New Scheme for IP Routing and Traffic Engineering

Girish P. Saraph and Pushpraj Singh
Department of Electrical Engineering
Indian Institute of Technology, Bombay, Mumbai – 400076, India
Email: girishs@ee.iitb.ac.in

***Abstract*—A novel scheme for routing data packets in high-speed communication networks is presented here. The detailed scheme is described in the paper.* Simulations are performed on randomly constructed 25 and 100 node networks, which demonstrate excellent IP throughput and capability to adapt to the dynamic network conditions. The most important benefit of this scheme is to simplify the route look-up tasks and save on resources required for route processing, routing updates, data storage, memory access, and information exchange. The scheme can be used as a stand-alone IP routing protocol or used with the conventional IP, ATM, or MPLS for path selection, QoS support, and traffic engineering. It can quickly adapt to the dynamic load conditions of traffic congestion or link breakage to enable QoS support or priority-based differential services. The scheme could be implemented in software or hardware to give a simple, fast, and cheap solution. The simulations show that the solution is robust and highly scalable for large high-speed networks.**

Index Terms— IP Routing, Look-up Table, MPLS, Protection, QoS, Traffic Engineering.

I. INTRODUCTION

THIS paper presents a novel scheme for routing packets in high speed communication networks. Internet protocol (IP) provides the most flexible, scalable, and robust platform for data communication and is currently the most widespread. However, with the recent explosion in the Internet traffic, IP routers are stretched to the limit to perform at wire-speed when each port operates at the OC-48 or OC-192 rates. The large size of the router look-up table remains a performance bottleneck for high-speed operation and the main reason for the high hardware and software costs in routers. Each router keeps track of the dynamic conditions in the network using routing information exchange and then updating the look-up table. Commonly used routing protocols rely on information about the network conditions to flow from the distant network nodes to any given router either directly (in link-state type protocols) or indirectly (in distance-vector type protocols). This requires excessive information exchange, processing, storage, leading to overhead hits in terms of the available bandwidth, processing speed, hardware costs, memory, and memory I/O. This information is used to build and update the route look-up tables based on dynamic network conditions [1]. Presently, a typical core IP router may have look-up table entries in excess of 100,000 [2], which makes the routing and forwarding tasks of the router rather complex and costly. The look-up task is further complicated due to the transition from IP-v4 to IP-v6 and the aggregation of the IP routing table entries leading to multiple matches, requiring the longest prefix match (LPM) selection [3,4]. The proposed scheme essentially eliminates the large look-up tables in routers and greatly simplifies both routing and forwarding functions. In addition it provides a convenient way to include the link cost function that reflects the dynamic conditions in the network. Although, the proposed scheme can be used as a stand-alone routing algorithm for IP networks, it can also be used with other protocols (ATM, MPLS, or GMPLS) as an additional tool to support traffic engineering, protection, and differential services. It can also be viewed as a central tool for network planning and management.

The paper first describes the routing scheme, including the philosophy behind this new approach, its formulation and the implementation details. Then, it presents the simulation results showing the efficacy of the proposed routing scheme. Then, the paper is concluded by describing the advantages of this scheme and proposing different applications of it.

II. ROUTING SCHEME

A. Basic Philosophy

In the proposed routing scheme the network information is separated into static (or slowly changing) physical network topology and the dynamic network conditions. The physical topology information is made available to each node in the most simplified form by expressing it in terms of the virtual space (VS) configuration. The dynamic conditions, including any traffic congestion or link breakage, *etc.*, are expressed in terms of link cost functions. The packets are routed towards

Manuscript is submitted to HPSR 2003 on December 20, 2002.

*The detailed routing scheme presented in this paper is being submitted for a U.S. patent application, please contact Dr. G. P. Saraph for details.

the destination by using the VS coordinates and by taking into account the link costs. This scheme reduces the routing information exchange, processing, and storage significantly.

As an analogy, in a well-connected roadway network, it is sufficient to head towards the distant destination city without knowing the exact road conditions near the destination. A suitable detour can be found in the vicinity of the problem area having some blockage or congestion. Similarly, in a well-connected mesh-type network, the simplified network topology information and local dynamic link conditions can be sufficient to direct packets and achieve efficient routing. The path directivity envisioned in this representation is enabled by virtual space embedding. The VS configuration is such that a directed VS distance to any destination node indicates the available path options with the least (or low) number of hops through the network.

