

Choosing Reputable Servents in a P2P Network

F. Cornelli E. Damiani S. De Capitani di Vimercati S. Paraboschi P. Samarati

fcornelli@crema.unimi.it {damiani,samarati}@dti.unimi.it decapita@ing.unibs.it parabosc@elet.polimi.it
<http://seclab.dti.unimi.it>.

DTI-Università di Milano DEA-Università di Brescia DEI-Politecnico di Milano

Outline

- Motivation
- P2P basics
- P2P issues
- Our approach (**P2PRep** protocol)
 - Basic polling
 - Enhanced polling protocol
- Implementation of the P2PRep protocol
- Conclusions and future work

Motivation

- P2P solutions are currently receiving considerable interest
- Most P2P systems protect peers' anonymity
- Anonymity opens the door to possible misuses and abuses
 - malicious users exploit the P2P network to distribute Trojan Horses and viruses

Motivation

- P2P solutions are currently receiving considerable interest
- Most P2P systems protect peers' anonymity
- Anonymity opens the door to possible misuses and abuses
 - malicious users exploit the P2P network to distribute Trojan Horses and viruses

Peer review process: the peers' opinions is used to establish a reputation for peers

Peer to Peer Basics (1)

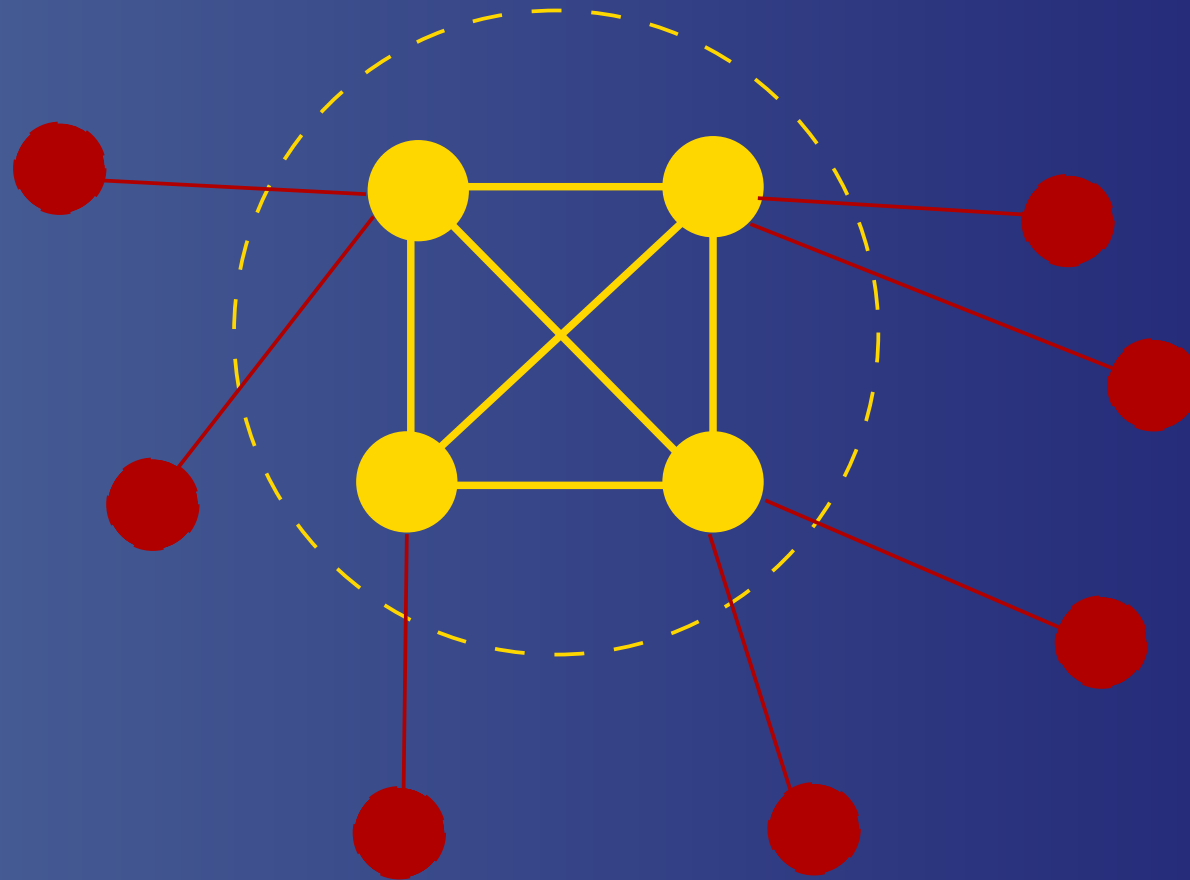
- All the nodes offer the same services and follow the same behavior (**server + client**)
- P2P networks for file exchange involves two phases:
 1. **Search** of the servers where the requested information resides
 2. **Download** from the exporting servers the requested information

