REPS: Scalable Reputation Management for P2P MMOGs

Guan-Yu Huang, Shun-Yun Hu, Jehn-Ruey Jiang
National Central University, Taiwan

MMVE'08, Reno, USA, 2008

2008/03/08

Outline

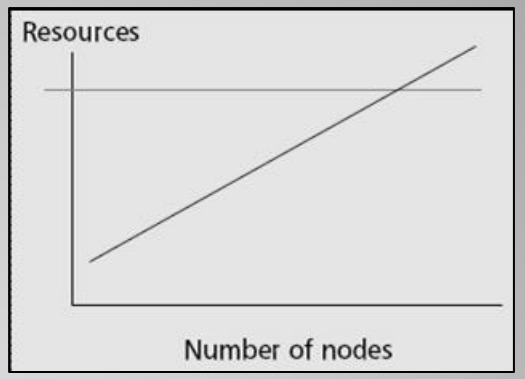
- Introduction
- Problem formulation and challenges
- Design of REPS
- Discussions
- Conclusion

- Booming of Virtual Environments (VEs)
  - Massively Multiplayer Online Games (MMOGs)

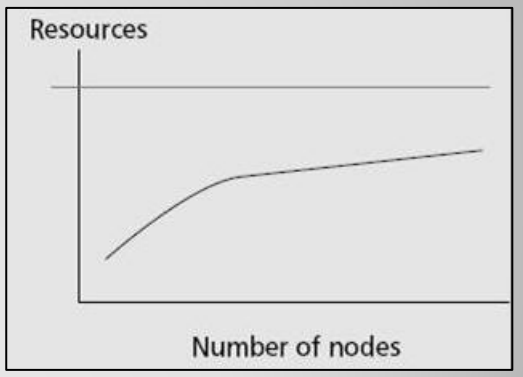
Virtual Environments

P2P-based MMOGs

- Architectures:
  - Client-server
    - All loads are on the server



  - Peer-to-Peer (P2P)
    - Distribute loads to all users
    - More scalable & affordable
    - Based on locality of interactions

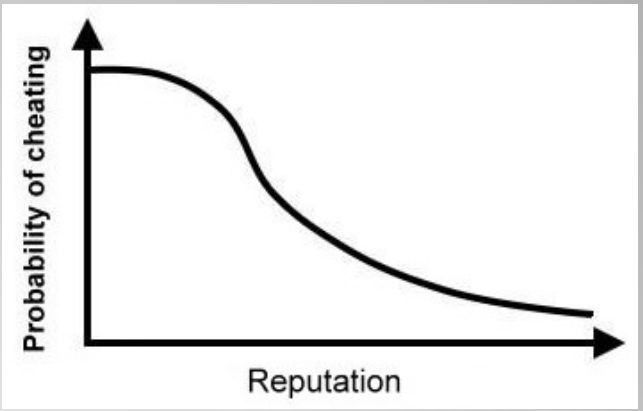


Problems in P2P-MMOGs

- MMOG requirements
  - Consistency
  - Persistency
  - Security
  - Responsiveness
  - Reliability
  - Scalability
- Security issues
  - Data modification
  - Game rules mis-processing

- **Reputation may identify trustworthy users**

- Reputation is feasible
  - MMOGs are socially-oriented
  - Players value in-game status



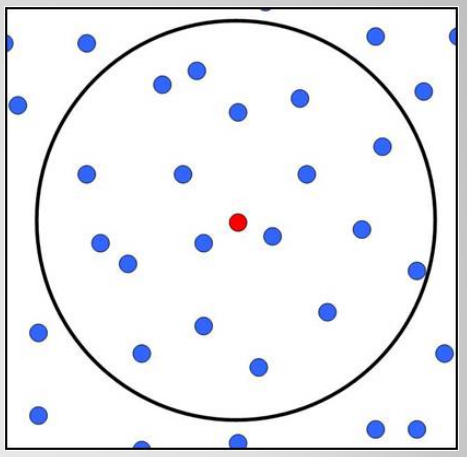
- Reputation is useful
  - To decide whether to interact
  - To delegate responsibilities

