Topology Information Condensation in Hierarchical Networks.

Piet Van Mieghem

Delft University of Technology[a]

ABSTRACT

Inspired by the PNNI protocol of the ATM Forum, this work focuses on the problem of node aggregation within peer groups and link aggregation between peer groups. It is assumed that the (large) network is already divided into peer groups. The objective is to maximally condense topology information subject to a given accuracy constraint.

The QoS measures can be reduced into three distinct classes: an additive QoS class, a min-max one and the last, a combination of additive and min-max QoS measures. A new method for node aggregation of the additive QoS class (even with multiple QoS measures) is presented. The min-max class is discussed and Lee's optimal solution (Lee, 1995) for a single min(max) QoS measure is reviewed. Finally, we discuss the extension of our new method to the combination of additive and min-max QoS measures.

A detailed example illustrates the presented algorithm for a single additive QoS measure. Subject to a given relative accuracy ε, it shows how to perform node and link aggregation on different hierarchical levels and how to establish the whole hierarchical structure of the original network.

KEYWORDS: hierarchical networks, information condensation, node and link aggregation

SUBMITTED TO : Computer Networks and ISDN Systems

Corresponding author: Piet Van Mieghem
Telecommunication and Traffic Control Systems Group
Faculty of Information Technology and Systems
Delft University of Technology
Postbus 5031
2600 GA Delft
The Netherlands
Tel.: +31-15-278 2397
Fax: +31-15-278 1774

email: P.VanMieghem@its.tudelft.nl

[a] Part of this work was done while at Alcatel Corporate Research Center in Antwerp.

1. HIERARCHICAL STRUCTURING

In today's networks, the number of nodes (switches, routers, terminals, ...) is growing fast. Just as in dictionaries, telephone books and large data systems, also in large networks *hierarchical structuring* proves efficient and highly desirable. The ATM-forum has incorporated this principle of hierarchical structuring in its PNNI (private network-network interface) specification.

In graph theory, there exists several 'standard' representations of a network topology (Cormen et al., 1990), such as a topology matrix, an adjacency list, a link state table, etc... Roughly, we may consider a hierarchical structure of a network as a pre-processing of a 'standard' topology representation to enhance routing. The hierarchical structuring of very large networks consists of (a) a partitioning of the network in smaller subsets of nodes (Van Mieghem, 1998a) and (b) a representation of this partitioning in an efficient, layered form, called a hierarchy. The partitioning of the network is, in fact, a recursive process because subsets of nodes may in turn again be grouped into an even smaller number of subsets. The recursive process ends after N recursions when one set contains all underlying subsets. This is the highest level N subnet (or peer group). Hence, the recursion creates a hierarchical tree in which child nodes represent the network in more detail than their parent nodes. Each recursion reflects a different hierarchical level. An interesting property of this recursive partitioning process is, that the union of all subsets of hierarchical level k again represents the original network, however, with different granularity. The lowest level - also called the physical level and further referred to as the $k = 0$ hierarchical level -possesses the finest granularity since that level consists of subsets with the original nodes as elements. Nodes are assumed to be indivisible into smaller parts. The subsequent levels have subsets as elements. The whole tree comprising all hierarchies is called the hierarchical structure of the network as shown in Figure 1.

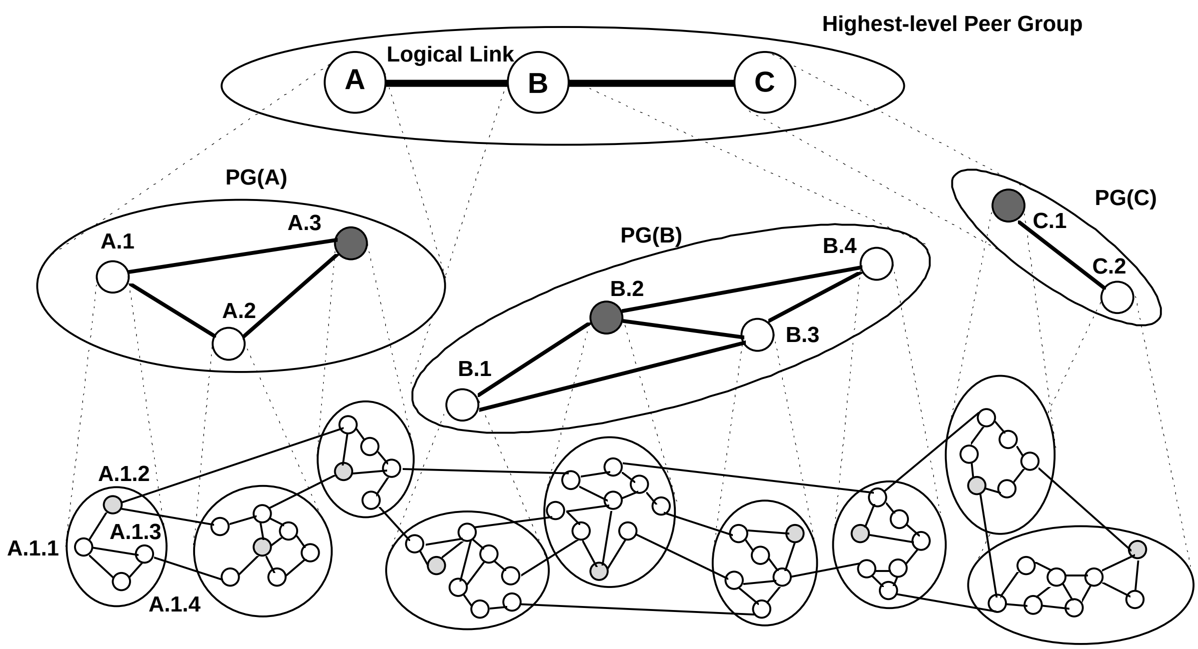


Figure 1 A hierarchical structure based on the specifications of PNNI where a subnet is coined a peer group, denoted as PG(.).

An attractive feature of hierarchical structures is the efficient representation. Since each hierarchical level actually represents the network, a full detailed description of each level leads to a N-fold redundancy. Thus, the hierarchical structure naturally calls for (topology) information condensation. It makes sense to represent a subset on the next higher level in some condensed form, which is called a complex node or an aggregated node. Actually, hierarchical structuring is based on a general principle stating that the immediate neighborhood is desired to be known in greater detail than the farther surroundings. For instance, it seems reasonable that a citizen of Antwerp roughly knows the street map of Antwerp and even that of Brussels, but it sounds odd to assume that he also knows the street map of Tokyo equally detailed.

Perhaps the most important benefit of a hierarchical structuring of a network is scalability leading to a decreased routing complexity (Van Mieghem, 1997), smaller routing tables (Kleinrock and Kamoun, 1977; Peleg and Upfal, 1988) and an enhanced human insight in the network. However, the information condensation can also result in routing solutions that are less optimal.

In the literature, we found some articles on hierarchical aggregation and decomposition (see e.g. references in Huang and Zhu, 1996), but these were limited to a routing algorithm that optimizes one QoS measure. The majority of articles dealing with hierarchical structures confine themselves to the proposal and evaluation of hierarchical routing algorithms dedicated to special network requirements. Montgomery and de Veciana (1998) present a hierarchical source routing algorithm based on the implied cost of a connection which measures the expected increase in future blocking that would occur from accepting this connection. Van Mieghem (1998) shows that a top-down strategy for routing in a hierarchy as shown in Figure 1 offers clear advantages over bottom-up strategies. Directly related to Internet scalability problems but ignoring QoS issues, Behrens and Garcia-Luna-Aceves (1998) explain a hierarchical routing algorithm, called area-based link-vector algorithm (ALVA), which uses link-state information to compute optimal paths but without replicating the complete topology information at every node.

Here, we concentrate on *topology information condensation*, in particular, on the question how we can represent a subset (determined above) as aggregated node on a next higher level (as shown in Figure 2). For an alternative point of view, we refer to W. Lee (1995a). We consider the information condensation process as consisting of two steps: first node aggregation, followed by link aggregation. The final result after link aggregation is advertised. The discussion, although inspired by PNNI, is very general including the full impact of quality of service (QoS), and hence it is valid for general hierarchical networks (e.g. a future Internet Hierarchical protocol, say H-OSPF).

2. NODE AGGREGATION.

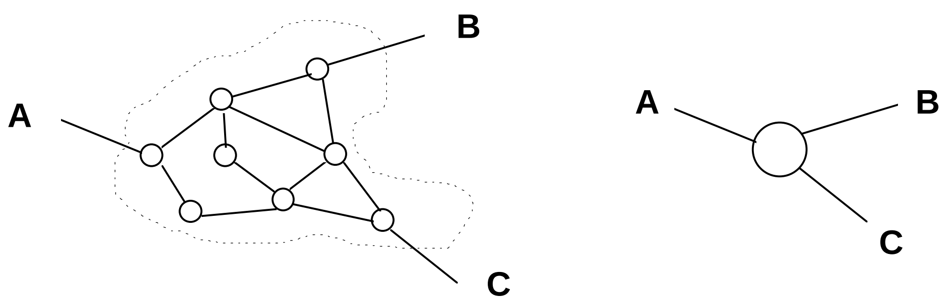


Figure 2 The original network (left) and the reduction of the network to a complex node (right). The nodes with connectivity outside the network (e.g. those with links leading to A, B, C) are called ingress or egress nodes.

Node aggregation, as illustrated in Figure 2, is concerned with the complex node representation of the original network. Ideally, when making abstraction of the details of the original network (enclosed by the dotted line in Figure 2), its replacement by a complex node should be transparent for the nearest neighbor nodes (A, B, C in Figure 2). Just as in systems theory, the network to be condensed is regarded as a black box and only via input/output relations (transfer functions) characteristics of this network are known. In information networks, the input/output relations are specified via QoS measures (such as delay, available bandwidth, link usage cost, packet loss,…) between ingress and egress. We can categorize the QoS measures into three important QoS classes: a class consisting of only additive QoS measures, of only min(max) QoS constraints or of a combination of both additive and min(max) QoS measures. We assume that the all ingress/egress QoS measure(s) can be computed via a routing algorithm. This settles the given data.

The second stage handles the representation or ‘nodal structure’ of the complex node. There exists two extreme situations: the full mesh representation containing all pairwise QoS measures between ingress/egresses and the symmetric-point case where the entire network is collapsed into one point and where merely an all-in-one characterizing parameter (known as the diameter) is advertised (Lee, 1995). The latter case also reflects the ultimate information condensation possible. In between these extremes lies the nucleus-spoke representation as adopted in PNNI and illustrated in Figure 3. Other representations (such as e.g. more than one nucleus[b]) seem to complicate more than to enhance the efficiency of node aggregation.

