بسم الله الرحمن الرحیم

Security and Privacy in Wireless Networks

Mohammad Hossein Manshaei

manshaei@gmail.com

Chapter 10: (secowinet.epfl.ch)

Packet Forwarding in Ad Hoc Networks

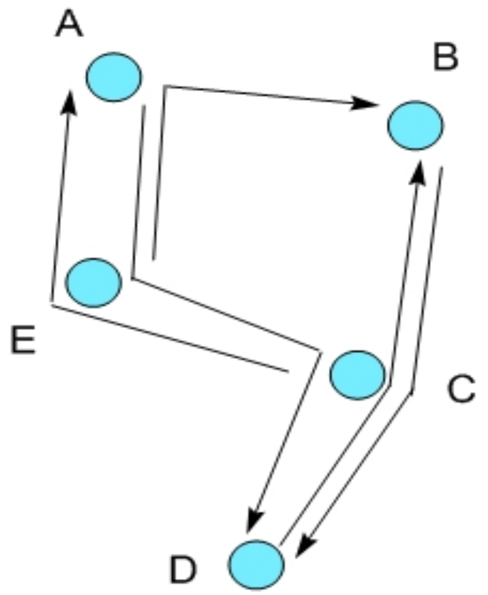
SELFISHNESS IN PACKET FORWARDING

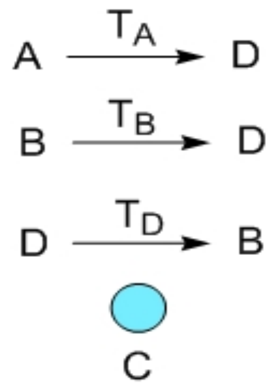
Introduction

- The operation of multi-hop wireless networks requires the nodes to forward data packets on behalf of other nodes
- However, such cooperative behavior has no direct benefit for the forwarding node, and it consumes valuable resources (battery)
- Hence, the nodes may tend to behave selfishly and deny cooperation
- If many nodes defect, then the operation of the entire network is jeopardized
- **Questions:**
  - What are the conditions for the emergence of cooperation in packet forwarding?
  - Can it emerge spontaneously or should it be stimulated by some external mechanism?

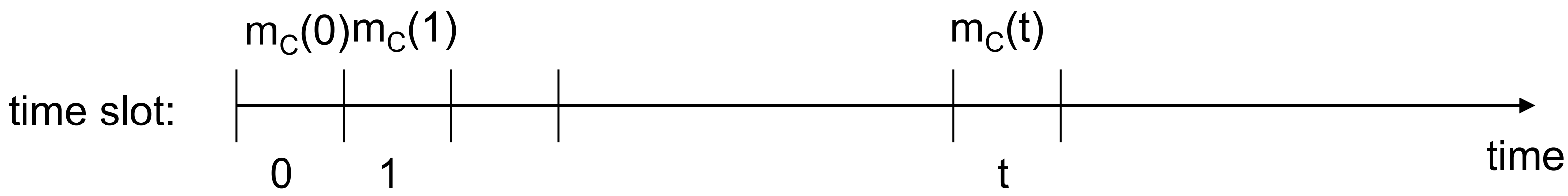
Modeling packet forwarding as a game

Players: nodes





Strategy:
cooperation level



Benefit (of node i):
proportion of packets sent by node i reaching their destination

Cost function

Cost for forwarder f_j:

$$c_{f_j}\left(r,t\right) = -T_s(r)\cdot C\cdot\hat{\tau}_j\left(r,t\right)$$

where:

$T_s(r)$ – traffic sent by source s on route r
C – unit cost of forwarding

Example :

$$\hat{\tau}_C\left(r,t\right) = \prod_{k\in\{E,C\}} m_{f_k}(t) = m_E(t)\cdot m_C(t)$$

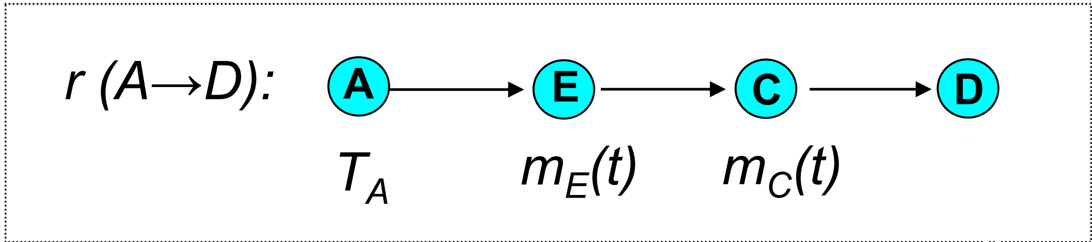
$$c_C\left(r,t\right) = -T_A(r)\cdot C\cdot\hat{\tau}_j\left(r,t\right)$$

