

A Protocol For Pervasive Distributed Computing Reliability

*Alberto Ferrante, Roberto Pompei,
Anastasia Stulova, Antonio V. Taddeo*

ALaRI, University of Lugano

e-mail: ferrante@alari.ch

Introduction

The Trusting Protocol

Simulations

Attacks

Conclusions and Future
Work

Introduction

The Trusting Protocol

Simulations

Attacks

Conclusions and Future Work



Distributed Pervasive Systems

Introduction

Distributed Pervasive
Systems

Delegation Protocol

Problem Statement

Related work

The Trusting Protocol

Simulations

Attacks

Conclusions and Future
Work

- Future pervasive systems will adopt innovative architectures;
- they will be distributed:
 - computational nodes organized in networks;
 - nodes may move from one network to another;
 - tasks can be delegated to other nodes.



Delegation Protocol

Introduction

Distributed Pervasive
Systems

Delegation Protocol

Problem Statement

Related work

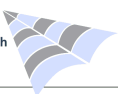
The Trusting Protocol

Simulations

Attacks

Conclusions and Future
Work

- Query for availability of nodes;
- evaluate the most appropriate node to execute the task;
- assign the task;
- receive the results.



Problem Statement

Introduction

Distributed Pervasive
Systems

Delegation Protocol

Problem Statement

Related work

The Trusting Protocol

Simulations

Attacks

Conclusions and Future
Work

- Will the node really complete the jobs assigned to it?
- Will the node do that on time?
- Will the node provide correct results?
- Identity of the nodes?



Problem Statement

Introduction

Distributed Pervasive
Systems

Delegation Protocol

Problem Statement

Related work

The Trusting Protocol

Simulations

Attacks

Conclusions and Future
Work

- Will the node really complete the jobs assigned to it?
- Will the node do that on time?
- Will the node provide correct results?
- Identity of the nodes?



Introduction

Distributed Pervasive
Systems

Delegation Protocol

Problem Statement

Related work

The Trusting Protocol

Simulations

Attacks

Conclusions and Future
Work

- Some works on trust for grid systems:
 - rely on high connectivity and computational resources.
 - e.g., weighted feedback.
- PGP Web of Trust:
 - decentralized trust mechanism for identities.
- Trust for peer to peer networks:
 - e.g., a reputation mechanism for P2P file-sharing;

What we miss:

- dynamic update of the trust value of the nodes;
- decentralized system;
- a lightweight protocol.



Protocol Overview (1/2)

Introduction

The Trusting Protocol

Protocol Overview

Trust Value

Trust Value Update

Simulations

Attacks

Conclusions and Future Work

- *Reliability*: capability of nodes to respect service agreements;
- a node is *trusted* if it is reliable;
- reliability is evaluated before assigning tasks to nodes;
- reputation-based protocol.



Protocol Overview (2/2)

