

Reputation system - STGDH

Julien Thomas

ENST Bretagne, RSM team

May 4, 2007

Outline

- 1 Goals
- 2 Existing approaches
- 3 Group decision principle
- 4 Reputation evaluations
- 5 Recommendation evaluation
- 6 Message exchange
- 7 Conclusion

- 1 Goals
- 2 Existing approaches
- 3 Group decision principle
- 4 Reputation evaluations
- 5 Recommendation evaluation
- 6 Message exchange
- 7 Conclusion

- Distributed group management: important to have a way to evaluate other nodes behavior
 - quality aspect of a group management operation
 - decisions have to be taken
- Desired reputation: shared among each nodes
 - not node-dependent

Outline

- 1 Goals
- 2 Existing approaches**
- 3 Group decision principle
- 4 Reputation evaluations
- 5 Recommendation evaluation
- 6 Message exchange
- 7 Conclusion

Existing approaches

- two main studies
 - Reputation and recommendation in the whole group
 - Reputation on local groups
- illustrate the general ideas found in the litterature
- are the most explicit studies

2 Existing approaches

- Reputation and recommendation in the whole group
- Reputation on local groups
- Performances comparisons

Existing approaches - In the whole group

- Reputation and recommendation system (bibliography)
- Jinshan Liu and Valérie Issarny

$SRep_a(o)^t$	node o 's reputation, declared by a , at time t
$RRep_a(o)^t$	o 's recommendation, about a , at time t
$SExp_a(o)^t$	Immediate experience of a about o
$Rec_a(o)^t$	Recommendation made by a about o , at time t for a correct node, $Rec_a(o) = SRep_a(o)$
ρ_e, ρ_c	information pondering

- Reputation evolution

$$SRep_a(o)^t = \rho_e \cdot SExp_a(o)^t + (1 - \rho_e) \cdot \frac{\sum_p (RRep_a(p) \cdot Rec_p(o))}{\sum_p RRep_a(p)}$$

- Recommendation evolution

- $a :: diff_1(p) = |Rec_p(o) - SExp_a(o)|$
- $diff = \frac{1 - diff_1}{\delta_a}$, δ_a : variation threshold
- $RRep_a(p)^t = RRep_a(p)^{t'} \cdot \rho_c^{(t-t')} + diff \cdot (1 - \rho_c^{(t-t')})$

2 Existing approaches

- Reputation and recommendation in the whole group
- Reputation on local groups
- Performances comparisons

- Conrad and al
- $reputation(c) = experience(c) \cdot p + (1 - p) \cdot hearsay(c)$
- $p = selfConfidence(c)$
- $experience(c) = \frac{immediateExperience(c) + experience(c)}{2}$

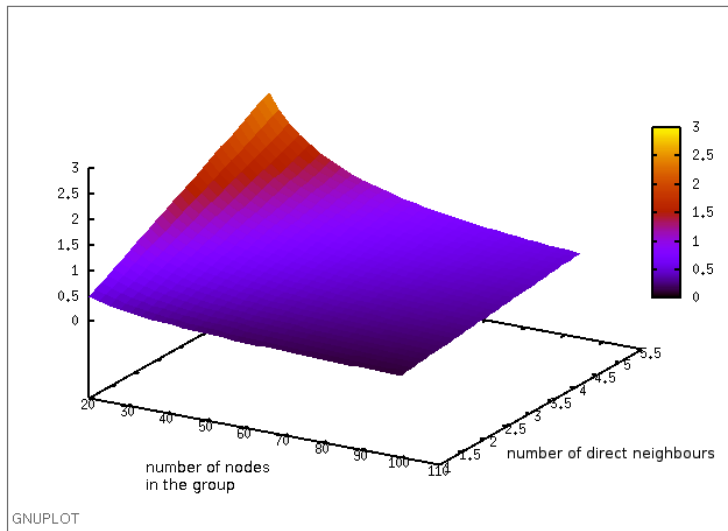
- Reputation : referees R
- $hearsay(c) = \frac{\sum_{r \in R} trust(r)}{|R|}$
- Simulations:
 - $|R| = 10$
 - $selfConfidence(c) = 30\%$
- number of referees, $selfConfidence$ value and reputation algorithm:
linked
 - but theoretical explanations are not given

