適於多域網路的路由演算法
A Routing Algorithm for Multi-Domain Networks*

陳任凱，陳金蓮，何聿昇
Jen-Kai Chen, Jean-Lien C. Wu, and Yue-Sheng Ho

國立台灣科技大學電子工程系
Department of Electronic Engineering
National Taiwan University of Science and Technology

Abstract

此篇論文提出一適於階層式網路的分散式路由演算法，我們採用 K-T 條件得到封包延遲做最佳化之必要條件，再導出充分條件。而每個路由器與其鄰近節點互換訊息，依此將流量往最短路徑調整。與 RIP 比較之模擬結果顯示，我們的分散式路由演算法在平面式或階層式網路皆比 RIP 好。

A distributed routing algorithm is proposed in this paper for hierarchical networks. We use the K-T conditions to get the necessary conditions and derive sufficient conditions for optimization of packet queuing delay. Every router exchanges routing information with its neighbors and adjusts its traffic from non-shortest paths to the shortest paths for each destination. We also provide the simulation results and comparison with the routing information protocol (RIP). It shows that the proposed routing algorithm is better than RIP for flat networks and hierarchical networks.

1 Introduction

In a flat network, every router needs to keep the knowledge of the whole network for optimal routing. It includes link capacity, traffic load per O-D (Origin-Destination) pair, etc. As the network grows, the non-linearly increased routing information results in the following drawbacks:

- The routing throughput will degrade due to costly search in a large routing information database that also makes routers expensive.
- A large flat network is difficult to manage and of less reliability.

The above drawbacks can be refined in a hierarchical network in which there are only a few visible neighboring nodes for each node. First, we illustrate the multi-domain concept proposed by Alaettinoglu [3]. A 28-node flat network and its corresponding hierarchical network are shown in Figure 1 and 2. A domain is a collection of physical nodes. A superdomain is a collection of logical nodes. Both are also called logical nodes. The visible nodes of d22 are d23 and d24, shown in Figure 3, because they are all within the same domain *H*. The other visible logical nodes are *I*, *D*, and *G* because *I* is the *H*'s sibling within the superdomain *J* whose neighbors are superdomains *D* and *G*. Therefore, the routing table size of d22 is 5 in such a hierarchical network, but 27 in a flat network. Every node or logical node has a unique address, such as J.H.d22 for the physical node d22, and J.H for its upper-level domain. Only the siblings and the ancestors' siblings are visible to each node.

* This research was supported by CCL, ITRI, ROC under grant G4-86023-j.

2 Optimal Routing in a Multi-Link Connected Network

Our algorithm is based on Gallager's optimal routing algorithm [2] which assumes of no more than one link is between any two nodes. Generally, it is very possible that there are multiple links between two physical/logical nodes, as shown in Figure 3. Therefore, we extend Gallager's algorithm to general plat networks before applying to multi-domain networks.

In Table 1, the $t_i(j)$ includes $r_i(j)$ and the bypass traffic destined to d_j, i.e.

$$t_i(j) = r_i(j) + \sum_{m=1}^{n}\sum_{l=1}^{L_{mi}} t_m(j)\phi_{mil}(j) \quad \ldots.(1)$$

By definition, we have

$$f_{ikl} = \sum_{j=1}^{n} t_i(j)\phi_{ikl}(j) \quad \ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Assume the cost function of link (i,k,l), denoted by $D(f_{ikl})$, is convex and strictly increasing. The optimization problem for routing in a multi-link

connected network can be defined as follows:

$$\text{minimize } D_T = \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{L_{ik}} D(f_{ikl}) \quad \text{.....(3)}$$

$$\text{such as } \sum_{k=1}^{n}\sum_{l=1}^{L_{ik}} \phi_{ikl}(j) = 1 \quad \text{..... (4)}$$

