Two Approaches for Aggregation of Peer Group Topology in Hierarchical PNNI Networks

Kiran R. Bhutani [*] Abdella Battou Bilal Khan [†]

October 26, 2000

Abstract

We propose two methods for aggregation of peer group topology in hierarchical ATM networks. In our proposed schemes, aggregation is performed in two steps: First, we transform the original peer group topology into a complete graph on border nodes. Then, we compute a compact representation of this complete graph. The final representations generated are terse, yet preserve the critical information of peer group structure that is needed for efficient call setup and routing. The first scheme is compatible with the current PNNI, while the second scheme extends the interpretation of these standards.

Both proposed aggregation methods transform a given peer group into a star graph representation. Our first approach optimally preserves, in a least square sense, the original costs of routing through the peer group. Our second approach assigns a weighted vector to the nucleus of the Logical Group Node, which quantifies the error in the compact representation. The two schemes are dual, in the sense that the first is best suited for peergroups where traffic patterns are unpredictable, and the second is suited for peergroups where traffic patterns can be characterized. Both the proposed schemes are practical: For peer groups with nodes V, links E, and n border nodes $B \subset V$, the approaches run in $O(n|V|\log|V| + n|E| + poly(n))$ time. The size of the final representation is small (linear in the number of border nodes) and can be computed efficiently. The scalability of the proposed algorithms makes them well-suited for use in practice. We also present a general method for measuring the degree of confidence in an aggregation scheme and give examples that illustrates the proposed techniques.

1 Introduction

As the advantages of ATM technologies are recognized, we anticipate the size of operational ATM clouds to grow at a rapid pace. In order to insure scalable operation of these networks, the PNNI standards for routing have been recently adopted. PNNI defines a set of protocols for a *hierarchic* network that provide the basis for efficient routing.

[*]Permanent address: The Catholic University of America. Department of Mathematics, Washington D.C. 20064. This work was funded by CIPS Corp., Bethesda, MD, under a subcontract from Kaman Sciences Corporation.

[†]Kaman Sciences Corp. At NRL

The PNNI hierarchy begins at the lowest level, where switches are clustered into *peer groups*. The members of a peer group execute a protocol to synchronize their topology databases in order to maintain concordant views of the group. As part of this protocol, each peer group elects a node that is responsible for aggregating and distributing information about the group, to higher levels of the PNNI hierarchy. Such a node is referred to as the *peer group leader*. A peer group at a higher level of hierarchy is a collection of *logical group nodes* (LGNs), each of which represents a lower level (or *child*) peer group. The function of a logical group node and the peer group leader of its child peer group are closely related: The peer group leader forwards reachability information and *aggregated topology* information about its group to the logical group node. Reachability information is a summary of (lowest level) ATM addresses that can be reached within that peer group. Aggregated topology, refers to a summary of peer group structure from which characteristics (such as bandwidth availability, transmission delay, etc.) for routes into and across the peer group can be estimated.

The PNNI hierarchy is defined recursively by this process of clustering nodes into peer groups and then representing each cluster by a logical group node at the next higher level. There can be up to 104 levels in the hierarchy. A higher level group uses the aggregated information advertised by its constituent lower level groups to decide which ones to route a connection through. Feeding information up the PNNI hierarchy is necessary for the creation of the hierarchy itself and for distributing routing information about child peer groups. Conversely, feeding information down the hierarchy is also essential to allow nodes in the lower level peer groups to route to all reachable destinations. The critical interface in this two-way flow of information is the one between a logical group node and the peer group leader of its child peer group. Because PNNI permits aggregation to take place during the upward flow of information across this interface, the protocol provides a natural mechanism for reducing the communication overhead involved in maintaining network routing information. In addition, aggregation permits the lower level domains to hide sensitive details about their internal structure, as mandated by security requirements in todays networks. The long term success of ATM as an end-to-end solution for networking depends largely on the success of PNNI. The success of PNNI, in turn, depends on how well peer group topology information can be summarized by aggregation schemes before it is reported to distant regions in the network.

As one degenerate case we might opt for no aggregation i.e. every peer group leader forwards the full information about the peer group to its logical group node. In this case, every node in the network will be maintaining a database that describes the entire network. Such a scheme would guarantee optimal route selection, but would result in prohibitively large amounts of network traffic, just to maintain accurate topology databases. Even if the resulting background traffic was not objectionable, its volume would clearly grow quadratically in the size of the network, yielding a non-scalable solution.

In the other extreme case, we might opt for total aggregation i.e. every peer group leader forwards nothing to the logical node. This scenario is equivalent to one where every node remains up-to-date about only switches and links in its own peer group. Such a scheme has an advantage that it produces minimal traffic due to PNNI database synchronizations. However switches, not having any knowledge of the larger network structure, would make uninformed, greedy routing decisions. This would result in the allocation of excessively long routes i.e. bigger call setup times, crankbacks and suboptimal network call admission rates, etc.

