

Security and trustworthiness in autonomic computing

Roberto G. Cascella

University of Trento - DISI



Autonomic Systems

- Autonomic networks are characterized by:
 - presence of heterogeneous devices
 - the possible coexistence of multiple administrative domains
 - high dynamicity -> churn
 - lack of a centralized authority -> self-management

- New communication paradigms:
 - User-centric (prosumers) → mainly strangers

- ➔ To function properly nodes must fulfill their obligation toward the system and other nodes
...but nodes can misbehave causing the system to be under attack.

Security, Survivability and Self-Preservation

- **Hard security** aims at providing traditional security properties: confidentiality, authentication, integrity, non-repudiation...
- **Soft security**, also called **social control** mechanism, protects against information/service providers who act deceitfully by providing false or misleading information.
- **Survivability** is the system's ability to protect itself and reconfigure as required by changes in the environment. Entities should detect malfunctioning components and adopt protection mechanisms (ex. DoS protection)
- **Self-preservation** is an umbrella term used for describing techniques that aim at protecting the so called "utility" of an entity from deterioration, because of the exposure of some of its resources to other entities.

Challenges to secure the systems

- Analyze the risks and threats.
- Guarantee protection of the system from traditional attacks.
 - Authentication
 - Confidentiality } Key management
- Define self-preservation mechanisms to maximize the resource utilization -> QoS guarantee.
 - Cooperation: model the system by treating entities as rational actors.
 - Trust management

Key Management



Key management issues

Lack of a centralized infrastructure.

– TTP, CAs

Ad hoc and p2p communication (dynamic node addition/removal).

Hard to initially deploy the system.

Require self-organize and self-healing mechanisms to handle the distribution of the keys and the management of the keys

Approaches to key management

- Symmetric key: create keys for pairs of nodes in the system.
 - Pre-distribution of keys: simple approach $\rightarrow O(n^2)$ keys.
 - Network-wide key
- Asymmetric key: establish keys from P_k/S_k keys.
 - Certificate based (Self-organized CA and PGP approaches)
 - Identity based (P_k derived from the identifier)
 - Simple approach \rightarrow distribute P_k of every node

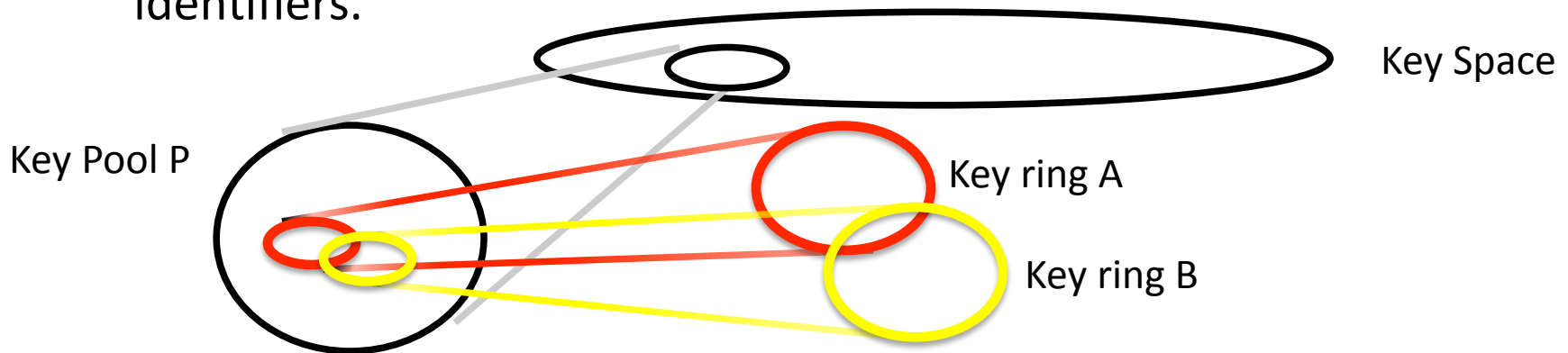
Pre-distributed key management

- Based on probabilistic key sharing among nodes of random graph: no need for all pairs of nodes to be able to communicate to get a connected network
- Keys loaded into nodes prior to deployment
- Simple shared-key discovery protocol for key distribution, revocation and node re-keying
- Three phases are involved: key pre-distribution, shared-key discovery, path-key establishment

- Any two nodes should find a common key in their set to communicate.

Key management: distribution

- A pool P of keys and identifiers are generated offline.
- Each node has a key ring consisting of k keys drawn randomly from the pool.
- Controller node (central entity) stores the nodes and keys identifiers.



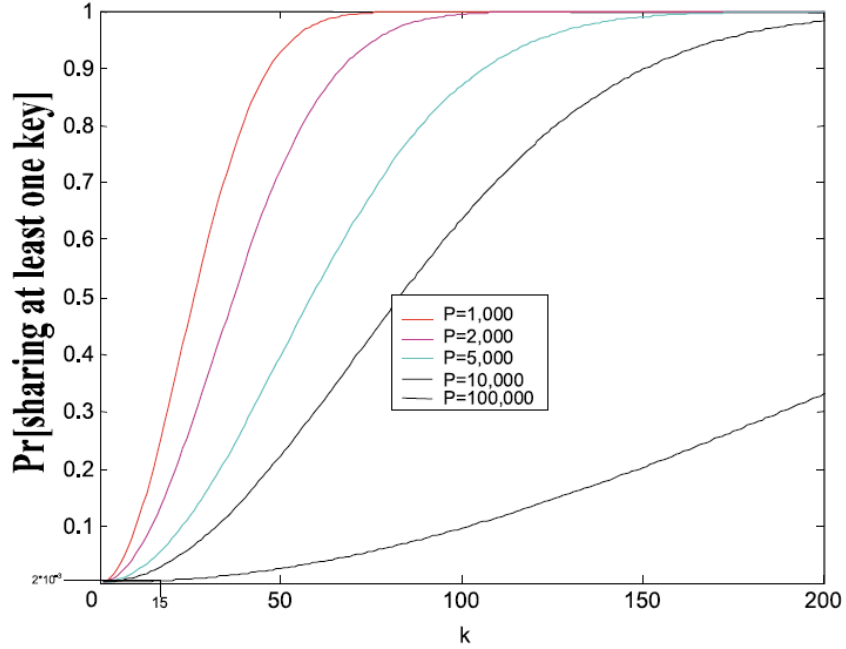
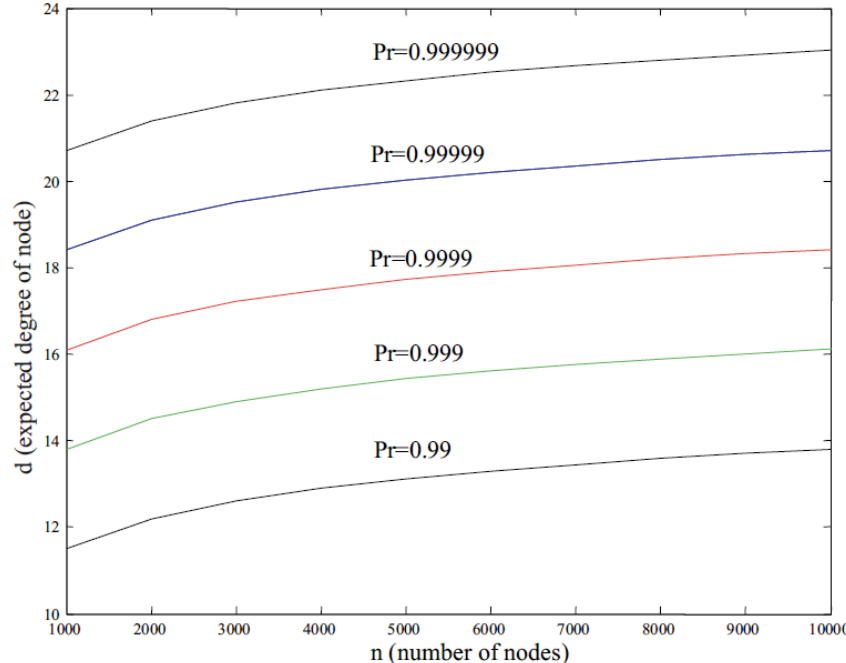
- Two nodes can communicate securely if they have a common key:
 - If a common key is found in their random set (broadcast of the key identifiers only), nodes use this key.
 - If there is no shared key, nodes try to find a path of nodes with whom they share a key (path-establishment phase)

Key management: revocation and re-keying

- When a node is compromised the controller broadcasts a single revocation message:
 - List of key identifiers signed with a key K_e communicate to nodes by an encryption channel
 - Comprised keys are removed by each node.
- Re-keying is implemented when the key expires.
 - There is no scheme defined (nodes should start path discovery).

Average degree of a node to have a probability Pr of connected graph.

Size of the key ring given the size of the pool to have an expected probability to find a common key between nodes



Source: Fig 1 and Fig 2 of L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, pages 41-47, Washington, DC, USA, 2002.

