Incentives for Opportunistic Networks

Greg Bigwood gjb4@st-andrews.ac.uk
Tristan Henderson tnhh@st-andrews.ac.uk

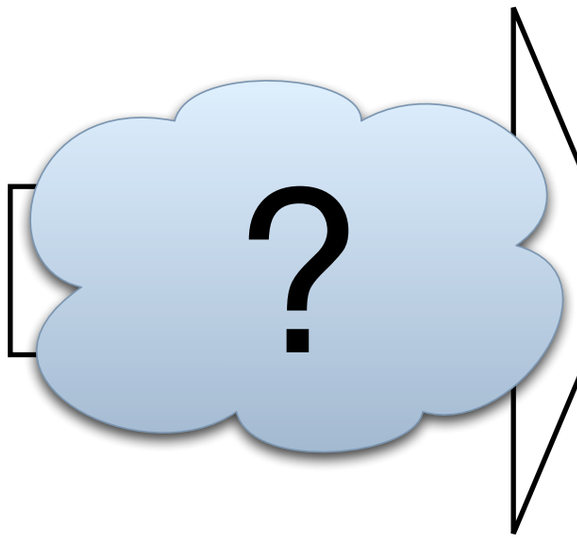
University of St Andrews

Opportunistic Networks

- Users carry wireless mobile devices
- Network leveraged from human encounters
  - Episodic connectivity
  - High delay
- Traditional TCP/IP does not cope
- Store-and-forward architecture
  - use Bluetooth/ WiFi to exchange messages
- Constrained power