2 Existing approaches

- Reputation and recommendation in the whole group
- Reputation on local groups
- Performances comparisons

- first study : scalability problems
 - Consider the following example:
 - detection mechanism: only on direct neighbors (often for lowest levels of the TCP/IP stack)
 - N nodes, each node has k neighbors
 - best case, as $k \ll N$: $rec(o) = 100\%$
 - $rep_t(i) = rep_{t-1}(i) \cdot ns_{rep} + \frac{1 - ns_{rep}}{N} \cdot (\sum_{j \in K} (rep_{t-1}(i, j) - \beta) + \sum_{j \notin K} rep_{t-1}(i, j) + \alpha)$

Existing approaches - Performances comparisons



- scalability problems
 - decrease rate of 3 when $k = 5$, $N = 20$
 - decrease rate of 0 when N increases
- second study
 - $hearsay(c) = \frac{\sum_{r \in R} trust(r)}{|R|}$
 - cannot be used to detect malicious nodes inside the network (each node: same reputation)
 - how can we evaluate R ?
 - What are the impact of the size of R in the reputation mechanism?

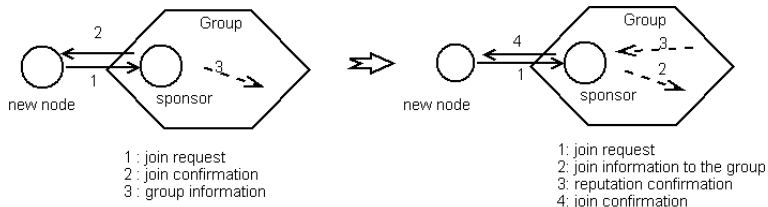
- 1 Goals
- 2 Existing approaches
- 3 Group decision principle**
- 4 Reputation evaluations
- 5 Recommendation evaluation
- 6 Message exchange
- 7 Conclusion

- 3 Group decision principle
 - Group operations
 - Group agreements

- Group operations: operations made by a single node
- Adding a node
 - n_i sends an adding message if $local_reputation \geq threshold_{Add}(i)$
 - several criteria, such as:
 - reputation, described in the following sections
 - group properties (routing criteria as connectivity, group proximity, power)
 - personal properties (notion of proximity, power (energy, transmission range), personal knowledge they share)

- Adding a node

- requires reputation analysis and thus behavior analyses
- node wants to join a group, it must first enter in a *neutral zone*
- most of the ad hoc routing protocol nowadays signature authentication



- Removing a node
 - can be node-dependent
 - but malicious node provokes impacts on the whole group
 - $threshold_{Evict}(i) = reputation(i)$

- 3 Group decision principle
 - Group operations
 - Group agreements

Group agreements

Property 1

In order to start an adding operation, a node must have receive τ_A adding message from distinct nodes among the network.

Property 2

In order to start an eviction, a node must have receive τ_L eviction message from distinct nodes among the network.

Property 3

A node message should be taken into account only if its group reputation is good enough.

Property 4

The difference between $threshold_{Add}(i)$ and $threshold_{Evict}(i)$ must be sufficient to prevent from having an unstable management (repetitive adding and leaving operations).

Property 5

Upon receiving a group management operation, each node of the group must take the same decision.

Outline

- 1 Goals
- 2 Existing approaches
- 3 Group decision principle
- 4 Reputation evaluations**
- 5 Recommendation evaluation
- 6 Message exchange
- 7 Conclusion

- 4 Reputation evaluations
 - Our reputation principle
 - Reputation evolution: main schemes
 - Reputation increases
 - Reputation decreases
 - Referees area and layers considerations

Our reputation principle

- similar decision on the whole group (Property 5) \Rightarrow global reputations
 \Rightarrow group recommendation, not node-dependent recommendation

- $$\text{group_reputation}_t(i) = \frac{\sum_{j \in R_i} \text{rec}(j) \cdot \text{rep}_t(i,j)}{\sum_{j \in R_i} \text{rep}_t(i,j)}$$

