

BPP: A Protocol for Exchanging Pricing Information between Autonomous Systems

Vincent Oberle, Hartmut Ritter, Klaus Wehrle
University of Karlsruhe – Institute of Telematics
Zirkel 2, D-76128 Karlsruhe, Germany
Tel.: +49 721 608 6416/6414, Fax: +49 721 388097
eMail: {Oberle,Ritter,Wehrle}@telematik.informatik.uni-karlsruhe.de

1 Introduction

In this paper we focus on the distribution of pricing information between autonomous systems in the Internet. Therefore we propose the Border Pricing Protocol (BPP) as a protocol for exchanging pricing information in analogy to the Border Gateway Protocol [1], the current Internet standard for inter-autonomous system routing protocols. The foundation is an Open Pricing Framework described more detailed in [2]. In the next section this Open Pricing Framework is sketched as well as its relation to the Differentiated Services architecture. In section 3 the mechanisms of the Border Pricing Protocol are presented and the content of the pricing and charging information base is discussed.

2 Principles of Pricing in the Internet

This section describes the proposed model for charging and accounting in the next generation Internet. It is aligned to the structure of the current Internet. A closer view onto the Internet reveals, that it is not a monolithic block but structured into domains and autonomous systems. These units are physically connected, but the main relationship between these units can be described in terms of services, provided by service providers. A data transmission from a sender to a recipient can be described as a recursive service usage. The sender passes its data to the service provider it is connected with and relies on its forwarding capabilities. The service provider passes the data on to another service provider who again passes it on recursively to successive service providers until the data reaches the service provider the recipient is connected to and the data gets delivered.

The domain-centered structure of the Internet causes a basic problem of charging and accounting which might be called the *Problem of Direction*. It is a typical problem between service providers (cf. also [3]): It is not obvious, whether a provider A has to pay for the traffic it sends to a provider B or for the traffic

it receives. If the clients of provider A generate huge amounts of traffic (e.g., by transferring files to other locations via FTP), provider A should pay provider B for forwarding the traffic.

On the other hand, if the clients of provider A mostly request data (e.g., by performing file downloads or browsing the web where a small HTTP request can cause the sending of large amounts of data from a server), provider A should pay for the data it receives. In order to overcome the problem to decide which one has to pay, we proposed in [2] a simple model oriented at the postal service. It is the basic paradigm of the postal service that the sender always pays for its data transmission. If the sender responds to a request, it can perform a billing of its transport cost afterwards. In most cases this will be part of billing the costs of the provided content or it will be granted for free in the case of advertisements.

This paradigm can be easily transferred to the Internet. Thus the problem of direction can be solved in a simple way. This strategy is plausible taking into account that it is not possible to prevent someone else from sending data, but only oneself can avoid sending data. As the sender does not want to care about the transportation of its data once passed to the service provider, the sender does not want to pay all service providers on the way to the destination for data transport, but only the one it is directly connected with. The single relationship can be set up using the model of recursive service usage again: The sender pays its directly connected provider for the service of data transportation to whatever direction. By analogy with the described recursive service usage (provider A passes the data to provider B and so on), the providers have to pay recursively for the service of data transportation. The recursive service usage thus divides the price the sender pays among the providers.

2.1 Pricing Information

In order to implement the model of recursive service usage in the area of pricing, network providers must know in advance what the transport of data to a destination will cost. When using services that need an explicit connection setup the settling of prices can eas-

ily be done. This has already been shown in [4]. However, at least the Differentiated Services architecture also supports services without such explicit connection setup. In order to get the current pricing information (PI) for these services it must be distributed throughout the Internet. This can be done in analogy to existing routing protocols like OSPF or BGP, as both protocols deal with the distribution of information between Autonomous Systems (AS). We propose the *Border Pricing Protocol (BPP)* enabling the spreading of pricing information in the Internet. Pricing information can be transmitted upon request, as a flooding of pricing information might cause a lot of overhead traffic. Entities implementing the BPP set up a pricing information data base for destinations and perform an aging of the entries. Therefore, a Management Entity (ME) for each AS has to be identified, as it is going to be discussed in the context of the Differentiated Services architecture.

When an entry changes, the BPP can be used to transmit the change information to entities that requested price information before via a notification request. We suggest adapting existing procedures for distributing and utilizing routing information in order to distribute and utilize pricing information additionally. Routing protocols have been extensively investigated and it is reasonable to make use of them instead of developing new protocols for distributing information similar to routing information with respect to where and when it is required. Routing is also a common factor of all QoS approaches for the Internet and thus this strategy makes it possible to adapt the distribution of price information to different QoS approaches.