Peer to Peer Basics (2)

- Centralized indexes (Napster)

Peer to Peer Basics (2)

Centralized Indexes

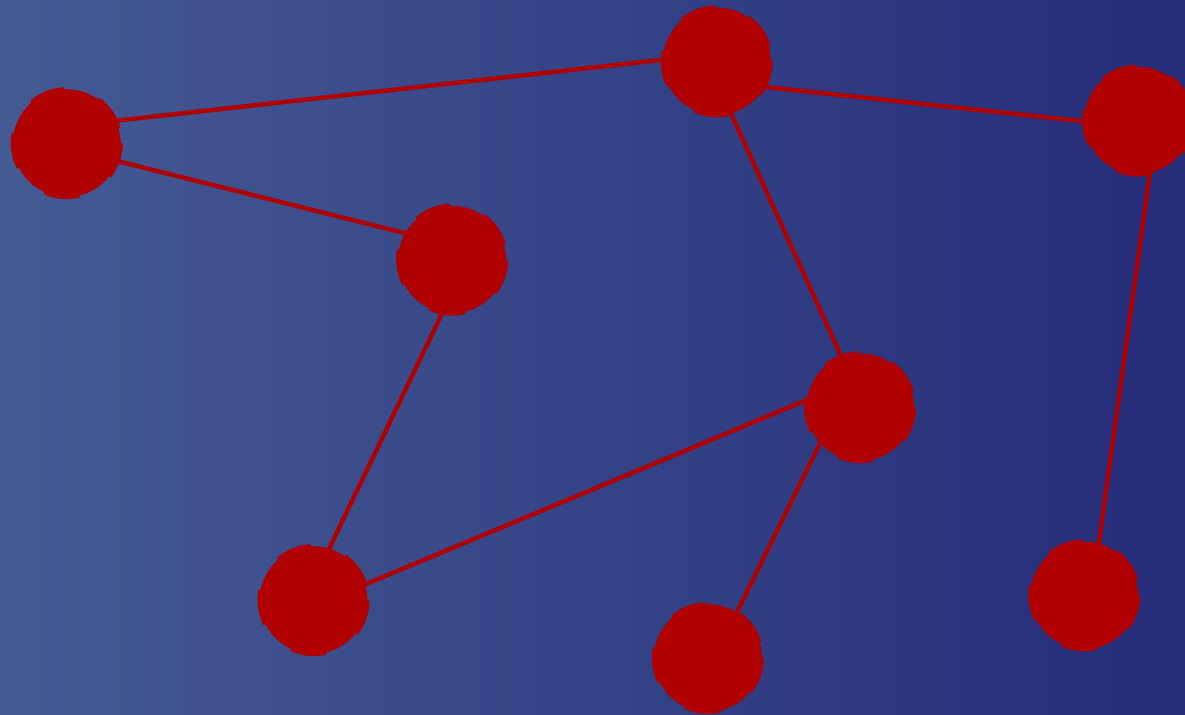


Peer to Peer Basics (2)

- Centralized indexes (Napster)
- Pure P2P architecture (Gnutella)

Peer to Peer Basics (2)

