

# XLowPan

The implementation of XLowPan, a subset of 6LowPan

# Aims

- Implement robust data-link layer protocol
- Implement on top of 802.15.4 wireless stack
- Mesh routing
- Suitable for embedded devices (Prospeckz)

# Features

- 64 Bit addressing, broadcasting
- 802.15.11 compatible
- Ad-hoc mesh routing

# Implementation

- Fully portable C implementation
- Well defined C interface
- Optimised for low-memory environments.

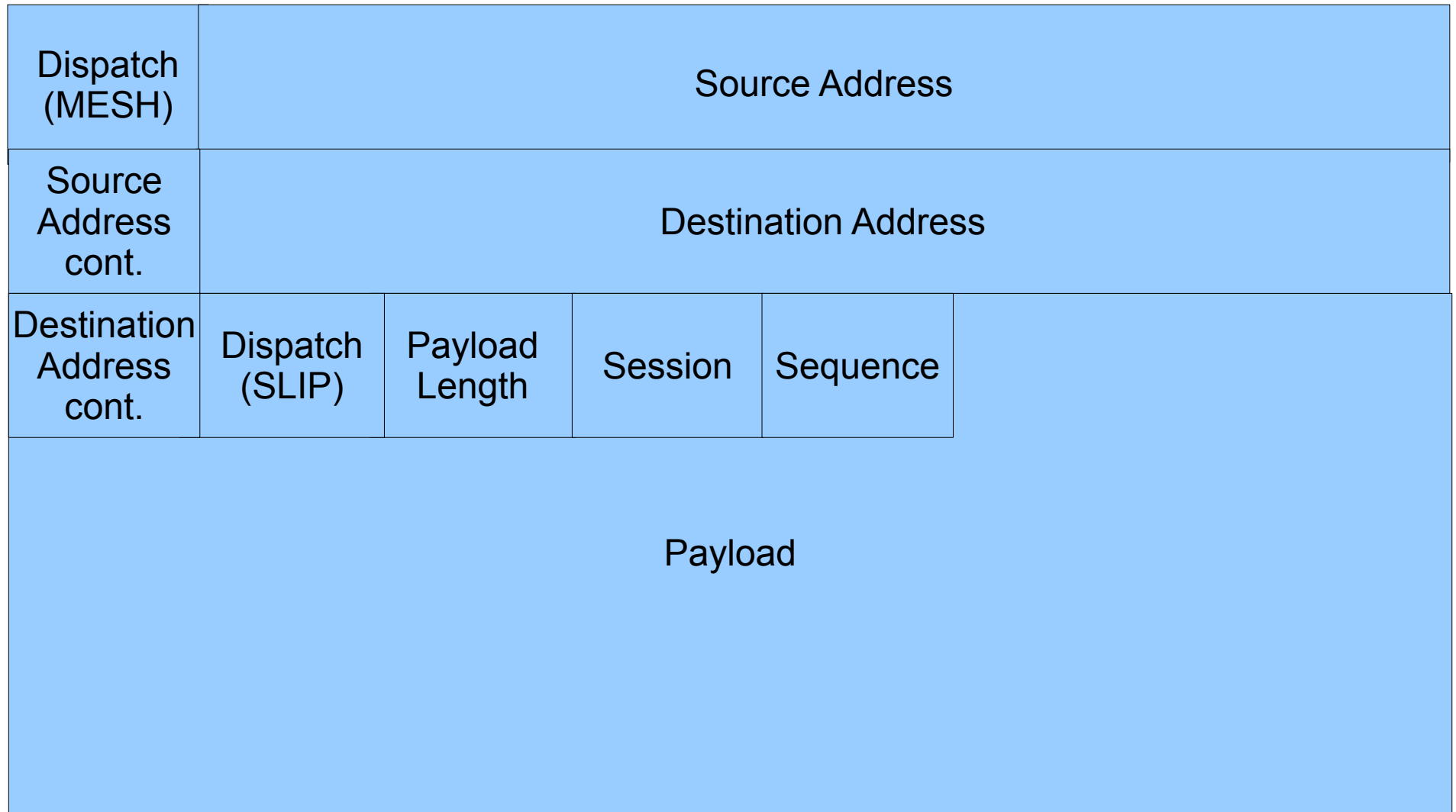
Transport and/or Application Layer

**Data Link Layer - XLowPan**

Physical and Mac Layer (802.15.4)

# Packet Outline

← 8 bytes →



# Mesh Routing

- No central node or basestation
- Flexible Ad-hoc mesh routing
- Mapping source addresses to session and sequence number.
- Sequence number is to prevent a packet being forwarded more than once and to only be consumed once at the destination

# Requirements

- Small footprint
  - Prospeckz only have small amounts of memory
- Ad-hoc
  - Mainly for Awesomegotchi – Ad-hoc (base-station-less) routing needs to be supported
- Low power
  - Needs to run easily on batteries
- Mobile
  - Nodes can move arbitrarily

# Flooding

- **Small Footprint**
  - Very simple to implement
  - Low packet overhead – just need a hop counter
- **Ad-hoc**
  - All nodes in range are always routable
- **Low power**
  - Flooding requires a lot of power – every packet is forwarded multiple times by most nodes
- **Mobile**
  - No state is kept so mobility is implicit

