An Architecture for Virtual Circuit/QoS Routing

Alvaro Guillen , Ramin Najmabadi Kia, Bernard Sales.

Brussels Universities-Service Télématique & Communication
Bld. du Triomphe CP 230 B-1050 Brussels Belgium

Abstract

With the emergence of multimedia applications, networks have to provide more and more bandwidth and should guarantee the quality of the communication between users. To address these issues, high speed architectures based on Virtual Circuit model such as ATM have been designed and successfully implemented. The current trends in telecommunication deployment indicate that very large Internets based on such model will be available in the near future. This paper addresses the routing problem in this context and analyzes the existing solutions. We conclude that the network has to provide pre-computed and on-demand routes using a variation of the Distance Vector algorithm. Among other things, source routing and complete route recording are used to prevent loop formation.

Keywords: *Routing, Distance Vector, Link State, VC Routing, QoS based routing, Network Interconnection.*

1. Introduction

Routing is a fundamental aspect of computer communication. In particular, efficient and operational routing procedures are a pre-requisite for the global networking environment. The routing procedures may be characterized by aspects related to distributed versus centralized computation of network paths and by their adaptability to environment changes.

For the datagram case, the Internet community has pioneered the routing protocol development. They have defined routing protocols for different environments. Examples of such protocols are [RIP] [ARP] [OSPF] [BGP]. This situation is continuously evolving and new protocols are currently designed and implemented [BGP4] [IDPR]. On the other hand, ISO has taken benefit of the valuable experiences of the Internet community and proposed a solution for the datagram routing based on three protocols: ES-IS [ISO9542], IS-IS [ISO10589] and IDRP [ISO10747].

However, with the emergence of multimedia applications, the network has to provide more and more bandwidth and should guarantee that the communication between ESs satisfies certain characteristics in terms of performance. To address these issues, high speed architectures based on Virtual Circuit (VC) model such as ATM have been designed and successfully implemented. The current trends in telecommunication deployment indicate that a very large Internet based on such model will be available in the near future. In this environment, both public and private networks have to collaborate to support the communication between end users. This raises the question on how systems can calculate and maintain routes in a VC context having in mind that these routes have to match the user requirements in terms of Quality of Service (QoS).

2. Environments

2.1. VC/QoS model

The following terms are used in this paper:
- End System (ES): a system which is attached to the VC-network and hosting the application process,
- Switching node (SN): systems performing a packet switching based on the VC model,
- System: an ES or a Switching node,
- Source node: the switching node to which the Source ES is attached,
- Destination node: the Switching node to which the Destination ES is attached,
- Node-id: a linguistic construction, built on a given set of symbols and identifying unambiguously a system within the VC network,
- Set-up packet: the packet used by the network to establish a VC.

The environment we are considering may be viewed as composed of a set of ESs (i.e. systems hosting different application processes) and a set of Switching

nodes arbitrarily interconnected by means of links. Each System is identified by an address which is unique in the context of the network. In this context, a set of QoS values are associated with each link interconnecting two Systems which give an indication of the quality of this link. Among other things, these values depend on the Switching nodes' status, the basic capacity of the links and the traffic currently supported by the links and by the Switching nodes.

The switching technique performed by the Switching nodes is based on the point-to-point Virtual Circuit model (e.g. X.25, ATM, ST-II, ...). For the purpose of this paper, a VC is a communication path between two ESs consisting of a set of Switching nodes and links.

We do not place any constraints on the protocols used in our environment, but only mention some (minimal) characteristics of them. We consider that the system has to be capable of establishing a VC consisting of a concatenation of Switching nodes and links in the network. The data communication requires that first a VC has to be set up. Once the VC set-up phase is completed, the data is transferred on this VC.

A complete route joining two ESs can not be obtained only by considering the ES addresses. The reason for this is that, in the context of a VC, the ES can ask for some characteristics to be met by the VC. These characteristics are known as the Quality of Service (QoS) over the VC. These QoS values result from the end users requirements who expect that the communication between them have specific characteristics corresponding to the needs of the application that will be used on this VC. The QoS is negotiated during the VC set-up phase. As a result of this negotiation, a VC set-up request may be rejected either because the QoS required by the ESs can not be satisfied by the network or because of an action initiated by the destination end user.

On a given VC, the QoS is not re-negotiated. Instead, when the QoS provided for a given VC no longer matches the values negotiated during the set-up, the network may clear the VC.

For the purpose of this paper, we consider that the links are symmetric (i.e. at each moment the same QoS is available for both directions of data transfer). However, the method presented here can be easily adapted to be used with asymmetric links.