Introduction

The Trusting Protocol

Protocol Overview

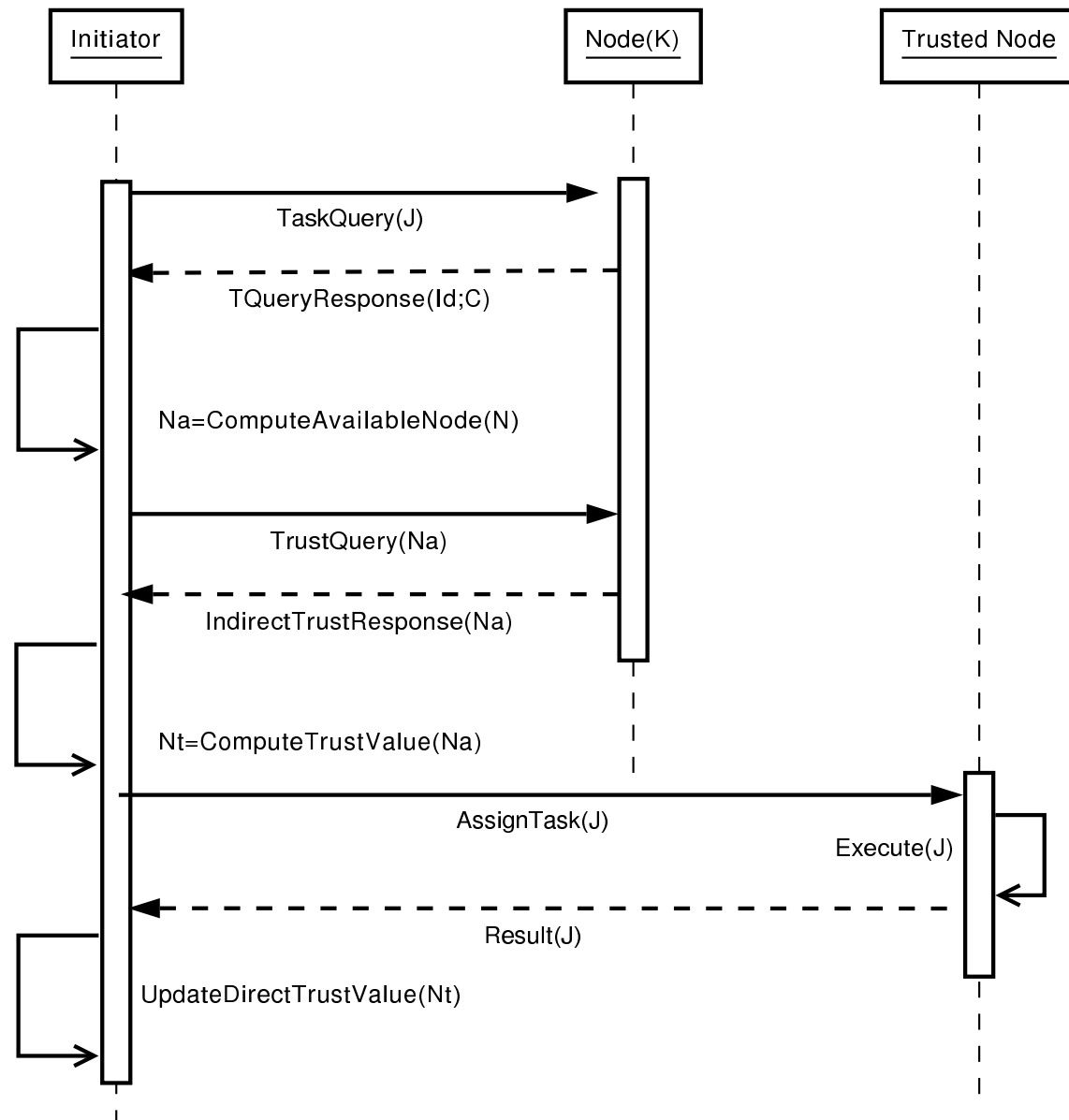
Trust Value

Trust Value Update

Simulations

Attacks

Conclusions and Future Work



Introduction

The Trusting Protocol

Protocol Overview

Trust Value

Trust Value Update

Simulations

Attacks

Conclusions and Future Work

Computed:

- By the initiator before assigning the task;
- by using the past experiences with the nodes considered;
- by mixing:
 - personal experience;
 - indirect experience.

$$T_j^d = w_p \times V_j^d + w_i \times \frac{\sum_i V_j^i}{m} \quad i \neq j; \quad i \neq d$$

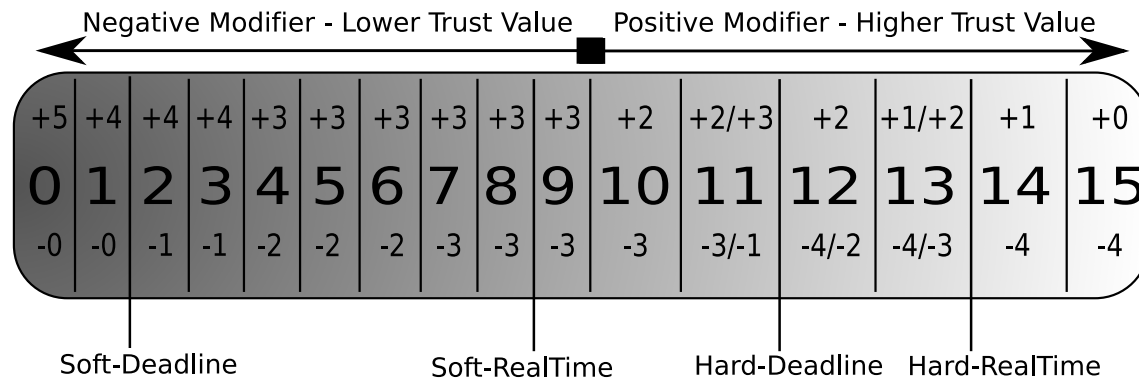


Trust Value Update

Personal trust value is updated after each task execution:

- based on the outcome of the execution:

$$V_j^d(t') = V_j^d(t) + M$$



- Introduction

- The Trusting Protocol

- Protocol Overview
- Trust Value
- Trust Value Update

- Simulations

- Attacks

- Conclusions and Future Work

Introduction

The Trusting Protocol

Simulations

Simulation Model

Simulation Results

Attacks

Conclusions and Future Work

- A SystemC model:
 - it models the delegation process;
 - it models the evolution of trust.