Pure P2P Architecture

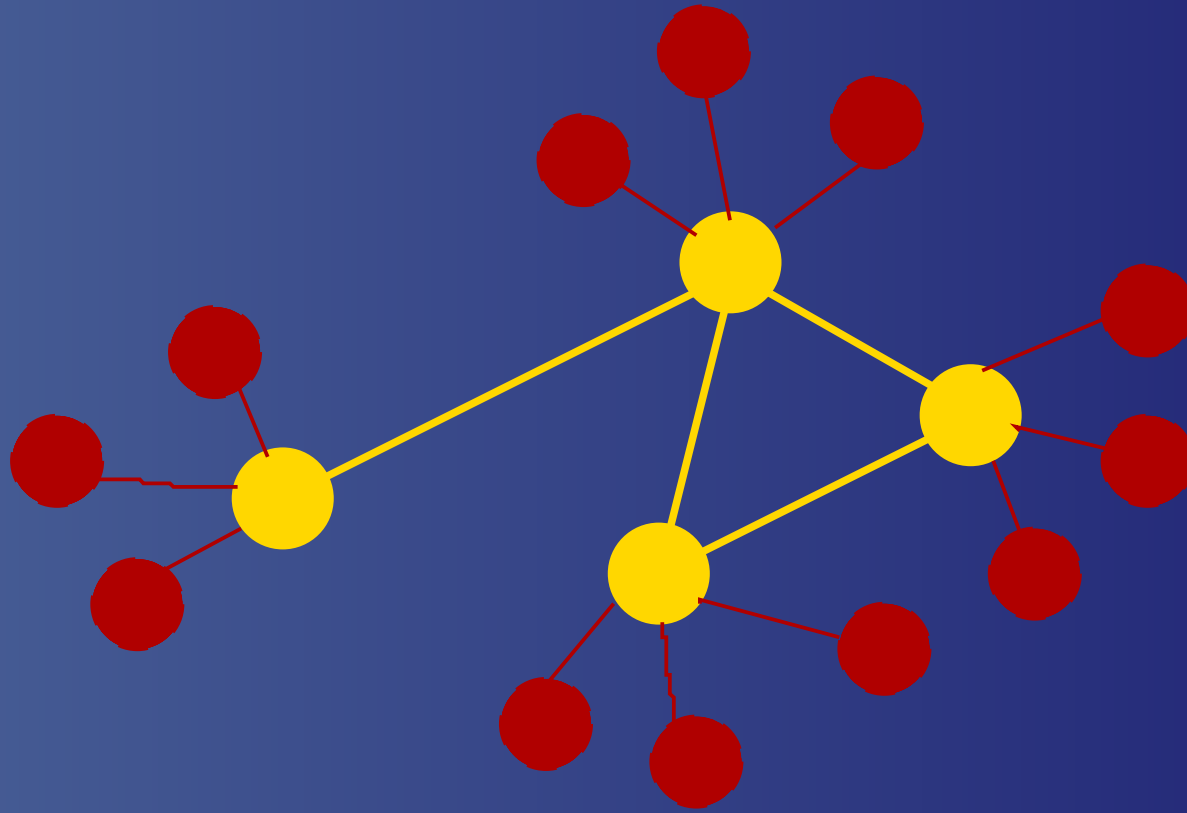


Peer to Peer Basics (2)