In this paper we probe the space of solutions between these two trivial aggregation schemes. The topology information we seek to aggregate includes the graph representation of a peer group and associated quantities or *topology state parameters*.

Topology state parameter refers to a quantity that describes individual nodes or links in a peer group. Topology state parameters fall into two classes: the metrics and the attributes. The metrics are those topology state parameters that require the values of the parameter for all links and nodes along a given path to be combined in order to determine whether the path is acceptable for carrying a given connection (e.g. link delay). Attributes are those topology state parameters that are considered individually to determine whether a specific (single) link or node is acceptable for carrying a given connection (e.g. available bandwidth).

Some topology parameters required by PNNI are: cell loss ratio, maximum cell rate, available cell rate, and administrative weight, Additionally, cell delay variation for CBR and real-time VBR service categories, and maximum cell transfer delay for CBR, real-time VBR and non real-time VBR service categories.

2 Topology Aggregation

A desirable aggregation scheme must

1. **Transform a peer group structure into a compact form.** This is necessary to capitalize on the intended purpose of aggregation, namely to reduce background traffic required to maintain routing information for large networks.

2. **Retain information about those aspects of the topology that are most critical to making good routing decisions.** The aggregation process results in each switch seeing a suitably coarsened (and hopefully adequate) view of the entire network topology: A switch sees the true topology only for its own peer group, but has progressively more aggregated views of farther regions of the network. Since this aggregated topology information will be used to select the routes for connection requests, we must make sure that our aggregation scheme does not have too detrimental effect on route optimality.

3. **Be efficiently computable.** Aggregation will need to be re-performed every time peer group topology changes significantly. Having a lightweight aggregation algorithm ensures that we can have a reasonable criterion defining a “significant change”, without putting an unacceptable burden on the network switches that have to compute the aggregated representation.

.

There are two paradigms for topology aggregation currently supported in PNNI: the *Symmetric-node* approach and the *Star* approach. In a Symmetric-node model the peer group topology is collapsed to a single node, and one value (usually the worst-case value), called the *radius*, is advertised for each parameter. While this scheme greatly reduces the information propagated, it may result in very poor routing decisions, particularly if the number of border nodes and the variance in their connectivity is large. The Star model augments the Symmetric-node approach as follows:

topology is aggregated into a representation with one central virtual node called the **nucleus** and n border nodes. This central node is explicitly connected to all of the border nodes via *spokes*. The spokes may be given weights different from the radius, in which case they are referred to as *exceptions*. In the final level of sophistication, PNNI aggregation allows the aggregated representation to specify logical links directly between borders; such links are called *bypasses*.

To date, no specific Symmetric-node or Star aggregation schemes have been defined in the PNNI standard. In this paper, we present two approaches for topology aggregation with respect to a single metric parameter. Specifically, we describe two techniques for computing good Star representations of a peer group.

3 Mathematical Model

We represent a peer group as a multi-weighted undirected graph $G(V, E, B, P_1, \cdots, P_k)$ with nodes V, edges E, where $B \subseteq V$ is the fixed set of n border nodes, and $P_1, \cdots, P_k$ are the required PNNI topology state parameters. The edge vector $e_{ij} = ({P_1}^{ij}, {P_2}^{ij}, \cdots, {P_k}^{ij})$ consists of the values of the parameters for the link joining V_i and V_j.

3.1 Preliminaries

Let $\delta_{P_i}(W, B_i)$ be the value of the best path between the non-border node W and border node B_i with respect to P_i. Specifically,

- for additive metric: best path means the shortest path.
- for bandwidth: best path means the path with maximum available bandwidth. The bandwidth available on a path is the minimum of the bandwidth available on the edges that form the path.
- for cell-loss ratio: best path means a path with minimum value for the cell loss parameter. (Cell loss ratio along a path is approximated as the maximum value of the cell loss ratio along the edges that form the path[1]).

Define $bd(B_i, B_j)$, the *boolean distance* between a pair of border nodes B_i and B_j;

$$bd(B_i, B_j) = \begin{cases} \phi, \text{ if } \; B_i = B_j \\ \{\, w \mid w \text{ is non-border node on a simple path between } B_i \text{ and } B_j\}, \text{ if } \; B_i \neq B_j \end{cases}$$

Note: The boolean distance between two nodes is a set. It is easy to see that the boolean distance is a metric, defined in terms of equality, containment and union of sets.

Define ω, the set of *candidate nodes* as

[1]This is a valid approximation only under the assumption that the average length of a path is much smaller than the reciprocal of the cell loss ratio (in present networks cell loss ratios are on the order of 10^{-9}).

$$\omega = \{W \mid W \in V - B,\ W \notin \bigcup_{1 \le i < j \le n} bd(B_i, B_j)\}$$

.

It is clear from this definition, that ω consists of all non-border nodes in the peer group that lie on a simple path between some pair of border nodes.