$$\phi_{ikl}(j) \geq 0 \quad \text{..... (5)}$$

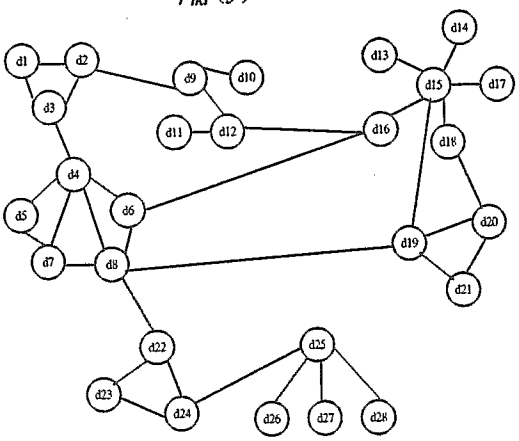


Figure 1. An example of a network with 28 nodes

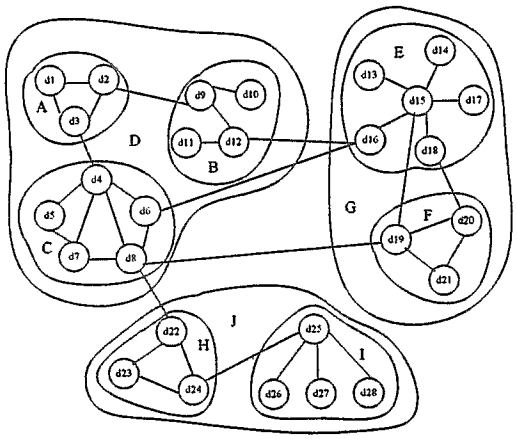


Figure 2. An example of superdomain hirerachy

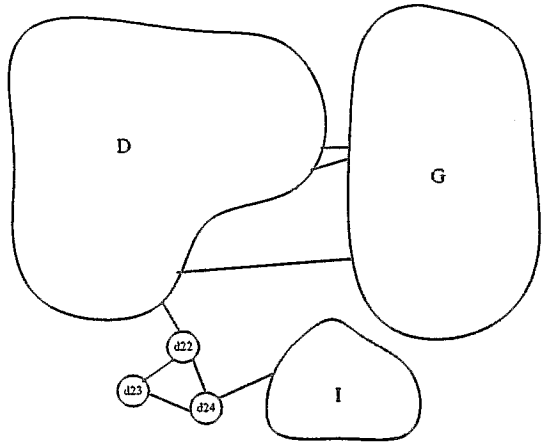


Figure 3. View from a router of d22

d_i	the node with identifier i, $1 \leq i \leq n$
L_{ik}	the number of links et. d_i and d_k
(i, k, l)	the l-th link bet. d_i and d_k
$r_i(j)$	traffic entering the network at d_i and destined for d_j
$t_i(j)$	total traffic at d_i destined for d_j
$\phi_{ikl}(j)$	a fraction of the total traffic on the l-th link bet. d_i and d_k, that is destined for d_j
f_{ikl}	traffic on the l-th link between d_i and d_k

Table 1. Defined Symbols

To minimize D_T, its first partial derivatives with respect to r and ϕ are calculated first:

$$\frac{\partial D_T}{\partial r_i(j)} = \sum_{k=1}^{n}\sum_{l=1}^{L_{ik}} \phi_{ikl}(j)[D'(f_{ikl}) + \frac{\partial D_T}{\partial r_k(j)}] \quad \text{...(6)}$$

$$\frac{\partial D_T}{\partial \phi_{ikl}(j)} = t_i(j)[D'_{ikl}(f_{ikl}) + \frac{\partial D_T}{\partial r_k(j)}] \quad \text{... (7)}$$

By applying the well known K-T conditions to the objective function and constraints, we can get the necessary condition

$$\frac{\partial D_T(\phi^*)}{\partial \phi_{ikl}(j)} \begin{cases} = \alpha_{ij}, & \phi^*_{ikl}(j) > 0 \\ \geq \alpha_{ij}, & \phi^*_{ikl}(j) = 0 \end{cases} \quad \text{..... (8)}$$

By substituting with eq. (6) and (7), we get

$$D'(f_{ikl}) + \frac{\partial D_T}{\partial r_k(j)} \geq \frac{\partial D_T}{\partial r_i(j)} \quad \text{.....(9)}$$

for all $i \neq j$, $(i,k,l) \in L$ (set of links).

