Ilmari Juva. 2007. Robust load balancing. In Proceedings of the 50th Annual IEEE Global Communications Conference (GLOBECOM 2007), Washington D.C., United States, 26-30 November 2007, pages 2708-2713.

Robust Load Balancing

Ilmari Juva
Helsinki University of Technology, Finland
ilmari.juva@tkk.fi

***Abstract*— We study the problem of balancing the traffic load in a network by route selection. The traditional approach starts from a given traffic matrix. However, as the traffic matrix is seldom available, estimated traffic matrices have to be used. Thus, the solution of the load balancing algorithm is not optimal due to errors in traffic matrix estimate. In order to overcome this, we present two variations of a robust load balancing scheme. One where no knowledge of the traffic matrix is needed, and a novel variant of the robust algorithm that does require a traffic matrix estimate but takes into consideration the estimation error involved. We compare the performance of these methods to that of traditional load balancing by a simulation study.**

Keywords: Load Balancing, Traffic Matrix.

I. Introduction

Load balancing is a common traffic engineering task in which the traffic in the network is routed in a way that optimizes some performance criterion. In this paper we use the minimization of the maximal link utilization in the network as the target. This criterion leads to network usage which minimizes the relative congestion on the links throughout the network.

Traffic is moved from heavily congested links to other routes and links in order to ease the congestion in that part of the network. To do this, the traffic matrix is typically assumed to be known. However, traffic matrices are not generally readily obtainable. Estimated traffic matrices obviously are not entirely accurate, but come with some estimation error. Thus, the actual traffic load on a given link might significantly differ from what is expected on the basis of to the estimated traffic matrix.

On the other hand, a method which does not require knowledge of the traffic matrix, or takes into consideration the estimation errors in the traffic matrix estimate, would be more robust. The basic idea of such a method is to balance the load in a way that does not optimize the network utilization for a single traffic matrix, but for a large polytope of matrices, which is selected so that it surely includes also the real traffic matrix. This kind of robust approach to traffic engineering has therefore gained interest in recent years. In particularly with regard to provisioning for Virtual Private Networks (VPN). In the hose model proposed in [1] and used by [2] each VPN endpoint specifies bounds for its traffic demand, and the provisioning is done so that there is sufficient bandwidth for any traffic matrix that is consistent with the specified bounds. Ben-Ameur and Kerivin [3] generalize the hose model and introduce the concept of routing a polytope. Johansson [4] proposes the use of this concept for load balancing in the network without a traffic matrix estimate. Applegate and Cohen [5] study the performance of robust routing for different size uncertainty sets.

In this paper we study the performance of robust load balancing methods against traffic matrix based load balancing methods. The traditional method is used with both gravity model and maximum likelihood traffic matrix estimates. For the robust methods, we study two variants. The first variant, the Robust method, defines the demand polytope as the set of all traffic matrices consistent with the link count measurements, as in [4]. Thus, the method does not require any traffic matrix estimate, instead it finds the optimal routing for the entire polytope of possible traffic matrices. We study the average and worst case performance of this approach through an extensive simulation study, and compare it to traditional load balancing methods.

While traditional load balancing relies on a single traffic matrix and the Robust method on a polytope containing all possible traffic matrices, these are obviously extreme points. A middle ground between the two would be a polytope around the estimated traffic matrix, that is smaller than the set of all possible traffic matrices. Surely there are some cases so implausible that we do not need to consider them, and on the other hand it is unlikely that our traffic matrix estimate is so accurate that no error margins are needed. To obtain these bounded polytopes we propose a novel variant of the robust approach, the Bounded Robust Method. We analytically derive statistical standard error for the elements of the traffic matrix estimate and use different confidence intervals to obtain different size polytopes.

The differences in our approach compared to the insightful work in [5] is that we do not assume the actual traffic matrix and the estimated traffic matrix to coincide. We cannot construct the uncertainty set around the real demand, since it is unknown in reality. Only the estimate is available to us. Also, our error margins are statistical confidence intervals as opposed to multiples of the traffic amounts. As these confidence intervals are larger for origin-destination(OD) pairs that are difficult to estimate and smaller for those OD pairs that are easier to estimate, we avoid making the polytope unnecessarily large in directions where there is not that much uncertainty.