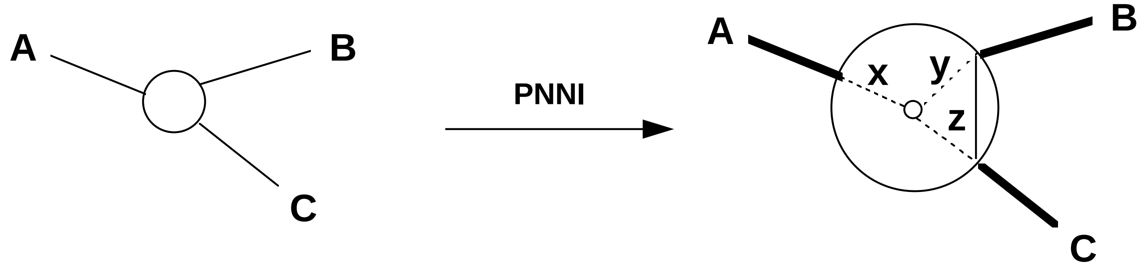


Figure 3. The complex node structure in PNNI. The vertices x, y, z connecting a fictitious nucleus and the ingress nodes are called spokes (dotted lines) while an exceptional link (full line) is a direct link between an ingress/egress pair.

PNNI assumes that the peer groups can be represented by a wheel with a nucleus and several spokes. The concept of a nucleus is attractive, because it represents in some way the point of gravity of the underlying peer group. Hence, the concept assumes a reasonable amount of symmetry in the sense that each node of the underlying peer group lies at approximately the same 'QoS-measure-distance' from that nucleus. In case the path ends in a complex node, the nucleus is the best representative for each node in the underlying peer group. This property makes the concept attractive. Network asymmetry is dealt with via exceptional links (see Figure 3 and the discussion below).

In spite of the attraction of the nucleus-spoke concept, there are also limitations. Consider as an example the routing from a physical node B.4.x to another one B.1.y in Figure 1. Using the top-down level routing principle (Van Mieghem, 1998), a path is computed in peer group PG(B) from the nucleus of B.4 to the nucleus of B.1. Let us assume that the path runs over the complex node B.2. After routing at this level, the egress port at B.4 and the ingress port at B.1 are known. From these ports, a path is computed on the next lower level (in this case the physical level) to B.4.x and B.1.y respectively. In this way, the complete, hierarchical path from B.4.x to B.1.y is known. Since the nucleus is the best representative for the *entire* peer group, it possible that the nucleus of B.4, for instance, is not the best representative for the *particular* node B.4.x such that a better path to B.1.y may exist (which traverses e.g. B.3 instead of B.2). This phenomenon is, in general, characteristic for any information condensation process. If small losses or inaccuracies are not tolerable, condensation may not be possible.

The overall objective is to achieve maximal information condensation, defined as minimizing the number of exceptional links in the nucleus-spoke representation while simultaneously approximating each ingress/egress QoS measure to within a given accuracy. The optimum is reached when only spokes are needed in the complex node Figure 3 and the ultimate limit, the 'symmetric-point' condensation, can be achieved for the particular network whose ingress/egress equations are the same for each pair.

2.1 ADDITIVE QOS MEASURES.

An additive QoS measure has the property that the value along a path is the sum of the values of the edges along that path. Typical examples of additive QoS measures are the delay, the hop count, cost,...

2.1.1 Single QoS measure.

We assume that we dispose of the best QoS value for paths AB, AC, BC. These ingress/egress values QoS_{AB}, QoS_{AC}, QoS_{BC} are computed from the original network (Figure 2) via some shortest path algorithm and are all non-negative. According to the nucleus-spoke structure (Figure 3), these ingress/egress values QoS_{AB}, QoS_{AC}, QoS_{BC} must be mapped into the spoke values QoS_x, QoS_y, QoS_z. We have 3 unknown and precisely 3 equations,

$$QoS_x + QoS_y = QoS_{AB}$$

[b] Placing additional nuclei is analogous to using Steiner points in a multicast tree to obtain a better overall tree

$$QoS_x + QoS_z = QoS_{AC}$$
$$QoS_y + QoS_z = QoS_{BC}$$

The solution of this linear set is straightforward and the exact solution is found in the appendix.

However, in general we have *m* vertices connecting the neighbors (via symmetrical links). These *m* vertices correspond to *m(m-1)/2* ingress-egress pairs (all combinations of 2 out of *m*). Only in the case $m = 3$, it happens that $m(m-1)/2 = 3$. Thus, in general, we have an over-determined set of *m(m-1)/2* equations in only *m* unknowns. Although an exact solution clearly does not exist, we can always find the best[c] possible set of *m* unknowns subject to the *m(m-1)/2* linear equations. The linear set has an interesting structure,

$$\begin{matrix} 1\ 1\ 0\ 0\ 0 \ldots 0 \\ 1\ 0\ 1\ 0\ 0 \ldots 0 \\ 1\ 0\ 0\ 1\ 0 \ldots 0 \\ \ldots \\ 1\ 0\ 0\ 0\ 0 \ldots 1 \\ 0\ 1\ 1\ 0\ 0 \ldots 0 \\ 0\ 1\ 0\ 1\ 0 \ldots 0 \\ \ldots \\ 0 \ldots \quad 0\ 1\ 1 \end{matrix} \quad x \quad \begin{matrix} QoS_1 \\ QoS_2 \\ QoS_3 \\ \ldots \\ QoS_m \\ \\ \\ \\ \\ \end{matrix} \quad = \quad \begin{matrix} QoS_{12} \\ QoS_{13} \\ QoS_{14} \\ \ldots \\ QoS_{1m} \\ QoS_{23} \\ QoS_{24} \\ \ldots \\ QoS_{m-1,m} \end{matrix} \qquad (1)$$

We further denote this set by $M_{m(m-1)/2\ x\ m}\ Q_{m\ x\ 1} = F_{m(m-1)/2\ x\ 1}$ where the component F_{ij} refers to the best ingress/egress value corresponding to the best QoS path from ingress i to egress j. On each row there are precisely two non-zero elements and these non-zero elements are equal to one. Geometrically, each equation represents a hyperplane in the *m*-dimensional space orthogonal to the bisectrice-vector[d] of two axes *i* and *j* and intersecting those axes at a distance QoS_{ij} from the origin. The well specified form of the matrix M encourages a further analysis. The least-square solution of $M_{m(m-1)/2\ x\ m}\ Q_{m\ x\ 1} = F_{m(m-1)/2\ x\ 1}$ follows (Lanczos, 1988) by multiplying both sides with the transpose of M, resulting in $M^TM\ Q = M^T F$, such that

$$Q = (M^TM)^{-1}\ M^T\ F \qquad (2)$$

where we compute the inverse of a symmetrical matrix. Since the matrix M is known, the right hand side can be pre-calculated as demonstrated in the appendix.

So far, we are in the position to determine a best possible solution (QoS_1, QoS_2, ..., QoS_m) that characterizes the spoke values for a certain QoS. However, a valid solution QoS_i must be non-negative to make any sense[e]. Geometrically, this means that the best solution must have all its coordinates non-negative. When reasoning in three dimensions (see Figure 4), we observe that such a point can be found in case the distances of the planes from the origin are not too far apart (i.e. when the QoS_{ij} are comparable in magnitude). If the distances broadly vary, the point of intersection is likely not to lie in the positive octant. In higher dimensions $m > 3$, the probability that the point of intersection lies in the only 'positive subspace' decreases sharply, because there is only 1 such a positive subspace out of 2^m similar subspaces! Hence, we see that the non-negative condition of QoS_i puts additional constraints to the problem.

Some paths may be significantly better in QoS performance than others. The *m* unknowns are seriously affected by the exceptional QoS values of a particular path, even to the extend that a valid best fitting solution does not exist because the non-negativity condition is violated. In order to circumvent this problem, an exceptional link between a particular ingress-egress is introduced. This link value is immediately advertised without affecting the determination of the other spoke values ((QoS_x, QoS_y, QoS_z) in Figure 3 and (QoS_1, QoS_2, ..., QoS_m) in general). Thus, we have means to always find a good representation by adding 'enough' exceptional links. Clearly, the maximum number of exceptional links for one QoS measure equals *m(m-1)/2*, a situation often

---

[c] Best possible is to be understood in the least squares sense. This stems from the minimization of the square of the error r. Specifically, $r^2 = \| M.Q - F \|$ where $\|.\|$ denotes the Frobenius norm, which can also be written as $r^2 = (M.Q - F)^T (M.Q - F)$ and further as $r^2 = Q^T M^T MQ - 2 F^T M.Q + F^T F$. Hence, minimizing this quadratic form via formal derivation $dr^2/dQ = 0$ yields $M^T MQ = M^T F$ that is equivalent to (2). The general theory relies on singular value decomposition (SVD) for which we refer to the book of Golub and Van Loan (1989).

[d] This is the vector at an angle of 45^o lying in the plane formed by two axes.

[e] Mathematically, of course, we can deal with negative values. However, if we represent the underlying network by a particular hierarchical level, the complex nodes on this level should reflect physical QoS measures.

coined as a full mesh solution. In this extreme situation, there is no fictitious nucleus and information condensation has failed because the complete set of QoS values between ingress/egress pairs is unaltered advertised.

