

---

# BPP: A Protocol for Exchanging Pricing Information between Autonomous Systems

Vincent Oberlé

*June 13, 2001*



# Outline

---

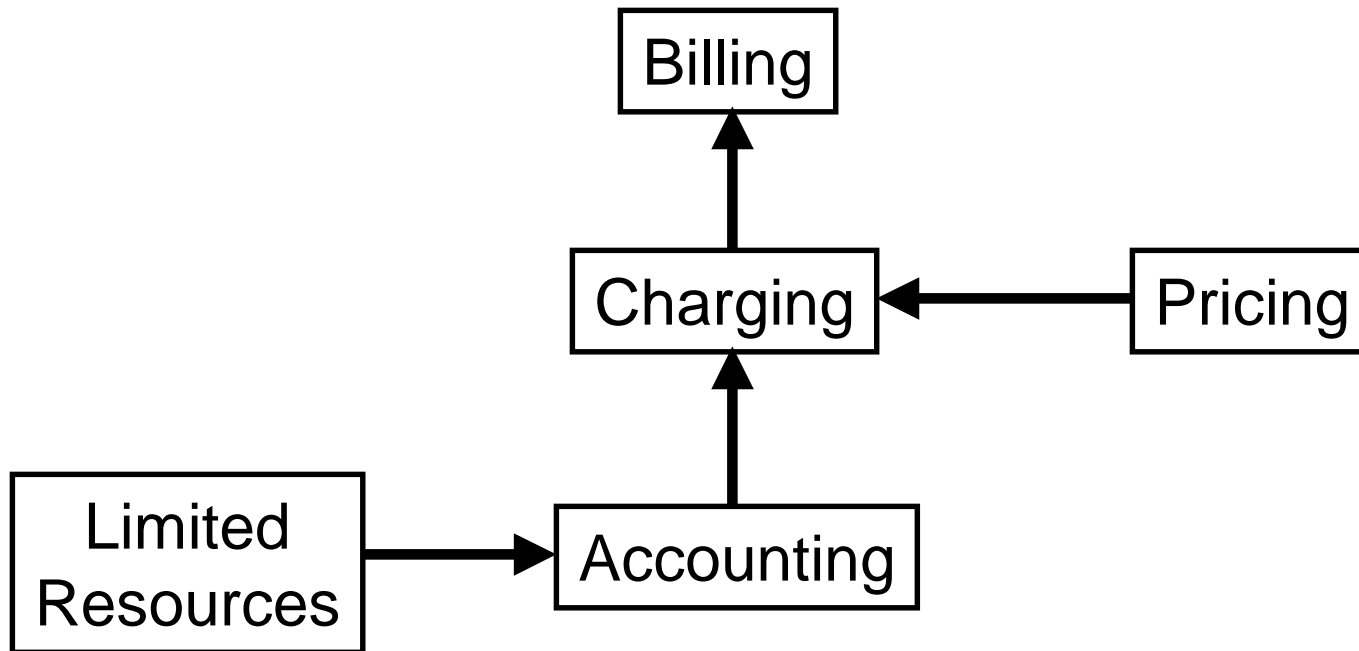
- Motivations and Requirements
- Principles of Pricing
- Border Pricing Protocol (BPP)
  - Principles of BPP
  - Role of BPP in the Process of Sending Data
  - Spreading of Pricing Information
  - Information Bases
- Simulations
- Conclusions



# Motivation and Definition of Pricing

---

- Need of financial incentives in QoS networks
- Pricing = *process of setting a price on a service, on a product or on a content*



# Requirements of Pricing

---

End-to-end pricing framework

Requirements

- Flexibility, regarding:
  - QoS architecture
  - Bandwidth
  - Different pricing schemes
- Low overhead
- Migration
- Functionality