We proved that equation (9) is also the sufficient condition besides necessary condition.

3 A Multi-Domain Routing Algorithm

We propose a heuristic aggregation method as follows to simplify the muti-domain routing problem as a flat-network routing problem.

- Every node exchanges its information with its siblings
- Every domain leader aggregates all of the information in its domain to exchange with its sibling leaders.
- Every domain leader sends the aggregated information obtained from its sibling leaders to its children (logical) nodes.

The routing algorithm has two parts: calculation of $\partial D_T / \partial r_i(j)$ and routing variables $\phi_{ikl}(j)$. The following steps are used to calculate $\partial D_T / \partial r_i(j)$:

- For each destination node j, each node i calculates $\partial D_T / \partial r_i(j)$ according to eq. (6) when it receives

the traffic information from its all immediate successors. The results are flooded.

- Each node k floods $\partial D_T/\partial r_k(k)=0$.
- If node i receives two different $\partial D_T/\partial r_k(j)$, eliminate the larger one.

Note that his procedure is free of deadlocks if and only if ϕ is loop free.

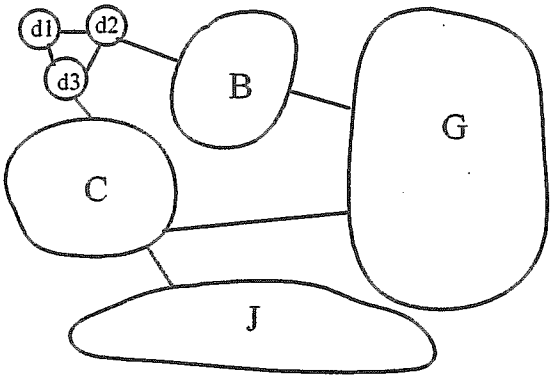


Figure 4. View of d1, d2, d3

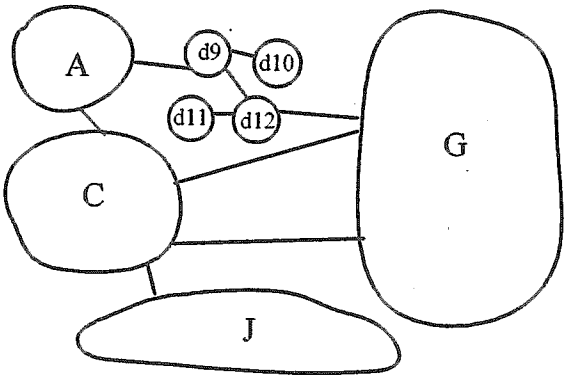


Figure 5. View of d9, d10, d11, d12

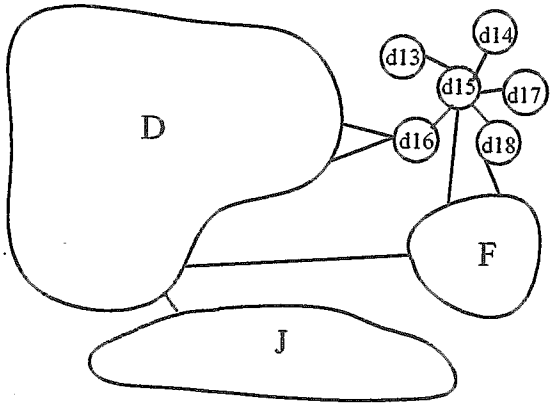


Figure 6. The view of d13, d14, ..., d18

In Figure 4, after having received $\partial D_T/\partial r_{d1}(d1)=0$ from d1, d2 and d3 can calculate $\partial D_T/\partial r_{d2}(d1)$ and $\partial D_T/\partial r_{d3}(d1)$, respectively. The border node d2 of domain A floods $\partial D_T/\partial r_{d2}(d2)=0$ to its neighbor B.d9 as $\partial D_T/\partial r_A(A)=0$ in aggregation view, see Figure 5. Then $\partial D_T/\partial r_{d9}(A)$ is calculated by d9. Similarly, when d16 receives information from superdomain D as $\partial D_T/\partial r_D(D)=0$, the nodes in E can calculate their $\partial D_T/\partial r_i(D)$. In Figure 7, d20 and d19 have different values about $\partial D_T/\partial r_E(D)=0$. It implies there are two different path lengths from E to D in the viewing points of d19 and d20.