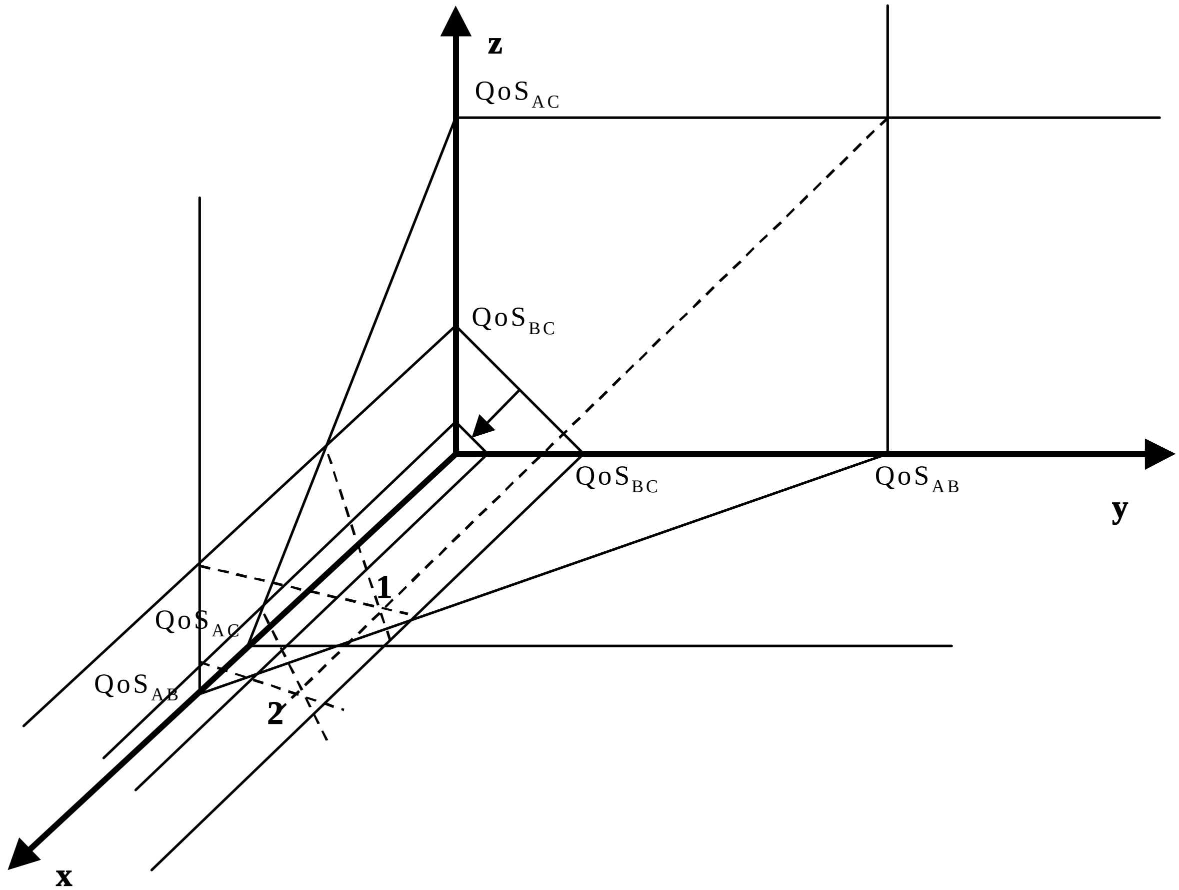


Figure 4 Geometry in three dimensions of intersecting bisectrice-planes. In three dimensions, we have a unique solution which is seen as the intersection point of the three dotted straight lines. Two situations are drawn for two values of QoS_{BC} leading to the solutions 1 and 2. We observe that solution 1 has all three components positive while solution 2 has a negative z-component which is attributed to the fact that, in case 2, QoS_{BC} is much smaller than the QoS_{AB} and QoS_{AC} .

A last point concerns the accuracy of the result. The non-negativity condition is necessary for the physical interpretation, but not sufficient to guarantee a certain level of accuracy. The accuracy is taken into account by the maximum relative error, denoted by ε.

2.1.2 Strategy : Optimizing condensation and accuracy simultaneously.

The analysis above naturally leads to a strategy that determines when to advertise exceptional links:

1. Determine $Q_{m \times 1}$ from equation (2).

2. if $(|\Sigma_{1\leq j\leq m} M_{i\,j}.QoS_j - F_i| \leq \varepsilon F_i)$ for each i, go to 3
   else go to 4

3. if $Q_{m \times 1} = (QoS_1, QoS_2, ..., QoS_m)$ obeys $QoS_i \geq 0$ for each i, we are done
   else go to 4

4. Search for the minimum component in $F_{m(m-1)/2 \times 1}$. Advertise the path and QoS value that minimizes this component in F as an exceptional link. Omit the corresponding linear equation and go to 1.

In each condensation cycle (step 2 to step 4), the accuracy of the solution $Q_{m \times 1}$ is checked (in step 2): the square error per component (i.e. ingress/egress pair) must lie within a given relative error ε . Step 3 verifies the non-negativity of the result and step 4 selects an exceptional link based on geometrical considerations as outlined above. If $\varepsilon = 0$, there will be no information condensation while for $\varepsilon \to\infty$, there is maximal condensation, but no accuracy control. Obviously, the value of ε is critical.

The worst case complexity of the proposed strategy occurs when $\varepsilon = 0$, and it is determined according to the notion explained in the appendix,

$$C_{worst} = \Sigma_{0\leq j < m(m-1)/2} [C_1(j) + C_2(j) + C_3 + C_4(j)]$$

where the worst case complexities of each step in the strategy are $C_1(j) = O((m(m-1)/2-j)^3)$ which ignores the special structure of M , $C_2(j) = O(m^2(m-1)/2-mj))$, $C_3 = O(m)$ and $C_4(j) = O(m(m-1)/2 - j)$. Hence,

$$C_{worst} = O(\Sigma_{0\leq j < m(m-1)/2} ((m(m-1)/2-j)^3] = O(m^8/64)$$

This worst case complexity should be confronted with the best case (only step 1, 2 and 3 once) complexity $C_{best} = O(m^3/2)$ as derived in the appendix.

It is instructive to relate our strategy to the concept of a *t*-spanner of a graph *G* which has been broadly studied in graph theory (Peleg and Shäffer, 1989; Althöfer *et al*., 1993). A *t*-spanner of a graph *G(V,E)* is a subgraph *G'(V,E')* of *G* where for all nodes *u,v* *Î* *{V}* holds that the distance from *u* to *v* in *G'* is at most *t* times longer than the distance in *G*. Althöfer *et al*. (1993) have proposed a simple algorithm to compute a *t*-spanner, strongly based on Kruskal's minimum spanning tree algorithm. In addition, they demonstrate that for an undirected graph, there exists a polynomially constructable *(2t+1)*-spanner such that the number of links $E' < V.[V^{1/t}]$ (where *[x]* denotes the integer just exceeding or equal to *x*) and the sum of all the edge weights of *G'* is smaller than the *(1+V/(2t))* times the weight of the minimum spanning tree of *G*. Roughly, the maximum relative error ε plays a role analogous to *t*. The difference clearly is that, in our proposal, the resulting complex node (nucleus-spoke structure with exceptional links) is not a subgraph of the original graph so that there is no confinement to minimum spanning tree techniques. Therefore, it is expected that, for a same accuracy, a closer agreement to the original graph can be obtained with a smaller number of links.

Finally, from experiences via simulation, we found that this strategy constitutes a very robust technique that perfectly deals with possibly redundant information (e.g. triangle (in)equalities such as $QoS_{AB} + QoS_{Bc} = (<) QoS_{Ac}$).

2.1.3 Multiple QoS measures.

Suppose now each link is characterized by *n* additive measures and assume that for all ingress *i* -egress *j* pairs, a best representative QoS vector, $\mathbf{QoS_{ij}} = [QoS_{ij1}, QoS_{ij2}, ..., QoS_{ijn}]$ is available. How this best QoS vector is computed is here regarded as beyond the scope, but we refer to our QoS routing algorithm, TAMCRA, a Tunable Accuracy Multiple Constraints Routing Algorithm (De Neve and Van Mieghem, 1998, 1998a). We further consider every component as equally important. Then, all previous unknowns become unknown *n x 1* vectors. Reasoning on the example drawn in Figure 3, this means that QoS_x transforms to $\mathbf{QoS_x} = [QoS_{x1}, QoS_{x2}, ..., QoS_{xn}]$ and likewise all other parameters. Thus, we have, instead of *3* unknown now *3 n* unknowns. The linear set (1) then generalizes to

$$\begin{matrix} I_{nxn} & I_{nxn} & O_{nxn} & O_{nxn} & O_{nxn} & \ldots & O_{nxn} \\ I_{nxn} & O_{nxn} & I_{nxn} & O_{nxn} & O_{nxn} & \ldots & O_{nxn} \\ I_{nxn} & O_{nxn} & O_{nxn} & I_{nxn} & O_{nxn} & \ldots & O_{nxn} \\ \ldots & & & & & \ldots & \\ I_{nxn} & O_{nxn} & O_{nxn} & O_{nxn} & O_{nxn} & \ldots & I_{nxn} \\ O_{nxn} & I_{nxn} & I_{nxn} & O_{nxn} & O_{nxn} & \ldots & O_{nxn} \\ O_{nxn} & I_{nxn} & O_{nxn} & I_{nxn} & O_{nxn} & \ldots & O_{nxn} \\ \ldots & & & & & & \\ O_{nxn} & .. & & & O_{nxn} & I_{nxn} & I_{nxn} \end{matrix} \; x \; \begin{matrix} \mathbf{QoS_1} \\ \mathbf{QoS_2} \\ \mathbf{QoS_3} \\ \ldots \\ \mathbf{QoS_m} \end{matrix} = \begin{matrix} \mathbf{QoS_{12}} \\ \mathbf{QoS_{13}} \\ \mathbf{QoS_{14}} \\ \\ \mathbf{QoS_{1m}} \\ \mathbf{QoS_{23}} \\ \mathbf{QoS_{24}} \\ \\ \mathbf{QoS_{m-1,m}} \end{matrix} \qquad (3)$$

where I_{nxn} and O_{nxn} are the *n x n* identity and zero matrix, respectively. Formally, the structure of (1) is maintained if the elements are substituted by the corresponding block matrices. In fact, using the Kronecker product $\otimes$, the block matrix in (3) can be written as $M \otimes I_{nxn}$. This formal resemblance is believed to be advantageous because efficient matrix manipulations analogous to those presented in the appendix seem possible.

The choice of exceptional links is inspired by the geometrical considerations of the 'one-dimensional' case above. Specifically, when the $\mathbf{QoS_i}$ vector has negative components, it triggers the advertisement of an exceptional link. Now, we select that exceptional link that geometrically lies farest from (or nearest to) the origin because, by removing the corresponding block row, the remaining set of block rows will only be better approximated (in least square sense) if the most extreme-valued block row disappears. To compute the distance, various distance metrics are possible, for instance, the Holder norm $|\mathbf{QoS_i}|^p = \Sigma_{1\leq j \leq n} QoS_{ij}^p$ (where *p* is real and *p* ³ *1*) which reduces in the case *p = 2* to the simple, well-known Euclidian distance measure, while for *p*®¥, the maximum component equals the norm. The chosen norm should agree with the one used in the QoS routing algorithm, e.g. TAMCRA is based on the *p*®¥ norm.

In the sequel, we confine ourselves to the simpler 'one-dimensional' case, because we have shown here how to extend to multiple, independent, additive QoS measures.

