Cooperation in Decentralised Networks

Michael Rogers

m.rogers@cs.ucl.ac.uk

Cooperation in Decentralised Networks

1) Reciprocation
2) Cooperation over longer distances
3) Delivery receipts

Background

- Peer-to-peer, mobile ad hoc networks
- Infrastructure provided by users
- Incentives to contribute resources

Incentives

Three approaches:

- Micropayments
- Reputations
- Reciprocation

Reciprocation

- “Payment in kind” between neighbours
- Doesn't require currency
- Doesn't require system-wide identities

Requirements for Reciprocation

1. Authentication between neighbours
2. Expect a continuing relationship
3. Measure the benefit provided by neighbours

Is Reciprocation Rational?

- Will selfish nodes reciprocate?
- Need to define selfishness
- Economics: utility-maximising behaviour

Definition of Expected Utility

- Every action has a cost and one or more possible outcomes
- Every outcome has a benefit and a probability
- Expected benefit = mean benefit of all outcomes, weighted by probability
- Expected utility = expected benefit ÷ cost

Definition of Selfishness

- Given a choice of actions, a selfish node always chooses the action with the highest expected utility
- Maximise benefit for a given cost
- Minimise cost for a given benefit
- Costs and benefits are subjective

A Selfish View of Reciprocation

- Assume neighbours are selfish
- Measure the benefit provided by each neighbour
- Benefit is attributable to the work done for the neighbour
- Benefit per unit of work done tells us the expected benefit of doing more work

Prioritising Requests

- “A selfish node always chooses the action with the highest expected utility”
- Serve requests in decreasing order of expected utility, regardless of the order of arrival
  - Cooperative neighbours get higher priority
  - As a neighbour is served, its priority decreases

Summary: Reciprocation

- Cooperation between immediate neighbours
- Only requires local information
  - Decentralised
  - Scalable
- Rational for selfish nodes to reciprocate

Cooperation in Decentralised Networks

1) Reciprocation
2) Cooperation over longer distances
3) Delivery receipts

Cooperation Over Longer Distances

- Reciprocation is local (single-hop)
- What about multi-hop interactions?
  - Peer-to-peer search
  - Packet forwarding

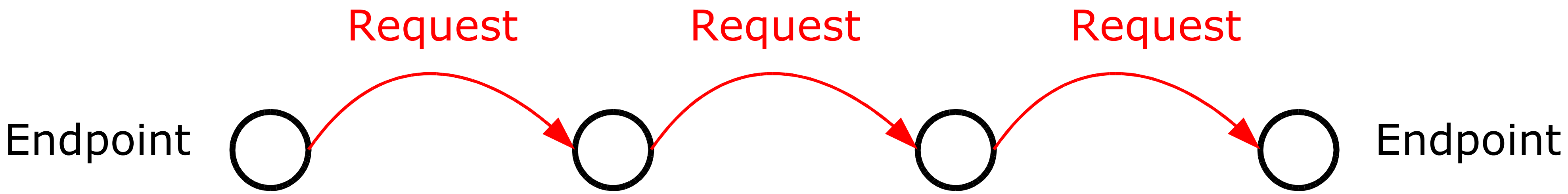
Cooperation Over Longer Distances

- Break each multi-hop interaction into a series of single-hop interactions
- Each node:
  - Provides a service to its upstream neighbour
  - Requests a service from its downstream neighbour

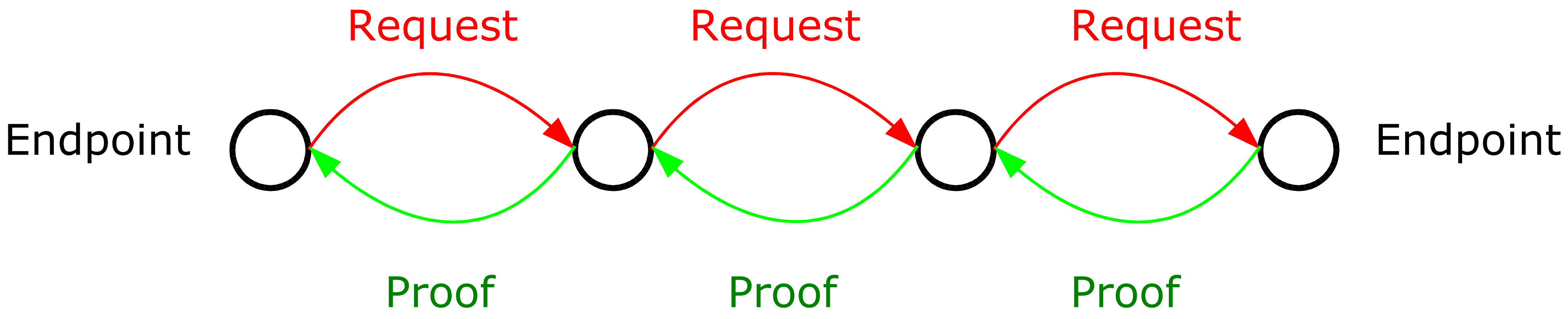
Proof of Work

- Need to measure the benefit provided by the downstream neighbour
- Final node creates a proof of work
- Each node verifies the proof and forwards it upstream

Proof of Work



Proof of Work



Proof of Work

- What is a suitable proof of work?
- Depends on the nature of the work...

