RZ 3383 (# 93429) 12/10/01
Computer Science 5 pages

Research Report

A New Minimum-Interference Routing Algorithm Based on Flow Maximization

Daniel Bauer

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

A New Minimum-Interference Routing Algorithm Based on Flow Maximization

Daniel Bauer

IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract

A new on-line routing algorithm based on the notion of minimum interference is presented. The algorithm maximizes the sum of residual flows of ingress–egress pairs, using a simple heuristic method. It achieves good results in terms of total bandwidth routed.

1 Introduction

This research note presents a new on-line routing algorithm that can be used for dynamic bandwidth provisioning in connection oriented networks such as ATM or MPLS. On-line routing algorithms compute bandwidth guaranteed connections, where connection setup requests arrive one-by-one and future demands are unknown. They determine a path through the network that provides sufficient bandwidth to satisfy a request.

Ordinary routing algorithms compute paths from a source to a destination based on the current network load and the requested bandwidth. However, such a path easily consumes resources which are crucial for other source-destination pairs. Often, this results in a poor overall utilization of the network resources.

More sophisticated algorithms, called minimum-interference routing algorithms, use the information regarding source-destination pairs. They find paths that do not "interfere too much" with paths that may be critical for future requests. In this research note, we present a new minimum-interference routing algorithm that achieves competitive performance in terms of network resource utilization and outperforms traditional algorithms. Compared to other minimum-interference algorithms such as SMIRA [1], the new approach does not require any parameter tuning.

2 Mathematical Formulation

In the following mathematical formulation, which has been adapted from [2], minimum interference routing is described as a weighted-sum-maximization problem. Let $G(N, L, B)$ describe the given network, where N is the set of routers (nodes) and L the set of directed links (edges) and B the capacity of the links. Let n denote the number of nodes and m denote the number of links in the network. The ingress–egress pairs are given by set $\mathcal{P}$ and the number of pairs is given by p. Let M represent the node-arc incidence matrix. Each row in this matrix corresponds to a node in the graph and each column corresponds to a link. Each column as exactly two non-zero entries, a $+1$ in the row of the source node and a -1 in the row of the destination node. Let x^{sd} be an m-vector denoting the flow of pair $(s, d) \in \mathcal{P}$. Each element in x^{sd} represents the flow on a link. Let θ_{sd} represent a scalar that is the maximum flow that can be sent from source s to destination d in the network. Let R be an m-vector of residual capacities. The residual capacities are initialized to B and subsequently reduced as resources for path x^{ab} are allocated. Let e^{sd} represent an n-vector with a $+1$ in position d and a -1 in position s. Let $(a, b) \in \mathcal{P}$ be the ingress–egress pair for which a demand of D units is to be routed in such a way that the sum of the flows among the other ingress–egress pairs is maximized. This problem is called WSUM-MAX and can be described as follows:

$$\max \sum_{(s,d)\in\mathcal{P}\backslash(a,b)} \alpha_{sd}\theta_{sd} \tag{1}$$

$$\begin{aligned}
Mx^{sd} &= \theta_{sd}e^{sd} && \forall (s,d) \in \mathcal{P}\backslash(a,b) && (2)\\
Mx^{ab} &= De^{ab} && && (3)\\
x^{sd} + x^{ab} &\leq R && \forall (s,d) \in \mathcal{P}\backslash(a,b) && (4)\\
x^{sd} &\geq 0 && \forall (s,d) \in \mathcal{P}\backslash(a,b) && (5)\\
x^{ab} &\in \{0, D\}^m && && (6)\\
& && && (7)
\end{aligned}$$

Equation 1 defines the optimization goal, which is to maximize the weighted sum of the maximum flow values. Equation 2 formulates the maximum flow problem for all ingress–egress nodes except pair (a, b). It makes sure that the source is the only producer and the destination is the only consumer of flow. Equation 3 states that D units of bandwidth are flowing from ingress a to egress b. The rest of the equations formulate necessary constraints. Equation 4 ensures that the capacity constaints in the residual networks are met. Equation 5 makes sure that all flow values are non-negative. Finally, Equation 6 defines that demand D is routed from a to b on a single path; i. e. that the demand is not split among multiple paths. This last equation makes the problem NP-hard.

3 Solution Approach

The solution approach is to separate the max-flow problem and the problem of finding a path. Those two problems are coupled by Equation 4. We remove summand x^{ab} from this equation and we obtain $p-1$ independent max-flow problems. These problems can be solved using a traditional max-flow computation algorithm, such as [3] or [4]. As a result, we obtain an x^{sd} vector for each $(s,d) \in \mathcal{P}$. In a second step, we tie up x^{ab} to the max-flow problems by formulating the following linear optimization:

$$\max \left(\sum_{(s,d)\in\mathcal{P}\setminus(a,b)} (R - x^{sd} - x^{ab}) \cdot \bar{1} \right) \tag{8}$$

$$Mx^{ab} = \quad De^{ab} \tag{9}$$
$$x^{ab} \leq \quad R \tag{10}$$
$$x^{ab} \in \quad \{0, D\}^m \tag{11}$$
$$\tag{12}$$

