

# Distributed Connection Management Architecture for Optimal VPC Provisioning on Hierarchical ATM Transport Network

Won-Kyu Hong , Dong-Sik Yun

Telecommunications Network Lab., Korea Telecom  
463-1, Junmin-dong, Yusung-gu, Taejeon, Korea  
{wkhong, dsyun}@kt.co.kr

**Abstract.** It was impossible to find a globally optimal route on hierarchical transport network because of the successive subnetwork partitioning and network topology abstraction. So, this paper proposes a hierarchical routing model that provides a globally optimal route on hierarchical ATM transport network. In addition, this paper proposes distributed connection management architecture adopting the proposed routing model. The hierarchical routing model and the distributed connection management architecture are specified in terms of information, computational and engineering viewpoints of RM-ODP. We implement the proposed routing model and the distributed connection management architecture, and show that the proposed connection management architecture always provides a globally optimal route with admissible performance by empirical performance analysis.

## 1 Introduction

ITU-T G.805 defines the generic transport network architecture from the perspective of the information transfer capability and defines especially the functional and structural architecture of transport networks independent of networking technology [1]. G.805 layering concept can be applied not only to transmission networks but also to switching networks. For example, there are ATM VC layer network, ATM VP layer network and SDH transmission layer network. A Layer Network (LNN) can be partitioned into Subnetworks (SNW) connected by Topological Links (TL). Each SNW can be successively decomposed into smaller SNW that is equivalent to a single switching element.

The deployment of hierarchical transport network based on the concepts of ITU-T G.805 layering and partitioning provides some strength in telecommunications network management. For one example of ATM VP connection service provisioning, SNW partitioning of VP layer network can narrow down the management scope of connection control, restoration, fault localization, performance monitoring, and so on.

However, it gives a disadvantage that hardly finds a globally optimal path. In the partitioned subnetwork architecture, every subnetwork can view only its own subordinate subnetwork partitioning topology that is used for routing.

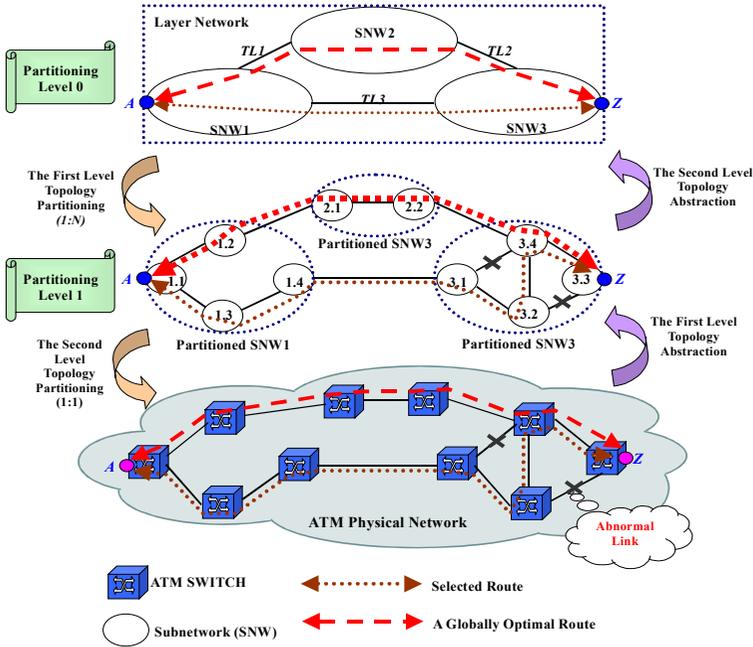


Fig. 1. Layer Network Partitioning

Fig. 1 describes a VP layer network partitioning structure and a route selection scheme based on ITU-T G.805 [1] and TINA-C NRM [2]. A layer network and each subnetwork can select only the locally optimal path with their topology information that is composed of sets of subnetworks and topological links.

