DARWIN: Distributed and Adaptive Reputation Mechanism for Wireless Ad-hoc Networks

Xiaoqi Li, Gigi

CSE, CUHK

April 9, 2008

This talk is based on paper: [J.J.Jaramillo and R. Srikant. DARWIN: Distributed and Adaptive Reputation Mechanism for Wireless Ad-hoc Networks. In Proc. of ACM Thirteenth Annual International Conference on Mobile Computing and Networking (MobiCom’07), Montreal, Canada, Sept. 2007]
1 Introduction

2 Basic Game Theory Concepts

3 Network Model

4 Analysis of Prior Proposals
   - Trigger Strategies
   - Tit For Tat
   - Generous Tit For Tat

5 DARWIN
   - CTFT
   - Definition
   - Performance Guarantees
   - Algorithm Implementation

6 Simulations
   - Settings
   - Results

7 Conclusions
Nodes are born to be selfish in wireless networks.
Nodes are born to be selfish in wireless networks.

Incentive mechanisms are needed to enforce cooperation.
Nodes are born to be selfish in wireless networks.
Incentive mechanisms are needed to enforce cooperation.
Two types of incentive mechanisms:
1. Credit exchange systems: by payment
2. Reputation based systems: by neighbors’ observation
Nodes are born to be selfish in wireless networks.

Incentive mechanisms are needed to enforce cooperation.

Two types of incentive mechanisms:

1. **Credit exchange systems**: by payment
2. **Reputation based systems**: by neighbors’ observation

Main issue concerned in this reputation system:

Nodes can be perceived as being selfish falsely due to silently dropping collision packets.
Introduction

- Nodes are born to be selfish in wireless networks.
- Incentive mechanisms are needed to enforce cooperation.
- Two types of incentive mechanisms:
  1. Credit exchange systems: by payment
  2. Reputation based systems: by neighbors’ observation
- Main issue concerned in this reputation system:
  Nodes can be perceived as being selfish falsely due to silently dropping collision packets.
- Contributions:
  1. Analyze prior reputation strategies’ robustness
  2. Propose a new reputation strategy and testify it
# The Prisoners’ Dilemma Game

## Table 1: Payoff Matrix of the Prisoners’ Dilemma Game

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>1, 1</td>
<td>-1, 2</td>
</tr>
<tr>
<td>Defect</td>
<td>2, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

A Nash equilibrium is a strategy profile having the property that no player can benefit by unilaterally deviating from its strategy.

Repeated Prisoners' Dilemma Game:

Total payoff function is the discounted sum of the stage payoffs:

Question: What's the NE of Repeated Prisoners' Dilemma Game?
The Prisoners’ Dilemma Game

A Nash equilibrium is a strategy profile having the property that no player can benefit by unilaterally deviating from its strategy.

Table 1: Payoff Matrix of the Prisoners’ Dilemma Game

<table>
<thead>
<tr>
<th></th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cooperate</td>
</tr>
<tr>
<td>Player 1</td>
<td>Cooperate</td>
</tr>
<tr>
<td></td>
<td>Defect</td>
</tr>
<tr>
<td></td>
<td>Defect</td>
</tr>
</tbody>
</table>
The Prisoners’ Dilemma Game

A Nash equilibrium is a strategy profile having the property that no player can benefit by unilaterally deviating from its strategy.

Repeated Prisoners’ Dilemma Game:
Total payoff function is the discounted sum of the stage payoffs:

\[ U_i = \sum_{k=0}^{\infty} w^k u_i^{(k)} \]

Question: What’s the NE of Repeated Prisoners’ Dilemma Game?

Table 1: Payoff Matrix of the Prisoners’ Dilemma Game

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Defect</td>
<td>2</td>
<td>-1</td>
</tr>
</tbody>
</table>
Network Model

Assumptions

- Nodes are selfish and rational, not malicious
- Nodes operate in promiscuous mode
- Game time is divided into slots
Network Model

Assumptions
- Nodes are selfish and rational, not malicious
- Nodes operate in promiscuous mode
- Game time is divided into slots

We model the interaction between any pair of nodes as a repeated two-player game.
- Receive $\alpha$ if a node’s packet is forwarded
- Cost 1 if a node forwards a packet
Define $p \in (0, 1)$ to be the probability of a packet that has been forwarded was not overheard by the originating node.