2.2. Routing topology

In a large network consisting of several thousands of systems, it is likely that these systems are administered and managed by different organizations (telecom operators, etc.) each of them having their own policy in

terms of routing algorithms, supported protocols for routing information exchange, resource allocation scheme and management, tarification, etc.

With that respect, a large network environment is generally partitioned in different autonomous entities which have to cooperate in order to support the communication between users belonging to different communities. The concept of routing domains[1] (or any other equivalent term) has been introduced to cover these requirements in several architectures. A Routing Domain (RD) is defined as a group of ESs and Switching nodes sharing the same routing algorithm, the same protocols for routing information exchange, etc. Such a domain can define its own policy with respect to other RDs (the type of routes available through it, traffic constraints,...). Routes between Source and Destination ESs belonging to different domains are composed of two parts: intra-domain part (the route supported within a given domain) and inter-domain part (the different RDs supporting the route). These different routes are combined to support the end-to-end route.

The interaction between ESs and the network (Switching nodes) is referred to as ES-IS routing.

For the purpose of this paper, only the Intra RD case will be discussed. The Inter RD routing in a VC/QoS environment is addressed in [NajSal]

2.3. More about QoS

In a distributed environment, each application has its own set of requirements concerning the "quality of the transmission". For instance, a file transfer might be characterized by the transfer of an high volume of information without any error. High throughput and error free transfer would typically be the most critical parameters. It is on a par with the fact that, for different data units, the transit delay between the users may vary within a certain range of values. A real-time HDTV (High Definition TeleVision) transmission requires also a high throughput but, on the other hand, it imposes a constant transit delay and tolerates that the error rate oscillates within a given range of values. As far as applications are concerned, we can consider that each application has its own well defined requirements concerning the underlying data transmission service.

---

[1]This structure was first proposed by the Internet community and was adopted later by ISO as a framework for routing architectures [TR9575] [ETG14].

The term Type of Service (ToS) or Quality-of-Service (QoS) is used to refer to the parameters allowing to control the characteristics of the data transmission.

The semantics of each QoS have to be unique within the network. This means that all the Switching nodes of the network have to agree on the same semantics for each QoS parameter and treat each of them based on their semantics in the same way. Even though the definition of these semantics is important, it is not directly related to the routing decisions. Instead, the values associated to QoS parameters and their variation have to be considered by the routing mechanism.

Another important aspect is the way ESs specify values for QoS parameters. Some applications require strict values for QoS parameters, but some other parameters could be viewed as topology dependent. An example of this is the Transit Delay parameter (e.g. international vs. national connections). A first solution is to consider that the transit delay is specified by relative values such as "high - medium -low". The significance of these relative values is entirely determined by the network (this could be unacceptable for some applications). Another solution is to consider that, in general, an application should consult some kind of directory services (DNS, X.500, ...) for application address resolution. The purpose of such a directory is to store information which is relatively static. This information can include the values of QoS which are "statically" known as possible to join the requested destination from the source (e.g. information related to geography,...). In this way, the user will have an indication of what is feasible.

2.4. More about VC set-up model

In a VC context, the QoS of a communication results from a negotiation between the user and the network. In the existing (point to point) VC model, the connection is first established from the Source ES to the Destination ES. For this purpose a route consisting of links and Switching nodes is selected on the basis of the QoS values requested by the Source ES. During this process, the intervening nodes may participate to the routing decision and diminish the original QoS values. When the incoming call is initiated at the destination, a set of QoS values will be proposed. These values match both the Source ES requirements and the network capabilities (the Destination ES does not intervene in the selection of the route).

For each parameter, the destination user selects one value within the range received from the network, indicates it in the call accepted packet and sends it across the established connection (e.g. see the model

described in [ISO8348] and [ISO10028],...). The major drawback of this method is that the connection is established only on the basis of the Source ES requirements and the network capabilities without considering those of the Destination ES.

Another model proposes to skip the negotiation process in the following way: the Source ES proposes the minimum service it agrees to use, and the network tries to establish a communication satisfying at least these requirements [RFC1363]. If the Source ES is always able and willing to handle these QoS, this method works well. However, if the destination can not or will not handle the call with these QoS, it may refuse the call.