At the partitioning level 0, optimal routing path from A to Z is the direct link of TL3 between SNW1 and SNW3. Subsequently the optimal path selected by SNW1 traverses SNW1.1, SNW1.3 and SNW1.4. The optimal path selected by SNW3 traverses SNW3.1, SNW3.2, SNW3.4, and SNW3.3. Therefore the selected route by SNW1 and SNW3 at the partitioning level 1 is not a globally optimal path from the perspective of the physical network. According to this hierarchical routing model, the optimal path selected at upper partitioning level is different from the optimal path selected at the lower partitioning level.

To solve this problem, this paper proposes a new hierarchical routing model that is composed of several routing information objects: *Routing Information Base (RIB)*, *Routing Metric (RM)*, *Routing Table (RT)*, and *Internal Cost and Reachability Table (ICRT)*. This paper also proposes distributed connection management architecture to

provide a globally optimal route on hierarchical transport network and to adopt the generic telecommunications networking system functions: addressing, routing and switching.

This paper specifies the proposed routing model and distributed connection management architecture using the RM-ODP information, computational and engineering viewpoint [12]. We implement the proposed hierarchical routing model and distributed connection management model. Our empirical performance analysis shows that the performance of the proposed distributed connection management model is nearly same as traditional TINA-C hierarchical connection management model in spite of providing the globally optimal route.

## 2 Hierarchical Routing Information Model

To describe the generic transport network topology, ITU-T G.805 [1] and TINA-C [2] defines several information objects that are mainly composed of Subnetwork (SNW), Topological Link (TL) and Link Termination Point (LTP). Subnetwork represents the node or the administrative and routing scope. Topological link represents the topological relationships between two subnetworks and the potential connectivity between the subnetworks. Link termination point represents an ending point of a topological link. According to the partitioning concept, a layer network is decomposed into subnetworks. Each subnetwork may be decomposed into smaller subnetworks until the subnetwork is equivalent to a single network element (switch or cross-connect). The subnetwork that is corresponding to a single network element is called as Element Management Layer Subnetwork (emlSNW). The subnetwork that is decomposed into smaller subnetwork is named as Network Management Layer Subnetwork (nmlSNW).

With these information objects, we can not provide a globally optimal route because of successive subnetwork partitioning and topology aggregation as depicted in Fig. 1. To solve this problem, we newly define four information objects: *Routing Information Base (RIB)*, *Internal Cost and Reachability Table (ICRT)*, *Routing Metric (RM)* and *Routing Table (RT)* as depicted in Fig. 2.

*Routing Information Base (RIB)* information object represents the network topology largely based on the weighted graph model. It is defined from the perspective of network topology to construct routing table and is composed of *ribNode* to represent SNW, *ribPort* to represent LTP and *ribLink* to represent TL as depicted in Fig. 2. Every *ribLink* has a state measured by the QoS metric of concern. Every *ribNode* also has a state defined by the *Internal Cost and Reachability (ICR)* metric that takes a key role to provide a globally optimal route.