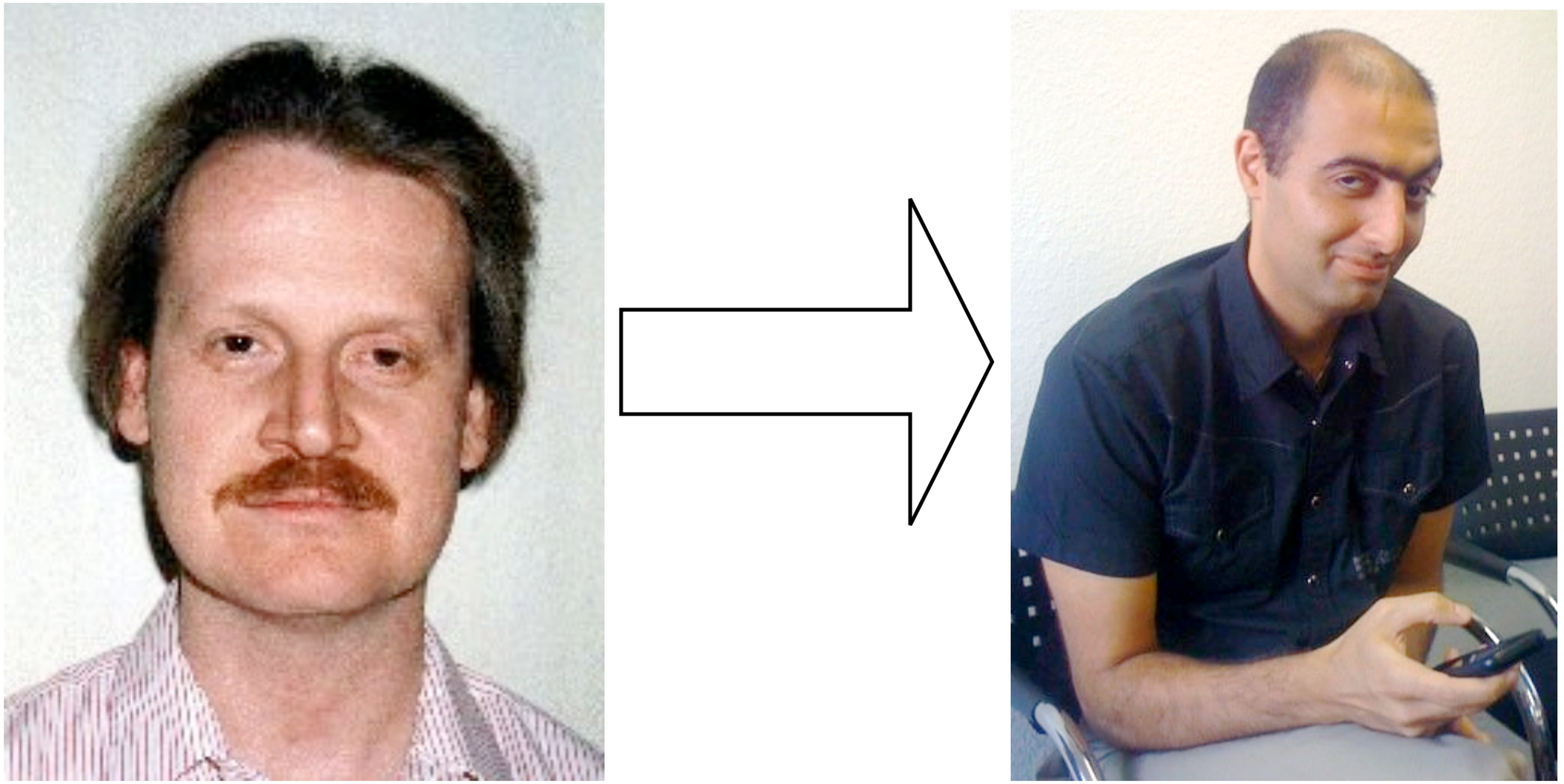
An Opportunistic Messaging Scenario

- Let's say James wants to message his friend Jon using the opportunistic network



He needs someone to forward his msg

- He meets Hämed, and gives him the message



Hämed is a clever guy...

- He worries about the battery cost of forwarding

He drops the message

- A rational, but selfish act

When he later gets to the pub...

- Hämed doesnt pass on the message

Later James meets Jon

- James asks Jon if Hämed gave him the message

?

Jon now knows Hämed is selfish

- Every time he encounters someone he can tell them that Hämed is selfish

When people know Hämed is selfish

- They wont forward his messages for him

Hämed is incentivised to be nice

- He must forward messages for other nodes

How can we incentivise participation?

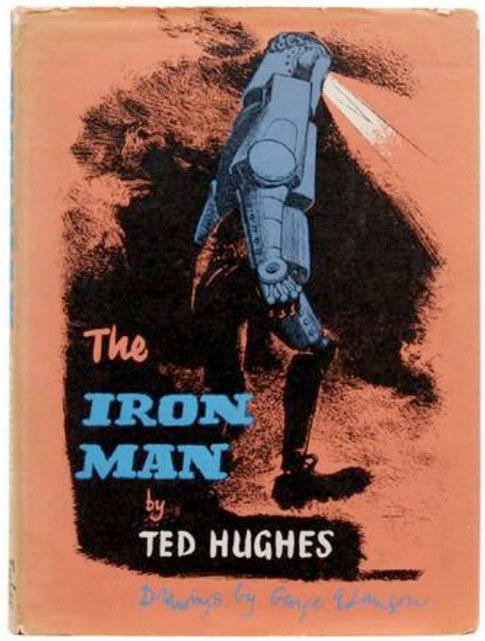
- Encounter histories allow us to detect selfishness
- We can build a concept of reputation
- We can then punish selfish nodes:
  - Drop selfish nodes' messages
  - Encouraging selfish nodes to send messages created by other nodes
- But who can we trust in an opportunistic network?
- Rational behaviour would be for everyone to be selfish
- Therefore: trust no-one?
- But if nobody trusts anyone, nobody can forward!

Trust your “friends”

- **Self-Reported Social Network (SRSN)**: declared social contacts
- For example participants’ Facebook “friends” to give declared social network