2.2 Relation to the Differentiated Services Architecture

As with the global deployment of Differentiated Services, not all service providers will adopt the pricing framework and will support the Border Pricing Protocol in the beginning. As the recursive service usage does not require a protocol communication from end to end this is no obstacle. In the case of a service provider not supporting BPP the financial settlement with this provider can be done the same way as before, in most cases by means of long-term contracts. Additionally, here is no need for a pricing framework to define the way charging and accounting inside a domain are realized. Different providers could implement different internal mechanisms or *Interior Pricing Protocols (IPP)*, protocol standardization is only needed for the communication between providers. The methods of an Interior Pricing Protocol are not focus of this paper, as charging and accounting inside an AS will remain an area of provider-specific interest and will be dominated by many non-technical aspects. Nevertheless, combined usage of BPP and a dedicated IPP

inside the Management Entity of the Differentiated Services domain will provide the highest benefits.

Thus, a Management Entity inside an AS is needed for realizing the BPP. In the Differentiated Services Architecture, different suggestions have been made already for some central entity, be it the Bandwidth Broker [5] or the Domain Manager [6]. These elements can be easily augmented with the BPP capabilities.

3 Border Pricing Protocol

The Border Pricing Protocol deals with the exchange of Pricing Information (PI) between Autonomous Systems. It was designed in analogy to the BGP routing protocol, the widely used protocol for exchanging routing information between Autonomous Systems in the Internet.

Like BGP, BPP is designed to run with TCP, so that the protocol needs not to be concerned about correct reception of traffic, segmentation, etc. which are handled by the transport layer. And like BGP, the protocol manages sessions, using open, keep-alive and close messages. Common information about all PI exchange between two peers (like security information) can be put in an open message and does not need to be repeated in all PI exchange messages.

The following subsections give a brief introduction to the architecture and the operations of the Border Pricing Protocol. A more detailed description of the BPP protocol is given in [7].

3.1 Role of BPP in the Process of Sending Data

This subsection presents the operations that are necessary for sending data in case of a system supporting the Border Pricing Protocol.

1. As in a transmission without pricing, the sender chooses the destination. In addition, the sender needs to choose the QoS factors for the data flow.
2. The QoS management mechanism tests if such a route is available and indicates it. Eventually, it is already reserved.
3. The price of this route (this specific destination and the desired QoS) is requested using BPP. It works recursively, from Management Entity (ME) to ME, using the information returned from the second step.
4. If the sender accepts the returned price, it starts the transmission. On the way, a charging database increments its corresponding record.

If the sender asks "I want the cheapest route for destination X", then the price becomes a QoS factor and probably BPP will already be used in the second step. But the protocol itself is not concerned with this issue.

3.2 Spreading of Pricing Information (PI)

Pricing Information is transmitted upon request: Only Management Entities (MEs) that have previously requested a certain PI are informed of a change of it. Moreover, some age counters prevent the spreading of PI that is not needed anymore.

Two MEs set up a TCP connection between each other. They exchange messages to open and confirm the connection parameters. The system needing a PI asks it with a request message, and the peer answers with an update message. If a previously requested PI changes, the updates are sent to the systems that had requested it before. Keep-Alive messages are sent periodically. Notification messages are sent in response to errors or special conditions. If a connection encounters an error condition, a notification message is sent and the connection is closed.

3.3 Advertisement of PI

PI are advertised between a pair of BPP speakers in UPDATE messages. They are couples of a price associated to one or more destinations. The destinations are the systems whose IP addresses are reported in the Destination Information field (similar to the Network Layer Reachability Information field in BGP). But since each PI must be given for a destination address with a specific QoS, a Destination Information is an IP prefix associated with the desired QoS factor. The price is reported in the price attributes fields of the same UPDATE message.

Contrary to BGP, the path to the final destination does not need to be specified in UPDATE messages, since BPP works associated with a routing protocol and QoS mechanisms, which take care of investigating and choosing the right path to the final destination. No routing mechanisms are reproduced in BPP.

Moreover, BPP does not provide mechanisms for informing a peer that a previously advertised PI is no longer accurate (there are no WITHDRAWN ROUTES field as in BGP). The routing protocol will take care of withdrawing unfeasible routes from routing tables, and the corresponding PI will be discarded with the timers expiring. To change an already existing PI, it must be replaced by a new one. And also contrary to BGP, the closing of a BPP connection does not remove all prices, which the peers had advertised to each other.

