Trust and Reputation in Peer-to-Peer Systems

Leonardo Leiria Fernandes
PhD Student
University of Trento
Research Methodology Course - 2007

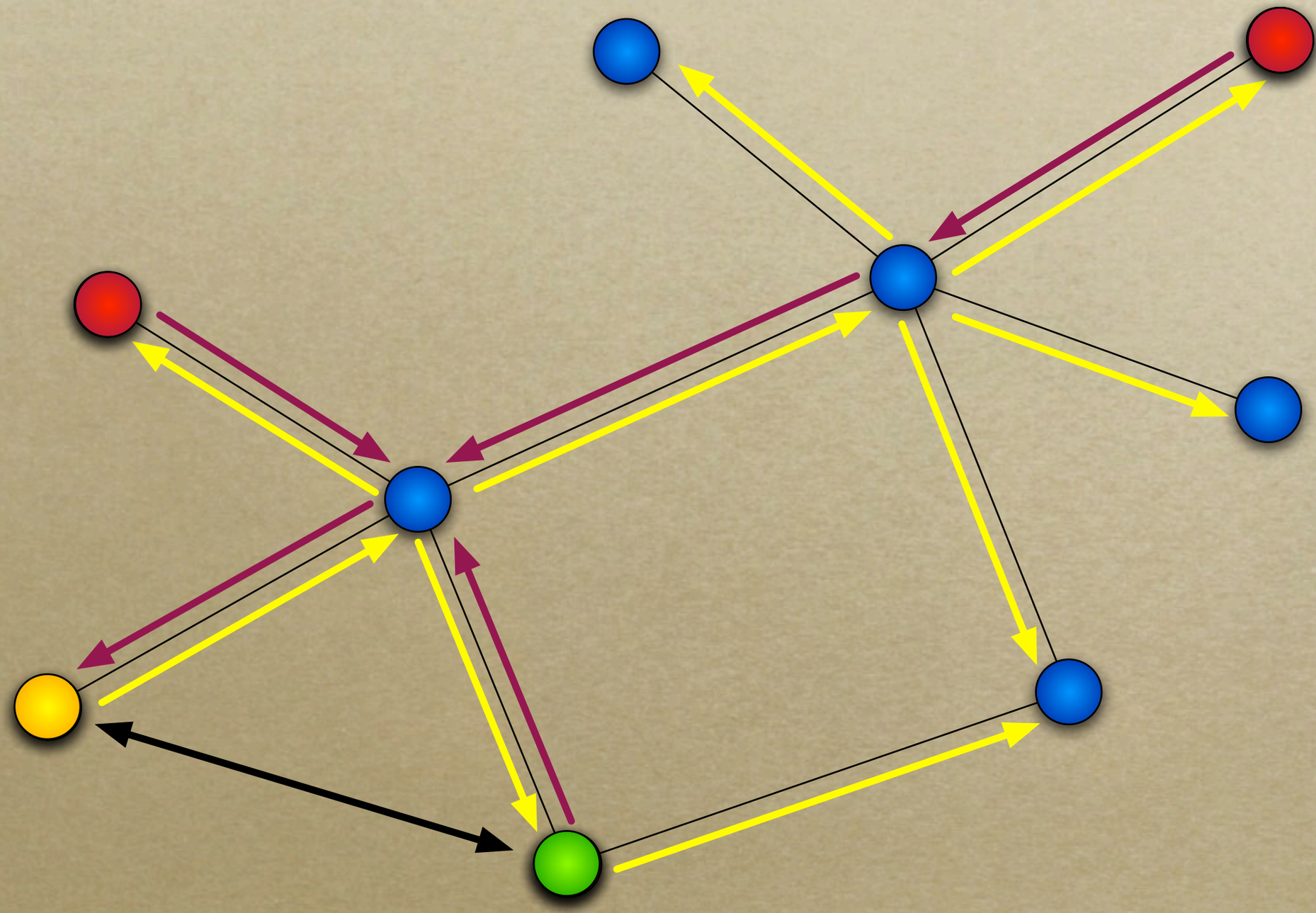
Contents

- *Motivation*
- *P2PRep: Providing Trust For Gnutella*
- *EigenTrust*
- *TrustMe*
- *Dealing With Free-Riders (BitTorrent)*
- *Conclusions*