# Principles of Pricing

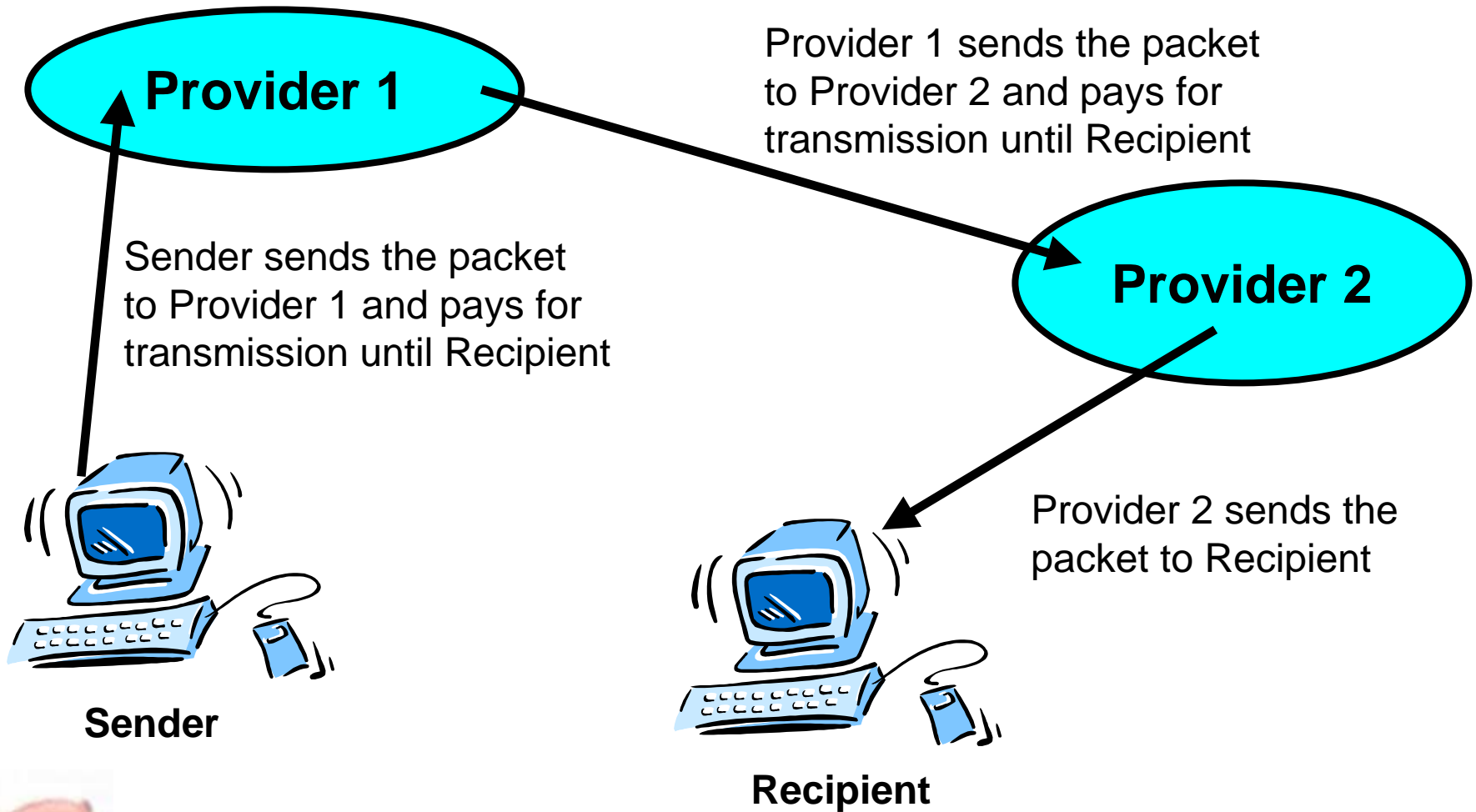
---

- Internet: Structured in blocks:  
    Autonomous Systems (AS)  
    Relations between AS = Services
- Data transmission = Recursive service usage
- Problem of direction: Who pays?
  - ⇒ The Sender always pays (// Postal service)
  - ⇒ Recursive payment



# Recursive service usage / payment

---



# Pricing Information: BPP

---

- Need to know in advance cost of a transmission
- For services without explicit connection setup, **Pricing Information (PI) must be distributed** throughout the Internet
- Adapting methods for distributing routing information to the distribution of PI
- Proposition: the

## **Border Pricing Protocol**

in analogy to the routing protocol BGP



# BPP: Principles

---

- No global deployment necessary: Providers not supporting BPP do the financial settlement as before
- Pricing **not defined inside a domain** (Interior Pricing Protocols - IPP)
- Need of a **Management Entity (ME)** inside an AS
  - Bandwidth Broker
  - Domain Manager