- $$\text{rec}_t(i) = \text{rec}_{t-1}(i) \cdot ns_{rec} + (1 - ns_{rec}) \cdot \frac{\sum_{j=0}^n \text{diff}(\text{rep}_{t-1}(j,i), \text{group_reputation}_t(j))}{n}$$

- $$\text{rep}_t(i) = \frac{\sum_{j \in R_i \text{ and } j \neq \text{myself}} \text{rec}(j) * \text{rep}_{t-1}(i,j) + 100 \cdot \text{experiences}}{\sum_{j \in R \text{ and } j \neq \text{myself}} \text{rec}(j) + 100}$$

- but .. variables values?

- 4 Reputation evaluations
 - Our reputation principle
 - Reputation evolution: main schemes
 - Reputation increases
 - Reputation decreases
 - Referees area and layers considerations

Numerous studies: two principal parameters in the reputation evolution mechanism:

- $\forall n_c, n_i$ acts well between: increases its reputation of α ,
 $n_c : rep(n_i) = n_c : rep(n_i) + \alpha$
- *if act well* reflects two cases
 - really act well
 - n_c cannot detect the bad behavior
- bad behavior, decrease its reputation of β

- what are the impacts the different parameters have?
- study the worst cases \Rightarrow we simplify the reputation model:
 - $rep_t(i) = \frac{\sum_{j \in R} rec(j) \cdot rep_{t-1}(i,j) + 100 \cdot personal_experiences}{\sum_{j \in R} rec(j) + 100}$
 - $ns_{rep} = 0$, reputation mechanism : more affected by malicious nodes attacks.

- 4 Reputation evaluations
 - Our reputation principle
 - Reputation evolution: main schemes
 - **Reputation increases**
 - Reputation decreases
 - Referees area and layers considerations

Property 1

The collusion of malicious nodes must not engender an eviction of a correct node.

Property 2

The increase rate must not be too important, in order to prevent malicious nodes from recovering correct reputation.

Reputation increases

- Worst case 1
 - $\tau - 1$ nodes act badly: they declare a reputation of 0
 - the others increase the reputation value of α
- First property: $\tau_{eviction}$ ($\tau_{eviction} > \tau$) nodes send an eviction message
 - at least a correct node: send an eviction message
 - reputation exceed the eviction threshold
- $rep_t = \frac{0 \cdot (\tau - 1) + (rep_{t-1} + \alpha) \cdot (|R| - \tau + 1)}{|R|}$
- $V_n = rep_t = V_0 \cdot a^n + b \cdot \sum_{i=0}^{n-1} a^i$ where
 $a = \frac{|R| - \tau + 1}{|R|}$ and $b = \alpha \cdot \frac{|R| - \tau + 1}{|R|}$

- Worst case 1

- suppose we have $V_k = Evi_{c_{threshold}}$,
- $\alpha_{min} / V_{k+1} \geq Evi_{c_{threshold}}$
- $(Evi_{c_{threshold}} + \alpha) \cdot \frac{|R| - \tau + 1}{|R|} > Evi_{c_{threshold}}$ or $\alpha > Evi_{c_{threshold}} \cdot \frac{1-a}{a}$.

$Evi_{c_{threshold}}$	α_{min}	$Evi_{c_{threshold}}$	α_{min}
10	4	30	10
20	7	40	14

Table: minimal value of α depending on $Evi_{c_{threshold}}$, $|R| = 4 \cdot \tau$

- Worst case 1
 - second requirement,
 - $V_n = rep_t = V_0 \cdot a^n + \alpha \cdot a \cdot n$, as $a \leq 1$
 - $\alpha = 4$, $V_0 = 50$, $|R| = 2 \cdot \tau$
 - $\alpha = 4$, $V_0 = 50$, $|R| = 4 \cdot \tau$
 - $\tau = 20$, $V_0 = 50$, $|R| = 4 \cdot \tau$
 - $|R| = 4 \cdot \tau$ for several reasons.
 - increase rate quite correct
 - $|R| = 6 \cdot \tau$ or $|R| = 2^T$: better results
 - but the size of $|R|$ increased very quickly

