynomial in N. Hence, the amortized complexity of SAMCRA is (2) with $k = (\ln T)^{m-1} = (\ln(\lfloor e(N-2)!\rfloor))^{m-1}$, which is polynomial in time for fixed m.

In the limit $m \to \infty$ and for w_j independent random variables, all paths in $G(N, E)$ are non-dominated, which leads to the following lemma (proved in [17]).

Lemma 2: If the m components of the link weight vectors are independent random variables and the constraints L_j are such that $0 \leq \frac{w_j}{L_j} \leq 1$, then any path with K hops has precisely a length (as defined by (1)) equal to K in the limit $m \to \infty$.

This means that for $m \to \infty$ it suffices to calculate the minimum hop path, irrespective of the weight structure of the m independent components. Since the minimum hop problem is an instance of a single metric shortest path problem, it has polynomial complexity.

Summarizing we can say that if the link weights are independent random variables, there are two properties reducing the search-space (the number of paths to be examined) of SAMCRA. For m small the concept of non-dominance is very powerful, resulting in the presence of only a small number of non-dominated paths between two points in a graph. At the other extreme, for m large, the values L_j of the constraints cause the largest search-space reduction, because only a few paths between the source and destination lie within the constraints. Even if the constraints are chosen infinitely large, SAMCRA may lower them in the course of the computation (by means of `endvalue`, line 25/26 meta-code) without affecting the solution.

The two properties complement each other, resulting in an overall good average performance of SAMCRA. The simulation results of the next paragraph indicate that the average complexity of SAMCRA is $O(N \log N + mE)$, i.e. (2) with $E[k] \approx 1$, for the class $G_p(N)$ of random graphs [4], with independent uniformly distributed link weights.

2.3 Simulation results on complexity

The simulations were performed on a large number (minimum of 10^5) of random graphs of the type $G_p(N)$ [4], where p is the expected link-density ($p = 0.2$) and N the number of nodes. The m link weights are independent uniformly distributed random variables.

We will start by presenting the simulation results for $m = 2$. Figure 1 gives the minimum queue-size ($k_{\min}$) needed to find a feasible path. If TAMCRA had used that particular $k_{\min}$ under the same conditions, it would have found the same shortest feasible path as SAMCRA did, but if a smaller value had been used TAMCRA would not have found the shortest path. Figure 2 gives the statistics corresponding to the data in Figure 1.

Figures 1 and 2 show that an (exponential) increase of N does not result in a significant increase in $k_{\min}$. The expectation $E[k]$ remains close to 1 and hardly increases with N. If we extrapolate these results to $N \to \infty$, figures 1 and 2 suggest that for the class of random graphs $G_p(N)$ with 2 independent uniformly distributed link weights, the average complexity of SAMCRA is approximately $O(N \log N + 2E)$.

Figures 3-6 show a similar behavior ($E[k] \approx 1$). The only difference is that the worst-case values ($\max[k]$) have slightly increased with m, as a result of the increased expected number of non-dominated paths. However, since $E[k]$ stays close to one, the simulation results suggest that the two search-space-reducing concepts, dominance and constraint values, are so strong that the average complexity of SAMCRA is not significantly influenced by the number of metrics m. The behavior of $k_{\min}$ as a function of the number of constraints m is illustrated in Figure 7.