⇒ Augmented with BPP capabilities



# BPP: Principles (continued)

---

- PI transmitted upon request
- Intermediate ME keep PI (caching mechanism)
- Designed to run with **TCP**: not concerned about correct reception of traffic, segmentation, etc
- Manage sessions (**Open**, **Keep-Alive**, **Close** messages)
  - ⇒ Common information put in Open message



# Role of BPP in the Process of Sending Data

---

- ① Sender chooses the Destination (Address) and the wished Services (QoS factors)
- ② Routing + QoS architecture indicate availability of such a route (*facultative*)
- ③ The price for this route is requested using BPP, recursively from ME to ME (*with information from ②*)
- ④ If the sender accepts the returned price, the transmission can start. On the way, charging databases are updated



# Spreading of Pricing Information

---

BPP session:

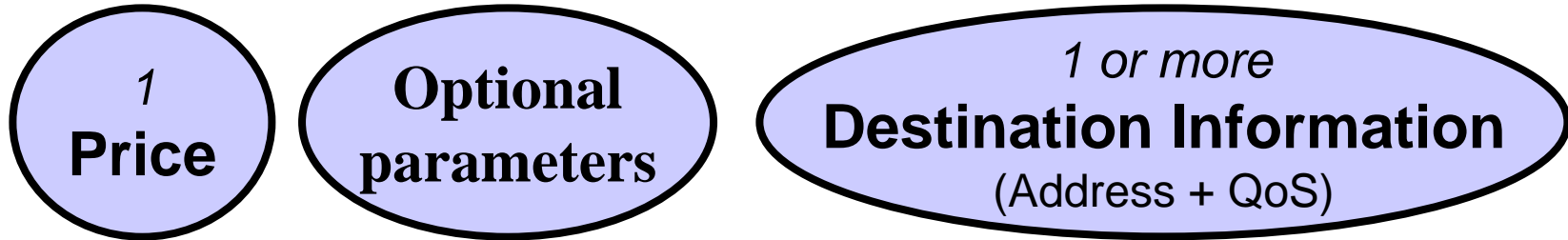
- 2 MEs set up a TCP connection
- Exchange Open messages to initialize and confirm connection parameters
- System needing a PI asks with a **Request** message
- The peer answers with an **Update** message
- If a previously requested PI changes, updates are sent only to systems that had requested it before
- **Keep-Alive**, **Notification** messages



# Advertisement of PI

---

Price are advertised in **Update messages**:



≠ BGP:

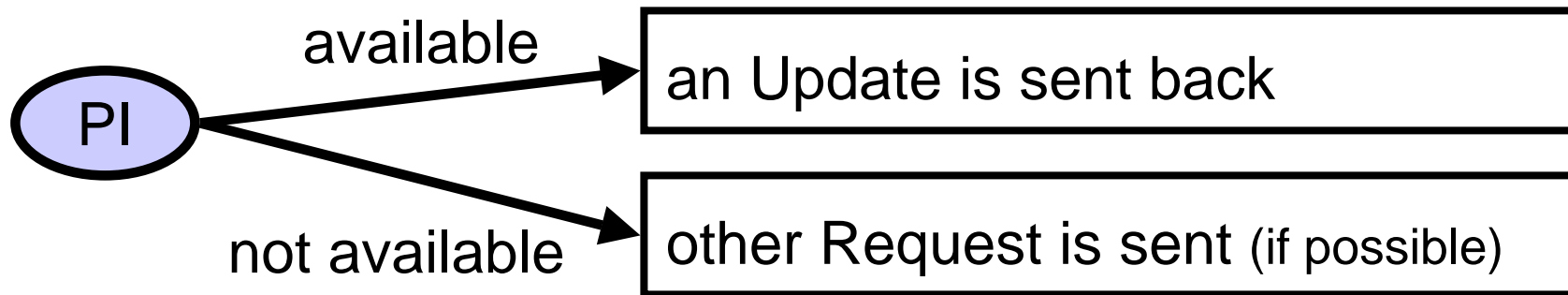
- Path to the final destination not given (requester addresses are known, uses routing mechanisms)
- No mechanism to indicate a PI is no longer accurate
  - routing protocol withdraws unfeasible routes
  - age counters (PI time-to-live)



# Request of PI

---

Price are requested using **Request message**:



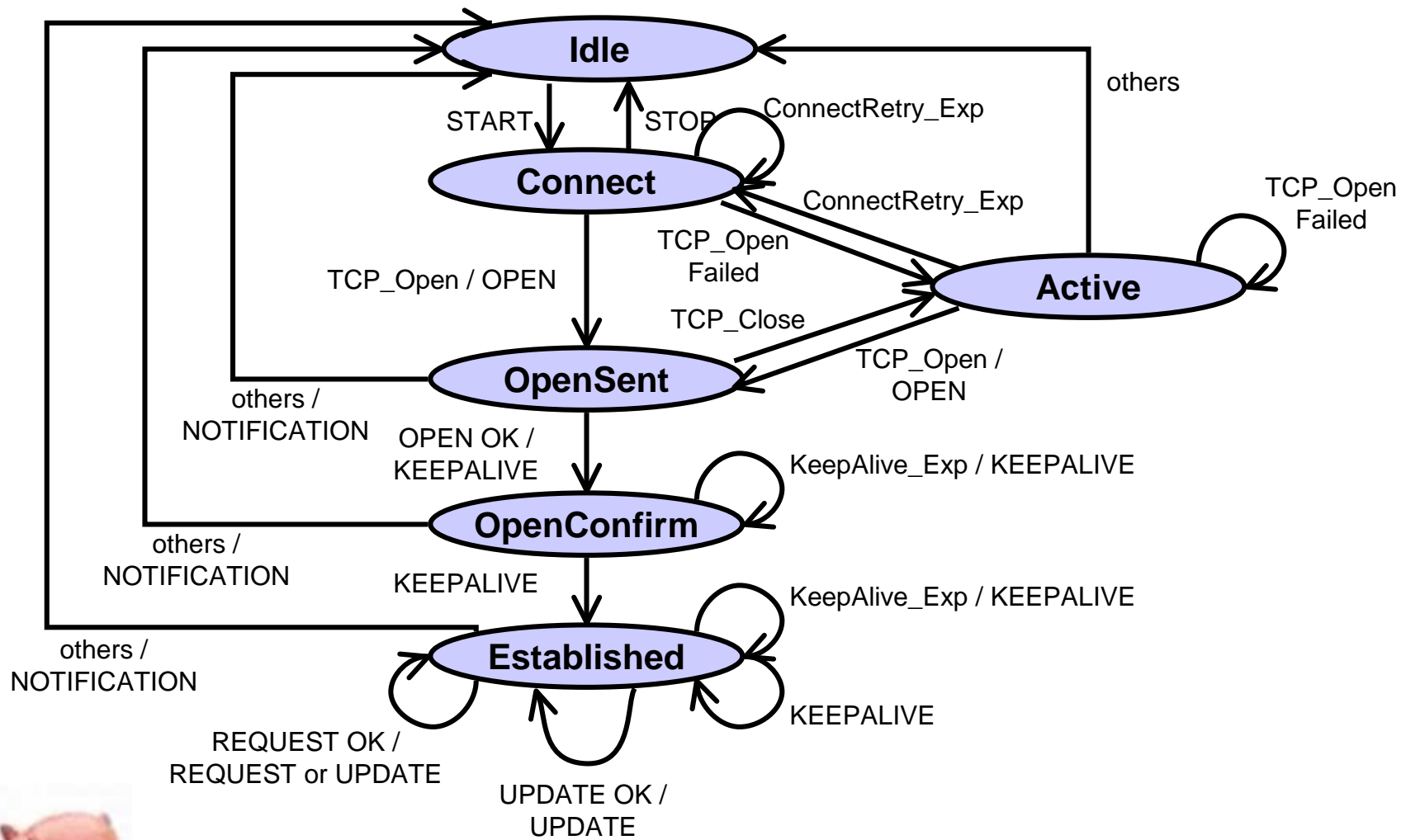
# Information Bases

---

- Pricing Information Base (PIB)
  - Stores PI for a particular destination
  - **Updated-PIB**: PI that have been learned from Update messages
  - **Local-PIB**: Local PI, known from an IPP or a manual entry
  - **Requested-PIB**: Destination for which some PI has been requested
- Charging Information Base
  - QoS and tariffing models specific



# Finite State Machine



# Simulation model

---

Simulation developed using:

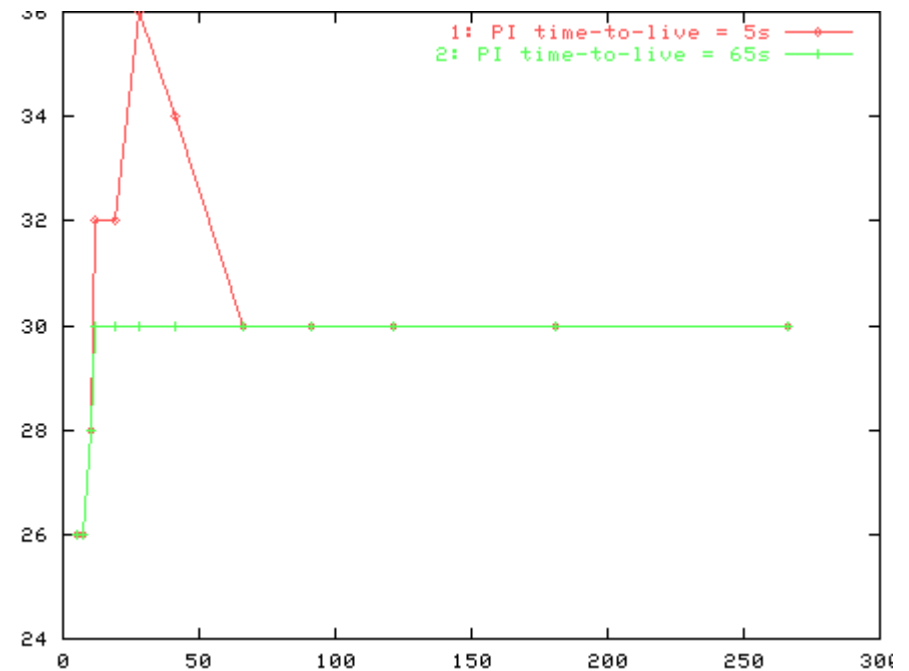
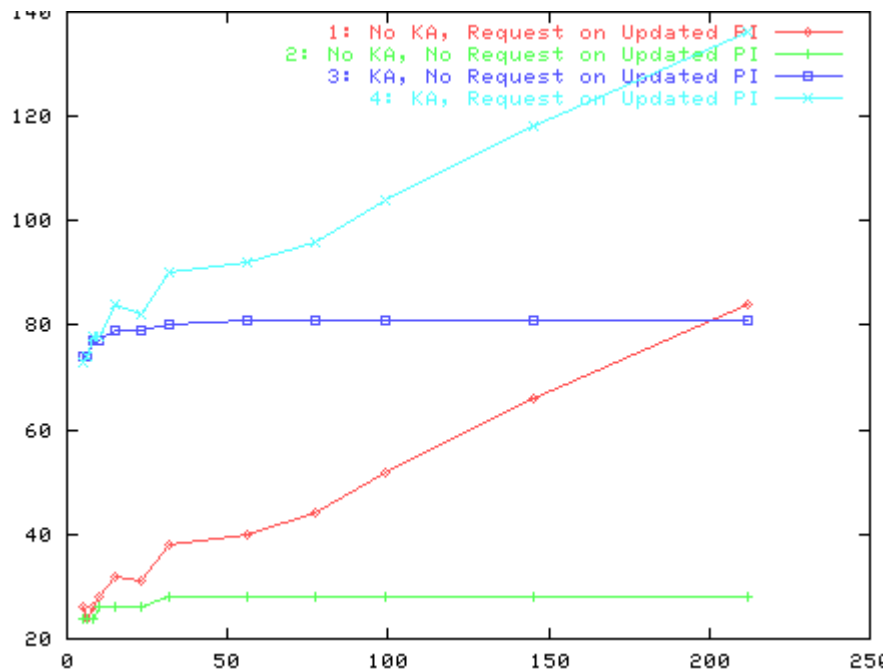
- **OMNeT++** 2.0 (from András Varga)  
Object-oriented modular discrete event simulator,  
programmed using C++
- **IP-Suite**: TCP/IP protocols suite for OMNeT++  
Developed by the University of Karlsruhe