IRONMAN

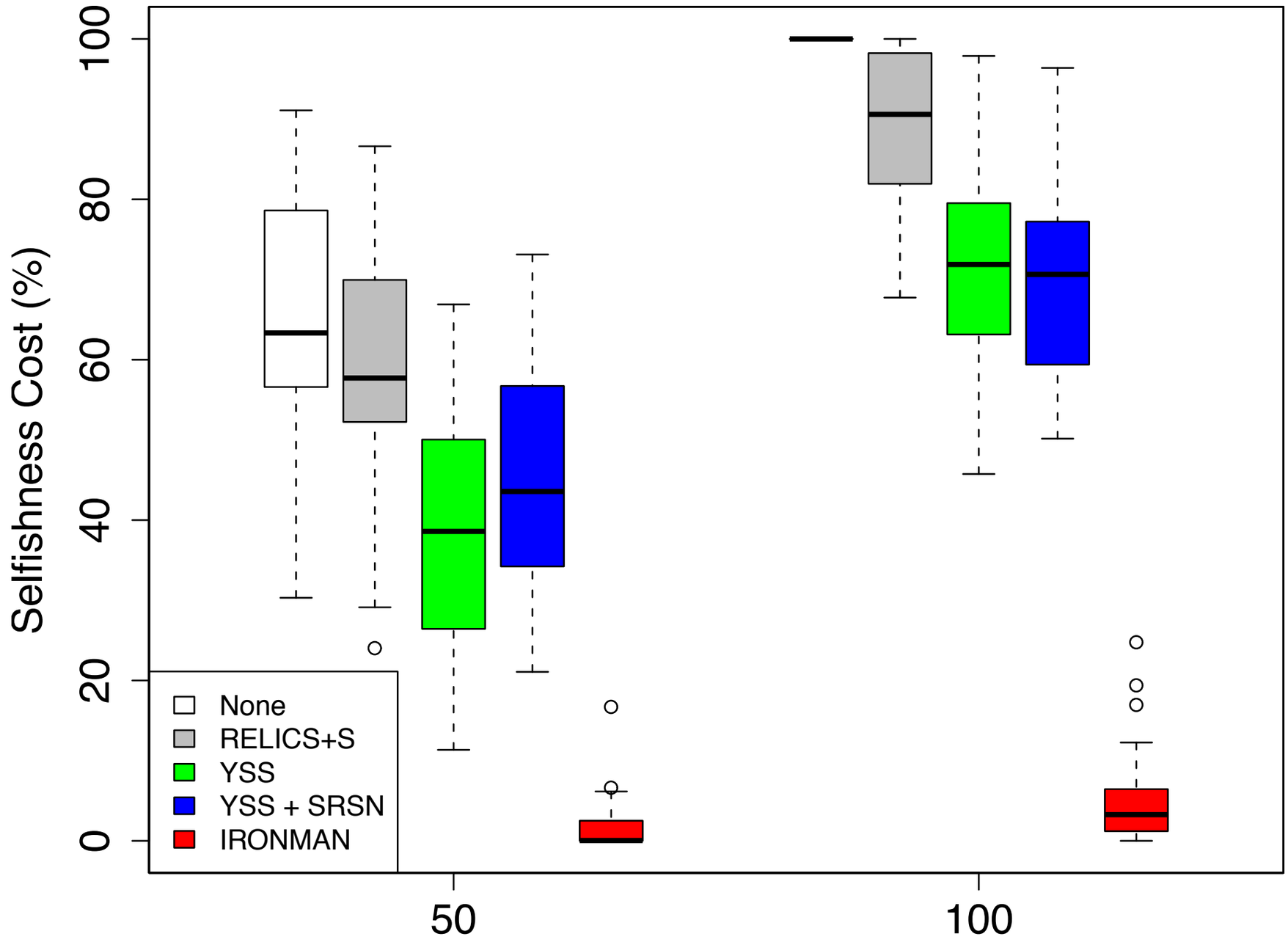
- Incentives and Reputation for Opportunistic Networks using sociAl Networks
- Store history of encounters and message forwards
- Detect selfishness
  - Decrease nodes rating for each msg dropped (additive decrease)
  - Increase nodes rating for each non-selfish forward (additive increase)
- Trust is based on local and global opinions
  - Nodes are initially untrusted, unless in SRSN
  - Opinions exchanged during encounters