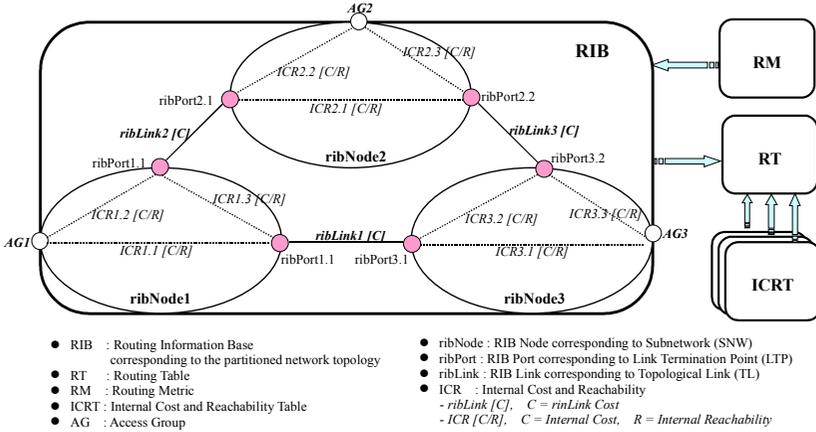


Fig. 2. Proposed Hierarchical Routing Model

We adopt the *Access Group* (AG) information object of ITU-T G.805 that is defined as a group of co-located "trail termination" functions that are connected to the same „subnetwork“ or „link“[1]. We extend it to minimize routing table size. There is one AG in the ribNode that can be termination node to calculate routing path by routing algorithm. It physically represents the route path termination and logically represents all the access ribPorts in a ribNode. Because the source and destination to calculate route path are AGs not ribPorts, the routing table size can be remarkably reduced. AG can be comparable with the Internet routing mechanism that most hosts and routers use a default route in order to avoid routing entries for every possible Internet destination

*Routing Metric* (RM) information object maintains the necessary information to assign cost to ribLinks and to keep optimal route selection criteria. In order to assign ribLink cost we define three states: bandwidth, delay, and reliability. It also maintains the administrative weight (hop count) that is used to trim the routes to exceed the assigned hop count from the perspective of network administration. This is very useful to moderate the size of computed routing table and subordinate route information propagation complexity. Let's assume that  $B_d$  is the degree of residual bandwidth,  $R_d$  is the degree of reliability,  $H_c$  is the cost of hop count,  $D_d$  is the degree of transit delay and  $W_{lnw}$  is administrative weight applied to VP LNW. We allocate cost to ribLink ( $L_c$ ) with the following criteria. If a ribLink is in fault, we allocate  $\infty$  to  $L_c$ .

$$L_c = B_d + D_d + R_d + H_c, \quad \text{where } H_c \geq B_d + D_d + R_d. \quad (1)$$

*Routing Table* (RT) information object contains all possible route information at any partitioning level. Most of existing routing algorithms rely on Dijkstra's algorithm to create a shortest-path route to the destination, based on a single quantity or metric. In ATM networks, links and connections are characterized by several metrics. It

makes Dijkstra’s algorithm unsuitable to apply to ATM network. Our routing algorithm adopts the pre-computed path driven approach to meet various kinds of QoS requirements and to provide a globally optimal route. At any partitioning level, if there are  $n$  ribNodes, there are  $n(n-1)/2$  route path entries. For example, at partitioning level 0 of Fig. 1, there are three AGs per ribNode: AG1, AG2, and AG3. We compute all possible routes among all combination of AGs. So there are three route entries: AG1 to AG2, AG1 to AG3, and AG2 to AG3.

Because we compute all possible routes on all combination of AGs in RIB, the complexity of route computation is very high. Fortunately, most of the large scaled backbone networks like VP layer network is connected each node with duplicated full mesh. It is also manifested that the more hop count is increased, the more the end-to-end transit delay is increased. So one should determine the reasonable end-to-end transit delay that is accomplished by the administrative weight ( $W_{lmw}$ ) in Routing Metric (RM) information object.  $W_{lmw}$  represents the maximum hop count of every route. We trim the routes that exceed the  $W_{lmw}$  off in the process of route computation to moderate route computation complexity and to provide the optimal VPC that meets the end-to-end transit delay.

*Internal Cost and Reachability Table (ICRT)* information object contains the summarized subordinate RT information. In line with the network topology of Fig. 1, there are three other RIBs and RTs that are corresponding to the partitioned SNW1, SNW2, and SNW3. The summarized subordinate RTs of SNW1, SNW2, and SNW3 are composed of source/destination AG pair and the cost and reachability of its first priority route. Our routing information model can be represented using the generic model of Rumbaugh OMT diagram [15] like Fig. 3.