Random Key Pre-distribution Scheme

➤ Advantage

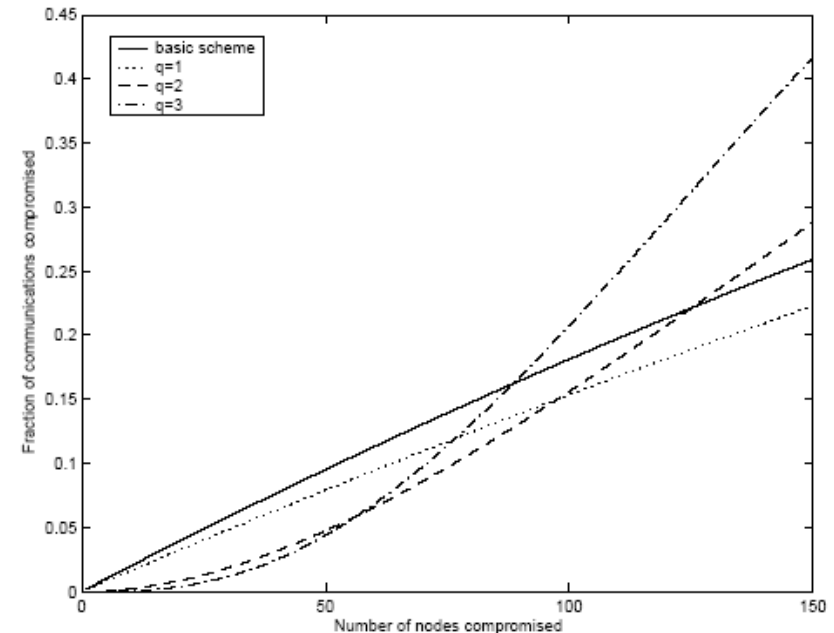
- Fewer keys needed for nodes than pairwise key approach

➤ Drawbacks

- Connectivity cannot be guaranteed (ex. a node does not share keys with its neighbours).
- Low resiliency against node capture attack.
- High overhead: shared-key discovery phase and path-key establishment phase produce too much communication and computational overhead

Pre-distribution keys: enhancement

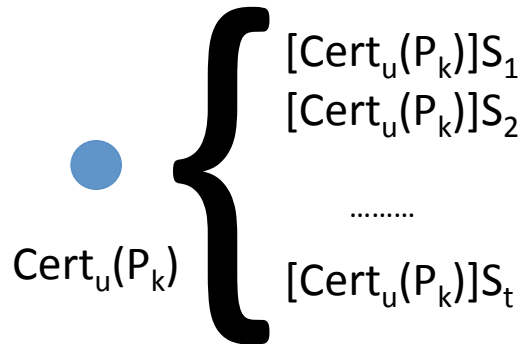
- “q-composite” approach: two nodes should share at least q ($q \geq 1$) keys to setup a secure link between them.
 - Final key is the hash of all q keys
- Better resiliency against node capture attack if only a small number of nodes were captured
- But if a large number of nodes is captured exposes larger portion of the keys



Source: Haowen Chan, Adrian Perrig, and Dawn Song. Random key pre-distribution schemes for sensor networks. In SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy, pages 197–213, Washington, DC, USA, 11-14 May 2003.

Self-organized CA

- Based on threshold cryptography.
- Group of nodes to sign certificates on behalf of the system.
- Private key shared among n nodes and $t+1$ partial signatures for any single certificate.
- Verification of the certificate by using the group P_k



Self-organized CA

Advantages:

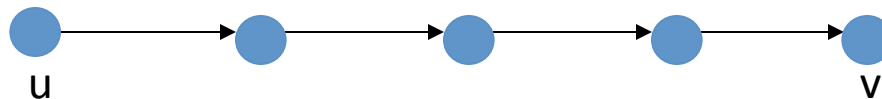
- Provides a fully self-organize solution
- Provides a distributed solution: secret sharing prevents t malicious nodes from reconstructing CA private key

Problems:

- Suffers from sybil attacks
 - An attacker creates enough identities to collect the shares
- Requires an initial server to distribute the shares and the public key.

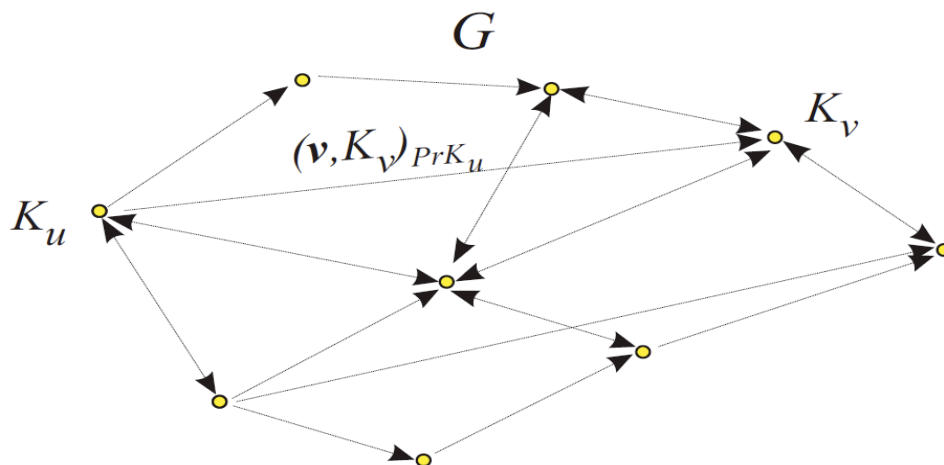
Self-organized public key management

- Based on PGP approach -> storage and distribution of certificates in a self-organized way.
- Certificate Graph:
 - Vertices -> Public Key of the node (assumption: each node has one identity).
 - Edges (u - v) -> Certificates signed by node u that binds the public key of node v to its identity (inbound = certificate signed for the node; outbound = certificates signed from the node)
 - Direct path (u->v) -> certificate chain from node u to node v
- Two certificate graphs:
 - Updated
 - Non-updated (expired certificates)
- Authentication
 - Merge the updated certificate repositories. (note that they are unidirectional)
 - Find chains of public-key certificates.
 - In case no common path is found, use the non-update certificate graph and ask for the renewed certificates of those expired on the path.



Certificate Creation

- Node u gets the public key of v by using a side channel.
- Node u verifies the identity of v and creates a certificate for v .
- Node u stores the certificate of v in its local repository (certificate graph) and sends the certificate to v .
- Small world phenomena \rightarrow certificate graph will be (strongly) connected if the number of branches per node is around the natural logarithm of the total number of nodes.

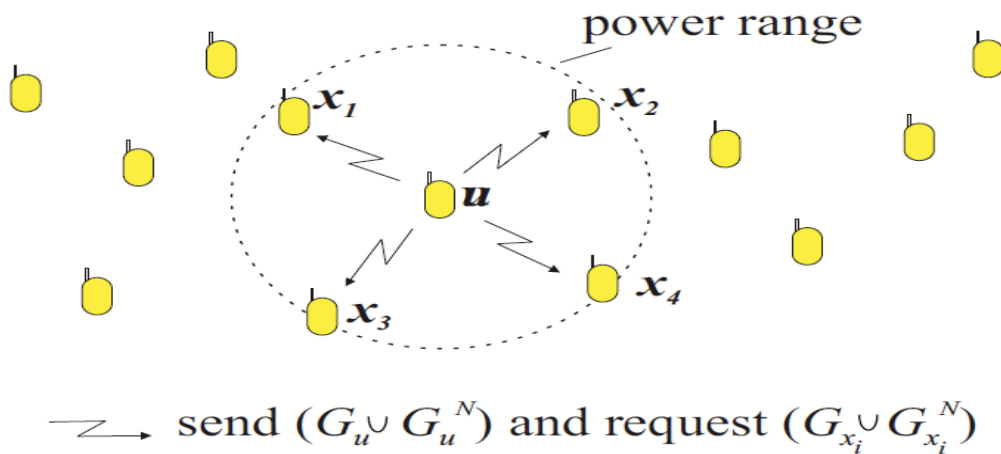


Source: Fig. 1 of S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," IEEE Transactions on Mobile Computing, vol. 2, no. 1, pp. 52-64, 2003.

Certificate exchange

Nodes exchange periodically their certificate graphs with their neighbours.

- A hash is computed for every certificate.
- The hash values are exchanged.
- Nodes verify the hash to check whether they do not have the same certificate.
- Only missed certificates are exchanged.



Source: Fig. 1 of S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," IEEE Transactions on Mobile Computing, vol. 2, no. 1, pp. 52-64, 2003.

Revocation

➤ Certificate revocation

– Explicitly

- A node issues an explicit revocation statement to those nodes that requested an update of the certificate issued.

– Implicitly

- Expiration time of the certificate -> simple but it needs time synchronization of the nodes and the expiration time should be chosen properly

➤ Key revocation

- Inform the issuers of its certificate, then the certificate is invalid.

Drawbacks

- Paths are unidirectional -> to have mutual authentication two (possibly disjoint)certificate paths need to be found.
- Nodes can create false certificates that create conflicts which can be solved if other certificate paths exist.
- A relatively large size of local repositories is required to achieve low probability of public key authentication failure.
- The revocation of certificates requires nodes to be in the proximity (communication range).

Self-healing key management

- Self-healing key distribution implies:
 - Forward and backward secrecy
 - Resilient to collusion
 - Polynomial based with parameters:
 - t : resilience to collusion
 - m : the number of sessions
- Efficient mechanism for group key management in high dynamic system.
 - Reduce impacts of key compromise
 - Adapt to group member changes
 - Recover the current key by combining information obtained in previous or subsequent sessions.