B. Virtual Space Forces

A major step in the VS based routing is to first embed the network topology information into the VS-based geometric form. The transformation of a planar network with a given topology into the VS representation is achieved by letting the network evolve under the influence of a set of forces. These forces act on each network node to displace it in the multi-dimensional virtual space. The forces are defined as follows:

1) Force on the directly connected nodes: This force, called $\mathbf{F}_{1ij}$, acts like a spring force between the nodes n_i and n_j that are directly connected and tends to make the distance between the directly connected nodes close to unity. When the distance between the two nodes, d_{ij}, is less than one unit, *i.e.* $d_{ij} = |\mathbf{d}_{ij}| < 1$, then $\mathbf{F}_{1ij}$ is a repelling force directed along the distance vector $\mathbf{d}_{ij}$ and acts on both the nodes to push them apart. Whereas, when $d_{ij} = |\mathbf{d}_{ij}| > 1$, then $\mathbf{F}_{1ij}$ is an attracting force. The functionality of its magnitude can be suitably chosen (*e.g.* linear, logarithmic, *etc.*) and optimized for a wide variety of network topologies. The simulation results presented here use the following relationship:

$$F_{1ij} = |\mathbf{F}_{1ij}| = k_1 * \log (d_{ij}) \qquad (1)$$

with $k_1 = 12$. All such forces acting on the node n_i due to all its directly connected neighbors add vectorially, to give the net force $\mathbf{F}_{1i}$ given by,

$$\mathbf{F}_{1i} = \Sigma_j \mathbf{F}_{1ij} \qquad (2)$$

where n_i and n_j are direct neighbors

2) Repulsive force on the nodes not connected directly: This force, called $\mathbf{F}_{2ik}$, acts on the nodes n_i and n_k that are not directly connected to push them apart, along the distance vector $\mathbf{d}_{ik}$. The force magnitude is made dependent on the least number of hops, N_{ik}, between the two nodes. It ensures that the two nodes with large N_{ik} are pushed apart in the VS configuration. The simulation results presented here use the following relationship:

$$F_{2ik} = |\mathbf{F}_{2ik}| = k_2 * (N_{ik} - 1)^{\alpha} / d_{ik} \qquad (3)$$

with $k_2 = 0.025$ and $\alpha = 2$. In order to limit the computation involved, the force F_2 is considered a short-range force and applied only over a limited distance of less than 4 units *i.e.* $d_{ik} < 4$. The maximum value of N_{ik} is also limited to 10. The different repulsive forces acting on the node n_i due to all the applicable nodes n_k are added vectorially, to give the net force $\mathbf{F}_{2i}$ given by,

$$\mathbf{F}_{2i} = \Sigma_k \mathbf{F}_{2ik} \qquad (4)$$

For all n_k nodes that are not directly connected and $d_{ik} < 4$.

3) Random kick force: This is a deterministic, pseudo-random force $\mathbf{F}_3$ used to kick individual nodes in the multi-dimensional VS configuration. It facilitates evolution of the VS configuration solution to reach the global energy minima instead of the local ones. It also seeds the initial evolution in the multi-dimensional space with the dimensionality N_d. The random kick $\mathbf{F}_{3i}$ given to the node n_i has an arbitrary amplitude and direction, given by a pseudo-random number (R_n) sequence in the range of [0,1]. The q-th component of the kick is given by the q-th R_n as,

$$F_{3iq} = |\mathbf{F}_{3i} \cdot \mathbf{q}| = k_3 * R_{nq} \qquad (5)$$

with $k_3 = 0.1$ and $q = 1, 2, \dots, N_d$.

C. Evolution of Virtual Space Configuration

The total force $\mathbf{F}_{ti}$ acting on the node n_i is a vector sum of the three forces listed above and is given by, $\mathbf{F}_{ti} = \mathbf{F}_{1i} + \mathbf{F}_{2i} + \mathbf{F}_{3i}$. For every iteration, each node, n_i, moves under the action of the total force, $\mathbf{F}_{ti}$, by a distance, $\mathbf{\Delta}_i$, given by,

$$\mathbf{\Delta}_i = 0.1 * \mathbf{F}_{ti} \qquad (6)$$

The nodes move without acquiring any kinetic energy or momentum in the process.

The initial conditions of a given network are specified using a 2-D network topology using the node coordinates and connections or only in terms of a connectivity matrix and the 2-D node coordinates are initialized randomly. All other VS-based multi-dimensional coordinates can be either initialized to zero or chosen randomly. The multi-dimensional random kick and other forces ensure that the network configuration evolves in the full N_d-dimensions. The network evolves into the final VS configuration that enables efficient VS-based routing in spite of the randomly chosen initial conditions.