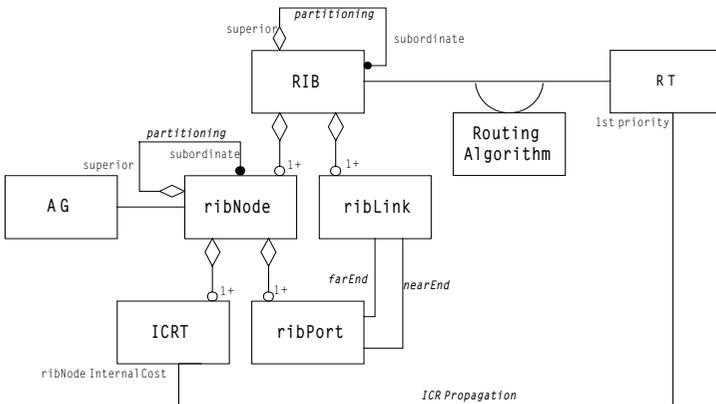


Fig. 3. Routing OMT Diagram

To provide a globally optimal route with the proposed information objects, we do following steps:

1. construct RIB with partitioned network topology information at every partitioning level
2. assign cost to every ribLinks using the Routing Metric (RM)
3. if RIB is composed of ribNodes corresponding to emlSNWs
  - prepare summarized RT information and propagate it to its superior partitioning level
  - otherwise, wait until it receives summarized RT information from all subordinate partitioning level
4. if RIB is composed of ribNodes corresponding to nmlSNWs,
  - construct ICRTs with the propagated summarized RT information from its lower partitioning level
5. when a source and a destination are specified,
  - make relation between ICR in ribNodes and summarized RT information cost of ICRT,
  - calculate total cost of all routes pertaining to the specified source and destination AGs, and
  - assign priority to all routes according to the total cost of each route.
  - The total cost of a  $k$ -hop path  $P$  is

$$C(P) = \sum_{i=1}^k Lc + \sum_{i=1}^{k+1} ICR_{(n)}, \text{ where } n = \text{ribNode that traverse the route } P. \quad (2)$$

If the specified source and destination are respectively  $A$  and  $Z$  in Fig. 1, the RT of partitioning level 0 will be changed to reflect the ICR information of subordinate partitioning level.  $A$  implies the AG-A and  $Z$  implies AG-Z in Fig. 4. So, we adjust cost and priority of routes in AG-A and AG-Z route entry to reflect the ICRT information. Reflecting the ICR information to each ribNodes, the total cost and priority of routes are changed as depicted in Fig. 4.

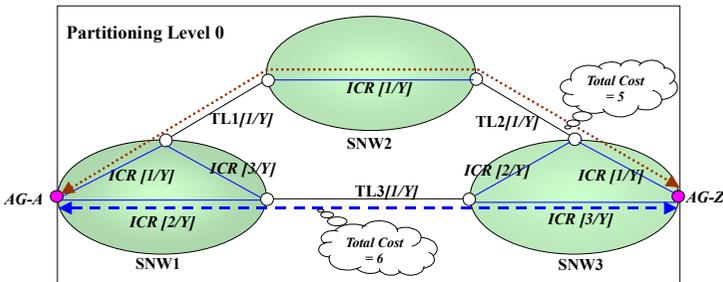


Fig. 4. RIB information to be applied ICR information

As a result of reflecting ICR information, we can select a route that traverses TL1 and TL2 as a globally optimal route to the contrary of Fig. 1. If we do not consider the ICR in the process of route selection, the route along the TL3 will be the optimal route. However, if the ICR is reflected into the process of selection, the route along the TL1 and TL2 will be more optimal route than the route along the TL3.

Network status is dynamically changed due to transient load fluctuation, connection in/out, and links up/down. As a result of above, the ribLink cost is subsequently changed, and so the pre-computed route cost and priority are adjusted. Then, if the pre-computed route of the highest priority is changed, subordinate route information should be propagated to superior partitioning level, else it needs not propagate.

### 3 Distributed Connection Management Computational Model

This paper fully adopts the TINA-C computational models of connection management and configuration management [3]. However, this paper describes the extended computational model in terms of generic networking system functions: addressing, routing, and switching. We define VP connection management architecture that is restricted within EML and NML of the TMN functional layering architecture. Our proposed distributed connection management computational model is like Fig. 5.