Reputation Management Systems



Threats: Adversarial Model

- Two broader classes of attack sources:
 - Selfish nodes
 - Malicious nodes
- Selfish or **rational** nodes maximize their own utility by prediction of the transactions' outcome
 - Selfish behavior prevents the realization of the system objective
 - ➔ Do not share the content/data they own (free-riders) or contribute with minimal resources
- Malicious nodes actively attack the system with the intent of disrupting the normal functionality
 - ex. False content – virus

In reality the fraction of malicious nodes is low compared to free-riders

Threats: Adversarial Model (Identity and trust)

- Sybil attack
 - Forge identities and appear in the system with new identifiers – multiple identities
- Whitewashing
 - Change identity after behaving maliciously
- Impersonation
 - Steal an identity
- Repudiation
 - Deny an action
- DoS
 - Saturate resources to deny services to legitimate users

Hard security solutions (Cryptography-based)

Threats: Adversarial Model (Behavioral)

- Inauthentic
 - Contribute with different content from requested
- Traitors
 - Behave inconsistently in transactions
- Collusion
 - Join a “community” to damage the system
- Front peers
 - Promote malicious activity of other nodes
- Bad Mouthing
 - Send false information on other nodes
- Ballot Stuffing
 - Report false transactions

Soft-security solutions (Reputation based)

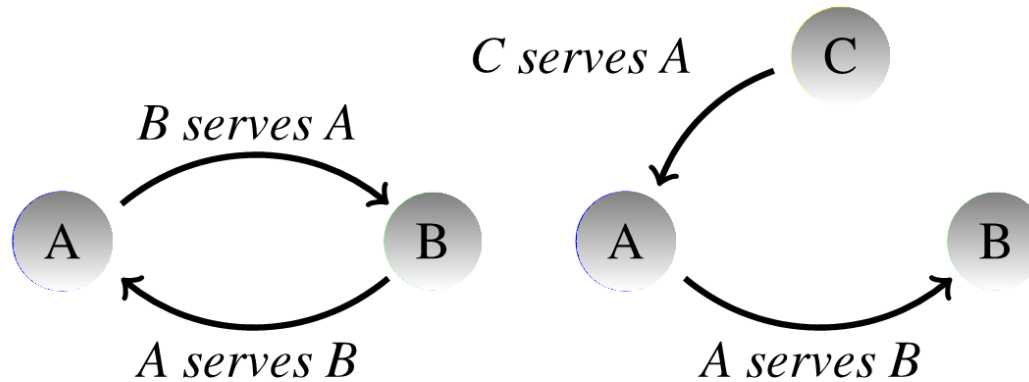
Soft-security solutions

- Can we define a Web of Trust?
 - A worldwide PKI is difficult to achieve
 - A PGP-like solution might require personal acquaintances

- In many cases defining the risk of an interaction is more useful than unconditional trust.

- Reputation based approach:
 - Reputation management systems create a framework to foster cooperation
 - Autonomic systems must create and maintain trust to function properly.

Social science



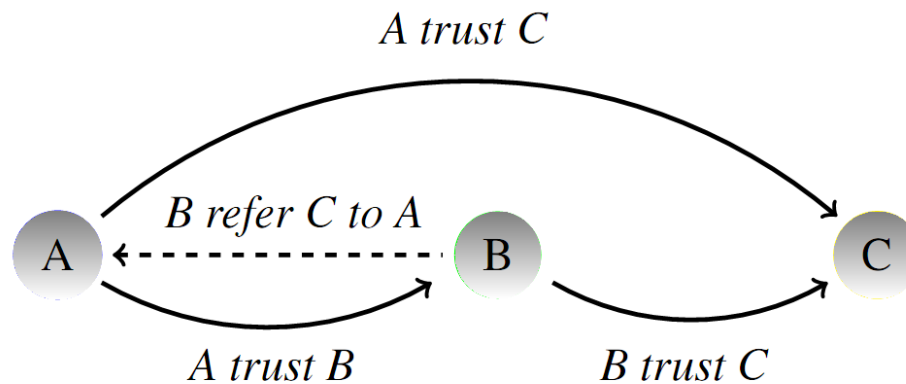
(a) Direct Reciprocity

(b) Indirect Reciprocity

- Reciprocal altruism: entities do not expect any service in return
- Indirect reciprocity possible only if transactions are monitored

Reputation

- Peer-to-peer systems must create and maintain trust to function properly.
 - Provision trust is users' knowledge about the reliability of authenticated parties
- Reputation is an important component of all human (and machine) interactions



Trust transitivity

Reputation Management Systems:

Definitions and Metrics



Dissecting reputation management systems

- In a reputation management system the reputation information needs to be
 - 1) collected from the feedback providers (how a node behaved in the past)
 - 2) aggregated to form a useful measure of trustworthiness (where?)^I
 - 3) disseminated to members requesting the reputation value of a particular node

- Reputation management systems to be useful must have three properties^{II}:
 - Nodes should last for long in the system
 - Nodes should distribute feedbacks
 - Feedbacks should be useful to the community

I) S. Marti and H. Garcia-Molina. Taxonomy of trust: categorizing p2p reputation systems. *Comp. Net.*, 50(4):472–484, 2006.

II) P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems: Facilitating trust in internet interactions. *Communications of the ACM*, 43(12):45–48, 2000.

Reputation types and goal

- The type of trust is application dependent:
 - Opinion
 - Credibility of reporting nodes
 - Reputation (community judgment)

Opinion is the judgment that a node forms after a transaction on the quality of service received by the counter part.

It is personal and the scope is limited to a single interaction. An opinion forms the so called private or first hand information resulting from own experience.

Reputation types and goal

- The type of trust is application dependent:
 - Opinion
 - Credibility of reporting nodes
 - Reputation (community judgment)

Credibility is the confidence that a node forms on the judging capabilities of another node in reporting opinions. It is personal and called second order reputation.

Reputation types and goal

- The type of trust is application dependent:
 - Opinion
 - Credibility of reporting nodes
 - Reputation (community judgment)

Reputation measures the trustworthiness of a peer in a system.

It is the global system-wide view of a node
or what is believed about this node.

In short, reputation is the collective measure of trustworthiness based on the judgement of a community. It is quantified and it is calculated by considering the action of a node in the view of a community of users.

Reputation types and goal

➤ The type of trust is application dependent:

- Opinion
- Credibility of reporting nodes
- Reputation (community judgment)

➤ Reputation to be useful must be objective

- Algorithms for aggregation of reported values

$$T_{xj} = (1 - w_p)O^{avg} + w_p \frac{\sum_d R_{dj}^{avg} \cdot C_{xd}}{\sum_d C_{xd}}$$

➤ The goal of the reputation might be context and application dependent:

- A node can be trustworthy for providing service of type 1 or/and untrustworthy for providing service of type 2

Trust

Trust is a relationship of reliance and decision in social science.
A trusted party proves to benefit the belief of other peers to fulfill its obligation.
The definition of trust might include also the concept of risk,
when the value of the outcome of a transaction is high and
there exists the probability of failure.
The concept of trust is stronger than reputation as a node risks in person.

- The trustworthiness of the node is subjective
 - Function of reputation and opinion
 - Quantification of the risk

Simple algorithms for aggregation

➤ Average

➤ Weighted aggregation:

- Age of the input ($e^{-\gamma t}$ where α depends on network conditions and characterize the aging)

$$R_{xj} = \frac{\sum_i F_i e^{-\gamma t_i}}{\sum_i e^{-\gamma t_i}}$$

- Likelihood a node lies for reputation values (C credibility factor)

$$R_{xj} = \frac{\sum_i F_{ij}^{avg} \cdot C_{xi}}{\sum_i C_{xi}}$$

More complex mechanisms

➤ Beta probability density function

$$\text{Beta}(\theta, \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

$\alpha=p+1$ $\beta=n+1$, Γ is the Gamma Function

➤ Friend of friend

– Nodes are vertices of the graph

$$R_{xj} = \sum_{e \in \text{incoming}(j)} w_e \cdot \frac{R_{uj}}{\sum_{f \in \text{incoming}(j)} R_{uf}}$$