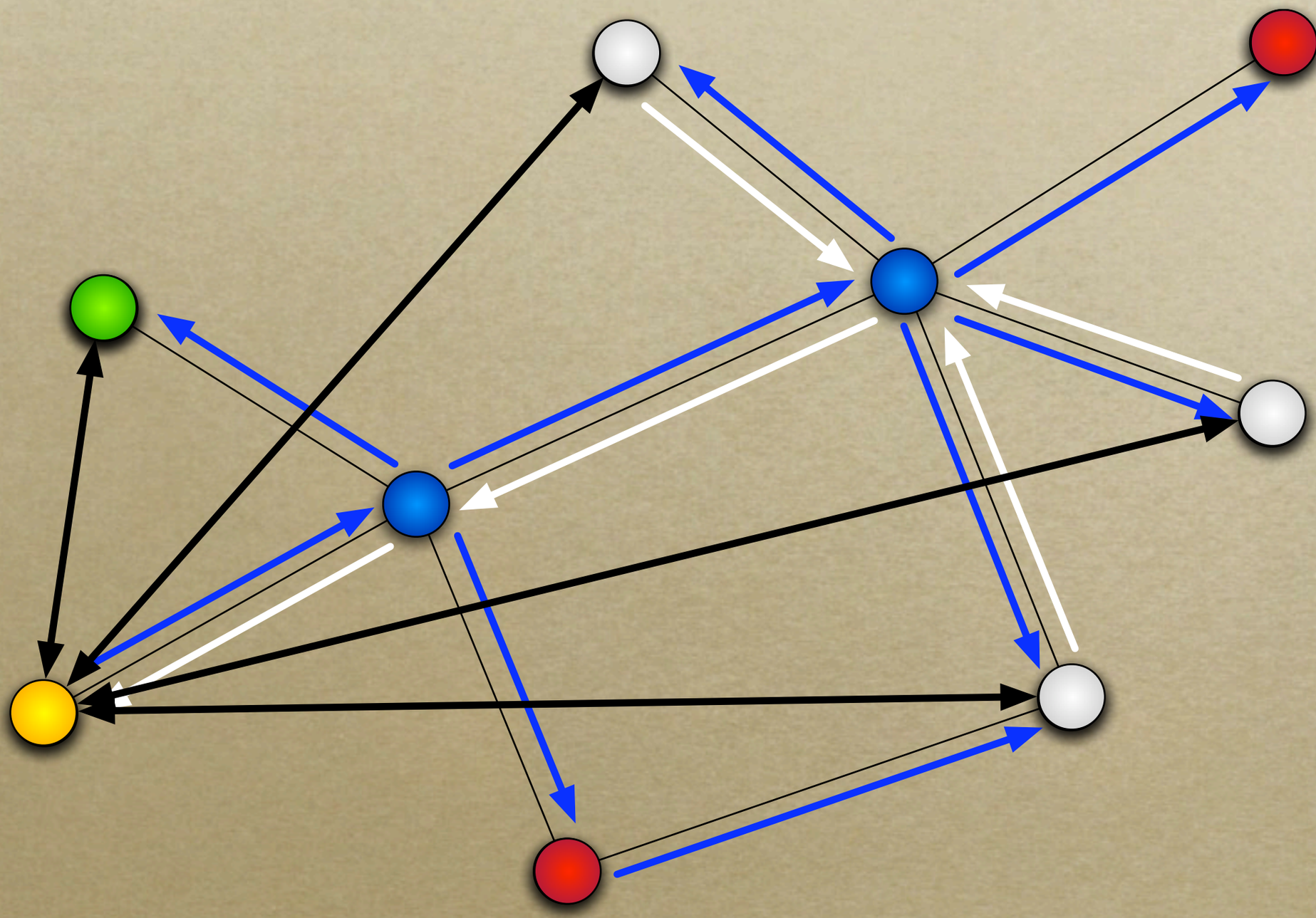
Motivation

- *Trusting Thousands of Anonymous Peers*
- *Trusted Peers Required in Many Protocols (i.e. Peer Sampling Service)*
- *No Trusted Central Authority*
- *"Free-Riders" Overhead*