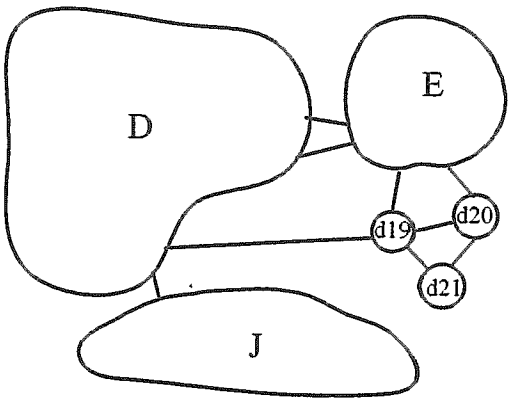


Figure 7. View of d19, d20 and d21

Define the length of link (i,k,l) as $D'(f_{ikl}(\phi,\mathbf{r}))$. Let $S_i(\phi,\mathbf{r})(j)$ be the length of the shortest path from d_i to d_j under the input traffic $\mathbf{r}$ and routing variable ϕ. A sufficient condition for optimization of ϕ is

$$\frac{\partial D_T(\phi,\mathbf{r})}{\partial r_i(j)} = S_i(\phi,\mathbf{r})(j), \text{ for all } i,j.$$

The advantage of our protocol is to include the intra-domain path length information in calculating $\partial D_T/\partial r_i(j)$.

If there exists a link such that $\phi_{ikl}(j)>0$, d_i is the *immediate predecessor* of d_k, and d_k is the *immediate successor* of d_i with respect to d_j. A routing variable set ϕ is *loop-free* if there don't exist d_i and d_k ($i \neq k$) such that d_i is immediate successor and predecessor of k.

Our algorithm is based on the sufficient condition, described as follows:

The set $\mathbb{B}(i,j,\phi^n)$, referred to be the set of *blocked nodes* at i for j and ϕ^n, is the set of all k such that $\phi_{ikl}(j)=0$ for all $l \in L_{ik}$, and either

$$\frac{\partial D_T(\phi^n,r)}{\partial r_i(j)} \le \frac{\partial D_T(\phi^n,r)}{\partial r_k(j)}$$

or there exists a link (a,b,c) such that a is downstream of k, $\phi^n{}_{abc}(j) > 0$, and

$$\frac{\partial D_T}{\partial r_a(j)} \le \frac{\partial D_T}{\partial r_b(j)}.$$

Such a link (a,b,c) is referred to be an *improper link.*

For each iteration n, if $k \in \mathrm{B}(i,j,\phi^n)$, then $\phi^{n+1}{}_{ikl}(j) = 0$, $\Delta^n{}_{ikl}(j) = 0$, for all $l \in L_{ik}$. If $k \notin \mathrm{B}(i,j,\phi^n)$, we define

$$a_{ikl}(j) = D'_{ikl}(f_{ikl}) + \frac{\partial D_T}{\partial r_k(j)} - \min_{m,l;l\in L_{im};m\notin B(i,j,\phi^n)} [D'_{iml}(f_{iml}) + \frac{\partial D_T}{\partial r_m(j)}] \ldots (10)$$

$$\Delta^n{}_{ikl}(j) = \min\{\phi_{ikl}(j), \frac{\eta a_{ikl}}{[t_i(j) * D''(f_{ikl}(j))]}\} \ldots (11)$$

Let $k_{\min}(i,j)$ be the value of m that achieves the minimization in eq. (10). Then

$$\phi^{n+1}_{ikl}(j) = \begin{cases} \phi^n_{ikl}(j) - \Delta^n_{ikl}(j), & k \ne k_{\min}(i,j) \\ \phi^n_{ikl}(j) + \sum_{k\ne k_{\min}(i,j)} \Delta^n_{ikl}(j), & k = k_{\min}(i,j) \end{cases} \ldots (12)$$

Proposition: Let ϕ^k represents the k-th iteration result of our algorithm. If ϕ^k is loop free, then ϕ^{k+1} is also loop free.