Normalized_throughput at forwarder f_j:

$$\hat{\tau}_j\left(r,t\right) = \prod_{k=1}^{j} m_{f_k}(t)$$

where:

r – route on which f_k is a forwarder
t – time slot
f_k – forwarders on route r
m_{fk} – cooperation level of forwarder f_k



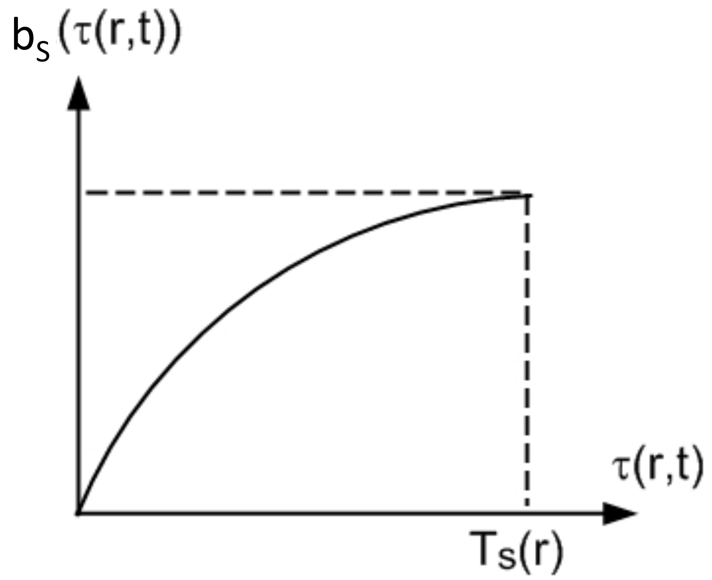
Benefit function

Experienced throughput :

$$\tau(r,t) = T_s(r) \cdot \prod_{k=1}^{l} m_{f_k}(t)$$

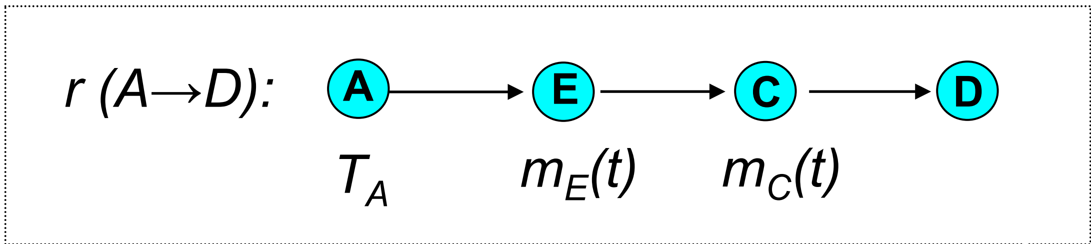
where: s – source
r – route on which s is a source
t – time slot
f_k – forwarders for s
p_{fk} – cooperation level of forwarder f_k

benefit function :



Example :

$$\tau(r,t) = T_A(r) \cdot m_E(t) \cdot m_C(t)$$



Total payoff

Payoff = Benefit - Cost

$$u_i(t) = \sum_{q \in S_i(t)} b_i\big(\tau(q,t)\big) + \sum_{r \in F_i(t)} c_i(r,t)$$

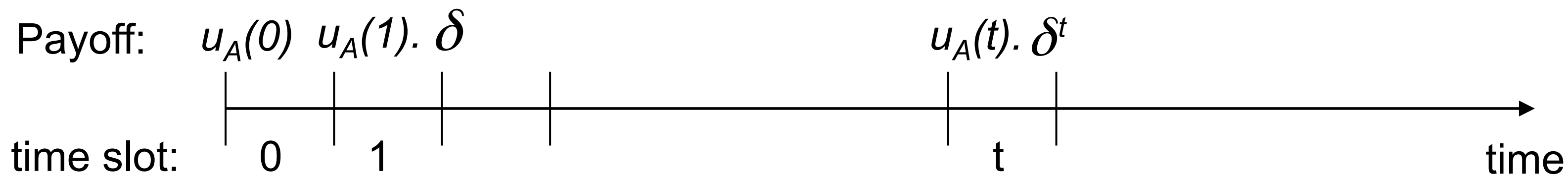
where: $S_i(t)$ – set of routes on which *i* is a source
$F_i(t)$ – set of routes on which *i* is a forwarder