Define $\hat{p}(k) - i$ to be the perceived dropping probability of node $i$'s neighbor at time slot $k \geq 0$ estimated by node $i$.

Define $\tilde{p}(k)_i S$ to be the actual dropping probability node $i$ should use at time slot $k$ according to strategy $S$.

Table 2: Payoff Matrix of the Packet Forwarding Game

<table>
<thead>
<tr>
<th></th>
<th>Forward</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forward</td>
<td>$\alpha - 1$</td>
<td>$-\alpha - 1$</td>
</tr>
<tr>
<td>Drop</td>
<td>$\alpha$</td>
<td>$-\alpha$</td>
</tr>
</tbody>
</table>

Table 3: Normalized Payoff Matrix of the Packet Forwarding Game

<table>
<thead>
<tr>
<th></th>
<th>Forward</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forward</td>
<td>$\frac{1}{2\alpha} - 1$</td>
<td>$\frac{-1}{2\alpha + 1}$</td>
</tr>
<tr>
<td>Drop</td>
<td>$\frac{2\alpha}{2\alpha + 1}$</td>
<td>0</td>
</tr>
</tbody>
</table>
Define $p_e \in (0, 1)$ to be the probability of a packet that has been forwarded was not overheard by the originating node.

### Table 2: Payoff Matrix of the Packet Forwarding Game

<table>
<thead>
<tr>
<th></th>
<th>Forward</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>Forward</td>
<td>$\alpha - 1$</td>
</tr>
<tr>
<td></td>
<td>Drop</td>
<td>$\alpha$</td>
</tr>
</tbody>
</table>

### Table 3: Normalized Payoff Matrix of the Packet Forwarding Game

<table>
<thead>
<tr>
<th></th>
<th>Forward</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1</td>
<td>Forward</td>
<td>$\frac{1}{2\alpha}$</td>
</tr>
<tr>
<td></td>
<td>Drop</td>
<td>$\frac{2\alpha}{2\alpha - 1}$</td>
</tr>
</tbody>
</table>
Define $p_e \in (0, 1)$ to be the probability of a packet that has been forwarded was not overheard by the originating node.

Define $\hat{p}_{-i}^{(k)}$ to be the perceived dropping probability of node $i$'s neighbor at time slot $k \geq 0$ estimated by node $i$.

\[
\hat{p}_{-i}^{(k)} = p_{-i}^{(k)} + (1 - p_{-i}^{(k)})p_e = p_e + (1 - p_e)p_{-i}^{(k)}, \tag{1}
\]
Define $p_e \in (0, 1)$ to be the probability of a packet that has been forwarded was not overheard by the originating node.

Define $\hat{p}^{(k)}_{-i}$ to be the perceived dropping probability of node $i$’s neighbor at time slot $k \geq 0$ estimated by node $i$.

\[
\hat{p}^{(k)}_{-i} = p^{(k)}_{-i} + (1 - p^{(k)}_{-i})p_e = p_e + (1 - p_e)p^{(k)}_{-i},
\]

Define $\tilde{p}^{(k)}_i S$ to be the actual dropping probability node $i$ should use at time slot $k$ according to strategy $S$. 

