A distributed algorithm for resources provisioning in networks

Marc-Antoine Weisser, Joanna Tomasik
Département d'Informatique, Supélec
3, rue Joliot-Curie, 91192 Gif-sur-Yvette cedex, France
Marc-Antoine.Weisser@supelec.fr, Joanna.Tomasik@supelec.fr

Introduction

Internet is a network based on IP (*Internet Protocol*). Routing between its independent subnetworks (domains) is assumed by BGP (*Border Gateway Protocol*) which uses IP. There was no need for QoS (*Quality of Service*) guarantees when these two protocols were designed and these guarantees aren't integrated in Internet. Today, the expectations changed. The introduction of new applications, like voice-over-IP, video-conference or e-commerce constraints the providers to adapt their network to include the QoS requirement. Resources reservation seems to be the only way to give strong QoS guarantees from peer to peer and not only probabilistic guarantees as an access protocol on the boundaries between the domains would give.

The existing reservation protocol RSVP (*ReSerVation Protocol* [1]) was designed for IP network seen as a single domain network. RSVP isn't scalable to the interdomain level. The scalability problem occurs because RSVP is treating each flow individually. Indeed for each flow, RSVP has :

- to create in each router in the path a *soft state table* containing flow information;
- to dedicate a leaky bucket mechanism to control the traffic of each flow for each router;
- to send *refresh messages* periodically to keep the reservation alive.

RSVP distinguishes a flow by its sender and destination addresses. In particular, RSVP may reserve paths for a great number of "small" flows. In the inter-domain context, the need of reserving path "flow by flow" make the use of RSVP unrealistic. The inter-domain resources reservation needs an introduction of a flow aggregation mechanism.

Let us take the hypothesis that this kind of mechanism is implemented in the inter-domain routers [2]. To establish a reserved path for aggregated flows, we will need a protocol which :

1. does not have to know the global network state, this means, the resources available in all the links and in all the routers;
2. should use another information than those obtained by BGP to construct the paths;
3. finds paths which satisfy a set of criteria as delay, cost, bandwidth, jitter, loss rate;
4. can create reserved paths independently from BGP routing tables.

The points 2, 3 and 4 eliminate RSVP from consideration. Indeed, RSVP use only the BGP routing table to construct the reserved path. Moreover the only constraints integrated today in the protocol are the delay and the bandwidth.

In the RSVP approach, the algorithm finds a complete path from the source to the receiver and then asks each router on the path if the demand is accepted. The path's finding is downstream, but the reservation is made upstream. If one of the routers rejects the demand, the whole path is rejected. RSVP uses the routing table of BGP to find the path. The path is created independently of the constraints, so it may be inadequate to the QoS request. It seems more efficient to construct the path downstream and to reserve the resources simultanously. Then it is possible to verify hop by hop that the constraints are still satisfied. If they are not, the algorithm can return one hop and try another solution without rejecting the whole path. Finally, RSVP allows the receiver to choose the QoS. It is also possible in the downstream reservation approach if a sender asks the receiver about the QoS request at the beginning of the reservation. The drawback is that it increases the number of messages but it can improve the paths searching. We think that it is a little loss in comparaison with the difficulties of the multi-constraints path finding.

Goals

Our work focuses on the inter-domain routing including a mechanism of aggregation. Our goal is to provide an algorithm finding multi-constraints paths between a router in a domain to a router in another domain. In this context, the algorithm has to satisfy some requirements. The providers want to limit the broadcast of information about their domains. For this reason, the algorithm should be distributed. The network's state, ie. the resources available on links and routers, is continuously changing. Today, keeping the network's state in each router implies a too important message overhead. So the algorithm has to work without the knowledge of the global network's state. As domains are managed independently by their operators, it may happen that the proposed reservation protocol is not applied in some of them. The algorithm should be able to work and to reserve a path excluding domains which do not use it. Finally, with new applications may appear also new needs for QoS. The algorithm has to be adaptable to new constraints. The algorithm has to work with at least additive, multiplicative and convex constraints whose contain the main known constraints: delay, cost, bandwidth, loss rate, jitter.

The problem of finding multi-constraint paths in a network is a *NP*-complete if the number of additive and multiplicative constraints are at least two [3]. This result implies that, even in a centralized model, to compute a path in a polynomial time we need to use an heuristic algorithm. The leak of network's state information leads us to choose a probabilistic approach.

We will present an algorithm finding multi-constraints paths which satisfy the following properties :

- it is distributed;
- it is able to work without global information about the resources available in the links and in the routers;
- the installation can be incremental, this means that it works even if only a subset of the domains uses it;
- it can support any additive, multiplicative or convex constraints;
- it is an algorithm in polynomial time: in each node the algorithm is in polynomial time and the number of communication steps is also polynomial;
- it is a *Las Vegas* algorithm (it is randomized and it produces correct results but don't always find a solution, even if it exists).

Our algorithm is based on a work presented in [4]. Its main idea is to send a *probe message* from a source router in a domain to a destination router in another domain. The probe is passing from domain to domain through the network. In each visited domain, the probe demands a reservation. The reserved path is constructed hop by hop. When the probe reaches the destination, a *validating message* is sent back to the source validating the reservation demands made by the probe message. The probe limits the number of domains visited. If the number of visited domains becomes greater than the limit, a *failure message* is sent back to the source, discarding the reservation's demands.

The choice of the path for a probe message is randomized. In each domain, when a probe arrives, a reservation's demand is made. If the reservation is accepted, the message is sent to a neighbor domain which is choosen randomly and has never be visited. If there isn't any unvisited domain in the neighborhood or if the demand is rejected, the message is sent back. When a sent back probe arrives into a domain, the reservation is already made so the message is directly sent to another domain in the neighborhood which has never be visited.

We need simulations to validate our algorithm because it is not deterministic. We have developed a simulator with OMNeT++ [5]. We first run simulation on regular topologies (rings, trees and grids) then on topologies generated with BRITE [6]. The first results have shown that the algorithm works correctly but that the limit of visited domains is a crital parameter. We have to study this parameter because the quality of the results depends on it.

References

[1] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. RFC 2205. Sept 1997.

[2] J. Schmitt, M. Karsten, and R. Steinmetz. On the aggregation of deterministic service flows. *Computer Communications*, 24(1):2–18, 2001.

[3] J. M. Jaffe. Algorithms for finding paths with multiple constraints. *Networks*, (14):95–116, 1984.

[4] T. Korkmaz and M. Krunz. A randomized algorithm for finding a path subject to multiple QoS requirements. *Computer Networks*, 36(2–3):251–268, 2001.

[5] OMNeT++. http://www.omnetpp.org/.

[6] BRITE. http://www.cs.bu.edu/brite/.