- **REPS** for P2P MMOGs
  - Localized trust evaluation with *rating right*
  - Selection of *trust nodes* to store & query reputations

Motivation & Proposed method

Problem formulation

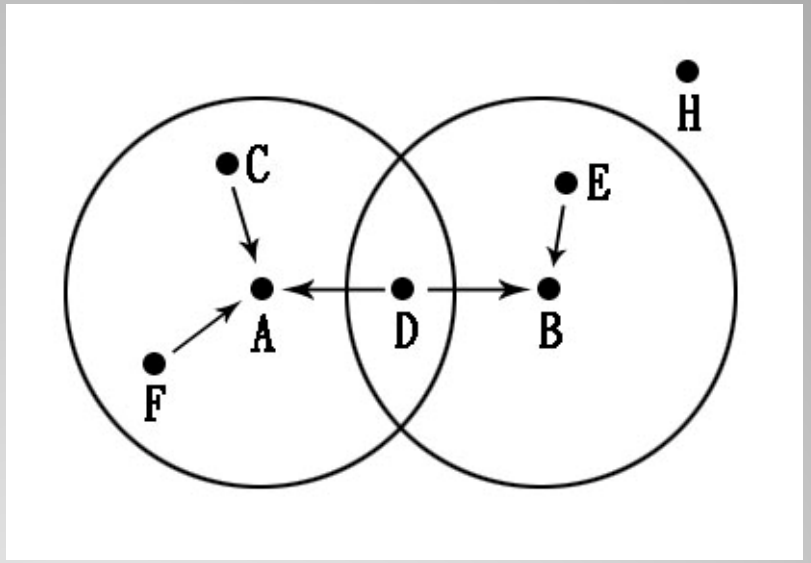
- How to store reputation scores on reliable peers and query them effectively?
- Assumptions
  - Fixed AOI radius
  - P2P-VE overlay provides *AOI neighbors*
  - Users may mutually rate each other



Challenges

- **Reputation evaluation**
  - Precise
  - Simple
- **Storage and query**
  - Scalable
  - Efficient
- **Reliability**
  - Cheat-proof
  - Failure-proof

- Mutual rating
- AOI-only



- **Rating Right**
  - Given only after interactions within AOI
  - Rate once, modify later any time
    - Positive (1), Negative (-1), Neutral (0)

Local reputation evaluation

Reputation storage and query

- **Trust nodes** to store reputation values
  - Chosen from AOI neighbors (may time-out)
  - List of trust nodes stored as **trust list** at each user
- Storage
  - Obtain trust list
  - Send evaluations to trust nodes directly
- Query
  - Obtain trust list
  - Randomly choose *n* trust nodes (out of *N* total)
  - Majority decision