Cooperation in Decentralised Networks

1) Reciprocation
2) Cooperation over longer distances
3) Delivery receipts

Delivery Receipts

- Proof of work for multi-hop packet forwarding
- Based on one-way hash functions
- Endpoints share a secret authentication key
- Relays don't need any keys

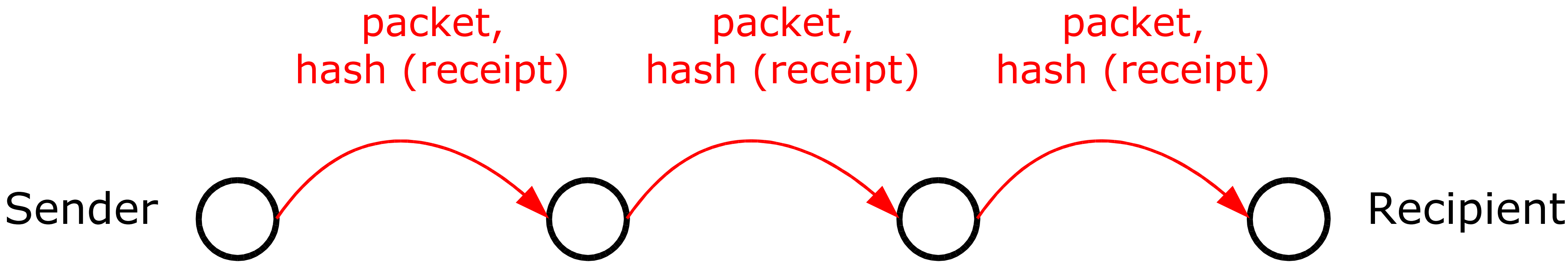
Delivery Receipts: Downstream

- Sender creates a unique secret for each packet
- The hash of the secret is attached to the packet
- Relays store the hash of the secret

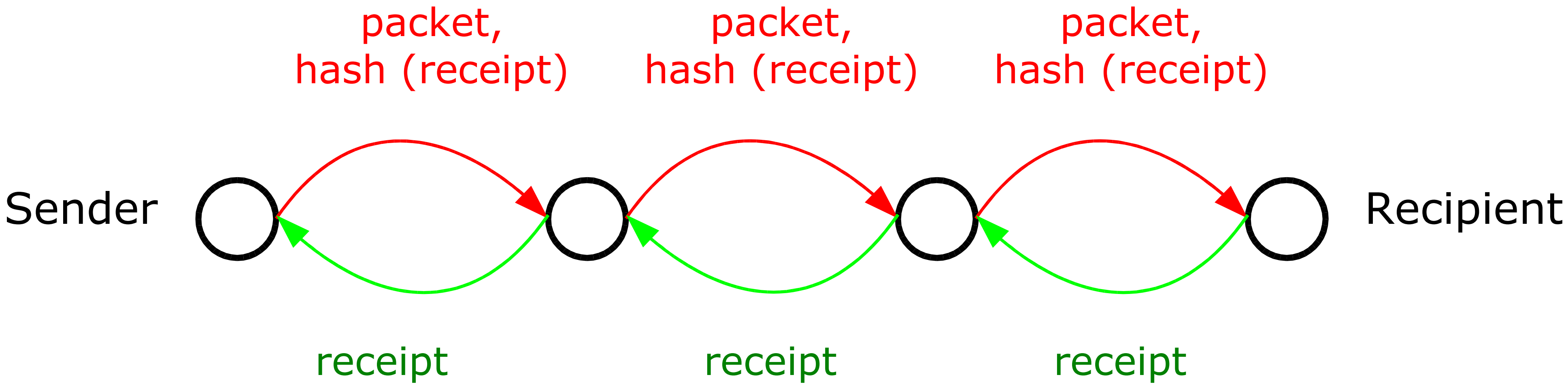
Delivery Receipts: Upstream

- Recipient generates the same secret and sends it as a receipt
- Relays hash the receipt and compare it to the stored value
- The receipt is forwarded back to the sender
- Each node has proof that its downstream neighbour forwarded the packet

Delivery Receipts Example



Delivery Receipts Example



Security Requirements

- Relays must not be able to forge receipts
  - No two packets have the same receipt
  - Receipts don't leak information about the authentication key
  - The hash function is 2nd-preimage-resistant
- Modifying the packet invalidates the receipt

Implementation Using MACs

- Message authentication codes have all the required properties
- The receipt is the MAC of the packet
- The hash of the MAC is attached to the packet
- Relays don't verify the receipt *as a MAC* – they just hash it and compare it to the stored value

Overhead of Delivery Receipts

- Bandwidth overhead: 20 bytes/packet
- Processing overhead: one hash/packet
- Storage overhead: 20 bytes/packet × RTT
  - OK if packets are large (file transfer)
  - May not be OK if packets are small and frequent (games/voice)

Reliability

- What if a node forwards a packet that is later dropped downstream?
- The node has done work but can't prove it

Measuring Reliability

- Measure the reliability of the downstream path
- Estimate the probability of getting a receipt
- “Expected benefit = mean benefit of all outcomes, weighted by probability”
- Forwarding on unreliable paths has lower expected utility

Summary

- Decentralised networks need incentives
- Reciprocation can support multi-hop interactions
  - With a suitable proof of work
- Delivery receipts provide proof of work for packet forwarding

Questions?