Preventing Selfish Behavior in Ad Hoc Networks

Marcin Seredynski
Pascal Bouvry
Mieczyslaw A. Klopotek
Our contribution

1. Game theoretical model of ad hoc networks

2. Strategy based cooperation enforcement approach
Plan of the presentation

- introduction to ad hoc networks
- definition of a selfish behavior
- our approach to the problem
- results
- conclusions

Marcin Seredyński, Pascal Bouvry, Mieczysław A. Klopotek

September 28, 2007
Ad hoc networks: introduction
Network properties

Basis requirements
- mobile device
- wireless network card

Network properties
- no additional infrastructure, all nodes are equal
- each node acts as a router (forwarding packets for others) and as terminal.
- nodes can be mobile $\Leftrightarrow$ frequent changes of topology
- limited resources: computational power and battery

Communication
a) **direct communication:** possible when nodes are within their radio range. Example: node A and B.
b) **multihop communication:** when nodes do not “see” each other they should use intermediate nodes. Example: node B and E can communicate with each other using node D
Watchdog mechanism - controlling the behavior of its neighbors
Example: node A can actually overhear the communication between node B and C
Sending various types of packets

**role 1**: node B acting as terminal

**role 2**: node B acting as router
Selfish behavior
What causes selfish behavior

Limited battery resources

- At some point **power saving** becomes an important issue for each network participant.
How to save energy (behave selfishly)?

- **Power consumption** of a typical IEEE 802.11 wireless network card

Four **modes** available:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Consumption (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit</td>
<td>1300</td>
</tr>
<tr>
<td>Receive</td>
<td>960</td>
</tr>
<tr>
<td>Idle</td>
<td>840</td>
</tr>
<tr>
<td>Sleep</td>
<td>66</td>
</tr>
</tbody>
</table>

When being in a sleep mode node is unavailable for any service requests

Two ways of saving the energy:

- **discard intermediate packets** (idle mode instead of transmit mode)
- **sleep mode** (instead of idle mode)
Why is selfish behavior bad for the network?

The correct operation of the network requires **not only** the correct execution of critical network functions by each participating node but it also requires that each node performs **a fair share of the functions**.

Reliable network requires **cooperation** of its participants.

Selfishness is a **rational behavior** in an environment free of any cooperation enforcement mechanism.

**No classical security mechanism can help to counter a misbehaving node in this context.**
Cooperation vs. Security
Cooperation vs. Security

- Cooperation and security are closely related issues:
  - **Secure network** is robust against attacks (malicious users wanting to jeopardize the network)
  - **Strategy proof network** is robust against strategic behavior of users which want to exploit the network: *cooperation is the most beneficial choice for the users*
Security related issues

Security issue

- Security objectives
  - Confidentiality
  - Data integrity
  - Non-repudiation
  - Authentication

- Cryptographic tools
  - Symmetric-key primitives
  - Public-key primitives

- Intrusion detection system
  - Key management

- Intrusion detection

Cooperation issues

- Second hand information
- First hand information

- Monitoring techniques
- Reputation evaluation

- Information exchange between nodes

- Identity trust
  - Trust based on peer knowledge
    - "I know you"
    - (Verification required)

- Behavioral trust
  - Trust based on observation
    - "You behave correctly"
    - (Monitoring required)

- Cooperation enforcement

Goals

Marcin Seredynski, Pascal Bouvry, Mieczyslaw A. Klopotek

September 28, 2007
Our solution
Cooperation enforcement

Solution: enforce cooperation using the notion of trust and activity

Node has follow these steps:

1. Each node is collecting reputation data concerning the behavior of network participants

2. Using reputation data, trust and activity of other known nodes can be calculated

3. When it receives a packet that should be forwarded, first it checks the trust and activity levels of the source of the packet (original sender) and then makes a decision according to some predefined strategy

Marcin Seredynski, Pascal Bouvry, Mieczyslaw A. Klopotek

September 28, 2007
Definition of trust

- **Trust** is a particular level of the *subjective probability* with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently or his capacity ever to be able to monitor it) and in a context in which it affects his own action [D. Gambetta].

\[
\text{Trust} = \frac{\text{number of our packets forwarded by particular node}}{\text{number of our packets received by particular node}}
\]

The obtained ratio is exchanged into one of four predefined trust values (trust lookup table), from 0 (complete distrust) to 3 (complete trust).
Definition of activity

- Activity describes how often a node is available for routing purposes (idles mode vs. sleep mode)

\[
\text{Activity} = \frac{\text{number of all packets forwarded by particular node}}{\text{average number of packets forwarded by nodes in the network}}
\]

- The obtained ratio is exchanged into one of three possible activity values (activity lookup table), from LO (low activity) to HI (high activity)