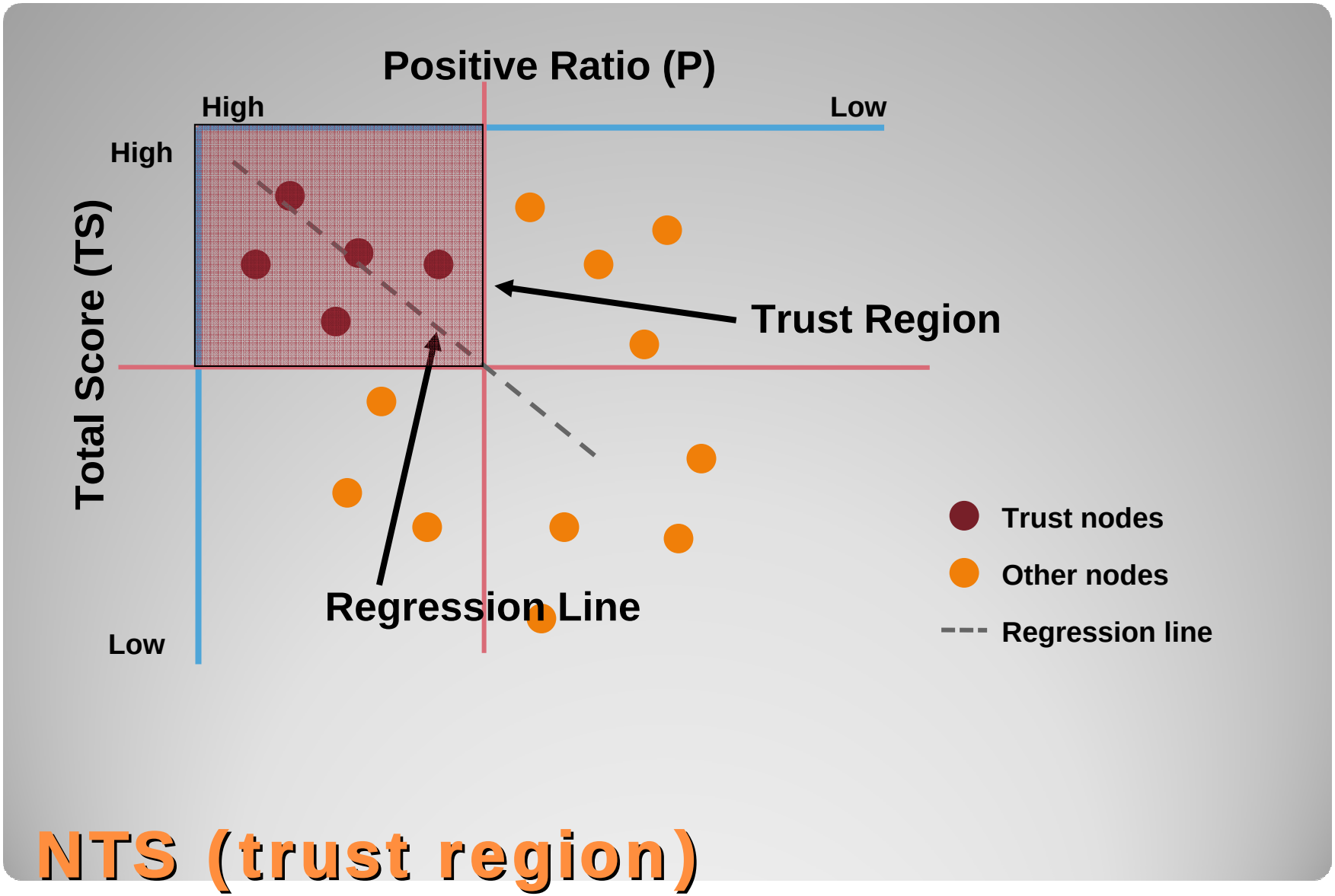
- Combines two intuitive factors
  - **Total score**: *TS (u)*
  - Total rating: *V (u)*
  - **Positive ratio**: *P (u)*