The rest of the paper is organized as follows. In the next section we introduce the general framework of the problem and the notation. In Section III the traditional load balancing approach is discussed. As this is based on a known traffic matrix, we briefly review typical traffic matrix estimation methods. Section IV presents the robust load balancing scheme,

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

and in section V we develop the modified Bounded Robust method. Section VI gives results of simulation studies and finally section VII concludes the paper.

II. General setting

We denote the measured link loads, available from SNMP measurements, by $\boldsymbol{y}_0$, which is a vector of length L, the number of links in the network. The elements of the vector are the link load measurements for each link, with the links indexed by l. The link - OD pair incidence matrix, or the routing matrix, is denoted by $L \times K$ matrix $\boldsymbol{A}$, where K is the number of OD pairs in the network, indexed by k. The element $A_{l,k}$ of the routing matrix gives the proportion of the traffic of the kth OD pair that is routed through link l. If the OD pair does not use the link in question, then $A_{l,k} = 0$.

The routing matrix during the measurements is assumed to be known and is denoted by $\boldsymbol{A}_0$. The traffic matrix $\boldsymbol{x}$ is a vector of length K, composed of traffic flows x_k between origin-destination pairs. It must satisfy the link count relation

$$\boldsymbol{A}_0\boldsymbol{x} = \boldsymbol{y}_0. \tag{1}$$

Each non-negative vector $\boldsymbol{x}$ that satisfies the above relation is a possible traffic matrix. Denote the set (polytope) of such traffic matrices by $\mathcal{D}$.

$$\mathcal{D} \quad = \quad \{\boldsymbol{x} \geq 0 : \boldsymbol{A}_0\boldsymbol{x} = \boldsymbol{y}_0\}. \tag{2}$$

Let $\boldsymbol{C}$ denote the link capacity vector. Our performance metric u refers to the relative utilization of the most heavily congested link,

$$u = \max_l \frac{(\boldsymbol{A}\boldsymbol{x})_l}{C_l} = |\boldsymbol{A}\boldsymbol{x}/\boldsymbol{C}|, \tag{3}$$

where in the latter form we have introduced the notation $|\boldsymbol{a}/\boldsymbol{b}| = \max_l a_l/b_l$. In the starting situation the utilization is

$$u_{init} = |\boldsymbol{y}_0/\boldsymbol{C}|, \tag{4}$$

Let $\boldsymbol{L}$ denote the node-link incidence matrix with element $L_{nl} = +1$ and $L_{n'l} = -1$ if (directed) link l leads from node n to node n', and 0 otherwise, while $\boldsymbol{R}$ denotes the node - OD pair incidence matrix with element $R_{n_o,k} = +1$ and $R_{n_d,k} = -1$ if OD pair k enters the network at node n_o and exits at node n_d, and 0 otherwise.

To ensure that all the traffic between nodes gets routed, the routing matrix must satisfy the flow conservation condition

$$\boldsymbol{L}\boldsymbol{A} = \boldsymbol{R}, \tag{5}$$

where the routing matrix $\boldsymbol{A}$ is free variable to be optimized by the load balancing algorithms.

III. Traditional load balancing approach

In the traditional approach, the traffic matrix is assumed to be known. In reality we usually need to estimate it from the available information and the estimate is always somewhat erroneous. In this section we first review traffic matrix estimation methods and then define the traditional load balancing problem.

A. Traffic matrix estimation

Typically the routing matrix and link counts are the only readily available information. We denote the link count measurements by $\boldsymbol{y}_0$ and the routing matrix under which the measurements were performed by $\boldsymbol{A}_0$. In order to be consistent with the link counts the traffic matrix $\boldsymbol{x}$ has to satisfy the link count relation (1). However, solving $\boldsymbol{x}$ from this equation is an underconstrained problem. Thus, some extra information needs to be brought in to the situation.