- Nodes joining the network or nodes spending a lot of time in sleep mode will have lower activity levels.
Strategy

Decisions:
- D: discard intermediate packet
- F: forward intermediate packet

Activity levels:
- LO - low
- MI - medium
- HI - high

Trust levels and activity levels:
- trust 0: LO MI HI
- trust 1: LO MI HI
- trust 2: LO MI HI
- trust 3: LO MI HI

Decisions against a forwarding request coming from a known player:
- trust 0: D
- trust 1: D
- trust 2: D
- trust 3: F

Decision against a forwarding request coming from an unknown player:
- trust 0: F
- trust 1: F
- trust 2: F
- trust 3: F
Several preferences do be defined

- Before finding a solution players’ preferences (goals) have to be defined

Modeling preferences of network participants

All we care about, our goals...

- number of packets to be send
- approach towards battery saving
- social relations with others
- battery state
- joining and leaving the network conditions when leaving...
- network liveliness

Players having the same preferences belong to the same class of players
We have defined a strategy so next:

Evaluation and evolution of strategies
Evaluation phase
Several types of players compete in a single network

Evolution phase
Players belonging to the same class evolve using a genetic algorithm (GA) independently of other classes
Evaluation of strategies (evaluation phase)

Tournament is a sequence of games (described latter)

In each game players receive payoffs that depend on their decisions.
Parallell evolution of strategies (evolution phase)

An example:

two populations (A and B) and two kinds of special (SP/LAP)

1: load players to the tournament
2: play ad hoc games
3: evaluate strategies of the population A and B
4: using GA operators new populations are created (selection + reproduction)
Game and payoffs
Game based model of the network

- Similar to iterated **Prisoner's Dilemma under the Random Pairing game**
- Nodes receive **payoffs** corresponding to the decisions made
- As a result strategies can be evaluated (better strategies receive higher payoffs during subsequent games)

N. Namikawa and H. Ishibuchi, “Evolution of cooperative behavior in the iterated prisoner’s dilemma under random pairing in game playing,” in *Congress on Evolutionary Computation (CEC’05)*
Definition of the game

**Game:** represents packet delivery process

GP - game participants
A - source of the packet
B, C intermediate nodes asked to forward the packet

Each decision is followed by a **payoff** corresponding to the decision made
After each game, its participants receive payoffs according to Table a and Table b. Additionally at the end of the tournament all participants receive payoffs for spending time in sleeping mode (Table c.)

### Payoff tables

<table>
<thead>
<tr>
<th>a) sending own packets</th>
<th>b) forwarding</th>
<th>c) sleeping</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>transmission status</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>success (S)</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>failure (F)</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>reliability of the src node</th>
<th>payoff</th>
<th>mode of the interface in a round</th>
<th>payoff (per round)</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust</td>
<td>activity</td>
<td>forward(F)</td>
<td>discard(D)</td>
</tr>
<tr>
<td>HI</td>
<td>14</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ME</td>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>LO</td>
<td>3</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>mode of the interface in a round</td>
<td>sleep</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>idle</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Types of players

<table>
<thead>
<tr>
<th>Types of players</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal (NP)</td>
<td>plays according to the strategy</td>
</tr>
<tr>
<td>low activity (LAP)</td>
<td>sleeps half of the time, always forwards</td>
</tr>
<tr>
<td>selfish (SP)</td>
<td>always awake (idle mode), newer forwards</td>
</tr>
</tbody>
</table>

Two classes of normal players (NP):

**Class 1:**
Player goes to sleep mode when more than 90% of its own packets reach the destination

**Class 2:**
Player goes to sleep mode when more than 40% of its own packets reach the destination
Tournament settings

<table>
<thead>
<tr>
<th></th>
<th>Experiment 1</th>
<th>Experiment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>tournament size</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>population A size</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>population B size</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>number of SP</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>number of LAP</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>
The evolution of cooperation

Successful games:
- Average: 33%
- Population A: 53%
- Population B: 32%
- LAP: 20%
- SP: 3%

Parameter values:
- Tournament size: 70
- Population A size: 50
- Population B size: 50
- Number of SP: 10
- Number of LAP: 10
### Evolved sub-strategies for the population A.

<table>
<thead>
<tr>
<th>Trust 0 (31%)</th>
<th>Trust 1 (60%)</th>
<th>Trust 2 (95%)</th>
<th>Trust 3 (86%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>011</td>
<td>011</td>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>001 (26%)</td>
<td>001 (26%)</td>
<td>111 (3%)</td>
<td>111 (13%)</td>
</tr>
<tr>
<td>010 (24%)</td>
<td>111 (7%)</td>
<td>001 (2%)</td>
<td>-</td>
</tr>
<tr>
<td>000 (18%)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Evolved sub-strategies for the population B.