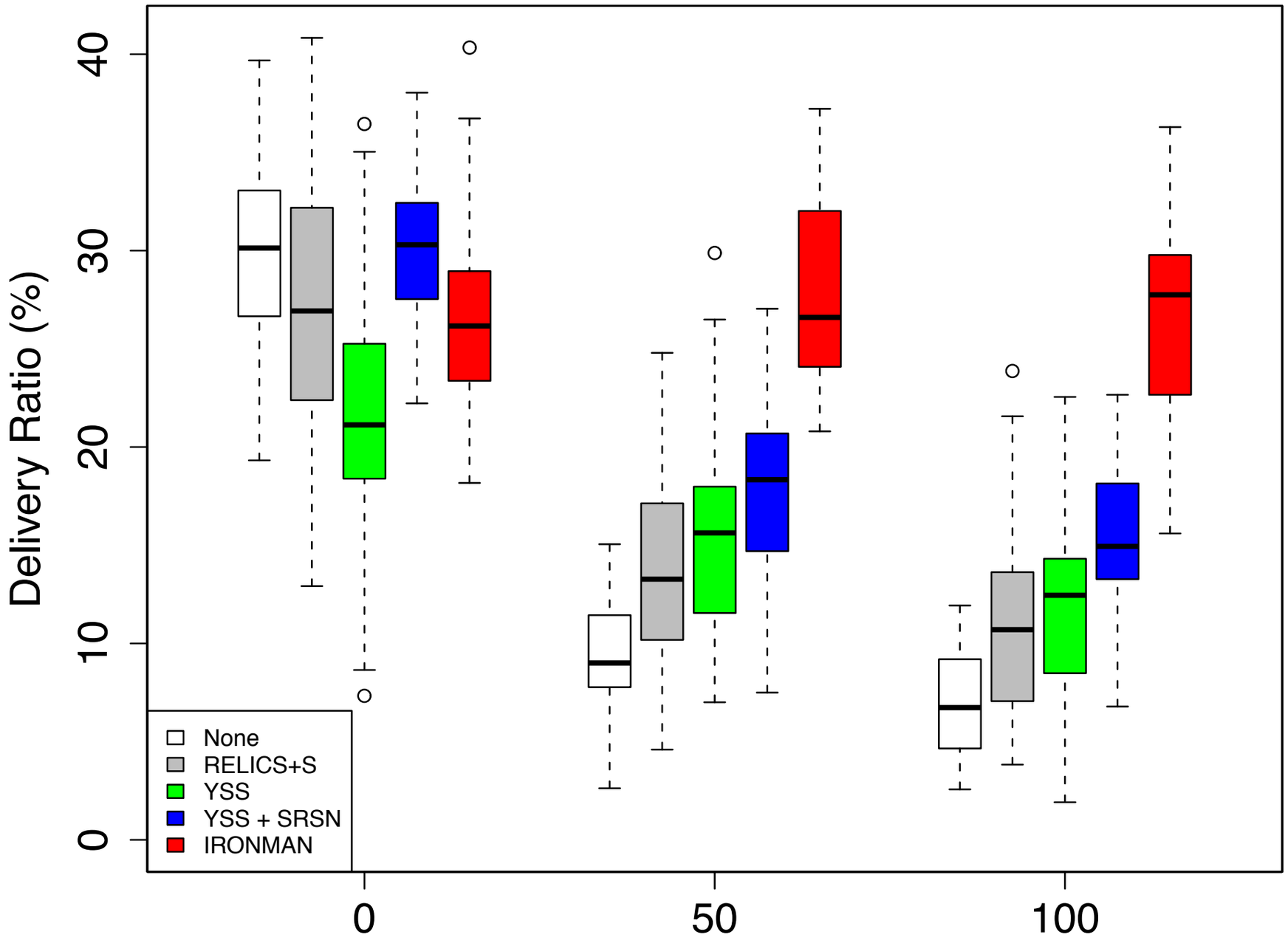
Evaluation

- Compare against existing incentive mechanisms:
  - YSS [Yu, Singh and Sycara 2004]
  - YSS + SRSN
  - RELICS+S [Uddin, Godfrey and Abdelzaher 2010]
- Use trace driven simulation of message passing using Epidemic routing:
  - SASSY dataset (facebook and ZigBee encounters)
  - Reality Mining (phone logs and Bluetooth encounters)
  - HOPE (talk interest and RFID encounters)
  - (all on CRAWDAD crawdad.org)

IRONMAN: deters selfish behaviour



IRONMAN: as good as no selfishness



Conclusions

- IRONMAN does not require an oracle or infrastructure network, nor delivery receipts
- Outperforms existing mechanisms, approaching performance when no selfishness in the network
- Could social-network information be used to improve incentive mechanisms for p2p or ad hoc networks?
- Could social network applications benefit from the reputation layer information?
- Improve selfishness model:
  - Selfish nodes can hide out inside detection time

Now he has his revenge...

Problem

- People are rational
- People are worried about the cost of opportunistic networks
- People are selfish
- People are not going to follow your opportunistic routing protocol
- People are going to drop your messages while expecting you to forward theirs

