DUAL-MODE ROUTING: A LONG TERM STRATEGY FOR IP OVER ATM

Bernard Sales, Philip Dumortier, Piet Van Mieghem

Alcatel Alsthom CRC, AReNA team, F. Wellesplein 1, 2108 Antwerp, Belgium

The purpose of this paper is to investigate how to design an integrated routing architecture for IP and ATM meeting the requirements for a large scale Internet based on IP and ATM. Integration of IP and ATM at the routing level leads us to consider two separate aspects: using a common routing architecture for IP and ATM on one hand (layer integration) and, on the other hand, integrating best-effort and QoS traffic support in the same routing architecture (service integration). The first level of integration is, for obvious reasons, highly recommended. In contrast, we show that the second level of integration is not desirable because best-effort and QoS traffic flows have, in terms of routing, contradictory requirements. To conduct this analysis, we feel that, because of the inherent complexity of the problem, confronting the existing proposals is too restrictive. Instead, we propose to go one step back in the design process and identify the basic design options to be considered when designing a routing architecture. We identify three options, namely, route updating vs. route pinning, hop by hop vs. explicit routing and pre-computed routes vs. on-demand route computation. Using this framework, we conclude that best-effort traffic flows are well served by a combination of route updating, hop by hop routing and pre-computed routes while QoS flow routing is built on route pinning, explicit routing and on-demand route computation.

1. INTRODUCTION

Up to now, most traffic generated over computer networks corresponds to applications which are relatively basic and which do not include features to support inter-personal and human oriented information exchange. Examples of those applications are file transfer, remote terminal access and basic electronic mail. For these applications, the Internet protocol suite built around the best-effort service model has proven its flexibility and its efficiency.

However, since the early 90s, there is a tendency to run more and more sophisticated services over the Internet including the transport of voice (Internet telephony and audio conference), video (video-conference, ...) and Work-group based applications (collaborative work). Other applications mix audio, video, text, fixed images, ... This emerging class of applications is generally referenced under the generic name of multimedia. It is expected that these applications will be very popular in the short term perspective. Such applications require new services from the network including generalised multi-peer/multi-flow communications, high bandwidth and quality of service (QoS) guarantees.

The deployment of optical fibers at a large scale combined with the development of the cell switching technology (namely ATM) provides a flexible infrastructure to support the requirements imposed by these demanding applications. ATM has an advantage regarding multimedia applications as it has been designed with QoS in mind. However, the ATM model is not well-suited for a number of applications for which a datagram best-effort model is sufficient. Under pressure of the research community, Internet is moving to a multi-service network and thus can be perceived as a competing solution to ATM.

Rather than to oppose them, we advocate that it would be better to design solutions allowing a convergence and take advantage of the respective complementarity: Internet is designed with global connectivity in mind (“IP over everything”) and is providing best-effort services, while the ATM model is built on sophisticated resource management model and is offering the hard guarantees required by multimedia applications.

A typical (data) flow can be viewed as a two phases process: first, determine the network resources needed for this data flow and next, set-up for this data flow a network path with sufficient resources to meet the user requirements. This path determination function is generally referred to as *routing*. This illustrates the very centric role that routing plays in this process.

Considering the myriad of proposals for running IP over ATM, we are facing the fact that routing procedures for IP and ATM are badly integrated. In this paper, we investigate how to design an integrated routing architecture filling the requirements imposed by a very large scale Internet based on IP and ATM. To conduct this analysis, a classical approach is to start from the existing proposals for IP over ATM, and confront them to select the architecture matching best the targeted objectives. We feel that, because of the complexity of integrating IP and ATM, a better approach is to go one step back in the design process and identify and discuss the basic design options which must be considered when engineering a routing protocol.

2. BASIC DEFINITIONS

The network environment we are considering is a mix of *host systems (HS)* and *network nodes (NN)*, interconnected by *links*. Host systems are running applications that from a network perspective are the source and/or sink of information. Network Nodes (NN) basically forward data and can be *routers* (performing layer 3 forwarding), *switches* (supporting layer 2 forwarding) or *integrated switch-routers* (performing both layer 2 and layer 3 forwarding). In this context, the considered layer 3 protocol is IP and the layer 2 one is ATM. However, the study here is to a large extent generic and therefore applicable to any combination of layer 2 and layer 3 technologies.

For the purpose of this paper, a *(data) flow* is defined as a set of correlated data transferred from a source HS or NN to one or more destination HS or NN (a flow is therefore unidirectional). A flow is characterised by a set of service requirements to be met by the network while transporting the flow. When such service requirements include firm guarantees in terms of traffic pattern profiles, the flow is called *a QoS flow*. A flow for which the only service requirement is that the major proportion of the packets reaches its specified destination(s), is defined as a *best-effort flow*. A flow can be defined recursively, in the sense that a flow can be an aggregation of a number of finer-granularity flows. The correlation is then done on a subset of the characteristics of the constituting finer-granularity flows, and can have a more limited reach in network.