1) Maximum likelihood method [6]: This approach assumes that there is a relation between the mean and variance of the OD pair counts. In order to make use of the relation, a time-series of measurements is required to obtain a sample covariance of the link counts. The variance is then used through the relation along with the link counts in a maximum likelihood framework to calculate an estimate for the traffic matrix. The fact that we can gain knowledge about the mean from the sample variance makes the problem identifiable. The accuracy of the maximum likelihood estimate (MLE) depends on sample size and the validity of the mean-variance relation.

2) Gravity model method [7], [8]: The method makes the assumption that source and destination are independent and that the traffic between any two nodes is proportional to the total traffic originating from the source node and terminating to the destination node. This assumption can be used to obtain a starting point estimate, which is then used together with the link count information to yield the final estimate.

The accuracy of the method depends strongly on the validity of the gravity model assumption. To obtain a realistic situation we generate synthetic traffic in our simulation study so that the goodness of fit in the sense of sum of squares error between gravity model $\boldsymbol{\lambda}_g$ and actual traffic matrix $\boldsymbol{\lambda}$ is approximately $R^2 = 0.84$. This value was obtained from a study of the Abilene[1] network traffic properties in [9].

B. Load balancing algorithm

With the estimated traffic matrix, denoted by $\hat{\boldsymbol{x}}$, we can formulate the load balancing problem as an LP problem

Problem 1 (Traditional Load Balancing Problem):

$$\min_{\boldsymbol{A} \geq \boldsymbol{0}} u \tag{6}$$

such that

$$\begin{aligned} u\boldsymbol{C} &\geq \boldsymbol{A}\hat{\boldsymbol{x}} && (7) \\ \boldsymbol{L}\boldsymbol{A} &= \boldsymbol{R} && (8) \end{aligned}$$

which yields the routing matrix that minimizes the maximum link utilization. The solution $\boldsymbol{A}(\hat{\boldsymbol{x}})$ is a function of the assumed traffic matrix $\hat{\boldsymbol{x}}$. This solution is optimal only with regard to the maximum link load, and usually is not unique. A second optimization is needed to ensure that traffic is optimally balanced in less loaded links also, not just in the bottleneck link.

$$\min_{\boldsymbol{A} \geq \boldsymbol{0}} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{A}\hat{\boldsymbol{x}} \tag{9}$$

[1]http://www.cs.utexas.edu/users/yzhang/

such that

$$\hat{u}\boldsymbol{C} \geq \boldsymbol{A}\hat{\boldsymbol{x}} \quad (10)$$

$$\boldsymbol{L}\boldsymbol{A} = \boldsymbol{R}, \quad (11)$$

where $\hat{u}$ is the solution of (6).

The above sequential optimization would result in the lowest possible achievable value for u if the traffic matrix estimate would be accurate. In reality, however, the estimate is more or less inaccurate. This introduces error to this approach, which leads to the real value of u being higher than the algorithm lets to believe.

IV. Robust method

In the Robust method, instead of using a fixed traffic matrix estimate, we try to find a routing matrix such that the worst case performance is optimized over all feasible traffic matrices in the set $\mathcal{D}$, defined in equation (2). By finding the routing matrix that performs well over the whole set $\mathcal{D}$ we do not have to estimate the traffic matrix.

This approach is proposed in [4] and uses the algorithm introduced in [3]. Our formulation is different from these in that we use the flow conservation constraints instead of making use of the arc-path formulation. The optimization problem in our framework is

$$\min_{\boldsymbol{A}\geq\boldsymbol{0}} u \quad (12)$$

such that

$$u\boldsymbol{C} \geq \boldsymbol{A}\boldsymbol{x} \qquad \forall \boldsymbol{x}\in\mathcal{D} \quad (13)$$

$$\boldsymbol{L}\boldsymbol{A} = \boldsymbol{R}. \quad (14)$$

Again a second step is needed to ensure that traffic is optimally balanced in less saturated links also.

$$\min_{\boldsymbol{A}\geq\boldsymbol{0}} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{A}\,\boldsymbol{e} \quad (15)$$

such that

$$\hat{u}\boldsymbol{C} \geq \boldsymbol{A}\boldsymbol{x} \qquad \forall \boldsymbol{x}\in\mathcal{D} \quad (16)$$

$$\boldsymbol{L}\boldsymbol{A} = \boldsymbol{R}. \quad (17)$$

Equation (13) defines the link constraints, stating that the maximal link utilization u has to satisfy the link constraints for each traffic matrix $\boldsymbol{x}$ belonging to the polytope $\mathcal{D}$. Equation (14) defines the flow conservation constraints.