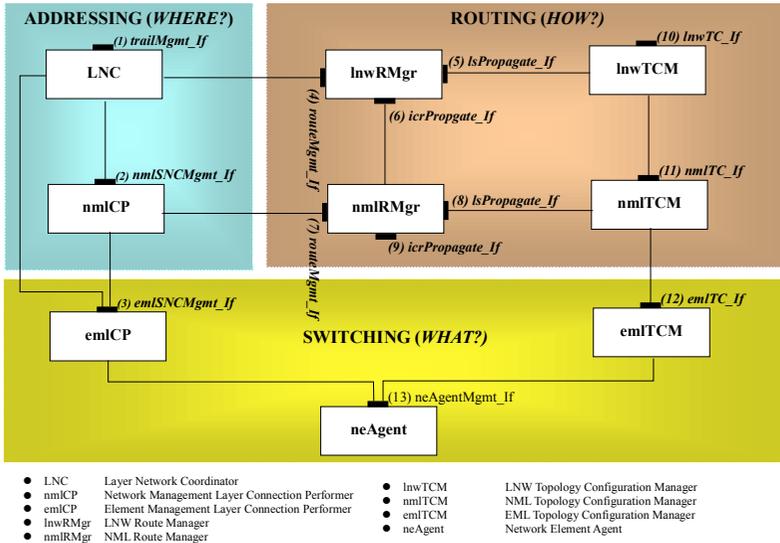


Fig. 5. Distributed Connection Management Computational Model

The *Layer Network Coordinator (LNC)* takes the role of establishment or release of trails within a layer network. We newly allocate addressing function to it. Our addressing function can be broken down two categories: resolving end user address (E.164) to network resource name (LTP) and selecting proper subordinate nmlCPs or emlCPs as a result of routing. There is one LNC for each administrative domain of every layer networks.

The *Connection Performer (CP)* provides interconnections of termination points of a subnetwork. There are two CPs of nmlCP (Network Management Layer CP) and

emlCP (Element Management Layer CP). The nmlCP is allocated to every nmlSNW and selects proper subordinate nmlCPs or emlCPs as a result of routing, as does LNC. On the other hand, the emlCP interacts with neAgent in order to control switch. The emlCP is allocated to every emlSNW.

The *Network Element Agent (neAgent)* takes the roles of mapping between our network transport entity and switch resource entity, issues operation to control switch and receives various kinds of notification from switches. Especially the switching function can be achieved by the mutual interaction between emlCP and neAgent.

The *Layer Network Route Manager (lnwRMgr)* and *Network Management Layer Route Manager (nmlRMgr)* take the role of routing function. They gather network topology information from lnwTCM or nmlTCM in order to compose RIB and make routing table using this RIB. The lnwRMgr is only one, but nmlRMgr can be more than one according to the network partitioning. lnwRMgr is allocated to every LNW and nmlRMgr is assigned to every nmlSNW. lnwRMgr makes ICRTs with the propagated ICR information from subordinate nmlRMgrs. If there are successive subordinate nmlRMgrs, the nmlRMgr also composes ICRTs, as lnwRMgr does. However, if there is no more successive subordinate nmlRMgrs, nmlRMgr does not make ICRTs. It provides not only optimal path at normal connection setup procedure but also alternative route excluding the abnormal ribLinks.

The *Topology Configuration Manager (TCM)* configures network topology resources of LNW, nmlSNW, emlSNW, TL and LTPs. It provides network topology information to Route Manager for RIB and RT construction. Especially it monitors network resource status and gives Route Manager information of changed resource status to affect route selection.

There are important aspects of our connection management computational model. One is subordinate CP selection procedure that LNC can directly control emlCPs without mediation of nmlCPs. The other is concurrent connection management procedure that if either LNC or nmlCP selects more than one subordinate CPs in consequence of routing, it sends SNC manipulation requests to the multiple CPs in current manner. So it improves connection manipulation performance and minimizes the number of managed objects.