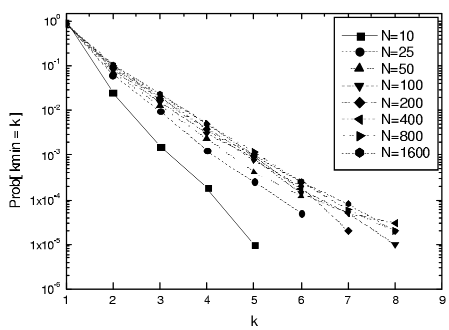

Figure 1: P.d.f. of $\mathrm{k_{min}}$, m=2

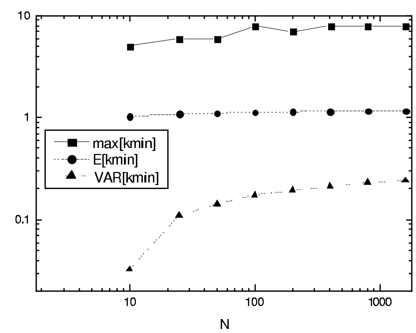

Figure 2: $\mathrm{k_{min}}$-statistics, m=2

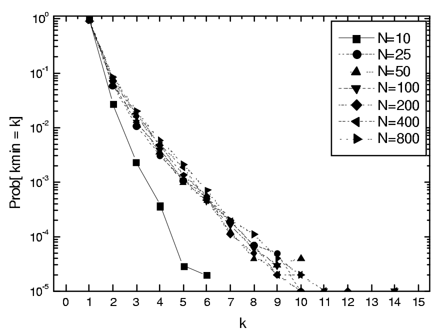

Figure 3: P.d.f. of $\mathrm{k_{min}}$, m=4

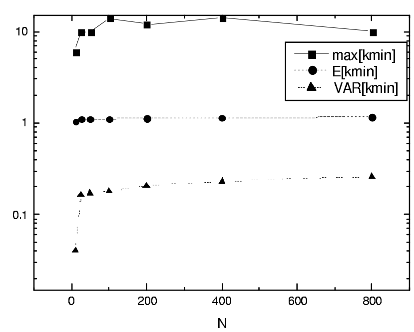

Figure 4: $\mathrm{k_{min}}$-statistics, m=4

In the previous paragraph we indicated that there are two concepts reducing the search-space of SAMCRA. For small m the dominant factor is the non-dominance property, whereas for $m \to \infty$ the constraint values are more dominant. Because these properties are most effective in a certain range of m's, we expect the worst-case behavior to occur with an m that is neither small nor large. Figure 7 shows the k-distribution for different values of m. The best performance is achieved with $m = 2$ and the worst performance is for m around 8. However, as figures 5 and 6 illustrate, $E[k]$ for $m = 8$ is still approximately 1, leading us to believe that for the class of random graphs $G_p(N)$ with independent uniformly distributed weights, the average complexity of SAMCRA is approximately $O(N \log N + mE)$ for every m.

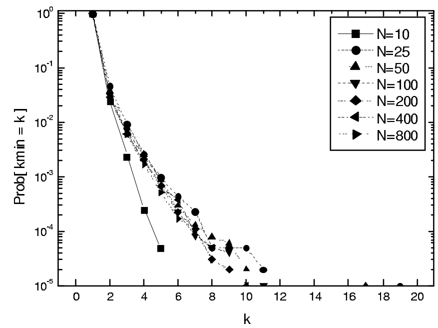

Figure 5: P.d.f. of k_{min}, m=8

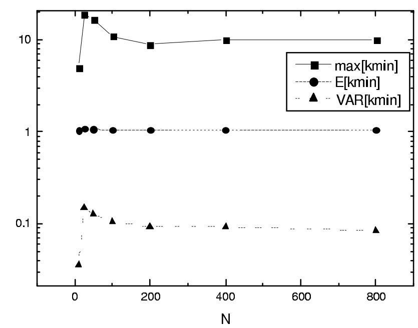

Figure 6: k_{min}-statistics, m=8

Our last simulation concerns the granularity of the constraints/metrics. When the granularity is finite, an upper-bound, in terms of the constraints, on the number of non-dominated paths can be found (3). The finer the granularity, the larger this upper-bound. Figure 8 confirms this behavior (for $N = 20$ and $p = 0.2$). In practice the constraints/metrics have a finite granularity, which according to Figure 8 will positively influence the running-time of SAMCRA.

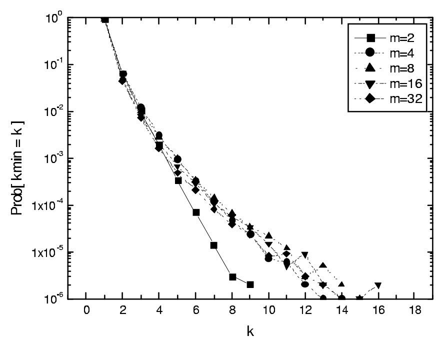

Figure 7: P.d.f. of k_{min}, N=20

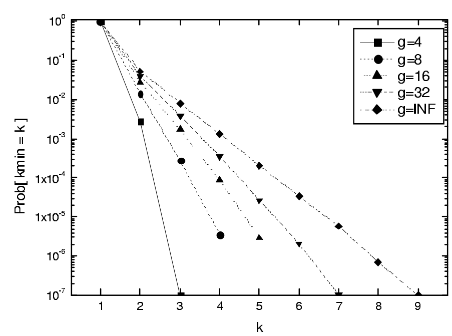

Figure 8: k_{min} for different granularity

Because SAMCRA solves the MCP problem exact, and since the simulations suggest that SAMCRA's average complexity is polynomial for $G_p(N)$ with independent uniformly distributed link weights, the MCP problem for that class of graphs seems, on average, solvable in polynomial time. We have seen that