**Table 2: Payoff Matrix of the Packet Forwarding Game**  

<table>
<thead>
<tr>
<th></th>
<th>Node 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Forward</td>
<td>Drop</td>
</tr>
<tr>
<td>Node 1</td>
<td>$\alpha - 1$</td>
<td>$\alpha - 1$</td>
</tr>
<tr>
<td></td>
<td>$1$</td>
<td>$-1$</td>
</tr>
<tr>
<td></td>
<td>$2\alpha - 1$</td>
<td>$2\alpha - 1$</td>
</tr>
</tbody>
</table>

**Table 3: Normalized Payoff Matrix of the Packet Forwarding Game**  

<table>
<thead>
<tr>
<th></th>
<th>Node 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Forward</td>
<td>Drop</td>
</tr>
<tr>
<td>Node 1</td>
<td>$1$</td>
<td>$-1$</td>
</tr>
<tr>
<td></td>
<td>$2\alpha - 1$</td>
<td>$2\alpha - 1$</td>
</tr>
<tr>
<td></td>
<td>$1$</td>
<td>$0$</td>
</tr>
<tr>
<td></td>
<td>$2\alpha - 1$</td>
<td>$2\alpha - 1$</td>
</tr>
</tbody>
</table>
### Payoff Function

**Table 2: Payoff Matrix of the Packet Forwarding Game**

<table>
<thead>
<tr>
<th></th>
<th>Node 2 Forward</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1 Forward</td>
<td>$\alpha - 1$</td>
<td>$\alpha - 1$</td>
</tr>
<tr>
<td>Drop</td>
<td>$\alpha - \alpha - 1$</td>
<td>$-\alpha - \alpha$</td>
</tr>
</tbody>
</table>

**Table 3: Normalized Payoff Matrix of the Packet Forwarding Game**

<table>
<thead>
<tr>
<th></th>
<th>Node 2 Forward</th>
<th>Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 1 Forward</td>
<td>$\frac{1}{2\alpha - 1}$</td>
<td>$\frac{2\alpha}{2\alpha - 1}$</td>
</tr>
<tr>
<td>Drop</td>
<td>$\frac{-1}{2\alpha - 1}$</td>
<td>$\frac{0}{2\alpha - 1}$</td>
</tr>
</tbody>
</table>

**Average payoff $u_i^{(k)}$ at time slot $k$ is:**

$$u_i^{(k)} = (1 - p_i^{(k)})(1 - p_{-i}^{(k)}) + \frac{2\alpha}{2\alpha - 1}p_i^{(k)}(1 - p_{-i}^{(k)}) - \frac{1}{2\alpha - 1}(1 - p_i^{(k)})p_{-i}^{(k)}.$$  

Rearranging terms:

$$u_i^{(k)} = 1 + \frac{1}{2\alpha - 1}p_i^{(k)} - \frac{2\alpha}{2\alpha - 1}p_{-i}^{(k)}. \quad (2)$$
Network Model

Payoff Function

Table 2: Payoff Matrix of the Packet Forwarding Game

<table>
<thead>
<tr>
<th></th>
<th>Node 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Forward</td>
<td>Drop</td>
</tr>
<tr>
<td>Node 1</td>
<td>α - 1</td>
<td>-α - 1</td>
</tr>
<tr>
<td>Forward</td>
<td>α - 1</td>
<td>-α - 1</td>
</tr>
<tr>
<td>Drop</td>
<td>α</td>
<td>-α - 1</td>
</tr>
</tbody>
</table>

Table 3: Normalized Payoff Matrix of the Packet Forwarding Game

<table>
<thead>
<tr>
<th></th>
<th>Node 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Forward</td>
<td>Drop</td>
</tr>
<tr>
<td>Node 1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Forward</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Drop</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

Average payoff $u_i^{(k)}$ at time slot $k$ is:

$$u_i^{(k)} = (1 - p_i^{(k)}) (1 - p_{-i}^{(k)}) + \frac{2\alpha}{2\alpha - 1} p_i^{(k)} (1 - p_{-i}^{(k)}) - \frac{1}{2\alpha - 1} (1 - p_i^{(k)}) p_{-i}^{(k)}.$$  

Rearranging terms:

$$u_i^{(k)} = 1 + \frac{1}{2\alpha - 1} p_i^{(k)} - \frac{2\alpha}{2\alpha - 1} p_{-i}^{(k)}. \quad (2)$$

Average discount average payoff $U_i^{(n)}$ of player $i$ starting from time slot $n$ is then given by:

$$U_i^{(n)} = \sum_{k=n}^{\infty} w^{k-n} u_i^{(k)}, \quad (3)$$
Trigger Strategies