This problem is difficult to solve because of the infinite number of constraints in (13). Therefore, we use the demand satellite approach, which was part of the iterative solution by Ben-Ameur and Kerivin [3]. The problem is divided to two optimization problems. The first one is the link load optimization problem. It does not consider the whole set $\mathcal{D}$. This set of infinite number of constraints is substituted by a finite set of constraints D^*, which are generated by the second optimization problem. The link load optimization is now a simple LP problem, with the flow constraints (20) just as (8) in the traditional approach, and the set of constraints (19) in the place of the link constraints (7).

Problem 2 (Link Load Optimization):

$$\min_{\boldsymbol{A}\geq\boldsymbol{0}} u \quad (18)$$

such that

$$u\,\boldsymbol{C}_l \geq (\boldsymbol{A}\boldsymbol{x})_l \qquad \forall (l,\boldsymbol{x})\in D^* \quad (19)$$

$$\boldsymbol{L}\boldsymbol{A} = \boldsymbol{R}, \quad (20)$$

and such that a secondary objective function is used to obtain optimal balancing throughout the network.

The set D^* is initially empty. Problem 3 is solved to obtain constraints. The iteration can be started using the initial routing $\boldsymbol{A}_0$. For each link we find the traffic matrix $\boldsymbol{x}^*\in\mathcal{D}$ that maximizes the traffic on that link. If the achieved link utilization is larger than the current value for u, the corresponding constraint

$$u\,\boldsymbol{C}_l \geq (\boldsymbol{A}\boldsymbol{x}^*)_l$$

is added to the set D^* to be used as a constraint in (19) in the next iteration of problem 2. As the constraint can be identified by the pair $(l,\boldsymbol{x}^*)$ we denote

$$D^* \leftarrow D^* \cup (l,\boldsymbol{x}^*).$$

The problem is formulated as follows.

Problem 3 (Constraint Generation): For each link l solve with current values $\boldsymbol{A}^{(i)}$, $u^{(i)}$

$$\boldsymbol{x}^* = \arg\max_{x\in\mathcal{D}} (\boldsymbol{A}^{(i)}\boldsymbol{x})_l$$

If

$$u^{(i)}\,\boldsymbol{C}_l < (\boldsymbol{A}^{(i)}\boldsymbol{x}^*)_l$$

then

$$D^* \leftarrow D^* \cup (l,\boldsymbol{x}^*).$$

The iteration starts by generating constraints for the initial routing matrix $\boldsymbol{A}_0$. Problem 2 is then solved with these constraints. This yields a new routing matrix, denoted by $\boldsymbol{A}^{(i)}$ in the ith iteration. More constraints are generated using this routing matrix and these constraints are added to the set of constraints along with the previous constraints. Then the Link Load Optimization problem is solved again. The iteration continues until the Constraint Generation problem does not yield any new constraints. The iteration is guaranteed to stop, as each generated new constraint corresponds to an extreme point of the polytope $\mathcal{D}$, which, of course, constitute a finite set. On a 12 node test topology the number of constraints in D^* is typically between 500 and 1000.

V. Bounded Robust method

In this section we introduce a novel approach for load balancing, the Bounded Robust Method, which combines the idea of the Robust method with the use of the traffic matrix estimate.

In Problem 3 we find the traffic matrix $\boldsymbol{x}^*\in\mathcal{D}$ that maximizes the traffic on a particular link. This traffic matrix is typically one with extreme values for OD pairs that use the link in question and a lot of zero values for the OD pairs that do not use the link. These cases are in the set $\mathcal{D}$, but may, in reality, be very unlikely. If we are interested in a plausible maximum link load, we might like to add some more constraints to eliminate the cases which are practically impossible. This can be achieved by setting an upper bound on the value that an estimate for an OD pair may obtain.