Detection

Algorithm 1 IRONMAN Selfishness detection

1: $x \leftarrow$*behaviour constant*

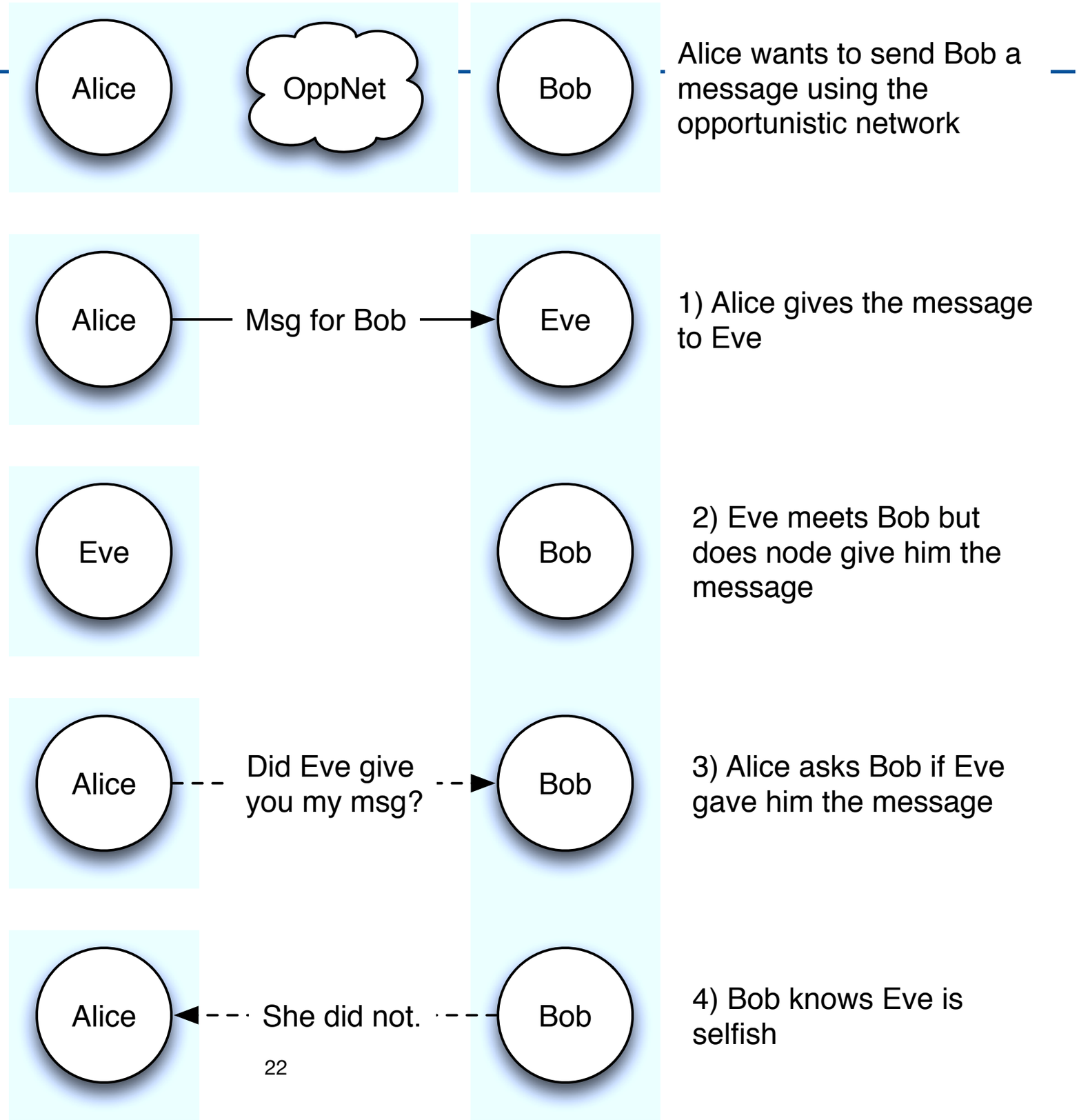
function EncounterNode(B):

1: history_tuples $\leftarrow$ [(*exchange_time*, *msg_id*, *msg_source*, *node_seen*)]
2: exchange forwarding history with B
3: **for all** *message_exchanges* in foreign_history **do**
4: **if** *exchange_time* $>$ last encounter with B **then**
5: **if** *msg_destination* $==$ *my_id* **then**
6: **if** last encounter with *node_seen* $>$ last encounter with B **then**
7: **if** *node_seen* did not give us *msg_id* **then**
8: $\text{Rating}_{node\_seen} \leftarrow \text{Rating}_{node\_seen} - x$

function ReceiveMessage(*other_node*, *msg_src*):

1: **if** *other_node* $\neq$ *msg_src* **then**
2: $\text{Rating}_{other\_node} \leftarrow \text{Rating}_{other\_node} + x$

Detection


Alice
OppNet
Bob
Alice wants to send Bob a message using the opportunistic network
Alice
Msg for Bob
Eve
1) Alice gives the message to Eve
Eve
Bob
2) Eve meets Bob but does node give him the message
Alice
Did Eve give you my msg?
Bob
3) Alice asks Bob if Eve gave him the message
Alice
She did not.
Bob
4) Bob knows Eve is selfish