Relevant “context” information

- Importance of the transaction
 - opportunistic model

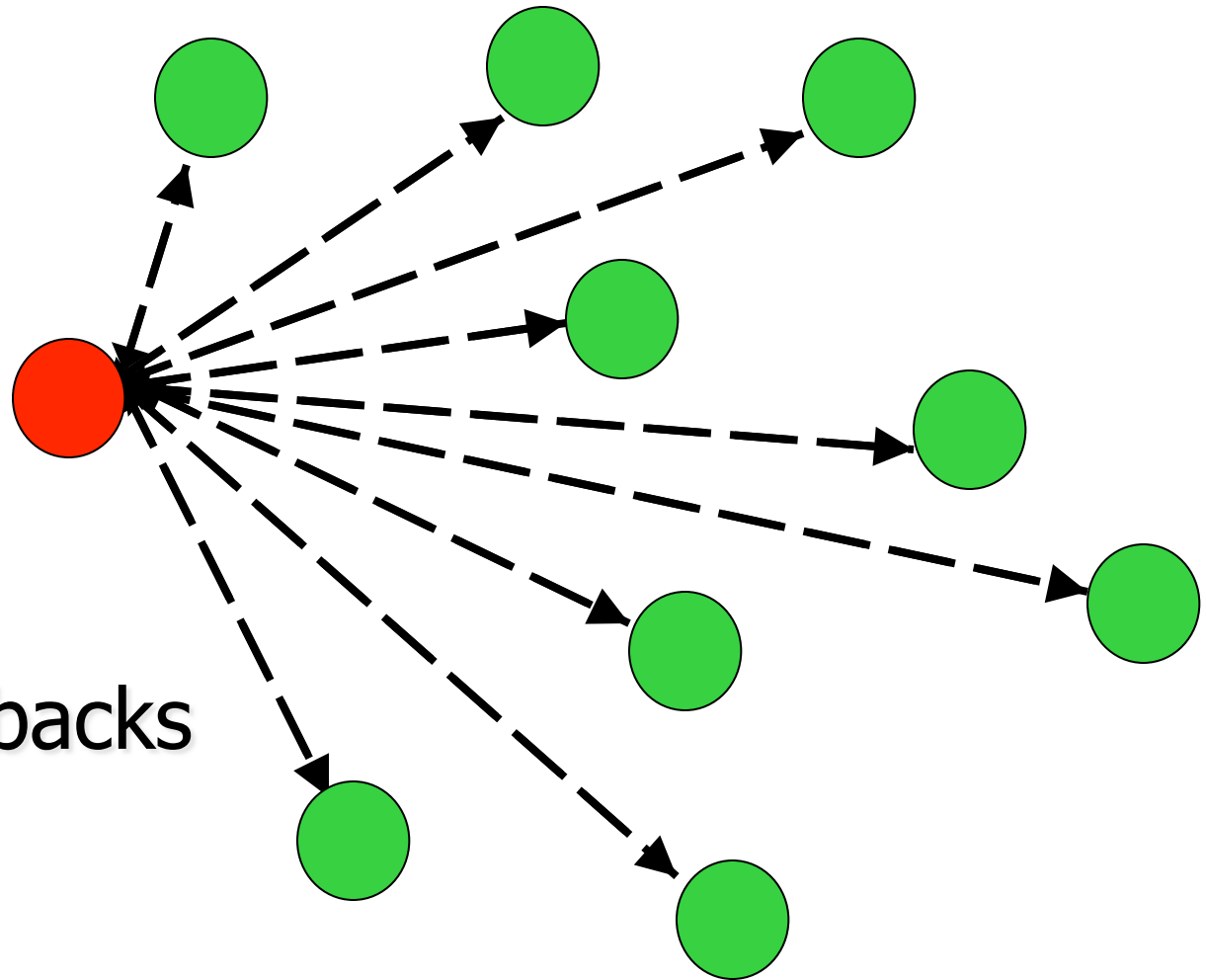
- Communication model
 - network capacity and topology

- Nodes capabilities:
 - computation
 - storage

Collection of feedbacks

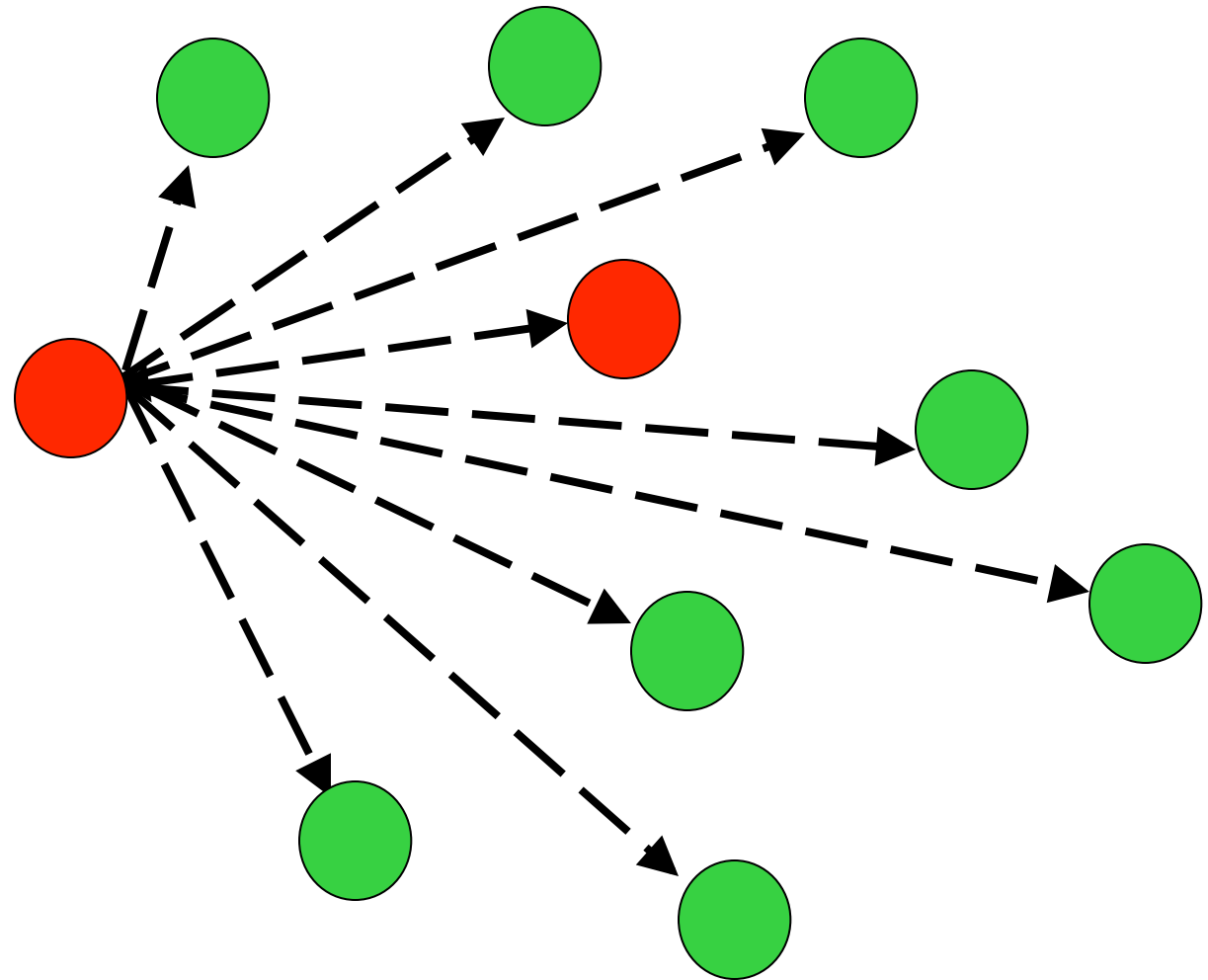
- This is essential as the trustworthiness of a node is dependant on how a node has behaved in the past.
- The gathered information represents the input to the reputation aggregation function.
- Possible approaches:
 - Reactive
 - Proactive
 - Hybrid (Proactive and reactive)

Collection of feedbacks: reactive



Send feedbacks

Collection of feedbacks: proactive



Interaction back

Dissemination of Trust

- This can be done with similar techniques like collecting feedbacks:
 - Reactive
 - Proactive
- Proactive schemes require the receiving node to store trust information
 - Recent information can be more valuable
 - > timestamps

Metrics

- Success Rate $= \frac{\#Tr_{good} + \#Av_{malicious}}{\text{Total \# of transactions}}$
- Detection of malicious nodes
 - ➔ Reputation value
- Communication overhead
 - ➔ Messages to send reputation information
- Computational overhead
 - ➔ Cost to process messages
- Storage
 - ➔ Maintenance of the history

Existing approaches

➤ EigenTrust^I

- The algorithm is based on the notion of transitive trust (Friend of friend approach)
- The algorithm has faster converge with a set of pre-trusted peers (malicious peers can lie)
- Secure trust storage (nodes can lie for themselves)

➤ ROCQ^{II}

- Nodes send feedback after every transaction
- Feedback is aggregated to form each node's reputation
- Collection, storage, aggregation and dissemination of trust data happens in a distributed fashion (**score managers**)

I) S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. "The Eigentrust algorithm for reputation management in P2P networks". In Proceeding of WWW 2003: 640-651

II) A. Garg, R. Battiti, and R. Casella. Reputation management: Experiments on the Robustness of ROCQ. In WAGEN Workshop at ISADS, pp. 725–730, China, Apr. 2005.