2.2 MIN (MAX) FUNCTION OF QOS MEASURES ALONG A PATH.

A min(max) QoS measure has the property that the value along a certain path in a topology consists of the minimum (maximum) of the values of the edges that constitute that path. Typical examples of min (max) QoS measures are the bandwidth, (policy related) transit flags,... The boolean ANDing of QoS measures that are boolean numbers (either 0 or 1) is a subclass of min(max) QoS measures.

Again starting from the simple example in Figure 3, we now have the following non-linear equations in the 3 unknown,

$$\min(QoS_x, QoS_y) = QoS_{AB}$$
$$\min(QoS_x, QoS_z) = QoS_{AC}$$
$$\min(QoS_y, QoS_z) = QoS_{BC}$$

or, equivalently,

$$QoS_x \geq QoS_{AB} \text{ and } QoS_y = QoS_{AB} \quad \text{or} \quad QoS_y \geq QoS_{AB} \text{ and } QoS_x = QoS_{AB}$$
$$QoS_x \geq QoS_{AC} \text{ and } QoS_z = QoS_{AC} \quad \text{or} \quad QoS_z \geq QoS_{AC} \text{ and } QoS_x = QoS_{AC}$$
$$QoS_y \geq QoS_{BC} \text{ and } QoS_z = QoS_{BC} \quad \text{or} \quad QoS_z \geq QoS_{BC} \text{ and } QoS_y = QoS_{BC}$$

Without loss of generality, we assume $QoS_{AB} \leq QoS_{AC} \leq QoS_{BC}$ for otherwise, we re-label *x, y, z* in Figure 3. In case $QoS_{AB} < QoS_{AC} < QoS_{BC}$, we end up with conflicting requirements. To see this, consider first the inequalities. The result obeying $QoS_x \geq QoS_{AB}$ and $QoS_x \geq QoS_{AC}$ is $QoS_x \geq QoS_{AC}$ and similarly, $QoS_y \geq QoS_{BC}$ and $QoS_z \geq QoS_{BC}$. But, the equations on the first line of the equivalent set require that at least one of the values (QoS_x, QoS_y) achieves the minimum value QoS_{AB}. There are solutions in degenerate cases where two of the right hand side values are identical. Let $QoS_{AB} = QoS_{AC} < QoS_{BC}$. Then, the solutions are $QoS_x = QoS_{AB}$ and either $QoS_y = QoS_{BC}$ and $QoS_z \geq QoS_{BC}$ or $QoS_z = QoS_{BC}$ and $QoS_y \geq QoS_{BC}$. In case $QoS_{AB} = QoS_{AC} = QoS_{BC}$, we readily verify that $QoS_x = QoS_y = QoS_z$. This situation corresponds with a perfect symmetrical case. However, in case $QoS_{AB} < QoS_{AC} = QoS_{BC}$, there is again no solution.

Curiously, all anomalies will not occur in our problem, because we have more information in that the set QoS_{AB}, QoS_{AC} and QoS_{BC} are not arbitrarily non-negative numbers, but, in fact, correlated. Indeed, e.g. when the QoS measure is maximum bandwidth, we have in addition that

$$\min(QoS_{Ac}, QoS_{BC}) \leq QoS_{AB}$$
$$\min(QoS_{AB}, QoS_{BC}) \leq QoS_{AC}$$
$$\min(QoS_{AB}, QoS_{AC}) \leq QoS_{BC}$$

which implies that at least two ingress-egress Qos values must be equal. Just these cases are shown to have a solution without exceptional links.

Before proceeding with the general case where m > 3, we will invoke results obtained from Lee's (1995) spanning tree method. If we relax the complex node representation and only ask for a reduction in the *m(m-1)/2* ingress-egress relations without demanding a nucleus-spoke structure, an optimal condensation is possible using properties of a minimum spanning tree (Cormen *et al*., 1990). Lee (1995) has demonstrated that min(max) QoS measures - he concentrates on maximum bandwidth - can be exactly condensed into *m-1* relation, just the number of links in a spanning tree. The complexity to compute an optimal spanning tree is *O(m log m).*

A maximum weight spanning tree[f] has the valuable property that the weight of the link connecting a pair of nodes must be bounded from above by the minimum weight among the links along the unique path connecting the nodes on the spanning tree. Indeed, otherwise, one can increase the total weight of the spanning tree by substituting this link for the minimum weight along the unique path. Further, Lee (1995) observes that the bandwidth of the unique path between any pair of nodes on a spanning tree cannot be larger than the bandwidth of the original ingress-egress value of that same pair of nodes. Combining both bounds, it follows that the full-mesh based on bandwidth between ingress-egress pairs can be exactly encoded by a maximum weight spanning tree. In other words, the 'original ingress-egress bandwidth' equals the minimum of the bandwidths of the unique path along the spanning tree connecting these particular ingress and egress.

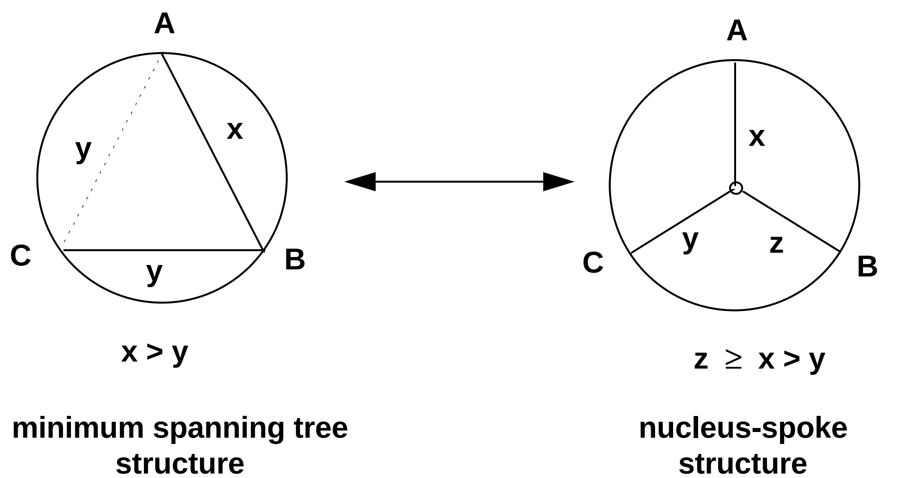


Figure 5. Transformation between minimum spanning tree structure and a nucleus-spoke structure.

Concentrating again on the general case analogous to (1) where m > 3, Lee's results implies that at most *m-1* of the *m(m-1)/2* right hand sides min(max) QoS values are different. It is doubtful whether there actually exists a pure nucleus-spoke structure without exceptional links. For, the minimum operator requires that, for all $1 \leq j \leq m$

$$QoS_j = \max(\{QoS_{kl}\}) \quad \text{where either } k = j \text{ or } l = j$$

Hence, all equations with a right hand side value smaller than $\min_j(QoS_j)$ (with $1 \leq j \leq m$) cannot be obeyed. We may wish then to advertise all these equations as exceptional links. The difficulty lies in the determination of that number of exceptional links. Recall that the overall strategy is to condense information as much as possible. Apart from this, the remaining equations are not guaranteed to be consistent.

In conclusion, the attempt to force an information condensation be conform to a complex node representation with de facto spokes, does not seem to be a satisfying strategy. Hence, the nucleus-spoke structure is the preferable candidate for additive QoS measures, while the spanning tree structure is more suited for a single min(max) QoS measure. However, the extension to multiple min(max)-QoS measures requires a spanning tree algorithm for multiple min(max) QoS measures which is very likely to be NP-complete. Moreover, the basic property of a maximum weight spanning tree, derived in the single min(max)-QoS case, does not extend to multiple min(max)-QoS because it is unclear whether there always exist a spanning tree that is a good reflection of the full mesh between ingress-egress pairs.

2.3 COMBINATION OF ADDITIVE AND MIN (MAX) QOS MEASURES ALONG A PATH.

The minimum spanning tree method of Lee (1995) only yields approximate results for the additive QoS measure, though an upper bound on the error can be derived. However, Lee did not suggest ways to adjust or control this upper bound on the error. Moreover, in the computation, the additive QoS measure (delay) is dominated by the min(max) QoS measure (bandwidth). In case bandwidth and delay are treated equally fair, the attractive, exact property of a maximum weight spanning tree does not hold any longer. Indeed, since delay is

[f] A maximum weight spanning tree of a topology is a spanning tree that maximizes the total link weight.

sensitive to the length of a path (because it is an additive measure) whereas maximum bandwidth is not, there may exist paths with larger bandwidth between ingress and egress if the delay component is ignored as illustrated in Figure 6.

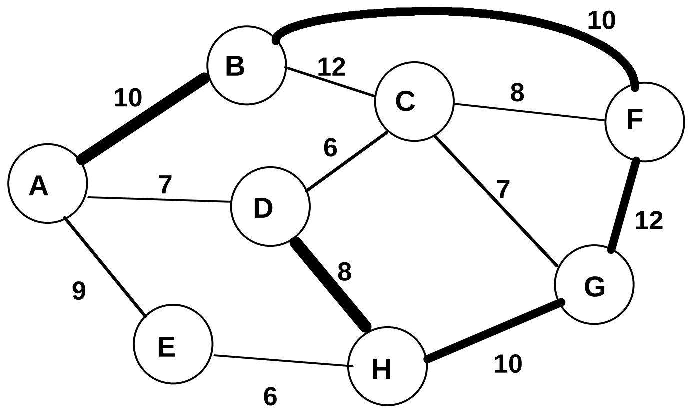


Figure 6. The link metrics is maximum bandwidth. The best maximum bandwidth path from node A to node D, denoted as P = A-B-F-G-H-D, is shown in bold and can carry 8 bandwidth units. If, in addition, each link has a unit delay component, the delay along that path P equals 5 delay unit. In case bandwidth and delay are treated equally fair, the path A-D with (7,1) can be preferable over path P with (8,5) depending on the comparison criterion.

The combination of additive and min(max) QoS measures assumes a criterion that allows to compare two (maximum bandwidth, delay) vectors, e.g. a convex functional $F_{ij} = w_B B_{ij}^{-1} + w_d d_{ij}$ where w_B and w_d are some positive (given) weights reflecting a cost and $B_{ij} = min_p(B_p)$ while $d_{ij} = S_p B_p$ and p denotes a path between i and j. This criterion should be the same as the one used to compute the 'best' vector values between all in- and egress pairs. An extension of our strategy amounts in solving a nonlinear set of equations in unknowns for the spoke vectors subject to non-negativity requirement for components. Again, applied to the simple example in Figure 3 and using the convex functional suggested above, the non linear set becomes

$$w_B \min^{-1}(B_x, B_y) + w_d (d_x + d_y) = F_{AB}$$
$$w_B \min^{-1}(B_x, B_z) + w_d (d_x + d_z) = F_{AC}$$
$$w_B \min^{-1}(B_y, B_z) + w_d (d_y + d_z) = F_{BC}$$

These settings are easily generalized to the case m>3. The complication clearly lies in finding the solution of such a non-linear set of equations with boundary conditions. In addition, the choice of the convex functional (a cost function) will always be debatable.