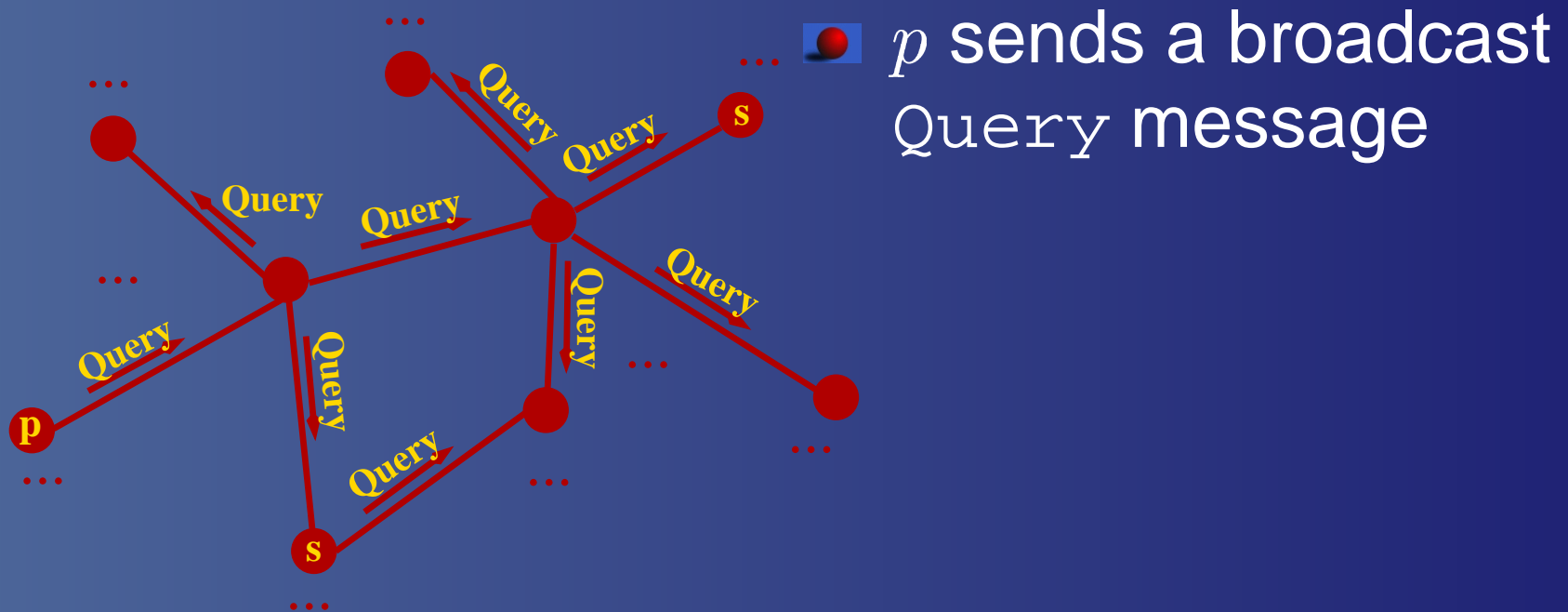
- Centralized indexes (Napster)
- Pure P2P architecture (Gnutella)
- Intermediate solutions (Fasttrack)


Peer to Peer Basics (2)

Intermediate solutions





The Gnutella Architecture

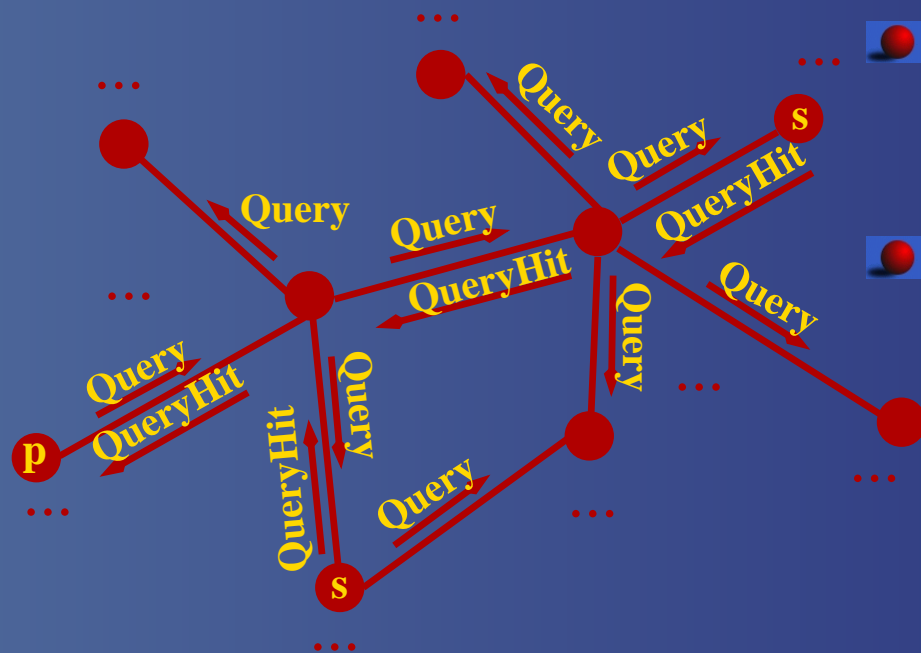


 p sends a broadcast Query message

Legend

-  p servent looking for a resource
-  s servents willing to offer the requested resource