Finally, we define the following terms used in the next sections:

- Calling NN: for a data flow, the NN which has originated the request;
- Called NN: for a data flow, the NN to which the flow is directed.

3. BASIC CHOICES IN THE DESIGN OF A ROUTING ARCHITECTURE

The design of an advanced routing architecture is a process where a number of independent basic options should be evaluated and combined.

3.1 Route updating vs. route pinning

Route updating indicates that the route of a data flow is dynamically altered after any environment change that could affect the route (failures, topology update, congestion,...). This technique tends to maintain optimal routes (according to the applied metrics) in all circumstances. However, it is vulnerable to stability problems when routing metrics are too dynamic.

Route pinning implies that the route of a data flow is kept unchanged for a specified duration of time, unless some exceptional condition occurs (node failure, link failure, network management action, ...). With this technique, data flows are supported by stable and well controlled routes.

When choosing between route updating and route pinning, we have to take into account that dynamic distributed routing schemes can generate cyclic paths (called *routing loops*). Two types of routing loops are possible [Garcia] [GulNajSal]: *long term loops* caused by malfunctioning algorithms that computing the routes, and *transient loops* which result from inconsistencies between network node routing tables, due to the inherent delay in the distribution of routing information.

An architecture with route updating must only deal with the first type of loop: potential loops formed as a result of routing base inconsistencies will only persist during the convergence phase of the algorithm; they will naturally disappear when the routing bases get synchronised. In contrast, when route pinning is applied, the routing algorithms must obviously avoid to take routing decisions that might generate routing loops [NajSal]. Otherwise, there is a risk that routes with loops will be maintained until the routes are unpinned.

3.2 Hop by hop vs. explicit routing

In *hop by hop routing*, the network nodes along the path determine the route to be followed (according to the flow requirements) and consecutively specify the next node on the route. Each node needs to apply the same routing algorithm and the route is built and maintained in a hop by hop way with the collaboration of all network nodes along the path.

With *explicit routing*, one particular node (e.g. flow source node) selects and specifies an appropriate route for the flow. The explicit route is then notified to the other network nodes along the path. In this way, routing decisions are taken in a pseudo-centralised way.

As hop by hop routing maintains routes in an incremental way, only the part of the path to be modified will be affected. With explicit routing, as only the network node that specified the route can edit the route, the complete route needs to be re-installed in all network nodes along the path.

Explicit routing on the other hand is more robust against routing loops, as the complete route is determined by a single network node, i.e. based on the same (by definition synchronised) database. In contrast, hop by hop routing is potentially subject to both permanent and transient loops. In this mode of routing, the formation of loops can be avoided by implementing routing protocols that keep the network node routing bases in a sufficiently synchronised state, such that inconsistent routing decisions will not occur.

3.3 Pre-computed routes vs. on-demand route computation

Pre-computed routes means that routes are computed and maintained in advance by the network, independent of actual flow routing requests. *On-demand route computation* on the other hand implies that the routing process is activated only when a data flow actually needs to be routed.

Pre-computing routes offers a better performance in terms of flow set up delay. Indeed, when the node needs to route a flow, it just has to fetch the correct entry from the routing table, without the burden of a costly route computation. Nevertheless, this method has drawbacks since the performance of pre-computed routes is bounded by the reliability (e.g. ageing) and accuracy of the information stored in the routing base at the time it is consulted.

Pre-computations also require that all possible routing requests need to be computed. The efficiency of pre-computing routes is very much dependent on the number of combinations that need to be computed, compared to the probability of consulting a specific entry.

4. INTEGRATED ROUTING

Our goal is to determine the routing mechanisms that are needed in an integrated IP and ATM environment such that both best-effort and QoS flows can be efficiently supported. In this particular context, integration can have two flavours:

- *layer integration*: a single routing architecture covering both IP and ATM vs. two specific routing architectures, one for IP and another one for ATM (layered routing);
- *types of service integration*: a single general-purpose routing paradigm independent of type-of-service vs. a dedicated routing paradigm for best-effort and another one for QoS flows.

Design a single layer for IP and ATM routing is for obvious reasons better that supporting two separate routing architectures, one for IP and the other one for ATM (as shown in [SalesDum]). Thus, for the purpose of this paper, we will focus on the services integration aspects which was not yet systematically studied up to now in the literature.