# RPL

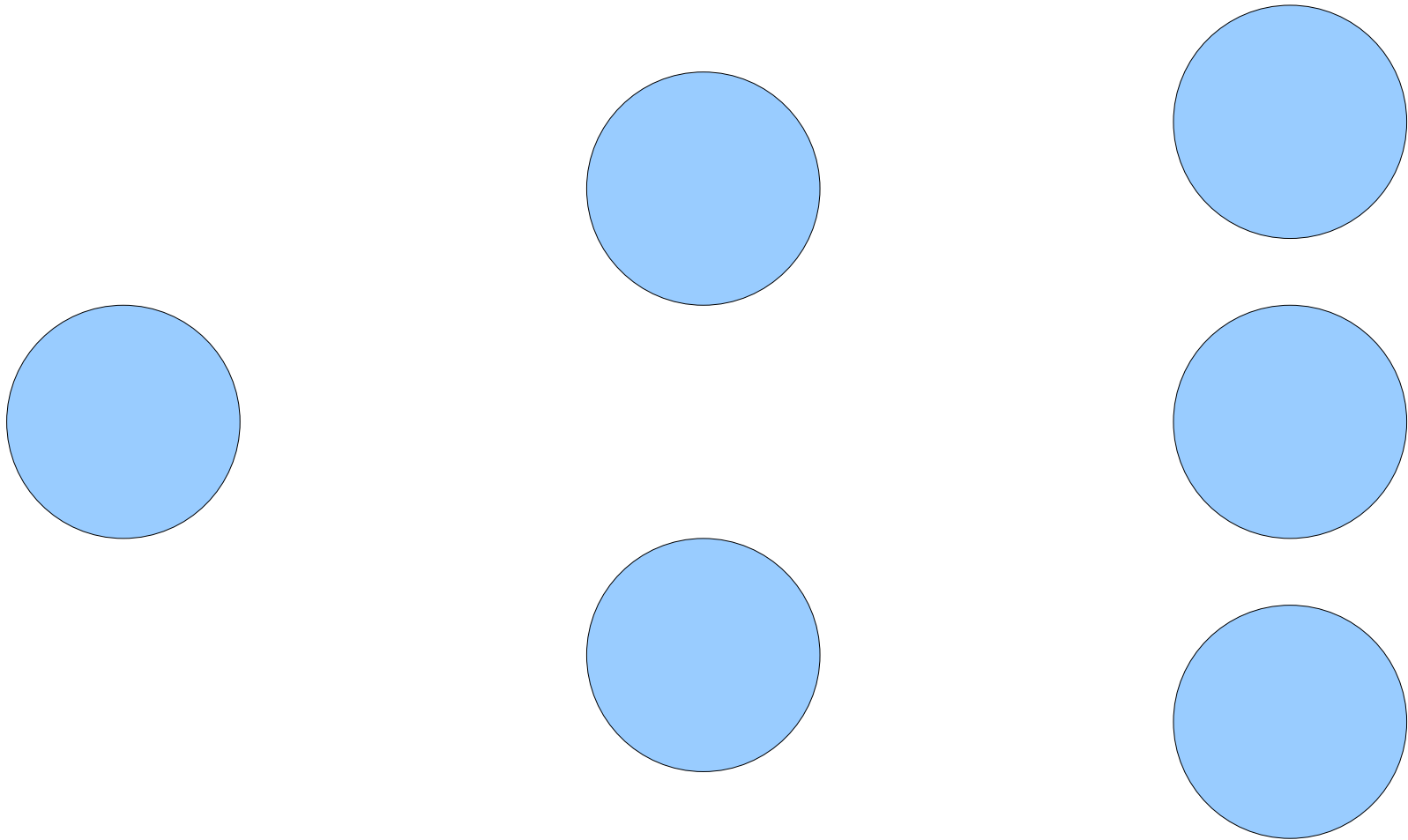
## Routing for Low power Devices

- Small footprint
  - RPL Designed for embedded devices – but requires some RAM for routing tables
- Ad-hoc
  - Fully ad-hoc, peer-to-peer routing modes not supported
- Low Power
  - RPL Designed for embedded devices
- Mobile
  - Mobility of devices is supported