We define the *status* of every candidate node $W \in \omega$ with respect to the parameter P_i, and denote this quantity as $S_{P_i}(W)$. We also define the *status* of the peer group with respect to parameter P_i, and denote this quantity by r_i. The precise definitions of r_i and $S_{P_i}(W)$ depend on the particular parameter P_i, and are given below:

P_i is an **additive metric**:	$S_{P_i}(W) = \sum_{B_i \in B} \delta_{P_i}(W, B_i)$	$r_i = \min_{W \in \omega} S_{P_i}(W)$
P_i is **bandwidth**:	$S_{P_i}(W) = \min_{B_i \in B} \delta_{P_i}(W, B_i)$	$r_i = \max_{W \in \omega} S_{P_i}(W)$
P_i is **cell-loss ratio**:	$S_{P_i}(W) = \max_{B_i \in B} \delta_{P_i}(W, B_i)$	$r_i = \min_{W \in \omega} S_{P_i}(W)$

The *central status nodes* of the peer group relative to the pair (B, P_i) are defined to be the set of nodes $W \in \omega$ such that $S_{P_i}(W) = r_i$. A central status node may be thought of as a node on which one can center the smallest radius sphere (with respect to parameter P_i), and still capture all of the border vertices.

4 Algorithms

4.1 Preprocessing

Whenever we need to aggregate, we first generate a weighted complete graph on border nodes from the original peer group. We present two methods for generating the weights d_{ij} (between borders B_i and B_j) on the edges of this complete graph:

(I) d_{ij} is the measure of the shortest path between B_i and B_j.

Complexity. We compute the best paths from every border node, to every other node in the network, by running Dijkstra's algorithm n times, starting once from each of the border nodes. This requires $O(n|V|\log|V| + n|E|)$ steps. The space needed is $O(|V| + |E|)$, which is what is required to hold the adjacency list graph representation of the peer group.

(II) d_{ij} is the measure of the best path between B_i and B_j through a central status node.

Complexity. First, we run Dijkstra's algorithm from all border nodes, in $O(n|V|\log|V| + n|E|)$ steps, as in (I). This gives us the $\delta_{P_i}(W, B_i)$. Then for each non-border node W, we use the computed distances to the border nodes to compute $S_{P_i}(W)$, requiring $O(n|V|)$ additional steps. We select a central status node appropriately, by keeping track of the best value of the $S_{P_i}(W)$ as we compute them. Finally, since we have already computed the distances from the border nodes to the chosen central status node, we can compute the pairwise shortest distances between all border nodes in an additional $O(n^2)$ steps. The space requirement is the same as for method (I) above.

If d_{ij} is the weight on the edge joining B_i and B_j in the complete graph then this is the value that we wish to advertise in the parent group as the cost of traversing the peer group between border nodes B_i and B_j.

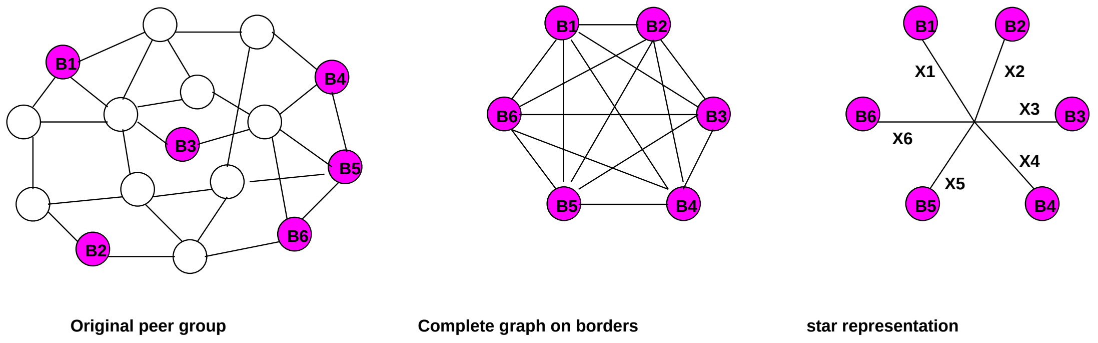

Figure 1: Basic steps for topology aggregation

4.2 Approach 1

We would like to represent the original peer group as a star graph with weights x_i advertised on the edges such that $x_i + x_j = d_{ij}$ for all $i, j (i \neq j)$.

In general, with n border nodes, such system of equations can be represented as $AX = D$ where, A is $\binom{n}{2} \times n$ matrix, D is $\binom{n}{2} \times 1$ column vector of the costs to be advertised, and X is $n \times 1$ column vector of unknown weights to be determined for the star representation of the Logical Group Node.

This system $AX = D$ is an over determined system of equations which is usually an inconsistent system. That is, given an $m \times n$ system $AX = D$ with $m > n$, we cannot in general find a vector $x \in R^n$ for which $AX = D$.