The Gnutella Architecture



■ *p* sends a broadcast Query message

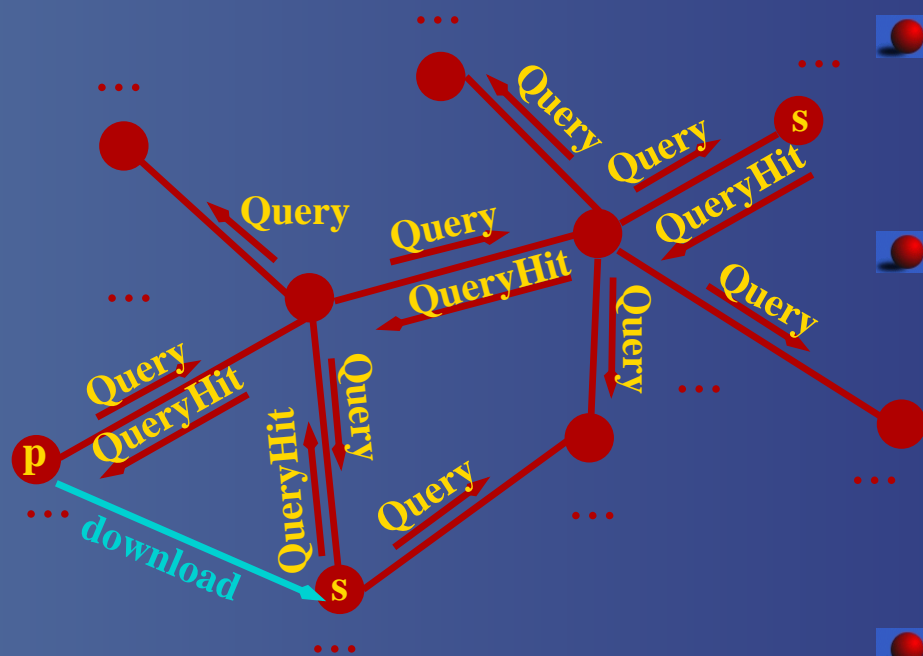
■ servers that have the requested file, answer with a QueryHit message

Legend

■ *p* servent looking for a resource

■ *s* servents willing to offer the requested resource

The Gnutella Architecture



Legend

- p** servent looking for a resource
- s** servents willing to offer the requested resource

- p** sends a broadcast Query message
- servents that have the requested file, answer with a QueryHit message
- p** selects a servent from which downloads the file

P2P Issues

- **Anonymous peer-to-peer communications:** under the anonymity, malicious peers can answer to virtually any query providing tampered-with information
- **Genericity of the shared information:** anyone can attach to the network and provides malicious content tailored to specific requests with relatively small chance of detection

The P2PRep Proposal (1)

- Each node keeps track and share with others information about the reputation of their peers
- Reputation sharing is based on a **distributed polling protocol**
- Before initiating the download, requestors can assess the reliability of sources by polling peers
- The protocol is easy to implement and to integrate into existing environments

The P2PRep Proposal (2)

- Each servent has a **servent_id** which is a digest of a public key obtained using a hash one-way function
- **Votes** are values expressing opinions on other peers
- **Servent reputation** represents the “trustworthiness” of a servent in providing files
- **Servent credibility** represents the “trustworthiness” of a servent in providing votes

Basic Polling

Phase 1: Resource searching. p sends a `Query` message for searching resources, and servers matching the request respond with a `QueryHit`