In conclusion, optimal information condensation of a combined set of min(max) and additive QoS measures is much more complicated than treating homogeneous sets and, in fact, still an open problem.

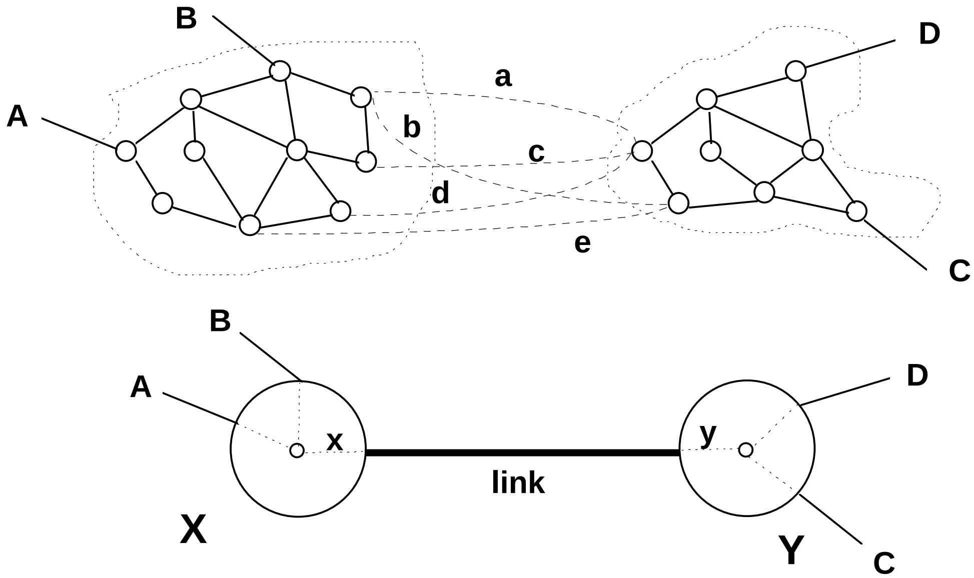


Figure 7 Link aggregation: the dashed links, a,b,c,d,e, between two peer groups are represented on the next hierarchical level by the bold aggregated link.

3. LINK AGGREGATION

In Figure 2, the relevant border nodes (ingresses or egresses) are visualized by links leading towards A, B and C. For simplicity, these links were thought of as single, not-aggregated links. The general problem of link aggregation is shown in Figure 7. We confine ourselves to additive QoS measures. We propose a node aggregation first on the two subnets (peer groups) via our strategy (sec. 2.12). Clearly (see Figure 8), the ingress-egress points of the peer groups are characterized via the link-sets {A,B,a,b,c,d,e} and {C,D,a,b,c,d,e}, respectively.

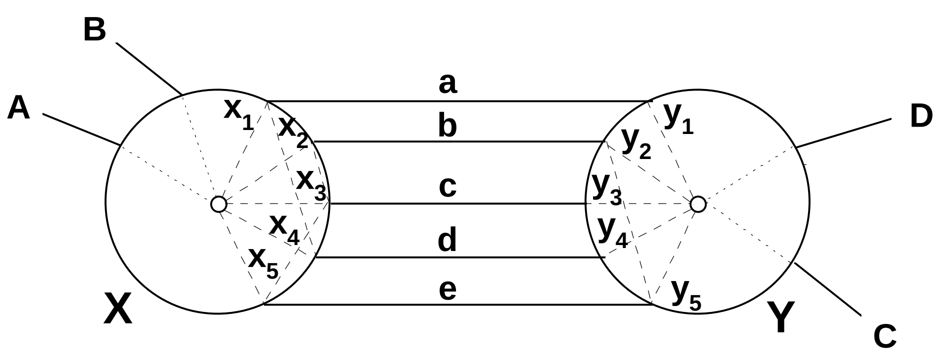


Figure 8 Link aggregation after node aggregation

Figure 8 assumes that the complete node aggregation procedure has been performed and that the spoke values x_i and y_i (and possibly the exceptional links that are also drawn in Figure 8) for each QoS measure are computed. The QoS values of the links *a, b, c, d, e* are known. From all possible paths between the nuclei of peer group *X* and *Y*, we choose the best QoS measure value and advertise this value in the aggregate picture (underneath in Figure 7). Thus, in case the QoS measure is additive, we advertise on the aggregated link that set {QoS_x, (exceptional link)$_x$, QoS_{link}, (exceptional link)$_y$, QoS_y} that minimizes the sum of the QoS values (possibly not all 5 path segments are needed). The exceptional links are important because they possess the best QoS measure value. Hence, although the path from nucleus X to nucleus Y using exceptional links can pass along more path segments, the QoS measure along that path may be the best one.

3.1 DISCUSSION AND ALTERNATIVES.

Because first node aggregation is performed, this strategy for link aggregation is fairly computationally intensive. In the end, possible exceptional links may have disappeared from the picture. Nevertheless, the best (final) value for a certain QoS, {QoS_x, (exceptional link)$_x$, QoS_{link}, (exceptional link)$_y$, QoS_y} may crucially depend on those exceptional link values. From the point of view of information condensation, link aggregation as presented here, contributes most significantly. One may wonder if the final results obtained by first aggregating the links (or what is more important the number of ingress-egress combinations) are still comparable, although obtained with considerably less effort. Judged a priori, we are in doubt about the quality of the results because it is far from obvious how to aggregate the links first. For instance, it is not difficult to condense the links *a, b, c, d, e* in Figure 7 to one with the best QoS value of the five original ones, but how do we connect the resulting condensed link to peer group nodes in *X* and *Y*?

The procedure of link aggregation connects complex nodes only by one (the 'best') logical link. Apart from simplicity, the advantage is a higher degree of uniformity over the hierarchical levels (at the lowest level every node is but connected by one link) resulting in the use of a same routing algorithm for the whole hierarchical structure. On the other hand, the drawback lies in the fact that this one link (which also is a physical link) is the only one used, even if there are more links possible to pass from one peer group to another. This implies that these best links may rapidly be 'outdated' and no longer the best QoS measure links. Hence, regular updates of the hierarchy (at least the link aggregation) seem required.

An alternative consists of advertising a number of *k* best QoS measure paths. This will result in less condensation, more complicated routing, but, a less rapid need to updating the hierarchy. Still another idea is to just omit the port numbers and to connect the nuclei of both peer groups by advertising the best value; on a lower layer, the routing may find out the best port and the corresponding specific link. The role of the hierarchy is then to announce over which peer groups a path must be followed (not which egress or port to use).

3.2 ADDITIONAL REMARKS.

Although the presented procedure is static, a dynamic extension may consist of a recomputation of (parts of) the hierarchy where significant changes have caused new flooding of topology state elements.

We define an *essential in- or egress* node on the physical level $k = 0$ as a node that has connectivity to other networks not belonging to the large original network. An important property of essential in(e)gresses is that they do not disappear on some level of the hierarchy due to condensation (see Figure 9, where node 'a' and 'o' are essential egresses). Hence, on each level the essential in(e)gress appears as a portnode in some complex node. These essential in(e)gress may be used by network management to achieve a more optimal hierarchical structure, as entry points for measurements and tests in the hierarchy or as attachment points for mobile networks (Dykeman *et al*., 1997).

4. EXAMPLES FOR ADDITIVE QOS MEASURES.

4.1 INFORMATION CONDENSATION OVER ONE HIERARCHICAL LEVEL.

To illustrate the proposed strategy for node and link aggregation, we have performed the computations for the additive QoS measure maxCTD or simply maximum delay. The network, randomly generated, is drawn in Figure 9.

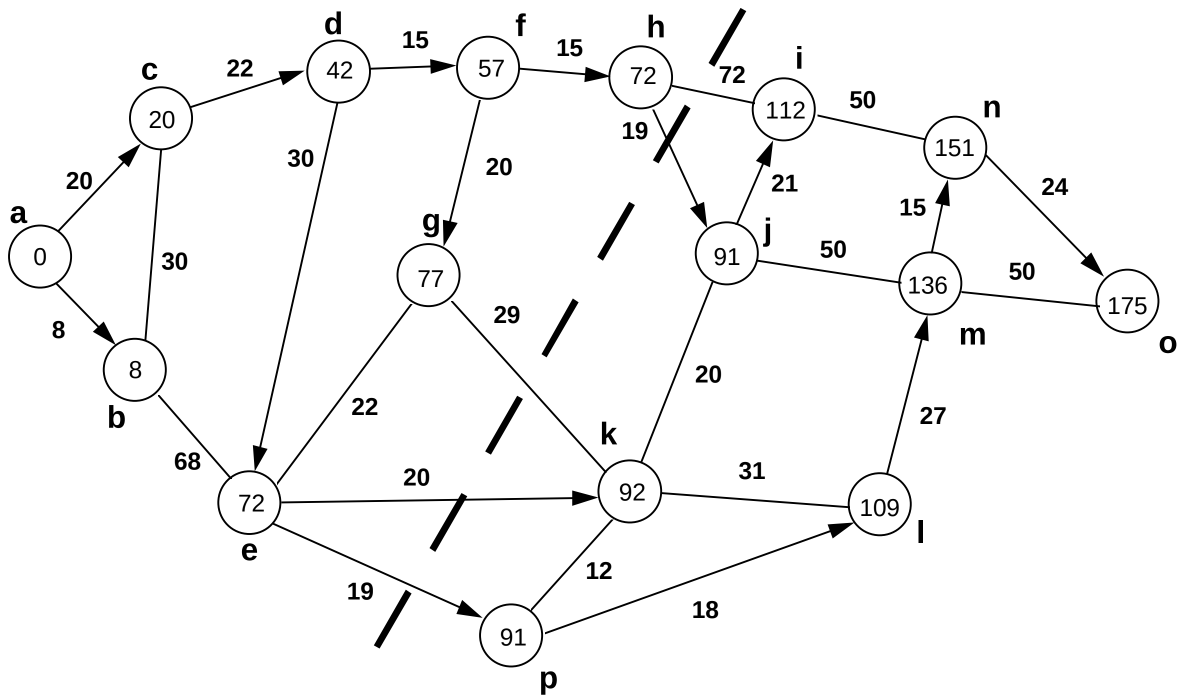


Figure 9 All links are bi-directional. The arrows indicate the shortest path from node a to any other node in the network. The QoS-distance (here the maximum delay) from any node to node a is the number inside the circle. The bold, dashed line shows two partitions of the network. Only node a and o are in-egresses of the orginal network

The shortest delay path from node a to node o is {a,c,d,e,p,l,m,n,o} and equals 175 units. We now present a node aggregation of the two parts of this network separated by the bold dashed line in Figure 9, and thereafter, a link aggregation. The result of this information condensation is compared with the value of the best delay path, namely 175 units.