In such cases, one resorts to finding a vector, $\hat{x} \in R^n$ for which $A\hat{x} = \hat{D}$ is closest to D in the least square solution to the system $AX = D$. Indeed, $\hat{x}$ in R^n is such that$\|r(\hat{x})\|^2 = \|D - A\hat{x}\|^2$ is minimum over all $x \in R^n$. Thus to solve $AX = D$, we solve

$$A^T AX = A^T D$$

The above represents an $n \times n$ system of linear equations, referred to as the normal equations. The $\hat{x}$ obtained are the values advertised as *exception* spoke edges of the star graph that represents the peer group in the next higher level. The size of this aggregated representation, and hence the communication incurred per peer group, is $O(n)$.

This approach is well-suited for peer groups which experience unpredictable traffic patterns, since it computes the star representation that minimizes, in a least squares sense, *the average distortion over all transits of the peer group.*

4.2.1 Computational Complexity of Approach 1

We compute $(A^TA)^{-1}A^T$ only once (off-line) beforehand in $O(n^4)$ steps[2].

Each time we need to re-aggregate: we multiply the precomputed $(A^TA)^{-1}A^T$ matrix by the current value of the vector D. This multiplication can be done in $O(n^3)$ time. The space required to hold the precomputed value of $(A^TA)^{-1}A^T$ and to perform the necessary multiplication is $O(n^3)$.

Thus the time requirement to perform an aggregation using Approach 1 (including preprocessing) is $O(n|V|\log|V| + n|E| + n^3)$, and the space requirement is $O(|V| + |E| + n^3)$.

4.3 Approach 2

A *transit* is an undirected path through the peer group whose endpoints are distinct border nodes. We call an unordered pair of border nodes (B_i, B_j) *critical* if we cannot tolerate any discrepancy between the information advertised in the aggregated representation, and the true value d_{ij} of the parameter for the transit between B_i and B_j. **Note:** In the entire discussion that follows, we deal with unordered pairs, implying that the element denoted (B_i, B_j) is equivalent to the element denoted (B_j, B_i).

Select a set of *preserved transits* $P^+ = \{(i,j) \mid (B_i, B_j) \text{ is critical}\}$. Our approach requires $|P^+| \leq n$; the precise conditions on this set P will be explained shortly.

Define the set of non-critical transits $P^- = \{(p,q) \mid B_p, B_q \text{ distinct border nodes}, (p,q) \notin P^+\}$. Note that $|P^+| + |P^-| = \binom{n}{2}$.

Now for each element $(i,j) \in P^+$ define the linear equation ϕ_{ij}

$\phi_{ij}: \quad x_i + x_j = d_{ij}$

And for each element $(p,q) \in P^-$ define the linear equation ψ_{pq}

$\psi_{pq}: \quad x_p + x_q + k_{pq} = d_{pq}$

The ψ_{pq} and ϕ_{ij} together specify a system of $\binom{n}{2}$ linear equations in $\binom{n}{2}$ unknowns, where n is the number of border nodes. The k_{pq} and $x_i's$ are the unknowns which have to be determined. We represent this system in matrix form $\mathbf{A}\vec{x} = \mathbf{D}$,

$$\left[\begin{array}{c} \vdots \\ \cdots \phi^*{}_{ij} \cdots \\ \vdots \\ \hline \vdots \\ \cdots \psi^*{}_{pq} \cdots \\ \vdots \end{array}\right] \left[\begin{array}{c} \vdots \\ x_i \\ \vdots \\ \hline \vdots \\ k_{pq} \\ \vdots \end{array}\right] = \left[\begin{array}{c} \vdots \\ d_{ij} \\ \vdots \\ \hline \vdots \\ d_{pq} \\ \vdots \end{array}\right]$$

[2]Computing (A^TA) requires $O(n^4)$ steps. The inverse $(A^TA)^{-1}$ is formed by solving the system $(A^TA)x = \vec{e_i}$, $i = 1 \cdots n$, where $\vec{e_i}$ is the n-dimensional vector which has value 1 in the ith component but is 0 elsewhere. Each of these n solutions can be computed using the LUP decomposition algorithm in $O(n^2)$ steps, so the matrix is inverted in $O(n^3)$ steps. The final product $(A^TA)^{-1}A^T$ is computed in $O(n^4)$ steps. So the entire procedure takes $O(n^4)$ time.

where ${\phi^*}_{ij}$ and ${\psi^*}_{pq}$ are the 0/1 coefficients corresponding to the linear equations ϕ_{ij} and ψ_{pq} respectively.

Since each ψ_{pq} introduces a new free variable k_{pq}, it is clear that the coefficient matrix $\mathbf{A}$ of the above system is non-singular if and only if the system consisting of just the ϕ_{ij} has a solution.