Initiator p

Servers S

`Query(search_string,min_speed)`



`QueryHit(num_hit,IP,port,speed,Result,server_idi)`



Basic Polling

Phase 2: Vote polling. p polls its peers about the reputation of a top list T of servers, and peers wishing to respond send back a `PollReply`

Initiator p

Servers S

`Poll (T, PKpoll)`



`PollReply ({(IP,port,Votes)} PKpoll)`

Basic Polling

Phase 3: Voter evaluation. p selects a set of voters, contacts them directly, and expects back a confirmation message

Initiator p

Servents S

TrueVote ($Votes_j$)



TrueVoteReply (response)



Basic Polling

Phase 4: Resource download. p selects a server s from which download the resource and starts a challenge-response phase before downloading

Initiator p

Servers s

$\text{challenge}(r)$

$\text{response}([r]_{sK_s}, PK_s)$

Enhanced Polling

Phase 1: Resource searching. p sends a `Query` message for searching resources, and servers matching the request respond with a `QueryHit`

Initiator p

Servers S

`Query(search_string,min_speed)`



`QueryHit(num_hit,IP,port,speed,Result,server_idi)`



Enhanced Polling

Phase 2: Vote polling. p polls its peers about the reputation of a top list T of servents, and peers wishing to respond send back a `POLLReply`

Initiator p

Servents S

`POLL (T, PKpoll)`

`POLLReply ({[(IP, port, Votes, servent_idi)]SKi, PKi })PKpoll)`

Enhanced Polling

Phase 3: Voter evaluation. p selects a set of voters, contacts them directly to avoid `servent_id` to declare fake IPs

Initiator p

Servents S

`AreYou (servent_idj)`



`AreYouReply (response)`



Enhanced Polling

Phase 4: Resource download. p selects a server s from which download the resource and starts a challenge-response phase before downloading

Initiator p

Servers s

challenge(r)



response($[r]_{SK_s}, PK_s$)



Reputations and Credibilities Storage

- **experience_repository**: set of triples (*servent_id, num_plus, num_minus*)
- **credibility_repository**: set of triples (*servent_id, num_agree, num_disagree*)
- **translation of reputations into votes**: votes are expressed on the basis of information available in the *experience_repository*
 - peers votes positively only for servents with which it never had bad experiences

Removing Suspects from the Poll

- Malicious peers can create or forge a set of peers with the purpose of sending positive votes to enhanced their reputations
- **Suspects identification** procedure: computing cluster of voters whose characteristics suggest that they may have been created by a single malicious user

P2PRep Impact

- Encourage servents to keep a persistent servent id across transactions
- Act as an adaptive selection mechanism of reliable information providers within a given horizon
- Message exchanges can be reduced by providing a **server-based** functionality:
 - servents keep a record of (positive) votes for them stated by others (**credentials**) that must be signed