$E[k] \approx 1$, what indicates that routing, for the sizes N considered, in multiple dimensions is analogous to routing in a single dimension ($m = 1$).

The effect that correlation between the link weights has on the complexity of SAMCRA is topic of further study.

3 The expected hopcount $E[h_N]$ for the random graph $G_p(N)$

As above, we consider the random graph $G_p(N)$ where each link is specified by a weight vector with m independent components possessing the same distribution function

$$F_w(x) = \Pr[w \le x] = x^\alpha 1_{[0,1]}(x) + 1_{(1,\infty)}(x), \quad \alpha > 0 \tag{4}$$

For this network topology, the expected hopcount $E[h_N]$ or the average number of traversed routers along a path between two arbitrarily chosen nodes in the network will be computed. The behavior of the expected hopcount $E[h_N]$ in multiple dimension QoS routing will be related to the single metric case ($m = 1$). That case $m = 1$ has been treated previously in [16], where it has been shown, under quite general assumptions, that

$$E[h_N] \sim \frac{\ln N}{\alpha}$$

$$var[h_N] \sim \frac{\ln N}{\alpha^2}$$

Lemma 2 shows that for $m \to \infty$ in the class of $G_p(N)$ with independent uniformly distributed link weight components, the shortest path is the one with minimal hopcount. Thus the derivation for a single weight metric in [16] for $G_p(N)$ with all link weights 1 is also valid for $m \to \infty$. The first order (asymptotic) calculus as presented in [16] will be extended to $m \ge 2$ for large N. In that paper, the estimate has been proved,

$$\Pr[h_N = k, w_N \le z] \simeq N^{k-1} p^k F_w^{k*}(z),$$

where the distribution function $F_w^{k*}(z)$ is the probability that a sum of k independent random variables each with d.f. F_w is at most z and is given by the k-fold convolution

$$F_w^{k*}(z) = \int_0^z F_w^{(k-1)*}(z-y) f_w(y)\, dy, \quad k \ge 2,$$

and where $F_w^{1*} = F_w$. By induction it follows from (4), that for $z \downarrow 0$,

$$F_w^{k*}(z) \sim \frac{z^{\alpha k} (\alpha \Gamma(\alpha))^k}{\Gamma(\alpha k + 1)}.$$

In multiple (m) dimensions, SAMCRA's definition of the path length (1) requires the maximum link weight of the individual components $w_N(\gamma) = \max_{i=1,\ldots,m} [w_i(\gamma)]$

along some path γ. Since we have assumed that the individual links weight components are i.i.d random variables, and hence $\Pr[w_N \leq z] = (\Pr[w_i \leq z])^m$, this implies for m-dimensions that

$$F_w^{k*}(z) \sim \left[\frac{z^{\alpha k}(\alpha\Gamma(\alpha))^k}{\Gamma(\alpha k+1)}\right]^m$$

such that

$$\Pr[h_N = k, w_N \leq z] \simeq N^{k-1}p^k \left[\frac{z^{\alpha k}\left(\alpha\Gamma(\alpha)\right)^k}{\Gamma(\alpha k+1)}\right]^m$$

We will further confine to the case $\alpha = 1$, i.e. each link weight component is uniformly distributed over $[0,1]$.