We can now more fully describe how to select the set P^+: We sample the actual probabilities of the transits, (or get some estimate of the traffic patterns between every pair of borders nodes B_i and B_j). Then we choose the n most demanding of these traffic routes, with the additional constraint that the resulting system of ϕ_{ij} be non-singular.

We compute the unique solution $\vec{\mathbf{x}}$, and give the following interpretation to the values of the components of the solution vector: The value of x_i will be the quantity advertised on the *exception* spoke between the nucleus and border node B_i in the star graph. The value of k_{pq} is the additive distortion between the truth and what is advertised about the (non-critical) transit between border nodes B_p and B_q.

For any fixed border node B_i we consider the multiset E_i consisting of the $n-1$ transit errors,

$$E_i = \left\{ k_{iq} \mid (i,q) \in P^- \right\} \cup \{0 \text{ for each } (i,q) \in P^+\}$$

We define $k_i = f(E_i)$, where f is some aggregation function[3], and consider k_i to be a measure of the average error inherent in the star representation for paths through border B_i of the peer group.

The final representation of the peer group is the set of x_i (on the exception spokes of the star), and the set of errors k_i. These k_i values serve as a "disclaimer", specifying the average error inherent in the star representation for transits through border B_i. The representation constructed by this approach is of size at most $2n = O(n)$.

Unfortunately, the current version of the PNNI standard does not have any provision for advertising the set of errors k_i . We recommend that the PNNI aggregation model be augmented to permit the assignment of topology state parameters to the nucleus. If such a scheme is adopted, the set of errors k_i could be associated with the nucleus and included as part of the aggregated representation. Remote switches trying to decide whether or not to route through a particular peer group could then take into account not only its aggregated representation, but also the k_i, which are the peer group's self-assessment of the fidelity of its advertised representation. In the absence of such an extension to PNNI, we can still use Approach 2, and only advertise the set of x_i on the exception spokes of the star (without the k_i).

This approach is well-suited for peer groups which experience regular, characterizable traffic patterns, since it computes the star representation that *completely preserves the truth about the most commonly occurring transits.*

4.3.1 Computational Complexity of Approach 2

We compute $\mathbf{A}^{-1}$ only once (off-line) beforehand in $O(n^6)$ steps, just as in Approach 1.

[3]In the example discussed later in this paper we take f to be the average absolute value function i.e. if $E = \{e_1, e_2, \cdots, e_{n-1}\}$, then $f(E) = \frac{1}{n-1} \cdot \sum_{i=1}^{n-1} |e_i|$

Each time we need to re-aggregate: we multiply the precomputed A^{-1} matrix by the current value of the vector D. This multiplication can be done in $O(n^4)$ time, and the total space required to store A^{-1} and perform this multiplication is $O(n^4)$. Finally, we compute the k_i using the aggregation function f as above, (and if f runs in time linear in the size of its argument set E_i), this is completed in at most $O(n^2)$ time.

Thus the time requirement to perform an aggregation using Approach 2 (including preprocessing) is $O(n|V|\log|V| + n|E| + n^4)$, and the space requirement is $O(|V| + |E| + n^4)$.

5 Remarks

5.1 Lowering Distortion Further

If this representation is still found to deviate unacceptably from the true topology, it can be augmented in a systematic way to reduce the distortion, but at the cost of a larger representation. Let us assume that we desire to finish with a representation that is within ϵ multiplicative distortion of the true topology. To achieve this we can add bypass links between borders B_i and B_j if

$$\mathbf{\frac{1}{1+\epsilon} \leq \frac{d_{ij}}{d^*{}_{ij}} \leq 1+\epsilon}$$

where d_{ij} is the actual cost of the (B_i, B_j) transit and $d^*{}_{ij}$ is the cost currently advertised. The weight of such the additional bypass link would be d_{ij}.

5.2 Ensuring Conservative Advertisement

We want to advertise d_{ij} as the cost of traversing the peer group between border nodes B_i and B_j. The least square solution in Approach 1 provides the best $\hat{x} = (\hat{x_1}, \cdots, \hat{x_n})$, and results in $\hat{x_i} + \hat{x_j}$ being advertised as the distance between border B_i and B_j. In some instances we may find $\hat{x_i} + \hat{x_j} \geq d_{ij}$. If we desire this advertisement to be conservative, we can add a bypass link with value exactly d_{ij} between border nodes B_i and B_j.

5.3 Measuring the Performance of an Aggregation Scheme:

We measure performance of an aggregation scheme K as follows:

Define

$$\mathbf{A = \sum_{1 \leq i,j \leq n, i \neq j} |d_{ij} - d^*_{ij}|}$$

where d_{ij} is the true cost of traversing the peer group through the borders B_i and B_j and d^*_{ij} is what the aggregated representation advertises. For perfect aggregation one would like d_{ij} and

d^*_{ij} to be equal for all i and j ($i \neq j$). To quantify the extent to which this is achieved, we define for each pair B_i, B_j of border nodes,

$$\mathbf{m_{ij} = 1 - \frac{|d_{ij} - d^*_{ij}|c_{ij}}{A}}$$

where c_{ij} are real numbers between 0 and 1, chosen to represent the frequency of transits between borders B_i and B_j. A value of c_{ij} that is closer to 1 signifies more frequent transits between the borders B_i and B_j, making it more important to minimize the distortion in the compact representation for the cost of that transit.