- Worst case 2
 - all of the malicious nodes : 100 of reputation, to quickly increase a node's reputation
 - $rep_t = \frac{100 \cdot (\tau - 1) + (rep_{t-1} + \alpha) \cdot (|R| - \tau + 1)}{|R|}$
 - which value of α engenders a correct reputation increase ?
- reputation of the bad node: evolves very quickly, no matter the value of τ
 - $|R| = 4 \cdot \tau$: fives iterations, starting from a value of 50
 - three iterations for $|R| = 2 \cdot \tau$

Reputation increases

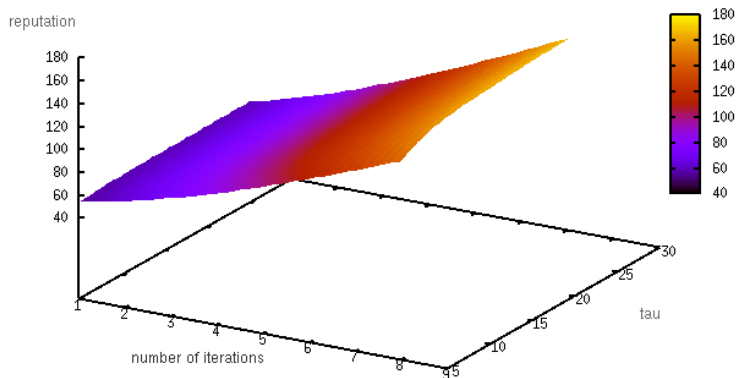


Figure: Reputation increase - maximal increase rate with $\alpha = 4$

- Common case

- $rep_t = \frac{(rep_{t-1} + \alpha) \cdot |R|}{|R|} = (rep_{t-1} + \alpha)$

- increase rate easy to evaluate: α

- 13 iterations are needed to get a maximal reputation, starting from a reputation of 50

- Conclusion

- 4 for α and a value of $4 \cdot \tau$ for $|R|$ are interesting

- 5: worst minimal value, with no recommendation decreases

- assumption made to simplify the equations.
 - increase rate lower in reality

- solution: prevent bad nodes from having malicious behavior

- diminishing correctly their reputation

- 4 Reputation evaluations
 - Our reputation principle
 - Reputation evolution: main schemes
 - Reputation increases
 - **Reputation decreases**
 - Referees area and layers considerations

Property 1

A collusion of malicious nodes must not prevent a malicious nodes from having a decrease of its reputation.

Property 2

The group must be able to evict a malicious node, when it exceeds a defined threshold.

Reputation decreases

- When a node n_m act maliciously, its reputation must decreases
- All the malicious nodes cooperate to prevent this decrease: reputation of 100
- We have to choose $|R|$ and β such that
 - the reputation is still able to decrease if a node often has a bad behavior.
 - decreases in a significant way the malicious node reputation,
 - increase the recovering time of this node.
- With β as a constant:
$$rep_t = \frac{100 \cdot (\tau - 1) + (rep_{t-1} - \beta) \cdot (|R| - \tau + 1)}{|R|}$$
 , where $rep_0 = 100$

Reputation decreases

decrease (V_1)	value of β	iterations to recover
10	7	3
20	25	5
30	43	8
40	60	10

- with $\beta = 25$
 - 7 iterations before obtaining the maximal reputation
 - acts maliciously during each reputation update intervals
 - reputation of 50 iterations after 5 iterations
 - reputation of 30 after 15 iterations
- drawback: may not be able to get a reputation of $threshold_{Evict}$
- a special case cannot be taken into account: node acts badly, waits, restarts to act badly (Moral Hazard)
 - requires a group history

- Ebay's reputation mechanism
 - trust of a node: correlation between positive rates (PR) and negative rates (NR)
 - $Trust = \beta_0 + \beta_1 \cdot \text{Log}(PR) + \beta_2 \cdot \text{Log}(NR)$
- In our case: limit for malicious node is NR_{max}
- $\beta(NR) = \beta_0 + f(NR) \cdot \beta_1$.
- choice of the different parameters (f , β_0 , β_1 and NR_{max}) engenders effects on:
 - how quickly a node can be evicted
 - how evolve the node's reputation