Proof: Suppose that ϕ^{k+1} is not loop free, thus there exists at least one directed loop along which $\phi^{k+1} > 0$. Walking along with the direction of the loop, we can find at least a pair of nodes, say m and n, such that

1. $\frac{\partial D_T(\phi^k,r)}{\partial r_m} \le \frac{\partial D_T(\phi^k,r)}{\partial r_n}$, and
2. There exists $l \in \mathrm{L}(m, n)$ such that $\phi^{k+1}_{mnl} > 0$.

From the above conditions, if $\phi^k_{mnp} = 0, \forall p \in \mathrm{L}(m, n)$ by condition 1 and according to the definition of blocking set, we get that n is a member of m's blocking set. By the property of the proposed algorithm, we know $\Delta^k{}_{mnl} = 0$ for all $l \in \mathrm{L}(m, n)$, i.e. $\phi^{k+1}_{mnl} = 0$ for all $l \in \mathrm{L}(m,n)$. It contradicts with condition 2. So, the link (m, n, p) is an improper link because $\phi^k_{mnp} > 0$ and $\partial D_T(\phi^k,r)/\partial r_m \le \partial D_T(\phi^k,r)/\partial r_n$). So, walking along upstream of m, there must exists a pair of nodes a and b, where b is the upstream of m for ϕ^k and $b \in \mathrm{B}(a,\phi^n)$ according to the definition of blocking set, such that $\phi^k{}_{abl} = 0$ for all $l \in \mathrm{L}(a, b)$. It contradicts with the hypothesis. So ϕ^{k+1} must be loop free.

4 Simulation Results and Discussions

We use the flat network and hierarchical network shown in Figures 1 and 2 for simulation with the following simulations :

- The propagation delay is negligible.
- The packet length, input traffic holding time, the input traffic interarrival time to node are all exponentially distributed.
- The maximum and minimum packet length are 65535 and 20 bytes, respectively.
- The output buffer policy is used with the infinite queue.
- The capacity of each link is 10Mbps.
- Mean transmission holding time is 20s.

Figure 8 shows the simulation result with mean traffic interarrival time to node being 20 seconds. Here we use the expected packet delay time as the cost function, that is,

$$D(f_{ikl}) = \frac{f_{ikl}}{C_{ikl} - f_{ikl}}.$$

This is the result of M/M/1 queue. We can see that the mean packet delay time is the least when using our routing algorithm in flat networks, while it is the largest when using RIP. As shown in Figures 1 and 3, domain d22 can sees 28 nodes in the flat network, but only 6 nodes in the hierarchical network. Therefore the routing table size in flat network/routing table size in hierarchical network is 27/5=5.4. Notice that larger routing table means smaller routing computation.

Consider an extreme condition, suppose that an n-node network is structured into a k-level multi-domain network, every g nodes are grouped into a group, $n = g^k$. The number of visible nodes in the lowest level nodes is g + (g-1)(k-1). Therefore, if the routing speed of an algorithm for an n-node network is $O(1/n^m)$, then the ratio of routing speed of the two types of networks is

$$\left([g + (g-1)(k-1)] / g^k\right)^m.$$

In general, the time complexity of a routing algorithm is $O(n^3)$. The size of the routing table is O(n), therefore the ratio of the required storage is

$$\frac{\text{flat net.}}{\text{hierarchical net.}} = \frac{g^k}{g + (g-1)(k-1)}.$$

Dimitrijevic proposed a centralized multi-

domain routing algorithm [1], compared to our distributed algorithm. Dimitrijevic's method takes a lot of efforts for an inconsistent flow problem described as follows, see Figure 9. Suppose that node 1 wants to transmit a 10Mb flow to node 8, and node 3 wants to transmit a 5Mb flow to node 5. To minimize the cost function in level 2, the routing parameters are as shown in Figure 2. But in order to avoid fully utilizing the link (3,5), some traffic has to be shifted to the link (3,4). Therefore, it makes the traffic flow between nodes B and C inconsistent.