A drawback of this method is the fact that specifying only the minimal requirements is not satisfying in some circumstances. For example, an application trying to get the best QoS within a range might have to try several times to establish a connection before succeeding, starting with the highest QoS values, which are rejected, and going down to the minimum acceptable. The rejection of the call with high QoS values might be a result of the Destination ES's incapability (or unwillingness) to accept these values. A possible way to take into account the Destination ES's requirements is to include these requirements in the refusal message. In this way, the Source ES can retry to establish a new connection with these new QoS requirements matching both Source and Destination ES requirements. The drawback of this method is that it implies an additional round-trip for the VC set-up phase.

As a result of this, a new model for VC set-up is needed. In this model, the communication is established on the basis of the Source and the Destination ES requirements. To do so, the network calculates an appropriate route and sets up a VC over this route. In this context, the ideal scheme would be:
1) the Source ES sends a message indicating its intention to communicate with a given Destination ES with certain characteristics in terms of QoS,
2) the Destination ES selects QoS values that it can support and that matches the request from the Source ES,
3) an appropriate route matching the QoS selected by the Destination ES is calculated,
4) the VC is set-up on the basis of this route,
5) the number of round-trips for the VC set-up should be one, as with the current models.

In general, this leads to a more accurate resource allocation scheme and management. The general problem is to design an architecture in terms of routing and VC set-up model which is as close as possible to the ideal case we mentioned above.

3. A review of routing methods for QOS/VC routing

3.1. SPF Distance Vector and Link State

The SPF methods used for packet switching routing can be classified in two main classes: the Distance Vector (DV) algorithm based on a method developed by Bellman-Ford [BellFord] and the Link-State (LS) based on a method originally proposed by Dijkstra [Dijk]. Examples of DV implementation are provided by RIP [RIP], DECNET Phase IV algorithms [SHW-86] and IDRP [ISO10747] (the OSI protocol used to route the ISO-IP traffic in the inter-domain case). Examples of LS protocols are IS-IS protocol [ISO10589] [RFC1195] [Callon] (designed for the ISO-IP and adapted for IP), OSPF protocol (designed for IP environment) [OSPF] and IDPR [IDPR] (a link state protocol for inter-administrative routing).

In the basic DV method, the node keeps trace of its "distance" to each Destination. When a node detects that a distance to one of its neighbors has changed or when it receives an update indicating that the distance to one Destination node has changed, it advertises this information to its neighbors (except to the one from which it received this information). In this way, nodes can maintain routes corresponding to the shortest path based on particular metrics between any pair of Source/Destination ESs. With this method, the route calculation process is performed in a distributed way, each system keeping track of one or more routes for each Destination node it can reach.

The storage complexity for basic DV is given by $O(m.d.g)$ where m is the number of metrics, d is the number of ESs in the RD and g is the maximum degree of the graph edges in this RD.

On the other hand, with the basic LS technique, a node is aware of local topological changes (e.g. metrics changes, failure of nodes, ...), it constructs a link state PDU which will be broadcasted in the entire network. In this way, each node of the network can maintain the complete topological map of the network. The route calculation is performed in a centralized way by using this global topological map (by using for example, a Dijkstra-like algorithm). The storage complexity is in the order of $O(m.n.g)$ for the graph and $O(m.d)$ for each SPF route (n is the number of Systems = ES+SN). Once a topological change is detected, this requires the execution of an SPF style algorithm for each Destination node m times (m is the number of metrics), i.e. a

processing complexity in the order of $O(m.d.n.g)$. In the basic LS algorithm, the SPF algorithm used should be the same in each node for the same Routing Domain in question otherwise long term loops may occur since two nodes may disagree on the "best" route to be used. A Routing Domain has a limited size in terms of nodes and ESs supported, both DV and LS methods scale in such environment.

3.2. Pre-computed routes vs. on-demand routes

In a datagram environment such as IP and CLNP, it is important to have pre-computed routes since each packet is to be routed and forwarded accordingly. This is possible since 1) these architectures use a limited number of metrics and 2) these metrics are never combined. This also means that routes are maintained on-line: for LS, each time the topology changes, the SPF algorithm is executed while in the basic DV, when in a node the distance to a Destination node changes, this node advertises the modification as soon as possible to its neighbors.

As far as QoS/VC routing is concerned, it is simply not possible (or far too expensive) to pre-compute and store a route for each source/destination/QoS combination. For the DV method, this means that a Switching node should continuously maintain a routing base in the order of $O(v^{**}q.d.g)$, where q is the number of different QoS parameters and v is the maximum number of different values that one QoS parameter can take.