The principal scheme of f is that:

- $f(0) = 0$
- $f(NR_{max}) = 100/\beta_1 - \beta_0$ (as $\beta(NR_{max}) = 100$)

- $beta(NR) = \frac{100}{2^{NR_{max}}} \cdot 2^N R$

■picture■

- 4 Reputation evaluations
 - Our reputation principle
 - Reputation evolution: main schemes
 - Reputation increases
 - Reputation decreases
 - Referees area and layers considerations

- R depends on several criteria.
 - other mechanism variables
 - protocol nature:
 - routing layer: checks are geographically-based
 - application layer: cryptotree-dependent, region-independent
- R may not be maximized by N but by a subset of it
- the maximal number of malicious nodes supported may diminish

Our case: group management, application-layer

- first case: actual update mechanism of S-TGDH
 - only the subtree whose root is the considered node's parent can perform check
 - $R = 4 \cdot \tau$
 - cryptotree structure: leaves problem
 - $\tau_{max} = 1$
 - the more the group size is important, the more nodes can be attacked

- second case: ameliorate the reputation detection: bound R to N
 - each node of the group receive each update message
- current design of S-TGDH, messages to the subtree whose root is the parent of the considered node
 - not possible to create a multicast address for each subtree of the group
 - each message is sent to each node of the subtree, in a unicast mode
 - n message per join operations
- group message: unique multicast address for the group
 - h multicast messages for a join operations, where $n = 2^{h+2} - 1$
 - propagated using the multicast aspect of the ad hoc routing protocol
 - limit the network overhead

Outline

- 1 Goals
- 2 Existing approaches
- 3 Group decision principle
- 4 Reputation evaluations
- 5 Recommendation evaluation**
- 6 Message exchange
- 7 Conclusion

- 5 Recommendation evaluation
 - Recommendation properties
 - Results comparisons

- prevent malicious nodes from interfering with correct information about a tierce node
 - their recommendation must decrease if they are erroneous.
 - stability value: 0 for malicious nodes

Property 1

a node recommendation must decrease if it acts maliciously

Recommendation function

- Existing function:

$$rec_t(i) = rec_{t-1}(i) * ns_{rec} + (1 - ns_{rec}) * \frac{\sum_{j=0}^n diff(rep_{t-1}(j,i), our_reputation(j))}{n}$$

- $our_reputation(j)$: $group_reputation_t(j)$ or our own reputation evaluation
- function linked to the size of the group, as for the reputation.
- simple attack: centralize attacks on a single node.
- recommendation would not decrease that much
- stabilized state example :
 - malicious nodes recommendation: 94%
 - attacked node's reputation: 12% or 30%, depending on τ
 - first property is not respected.

Recommendation function

- $rec_t(i) = rec_{t-1}(i) * ns_{rec} + (1 - ns_{rec}) \cdot \frac{\prod_{j=0}^n diff(rep_{t-1}(j,i), group_reputation_t(j))}{n}$
- intelligent malicious nodes, drawbacks can be found:
 - decrease rate: associated to *diff*
 - sending reputation that are slower than the group_reputation, but not too far
 - advanced attacks such as a binary state correct, malicious can have impact on the reputation mechanism
- attacks: more sophisticated, reputation rates: better
 - but still affected by malicious nodes collusions.

Recommendation function

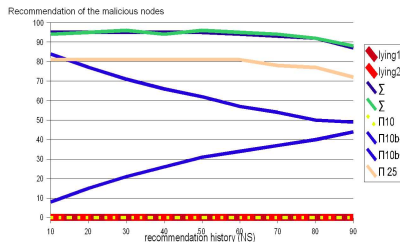
- In our case, we made a strong assumption :
 - R such that $R = 4 \cdot \tau$
 - $\tau - 1$ being the maximal number of malicious nodes
 - at least 75% of the nodes among R are not malicious
- each node among R is able to detect if a node is acting maliciously
- majority of the reputations is correct
 - compare a node recommendation with the majority value