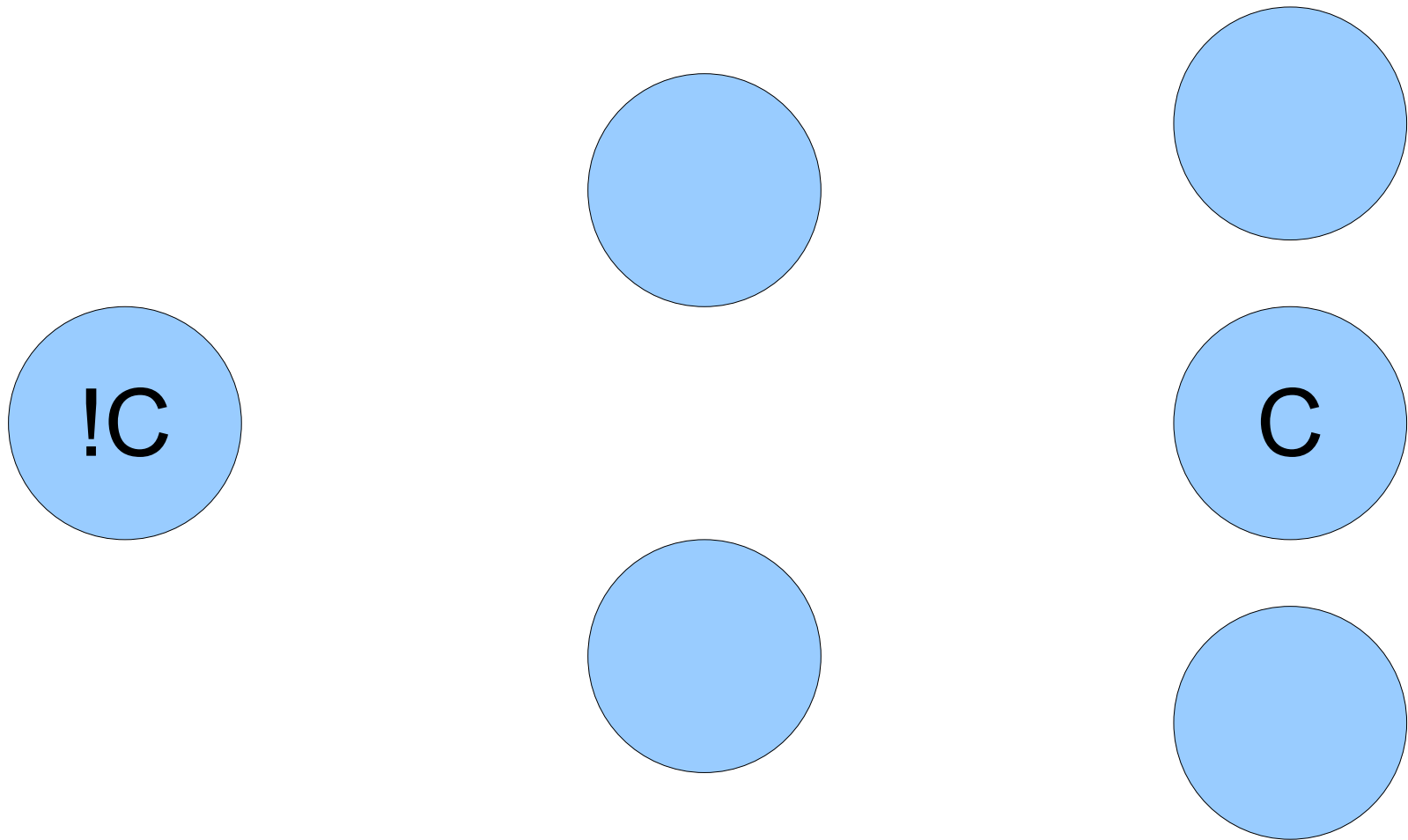
# Our Protocol

- **Small Footprint**
  - Uses more RAM than flooding, but less than RPL.  
Small ROM footprint.
- **Ad-hoc**
  - Designed for fully ad-hoc networks
- **Low Power**
  - Less than flooding, more than RPL (in general)
- **Mobile**
  - Supports fully mobile networks