Table 1. Computational Object Interfaces

<i>Interfaces</i>	<i>Role of Interface</i>
<i>trailMgmt_if</i>	Create, delete, modify and reroute Trail
<i>nmlSNCMgmt_if</i>	Create, delete, modify and reroute nmlSNC
<i>emlSNCMgmt_if</i>	Create, delete and modify emlSNC
<i>RouteMgmt_if</i>	Provide optimal route and alternative route including CAC
<i>icrPropagate_if</i>	Gather subordinate summarized RT information (ICR)

<i>lsPropagate_If</i>	Receive link State that will be effect the ribLink cost calculation Modify RIB and Re-construct RT only when the network topology is changed
<i>lnwTC_If</i>	LNW resource provisioning (LNW, nmlSNW, LTP, TL)
<i>nmlTC_If</i>	nmlSNW resource provisioning (nmlSNW, emlSNW, LTP, TL)
<i>emlTC_If</i>	emlSNW resource provisioning (emlSNW, LTP)
<i>neAgentMgmt_If</i>	Adaptation between management objects and switch dependent objects

## 4 Engineering Model and Implementation

### 4.1 Engineering Model

We have implemented our VP connection management model with the proposed hierarchical routing model. The network topology and connectivity related information objects are modeled as the *Persistent Engineering Objects (PEO)* to be kept in spite of system failure. It is realized into C++ objects and its information is stored into Oracle Database for persistency. On the other hand, the information objects related routing are realized into the *Transient Engineering Objects (TEO)* to be not kept in case of system failure, instead can be rebuilt based on the persistent engineering objects. All routing information objects are defined as TEOs because it can be reconstructed with the SNW, LTP and TL PEOs in case of route manager system failure.

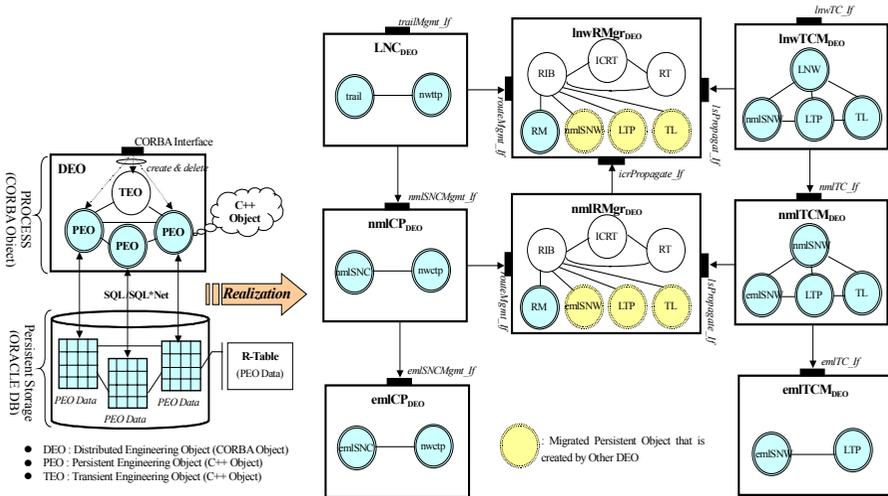


Fig. 6. Engineering Model

All computational objects are described with IDL and realized into the CORBA distributed objects of *Distributed Engineering Object (DEO)* executing on the distrib-

uted platform. It creates, modifies and deletes the necessary PEOs or TEOs on which operations are called by client DEO. After creating a PEO, its information is stored into database and deleted from memory at end of transaction. However, the TEOs for routing is not stored into database but kept only in memory after the end of transaction.

The connection management DEOs of LNC, nmlCP and emlCP create connectivity PEOs and store their information into database at the end of transaction. The network topology configuration management DEOs of lnwTCM, nmlTCM and emlTCM create network topology related PEOs and store their information to database, as connection management DEOs do. The routing management DEOs of lnwRMgr and nmlRMgr create RIB, ICRT and RT TEOs and keep their information only in memory.