We propose the Bounded Robust algorithm that is based on the idea that we do utilize a traffic matrix estimate, but also take into consideration the error of the estimator. We calculate the upper bounds for the traffic matrix estimate and the estimator's confidence intervals. To obtain a confidence interval we need to calculate the standard error of the estimator. When we use the maximum likelihood estimator, this can be achieved through the Cramér-Rao lower bounds (CRLB) for the variance of an estimator, derived in [10].

In the next subsection we review the use of the CRLB, and in the following subsection formulate the constraint generation algorithm for the bounded case.

A. *Confidence interval for estimator*

The variance/covariance matrix of any unbiased estimator cannot be lower than the inverse of the Fisher information matrix $\mathcal{I}$. This is the Cramér-Rao lower bound for the variance of an estimator. As the maximum likelihood method is unbiased and asymptotically efficient [11], it has the lowest variance of any estimator and thus its variance coincides with the bound. On the other hand, the gravity model based methods are typically biased, and thus the bound does not necessarily apply.

For the MLE we can calculate the variance/covariance matrix of the estimator using the CRLB. The vector of the standard errors $\boldsymbol{S}_E$ is then the square root of the diagonal elements of the CRLB matrix, and the confidence interval is given by

$$\hat{\boldsymbol{x}} - z \cdot \boldsymbol{S}_E \leq \boldsymbol{x} \leq \hat{\boldsymbol{x}} + z \cdot \boldsymbol{S}_E,$$

where z defines the width of the confidence interval. For instance, $z = 1.96$ is the 95% confidence interval.

B. *Bounded constraint generation function*

Now we can redefine the set $\mathcal{D}$ of traffic matrices to take into consideration the confidence intervals. We have no need to bound the low values, as we are looking for the maximal link loads. Thus we define this new set as

$$\mathcal{D}' = \{\boldsymbol{x} : \boldsymbol{A}_0\boldsymbol{x} = \boldsymbol{y}_0 \ , \ 0 \leq \boldsymbol{x} \leq \hat{\boldsymbol{x}} + z \cdot \boldsymbol{S}_E\}. \tag{21}$$

The corresponding constraint generation problem is then

Problem 4 (Bounded Constraint Generation):

$$\boldsymbol{x}^* = \arg\max_{x \in \mathcal{D}'} (\boldsymbol{A}^{(i)}\boldsymbol{x})_l \tag{22}$$

If

$$u^{(i)}\, \boldsymbol{C}_l < (\boldsymbol{A}^{(i)}\boldsymbol{x}^*)_l$$

then

$$D^* \leftarrow D^* \cup (l, \boldsymbol{x}^*).$$

A similar iteration is performed as described in the previous section with Problem 4 now in place of Problem 3. The result of the algorithm gives the optimal routing matrix over all reasonably conceivable traffic matrices.

VI. Simulation study

Robust methods have lower worst case link utilizations, while this provisioning for the worst case might have an adverse affect on the mean utilization obtained. In this section we evaluate this tradeoff between average link utilization and robustness by a simulation study with the two different topologies, shown in Figure 1.

A. *Compared methods*

In order to be able to use the Cramer-Rao bounds we need to use the maximum likelihood method for the traffic matrix estimation. Thus, it is natural to compare the traditional method calculated with this estimate to the robust methods using the same estimate. While specifically for small sample sizes the gravity methods could outperform the MLE in traffic matrix estimation [9], the behavior of the different metrics defined below is similar regardless of the estimator used for the traditional method. Therefore, we compare the following

1) The traditional load balancing with maximum likelihood traffic matrix estimate
2) The Robust method
3) The Bounded Robust method,

and choose sample sizes $T = 25$ for the smaller topology and $T = 100$ for the larger topology, so that the performance for the traditional method is approximately the same with MLE and gravity estimate.