4.1 Best-effort flows

The best-effort service is materialised by the datagram concept in the Internet today. Its extensive use in the last ten years has demonstrated its maturity to efficiently provide best-effort services. In short, the IP datagram concept supports the best-effort service better than any other available technology. No signalling message is needed to set up a flow, and an IP router can receive datagrams destined to any host system connected anywhere to the Internet. This gives the Internet its current high level of scaleability and robustness.

First, *route updating* is applied since there is no other service requirement than to deliver every datagram individually in the best possible way to its destination. As such, if a better route can be found because of an environment change, there is no advantage in maintaining the old route. Packets belonging to the same flow (the flow being defined here as ‘all datagrams destined to a particular cluster of networks’) can follow different paths in time without any penalty. Updating routes also eliminates the risk to freeze transient loops, created by applying hop by hop routing.

Next, *hop by hop routing* is applied since applying explicit routing with the CL datagram concept would require an unacceptable overhead to store the entire route in every datagram. The only way to avoid this would be to insert an index in the datagram, pointing to the actual route information stored in the nodes [GulNajSal]. However, this would require an advance binding of the index to the appropriately initialised state information which contains the explicit route. This is no more than turning to a form of CO forwarding, which would harm the scaleability that the Internet has enjoyed up to now with the pure datagram mode. As a result, the only solution is to use hop by hop routing, which minimises the state information to be maintained in the network while still being acceptable in terms of overhead.

Third, the current Internet uses *pre-computed routes* rather than computing routes on-demand. This is justified by the fact that

- the number of possible flows (i.e. number of pre-computed routes) is limited thanks to the large flow granularity (“all datagrams destined to a particular cluster of networks”) and because a single metric (hop count) is sufficient to provide a reasonable best-effort service;

- the pre-computed routes remain fairly stable in time (only link or node failures or major configuration changes may affect a route), while serving frequent requests for route information (due to the hop by hop routing, for every arriving datagram, route information is needed), which gives the pre-computation a high utilisation rate. Besides being stable, the metric also needs to be simple enough to allow for a complete re-computation of all routes after an environment change.

4.2 QoS flows

4.2.1 Role of QoS routing

To provide flows with QoS guarantees, a traffic management function must be supported. In short, this function is responsible to insure that the QoS contract, agreed between network and user during the flow set up, is respected [De Prycker]. To anticipate that new flows jeopardise the QoS contract of existing flows, a flow admission control function is needed. With this model, whenever an NN receives a flow set up message initially submitted to the network by a calling NN, the flow admission function of the NN examines this request with respect to the traffic and the QoS parameters requirements for this flow. According to this, it can either reject or accept the demand. In this latter case, the request is forwarded to the next NN in the path towards the called NN. In the former case, a message indicating the reason of the data flow set up failure is sent back by the NN towards the called NN. Using this diagnostic, the called NN can therefore attempt to establish the flow on a new route and it can subsequently re-initiate a flow set up phase along this new route.

In this way, the routing architecture should compute routes meeting the constraints imposed by the traffic/QoS parameters based on routing information distributed by the NNs and reflecting the resources which have not yet been allocated. With this mechanism, routing can compute paths using as input an abstracted view of the resources that remain available in the network. In short, the routing decision is depending on topology information which are function of the actual use of network resources. In addition, a flow is generally characterised by a combination of QoS and traffic parameters. Consequently, routing will utilise an abstraction where every link/node is weighted by more than one routing metric. The metrics' values are distributed by the NNs with respect to the state of their flow admission control functions. Among other things, these values depend on the NN status, the basic capacity of the link and the traffic currently supported by the network elements. These routing metrics are used by the routing algorithm in order to compute a path supporting the QoS values required by the calling NN for the flow being considered.

This indicates that the routing architectures for QoS should discover paths satisfying more than one constraints. The complexity of this algorithmic problem depends on the nature of the parameters being considered by the routing algorithm. However, in the most general case, the route computation process is NP-complete.

4.2.2 Basic design choices for QoS routing

All the above observations we made in the previous section, pose severe constraints on the design options that can be used to build a QoS based routing architecture:

- pre-computing all the routes is unfeasible

Pre-computing routes for QoS routing means that we have to continuously maintain routes for any pair of source/destination HSs and this for each combination of QoS parameter values. As a result, pre-computing the routes for any combination of QoS parameter values is not feasible. Notice that QoS requests often correspond to only a few well-known profiles of QoS values (e.g. file transfer service, remote terminal access, ...): those routes can be perfectly computed in advance and maintained by the network as mentioned in [GulNajSal]. However, in most cases, the QoS combination values are not predictable, implying that a QoS routing architecture should provide for on-demand route computation [GulNajSal].