The engineering model supports the object migration concepts of CORBA. Though the SNW, LTP, and TL PEOs are created and managed by topology configuration manager DEOs, these can be migrated to Route Manager DEOs when Route Manager does CAC or rebuilds routing related TEOs after system crash.

### 4.2 Implementation and Empirical Performance Analysis

For implementation, we choose IONA Orbix distributed platform for enhancing the distribution transparency. In order to support object persistency, we choose Oracle and store all PEOs into this database. We configure ATM backbone network with the FORE ASX200 switches and connect them with OC-3C links. We implement the neAgent that plays the role of adaptation between our management system and FORE ATM switch using the CORBA/SNMP gateway of JIDM approach.

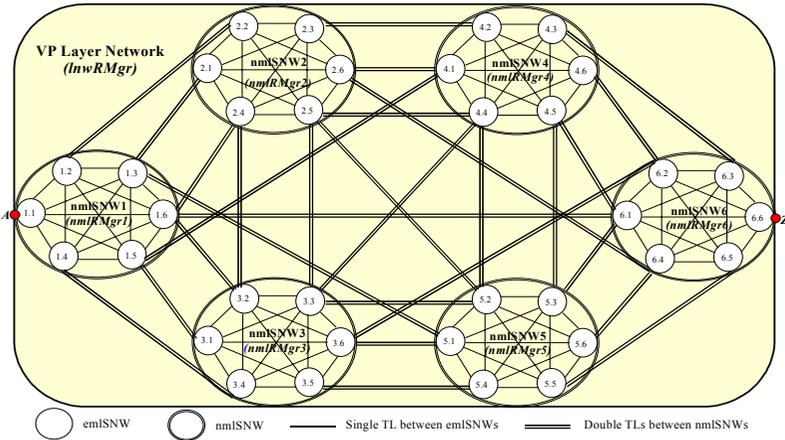


Fig. 7. Implementation Network Topology

The network topology for implementation is depicted in Fig. 7. There are one VP layer network, six nmlSNWs and thirty-six emlSNWs. The LNC, lnwRMgr and

lnwTCM are allocated to VP layer network. The nmlCP, nmlRMgr and nmlTCM are allocated to every nmlSNWs. The emlCP and emlTCM are assigned to every emlSNWs and the neAgent is allocated to every ATM switches. All route and connection management DEOs is implemented with thread for current connection management. We use seven UTLASPARC workstations that every workstation is allocated to each Route Manager. One is for LNC, lnwRMgr and lnwTCM DEOs, others are for nmlCP, nmlRMgr and nmlTCM DEOs.

This paper describes implementation experiences on connection manipulation and route management compared our connection management architecture with traditional TINA-C connection management architecture. We analyze the performance of routing table construction and connection setup of both of one to propagate ICR information and the other to not propagate.