4.1.1 Node Aggregation without accuracy ($\varepsilon \rightarrow \infty$).

The node aggregation of the whole network is shown in Figure 10. In both the subnetwork at left hand side, denoted by L, and the subnetwork at the right hand side, denoted by R, we have to determine the unknowns $\{x_1, x_2, x_3, x_4\}$ and $\{y_1, y_2, y_3, y_4, y_5\}$, possibly augmented with exceptional links (not yet drawn in Figure 10).

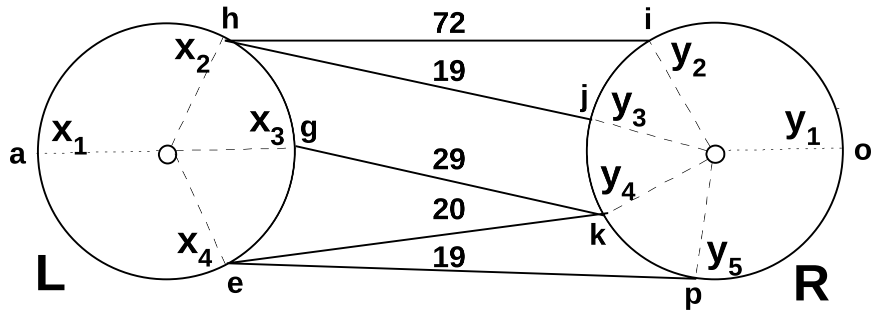


Figure 10 The aggregated node representation of the network drawn in Figure 9.

For the L subnetwork, we have to solve

					error
1 1 0 0	x_1		72	|80.33	8.83
1 0 1 0	x_2		77	|65.83	-11.16
1 0 0 1	x_3		72	|74.33	2.3
0 1 1 0	x_4	=	35	|37.33	2.3
0 1 0 1			57	|45.83	-11.16
0 0 1 1			22	|30.83	8.83

We invoke equation (2) to find $\{x_1, x_2, x_3, x_4\} = \{54.66, 11.16, 26.16, 19.66\}$. Since all components are positive, there is no need to introduce an exceptional link. Notice that the sum of the errors (~mean error) is almost zero (as expected since the overall square error was minimized) and that the distribution of the errors around the mean is symmetric.

Similarly, for the R subnetwork, the set linear equations are

1 1 0 0 0	y_1		101	|101.66
1 0 1 0 0	y_2		86	|82.66
1 0 0 1 0	y_3		79	|80.33
1 0 0 0 1	y_4	=	91	|92.33
0 1 1 0 0	y_5		21	|35.66
0 1 0 1 0			41	|33.33
0 1 0 0 1			53	|45.33
0 0 1 1 0			20	|14.33
0 0 1 0 1			32	|26.33
0 0 0 1 1			12	|24

and the corresponding best fitting solution $\{y_1, y_2, y_3, y_4, y_5\} = \{70.83, 19.5, 10.5, 12.83, 16.83\}$. Again, all components are positive and exceptional links are not needed. The reason is that the delay values in the network are more or less comparable.

In order to give a feeling for the quality of the node aggregation, we may simply fill in the results in the overdetermined linear set and compare the resulting values with the right hand side. These values are given in the last column, separated by a short line "|". Clearly, information condensation (without accuracy check as in step 2) gives rise to inaccuracies.

LINK Aggregation.

Since there are no exceptional links, the link aggregation is a quite easy process. We merely choose and advertise those values that minimize the delay path between the two nuclei. From Figure 10, we have that

$$x_2 + 72 + y_2 = 26.16 + 72 + 19.5 = 117.66$$
$$x_2 + 19 + y_3 = 26.16 + 19 + 10.5 = 55.66$$
$$x_3 + 29 + y_4 = 11.16 + 29 + 12.83 = 52.99$$
$$x_4 + 20 + y_4 = 19.66 + 20 + 12.83 = 52.49$$
$$x_4 + 19 + y_5 = 19.66 + 19 + 16.83 = 55.49$$

The best path is clearly that corresponding to the fourth equation. When performing the link aggregation, the resulting representation becomes

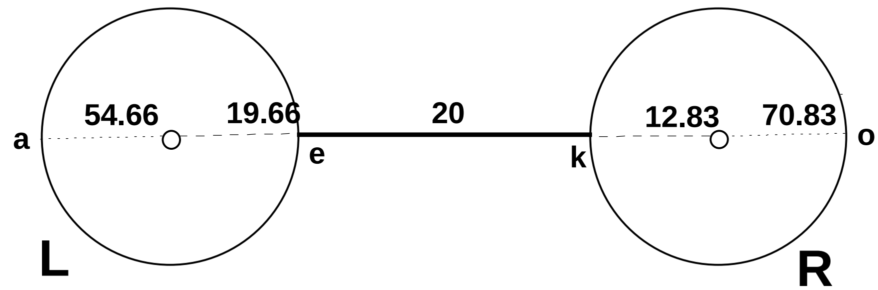


Figure 11 The result of information condensation (node and link aggregation) of the original network drawn in Figure 9

Comparing the best delay, computed from the original network in Figure 9, and that in the resulting aggregated representation, we obtain 177.98 against 175, which is not too bad without accuracy check, after all.

4.1.2 Information condensation with enhanced accuracy.

Using our strategy with an accuracy ε = 10%, we obtain for the same network (Figure 9), the result after aggregation plotted in Figure 12. One may verify that the original measure, e.g. from a to h, are approximated to within 10% using the best path from a to h (taking into account also exceptional links).

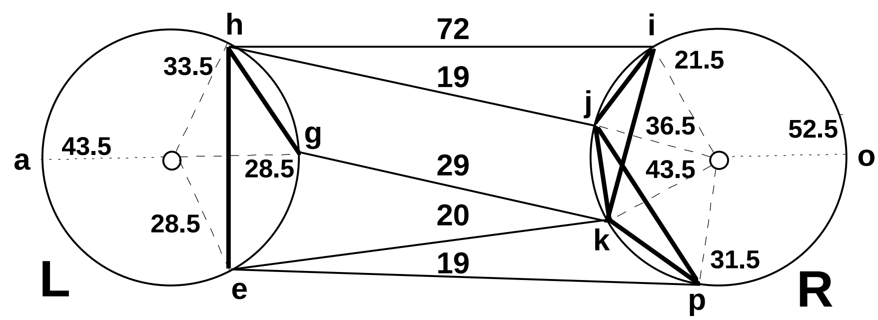


Figure 12 The aggregated node representation of the network drawn in Figure 9 with a 10% accuracy constraint. The values of the exceptional links in L are hg = 35 and he = 22. Those in R have the values ij = 21, ik = 41, jk = 20, jp = 32 and kp = 12.

The final picture after link aggregation is shown in Figure 13 with as best delay from a to o equals to exactly 175 (well within the required 10%) where as the previous not accuracy-constrained approach finished at 177.98.

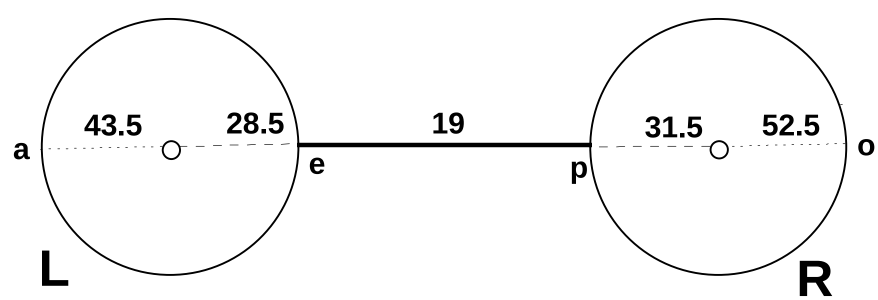


Figure 13 The result of information condensation (node and link aggregation) of the original network drawn in Figure 9 with a 10% accuracy constraint.

4.2 INFORMATION CONDENSATION OVER TWO HIERARCHICAL LEVELS.

Let us now illustrate, based on the same original network, how node aggregation (with $\varepsilon \rightarrow \infty$) on two hierarchical levels work. We divide the original network of Figure 9 in four different peer groups (as shown in Figure 14) instead of just two. In addition, only the nodes a,f,l,o are assumed to have links to other networks.

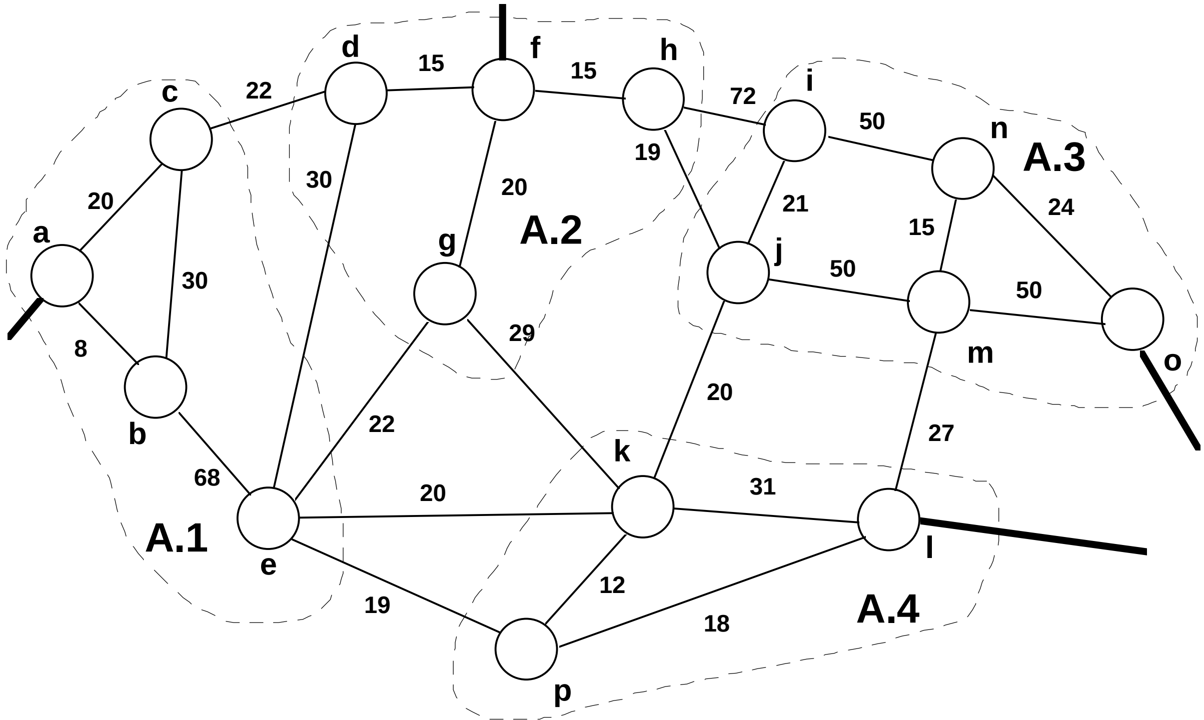
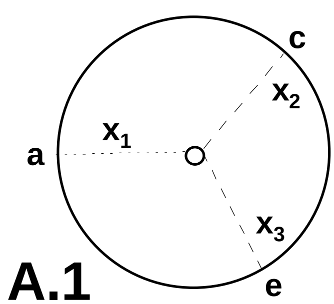


Figure 14 The original network of Figure 9 is organized in 4 different peer groups, A.1, A.2, A.3 and A.4.