The network is allowed to evolve under the influence of the forces specified above for a fixed number of (say, 40) iterations. The final VS configuration tends to have inter-nodal VS-distance d_{ik} that roughly matches with N_{ik}. To measure any deviation from it, a parameter, δ_{ik}, is defined as,

$$\delta_{ik} = |(d_{ik} - N_{ik})| / N_{ik} \qquad (7)$$

An average value of all the δ_{ik} values in the configuration is called average deviation, δ_{avg}. It is calculated after each iteration and tracked to ensure that it decreases initially and then saturates to a low value (about 0.2 or less) near the end of the evolution. The final VS configuration of the network has all the network topology information embedded in it.

D. *VS-based Routing*

The VS based routing at an intermediate node, n_i, in the VS domain (described later) is based on the VS address of the final destination node, n_D, of the packet. A cost function, C_{ij}, is evaluated for each outgoing link from the node n_i to its neighbor n_j given by

$$C_{ij} = C_1 * (1 - \cos\theta_j) + C_{2ij} + C_{3j} \qquad (8)$$

Where C_1 is a constant and is fixed at 0.5 for the simulation results presented here. The angle θ_j is an angle between the directed distance to the final VS destination $\mathbf{d}_{iD}$ from n_i to n_D and the outgoing link direction given by the distance vector $\mathbf{d}_{ij}$ from n_i to n_j in the multi-dimensional virtual space. It can be seen that when the final destination node is perfectly aligned to one of the outgoing links in the virtual space the first term in the cost function will become zero.

The second term in the cost function is a link cost function, C_{2ij}, for the link between n_i to n_j nodes. It reflects the dynamic traffic conditions and any direct cost factors involved. As the link and buffer capacity utilization increases this term increases in value making the link a less attractive choice. For the simulation results presented here C_{2ij} is chosen randomly between 0.2 and 0.8. A failure of a link is indicated by making this term go to 2.0, which is specified as a broken cost level and the link is automatically removed from contention. The third term C_{3j} is a node cost term for node n_j and is an average of all the outgoing links from that node. This gives the forward visibility for avoiding any network congestion downstream. Any direct costs associated with the node can also be included if required.

The routing can be either single-path or multi-path, depending upon the channel requirements. In single path routing, each packet is routed along the link having the lowest total cost. The advantage of single path routing is that as long as the link costs are unchanged, a given data stream will follow the same path and the sequence of packets remains unchanged. Whereas, in multi-path routing one or more low cost paths can be chosen. In the multi-path simulation results presented here, all the links having total costs within a fixed cost differential of +0.5 from the lowest available value are chosen with equal probability. The present routing scheme can incorporate multi-path and stochastic routing without any additional resources in terms of time, cost, storage, or complexity. It provides a quick recovery from any link or node failures.

Additional procedures for efficient completion of the path and overcoming any anomalies are not included due to lack of space in this short paper.

E. *Implementation Details*

A network domain can be established, which uses VS based routing for forwarding IP data packets. This routing method need not be used as a stand-alone IP routing scheme but can also be integrated with other protocols (IP, ATM, or MPLS). At the edge of the domain the address conversion from the destination IP address of each incoming packet to the destination VS address takes place by adding an extra header or modifying the existing one. It is estimated that 7 or 8 byte VS header can be sufficient depending upon the size of the network and dimensionality of the VS configuration. The header should include the destination VS address (8-10 bits per VS coordinate, 5-6 dimensions), its version tag (2-3 bits), priority level (2-3 bits), time-to-live or TTL field (6-8 bits), and other functionality.

The computation involved in the transformation from the ordinary network topology into the VS configuration can be handled by a specific central node in the network. After the task is completed the node can send relevant (specified in later sections) VS coordinate information to individual nodes in the network. The central node can also perform the role of an ARP-server (ARP: Address Resolution Protocol) to resolve the IP or ATM address in terms of the VS coordinates and vice versa. To avoid a single point of failure the role of the central node can be duplicated by using plurality of central nodes and the ARP task can also be subdivided into different network domains. As the algorithm for generating the VS configuration is fully deterministic, the information from the multiple computations remains consistent.