- route updating is not acceptable for QoS routing

The routing decision is based information reflecting the actual utilisation of network resources. This routing information is disseminated on a periodic and/or event-driven basis. Each time a QoS flow is set up/released, the resulting network state should be advertised in the network. For scaleability reasons however, the routing information updates are distributed only when a significant change occurs (e.g. [PNNI]). With this abstraction mechanism, the path computation function does not have an exact view of the resources in use, with the possible consequence that the blocking probability may increase.

Route updating cannot be used in conjunction with QoS routing. Indeed, consider the impact of the dynamics of routing topology information on route updating. As a part of the network resources are allocated to a QoS flow during its establishment, the flow set up itself may result in the distribution of new routing topology information. A naive application of route updating will lead to the following problem: the nodes which are responsible for maintaining this QoS flow can discover, according to the new routing updates received, that there is a better route for this QoS flow. As a result, one could decide that this QoS flow must use this new route. Thus, the resources used along the “old” route are released and the necessary resources are allocated along the new routes. Again this new situation may generate new routing updates implying the start of a recursive process that leads to route instabilities. The problem with this naive approach is that the nodes re-compute new routes for a QoS flow according to routing information which already include the resource utilisation for this flow. This problem is commonly referred to as *follow-your-own-shadow*.

More generally, as the topology information used in the context of QoS routing is by definition dynamic, the optimal route for a given flow will consequently also frequently change. Applying route updating will lead to unstable flows oscillating among these best routes. To prevent oscillations, route pinning is mandatory for QoS routing. Route pinning may have the drawback of maintaining sub-optimal routes, but at least it insures a reasonable degree of stability.

The fact that QoS routing is based on the route pinning mechanism implies de facto that hop by hop routing will not be a sound solution for QoS routing because of its fragility with respect to the loop formation.

4.3 Solution: dual-mode routing

As a result of the above discussion, routing for best-effort and QoS cannot be fully integrated. The route computation processes and the maintenance of routes are founded on different paradigms: best-effort routing is based on a combination of pre-computed, hop by hop and route updating while QoS relies on on-demand, explicit routing and route pinning. Integrating IP and ATM for the provision of QoS and best-effort flows means that IP and ATM should share the same routing architecture but that different routing modes should be used, one for the support of QoS flows and another for the support of best-effort flows. The route computation processes for QoS and best-effort may, however, use the same topological information, but they will use it in different ways. In fact, the routing protocols for the support of best-effort and for the support of QoS could even be different. We call this concept “dual-mode routing”, where the same routing is used by IP and ATM but where two routing modes are supported, one for best-effort and the other for QoS, while maintaining a fully integrated resource management.

5. CONCLUSION

In this paper, we have discussed the need for integration between IP and ATM routing. For this purpose, we go back one step in the design process and first identify the basic choices that drive the design of a routing architecture. These are: route updating vs. route pinning, hop by hop vs. explicit routing, pre-computing routes vs. on-demand route computations.

Routing can further be integrated at two levels: layer integration (IP and ATM) and service integration (best effort and QoS). While the layer integration is highly recommended, it appears that service integration is not desirable because best-effort and QoS traffic flows have, in terms of routing, contradictory requirements. This is explained by analysing the basic design choices that are required for the support of best effort and QoS flow routing respectively.

Using this framework, we conclude that best-effort traffic flows are well served by a combination of route updating, hop by hop routing and pre-computed routes, while QoS flow routing is built on route pinning, explicit routing and on-demand route computation.

As a result, integrating IP and ATM for the provision of QoS and best-effort flows implies that IP and ATM should share the same routing architecture but that different routing modes should be used, one for the support of QoS flows and another for the support of best-effort flows. We believe that the established framework for routing is generic and can be used to evaluate existing and future routing protocols.

6. REFERENCES

[DePrycker] M. De Prycker, Asynchronous Transfer Mode: Solution for Broadband ISDN, 3rd Ed., Prentice Hall, 1995.

[Garcia] J. Garcia-Luna, A Unified Approach to Loop-Free Routing Using Distance Vectors or Link States, proceedings of SIGCOMM 89, pp. 212-223, Austin, 1989

[GulNajSal] A. Guillen, R. Najmabadi Kia, B. Sales, An Architecture for Virtual Circuit/ QoS Routing, Proceedings of the First IEEE International Conference on Network Protocol, pp.80-87, IEEE, Los Alamitos, USA, 1993

[NajSal] R. Najmabadi Kia, B. Sales, Routing Architecture for the support of the Connection-mode Network Service, Proceedings of the 3rd Joint European Networking Conference, Computer Networks and ISDN Systems 25(4-5), pp 405-410, North Holland, 1992

[P-NNI] P-NNI Specification, ATM Forum, 1996.

[SalesDum] Bernard Sales, Philip Dumortier, Dual-mode routing in IP over ATM Networks, to be presented at the SPIE’97 conference and to be published in the conference proceedings.