Equation 8 is the optimization goal, which is to find a path from a to b such that the bandwidth in the network, after the flows x^{sd} are considered and after x^{ab} is routed, is maximized. $R - x^{sd}$ denotes the bandwidth in the flow-residual network for flow x^{sd}. The summation makes sure that the flow of each ingress–egress pair is considered. The vector dot product ensures that it is done for each link. This is an approximation of the WSUM-MAX problem. Let $\hat{\theta_{sd}}$ denote the max-flow values that are obtained by solving the $p-1$ independent max-flow problems. For $D = 0$, the max-flow values θ_{sd} in WSUM-MAX are equal to $\hat{\theta_{sd}}$. As D increases, the flow values θ_{sd} remain equal to $\hat{\theta_{sd}}$ if $x^{sd} + x^{ab} \leq R_i$. Above approximation finds a path x^{ab} that satisfies these capacity constraints, if such a path exists. If no such path exists, then above approximation finds a path the minimizes the 'overbooking'.

Note that above approximation can be computed using a shortest-path algorithm that uses link costs given by vector c as follows:

$$c = \sum_{(s,d)\in\mathcal{P}\setminus(a,b)} (x^{sd} - R)$$

4 Results

In this section, we compare the performance of the WSUM-MAX approximation with two SMIRA-type algorithms MI-BLA and MI-PA [1] and with the traditional shortest-widest-path (SWP) and widest-shortest-path (WSP). The experiments are carried out using the network topology used in [1] as depicted in Figure 1. Links are bi-directional with a capacity of 1200 units (thin lines) and 4800 units (thick lines). Each link l is assigned a static cost of one unit. The network contains the four ingress–egress pairs (S1→D1), (S2→D2), (S3→D3), (S4→D4). Path requests are limited to those pairs only.

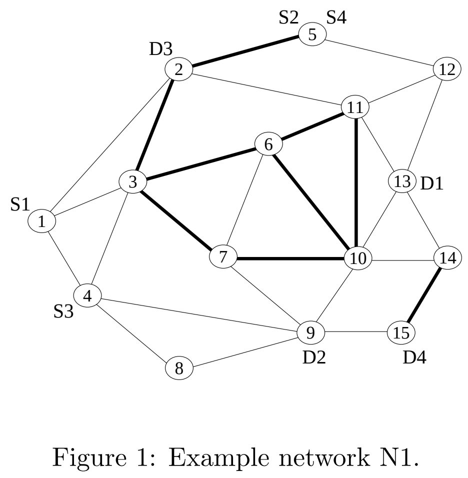

Figure 1: Example network N1.

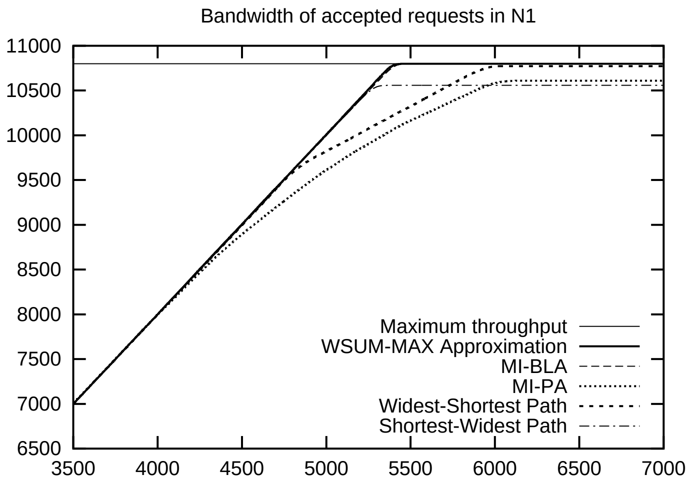

Figure 2: Throughput of accepted requests using demands of 1 to 3 in N1.

In a first experiment, network N1 is loaded with 7000 requests. The bandwidth demand of each request is uniformly distributed in the range of 1 to 3 units (only integer values are used).

The bandwidth of accepted requests of experiment 1 is shown in Figure 2. For each of the algorithm, the bandwidth increases with the number of requests until a saturation point is reached at which no more requests can be accommodated, and the network is saturated. SWP shows the weakest performance, with a saturation point around 10550 bandwidth units. The best performance is shown by the WSUM-MAX approximation and MI-BLA with 10800 units, followed closely by WSP with 10770 units, and MI-PA with 10610 units. Note that the theoretical maximum also is 10800 units [1].

In a second experiment, we increase the number of ingress–egress pairs and therefore increase the possibility of "interference". Figure 3 shows the example network after two additional ingress–egress pairs have been added. We refer to this network as network N1+. The link costs are one unit. As in the previous experiments, the requests are uniformly distributed among the six ingress–egress pairs and the bandwidth demand is an integer value uniformly distributed in the range from 1 to 3. Due to the increased number of ingress–egress pairs, the theoretical maximum throughput for greedy algorithms increases to 13600 units [1].