How ROQC Works

- Users send feedback after every transaction
- Feedback is aggregated to form each user's reputation
- Collection, storage, aggregation and dissemination of trust data happens in a distributed fashion

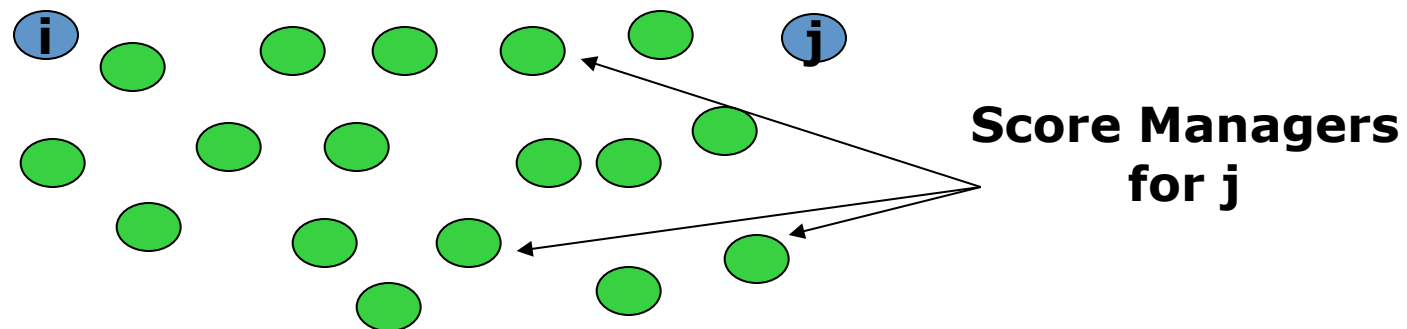
The ROCQ scheme: system model

Reputation of node formed by averaging opinions of all its transaction partners

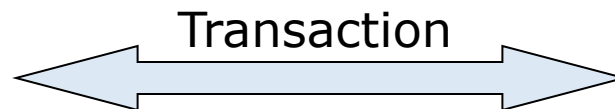
Opinion is formed by a node based on how other nodes have behaved during a transaction

Quality represents node's confidence in an opinion that it reports

Credibility measures node's honesty in the reputation system



**Evaluate trust
for j**



The role of “Credibility”

- Without credibility a system will be open to attacks based on falsified opinions
 - Nothing prevents me from lying about your behavior
- Credibility of a user is modified based on agreement
- Credibility modification is influenced by reported quality

The role of “Quality”

- A user’s confidence in an opinion that it reports
- Wrong opinions can cause loss of credibility
- A user may not be sure of its opinion
- Some interactions are more important than others
- Measured as confidence level that actual trust rating lies within $r\%$ of opinion

ROCQ: Equations

$$R_{mj} = \frac{\sum_i O_{ij}^{avg} \cdot C_{mi} \cdot Q_{ij}}{\sum_i C_{mi} \cdot Q_{ij}}$$

$$C_{mi}^{k+1} = \begin{cases} C_{mi}^k + \frac{(1 - C_{mi}^k Q_{ij})}{2} \left(1 - \frac{|R_{mj} - O_{ij}^{avg}|}{s_{mj}}\right), & \text{if } |R_{mj} - O_{ij}^{avg}| < s_{mj} \\ C_{mi}^k - \frac{C_{mi}^k Q_{ij}}{2} \left(1 - \frac{s_{mj}}{|R_{mj} - O_{ij}^{avg}|}\right), & \text{if } |R_{mj} - O_{ij}^{avg}| \geq s_{mj} \end{cases}$$

ROCQ: Equations

Quality is the likelihood that actual trust value lies within this range



$$O_{ij}^{avg} \cdot \left(1 \pm \frac{r}{100}\right)$$

$$Q_{ij} = 1 - B \left(\frac{(N_{ij} - 1)}{(N_{ij} - 1) + t^2}; \frac{1}{2} \cdot (N_{ij} - 1), \frac{1}{2} \right)$$

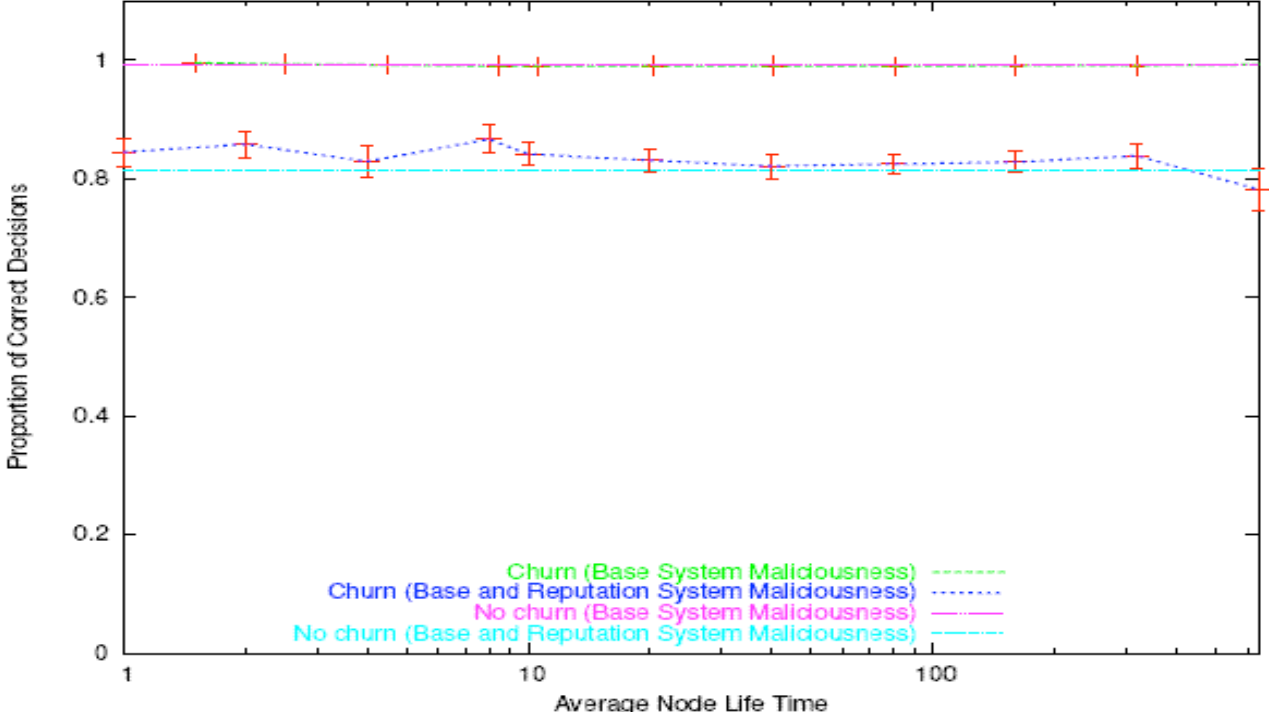
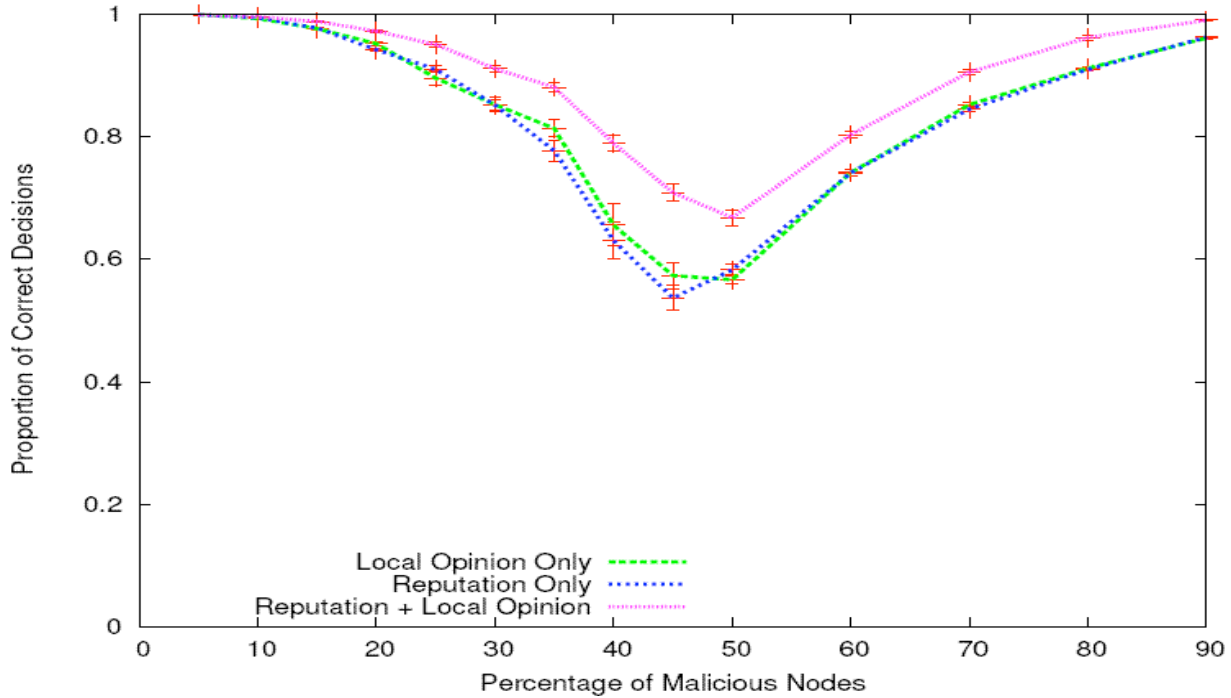
The t -value for the *Student's t-distribution* is given by the following equation:

$$t = \frac{r}{100} \cdot \frac{O_{ij}^{avg} \cdot \sqrt{N_{ij}}}{s_{ij}}$$

System Architecture

- Assume a structured overlay network that uses Distributed Hash Tables
- DHT is used to assign Score Managers (SM)
- Multiple SMs to ensure reliability and guard against malicious SMs
- SM for a peer stores all trust information related to that peer
- Opinions about a peer are reported to all of its SMs