4.2.1 First Level Node Aggregation.

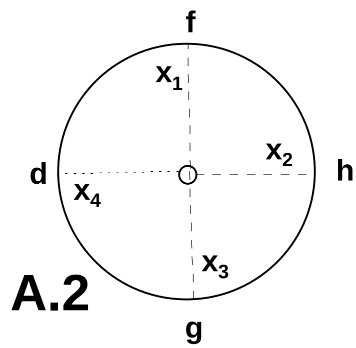


$$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} 20 \\ 72 \\ 98 \end{vmatrix}$$

Figure 15 Node aggregation of peer group A.1

When solving the linear set[g] for A.1 (see Figure 15), we obtain $\{x_1, x_2, x_3\} = \{-3,23,75\}$. Here, an exceptional link is needed. We choose as exceptional link the one with lowest value, namely ac. The remaining equation $x_1 + x_3 = 72$ and $x_2 + x_3 = 98$ constitute an underdetermined linear set. As additional equation, we choose $x_1 = x_3$ in order to stimulate symmetry. The solution then is $\{x_1, x_2, x_3\} = \{36,43,36\}$ complemented with exceptional link $x_{ac}= 20$.

[g] It is interesting to observe what the impact is of a small errors. The correct linear set for A.1 has as right hand side components {20,76,96} leading to $\{x_1, x_2, x_3\} = \{0,20,76\}$. Hence, in the correct case, we do neither have an exceptional link, nor a separate nucleus (because node a is nucleus).



$$\begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{vmatrix} = \begin{vmatrix} 15 \\ 20 \\ 15 \\ 35 \\ 30 \\ 35 \end{vmatrix}$$

Figure 16 Node aggregation of peer group A.2

The solution of the linear set for A.2 (see Figure 16) using equation (2) gives us the exact solution, $\{x_1, x_2, x_3, x_4\} = \{0,15,20,15\}$. This is not at all surprising, because we have tried to force a solution in the form as shown in Figure 16 because on the next hierarchical level, we will have to connect node f with another outside network.

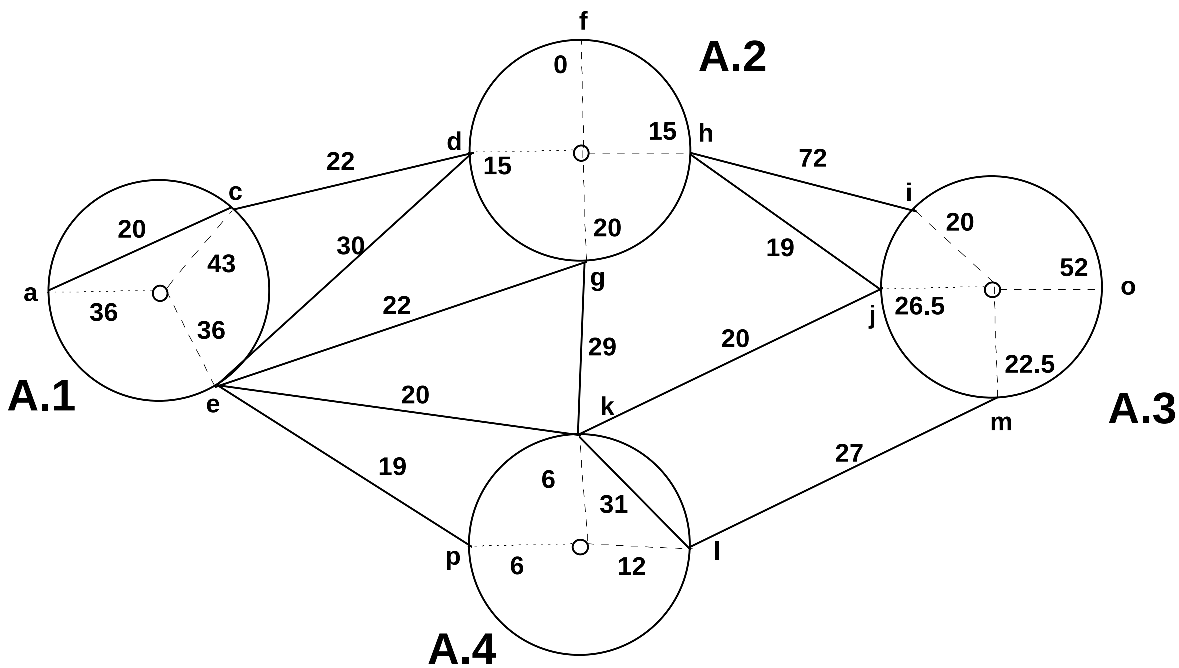


Figure 17. The result after node aggregation on the first hierarchical level

The computations for A.3 and A.4 are similar and we merely give the results referring to Figure 17 for the structure of each aggregated node. For A.3, we obtain as best fitting solution $\{x_1, x_2, x_3, x_4\} = \{52,20,26.5,22.5\}$ and the accuracy of the solution (by substituting the result in the linear equation) is {72, 155/2,149/2,91/2,85/2,48} as compared to the right hand side {74,100,50,21,65,50}. For A.4, similar to A.1, we must introduce an exceptional link and here we have chosen the worst value (to illustrate the difference with the proposed strategy). The result is $\{x_1, x_2, x_3\} = \{6,6,12\}$ complemented with exceptional link $x_{ki} = 31$.

<u>4.2.2 First Level Link Aggregation.</u>

Based on Figure 17, we have to compute the shortest delay path from a nucleus to his neighbor nuclei. This operation is quite basic. The result is drawn in Figure 18.

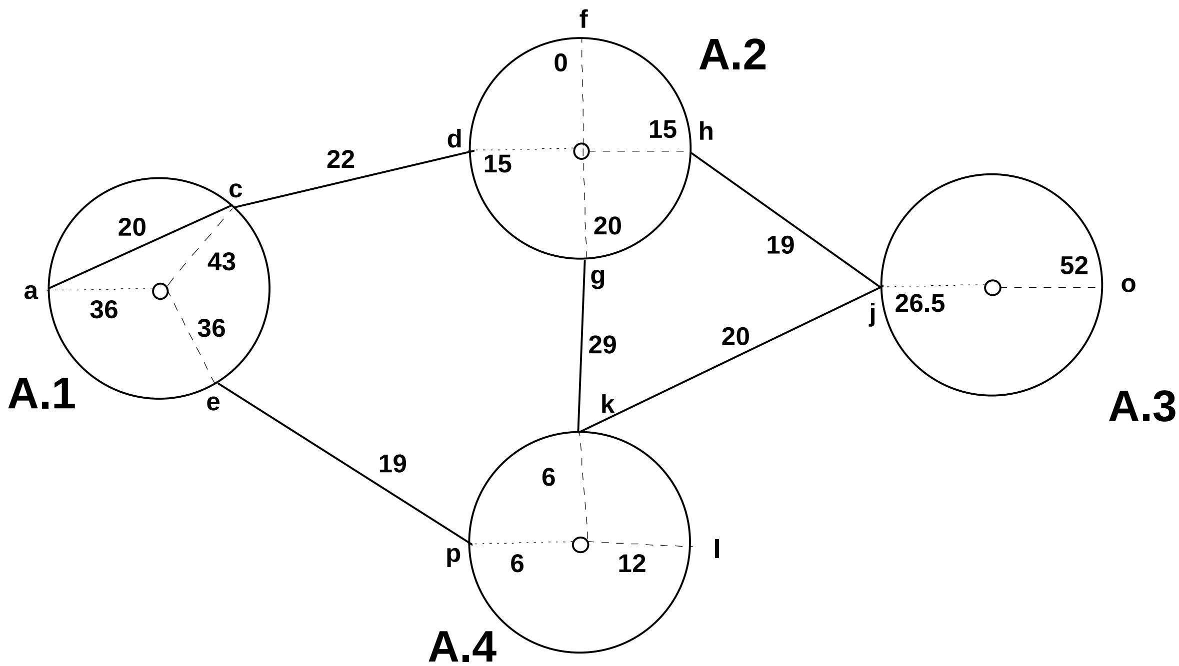
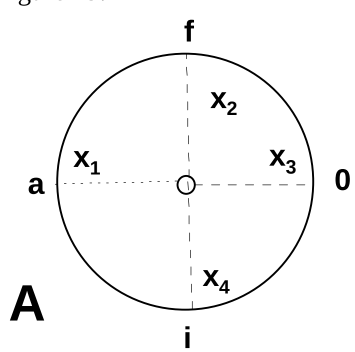


Figure 18 The final result (after node & link aggregation) on the first hierarchical level.

4.2.3 Second Level Node Aggregation.

The last step is to aggregate the subnet consisting of the aggregated peer groups A.1, A.2, A.3 and A.4. Since we need connection with other networks via the nodes a,f,i,o, the representation on this second hierarchical level is as drawn in Figure 19.



$$\begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{vmatrix} = \begin{vmatrix} 57 \\ 169.5 \\ 109 \\ 112.5 \\ 67 \\ 116.5 \end{vmatrix}$$

Figure 19 Node aggregation of peer group A

Before concentrating on the solution, we would like to point out that the goodness of the first level information condensation can be extracted from Figure 19. Indeed, we see that the shortest path from node a to node o (the reference taken before), now equals 169.5 compared to the exact value 175. Hence, the quality of this information condensation is slightly less than in previous case (where $\varepsilon \to \infty$) with just two peer groups.