$$\Pr[h_N = k, w_N \leq z] \simeq \frac{1}{N}\frac{(Npz^m)^k}{(k!)^m} \tag{5}$$

For a typical value of z, the probabilities in (5) should sum to 1,

$$1 = \frac{1}{N}\sum_{k=1}^{N-1}\frac{(Npz^m)^k}{(k!)^m}$$

At last, for a typical value of z, $\Pr[w_N \leq z]$ is close to unity resulting in

$$\Pr[h_N = k, w_N \leq z] \simeq \Pr[h_N = k]$$

Let us denote with $y = Npz^m$,

$$S_m(y) = \sum_{k=0}^{N-1}\frac{y^k}{(k!)^m} \tag{6}$$

subjected to

$$N+1 = S_m(y) \tag{7}$$

Hence, the typical value y of the end-to-end link weight that obeys (7) is independent on the link density p for large N. Also the average hopcount and the variance can be written in function of $S_m(y)$ as

$$E[h_N] = \frac{y}{N}S'_m(y) \tag{8}$$

$$var[h_N] = \frac{1}{N}\left[y^2 S''_m(y) + yS'_m(y) - \frac{y^2}{N}\left(S'_m(y)\right)^2\right] \tag{9}$$

We will first compute good order approximations for $E[h_N]$ in the general case and only $var[h_N]$ and the ratio $\alpha = \frac{E[h_N]}{var[h_N]}$ in case $m = 2$. Let us further concentrate on

$$V_m(y) = \sum_{k=0}^{\infty}\frac{y^k}{(k!)^m} \tag{10}$$

Clearly, $V_m(y) = \lim_{N\to\infty} S_m(y)$. In the appendix A it is shown in (21) that

$$V_m(y) = A_m(y) \exp\left[my^{1/m}\right] \tag{11}$$

with

$$A_m(y) = \frac{(2\pi)^{\frac{1-m}{2}}}{\sqrt{m}} y^{-\frac{1}{2}\left(1-\frac{1}{m}\right)} \tag{12}$$

After taking the logarithmic derivative of relation (11), we obtain

$$V_m'(y) = V_m(y)\left[\frac{A_m'(y)}{A_m(y)} + y^{\frac{1}{m}-1}\right]$$

In view of (7), y is a solution of $V_m(y) \sim N$, such that the average (8) becomes

$$E\left[h_N\right] \sim \frac{y}{N} V_m'(y) \sim \frac{V_m(y)}{N}\left[y\frac{A_m'(y)}{A_m(y)} + y^{\frac{1}{m}}\right]$$

or

$$E\left[h_N\right] \sim y^{\frac{1}{m}} + y\frac{A_m'(y)}{A_m(y)} \tag{13}$$

Using (12) in (13), we arrive at

$$E\left[h_N\right] \sim y^{\frac{1}{m}} - \frac{1}{2}\left(1-\frac{1}{m}\right) \tag{14}$$

where y is a solution of $V_m(y) \sim N$. Equivalently, y is a solution of

$$r(y) = \ln\left[\frac{(2\pi)^{\frac{1-m}{2}}}{\sqrt{m}}\right] - \frac{1}{2}\left(1-\frac{1}{m}\right)\ln y + my^{1/m} - \ln N = 0$$

As initial value in Newton-Raphson's method, for large N, we start with $y_0 = \left(\frac{\ln N}{m}\right)^m$. The next iteration is $y = y_0 + h$, where $h = -\frac{r(y_0)}{r'(y_0)}$. Since

$$r'(y) = \frac{1}{y}\left[-\frac{1}{2}\left(1-\frac{1}{m}\right) + y^{1/m}\right]$$

we have, with

$$Q = \frac{1}{2}\ln m - \frac{1}{2}(m-1)\ln\left(\frac{m}{2\pi}\right),$$

that

$$\begin{aligned} h &= -\left(\frac{\ln N}{m}\right)^m \frac{-Q - \frac{1}{2}(m-1)\ln(\ln N)}{\left[-\frac{1}{2}\left(1-\frac{1}{m}\right) + \frac{\ln N}{m}\right]} \\ &\sim \left(\frac{\ln N}{m}\right)^{m-1}\left[\frac{1}{2}(m-1)\ln(\ln N) + Q + O\left(\frac{\ln(\ln N)}{\ln N}\right)\right] \end{aligned}$$

and

$$y \sim \left(\frac{\ln N}{m}\right)^m + \left(\frac{\ln N}{m}\right)^{m-1} \left[\frac{1}{2}(m-1)\ln(\ln N) + Q\right] + O\left(\ln(\ln N)\ln^{m-2} N\right)$$