3.4 Request of PI

PI is requested from a BPP peer using REQUEST messages, which are lists of Destination Information. Moreover, the ASs that are crossed to reach each destination must be indicated. The list of AS numbers is given by the routing mechanisms. It will be used to forward the REQUEST message to the next ME on the route if the PI is not available. In case the PI is found, an UPDATE message is sent back to the requesters.

If a price request cannot be satisfied, the system keeps the information and sends another request if possible. But the requester is not informed that the PI is not available. Not having received an UPDATE message in a given time period indicates the requester that the request could not be fulfilled.

3.5 Information Bases

A ME must maintain two different information bases: The Pricing Information Base (PIB) is aimed to store PI for a particular destination. The Charging Information Base (CIB) is used to store the charging information (the "bill") of each sender.

3.5.1 Pricing Information Base

The PIB in a BPP system consists of three distinct parts (similar to the Routing Information Base of BGP):

Updated-PIB: Stores PI that has been learned from UPDATE messages. Its content is available for price request answer and for price calculation (in the charging process). Destinations of this base are outside the AS.

Local-PIB: Contains the local PI that the system knows by an IPP or by a manual entry. The destinations of this base are inside the AS.

Requested-PIB: Stores the Destination Information for which some PI has been requested by peers, but maybe is not yet available. Peers not in this base will not be informed of a PI update.

There are three ways to set up a record in the PIB:

- *Static records* are specified manually by the administrator. This can be done to connect networks that do not support the BPP yet, to specify the PI for a destination in the AS, or for other reasons (economic, marketing considerations).
- *BPP records* are set up using the Pricing Information gathered by the BPP protocol.

- *IPP records* are set up using the Pricing Information in the AS that is known using an Interior Pricing Protocol.

Common to all parts of the PIB are the following entries:

Destination: In general a network number, in most cases an IP prefix.

Associated QoS: The format of this field is associated to the QoS model, e.g. the Differentiated Services Architecture and its service classes.

3.5.2 Charging Information Base

The records stored in the Charging Information Base depend on the used QoS model and on the tariffing model a provider chooses. At least it will entail an identifier of the Management Entity sending data into the Autonomous System and a counter representing the price units it has to pay. This might be split up into more detailed counters, measuring usage of each service class per traffic source.

3.6 BPP state machine

BPP operations are specified in terms of a Finite State Machine (FSM), which actually is quite close to the BGP state machine.

The FSM has six states, as briefly summarized here:

Idle: BPP refuses all incoming connections. It only reacts to the Start event by initializing a transport connection to another BPP peer and changes its state to Connect.

Connect: BPP is waiting for the transport protocol connection to be completed. When done, the local system sends an OPEN message to its peer, and changes its state to OpenSent. If the transport protocol connection fails, the system continues to listen for a connection that may be initiated by the remote BPP peer, and changes its state to Active state.

Active: BPP is trying to acquire a peer by initiating a transport protocol connection. If the transport protocol connection succeeds, the local system sends an OPEN message to its peer and changes its state to OpenSent.

OpenSent: BPP waits for an OPEN message from its peer. When an OPEN message is received, all fields are checked for correctness. If there are no errors, BPP sends a KEEPALIVE message and changes its state to OpenConfirm, otherwise it

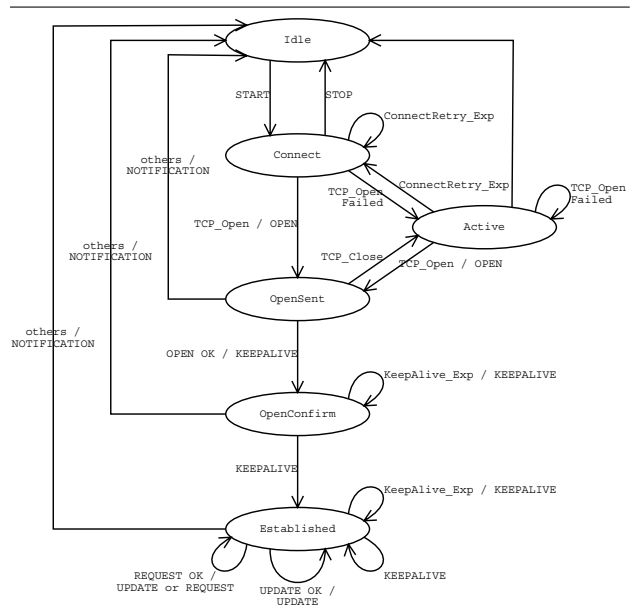


Figure 1 State machine of the Border Pricing Protocol

sends a NOTIFICATION message and changes its state to Idle.