Specifically, we view m_{ij} as the degree to which the advertised cost matches the true cost of the transit between borders B_i and B_j. It is easily verified that for any such transit, m_{ij} equals 1 if and only if the advertised cost d^*_{ij} exactly matches its true value d_{ij}. Furthermore, any deviation of the advertised value from the true value is reflected in the the corresponding m_{ij} to an extent that is linearly proportional to the importance assigned to the transit (as specified by the scalar c_{ij}).

We now define the performance of an aggregation scheme K with respect to a peer group with N border nodes as:

$$\mathbf{Perf(K) = \frac{\sum_{(i,j)} (m_{ij})}{\binom{N}{2}}}.$$

Then $0 \leq \text{Perf}(K) \leq 1$, and higher values of Perf(K) indicate better performance. Also, it is clear that if the aggregation scheme is perfect, that is, $d_{ij} = d^*_{ij}$, then Perf(K) $= 1$.

6 Conclusion

Both approaches presented are viable, according to the criteria presented in section 2:

The proposed schemes transform a peer group structure into a compact form; the final representation requires only $O(n)$ space, where n is the number of border nodes of the peer group.

Within the confines of an $O(n)$ space star, the proposed schemes optimally retain the information about peer group structure that is critical for making good routing decisions. Approach 1 seeks to minimize the average distortion over all transits in a least squares sense; approach 2 builds a star that completely preserves the n most critical transits. The schemes are dual: the first is best suited for peergroups where traffic patterns are unpredictable; the second is suited for peergroups where traffic patterns can be characterized.

The representations are efficiently computable since the most expensive operations (like inverting large matrices) are performed off-line and only need to be done once. In fact, if we we assume, not unreasonably, that the number of border nodes is significantly smaller than the number of nodes in the peer group, say $n \leq |V|^{\frac{1}{3}}$, then the time to aggregate using either of our schemes becomes[4] $O(n|V|\log|V| + n|E|)$. Note further that when a connection request arrives at the peer group, a route computation must be carried out to select a path for the connection. This route

[4]Since then $nV \log V$ dominates n^4

computation itself, using Dijkstra's algorithm, would take $O(|V|\log|V| + |E|)$ time. Thus, if the peer group leader performs re-aggregation every time n new connection requests have entered the peer group, then the computational work required to do the re-aggregation can be amortized against the unavoidable work required to compute the routes for the connections themselves. Our schemes can therefore be executed scalably in a temporal sense: larger peer groups will need to re-aggregate their topology less frequently if switches are not to be excessively burdened by the computational demands of re-aggregation, but the length of this minimum time interval between re-aggregations grows only linearly in n, the number of border nodes.

7 Examples

Let us consider a peer group with 6 border nodes. These six border nodes mean that there are 15 possible paths to route communication through this peer group. Let **D** be the 15×1 column vector of the actual costs of the routing through the 6 pairs of borders. These are obtained from the complete graph on 6 border nodes. These costs are the ones that we wish to advertise. The objective is now to reduce this peer group to a compact form "A Star" representation using both the approaches. Figure 2 shows the complete graph on six border nodes obtained from the original peer group.

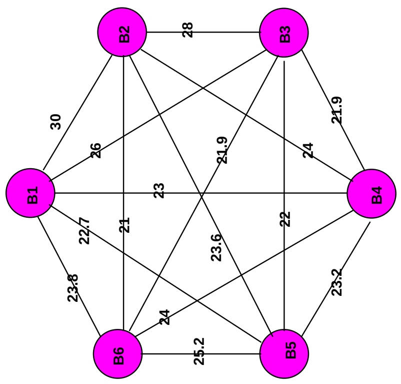

Figure 2: Complete graph on 6 border nodes

The vector of true distances D is given below, and the degree of importance C for each of the transits is as follows:

$$D =_{def} \begin{bmatrix} d_{12} \\ d_{13} \\ d_{14} \\ d_{15} \\ d_{16} \\ d_{23} \\ d_{24} \\ d_{25} \\ d_{26} \\ d_{34} \\ d_{35} \\ d_{36} \\ d_{45} \\ d_{46} \\ d_{56} \end{bmatrix} = \begin{bmatrix} 30 \\ 26 \\ 23 \\ 22.7 \\ 23.8 \\ 28 \\ 24 \\ 23.6 \\ 21 \\ 21.9 \\ 22 \\ 21.9 \\ 23.2 \\ 24 \\ 25.2 \end{bmatrix}, \quad and \quad C =_{def} \begin{bmatrix} c_{12} \\ c_{13} \\ c_{14} \\ c_{15} \\ c_{16} \\ c_{23} \\ c_{24} \\ c_{25} \\ c_{26} \\ c_{34} \\ c_{35} \\ c_{36} \\ c_{45} \\ c_{46} \\ c_{56} \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 0.5 \\ 0.7 \\ 0.5 \\ 0.6 \\ 0.6 \\ 1.0 \\ 0.9 \\ 0.9 \\ 0.9 \end{bmatrix} \tag{1}$$

We compute the aggregated representation of the network using both approaches.