6 score managers
 Deterministic threshold of 0.5
 Proactive dissemination
 Nodes 200
 Transactions 50,000



EigenTrust

➤ The EigenTrust algorithm is based on the notion of transitive trust

➤ Local Rating: $s_{ij} = \text{sat}(i, j) - \text{unsat}(i, j)$

➤ Normalized rating: $c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}$

➤ Local trust values: $t_{ik} = \sum_j c_{ij} c_{jk}$

➤ Friend of friend: $\vec{t} = (C^T)^n \vec{c}_i$

Note that for large values of n

➔ t will converge to the same vector

Left principal eigenvector of C

EigenTrust: refinements

- The algorithm has faster converge with a set of pre-trusted peers
 - Malicious peers lies

Definitions:

- A_i : set of peers which have downloaded files from peer i
- B_i : set of peers from which peer i has downloaded files

Algorithm:

Each peer i do {

Query all peers $j \in A_i$ for $t_j^{(0)} = p_j$;

repeat

Compute $t_i^{(k+1)} = (1 - a)(c_{1i}t_1^{(k)} + c_{2i}t_2^{(k)} + \dots + c_{ni}t_n^{(k)}) + ap_i$;

Send $c_{ij}t_i^{(k+1)}$ to all peers $j \in B_i$;

Compute $\delta = |t_i^{(k+1)} - t_i^{(k)}|$;

Wait for all peers $j \in A_i$ to return $c_{ji}t_j^{(k+1)}$;

until $\delta < \epsilon$;

}

- Secure trust storage
 - Nodes might report false trust values for themselves

Distributed version

Source: Sepandar D. Kamvar, Mario T. Schlosser, Hector Garcia-Molina. "The EigenTrust algorithm for reputation management in P2P networks". In Proceeding of WWW 2003: 640-651

Advances compared to existing approaches

- Analysis of the modes of operation of reputation management systems.
 - Collection of feedbacks: reactive or proactive approach
 - Dissemination of trust: reactive or proactive approach
- Determination of the costs (communication overhead) and benefits of their application:
 - Cost: # of messages to handle reputation information
 - Benefits: # of avoided transactions with malicious nodes and their correct identification

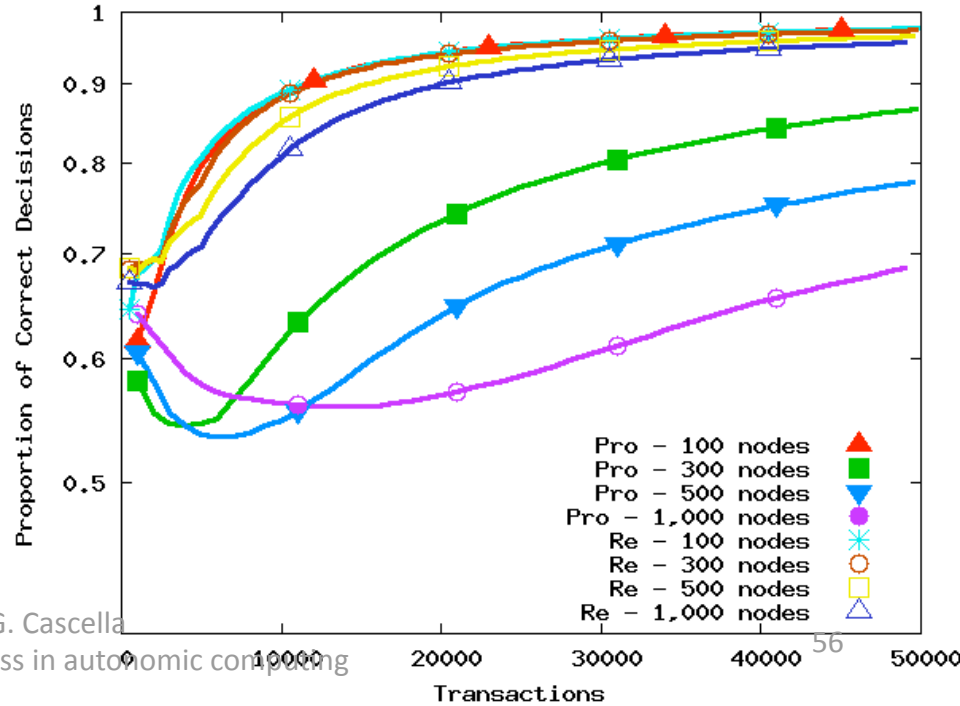
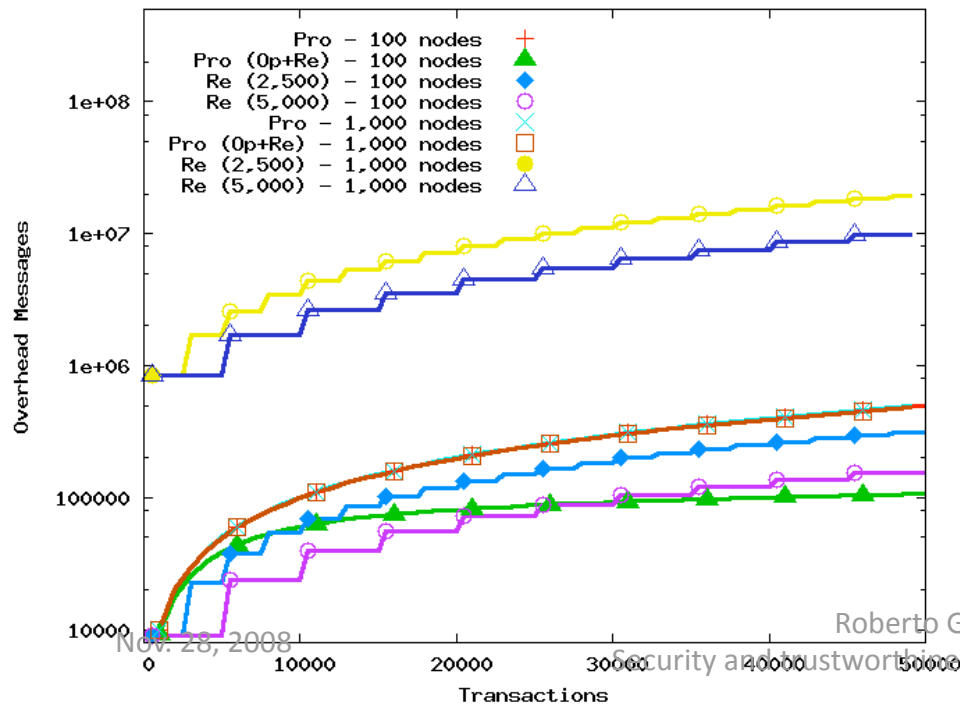
(reputation value should reflect the behaviour of a node)

Roberto G. Cascella. Costs and benefits of reputation management systems. In the 9th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM 2008), Newport Beach, CA, USA, June 23-27 2008. IEEE