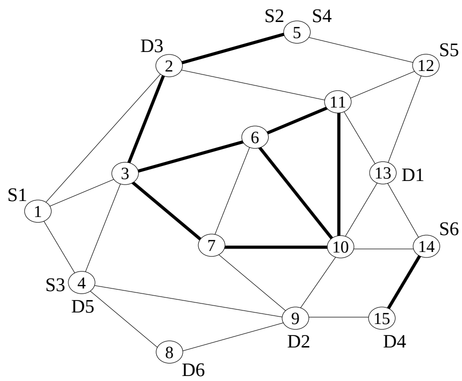

Figure 3: Example network N1+ with additional ingress–egress nodes.

Figure 4 shows the bandwidth of accepted requests in N1+. The best performance is achieved by the WSUM-MAX approximation and MI-PA, which reach a total throughput of 13340 resp. 13330 units. MI-BLA follows next with 13040 units, WSP and SWP are last with 12610 and 12130 units, respectively.

In a third experiment, the static link costs are changed to be inversely proportional to the capacity, i.e. high capacity links are assigned a static cost of one unit and low capacity links are assigned a static cost of four units. We refer to this network as N2+. Figure 5 shows the results. The best result is achieved by

MI-PA, which reaches the theoretical maximum of 13600 units. WSP and the WSUM-MAX approximation follow with 13405 and 13375 units of bandwidth, respectively. MI-BLA reaches 13035 units and SWP is last with 12575 units.

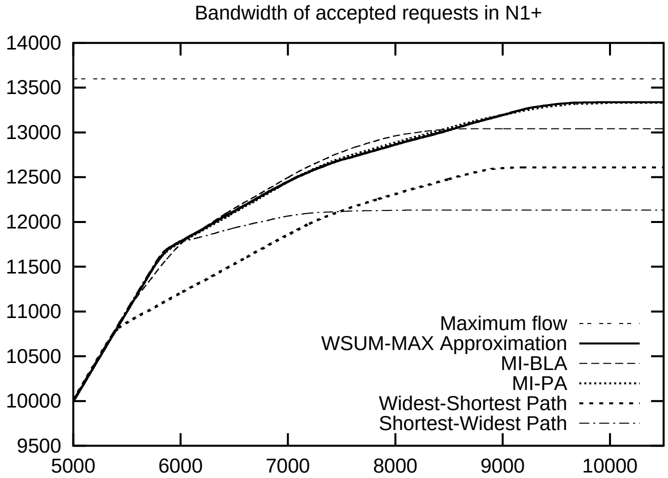

Figure 4: Throughput of accepted requests using demands of 1 to 3 in N1+.

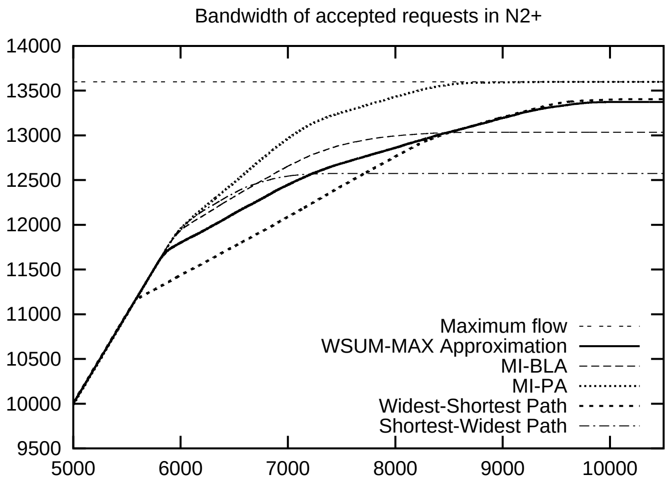

Figure 5: Throughput of accepted requests using demands of 1 to 3 in N2+.

5 Conclusion

The experiments showed that the WSUM-MAX approximation achieves competitive results. In the first experiment, the WSUM-MAX approximation was able to achieve the optimal performance. In the second experiment, the algorithm achieved the best performance, together with MI-PA. In the third experiment, WSUM-MAX was second best.

The main advantage of the WSUM-MAX approximation compared to MI-PA and MI-BLA is the fact that WSUM-MAX is parameterless and does not require any tuning. This is in contrast to the SMIRA algorithm family, which require several parameters to be set. MI-BLA and MI-PA are two examples of different parameter settings. While MI-PA achived the optimal performance in experiment 1, it showed a rather weak performance in the other experiments. The same can be said for MI-BLA, which performed poorly in experiment 1 but excellent in the others.

References

[1] I. Iliadis and D. Bauer. A new class of on-line minimal interference routing algorithms. Technical Report 3379, IBM Research, October 2001.

[2] K. Kar, M. Kodialam, and T.V. Lakshman. Minimum interference routing of bandwidth guaranteed tunnels with mpls traffic engineering applications. *IEEE Journal on Selected Areas in Communications*, 18(12):2566–2579, December 2000.

[3] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum flow problem. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 136–146, 1986.

[4] L. Ford and D. Fulkerson. Flows in networks. Princeton University Press, Princeton, NJ, 1965.