7.1 Solution using approach 1

- S: represents the 15×6 matrix of coefficients for the 15 linear equations in six unknowns x_1, x_2, x_3, x_4, x_5, x_6. These x's will be weights on the star topology of the Logical Group Node representing the given peer group in the parent group.
- D: represents the 15×1 column vector of the actual costs of traversing the peer group through pairs of border nodes. These are obtained from the complete graph on six border nodes. These costs are what we would like to advertise, and the precise values shown in equation (1).
- W: are the values we obtain for the unknown x's using least square approach ,that is, the Approach 1 for solving the system $SX = M$. These are advertised on the edges of the star of the LGN.
- D_1: represents the 15×1 column vector of the values we obtain from the star using approach 1 which will approximate the true values specified in D.

$$S = \begin{bmatrix} 1&1&0&0&0&0\\ 1&0&1&0&0&0\\ 1&0&0&1&0&0\\ 1&0&0&0&1&0\\ 1&0&0&0&0&1\\ 0&1&1&0&0&0\\ 0&1&0&1&0&0\\ 0&1&0&0&1&0\\ 0&1&0&0&0&1\\ 0&0&1&1&0&0\\ 0&0&1&0&1&0\\ 0&0&1&0&0&1\\ 0&0&0&1&1&0\\ 0&0&0&1&0&1\\ 0&0&0&0&1&1 \end{bmatrix}, \quad solving \quad S^T SW = S^T D \quad we \quad get \quad W = \begin{bmatrix} \text{exception}_1\\ \text{exception}_2\\ \text{exception}_3\\ \text{exception}_4\\ \text{exception}_5\\ \text{exception}_6 \end{bmatrix} = \begin{bmatrix} 13.36\\ 13.635\\ 11.935\\ 11.01\\ 11.16\\ 10.96 \end{bmatrix} \tag{2}$$

The left half of figure 3 shows the compact representation of the complete graph in Figure 1. The transit costs implied by this representation are given by D_1.

Now we measure the performance of this approach according to our formulation in section 5.3.

$$D_1 =_{def} \begin{bmatrix} d^*_{12}\\ d^*_{13}\\ d^*_{14}\\ d^*_{15}\\ d^*_{16}\\ d^*_{23}\\ d^*_{24}\\ d^*_{25}\\ d^*_{26}\\ d^*_{34}\\ d^*_{35}\\ d^*_{36}\\ d^*_{45}\\ d^*_{46}\\ d^*_{56} \end{bmatrix} = \begin{bmatrix} 26.99\\ 25.29\\ 24.37\\ 24.52\\ 24.32\\ 25.57\\ 24.64\\ 24.79\\ 24.59\\ 22.94\\ 23.09\\ 22.89\\ 22.17\\ 21.97\\ 22.12 \end{bmatrix}, \quad and \quad P_1 =_{def} \begin{bmatrix} m_{12}\\ m_{13}\\ m_{14}\\ m_{15}\\ m_{16}\\ m_{23}\\ m_{24}\\ m_{25}\\ m_{26}\\ m_{34}\\ m_{35}\\ m_{36}\\ m_{45}\\ m_{46}\\ m_{56} \end{bmatrix} = \begin{bmatrix} 0.87\\ 0.97\\ 0.94\\ 0.92\\ 0.97\\ 0.90\\ 0.98\\ 0.96\\ 0.92\\ 0.97\\ 0.97\\ 0.95\\ 0.96\\ 0.92\\ 0.88 \end{bmatrix} \tag{3}$$

$A_1 = \sum |d_{ij} - d^*_{ij}| = 24.56$, and P_1 shows the degree of match between d_{ij} and d^*_{ij} for all pairs i, j $(i \neq j)$.

Now assuming a degree of importance for each transit as specified in the vector C, and using the vector P_1, we compute that performance of this approach is 0.94 according to our formulation in section 5.3.

7.2 Solution using approach 2

- R: represents the 15×15 matrix of coefficients for the 15 linear equations in fifteen unknowns x_1, x_2, x_3, x_4, x_5, x_6, a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9. These x's will be weights on the star topology of the Logical Group Node representing the given peer group in the parent group. The values of the a's that we obtained will be used to compute the values for the k_i's for $i = 2, \cdots, n$, that will be assigned to the nucleus of the LGN.

$$R = \begin{bmatrix} 1&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&0&1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&1&0&0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&1&0&1&0&0&0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&1&0&0&0&0&0&1&0&0&0 \\ 0&0&0&1&1&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&1&1&0&0&0&0&0&0&0&0&1 \end{bmatrix} \qquad (4)$$

- D: represents the 15×1 column vector of the actual costs of traversing the peer group through pairs of border nodes. These are obtained from the complete graph on six border nodes. These costs are what we would like to advertise, and the precise values shown in equation (1).