From this asymptotic expression for y, we compute

$$y^{\frac{1}{m}} \sim \frac{\ln N}{m} + \frac{1}{2}\left(1 - \frac{1}{m}\right)\ln(\ln N) + \frac{Q}{m} + O\left(\frac{\ln(\ln N)}{\ln N}\right)$$

Finally, the average hopcount follows from (14) as

$$E[h_N] \sim \frac{\ln N}{m} + \frac{1}{2}\left(1 - \frac{1}{m}\right)\ln(\ln N) + \frac{\ln m}{2m} - \frac{1}{2}\left(1 - \frac{1}{m}\right)\left(\ln\left(\frac{m}{2\pi}\right) + 1\right) + O\left(\frac{\ln(\ln N)}{\ln N}\right) \tag{15}$$

This formula indicates that, to a first order, $m = \alpha$. The simulations (Figures 9, 10, 11) show that, for higher values of m, the expectation of the hopcount tends slower to the asymptotic $E[h_N]$-regime given by (15).

In order to compute the variance, higher order terms in (21) are needed. Although higher order terms can be computed, we confine ourselves to the case $m = 2$, for which $V_2(y) = \sum_{k=0}^{\infty} \frac{y^k}{(k!)^2} = I_0(2\sqrt{y})$ where $I_0(z)$ denotes the modified Bessel function of order zero [1, sec. 9.6]. The variance of the hopcount from (9) with $S_m''(y) = \frac{d^2 I_0(2\sqrt{y})}{dy^2} = \frac{I_0(2\sqrt{y})}{y} - \frac{I_1(2\sqrt{y})}{y\sqrt{y}}$

$$\begin{aligned} var[h_{N,2}] &\sim \frac{y}{N} I_0(2\sqrt{y}) - \frac{\sqrt{y}}{N} I_1(2\sqrt{y}) + E[h_{N,2}] - (E[h_{N,2}])^2 \\ &\sim y - (E[h_{N,2}])^2 \end{aligned}$$

At this point, we must take the difference between $I_0(x)$ and $I_1(x)$ into account otherwise we end up with $var[h_N] \sim 0$. For large x,

$$I_0(x) \sim \frac{e^x}{\sqrt{2\pi x}}\left(1 + \frac{1}{8x} + O\left(x^{-2}\right)\right)$$

and

$$I_1(x) \sim \frac{e^x}{\sqrt{2\pi x}}\left(1 - \frac{3}{8x} + O\left(x^{-2}\right)\right)$$

such that

$$I_1(x) \sim I_0(x)\left(1 - \frac{1}{2x} + O\left(x^{-2}\right)\right)$$

$$\begin{aligned} E[h_{N,2}] &\sim \frac{y}{N} I_1(2\sqrt{y})\frac{1}{\sqrt{y}} \sim \frac{I_0(2\sqrt{y})}{N}\left(\sqrt{y} - \frac{1}{4} + O\left(y^{-1}\right)\right) \\ &\sim \frac{\ln(N)}{2} + \frac{\ln(\ln(N))}{4} - \frac{1}{4} + O\left(\frac{1}{\ln(N)}\right) \end{aligned} \tag{16}$$

Thus,

$$var\left[h_{N,2}\right] \sim y - \left(\sqrt{y} - \frac{1}{4}\right)^2 = \frac{\sqrt{y}}{2} - \frac{1}{16} + O\left(\frac{1}{\sqrt{y}}\right) \qquad (17)$$

and

$$\alpha = \frac{E\left[h_{N,2}\right]}{var\left[h_{N,2}\right]} \sim 2 - \frac{1}{4\sqrt{y}} + O\left(y^{-1}\right) \sim 2 - \frac{\sqrt{2}}{4\sqrt{\ln N}} + O\left(\frac{1}{\ln(N)}\right) \qquad (18)$$

This corresponds well with the simulations shown in Figure 9. In addition, the average and variance of the hopcount for $m = 2$ dimensions scales with N in a similar fashion as the same quantities in $G_p(N)$ with a single link weight, but polynomially distributed with $F_w\left[w \leq x\right] = x^2$.