$$rec_t(i) = rec_{t-1}(i) * ns_{rec} + (1 - ns_{rec}) \cdot lieValue$$

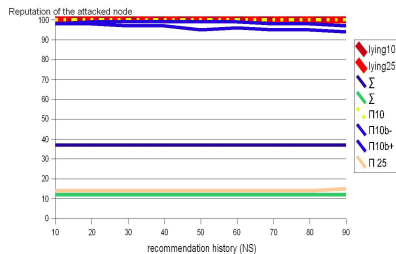
$$lieValue(n) = \begin{cases} 0 & \text{if } \exists i \in R / rep_t(i, n) \neq majority_reputation_t(i) \\ 100 & \text{otherwise} \end{cases}$$

- 5 Recommendation evaluation
 - Recommendation properties
 - Results comparisons

Results comparisons



Recommendation value of the malicious nodes



Reputation value of the attacked node

- first property is not assured by the standard recommendation evaluation
 - recommendation is superior to 85%, even with $\tau = 10\%$
 - attacked node reputation is really affected (12% or 37%)
- update recommendation evaluation (Π):
 - recommendation evaluation and the reputation evaluation:
 - correct in the case of basic malicious nodes, with $\tau = 10\%$
 - extreme case of $\tau = 25\%$, the attacked node's reputation is affected
 - advanced malicious nodes, the attacked node's reputation is not really affected and the malicious nodes' recommendations are neither good nor bad

- lying method provides really good results
 - malicious nodes recommendations are always null
 - number of iterations needed to obtain the stabilized state is in the weakest ones

Outline

- 1 Goals
- 2 Existing approaches
- 3 Group decision principle
- 4 Reputation evaluations
- 5 Recommendation evaluation
- 6 Message exchange**
- 7 Conclusion

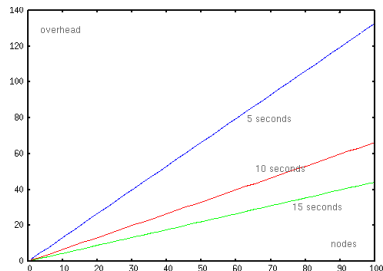
- $overhead = nb_{messages} * size(message) = nb_{messages} \cdot (< sub_layers_headers > + reputations_by_message \cdot packetsize)$

	$< sub_layers_headers >$	global overhead with $\tau = 10\%$
OLSR:	58 bytes	$n \cdot (58 + n \cdot 34)$
AODV:	60 bytes	$n \cdot (60 + n \cdot 34)$

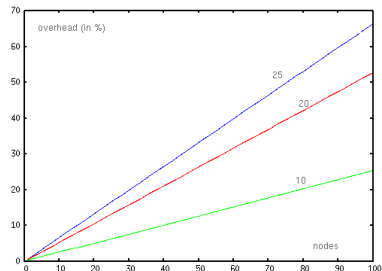
- classical comparisons: CBR model at 512 bytes.
- $increase_{rate} = \frac{60+n \cdot 34}{512 \cdot seconds_{intervals}}$

Traffic overhead

- TC messages: 5 seconds
- $k \cdot 5\text{seconds}$: add reputation messages in TC message
- $increase_{rate} = \frac{n \cdot 34}{512 \cdot seconds_i \cdot intervals}$



5 ,10 and 15 seconds intervals, $\tau = 25\%$



impact of τ , interval of 10 seconds

Outline

- 1 Goals
- 2 Existing approaches
- 3 Group decision principle
- 4 Reputation evaluations
- 5 Recommendation evaluation
- 6 Message exchange
- 7 Conclusion**

- designing a reputation and recommendation mechanism at the group layer:
 - complex problem
 - requires a shared reputation between the nodes, as a group reputation not local reputation
- this kind of system relies on many parameters
 - such as update rates, synchronisation intervals and thresholds
 - linked together in complex ways

- Based on a mathematic approach, we have
 - defined basic security properties
 - parameters values
- recommendation aspect is a far important
 - basic aspect: *a node recommendation must decrease if it acts maliciously*
 - not assured in extreme cases: engenders incorrect stabilized states
 - two updates of this evaluation