<table>
<thead>
<tr>
<th>Trust 0 (28%)</th>
<th>Trust 1 (48%)</th>
<th>Trust 2 (68%)</th>
<th>Trust 3 (77%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>011</td>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>010 (27%)</td>
<td>011 (41%)</td>
<td>111 (16%)</td>
<td>001 (17%)</td>
</tr>
<tr>
<td>011 (23%)</td>
<td>101 (5%)</td>
<td>001 (9%)</td>
<td>111 (5%)</td>
</tr>
<tr>
<td>000 (22%)</td>
<td>010 (3%)</td>
<td>101 (5%)</td>
<td>-</td>
</tr>
</tbody>
</table>
Conclusions
Conclusions

- Reliable network requires an existence of a cooperation enforcement system
- We propose a game theoretic model of an ad hoc network and a strategy based approach towards packet forwarding
- Strategies were evolved using a GA
- Nodes were forced to cooperate and limit time spend in a sleep mode in order to be able to use the network for its own purposes

Marcin Seredyński, Pascal Bouvry, Mieczysław A. Klopotek  September 28, 2007
Thank you!

Questions?

marcin.seredynski@uni.lu
Ad hoc networks: a wireless (rather anonymous) network

Selfish behavior: temptation to save limited resources (battery...)

Cooperation enforcement mechanism: encourage nodes to cooperate
- Reputation derivation
- Trust computation
- Rate other network participants
- Monitor/exchange reputation data

Activity level computation
- Cooperate or deny service
- Strategy
- Finding the right strategy

Generic algorithm: a search tool

Game theoretical network model: simulation + evaluation

Summary
Reputation and Trust evaluation

Reputation collection

\[ fr(B,A) = \frac{pf_A}{ps_A} \]

\( fr(B,A) \) - forwarding rate of the node A
\( pf_A \) - number of packets forwarded by the node A
\( ps_A \) - number of packets sent to the node A

Trust level evaluation

<table>
<thead>
<tr>
<th>( fr )</th>
<th>TLBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 0.9</td>
<td>3</td>
</tr>
<tr>
<td>0.9 - 0.6</td>
<td>2</td>
</tr>
<tr>
<td>0.6 - 0.3</td>
<td>1</td>
</tr>
<tr>
<td>0.3 - 0</td>
<td>0</td>
</tr>
</tbody>
</table>

\( TLBA \) - highest trust level
\( TLBA \) - lowest trust level

packet dropped

src:

A

B

C

D

dst:

E

reputation update (about B,C,D)

reputation update (about C,D)

reputation update (about B,D)
Types of ad hoc networks

- **Time**
  - short term networks
  - long term networks

- **Pre-configuration**
  - Pure (spontaneous)
    - no assumed preconfiguration
    - network created on the fly
  - Managed
    - requires an on line trust infrastructure or
    - pre-configuration of the nodes.
    - suitable for planned, long term ad hoc networks

- PKI
  - Certificate/Revocation Repository
  - End entity
  - Registration Authority
  - PKI users
  - PKI components
  - Certification Authority
Security in Pervasive computing

Anonymous communication (pure ad hoc network)
- not named or identified
- little or no security

Non anonymous communication (managed ad hoc network)
- requires method (usually third party) to attest to the identity of a participant
- this requires both parties to agree on the trusted party and method

Q: What solutions should be used?
A: Deployment context has to be specified!
Selfish and malicious misbehavior

What causes non-cooperation?

- **Selfishness**: save Energy
- **Malice**: Damage the Network
- Need of power saving, temporal node/communication failures

PROBLEM: How to distinguish real need of *power saving* or *temporal failure* and *selfish* behavior, fault tolerance...?
Network example
Network properties: multi hop communication

Let’s suppose that node A wants to communicate with node M. It should do the following:

1. find a route (using a routing protocol)
2. send packets using the path found by the routing protocol.

In the source routing protocols sender specifies the full path to the destination. In the presented example, path from node M to node A includes nodes L-H-E.
Requests coming from normal (evolving nodes) versus nodes selfish by default

<table>
<thead>
<tr>
<th></th>
<th>forwarding requests from normal nodes (NN)</th>
<th>forwarding requests from selfish nodes (SN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req. accepted</td>
<td>77%</td>
<td>4%</td>
</tr>
<tr>
<td>Req. rejected by NN</td>
<td>0.23%</td>
<td>53%</td>
</tr>
<tr>
<td>Req. rejected by SN</td>
<td>22%</td>
<td>43%</td>
</tr>
</tbody>
</table>