$$P(u) = \frac{TS(u)}{V(u)}$$

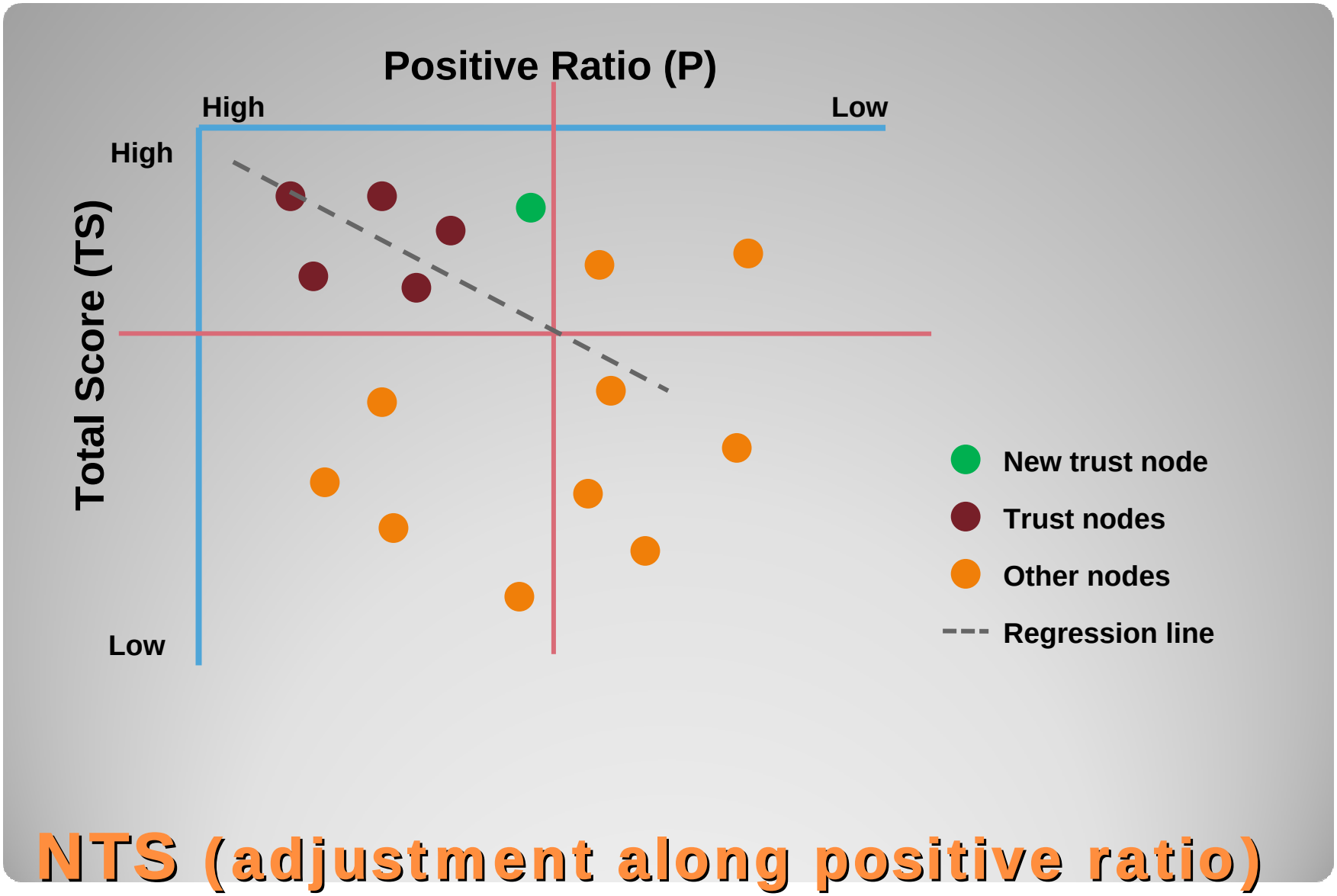
User	Total score TS (u)	# ratings V (u)	Ratio P (u)
A	30	100	0.3
B	9	10	0.9

- Which metric is more important for selecting a given number of trust nodes?