Now, solving the set in Figure 19 with (2) results in $\{x_1, x_2, x_3, x_4\} = \{62.5, 13, 94, 41\}$ and an indication of the error follows via substitution of this result in the original equation, yielding {75.5, 156.5, 103.5, 107, 54, 135}. These values should be compared with the right hand side of the equation given in Figure 19. This finally ends our second level node aggregation. Now, we observe that the shortest path from node a to node o is 156.5 to be compared to the exact value of 175. Hence, as expected, the more information condensation, the lower the quality of the results.

5. CONCLUSIONS

For (multiple) additive QoS measures, we have presented how to effectively construct a hierarchical structure. The procedure consists of two basic steps: node aggregation followed by link aggregation. Our strategy (sec. 2.1.2) is the best way to aggregate nodal topology information for (multiple) additive QoS measures subject to a given accuracy (the relative error ***e***). This means that, for additive QoS measures, always an optimal (in the sense that information is maximally condensed) node aggregation with a guaranteed accuracy can be constructed. The procedure for link aggregation advertises the one, best link between two complex nodes as the 'aggregated' logical link, which again offers maximal condensation. In addition, it enhances uniformity in the data structure of the hierarchy, which in turn simplifies routing, at the compensation of a more rapid updating of the hierarchy. The accuracy parameter ***e*** allows fast verification whether a connection demanding certain QoS guarantee, QoS*, should be refused (if $QoS > QoS^*$) or not. Indeed, the routing process can focus on that level of hierarchy, say k, that encloses both source and destination. If the computed path on that level k leads to a QoS value exceeding $(1\text{-}ke)\ QoS^*$, the connection is blocked, else, signaling is activated with the proper path-list[h].

For one min(max) QoS measure (like bandwidth), we have reviewed Lee's optimal minimum spanning tree approach. Although the method is uniquely supreme for one min(max) measure, it is doubtful whether it can be generalized to multiple min(max) QoS measures featuring the same, elegant properties. In addition, the existence of good strategies for the link aggregation of min(max) QoS measures is questionable. Finally, optimal node and link aggregation for combined additive and min(max) QoS measures, is still believed to be an open problem.

6. ACKNOWLEDGMENTS

Discussions with various colleagues, in particular Hans De Neve, are much appreciated. Valuable comments of the reviewers are acknowledged.

7. REFERENCES

Althöfer, I., G. Das, D. Dobkin, D. Joseph and J. Soares, 1993, "On Sparse Spanners of Weighted Graphs", Discrete Comput. Geom., vol. 9, pp. 81-100.

ATMF, 1996, *Private Network Network Interface*, specification version 1.0, March

Behrens, J. and J. J. Garcia-Luna-Aceves, 1998, "Hierarchical Routing Using Link Vectors", IEEE INFOCOM'98.

Cormen, T. H., C. E. Leiserson and R. L. Rivest, 1990, *Introduction to Algorithms*, The MIT Press, Cambridge, Massachusetts.

De Neve, H. and P. Van Mieghem, 1998, "A Multiple Quality of Service Routing Algorithm for PNNI", Proceedings of the IEEE ATM Workshop, May 26-29, Fairfax, USA, pp. 324-328.

De Neve, H. and P. Van Mieghem, 1998a, "TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm", submitted to Journal of ACM.

Dykeman, D., I. Iliadis, P. Scotton, L. Frelechoux and S. Ray, 1997, "PNNI routing support for mobile networks", ATMF 97-0766.

Golub, G. H. and C. F. Van Loan,1989, *Matrix Computations*, North Oxford Academic, Oxford.

Huang, G. M. and S. Zhu, 1996, *A New HAD Algorithm for Optimal Routing of Hierarchically Structured Data Networks*, IEEE Trans. on Parallel and Distributed Systems, vol. 7, No. 9.

Kleinrock, L. and F. Kamoun, 1977, *Hierarchical routing for large networks. Performance evaluation and optimization*, Computer Networks 1, pp. 155-174.

[h] In PNNI, this path-list is called the designated transfer list (DTL).

Lanczos, C.,1988, *Applied Analysis*, Dover Publications, Inc., London.
Lee, W. C., 1995, “Spanning Tree Method for Link State Aggregation in Large Communication Networks”, IEEE INFOCOM’95, pp. 297-302.
Lee, W. C., 1995a, “Topology Aggregation for Hierarchical Routing in ATM Networks”, ACM Sigcomm., vol. 25, No. 2, April, pp. 82-92.
Montgomery, M. and G. de Veciana, 1998, Hierarchical Source Routing Through Clouds, IEEE INFOCOM’98.
Peleg D. and A. A. Shchäffer, 1989, “Graph Spanners”, Journal of Graph Theory, vol. 13, No. 1, pp. 99-116.
Peleg D. and E. Upfal, 1988,”A trafeoff between space and efficiency for routing tables”, ACM Proceedings, pp. 43-52.
Van Mieghem, P., 1997, “Estimation of an Optimal PNNI Topology”, Proceedings of the IEEE ATM’97 Workshop, May 26-28, Lisboa, Portugal, pp. 570-577.
Van Mieghem, P., 1998, “Routing in a Hierarchical Structure”, Proceedings of the first IEEE International Conference on ATM, ICATM’98, June 22-24, Colmar, France.
Van Mieghem, P., 1998a, “Dividing a Network into Peer Groups to Build a Hierarchical Structure”, Proceedings of the first International Workshop on the Design of Reliable Communication Networks, May 17-20, Brugge, Belgium, O16.

8. APPENDIX

Lemma. Let $P = (M^TM)^{-1} M^T$ and $M_{m(m-1)/2 \times m}$ as defined in (1). Then P equals M^T where the ones in M^T are changed for $1/(m-1)$ and the zero’s for $-1/(m-1)(m-2)$.

This general solution of (2), $Q = P. F$, is computed very rapidly. In matrix algebra, the amount of work involved or the complexity is usually expressed in the number of multiplicative operations (Golub and Van Loan, 1989). Hence, the complexity to compute $Q = P. F$ equals $C(Q) = m^2(m-1)/2$ flops (floating point operations). In this paper, we have used the less precise notion of “order” to express the complexity, thus, $C(Q) = O(m^3/2)$ for large m.

The verification of the lemma is as follows.

Let us first concentrate on the simplest case m = 3. The set (1) can be solved exactly since the inverse of M exists, namely

$$M^{-1} = 1/2 \begin{vmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{vmatrix}$$

Hence, the solution (QoS_1, QoS_2, QoS_3) = 1/2 ($f_1+ f_2- f_3$, $f_1- f_2+ f_3$, $-f_1+ f_2+ f_3$) will only have a negative component if one of the components of F is larger than the sum of the two other components, thus, if $f_i+ f_j< f_k$ for any combination of (i,j,k).

The matrix M in case m = 5 equals

$$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{matrix}$$

and the corresponding $(M^TM)^{-1} M^T$ is

1/4	1/4	1/4	1/4	-1/12	-1/12	-1/12	-1/12	-1/12	-1/12
1/4	-1/12	-1/12	-1/12	1/4	1/4	1/4	-1/12	-1/12	-1/12
-1/12	1/4	-1/12	-1/12	1/4	-1/12	-1/12	1/4	1/4	-1/12
-1/12	-1/12	1/4	-1/12	-1/12	1/4	-1/12	1/4	-1/12	1/4
-1/12	-1/12	-1/12	1/4	-1/12	-1/12	1/4	-1/12	1/4	1/4

When m = 6, we have

$$\begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{matrix}$$

$$
\begin{array}{cccccc}
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

with corresponding $(M^TM)^{-1}\,M^T$,

1/5	1/5	1/5	1/5	1/5	-1/20	-1/20	-1/20	-1/20	-1/20	-1/20	-1/20	-1/20	-1/20	-1/20
1/5	-1/20	-1/20	-1/20	-1/20	1/5	1/5	1/5	1/5	-1/20	-1/20	-1/20	-1/20	-1/20	-1/20
-1/20	1/5	-1/20	-1/20	-1/20	1/5	-1/20	-1/20	-1/20	1/5	1/5	1/5	-1/20	-1/20	-1/20
-1/20	-1/20	1/5	-1/20	-1/20	-1/20	1/5	-1/20	-1/20	1/5	-1/20	-1/20	1/5	1/5	-1/20
-1/20	-1/20	-1/20	1/5	-1/20	-1/20	-1/20	1/5	-1/20	-1/20	1/5	-1/20	1/5	-1/20	1/5
-1/20	-1/20	-1/20	-1/20	1/5	-1/20	-1/20	-1/20	1/5	-1/20	-1/20	1/5	-1/20	1/5	1/5

The latter matrix has the following structure (more easily seen if we replace 1/5 by 1 and -1/20 by 0):

1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	1	1	0	0	0
0	0	1	0	0	0	1	0	0	1	0	0	1	1	0
0	0	0	1	0	0	0	1	0	0	1	0	1	0	1
0	0	0	0	1	0	0	0	1	0	0	1	0	1	1

We observe that the first row consists of (m-1) ones. The remaining (m-1) rows exhibit in the first (m-1) columns the $I_{(m-1)x(m-1)}$ identity matrix, in the (m-2) remaining row, in the next (m-2) columns, the $I_{(m-2)x(m-2)}$ identity matrix and so on. The other elements on the rows are (m-i) consecutive one's followed by zero's. The sum of the ones on each row equals (m-1). This observation is general and illustrates the lemma.

Now, the solution will have a negative component if for certain permutations of the set $(j_1, j_2, ..., j_m)$ holds that

$$\sum_{i=1}^{m-1} f_{j_i} < \frac{1}{m-2} \sum_{i=m}^{m(m-1)/2} f_{j_i} \qquad \text{(A.1)}$$

Observe that if all $f_i = f$, the relation (A.1) is never satisfied because the left hand side equals (m-1)f and the right hand side equals (m-1)f/2. Indeed, in this perfect symmetrical situation, the general solution is precisely $(QoS_1, QoS_2, ..., QoS_3) = f/2\ (1,1,...,1)$. Furthermore, if m = 3, (A.1) reduces to the well-known triangular inequality. No simple method to check (A.1) a priori seems to exist. Therefore, the proposed algorithm (with possibly many iteration) is hard to simplify. Moreover, in case of iterations, i.e. when exceptional links are introduced, the regular structure of M is destroyed by deleting rows corresponding to an exceptional path. Therefore, no such simple solution as Q = P.F can be hoped for and numerical computation will increase significantly.