On Routing with Forwarding on Gates

Florian Liers, Thomas Volkert, Andreas Mitschele-Thiel
{ florian.liers | thomas.volkert | mitsch }@tu-ilmenau.de
Ilmenau University of Technology

1. Introduction

Forwarding on Gates (FoG)[1] is a new clean-slate approach for an inter-network like the Internet. Its main goals are the generalization and the separation of the forwarding process from the routing process [1]. FoG is designed to work independently from protocols like IP, TCP or UDP. Furthermore, the approach supports inter-network wide quality of service (QoS) and the possibility to integrate any application functionality into the network. In terms of a mathematical graph, FoG uses edges, instead of nodes to describe paths for packets. This approach enables the support for hosts without addresses. Scalability issues, raised by using edges instead of nodes, are handled by mechanisms to dynamically distribute state information over the network. Additionally, responsibility areas for routing and QoS management are defined. FoG is able to adjust the position of state information according to the needs of the applications and the limitations of the underlying networks of autonomous systems (AS). This feature is not known in today's Internet where state information in form of routing tables is needed within each router. Moreover, these routing tables represent one of the main pillars for handling IP traffic. The size of these tables depends on the size of the Internet and its address distribution. Caused by the rapid growth of the Internet, the tables are growing year by year [2]. This, in turn, leads to an increased forwarding delay. FoG strictly separates routing from forwarding to enable dynamic state distribution. Therefore it uses a specialized service for routing and another for packet transport. This opens new possibilities, challenges and requirements for routing. This abstract gives a short overview about them and introduces some first ideas for G-Lab FoG-routing.

2. FoG's logical Components

Figure 1 shows the functional components of FoG. They are implemented in a distributed way, while each physical host, supporting FoG, is running an entity of each component. The minimum implementation (esp. of the routing service) is just a stub asking other hosts for providing the service. Since FoG decouples forwarding and routing, special services exist for each of them.

The basic service is the transport service, which is distributed over the existing physical nodes and is responsible for forwarding data packets from one functional block to the other. These functional blocks are called "gates" and they are placed automatically in-between the source and the destination application. In combination, they implement all desired functionalities on the way from the transmission source to the transmission destination. The resulting structure depends on the requirements given by the applications. For example, gates can represent links between physical hosts, higher application functions (e.g. video re-coding) or advanced network functions like packet filters or virus scanners. They are indentified in a local context by so called gate numbers. Hence, FoG can be seen as an index-based forwarding approach with routes consisting of gate numbers. Besides the described gates, the transport service of FoG includes forwarding nodes. They are responsible for the multiplexing of data packets and forward them to the next gate on their path towards the destination. Related to the mathematical view of the introduction, a gate is an edge and a forwarding node is a node in a mathematical graph. The management of the transfer service is done by special managers, which are also functional blocks within the transfer service. At least one manager instance has to be located on each physical host. The manager instances represent the control interface of FoG towards applications, other services and existing physical transmission equipment.

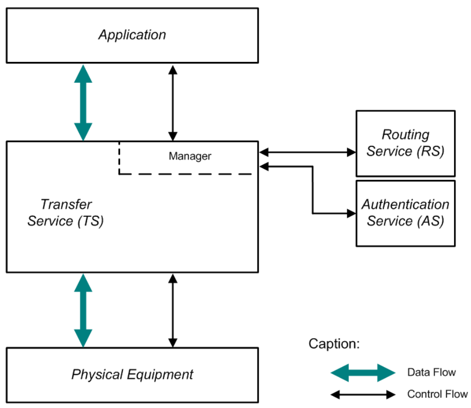

Figure 1: FoG Architecture

The selection of existing gates and the request of additional gates, which are needed for a communication between two applications, are done automatically by the routing service depending on the requirements which are given by applications. As mentioned before, the routing service is

[1] This work was funded by the German Federal Ministry of Education and Research (BMBF) with the project G-Lab_FoG, FKZ 01BK0935.

implemented by several entities spanning an overlay network above the physical network.

Additionally, FoG uses authentications for all its management and signalling processes. They are provided by an authentication service. However, this service is not in the scope of this abstract.