# Basic Simulations

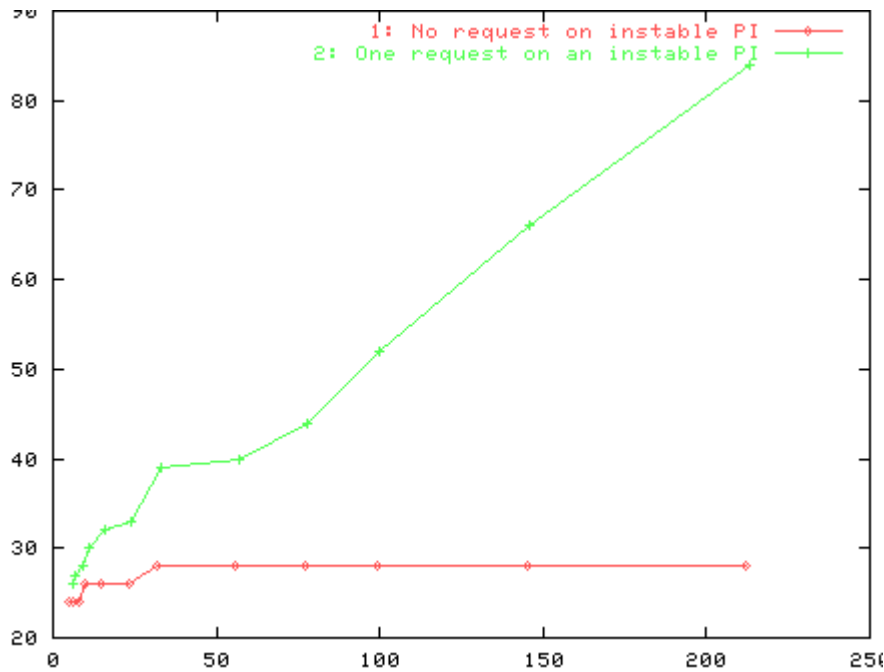
Simple example of request and caching effect

Effect of the **PI time-to-live**

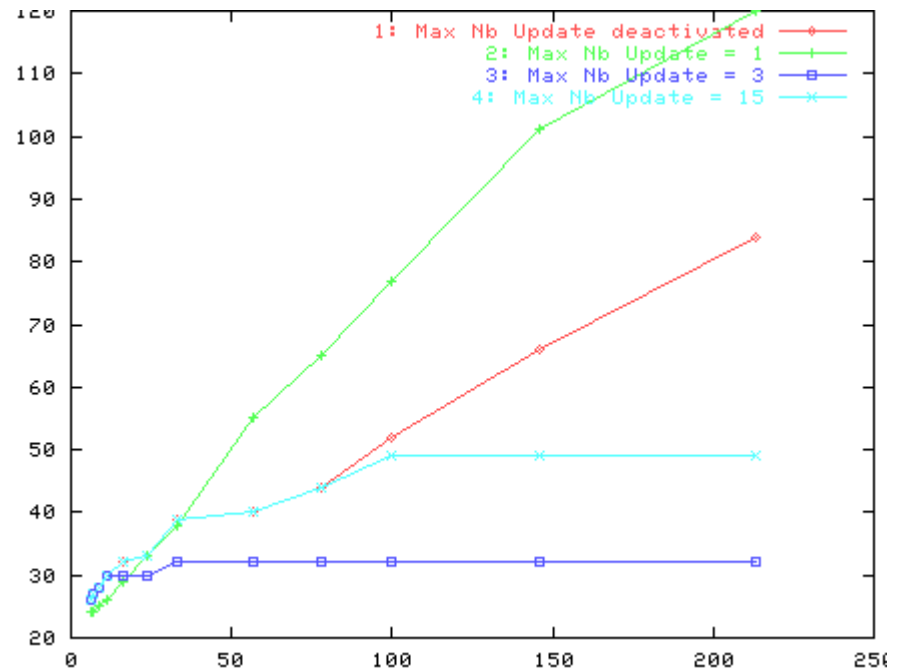


# PI instability problem

Highlighting of the PI instability problem



Correction of PI instability problem: Open message parameter Max Nb Update



# Conclusions

---

- Open framework and a protocol for distributing pricing schemes
- BPP
  - Similar to BGP routing protocol
  - PI transmitted upon request
- Simulation results
  - Request-based protocol, Caching limit the traffic to satisfying values



# Conclusions (continued)

---

- Future work
  - Better address aggregation - Scalability
  - Simulate big and complex networks
  - With some real traffic - DiffServ network
  - Pricing directly in BGP (HPSR feedback)
- <http://vincent.oberle.com/bpp.html>

*Gracias*