It is envisaged that the VS configuration would have to be recomputed after a sufficiently long duration (in weeks or months) as the network topology changes significantly. Any dynamic network conditions such as link buffer utilization or temporary failure of a link or a node, are reflected in the link and node costs. These dynamic conditions do not require recomputation of VS configuration. An extra 2-3 bit tag can be added to indicate the version of the VS address to maintain consistency between multiple central nodes. Also, the information regarding the network topology and any long-term changes in terms of new nodes or links has to be consistent. The topology information can be expressed in terms of the least number of hops (N_{ik}) matrix having dimensions N_M x N_M, where N_M is the total number of nodes in the VS based network domain. The row and column indices of each element indicate the starting and ending nodes, n_i and n_k, respectively. Naturally, the N_{ik} matrix is symmetric with the diagonal elements being zero and the direct connections indicated by $N_{ik} = 1$. If the maximum value of N_{ik} is limited to 10, then the least number of hops matrix $[N_{ik}]$ can be built using only 9 extra steps. Any node pairs with the least hop distance larger than 10 are treated as $N_{ik} = 10$ in this formulation.

In order to process the incoming packets at a fast speed each node should store the multi-dimensional VS coordinates of each direct neighbor as well as the most recent link costs and node costs associated with its direct connections. Some of the calculations associated with total cost functions for each link can be pre-computed. Since the cost calculations for each link are independent, they can be done in parallel. The proposed VS routing algorithm can be implemented in software and executed on the network processor or main CPU or on the forwarding engines at individual input ports. It can also be implemented in hardware using an FPGA or a dedicated low complexity ASIC. The implementations in both, software as well as hardware can be simple, fast, and cheap. The algorithm has natural parallelism that can be exploited for further speed-up.

III. Simulation Results

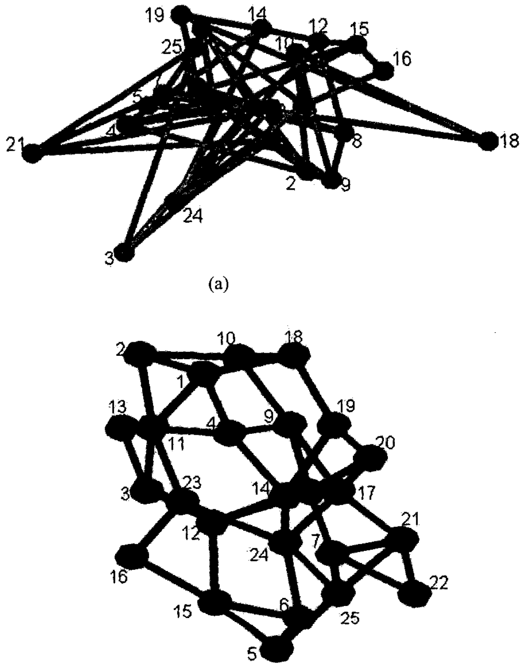
For simulation purposes arbitrary networks have been constructed using randomly chosen connectivity matrix and random node coordinates. In this case the network is specified in terms of only the total number of nodes and the average connectivity per node. We have carried out simulations for randomly constructed networks with 25 and 100 nodes with average connectivity of 4 and 6 each.

An example of the transformation from a planar 25-node network topology into the 3-D VS configuration is shown in Fig. 1. The figure shows how the nodes get rearranged to reflect the original network topology information in terms of the VS embedding. The VS configuration has all the nodes with direct connections separated by approximately one unit distance and the directed path between distant nodes tends to indicated path with the least or low number of hops. The larger networks of 100 and 200 nodes require VS based dimensionality of 4 or 5, depending upon the topology and the average connectivity of the network.

We have simulated several randomly constructed networks to check the performance of the VS based routing. The table in the next column shows the average throughput results for 100 node networks with the average connectivity of 4 and 6, using the five-dimensional VS based routing with the multi-choice and single-choice schemes. The link costs (indicating the loading level) are randomly distributed between 0.2 and 0.8. In order to demonstrate the VS based routing under the dynamic network conditions, simulations are carried out with arbitrary link failures. We have simulated 5 % and 10 % link breakages (randomly selected) as the conditions of extreme adversity in the network. The results show high throughputs under all conditions. The multi-choice routing scheme with the high connectivity networks performs better by finding alternate paths and achieves high throughput.

As a particular link gets more traffic, it leads to higher utilization of its link and buffer capacity, which increases the link cost, C_{2ij}. Hence, the subsequent data streams would tend to bypass the link by choosing alternate path, if possible. This automatically tends to balance the network loading and can bring down the capacity utilization of the busy links over time. This load balancing feature of the VS based routing scheme can be used to develop a traffic engineering tool for network management.