The goal of each node is to maximize its total payoff over the game:

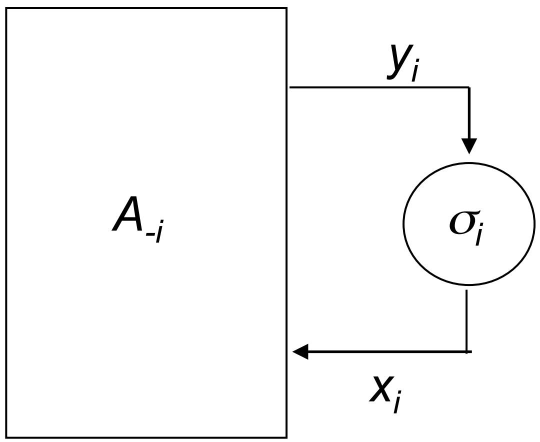
$$\sum_{t=0}^{\infty} u_i(t) \cdot \delta^t$$

where: δ – discounting factor
t – time

Example :



Representation of the Nodes as Players



Strategy function for node *i*:

$$m_i(t) = \sigma_i([\tau(r, t-1)])$$

where:

$\tau(r,t)$ – experienced throughput

Node *i* is playing against the rest of the network (represented by the box denoted by A_{-i})

Examples of Strategies

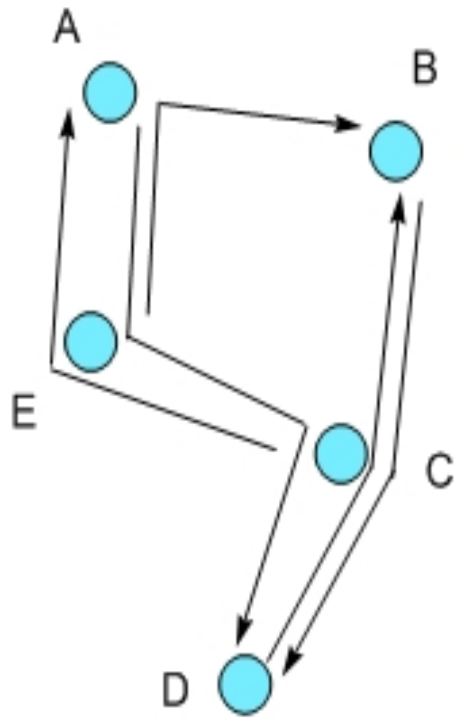
Strategy	Initial cooperation level	Function $\sigma_i(y_i) = x_i$
AllD (always defect)	0	$\sigma_i(y_i) = 0$
AllC (always cooperate)	1	$\sigma_i(y_i) = 1$
TFT (Tit-For-Tat)	1	$\sigma_i(y_i) = y_i$

where y_i stands for the input

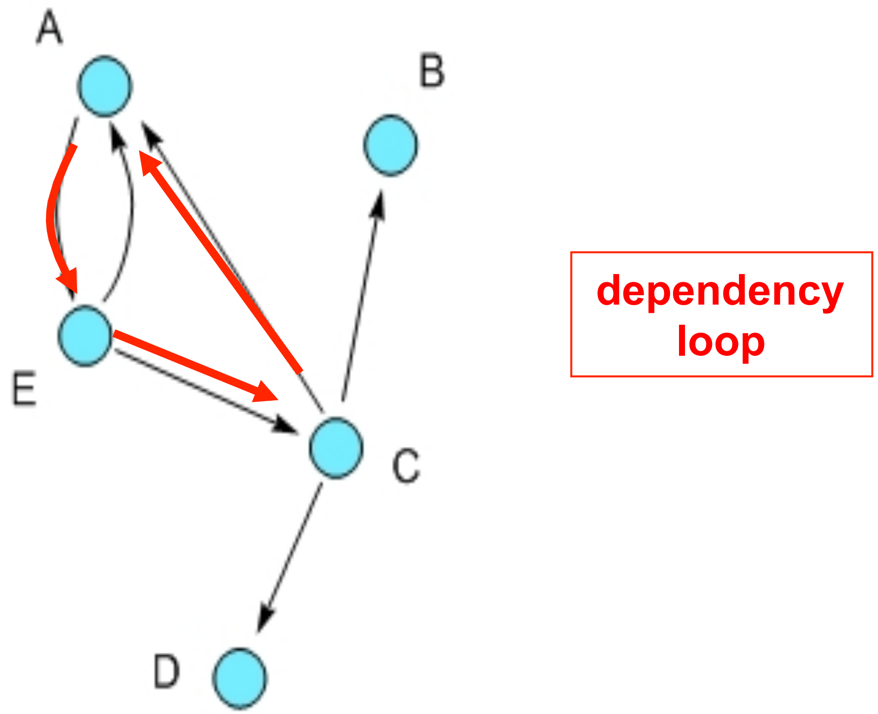
- **non-reactive strategies**:
  the output of the strategy function
  is independent of the input (example: AllD and AllC)
- **reactive strategies**:
  the output of the strategy function
  depends on the input (example: TFT)