B. *Metrics*

The algorithms yield a routing matrix $\widehat{\boldsymbol{A}}$. To evaluate the performance of each algorithm, we define three performance metrics.

1) The real case utilization of the most congested link.

$$u_{\mathrm{rc}} = |\widehat{\boldsymbol{A}}\boldsymbol{x}_0 / \boldsymbol{C}|, \tag{23}$$

where x_0 is the real value of the traffic matrix during the measurements.

2) The worst case maximal utilization.

$$u_{\mathrm{wc}} = \max_{\boldsymbol{x} \in \mathcal{D}} |\widehat{\boldsymbol{A}}\boldsymbol{x} / \boldsymbol{C}|. \tag{24}$$

We consider the largest value for u with the routing matrix $\widehat{\boldsymbol{A}}$, changing the traffic matrix within the polytope

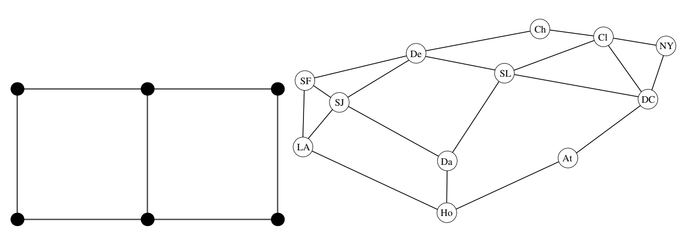

Fig. 1. Test topologies

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

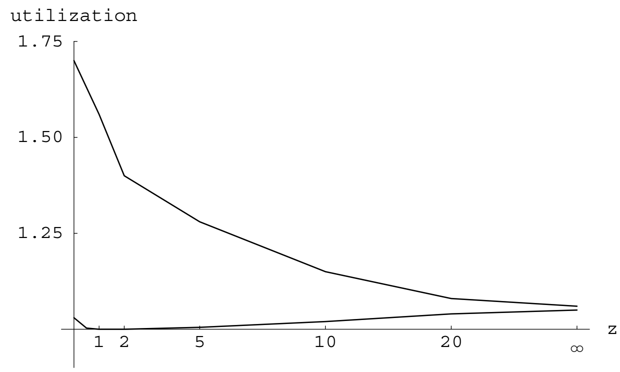

Fig. 2. Real case performance (lower curve) and worst case performance (upper curve) of the Bounded Robust method in the small topology as a function of the confidence interval coefficient

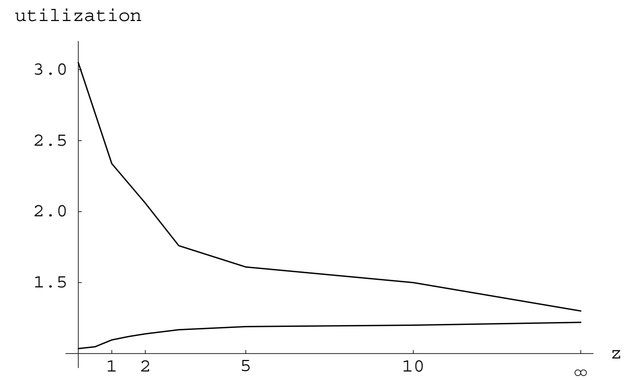

Fig. 3. Real case performance (lower curve) and worst case performance (upper curve) of the Bounded Robust method as a function of the confidence interval coefficient

specified in (2). This is the criterion the Robust methods aims to minimize.

3) The bounded worst case utilization is similar to the second one except we exclude implausible traffic matrices from consideration by finding the maximal u such that the traffic matrix is within the bounded polytope given in (21).

$$u_{\mathrm{bwc}} = \max_{\boldsymbol{x} \in \mathcal{D}'} |\widehat{\boldsymbol{A}}\boldsymbol{x}/\boldsymbol{C}|. \tag{25}$$

This is the criterion the Bounded Robust methods aims to minimize.

All the results in the sequel are given relative to the optimal load balancing that would be obtained using the exact traffic matrix for the traditional load balancing method.