- n-step Trigger Strategy

\[
\hat{p}_{i, nT}^{(0)} = 0
\]

\[
\hat{p}_{i, nT}^{(k)} = \begin{cases} 
0 & \text{if } \hat{p}_{i}^{(j)} \leq T \text{ for all } j \in \{k - n, \ldots, k - 1\} \\
1 & \text{else}
\end{cases}
\]
Trigger Strategies

- **n-step Trigger Strategy**

  \[ \hat{p}_{i\_n T}^{(0)} = 0 \]

  \[ \hat{p}_{i\_n T}^{(k)} = \begin{cases} 
  0 & \text{if } \hat{p}_{-i}^{(j)} \leq T \text{ for all } j \in \{k - n, \ldots, k - 1\} \\ 
  1 & \text{else} 
\end{cases} \]

- **Remember** \( \hat{p}_{-i}^{(k)} \) is

  \[ \hat{p}_{-i}^{(k)} = p_{-i}^{(k)} + (1 - p_{-i}^{(k)}) p_e = p_e + (1 - p_e) p_{-i}^{(k)}, \quad (1) \]

  If node \( i \)'s neighbor cooperates, then \( \hat{p}_{-i}^{(k)} = p_e \)

  \[ \Rightarrow \text{the optimal value of } T = p_e \]
Trigger Strategies

- **n-step Trigger Strategy**
  \[
  \hat{p}_i^{(0)}_{nT} = 0,
  \hat{p}_i^{(k)}_{nT} = \begin{cases} 
  0 & \text{if } \hat{p}_i^{(j)} \leq T \text{ for all } j \in \{k - n, \ldots, k - 1\} \\
  1 & \text{else}
  \end{cases}
  \]

- Remember \(\hat{p}_i^{(k)}\) is
  \[
  \hat{p}_i^{(k)} = p_i^{(k)} + (1 - p_i^{(k)})p_e = p_e + (1 - p_e)p_i^{(k)},
  \]

If node i’s neighbor cooperates, then \(\hat{p}_i^{(k)} = p_e\)
\(\Rightarrow\) the optimal value of \(T = p_e\)

- But in reality, \(p_e\) is hard to perfectly estimated, so we have:
  1. If \(T < p_e\) then we have that \(\hat{p}_i^{(k)}_{nT} = 1\) for \(k \geq 1\), so cooperation will never emerge.
  2. If \(T > p_e\) then player \(-i\) will be perceived to be co-operative as long as it drops packets with probability:
  \[
  p_i^{(k)} \leq \frac{T - p_e}{1 - p_e}.
  \]
Analysis of Prior Proposals

Trigger Strategies

- n-step Trigger Strategy

\[
\hat{p}_i^{(0)}_{nT} = 0
\]

\[
\hat{p}_i^{(k)}_{nT} = \begin{cases} 
0 & \text{if } \hat{p}_i^{(j)} \leq T \text{ for all } j \in \{k - n, \ldots, k - 1\} \\
1 & \text{else}
\end{cases}
\]

- Remember \( \hat{p}_{-i}^{(k)} \) is

\[
\hat{p}_{-i}^{(k)} = p_{-i}^{(k)} + (1 - p_{-i})p_e = p_e + (1 - p_e)p_{-i}^{(k)}, \quad (1)
\]

If node \( i \)'s neighbor cooperates, then \( \hat{p}_{-i}^{(k)} = p_e \)

\( \Rightarrow \) the optimal value of \( T = p_e \)

- But in reality, \( p_e \) is hard to perfectly estimated, so we have:

1. If \( T < p_e \) then we have that \( \hat{p}_{i\ nT}^{(k)} = 1 \) for \( k \geq 1 \), so cooperation will never emerge.