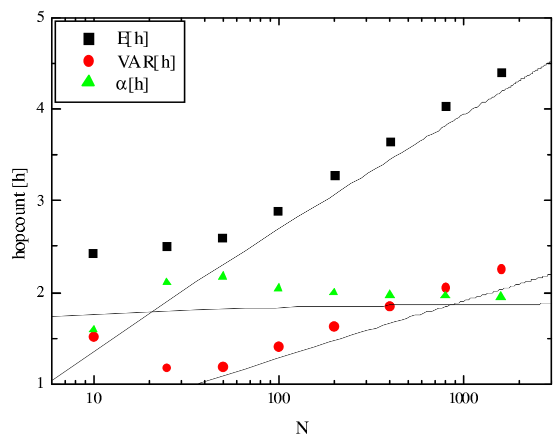

Figure 9: The average $E\left[h_{N,2}\right]$, the variance $var\left[h_{N,2}\right]$ and the ratio $\alpha = \frac{E[h_{N,2}]}{var[h_{N,2}]}$ of the shortest path found by SAMCRA, as a function of the size of the random graph N with two link metrics ($m = 2$). The full lines are the theoretical asymptotics

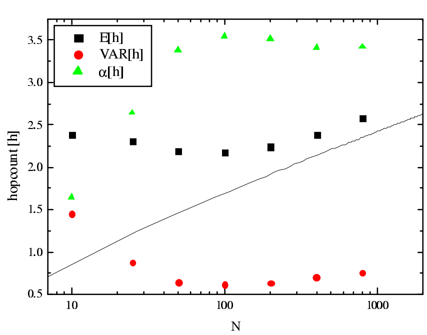

Figure 10: Hopcount statistics for m=4

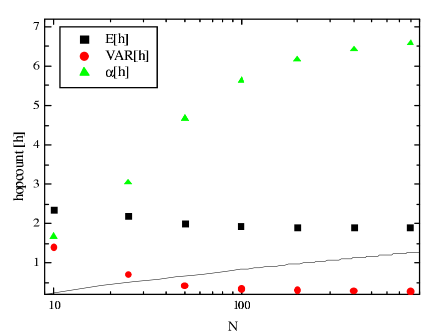

Figure 11: Hopcount statistics for m=8

In summary, the asymptotic analysis reveals that, for random graphs of the class $G_p(N)$ with uniformly (or equivalent exponentially) distributed, independent link weight components, the hopcount in m dimensions behaves similarly as in the random graph $G_p(N)$ in $m = 1$ dimension with polynomially distributed link weights specified via (4) where the polynomial degree α is precisely equal to the multiple dimension m. This result, independent of the simulations of the complexity of SAMCRA, suggests a transformation of shortest path properties in multiple dimensions to the single parameter routing case, especially when the link weight components are independent. As argued in [16], the dependence of the hopcount on a particular topology is less sensitive than on the link weight structure, which this analysis supports.

4 Conclusions

Since constrained-based routing is an essential building block for a future QoS-aware network architecture, we have proposed a multiple constraints, exact routing algorithm called SAMCRA. Although the worst-case complexity is NP-complete (which is inherent to the fact that the multiple constraints problem is NP-complete), a large amount of simulations on random graphs with independent link weight components seem to suggest that the average-case complexity is polynomial. For that considered class, the MCP problem thus seems tractable.

The second part of this paper was devoted the study of the hopcount in multiple dimensions as in QoS–aware networks. For random graphs of the class $G_p(N)$ with uniformly (or equivalent exponentially) distributed, independent link weight components, a general formula for the expected hopcount in m dimensions has been derived and only extended to the variance $var[h_N]$ as well in $m = 2$ dimensions, in order to compute the variance and the ratio of the expected hopcount and its variance. To first order, with the network size $N >> m$ large enough, the expected hopcount behaves asymptotically similar as the expected hopcount in $m = 1$ dimension with a polynomial distribution function $(x^\alpha 1_{[0,1]}(x) + 1_{(1,\infty)}(x))$ and polynomial degree $\alpha = m$.