The simulation results presented in Fig. 2 show how the path selection of a given link is affected by the link cost function. As seen from the plot the link usage tends to fall with its cost as the VS based routing tends to find bypass paths. Thus, the VS based routing can lead to a fairly even load balancing in a network even under dynamic, bursty, and unpredictable traffic conditions for both, single-path as well as multi-path routing. Hence, it can result in better network utilization, higher throughputs, and lower congestion.



(a)

(b)

Fig. 1. 25-node network topology with numbered nodes showing (a) the original planar network, and (b) the final 3-D virtual space configuration.

TABLE I
Performance Matrix for VS-based Routing

Link Breaks	Connectivity = 6	Connectivity = 6	Connectivity = 4	Connectivity = 4
↓	Multi-choice	Single choice	Multi-choice	Single choice
0 %	99.94 %	99.92 %	99.93 %	99.86 %
5 %	99.81 %	99.86 %	99.67 %	99.15 %
10 %	99.76 %	99.73 %	99.24 %	98.57 %

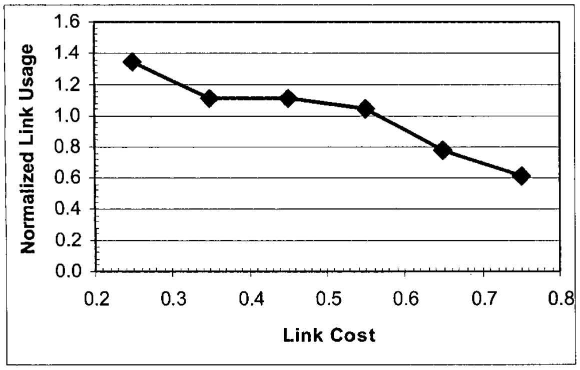


Fig. 2. Plot of the normalized link usage as a function of its cost for a 100-node arbitrary network with an average connectivity of 4.

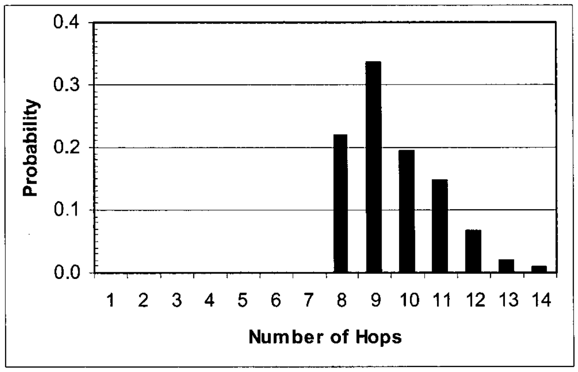


Fig. 3. Plot of probability of completing a path (Y-axis) in a given number of hops (X-axis) for all source-destination node pairs having the least number of hops of 8 for a 100-node arbitrary network with an average connectivity of 4.

Figure 3 shows a plot of the probability of completing a path in a given number of hops for all source-destination node pairs having the least hop distance of 8 for a 100 node network with an average connectivity of 4 and employing VS based multi-choice routing. The results show that 90% of the paths have between 8 to 11 hops, *i.e.* the VS based path selection closely matches with the least (or low) number of hops. In addition the VS based routing takes into account the dynamic link costs helping to alleviate traffic congestion with minimal exchange or storage of the routing information. Since the information about a link or a node failure, or any dynamic load conditions does not flow to all distant nodes, the network quickly adapts to the dynamic traffic conditions. This scheme can avoid the problems in traditional IP routing such as, load oscillations or delay in converging to new routing paths.

Simulations demonstrating the load balancing in the network are currently being carried out. Initially the link cost function that is linear with the link utilization is being considered. The traffic pattern evolves iteratively due to the feedback between the path selection and the routing decisions. Each path choice affects the link utilization, and hence the link costs, which in turn change the routing decisions. The link costs are modified in each iteration to reflect the current network traffic. It is possible to simulate time-varying traffic patterns to investigate the routing performance under real-life conditions.

Instead of randomly selected node connections, the network topologies can also be constructed by drawing connectivity on paper. Such topologies can have higher percentage of short-range connections (to the local or neighboring nodes), which resemble the real networks better. As a realistic example, a British Telecom backbone network topology with about 90 nodes presented at their website [5] has been considered (without knowing any additional details). The VS based routing for this topology yielded 100% throughput for 0, 5, and 10% random link breakages. Thus, the VS-based routing performs even better with the realistic network topologies as compared to the arbitrary topologies presented here.