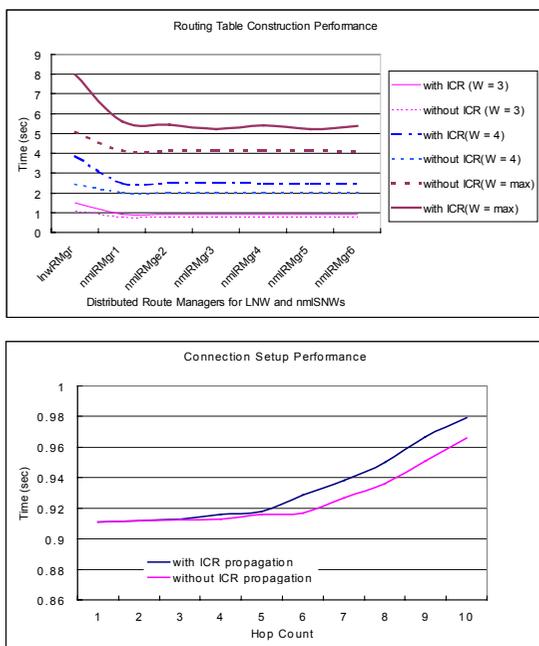


Fig. 8. Route and Connection Management Performance Evaluation

Inferring from Fig. 8, we learn that complexity of routing table construction including ICR metric propagation totally depends on the number of subordinate nmlRMgrs and the administrative weight ( $W_{LNW}$ ). The complexity of routing table construction of all nmlRMgrs that has no any successive subordinate nmlRMgrs is similar without regard to it that they propagate ICR information to superior route manager or not, as they do not create ICRTs. On the other hand, the lnwRMgr shows a prominent performance gap depending on whether lnwRMgr gathers ICR information and constructs its correspondent ICRT or not.

From the perspective of connection setup, there is no prominent performance gap without regard to it that the Route Manager chooses the proposed end-to-end optimal path selection mechanism or not. The slight gap in connection setup time is mainly caused by the route cost calculation to reflect ICR cost of route manager.

## 5 Conclusion

This paper proposed hierarchical routing model that always provides a globally optimal route on the hierarchical transport network architecture. It has been proved that the model provides a globally optimal path using the information objects newly defined. Among them, ICRT is key information object to resolve the problem to hardly find optimal route because of subordinate network topology abstraction.

We also proposed the distributed connection management architecture to integrate addressing, routing and switching. For addressing, we extended the subordinate Connection Performer (CP) selection procedure to minimize the number of connectivity related managed objects and to reduce the connection setup time. For routing, we newly defined two computational objects: `lnwRMgr` and `nmlRMgr` to take a key role to provide a globally optimal route on hierarchical transport network.

We showed that the proposed hierarchical routing model and distributed connection model could always provide the globally optimal route by implementation. In addition, we showed that the performance of the proposed model is nearly same as traditional hierarchical connection management in spite of providing the globally optimal route. The difference of connection setup time was less than 5% by our experience.

This paper focused only on the ATM Virtual Path Connection (VPC) management that provides a globally optimal route on the G.805-based hierarchical transport network architecture. In order to deploy proposed distributed connection management model into upcoming multi-layer and multi-service network, we should evaluate what is the most reasonable convergence point between abstraction and partitioning to maximize network resource utilization and restoration probability without prominent performance degradation.

## 6 References

1. ITU-T Recommendation G.805, „Generic Functional Architecture of Transport Networks,“ November 1995.
2. Network Resource Information Model Specification, Document No. NRM\_v2.2\_97\_10\_31, October 31, 1997.
3. Network Resource Architecture, Document No. NRA\_v3.0\_97\_02\_10, TINA-C, February 1997.

4. ITU-T Recommendation M.3100, „Generic Network Information Model,“ July 1995.
5. ITU-T Recommendation M.3010, „Principles for a Telecommunications Management Network,“ May 1996.
6. ITU-T Recommendation I.311, „B-ISDN General Network Aspects,“ March 1993.
7. ATM Forum, „PNNI Specification 1.0,“ af-pnni-0055.0000, March 1996.
8. Hui Xie and John S. Baras, „Performance Analysis of PNNI Routing in ATM Networks: Hierarchical Reduced Load Approximation,“ February 1996.
9. Tsong-Ho Wu and Noriaki Yoshikai, „ATM Transport and Network Integrity,“ Academic Press, 1997.
10. P. Lauder, R.J. Kummerfeld, and A. Fekete, „Hierarchical Network Routing,“ IEEE Conference on Communications Software, April 1991.
11. IONA Technologies Ltd., „Orbix Programming Guide – Release 2.0,“ November 1995.
12. OMG, „CORBAServices: Common Object Service Specification,“ March 31, 1995.
13. OMG, „The Common Object Request Broker: Architecture and Specification,“ July 1995.
14. ISO/IEC CD 10746-1, „Basic Reference Model of Open Distributed Processing – Part 1: Overview and Guide to Use“, July 1994.
15. James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy and William Lorenson, Object-Oriented Modeling and Design,“ Prentice Hall, 1991.