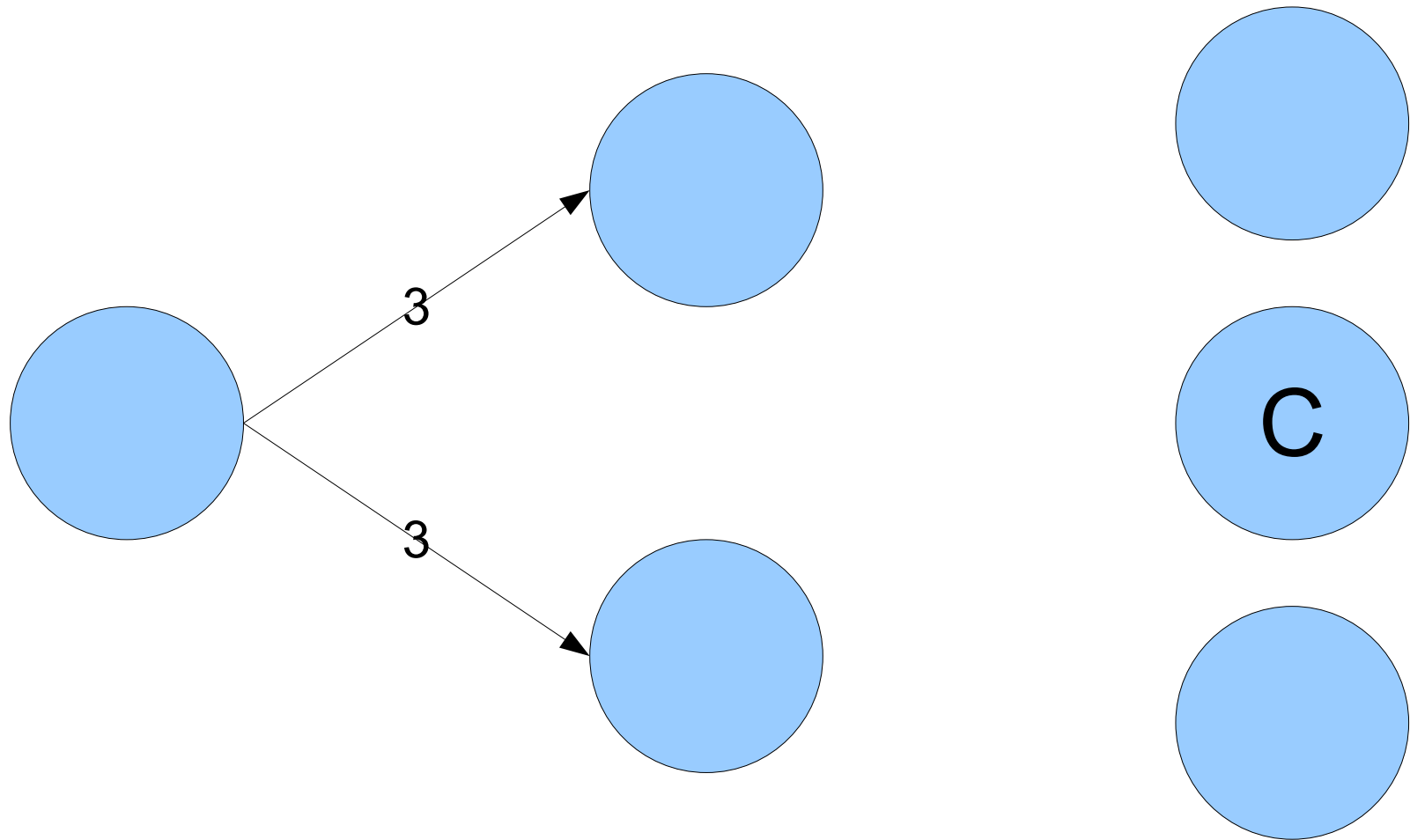
# Flooding Mechanism



# Flooding Mechanism

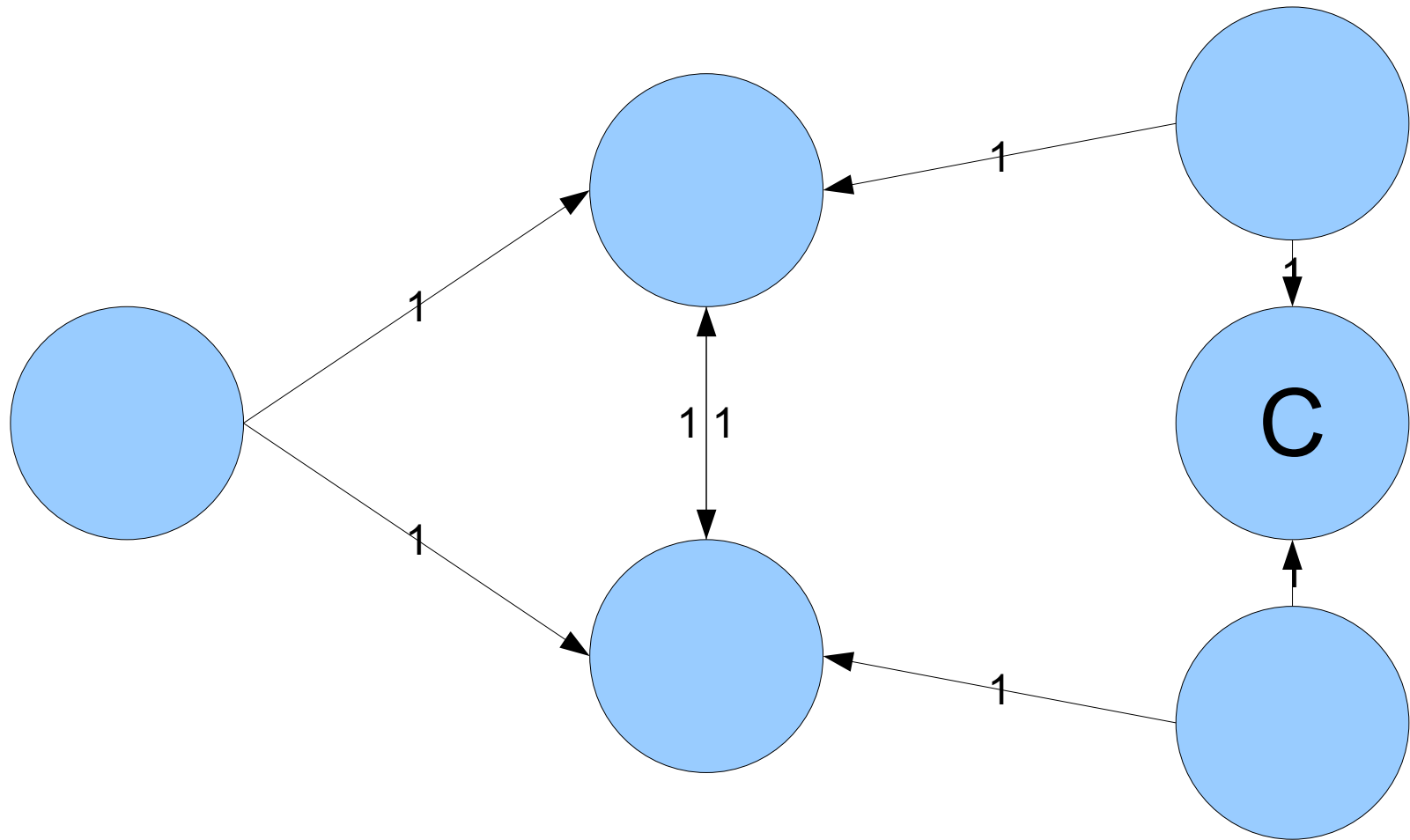