$$T_j^d = 0.7 \times V_j^d + 0.3 \times \frac{\sum_i V_j^i}{m} \quad i \neq j; \quad i \neq d$$



Simulation Results (1/2)

Introduction

The Trusting Protocol

Simulations

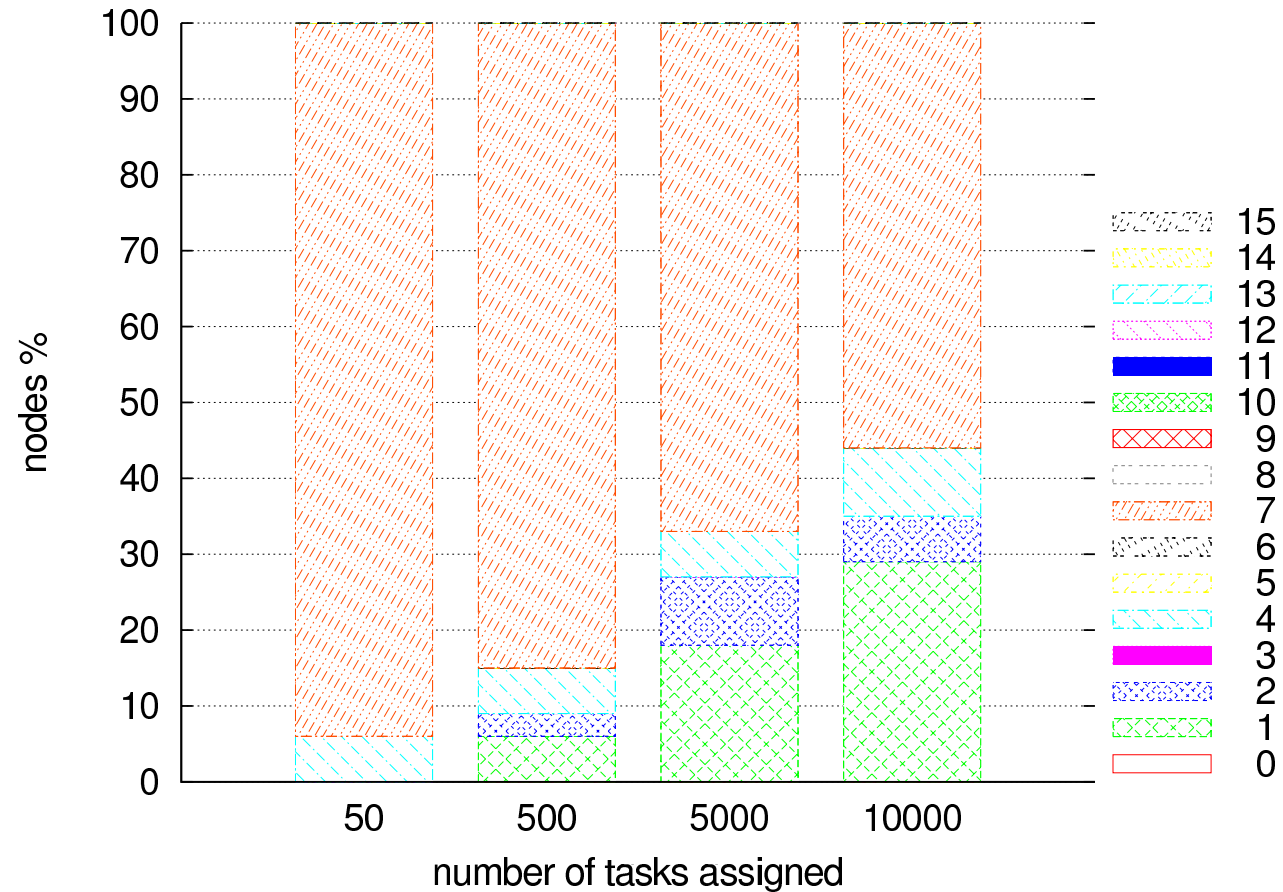
Simulation Model

Simulation Results

Attacks

Conclusions and Future Work

100% faulty node



58 tasks awarded to the faulty node during the whole simulation



Simulation Results (2/2)