Communication Overhead

Parameters

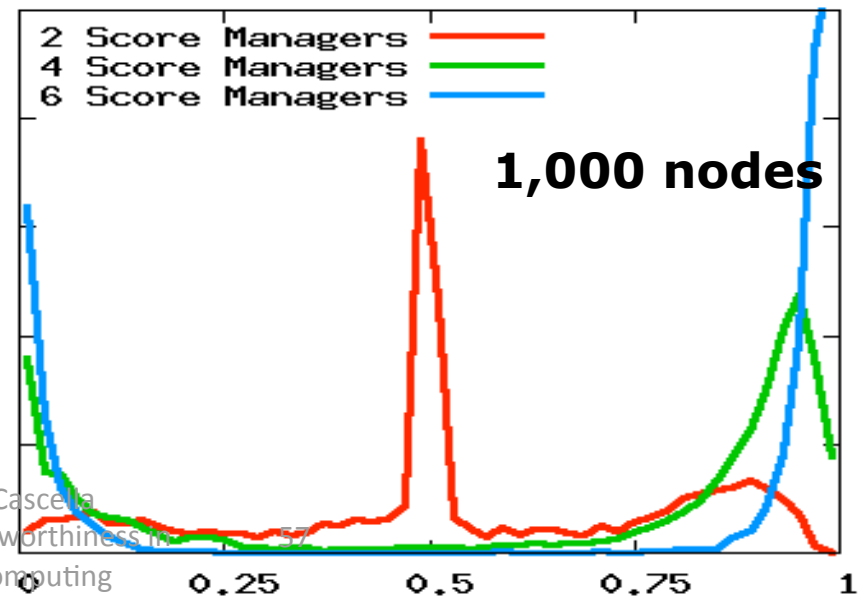
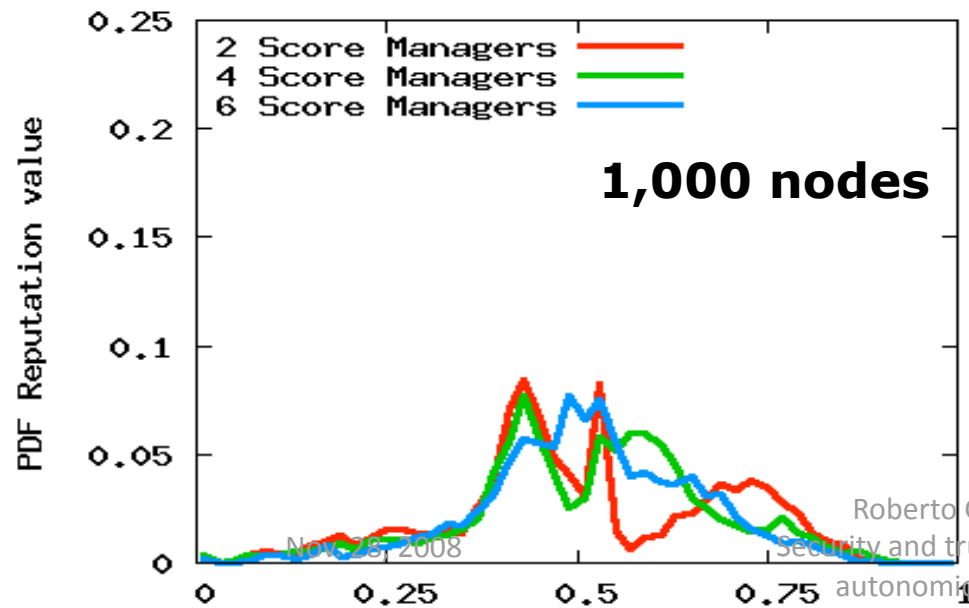
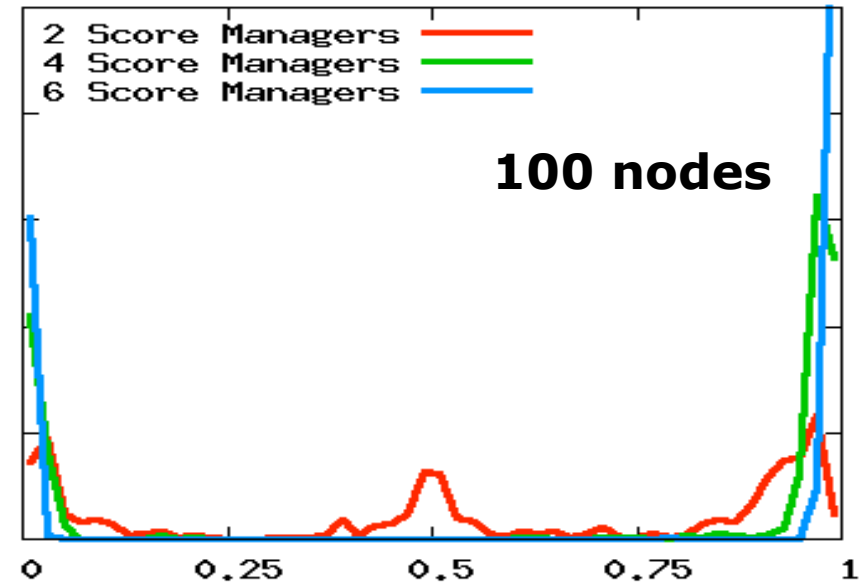
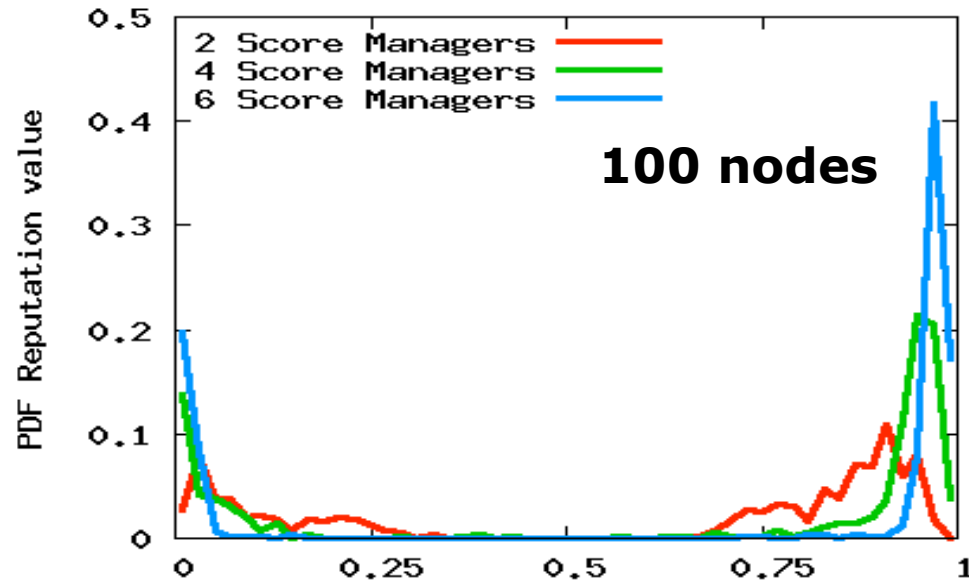
- Deterministic threshold 0.5
- Score Managers 5
- Malicious nodes 30%
- Malicious behaviour: transaction and feedback



Proactive approach

Reactive approach

(2,500)



Nov 28, 2008

Considerations

- Communication overhead must be considered to evaluate the benefits
- The design depends on the underlying topology and network
- The correct estimation of reputation depends on:
 - Amount of historical information
 - Size of the system
 - Frequency of interaction

Reputation Management Systems:

Theoretical analysis



Selfish nodes: Modeling reputation

- Theoretical approaches to study the dynamic of the system:
 - Mechanism design
 - Game theory

- ➔ Simplifications must be made to study the complexity of networked systems
 - ...but still useful to understand the behaviour of rational nodes

Reputation to foster cooperation

- Introduce reputation in the utility function
- Selective cooperation (reduce the impact of selfish nodes)
- Reputation as a metric to prioritize nodes' interactions

The Reputation Game: based on the Iterative Prisoner's Dilemma

Objective:

- Study if reputation is sufficient to sustain cooperation
- Determine how nodes “value” reputation

Settings:

- Non-cooperative and simultaneous move game
- At each stage 2-players are selected randomly

		Receiver	
		Cooperate	Defect
Serving	Cooperate	(R_s, R_r)	(S_s, T_r)
	Defect	(T_s, S_r)	(P_s, P_r)

$$T > R > P > S$$

$$R_r + R_c > S_c + T_r$$

$$R_c + R_r > S_r + T_c$$

Two available actions:

- Cooperate
- Defect

Outcome of the game stage:

- Reputation Value
- Asymmetric Payoff

Game settings

Service Provider

$$\pi_s^{t+1} = \begin{cases} -C + B - C_p(I_r) + f(I_s^t), & R_s \\ -C - C_p + f(I_s^t), & S_s \\ B + f(I_s^t), & T_s \\ f(I_s^t), & P_s \end{cases}$$

Service Consumer

$$\pi_r^{t+1} = \begin{cases} -B + S + g(I_r^t), & R_r \\ -B + g(I_r^t), & S_r \\ S + g(I_r^t), & T_r \\ g(I_r^t), & P_r \end{cases}$$