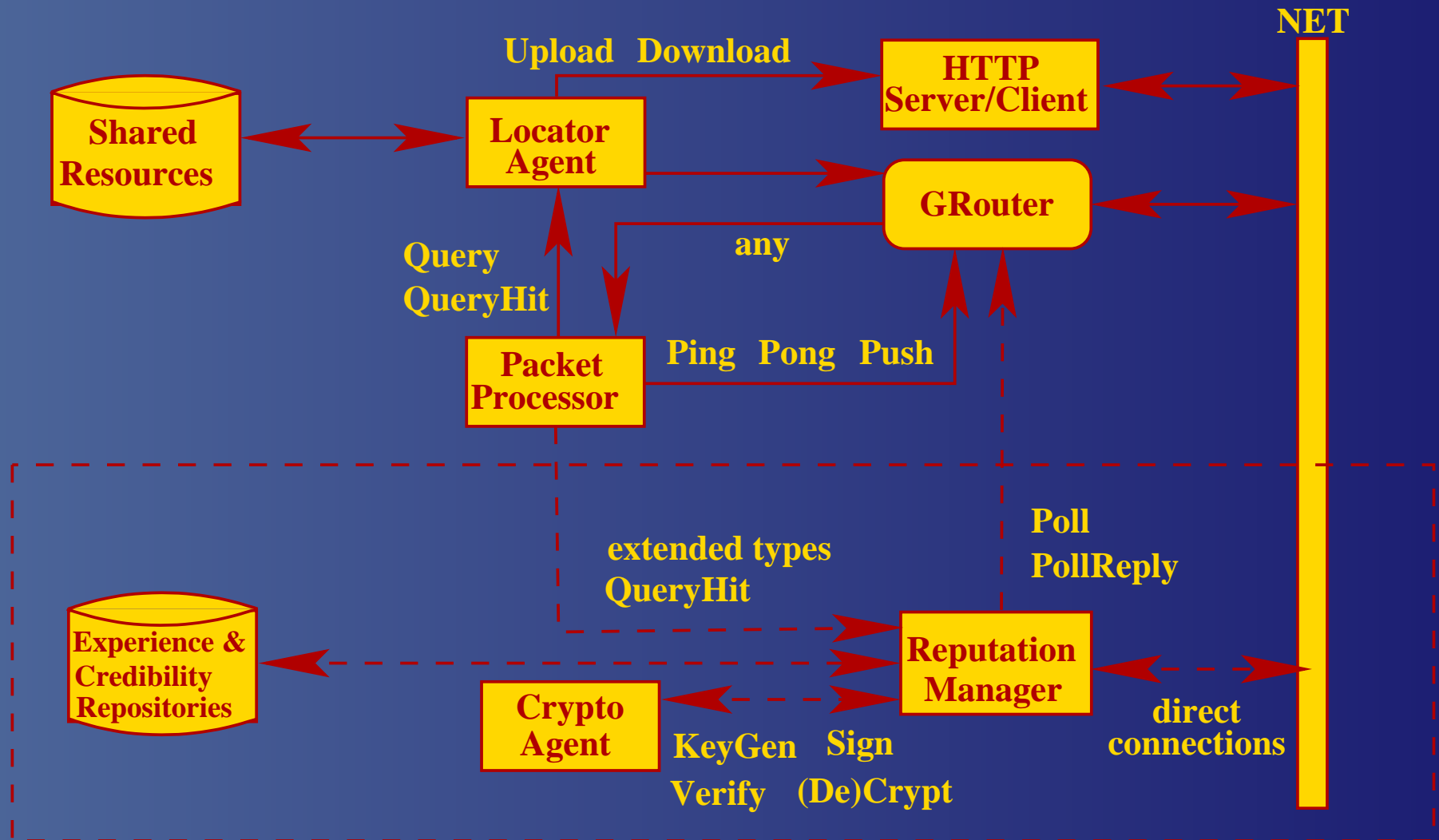
Security Improvements

- ❑ Malicious servers **cannot modify** the votes in transit because encrypted
- ❑ Servers **will not be able to selectively discard** votes (the recipient is not known and the content is not visible)
- ❑ Prevent distribution of tampered-with information
- ❑ The challenge-response mechanism **avoids impersonation**

P2PRep Implementation

- To keep the impact of our protocol extension to minimum, we use a *piggyback* technique:
 - P2PRep messages are carried as payload inside `Query` and `QueryHit` messages
- Gnutella-like architecture is complemented with three additional components:
 - Reputation Manager
 - Experience and Credibility Repositories
 - CryptoAgent

P2PRep Implementation



Comments and Discussion (1)

- **Limited cost:** the additional cost (storage capacity and bandwidth) is limited
- **Concentration of servers:**
 - the number of peers reachable by poll requests (and queries) is limited
 - good (or bad) reputation is supported only if servers have many votes
 - small number of offerers and greater number of free riders implies that servers exhibit an adequate reputation

Comments and Discussion (2)

■ Overload avoidance:

- a digest is associated with each file
- a reputable offerer is identified
- download from any peers exporting the resource with the same digest of the reputable offerer

■ Integration with intermediate P2P solutions: our reputation mechanism can take advantage from **ultrapeers**

Conclusions

- We described a reputation management protocol for anonymous P2P environments
- First step towards the development of a **self-regulating system** for preventing malicious behavior in P2P
- Future work:
 - resource-based reputation
 - reputation mechanism with ultrapeers