3. Routing based on Gates

Due to scalability reasons, a route is calculated incrementally. A routing service entity is allowed to calculate a partial route, which defines only the next steps towards the destination. In contrast to the hop-by-hop routing of IP, these partial routes can contain more than just a single hop. Is a route not calculated as a whole, the destination of a packet has to be stored as meta data within its header. If a partial route is processed completely, this meta data is used again for further route request to other routing service entities. This process has to be repeated until the packet reaches its destination.

A routing service receives topology information from transport service managers. Basically, it is informed about existing forwarding nodes and gates as well as their dependencies. A comparable announcement in today's systems is the announcement of a link to a neighbor host or autonomous system. If a gate is published to the routing service or not depends on the policy of the network and is a decision of the manager of the transport service. Every announced gate is used by the routing service in order to calculate answers for route requests. If gates are missing and a route request can't be resolved, the routing service requests additional gates from the corresponding transport managers. During this process, the manager has to decide if the requested gates should be created or if existing gates, which might be unknown by the routing service, can be used. Due to the used incremental calculation of the route, the decision of the manager is transparent to the routing service. Therefore, newly created gates can be used exclusively for a route request if the manager does not report the new gates to the routing service for general usage.

4. Routing Service for FoG

In general, the implementation of the routing service is independent from FoG. In the following, we introduce an idea for this kind of service using FoG's features.

In contrast to IP based routing with routing tables in every router, the path calculations in FoG can be based on routing zones. The zones can be calculated by distributed clustering algorithms. The members of each zone elect a coordinator for a zone. One possible solution for this can be based on the bully algorithm [3]. If the election is done successfully, the new coordinator is responsible for processing routing the requests for the whole zone. Caches for repeating requests can be used in every local routing service stub to improve the calculation performance. The entries in this kind of cache could remain valid until a forwarding node signalizes a forwarding problem or a timeout occurres. In this case a new request is triggered to the zone coordinator and the cache is updated. If very low delays, even in case of failures, are required it is imaginable to elect a second coordinator which works in hot-standby mode. In order to enable the manual selection of dedicated routing hosts as coordinators, some priorities could be integrated in the election process.

In general, coordinators enable the network to manage routes and QoS reservations in a locally centralized point. This method avoids a lot of communication overhead compared to a fully decentralized solution. However, without any further extension this leads to a structure similar to today's BGP networks because each BGP node needs to know the whole topology of the BGP based backbone. In case of FoG, this would require coordinators knowing all other coordinators. Without this knowledge, routing requests crossing zone borders are not possible to be handled. To face this issue, FoG could use a hierarchical approach which supports a dynamic hierarchy depth. Within the hierarchy every zone coordinator represents a zone member of the next higher hierarchy level as proposed in [4]. The zone selection for higher hierarchy levels is done in the same manner as for the lowest level. Additionally, necessary routing information is exchanged as well as coordinator priorities during the election process in higher hierarchy levels. As a result of this hierarchy, the zone coordinators need to know only their zone members. Hence, an optimal hierarchy depth for a given scenario has to be found leveling the amount of zone members and the delay between each member and the coordinator.

Furthermore, FoG could use hierarchical addressing to improve routing performance. Then the routing zone could be concluded from a known address. Moreover, it would be possible to use different routing algorithms for each routing zone with one common signaling interface in-between. This enables an adaption of the routing method based on the requirements of zones.

Within the presentation we will discuss the main differences between IP and FoG based routing. We will also present our idea of the FoG routing service and try to start a discussion about routing hierarchies.

[1] F. Liers, T. Volkert and A. Mitschele-Thiel: Forwarding on Gates: A clean-slate Future Internet Approach within the G-Lab project, *Euroview 2009*, Würzburg, 2009.

[2] G. Huston: Growth of the BGP Table - 1994 to Present, http://bgp.potaroo.net/, *Swinburne University of Technology*, 2010.

[3] H. Garcia-Molina: Elections in a Distributed Computing System, *IEEE Transactions on Computers*, Vol. C-31, No. 1, pg. 48-59, 1982.

[4] G. Pei, M. Gerla, X. Hong and Ching-Chuan Chian: A Wireless Hierarchical Routing Protocol with Group Mobility, *Wireless Communications and Networking Conference (WCNC)*, pg. 1538 - 1542 ol.3, 1999.