IV. Advantages and other Applications

The large size of the router look-up table (in excess of 100,000 entries in the core IP routers) remains a performance bottleneck for high-speed operation and the main reason for the high costs in the router hardware and software. The VS-based routing scheme presented here eliminates the routing table, making the routing functions simple, fast and low cost. The costs associated with the fast memory, routing updates, data management, and route processing in conventional routers can be reduced substantially. The VS based routing can be used as a stand-alone IP routing protocol. It achieves high throughput for single-path as well as multi-path routing. In both forms it achieves good load balancing in the network, reducing any chances and severity of network congestion.

The VS based routing can also be used along with other protocols (IP, ATM, and MPLS) as a additional tool for traffic engineering and network management. It uses multi-faceted link cost function to achieve traffic engineering under dynamic load conditions. It can support differential services by using multiple priority levels for different data flows or applications. The priority-based differential services can be supported by defining a different cost function coefficients and attributes for each priority level. A certain portion of the link and buffer capacity can be reserved for the high priority traffic or specific QoS requirements. This would ensure the highest quality of service to the highest priority traffic.

The VS based routing scheme can be used with the conventional IP, ATM, or MPLS for path selection, QoS support, protection, and traffic engineering. It can be used for establishing virtual circuits or label swapping paths in terms of ATM virtual path and circuit identifiers (VPI/VCI) or MPLS labels. The path establishment request can be routed using the VS configuration overlay as a constraint based routing and the required resources can be reserved using RSVP to provide the quality of service (QoS) guarantees. In case of multi-protocol environment the header should identify a type of the request packet being sent. An additional advantage of the VS based routing that can be exploited is in terms of establishing an quick alternate path if one of the links from the original path fails. The VS based routing can quickly establish back-up path in order to support critical real-time applications.

The VS based routing can be used as a control plane protocol for setting up back-up paths or protection facility in the generalized MPLS or GMPLS. Currently, many existing TDM networks use SONET or SDH ring topologies, such as unidirectional path-switched rings (UPSR) or bi-directional line-switched rings (BLSR) to provide the protection facility. Otherwise protection is provided by using dedicated back-up paths (or 1 + 1 protection) in mesh-type networks, such that the working and protection paths do not share any common links. The fixed back-up paths or ring topologies (UPSR or BLSR) can restore all the existing communication channels in less than 50 milli-seconds after detection of a link or node failure. This quick restoration is important for real-time applications such as voice, video, and mission critical tasks.

However, a lot of the network bandwidth is left under-utilized in order to provide the protection facility. Alternately a mesh-based optimal protection can be used, where new paths are determined for all channels affected by a failure. It requires sending the failure information back to the source and destination nodes to initiate rerouting process. Although it leads to a significant improvement in the network capacity utilization, the channel restoration times are rather poor. The rerouting task involves the participating nodes to converge to a new solution to reestablish all existing connections. Hence, the process is a lot slower than the SONET protection.

Whereas, the VS based routing uses mesh based network topology and restores the back-up paths quickly. Just like the BLSR protection, in the VS based routing only the nodes connected to the failed link complete the rerouting of the network traffic through the available links providing the necessary protection. Thus, the proposed scheme can enable both quick restoration and better network utilization. Hence, it can lower the costs associated with the new infrastructure build-outs and upgrades.

The VS based routing can be used to control multi-commodity flow systems, such as distribution of various commodities from manufacturing sites to the warehouses. In this application trucks carrying different goods act like packets in the communication network and roadways act as different links between the nodes. With the just-in-time deliveries to minimize the inventory the distribution plays an important role. The VS based routing can be used to quickly adapt to the dynamic flow conditions.

In conclusion, the VS based routing scheme presented here is envisioned to be useful for many possible applications. A detailed simulation study on various aspects of this scheme is currently being pursued. Additional simulation results will be presented during the conference.

REFERENCES

[1] S. Keshav, "An Engineering Approach to Computer Networking", Addison Wesley Publ., Reading, MA, U.S.A., 1997.

[2] B. Davie and Y. Rekhter, "MPLS Technology and Applications", Morgan Kaufman Publ., San Francisco, CA, U.S.A., 2000.

[3] Waldvogel, et al., "Scalable High Speed IP Routing Lookups", Proceedings of ACM SIGCOMM'97, Sept. 2002, pp. 25-36.

[4] Ruiz-Sanchez, et al., "Survey and Taxonomy of IP Address Lookup Algorithms", IEEE Network, Mar./Apr. 2001, pp. 8-23.

[5] http://www.ignite.com/internetservices/BTnet/network_backbone.htm