C. Results for the smaller topology

The accuracy of the traffic matrix estimate is dependent on the difficulty of the problem. The more OD pairs compared to the number of links there are in the considered topology, the more underconstrained the problem becomes. In heavily underconstrained situations the estimates are naturally less accurate.

We consider first the small topology depicted on the left hand side of Figure 1. It has six nodes, and thus 30 OD pairs. There are seven two-way links, so the number of links is 14. This is a relatively easy situation. The standard error is typically about 10% of the OD pair volume.

In this case our Bounded Robust method outperforms the traditional method relying on a single traffic matrix estimate. This is explained by the fact that while the estimate is inaccurate, the polytope used in the Bounded Robust method includes the real traffic matrix, and is not excessively large due to small confidence intervals.

This is in line with the results reported in [5]. They find that small error margins do not make the result worse. In their study the margins are centered around the real values. Thus the traditional method achieves optimal utilization. We use a traffic matrix estimate instead of the actual traffic matrix so the traditional method does not yield optimal utilization.

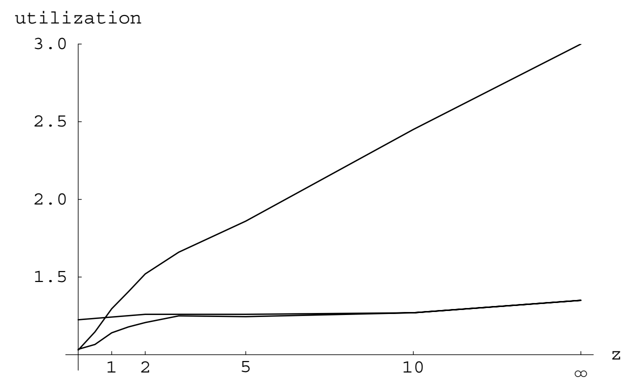

Fig. 4. Bounded worst case performance of Traditional (the fast rising curve), Bounded Robust (lower curve) and Robust methods (midlle curve) as a function of the confidence interval coefficient.

Thus, as can be seen in Figure 2, not only does using the Bounded Robust method make the utilization no worse, it actually makes it better, while at the same time making the routing more robust.

In the figure, the horizontal axis is the value of z defining the width of the confidence interval in multiples of the standard error, as given in equation (21). The vertical axis gives the maximal link utilization u obtained in the network by routing the corresponding polytope. The lower curve gives the real case utilization and the upper curve gives the worst case utilization. We can see that the real case utilization is near optimal when using the Bounded Robust method with moderate size confidence intervals. The case $z = 0$ corresponds to the traditional method and the case where z is so large that it does not bound the polytope and $\mathcal{D}' = \mathcal{D}$ corresponds to the Robust method. For traditional method $u_{rc} = 1.03$ and for the Robust method $u_{rc} = 1.06$. The worst case utilization is naturally highest with traditional method, and becomes smaller as a function of the confidence interval used.

D. Results for the larger topology

We now consider the larger topology shown on the right hand side of Figure 1. This is a more realistic size network with 12 nodes, 132 OD pairs and 38 links. Clearly, the situation is more underconstrained than the previous one, and thus more difficult. The standard error is more than twice the the standard error of the previous case.

Figure 3 depicts the real case utilization and the worst case utilization obtained by the different methods. The utilization is lowest with the traditional method with $u_{rc} = 1.04$. For $z = 0.5$ it is still quite low, but then grows quickly when z approaches 3. After that the utilization grows only moderately from 1.17 to 1.22 which is the real case utilization for the Robust method.

Figure 4 shows the bounded worst case utilizations for the different methods. On the horizontal axis is the width of the confidence interval, over which the bounded worst case utilization u_{bwc} is calculated. So, $z = 0$ corresponds to using no confidence interval at all. In this case u_{bwc} is exactly the same as the real case utilizations, as we calculate the worst case over a zero size polytope, which is the single point of the estimate. Then, for instance at the point $z = 1.96$ the curves give the bounded worst case utilizations for the polytope with 95% confidence intervals. Finally at $z = \infty$ u_{bwc} coincides with u_{wc} as the confidence interval is so large that we take the worst case over all possible traffic matrices. The highest curve is the traditional approach. The curve for Bounded Robust starts lower than the Robust, but they approximately coincide after $z = 3$.