- W_2: are the values we obtain on the edges of the star graph .

- K: is the vector that is assigned to the nucleus of the LGN .

- D_2: represents the 15×1 column vector of the values we obtain from the star as well as the vector at the nucleus using approach 2 which will approximate the true values given in M.

$$W_2 = \begin{bmatrix} \text{exception}_1 \\ \text{exception}_2 \\ \text{exception}_3 \\ \text{exception}_4 \\ \text{exception}_5 \\ \text{exception}_6 \end{bmatrix} = \begin{bmatrix} 14 \\ 16 \\ 12 \\ 9 \\ 8.7 \\ 9.8 \end{bmatrix}, \quad \textit{and} \quad K = \begin{bmatrix} \text{error k}_1 \\ \text{error k}_2 \\ \text{error k}_3 \\ \text{error k}_4 \\ \text{error k}_5 \\ \text{error k}_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.38 \\ .46 \\ 2.52 \\ 2.92 \\ 3.36 \end{bmatrix} \qquad (5)$$

The right half of figure 3 shows the compact representation of the complete graph in 1 using approach 2. Now we measure the performance of this approach according to our formulation in section 5.3.

$A_2 = \sum |d_{ij} - d^*_{ij}|$ =26.6 The transit costs implied by this representation are:

$$P_2 = \begin{bmatrix} m_{12} \\ m_{13} \\ m_{14} \\ m_{15} \\ m_{16} \\ m_{23} \\ m_{24} \\ m_{25} \\ m_{26} \\ m_{34} \\ m_{35} \\ m_{36} \\ m_{45} \\ m_{46} \\ m_{56} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0.98 \\ 0.97 \\ 0.90 \\ 0.97 \\ 0.97 \\ 0.99 \\ 0.81 \\ 0.82 \\ 0.76 \end{bmatrix}, \quad \textit{and} \quad D_2 = \begin{bmatrix} d^*_{12} \\ d^*_{13} \\ d^*_{14} \\ d^*_{15} \\ d^*_{16} \\ d^*_{23} \\ d^*_{24} \\ d^*_{25} \\ d^*_{26} \\ d^*_{34} \\ d^*_{35} \\ d^*_{36} \\ d^*_{45} \\ d^*_{46} \\ d^*_{56} \end{bmatrix} = \begin{bmatrix} 30 \\ 26 \\ 23 \\ 22.7 \\ 23.8 \\ 28 \\ 25 \\ 24.7 \\ 25.8 \\ 21 \\ 20.7 \\ 21.8 \\ 17.7 \\ 18.8 \\ 18.5 \end{bmatrix} \qquad (6)$$

Here P_2 shows the degree of match between d_{ij} and d^*_{ij} for all pairs i, j $(i \neq j)$.

Now assuming a degree of importance for each transit as specified in the vector C, and using the vector P_2, we compute that performance of this approach is 0.95 according to our formulation in section 5.3.

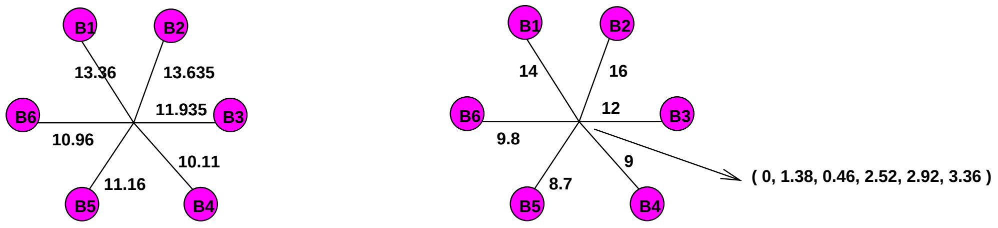

Figure 3: Compact representation using approaches 1 (left) and 2 (right)

8 References

[1] ATM Forum, *Private Network-Network Interface Specification, Version 1.0*, Sept.1996.

[2] W.C.Lee, Spanning tree method for link state aggregation in large communication networks, *In Proceedings of IEEE INFOCOM 95*, volume 1, pages 297-302, 1995.

[3] W.C.Lee, Topology aggregation for hierarchical routing in ATM networks. *Computer Communications Review*, pages 82-92, 1995

[4] Fred Buckley and Frank Harary, Distance in graphs. *Addisio Wesley*, 1990

[5] Steven J.Leon, Linear algebra with applications. *Macmillan College Publishing Company*, 1994