# Flooding Mechanism





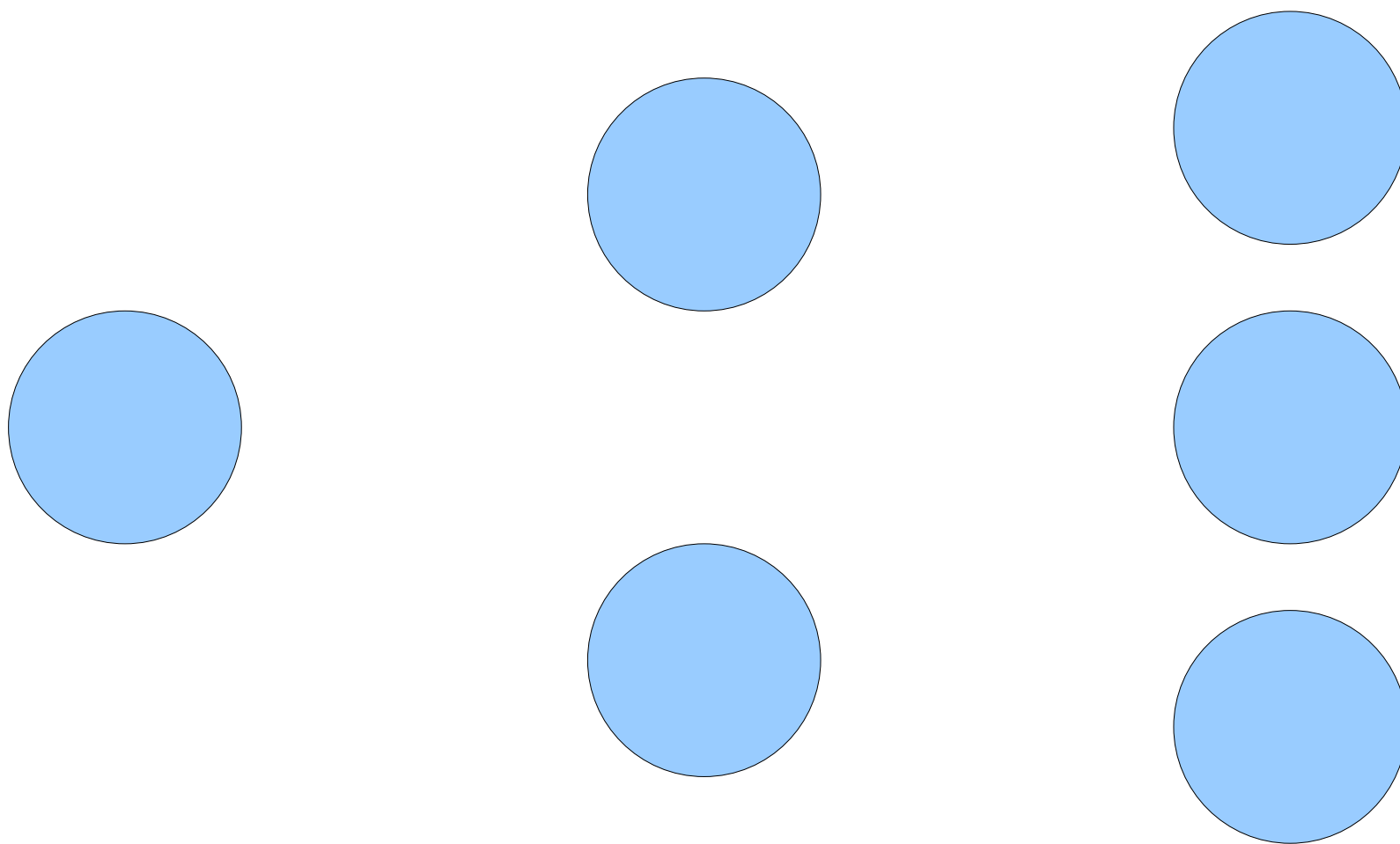
# Flooding Mechanism



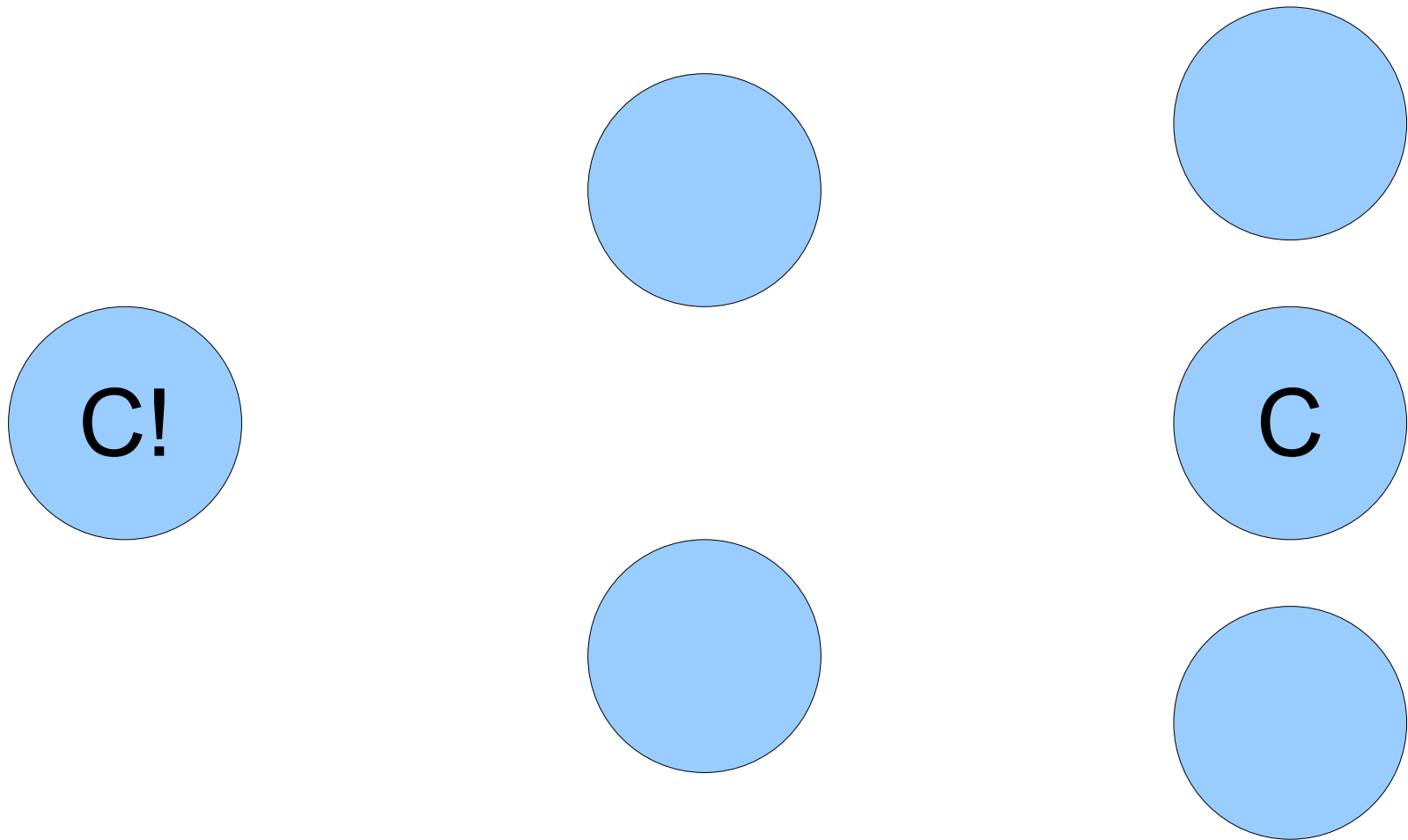
# Flooding

- 8 total transmits for a single useful transmission!
- Lots of interference caused by packet transmissions – can drown out each other
- Lots of power used transmitting these packets