For example, let us consider the 95% confidence interval for each element of the traffic matrix. We have three methods: Traditional method, Bounded Robust method with $z = 1.96$ and Robust method. We can read from Figure 4 the bounded worst case utilization for each. From Figure 3 we can read the real case and worst case utilizations using $z = 0$ for Traditional method, $z = 1.96$ for Bounded Robust and $z = \infty$ for Robust. These results are listed in Table I.

TABLE I
VALUES OF UTILIZATION FOR DIFFERENT METHODS WHEN $z = 1.96$.

	u_{rc}	u_{bwc}	u_{wc}
Traditional	1.04	1.52	3.05
Bounded Robust	1.13	1.20	2.06
Robust	1.22	1.26	1.31

VII. Conclusion

In this paper we addressed new approaches for the network load balancing task. Instead of relying on a estimated and inaccurate traffic matrix we study methods that take a different approach. The Robust method, which does not need any traffic matrix at all, was found to perform sufficiently well. The link utilization was in a less underconstrained situation only marginally higher than that of the traditional approach. The worst case utilization for the Robust method was only one percent higher than its real case utilization. The Robust method guarantees a maximal link utilization only 6% worse than optimal in that scenario. For the more difficult scenario, the utilizations were higher for the Robust method, but the worst case utilization is still only 31% higher than optimal, while worst case for the Traditional method is over three times the optimal utilization.

We proposed a novel extension for the robust approach, the Bounded Robust method, which requires a traffic matrix estimate, but takes the uncertainty in the estimation into account. The approach decreases the size of the polytope of considered traffic matrices by introducing confidence intervals for the estimator, thus eliminating the need to include the next to impossible extreme cases in the provisioning. The method uses maximum likelihood estimates for the traffic matrix and makes use of the Cramer-Rao lower bounds to obtain the confidence intervals. In some simulation scenarios the Bounded Robust method was shown not only to add the robustness that the traditional approaches lack, but actually outperform them with regard of the real case maximal link utilization. In any situation the method is close to the traditional method and outperforms the Robust method in real case link utilization, while still maintaining the robust performance over all plausible traffic matrices.

Acknowledgments

The work for this paper was carried out as a part of project Fancy supported by the Academy of Finland grant 210275.

The author wishes to thank Prof. Jorma Virtamo, Ph.D. Samuli Aalto and Lic.Sc.(Tech) Riikka Susitaival from TKK for their help.

References

[1] N. Duffield, P. Goyal, and A. Greenberg, "A flexible model for resource management in virtual private networks," in *in ACM SIGCOMM*, 1999.
[2] T. Erlebach and M. Ruegg, "Optimal bandwidth reservation in hose-model vpns with multi-path routing," in *in proceedings on Infocom 2004*, 2004.
[3] W. Ben-Ameur and H. Kerivin, "Routing of uncertain traffic demands."
[4] M. Johansson, "Data driven traffic engineering," Presentation in LSNI Workshop, Banff, Canada, 2005.
[5] D. Applegate and E. Cohen, "Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs," *IEEE/ACM Transactions on Networking, Vol.* 14*, No.* 6, 2006.
[6] J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography," *Journal of the American Statistical Association*, vol. 95, pp. 1063–1075, 2000.
[7] J. Kowalski and B. Warfield, "Modeling traffic demand between nodes in a telecommunications network," in *ATNAC 95*, 1995.
[8] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *ACM Sigmetrics*, 2003.
[9] I. Juva, "Sensitivity of traffic matrix estimation techniques to their underlying assumptions," in *ICC 2007*, Glascow, UK, 2007.
[10] P. Bermolen, S. Vaton, and I. Juva, "Search for optimality in traffic matrix estimation: A rational approach by cramér-rao lower bounds," in *NGI 2006*, Valencia, Spain, 2006.
[11] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley and Sons, Inc, 1997.