In the LS case, a LS PDU (Link Sate PDU - i.e. a message sent by a node indicating the state of its links) can carry the m QoS values (or for some QoS parameters, a min and a max. value indicating that any value in the range are supported) characterizing a given link. In this way, the complete topology map including QoS characteristics associated to each link can be easily maintained. However, a pre-computed method requires to execute in the order of $O(v^{**}q.d)$ times the route calculation algorithm in each node.

We can note that the request in terms of QoS required by the ESs in the network generally corresponds to few well-know pattern of QoS combination values. Examples of this are given in the previous section. This means that a number of requests are predictable in terms of QoS combination requirements for which routes can be computed by the network in advance. Nevertheless, we can not exclude that the users might require an unpredictable combination of QoS parameter values.

This means that the number of different routes to a single Destination ES that the network should maintain may be lower than P, with $P<<v^{**}q$. The value of P

depends on the resources that the network is capable to allocate for route storage and calculation purposes.

3.3. Link State vs. Distance Vector

About the LS method in a VC context, we formulate two observations: the first one concerns the topology distribution process and the second one the route calculation itself.

1) The process of updating the link state data base can be based on different strategies. At one extreme, an LSP PDU can be generated by a given node each time a new VC is established through it. At the other extreme, such LSP PDU is sent only in case of a topological change (i.e. node and/or link failure and node/link set-up). The second one is very poor in the sense that it induces a kind of distributed backtracking algorithm at the VC set-up phase because of lack of network load information. Surprisingly, the first one could be as poor as the second one since the network is always unstable. A solution for this problem is that the nodes send updates less frequently, e.g. when a particular condition in the nodes holds (e.g. a particular threshold is reached,...). However, in this case, the number of attempts to establish a VC before succeeding directly depends on the tuning of these threshold conditions. Other authors propose to geographically limit the advertisement of LS routing updates. However, these techniques can not always be applied to calculate routes to all the Destination nodes. In addition, the same problem mentioned above remains.

2) As in our VC environment the quality of a route depends on a combination of different QoS requirements. The route computation induces either to a backtracking algorithm (i.e. time complexity = $O(Exp(n))$) or to a storage complexity in the order of $O(Exp(n))$.

The distributed nature of the DV class of algorithms makes the on-demand route calculation more difficult. In the DV method, routes are calculated in a distributed way and the route calculation is initiated from the Destination node. A possible solution to fix this problem is, on a user request, to ask the Destination node to initiate the DV algorithm in order to establish the required route. However, this implies that additional round-trip time is needed to complete the VC set-up phase.

3.4. Loop problem

It has been shown in the literature that the basic DV solution suffers from "long term loops" which are inherent to the method used to establish routes [Cheng] and from "transient loops" during the convergence process of the algorithm due to the fact that inconsistent routing tables are present in systems. Solutions have been proposed to limit the long term loop problem (e.g. Split Horizon, Hold-count, complete route recording, ...). In contrast, LS protocols only induce transient loops. This only results from the dissemination of LS information because nodes have inconsistent copies of the complete map topology. In addition, LS has a convergence time that is better than DV (in the order of the "diameter" of RD). Furthermore a new class of DV algorithm has been designed to avoid loops during the DV route calculation process. In these circumstances, the DV and the LS algorithm have a similar behavior with respect to the convergence time [Shankar].

To provide loop free routes which are essential in a VC context, either a loop detection or a loop avoidance mechanism is to be supported. Examples of loop detection techniques include the conveyance of a connection identifier with the call packet or the support of a complete route recording function. Loop avoidance can be achieved for example by the use of a completely loop free routing algorithm (both transient and long term loops)[2] or by Source routing. These algorithms have been extensively discussed in the literature (permanent and transitory loops at distance-vector-like algorithms and transitory loops at link-state-like algorithms) [Cheng] [ZauGar]. However, usually the suggested solutions were related to SPF (Shortest-Path First) search which either prevents their adaptation to QoS based routing [Awerb] or this adaptation would imply a very poor performance [Garcia89]. In addition to loop avoidance, Source routing has a number of advantages:

1) For LS, one advantage is that only one system has to calculate the route. Another advantage is that different nodes can use different algorithms, this can be useful, e.g., for testing purposes.

2) For DV method, this allows to design on-demand DV algorithm in which the nodes are "memory-less " with respect to the DV method on the condition that the complete route is recorded during the route calculation process.

3.5. Our basic proposal

The method that will be described in this section aims to achieve a trade off between user requirements in terms of QoS sensitivity of routes and the network

———————————————

[2]DUAL [Garcia89a] provides solution to both DV and LS loop problem. However, we don't address this possibility in this paper.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.