Concept of Dependency Graph

dependency: the benefit of each source is dependent on the behavior of its forwarders



(a) routes

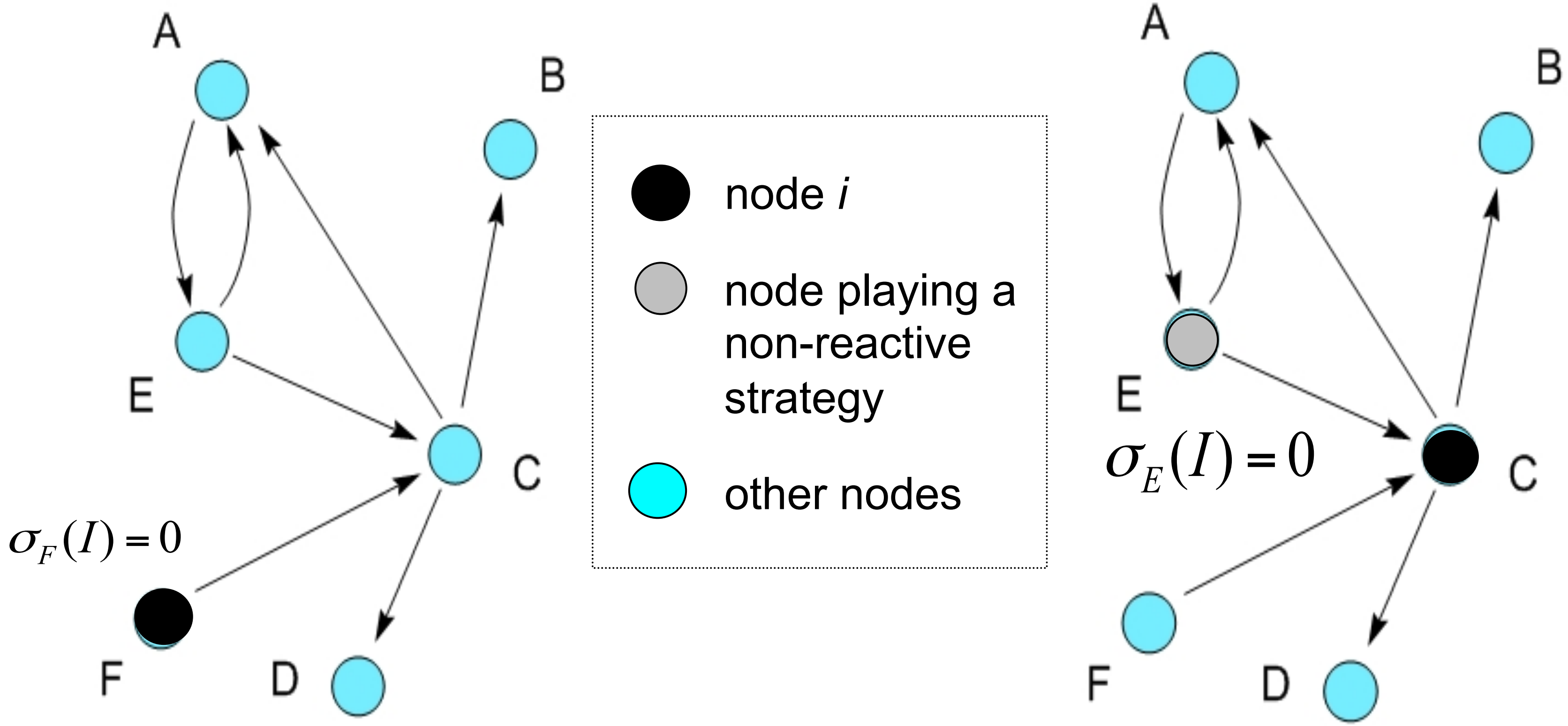


(b) dependency graph

Analytical Results (1/2)

Theorem 1: If node *i* does not have any dependency loops, then its best strategy is AllD.