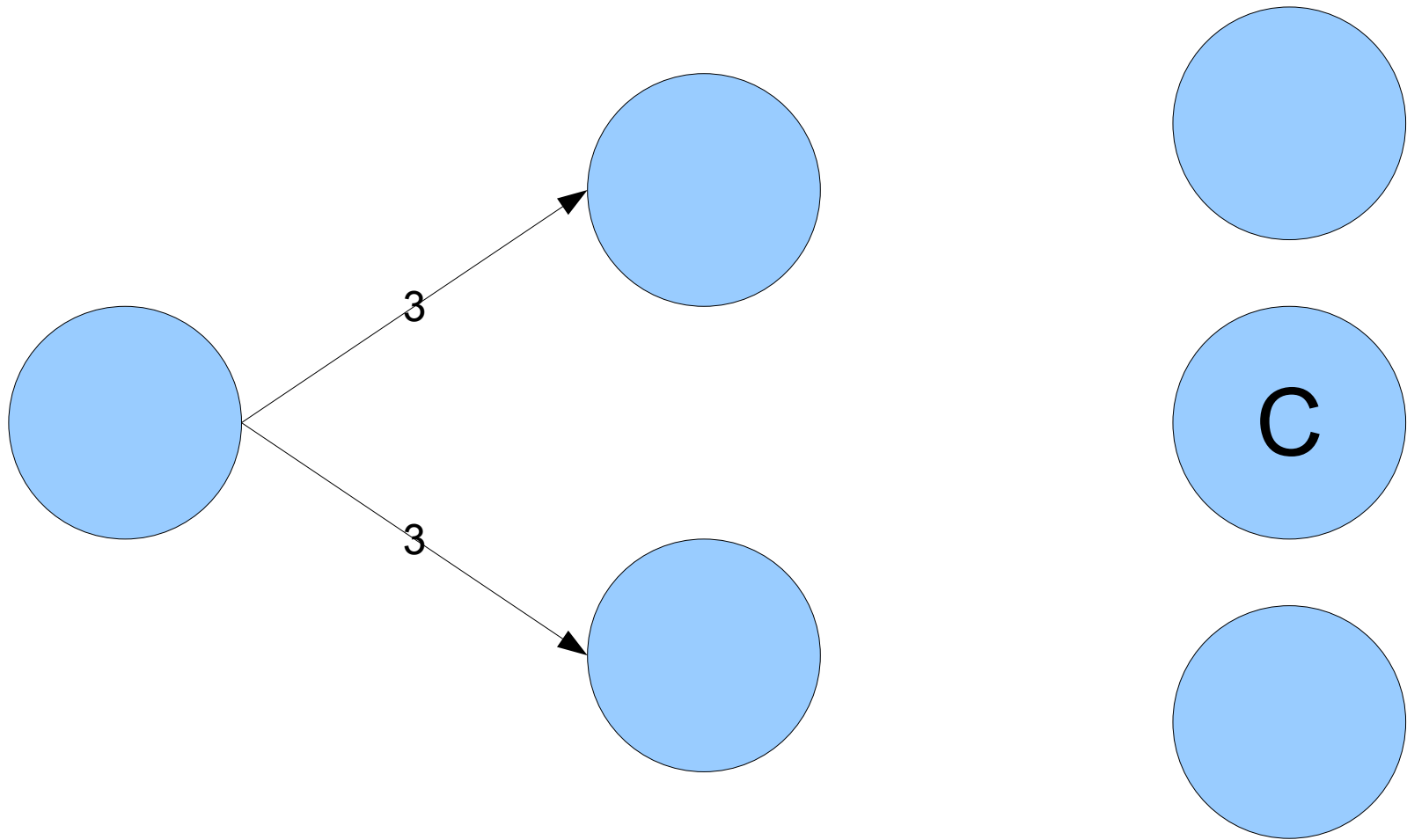
# Our Protocol



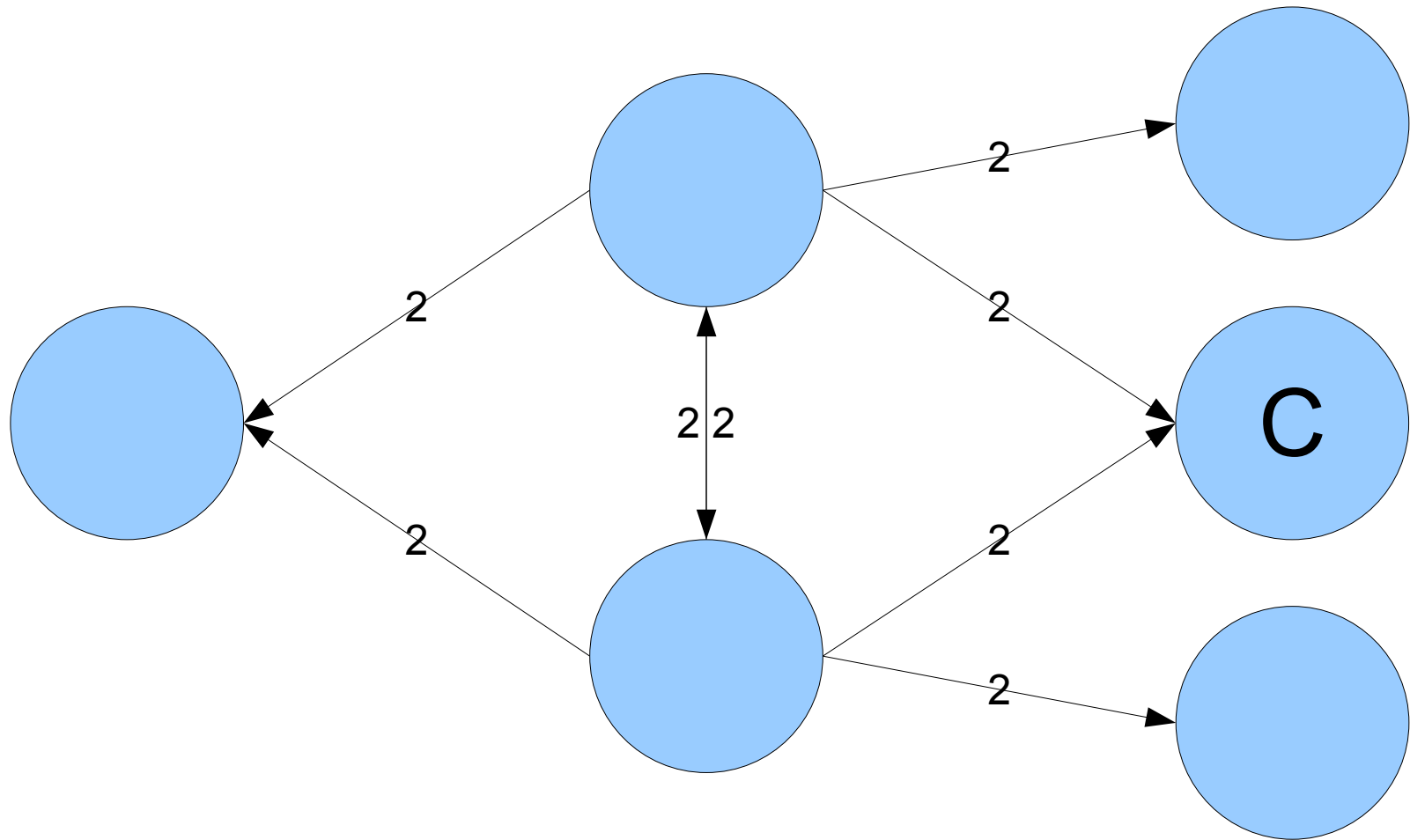
# Our Protocol



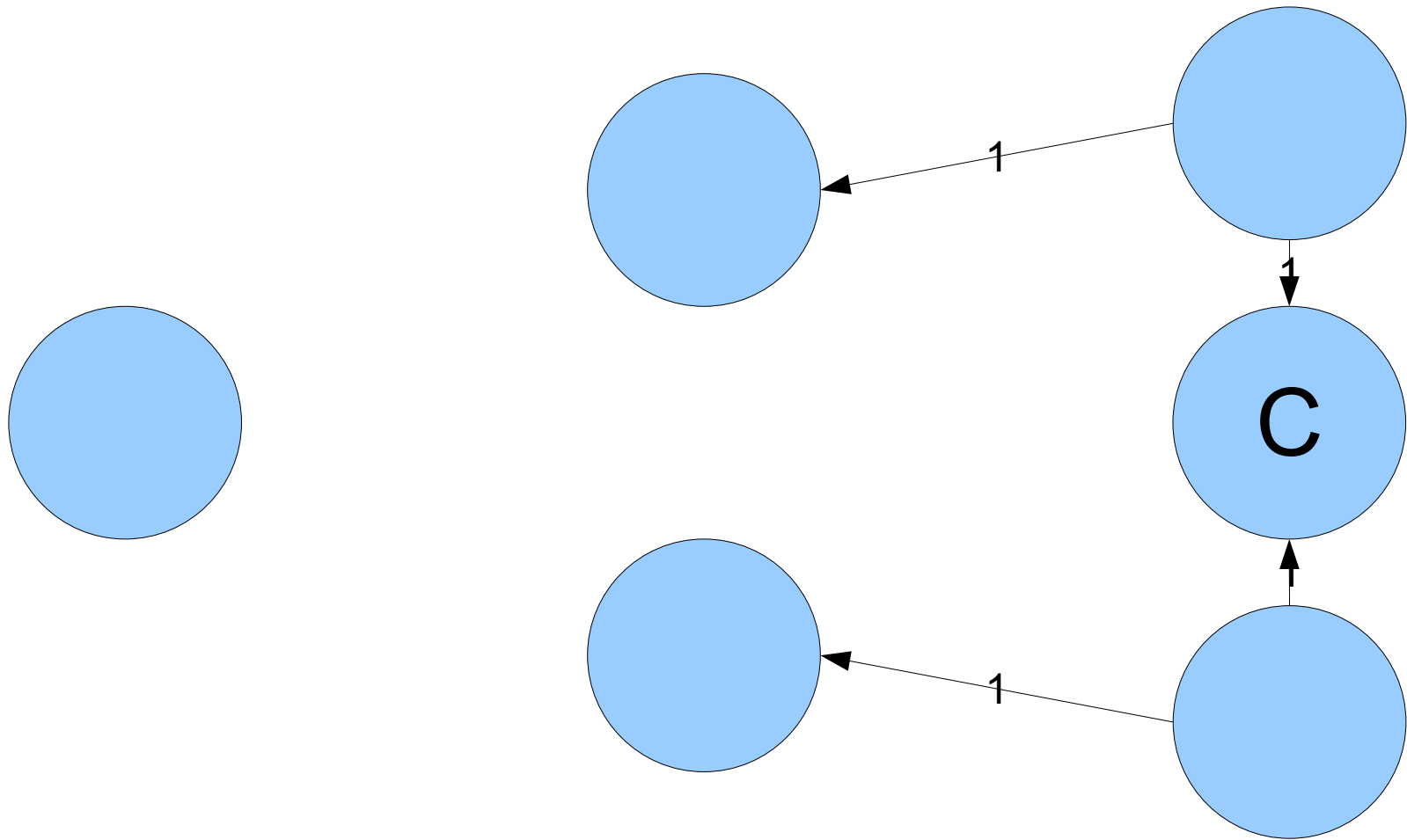
# Our Protocol



# Our Protocol



# Our Protocol



# Our Protocol

- Only 5 transmits – 3 saved (37.5%)
- Packets move only away from the source – 'outwards' through the network.

# Questions?

<https://github.com/mcover/uniwork-kelvin>

## References

- <http://www.ietf.org/rfc/rfc4944.txt>