Introduction

The Trusting Protocol

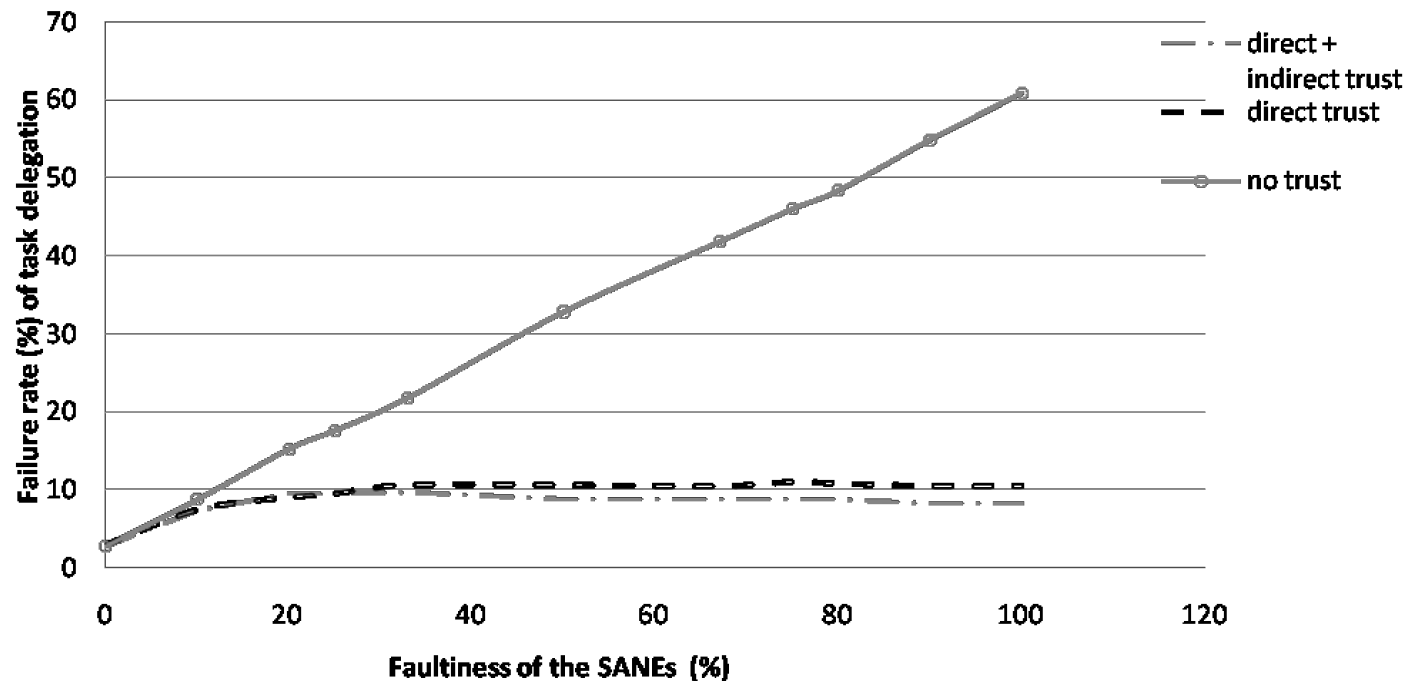
Simulations

Simulation Model

Simulation Results

Attacks

Conclusions and Future Work



- 50% of nodes are 10% faulty
- the others have different levels of faultiness

Attack Description

Introduction

The Trusting Protocol

Simulations

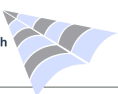
Attacks

Attack Description

Results

Conclusions and Future Work

- *Bad Mouthing*
nodes provide wrong recommendations.
- *Sybil*
nodes fake their identity.
- *On-off*
alternate behavior of a node.
- *Conflicting behavior*
node behaves in different ways depending on the peer.



Introduction

The Trusting Protocol

Simulations

Attacks

Attack Description

Results

Conclusions and Future Work

- Our protocol provides protection against:
 - Bad mouthing attack:
 - many malicious nodes (30% in simulations) are required to degrade performance of the system.
 - On-off attack:
 - a node behaving well in 50% of the cases quickly reaches a trust value of 7.



Introduction

The Trusting Protocol

Simulations

Attacks

Attack Description

Results

Conclusions and Future Work

- The conflicting-behavior attack only influences systems in which we also rate the reliability of nodes in providing recommendations.
- The Sybil attack can be prevented by using an identity check mechanism.



Introduction

The Trusting Protocol

Simulations

Attacks

Conclusions and Future Work

Conclusions

Future Work

We have proposed a protocol that:

- provides the ability to evaluate trust of the nodes;
- provides protection against some well known attacks.



Introduction

The Trusting Protocol

Simulations

Attacks

Conclusions and Future Work

Conclusions

Future Work

- Study the network overload due to the protocol;
- better study of the coefficient of the trust formula;
- study different forms of the trust formula;
- better study the modifier values;
- include identity checking and evaluate its overload.