OpenConfirm: BPP waits for a KEEPALIVE or NOTIFICATION message. If a KEEPALIVE message is received, the state is changed to Established.

Established: BPP can exchange UPDATE, REQUEST, NOTIFICATION, and KEEPALIVE messages with its peer. If the local system receives an UPDATE or a REQUEST message with no errors, the corresponding handling procedure is started, ending generally with the sending of multiple UPDATE or REQUEST messages. The state is not changed.

Figure 1 gives the complete BPP state machine.

4 Results and Conclusion

This paper is based on an open and flexible framework for charging and accounting. The framework only assumes certain entities within administrative domains that can exchange charging and accounting information but does not impose any certain pricing schemes on service providers. Similar to routing protocols the Border Pricing Protocol presented in this paper is defined to exchange price information between administrative domains, but leaves the exact definition of IPPs (Interior Pricing Protocols) to the providers. The entities handling charging and accounting can be co-located with, e.g., bandwidth brokers, as envisioned in the Differentiated Service architecture. To keep the overall overhead of exchanging pricing information low, this information can be exchanged in

the same manner as routing information. Thus it is possible to charge best effort transmissions across the Internet and not only at the access to the Internet. This is basically due to the recursive forwarding of pricing information from one provider to another following the path of data transmission.

We think that an open market will decide which pricing scheme fits which purpose best and thus we do not enforce a certain scheme but offer an open framework and a protocol for implementing and distributing pricing schemes. Currently, we are simulating the behaviour of the Border Pricing Protocol and evaluating the dynamics of exchanging Pricing Information between many Autonomous Systems. The experience gained from the simulation is progressively made available on BPP website at [7].

5 Related Work

An overview of approaches in the area of pricing is given in [8]. Many existing approaches concentrate on the network access and ignore the domain structure of the internet. A well known simple approach for pricing the network access is the Paris Metro Pricing described in [9]. A more recent approach was presented in [10]. The proposed Ressource Negotiation and Pricing Protocol RNAP has in common with BPP, that it also takes into account the domain-centered structure of the internet. The usage of the RNAP protocol does not prescribe a pricing model inside an autonomous system. It combines the exchange of pricing information with per-flow resource reservation, thus dealing with problems of scalability of many per-flow-reservation schemes.

References

- [1] Y. Rekhter, T. Li: *A Border Gateway Protocol 4 (BGP-4)*. RFC 1771. IETF, March 1995
- [2] J. Gerke, H. Ritter, J. Schiller, K. Wehrle: *Elements of an Open Framework for Pricing in the Future Internet*. In: Proc. of Conf. on Quality of future Internet Services (QofIS'2000), Berlin, Germany, September 2000
- [3] G. Huston: *Interconnection, Peering and Settlements*. In: Proc. of the INET99, San Jose, Kalifornien, 1999
- [4] G. Fankhauser and B. Stiller and C. Voegtli and B. Plattner: *Reservation-based Charging in an Integrated Services Network*. 4th INFORMS Telecommunications Conference, Boca Raton, Florida, USA, 1998
- [5] K. Nichols, V. Jacobson, L. Zhang, *A Two-bit Differentiated Services Architecture for the Internet*, RFC 2638, IETF, July 1999
- [6] R. Bless, K. Wehrle: *Management-Architecture for Enabling Multicast in Differentiated Services Networks*. Proc. of 1. GI Multicast Workshop, Braunschweig, Germany, May 1999
- [7] Oberle, V.: *Specification of the Border Pricing Protocol*. Available at: <http://vincent.oberle.com/bpp>
- [8] McKnight, L., Bailey, P. (eds.): *Internet Economics*. MIT Press, 1998
- [9] Odlyzko, A. M.: *Paris Metro Pricing fo the Internet*. In Proceedings of ACM Conference on Electronic Commerce (EC'99), 1999
- [10] Wang, X., Schulzrinne, H.: *RNAP: A Resource Negotiation and Pricing Protocol*. In: Proceedings of NOSSDAV 1999, Basking Ridge, New Jersey, 1999