Theorem 2: If node *i* has only non-reactive dependency loops, then its best strategy is AllD.



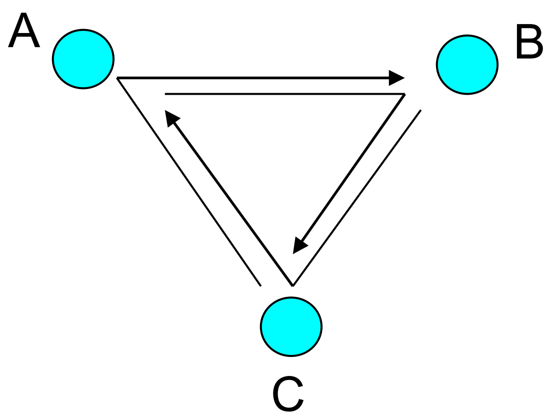
Corollary 1: If every node plays AllD, it is a Nash-equilibrium.

Analytical results (2/2)

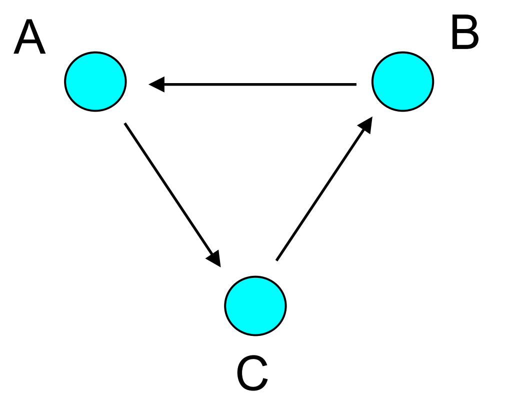
Theorem 3 (simplified): Assuming that node *i* is a forwarder, its behavior will be cooperative only if it has a dependency loop with each of its sources

Corollary 2: If Theorem 3 holds for every node, it is a Nash-equilibrium.

Example in which Corollary 2 holds:

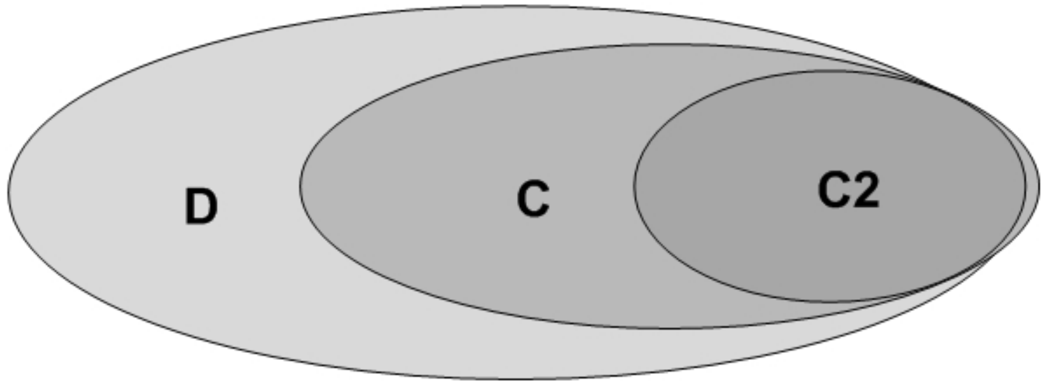


Network



Dependency graph

Classification of Scenarios



D: Set of scenarios, in which every node playing AllD is a Nash equilibrium

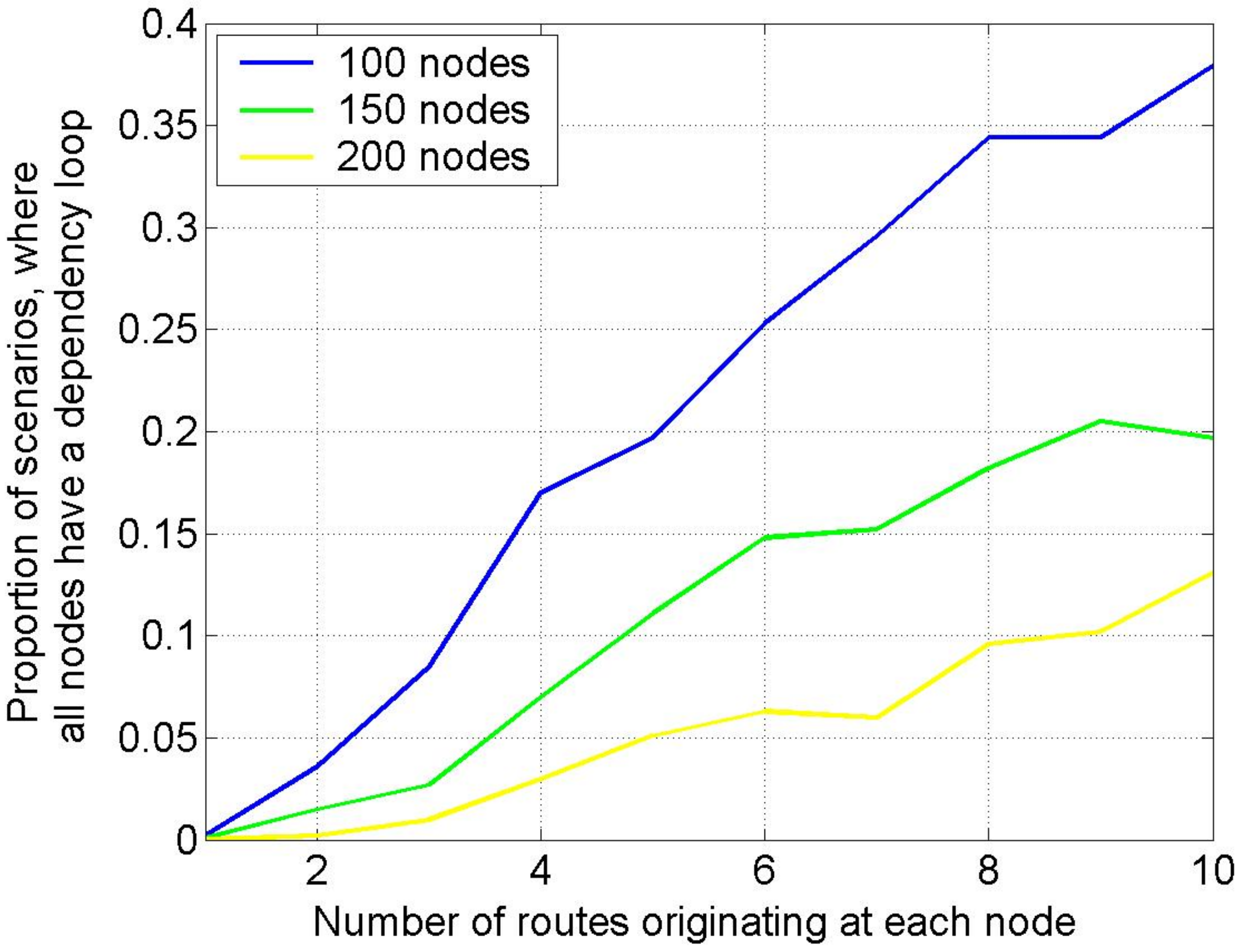
C: Set of scenarios, in which a Nash equilibrium based on cooperation is not excluded by Theorem 1

C2: Set of scenarios, in which cooperation is based on the conditions expressed in Corollary 2

Simulation settings

Number of nodes	100, 150, 200
Distribution of the nodes	random uniform
Area type	torus
Area size	1500x1500m, 1850x1850m, 2150x2150m
Radio range	200 m
Number of routes originating at each node	1-10
Route selection	shortest path
Number of simulation runs	1000

Simulation results



Summary

- **Analytical results:**
  - If everyone drops all packets, it is a Nash-equilibrium
  - **In theory**, given some conditions, a cooperative Nash-equilibrium can exist (i.e., each forwarder forwards all packets)
- **Simulation results:**
  - **In practice**, the conditions for cooperative Nash-equilibria are very restrictive : the likelihood that the conditions for cooperation hold for every node is extremely small
- **Consequences:**
  - Cooperation cannot be taken for granted
  - Mechanisms that stimulate cooperation are necessary
    - incentives based on virtual currency
    - reputation systems