$$f(I_s^t) = B * [I_s * (1 - \alpha) + \alpha * v]$$

$$g(I_r^t) = S * [I_r * (1 - \alpha) + \alpha * v]$$

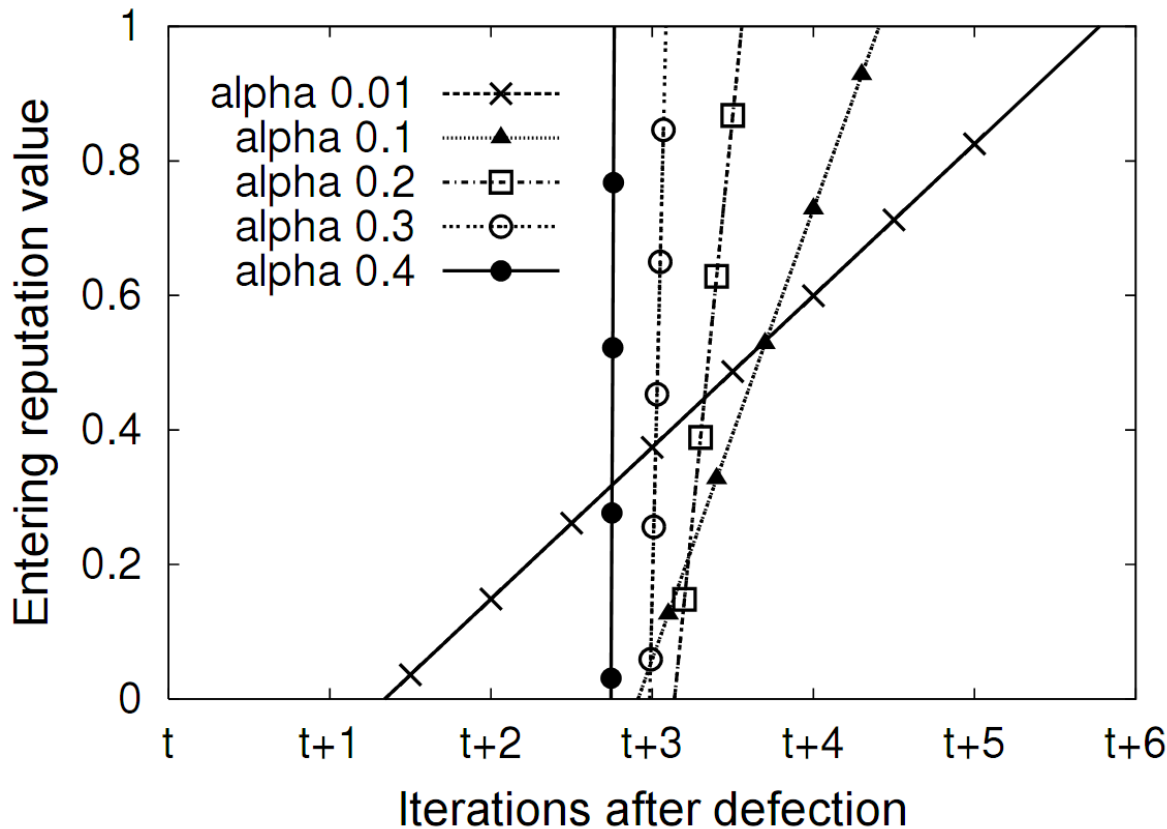
C_p : “punishment” cost

$$C_p(I_r) = \frac{B}{1 + e^{5I_r}}$$

Reputation is updated

$$I_{t+1} = \begin{cases} 0, & \text{if } t = 0 \\ I_t * (1 - \alpha) + v * \alpha, & \text{if } t > 0 \end{cases}$$

Nash Equilibrium



Parameters

- Benefit = 15
- Service = 25
- Cost = 5
- Nodes = 1,000
- t = 10 (iterations after defection)

Findings:

- Similar results with different t

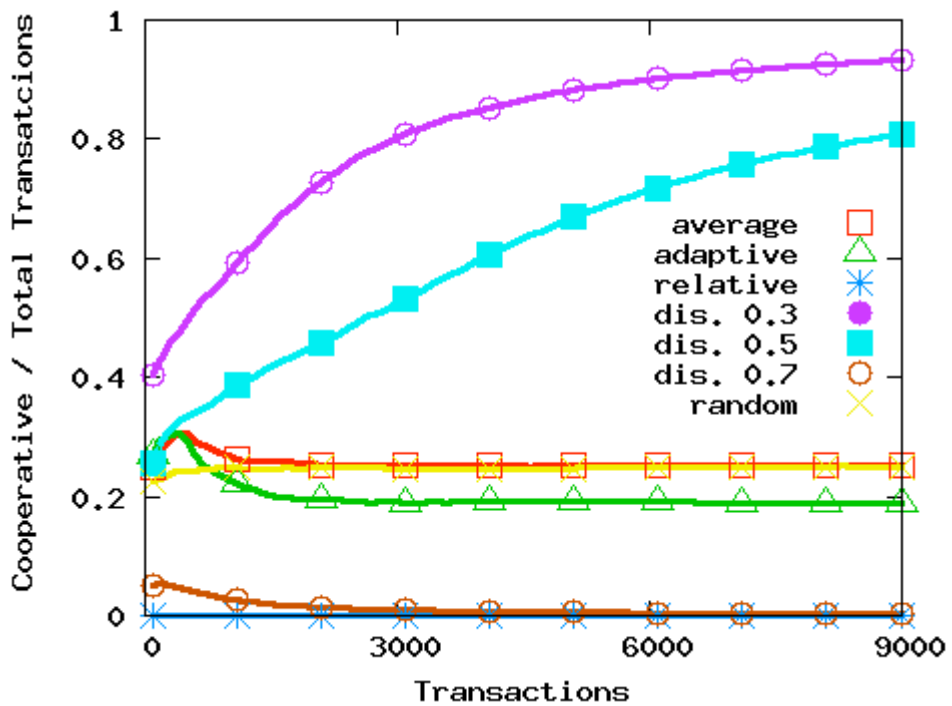
$$I_s(t=0) \leq \frac{10x + 15(2 - \alpha)[(1 - \alpha)^t - 1] - 15(1 - \alpha)^x + 15 / (1 + e^{5 * \{(1 - \alpha)^t [I_r(t=0) - 1] + 1\}})}{25(2 - \alpha)(1 - \alpha)^t}$$

Selection strategies

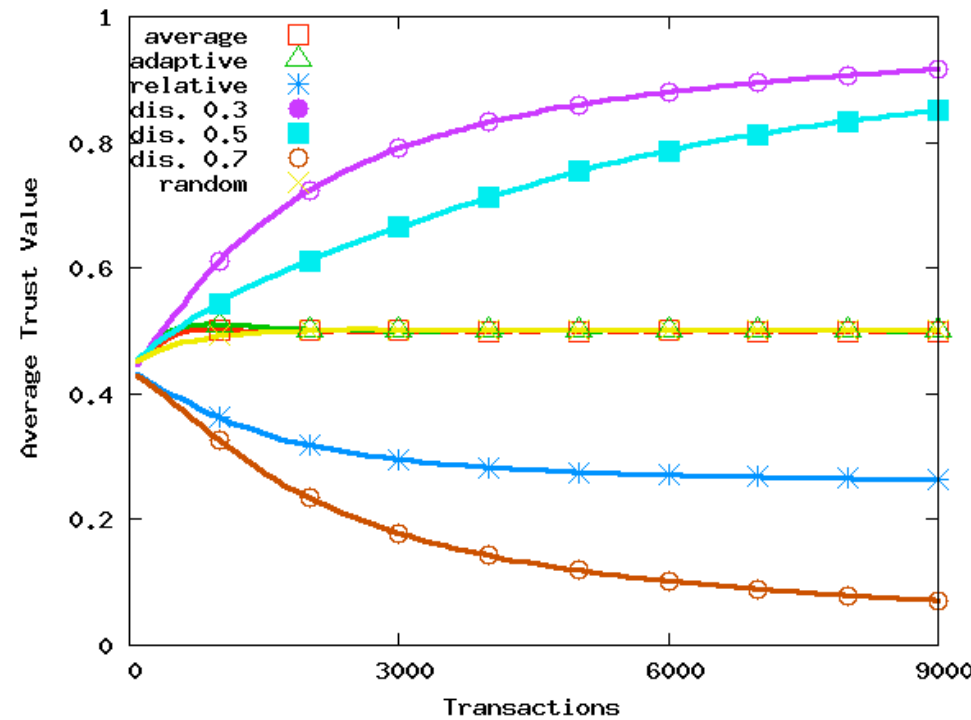
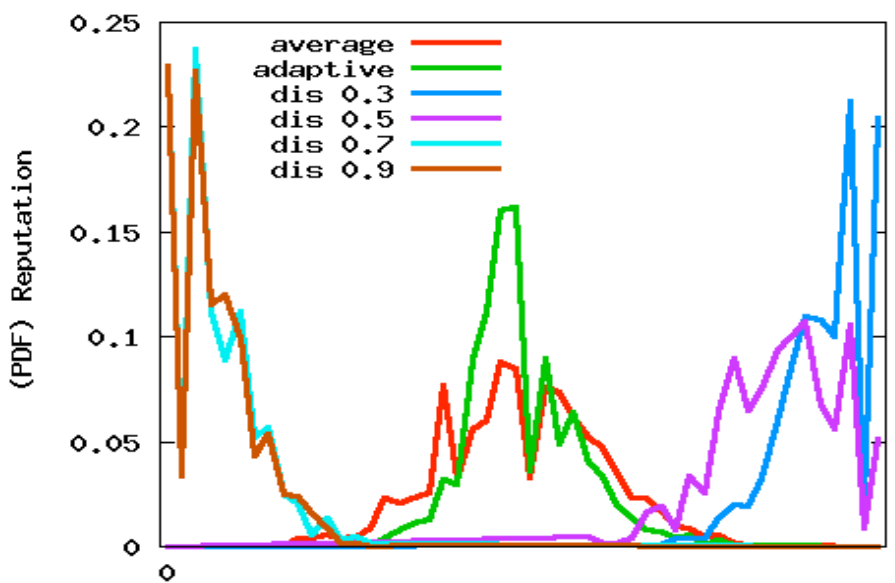
- **Discriminant:**
 - Cooperation is based on a fix threshold and correspondent trust value
- **Average:**
 - Cooperation is decided on the average trust value only
- **Relative:**
 - Cooperation is decided on the correspondent trust value
- **Adaptive:**
 - Cooperation is decided on the average trust, personal and correspondent trust value

➤ **Nodes = 1,000**

➤ **Topology random**



Distribution of reputation



Considerations

- The threshold is a critical parameter
 - Initial trust value must be set in accordance

 - The adaptive and the average strategies keep the reputation value “constant” in the system
 - The average reputation value might be not always available
 - Reputation is sufficient to sustain cooperation
- ➔ The results can be generalized

Practical considerations

- Design of reputation management systems:
 - The results obtained can guide the definition of new schemes
 - The models used for evaluation are general

- Reputation is not a substitute for security but:
 - It is a useful metric to predict future interactions
 - It is a self-preservation mechanism
(behavioral attacks)

Thanks!

Contact details:

cascella@disi.unitn.it

<http://disi.unitn.it/~cascella>