2. If \( T > p_e \) then player \( -i \) will be perceived to be cooperative as long as it drops packets with probability:

\[
p_{-i}^{(k)} \leq \frac{T - p_e}{1 - p_e}.
\]

- \( T \) cannot be set exactly to \( p_e \) \( \Rightarrow \) \( p_{-i}^{(k)} \) is always larger than 0.
Tit For Tat

- TFT Strategy

\[ \hat{p}^{(0)}_{i \text{ TFT}} = 0 \]
\[ \hat{p}^{(k)}_{i \text{ TFT}} = \hat{p}^{(k-1)}_{-i} \text{ for } k \geq 1 \]
Tit For Tat

- TFT Strategy

\[ \tilde{p}_i^{(0)}_{TFT} = 0 \]
\[ \tilde{p}_i^{(k)}_{TFT} = \tilde{p}_{-i}^{(k-1)} \text{ for } k \geq 1 \]

- Others proved that TFT does not provide the right incentive for cooperation in wireless networks.
Generous Tit For Tat

- Use a generosity factor $g$ that allows cooperation to be restored
Generous Tit For Tat

- Use a generosity factor $g$ that allows cooperation to be restored
- Generous TFT

$$
\begin{align*}
\hat{p}_{i \ GTFT}^{(0)} &= 0 \\
\hat{p}_{i \ GTFT}^{(k)} &= \max\{\hat{p}_{-i}^{(k-1)} - g, 0\} \text{ for } k \geq 1
\end{align*}
$$
Generous Tit For Tat

- Use a generosity factor $g$ that allows cooperation to be restored
- Generous TFT

$$\hat{p}^{(0)}_{i_{GTFT}} = 0$$
$$\hat{p}^{(k)}_{i_{GTFT}} = \max\{\hat{p}^{(k-1)}_{-i} - g, 0\} \text{ for } k \geq 1$$

- GTFT is a robust strategy where no node can gain by deviating from the expected behavior, even if it cannot achieve full cooperation
Generous Tit For Tat

- Use a generosity factor $g$ that allows cooperation to be restored
- Generous TFT
  
  \[
  \hat{p}_{i_{GFT}}^{(0)} = 0 \\
  \hat{p}_{i_{GFT}}^{(k)} = \max\{\hat{p}_{-i}^{(k-1)} - g, 0\} \text{ for } k \geq 1
  \]

- GTFT is a robust strategy where no node can gain by deviating from the expected behavior, even if it cannot achieve full cooperation

**Corollary**

*If both nodes use GTFT then cooperation is achieved on the equilibrium path if and only if $g = p_e$.***
Generous Tit For Tat

- Use a generosity factor $g$ that allows cooperation to be restored
- Generous TFT
  
  $\tilde{p}^{(0)}_i^{GTFT} = 0$
  
  $\tilde{p}^{(k)}_i^{GTFT} = \max\{\hat{p}^{(k-1)}_{-i} - g, 0\}$ for $k \geq 1$

- GTFT is a robust strategy where no node can gain by deviating from the expected behavior, even if it cannot achieve full cooperation

**Corollary**

*If both nodes use GTFT then cooperation is achieved on the equilibrium path if and only if $g = p_e$.***

- So GTFT also needs a perfect estimate of $p_e$
DARWIN’s goal: propose a reputation strategy that does not depend on a perfect estimation of $p_e$ to achieve full cooperation.
DARWIN’s goal: propose a reputation strategy that does not depend on a perfect estimation of $p_e$ to achieve full cooperation.

CTFT
- Basic idea: A player can avoid being punished by contrition
- A player is always in good standing on the first stage
- A player should cooperate if it is in bad standing or if its opponent is in good standing
- Otherwise, the player should defect
**Definition**

- **DARWIN Strategy**

\[
\hat{p}_{i,\text{DARWIN}}^{(k)} = \left[ \gamma \left( q_{-i}^{(k-1)} - q_i^{(k-1)} \right) \right]_0^1 \text{ for } k \geq 0, \quad (6)
\]

where we define for \( i = \{1, 2\} \):

\[
q_i^{(k)} = \begin{cases} 
\left[ \hat{p}_i^{(k)} - \hat{p}_{i,\text{DARWIN}}^{(k)} \right]_0^1 & \text{for } k \geq 0 \\
0 & \text{for } k = -1.
\end{cases} \quad (7)
\]