Both the complexity analysis as the hopcount computation suggests that for a special class of networks, namely random graphs of the class $G_p(N)$ with uniformly (or equivalent exponentially) distributed and independent link weight components, the QoS routing problem exhibits features similar to the one dimensional (single parameter) case. The complexity analysis suggested this correspondence for small N, whereas the asymptotic analysis for the hopcount revealed the connection for $N \to \infty$. Hence, the question arises whether the QoS routing problem in particular classes of graphs may possess a polynomial, rather than non-polynomial *worst* case complexity. And, further, what is the influence of the correlation structure between the link weight components because simulations suggest that independence of these link weight components seems to destroy NP-completeness. Moreover, we notice that the proof presented in [20] strongly relies on the choice of the link weights. At last, if our claims about the NP-completeness would be correct, how large is then the class of networks that really lead to an NP-complete behavior of the MCP problem? In view of the large amount of simulations performed over several years by now, it seems that this last class fortunately must be small, which suggests that, in practice, the QoS-routing problems may turn out to be feasible.

References

1. Abramowitz, M. and I.A. Stegun (eds.), *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*, Dover Pubns, June 1974, ISBN: 0486612724.
2. Barndorff-Nielsen, O. and M. Sobel, *On the distribution of the number of admissible points in a vector random sample*, Theory of Probability and its Applications, vol. 11, no. 2, 1966.
3. Bentley, J.L., H.T. Kung, M. Schkolnick and C.D. Thompson., *On the Average Number of Maxima in a Set of Vectors and Applications*, Journal of ACM, vol. 25, no. 4, pp. 536-543, 1978.
4. Bollobas, B., *Random Graphs*, Academic Press, London, 1985.
5. Chong, E.I., S. Maddila, S. Morley, *On Finding Single-Source Single-Destination k Shortest Paths*, J. Computing and Information, 1995, special issue ICCI'95, pp. 40-47.
6. Cormen, T.H., C.E. Leiserson and R.L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, 2000.
7. De Neve, H. and P. Van Mieghem, *A multiple quality of service routing algorithm for PNNI*, IEEE ATM workshop, Fairfax, May 26-29, 1998, pp. 324-328.
8. De Neve, H. and P. Van Mieghem, *TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm*, Computer Communications, 2000, vol. 23, pp. 667-679.
9. Edwards, H. M., *Riemann's Zeta Function*, Academic Press, New York, 1974.
10. Garey, M.R. and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.
11. Guo, L. and I. Matta, *Search space reduction in QoS routing*, Proc. of the 19th III Int. Conference on Distributed Computing Systems, III, May 1999, pp. 142-149.
12. Henig, M.I., *The shortest path problem with two objective functions*, European J. of Operational Research, 1985, vol. 25, pp. 281-291.
13. Korkmaz, K. and M. Krunz, *Multi-Constrained Optimal Path Selection*, IEEE INFOCOM 2001.
14. Levin, L.A., *Average Case Complete Problems*, SIAM J. Comput., 1986, 15(1):285-286.
15. Titchmarsh, E. C., *The Theory of Functions*, Oxford University Press, 1964.
16. Van Mieghem, P., G. Hooghiemstra and R. van der Hofstad, *A Scaling Law for the Hopcount in Internet*, Delft University of Technology, report 2000125, 2000, http://wwwtvs.et.tudelft.nl/people/piet/telconference.html.
17. Van Mieghem, P., H. De Neve and F.A. Kuipers, *Hop-by-Hop Quality of Service Routing*, to appear in Computer Networks, 2001.
18. Van Mieghem, P., *Paths in the simple Random Graph and the Waxman Graph*, to appear in Probability in the Engineering and Informational Sciences (PEIS), 2001.
19. Van Mieghem, P., *The Asymptotic Behaviour of Queueing Systems: Large Deviations Theory and Dominant Pole Approximation*, Queueing Systems 23, 27-55, 1996.

20. Wang, Z. and J. Crowcroft, 1996, *QoS Routing for supporting Multimedia Applications*, IEEE JSAC, 14(7): 1188-1234, September 1996.