Dimitrijevic's routing algorithm is done by cooperations of siblings. The proposed algorithm is done by ancestors and children so that the information in other domains is hidden. The inconsistent flow problem is mainly resulted by that the intranet flow in a domain is not known in other domains. Because our path length $\partial D_T / \partial r_i(j)$ includes the intranet path length, inconsistency will not occur.

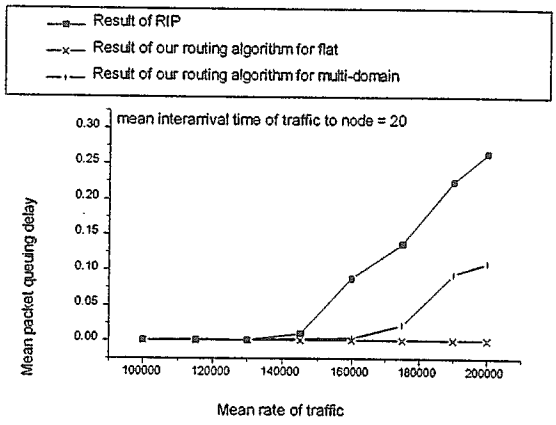


Figure 8. Comparison of Simulation Results

5 Conclusion

In this paper, we applied the Gallager's method to derive the optimal condition for minimum delay routing in a distributed multi-link connected network. For multi-domain routing, a simple flow aggregate method is used to reduce the multi-domain problem into a flat network problem. Then, the Alaettinoglu's distributed routing algorithm can be applied on such a multi-link flat network, and we proved that it would not incur any loop at operation. The simulation results showed that this multi-domain algorithm performs more well than the RIP algorithm. Although it has worse performance than applying to the flat network, it requires less space for the routing table and less computation time for calculating routing parameters. The another advantage is that it has not the inconsistence flow problem versus the Dimitrijevic's multi-domain routing algorithm. It says that our mulyi-domain routing algorithm is more simple for implementation.

References

[1] D. Dimitrijevic, B. Maglaris and R. R. Boorstyn, "Routing in Multidomain Networks", IEEE/ ACM transactions on networking, vol. 2, no. 3, pp.252-262, June 1994.

[2] R. Gallager, "Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Trans. on Comm., Vol. COM-25, No 1, pp.73-84, Jan. 1977.

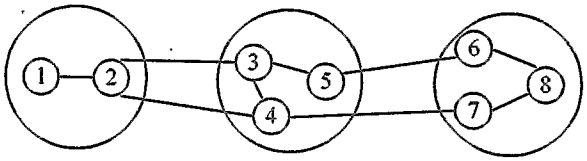
[3] C. Alaettinoglu, A. U. Shankar "An Approach to Hierarchical Inter-Domain Routing with On-Demand ToS and Policy Resolution", IEEE Proceedings 1993 International Conference on Network Protocols, pp.72-9, Oct. 1993.

[4] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison Wesley, May 1989.

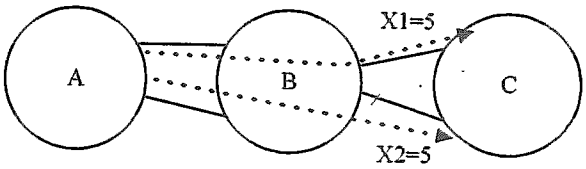
[5] Y. Rekhter, P. Traina, "Inter-Domain Routing Protocol, Version 2", Internet draft, June 1996.

[6] Y. Rekhter, and T. Li. "A Border Gateway Protocol 4", Request for Comment RFC -1771, Network Information Center, March 1995.

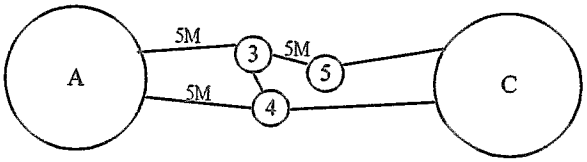
[7] M. Hedrick "Routing Information Protocol", Request for Comment RFC-1508, Network Information Center.



(a) An example hierarchy network



(b) The routing result of level-2 nodes



(c) The routing result of level-1 nodes in B

Figure 9 Ilustration of inconsistent flow problem