Additionally we define the function:

\[
[x]_0^1 = \begin{cases} 
1 & \text{if } x \geq 1 \\
x & \text{if } 0 < x < 1 \\
0 & \text{if } x \leq 0
\end{cases}.
\]
**DARWIN Strategy**

\[
\tilde{p}_{i_{DARWIN}}^{(k)} = \left[ \gamma \left( q_{i_{DARWIN}}^{(k-1)} - q_{i_{DARWIN}}^{(k-1)} \right) \right]_0 \text{ for } k \geq 0,
\]

where we define for \( i = \{1, 2\} \):

\[
q_{i}^{(k)} = \begin{cases} 
\tilde{p}_{i}^{(k)} - \tilde{p}_{i_{DARWIN}}^{(k)} & \text{for } k \geq 0 \\
0 & \text{for } k = -1.
\end{cases}
\]

Additionally we define the function:

\[
[x]^1_0 = \begin{cases} 
1 & \text{if } x \geq 1 \\
x & \text{if } 0 < x < 1 \\
0 & \text{if } x \leq 0
\end{cases}
\]

\( q_{i}^{(k)} \) acts as a measurement of the bad standing of a node.
**Definition**

- **DARWIN Strategy**

  \[ \hat{p}^{(k)}_{i,DARWIN} = \left[ \gamma \left( q_{-i}^{(k-1)} - q_i^{(k-1)} \right) \right]_0^1 \text{ for } k \geq 0, \]

  where we define for \( i = \{1, 2\} \):

  \[ q^{(k)}_i = \begin{cases} \left[ \hat{p}^{(k)}_i - \hat{p}^{(k)}_{i,DARWIN} \right]_0^1 & \text{for } k \geq 0 \\ 0 & \text{for } k = -1. \end{cases} \]

  Additionally we define the function:

  \[ [x]^1_0 = \begin{cases} 1 & \text{if } x \geq 1 \\ x & \text{if } 0 < x < 1 \\ 0 & \text{if } x \leq 0 \end{cases} \]

- \( q^{(k)}_i \) acts as a measurement of the bad standing of a node
- DARWIN’s dropping probability is determined by the difference in the two standings instead of the absolute value of the standing of its opponent.
DARWIN Strategy

\[ \tilde{p}_{i_{DARWIN}}^{(k)} = \left[ \gamma \left( q_{-i}^{(k-1)} - q_{i}^{(k-1)} \right) \right]_0^{1} \text{ for } k \geq 0, \quad (6) \]

where we define for \( i = \{1, 2\} \):

\[ q_{i}^{(k)} = \begin{cases} 
\left[ \tilde{p}_{i}^{(k)} - \tilde{p}_{i_{DARWIN}}^{(k)} \right]_0^{1} & \text{for } k \geq 0 \\
0 & \text{for } k = -1.
\end{cases} \quad (7) \]

Additionally, we define the function:

\[ [x]_0^{1} = \begin{cases} 
1 & \text{if } x \geq 1 \\
x & \text{if } 0 < x < 1 \\
0 & \text{if } x \leq 0
\end{cases} \]

- \( q_{i}^{(k)} \) acts as a measurement of the bad standing of a node.
- DARWIN’s dropping probability is determined by the difference in the two standings instead of the absolute value of the standing of its opponent.
- DARWIN assumes:
  1. Nodes share the perceived dropping probability with each other.
  2. Nodes do not lie about this perceived dropping probability.
Theorem

Assuming $1 < \gamma < p_e^{-1}$, DARWIN is subgame perfect if and only if

$$\omega > \max \left\{ \frac{1}{\gamma}, \frac{1}{2\alpha (1 - p_e \gamma) + p_e \gamma} \right\}$$

Lemma

If both nodes use DARWIN then cooperation is achieved on the equilibrium path. That is, $p_i^{(k)} = p_{-i}^{(k)} = 0$ for all $k \geq 0$. 
Estimate $p_e$