A Asymptotic formula for $V_m(y)$

We will apply the Euler-Maclaurin formula

$$\sum_{n=a}^{b} f(n) = \frac{1}{2}\left(f(a)+f(b)\right) + \int_a^b f(t)\,dt$$
$$+\sum_{n=1}^{N}(-1)^{n-1}\frac{B_{2n}}{(2n)!}\left[f^{(2n-1)}(b)-f^{(2n-1)}(a)\right]+R_N \qquad (19)$$

where B_n are the Bernoulli numbers [1, sec. 23] and

$$R_N = \frac{1}{(2N+1)!}\int_a^b B_{2N+1}(u)\, f^{(2N+1)}(u)\,du$$

Comparison with (10) indicates that $a=0$, $b=\infty$ and $f(x)=\frac{y^x}{(\Gamma(x+1))^m}$. The derivatives at b all vanish and the derivatives at a follow from the Taylor series of $f(x)$ around $x=0$,

$$\frac{e^{x\ln y}}{(\Gamma(x+1))^m} = \sum_{k=0}^{\infty} f_k x^k$$

Thus,

$$V_m(x) = \frac{1}{2} + \int_0^\infty \frac{y^x\,dx}{(\Gamma(x+1))^m} - \sum_{n=1}^{N}(-1)^{n-1}\frac{B_{2n}}{(2n)} f_{2n-1} + R_N$$

Let us first concentrate on the integral

$$I(y) = \int_0^\infty \frac{y^x\,dx}{(\Gamma(x+1))^m}$$
$$= \int_0^\infty \exp\left[x\ln y - m\ln\Gamma(x+1)\right]dx$$

We will approximate $I(y)$ by the method of the steepest descent for which an exact expansion was published earlier in [19]. Here, we confine ourselves to the leading order term,

$$\int_{-\infty}^{\infty} e^{x[f_1(z_0)\,z-f(z)]}\,dz = \frac{e^{-x[f_0(z_0)-z_0 f_1(z_0)]}}{\sqrt{x f_2(z_0)}}\left(\sqrt{\pi}+O\left(x^{-1}\right)\right) \qquad (20)$$

where $f_k(z_0)$ are the Taylor coefficients of $f(z)$ around $z=z_0$. The fastest convergence is expected if z_0 is the maximum of f. Applied to $I(y)$, the maximum of $f(x)$ is the same as the maximum of $g(x) = -x \ln y + m \ln \Gamma(x+1)$. Since $g'(x) = -\ln y + m\psi(x+1)$ where the digamma function equals $\psi(x+1) = \frac{\Gamma'(x+1)}{\Gamma(x+1)}$, we find that the maximum x_0 obeys

$$\psi(x_0+1) = \frac{\ln y}{m}$$

or for large y using the asymptotic expansion for the digamma function [1, 6.3.18],

$$\ln(x_0+1) - \frac{1}{2(x_0+1)} \sim \frac{\ln y}{m}$$

from which $x_0 \sim y^{1/m}$ and

$$\begin{aligned} g(x_0) &\sim -y^{1/m} \ln y + m \ln \Gamma(y^{1/m}+1) \\ &\sim -my^{1/m} + \frac{1}{2} \ln y + \frac{m}{2} \ln 2\pi \end{aligned}$$

The higher order derivatives at x_0 of $g(x)$ are polygamma functions $\psi^{(n)}$ [1, 6.4],

$$g^{(n)}(x_0) = m\psi^{(n-1)}(x_0+1)$$

The second derivative $g''(x_0) = m\psi'(x_0+1) \sim \frac{m}{x_0+1} + O\left(x_0^{-2}\right)$. Application of (20) yields

$$\begin{aligned} I(y) &\sim \frac{e^{x_0 \ln y - m \ln \Gamma(x_0+1)}}{\sqrt{\frac{m}{2}\psi^{(1)}(x_0+1)}} \sqrt{\pi} \\ &\sim \frac{e^{my^{1/m} - \frac{1}{2}\ln y - \frac{m}{2}\ln 2\pi}}{\sqrt{\frac{m}{2y^{1/m}}}} \sqrt{\pi} \end{aligned}$$

or

$$V_m(y) \sim \frac{(2\pi)^{\frac{1-m}{2}}}{\sqrt{m}} y^{-\frac{1}{2}\left(1-\frac{1}{m}\right)} e^{my^{1/m}} \tag{21}$$

It is well known [9] that in the n-sum in (19), the first neglected term is of the same order as the remainder R_N. It is readily verified by executing the Cauchy product that $f_k \sim O\left(\ln^k y\right)$ and, hence, negligible with respect to $I(y)$ for large y. Hence, for large y, we arrive at (21). This asymptotic expansion reduces for $m=2$ to that of the modified Bessel functions $I_v(x) \sim \frac{e^x}{\sqrt{2\pi x}}$ for $x = 2\sqrt{y}$.