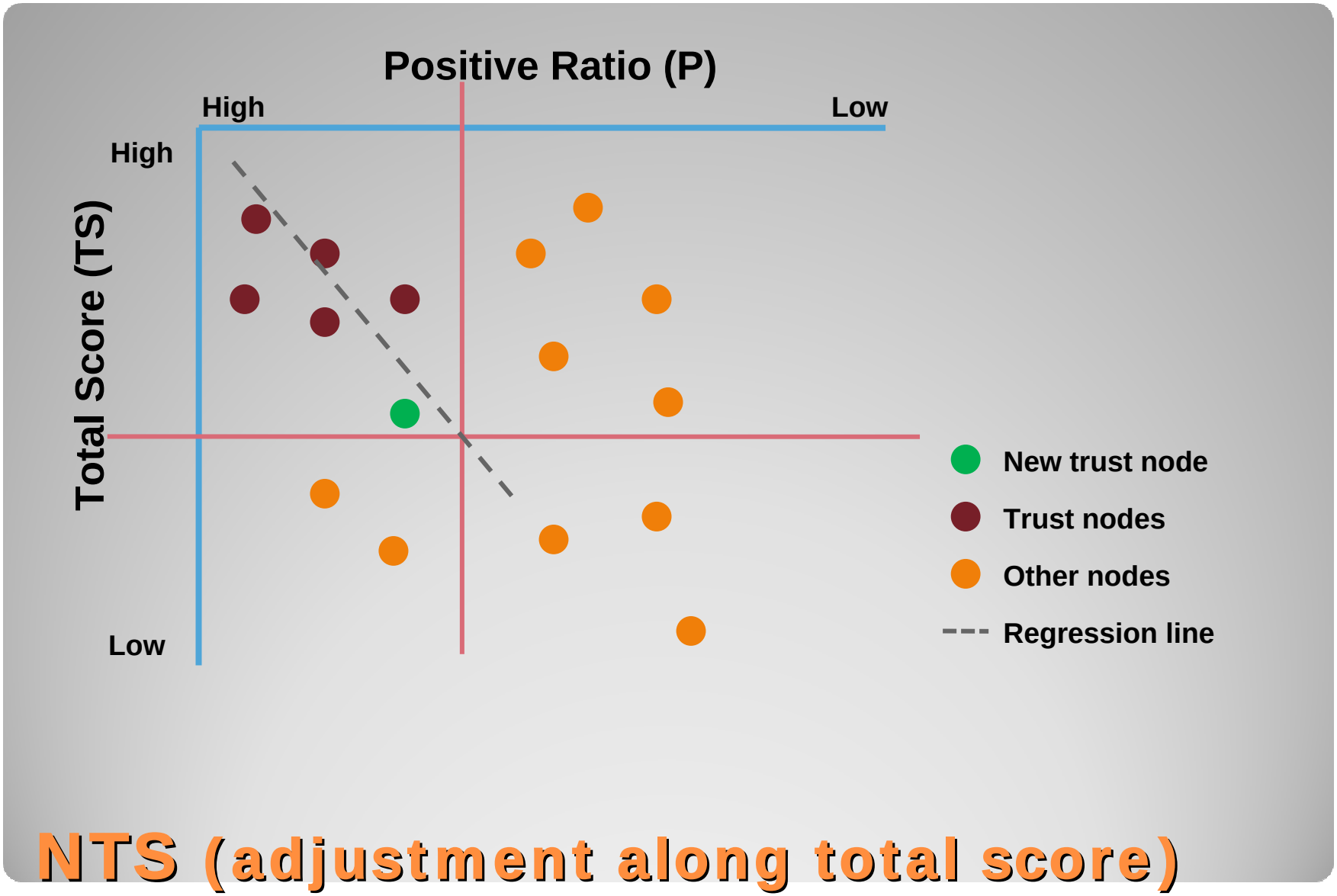
Neighbor Trust node Selection (NTS)



NTS (trust region)



NTS (adjustment along positive ratio)



NTS (adjustment along total score)

Discussions

- **Reputation evaluation**
  - Simple representation for reputations
  - Rating right control
- **Storage and query**
  - Remote storage prevents self-modification
  - Distribution enhances scalability
- **Reliability**
  - Socially enforced mutual monitoring
  - Majority-based value retrieval masks cheating

Issues

- Not 100% secure (but may converge)
- Incentives for rating (need actual tests)
- Query efficiency (due to replications)
- Bootstrapping (will converge?)

Conclusion

- Reputation management for P2P MMOGs
  - Mutual rating
  - Distributed storage (trust nodes)
- Characteristics
  - Low server overhead
  - Consensus-based monitoring
  - Cheat-proof measures

Thank You

Q & A