- For the theorem to hold, $p_e$ is still need to be estimated to meet

$$\gamma < p_e^{-1}$$
Estimate $p_e$

- For the theorem to hold, $p_e$ is still need to be estimated to meet $\gamma < p_e^{-1}$
- Define estimated error probability $p^{(e)}_e$ is equal to
  
  $$p^{(e)}_e = p_e + \Delta,$$

  where $\Delta \in (-p_e, 1 - p_e)$ is the estimation error.
Estimate $p_e$

- For the theorem to hold, $p_e$ is still need to be estimated to meet
  $$\gamma < p_e^{-1}$$

- Define estimated error probability $p_e^{(e)}$ is equal to
  $$p_e^{(e)} = p_e + \Delta,$$
  where $\Delta \in (-p_e, 1 - p_e)$ is the estimation error.

- We then have $\gamma < p_e^{-1}$ if and only if:
  $$\Delta > -p_e \left( \frac{1 - p_e}{2 - p_e} \right)$$
Algorithm Implementation

- $c_{ij}^{(k)}$ denotes connectivity, which is the forwarding ratio:

\[
c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}}
\]
$c_{ij}^{(k)}$ denotes connectivity, which is the forwarding ratio:

$$c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}}$$

Then $j$’s average connectivity ratio:

$$\hat{C}_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\} \setminus \{m\}, m \neq j} c_{im}^{(k)} \times c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\} \setminus \{i\}, m \neq j} c_{im}^{(k)}}$$
Algorithm Implementation

- $c_{ij}^{(k)}$ denotes connectivity, which is the forwarding ratio:

  $$c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}}$$

- Then $j$’s average connectivity ratio:

  $$\hat{c}_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\}} c_{im}^{(k)} \times c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\} \setminus \{j\}} c_{im}^{(k)}}$$

- Let $\hat{p}_j^{(k)} = 1 - \hat{c}_j^{(k)}$
Algorithm Implementation

- $c_{ij}^{(k)}$ denotes connectivity, which is the forwarding ratio:
  \[ c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}} \]

- Then $j$’s average connectivity ratio:
  \[
  \hat{c}_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\}} c_{im} \times c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)}}
  \]

- Let $\hat{p}_j^{(k)} = 1 - \hat{c}_j^{(k)}$

- Strategy using Equation (6) and (7)
Algorithm Implementation

- $c_{ij}^{(k)}$ denotes connectivity, which is the forwarding ratio:

  \[ c_{ij}^{(k)} = \frac{F_{ij}^{(k)}}{S_{ij}^{(k)}} \]

- Then $j$’s average connectivity ratio:

  \[ \hat{c}_j^{(k)} = \frac{\sum_{m \in N_i^{(k)} \cup \{i\}} c_{im}^{(k)} \times c_{mj}^{(k)}}{\sum_{m \in N_i^{(k)} \cup \{i\}, m \neq j} c_{im}^{(k)}} \]

- Let $\hat{p}_j^{(k)} = 1 - \hat{c}_j^{(k)}$
- Strategy using Equation (6) and (7)
- To meet $\gamma < p_e^{-1}$, we estimate $p_e$ as $\hat{p}_{ej}$, which is the fraction of time at least one node different from $j$ transmits.
Settings

- NS-2
- Dynamic Source Routing (DSR) Protocol
- $670 \times 670 m^2$
- 50 nodes
- 5 of them are selfish
- 14 source-destination pairs
- simulation time is 800s and each time slot is 60s
- $\gamma$ is set to 2
Results

Figure 4: Normalized throughput for different dropping ratio of selfish nodes

Figure 5: Normalized throughput for different connection rates (for a packet size of 512 bytes)

Figure 6: Normalized throughput for different number of selfish nodes
Conclusions

Studied how reputation-based mechanisms can help cooperation emerge among selfish users.

Proposed a new mechanism called DARWIN.

Showed that DARWIN is robust and is able to achieve full cooperation.