Gnutella vs P2PRep



Gnutella vs P2PRep



P2PRep Discussion

- *Extends a Real-World Deployed P2P System*
- *Nodes Only Need Information About Own Previous Experience*
- *Vulnerable to Some Collective Attacks*

Eigentrust

- *Global Reputation for Each Peer Obtained By Calculating the Left Principal Eigenvector of a Matrix of Normalized Local Trust Values*

Basic EigenTrust

$$s_{ij} = sat(i,j) - unsat(i,j)$$

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum_j max(s_{ij}, 0)}$$

$$t_{ik} = \sum_j c_{ij} c_{jk}$$

$$\overrightarrow{t_i} = C^T \overrightarrow{c_i}$$

$$\overrightarrow{t} = (C^T)^n \overrightarrow{c}$$

- *Vector $\overrightarrow{t}$ (left principal eigenvector of C)represents a global vector with the trust value of all peers*
- *Basic Eigentrust Assumes Matrix C is Known*

Distributed and Secure EigenTrust

- *Distributed EigenTrust: Each Node i Calculates Its Own Global Trust Value By Collecting c_{ki} Of Every Node k*
- *Expensive Process Since In Each of the n Calculation Interactions, partial values for c_{ki} Need to Be Collected*
- *Secure EigenTrust: Global Trust Value for Node i is Calculated and Kept By Other Peers Instead of By Itself*

EigenTrust Discussion

- *Ensures Anonymity*
- *Algorithm Converges Fast, i.e. 10 Iterations for 1000x1000 Matrix*
- *Although The Solution is Totally Distributed, Some Initially Trusted Peers Are Needed For Security Against Collective Attacks*

TrustMe

- *2 Public-Private Key Pairs for Each Peer*
- *Bootstrap Server (BS) Needed*
- *Set of Trust Holding Agents (THA's) Assigned to Each Peer By the BS By Randomly Choosing Nodes*
- *Special Public-Private Keys (SPi,SBi) Generated for Each Peer i By BS And Distributed to Respective THA's*

TrustMe In a Nutshell

- ***Query:*** *Peer j Broadcasts Trust Query for Peer "i"*
- ***Reply:*** *All THA's of "i" Reply With Trust Value For i Encrypted With SP(i)*
- ***Interaction:*** *j Interacts With i and Collect Proof of Interaction*
- ***Report:*** *j Reports The Result of The Interaction Encrypted Using SB(i)*

TrustMe Discussion

- *Ensures Anonymity Through The Use Of Opaque ID's*
- *Requires A Trusted BootStrap Server*
- *Protocol Cost Is Roughly of Two Broadcasts To The Whole Network*
- *Random Choice of THA's Provide Good Security Against Attacks*

Dealing With “Free-Riders”

- *Requesters Reputation is Considered*
- *On BitTorrent Nodes Upload Mostly to Peers That Provide them The Best Download Rate in Exchange (Tit-for-Tat)*
- *For New Nodes to Startup, BitTorrent Peers Keep “Optimistic” Upload*
- *No Persistent Information is Kept*

Conclusions

- *We Have Shown It Is Possible To Implement A Reasonable Level of Trust in P2P Systems, Assuming Most Nodes Cooperate, In Real World Applications*
- *However, All Studied Solutions Still Rely on Some Kind of Central Authority or Initially Trusted Peers. Due to This Fact We Believe That the problem of Trust in "Pure" P2P Systems is Still an Open Issue*

Thanks!

Leonardo L. Fernandes
http://dit.unitn.it/~fernand/

TrustMe in Detail

$$Node_i Keys : P_i, B_i, P_i', B_i'$$

$$BSKeys : P_{BS}, B_{BS}, SP_i, SB_i \forall i$$

$$BID_i = P_{BS}(\text{"}ValidNode\text{"}|B_i')$$

$$Query : Q(j, \{i_1, i_2, i_3, ...\} = ID_{i_1}|ID_{i_2}|ID_{i_3}...$$

$$Reply : R(x, i) = ID_i|B_i|SB_i|SP_i(TV|TS|BID_x|P_x'(TS))$$

$$Report : ID_i|SB_i(\text{"}Report\text{"}|V|B_j|P_j(P_i(TS|B_j|ID_j)))$$

Motivation

"The term Peer-to-Peer is a generic label assigned to network architectures where all the nodes offer the same services and follow the same behavior"