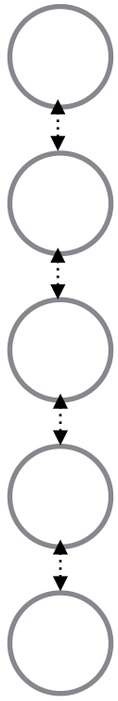
# [CE520] WSN Programming
# 2nd assignment
# Packet propagation to all nodes

Katerina Karakoula (1604)
Apostolos Tsaousis (1714)

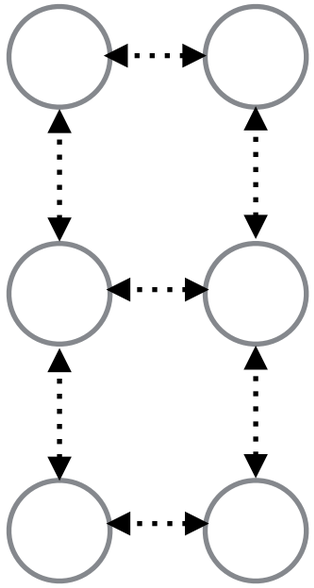
# Case study: Best-effort network-wide broadcast

- Given a topology of nodes (chain, tree, or grid)
- Each source node transmits a new packet periodically
- Each node that receives a packet, it propagates the packet to its neighbours
- Target is to distribute a packet through the whole network, to reach a steady silent state and to avoid collisions when transmitting packets

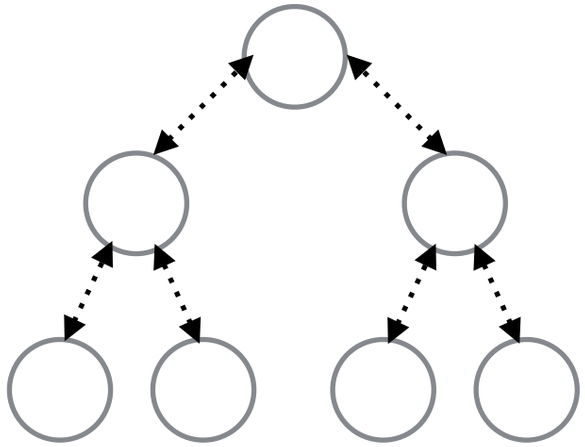
# Topologies



Chain



Grid



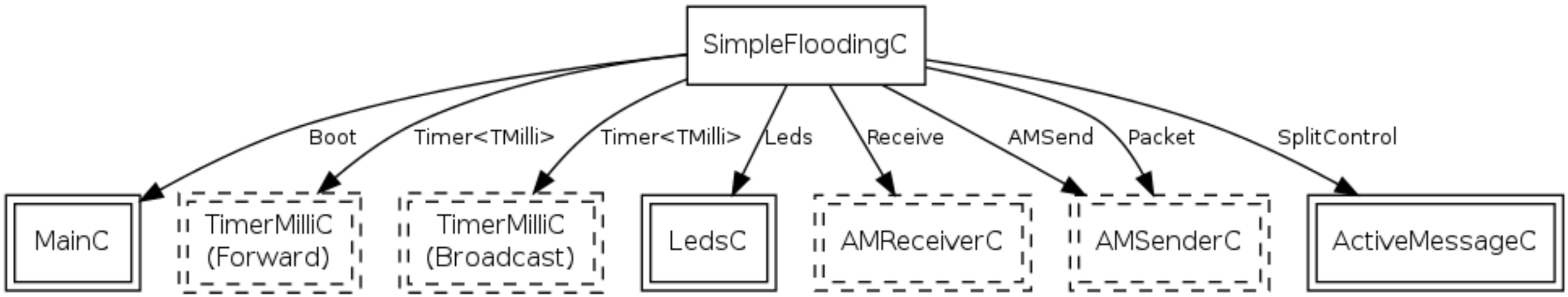
Tree

# Pseudocode

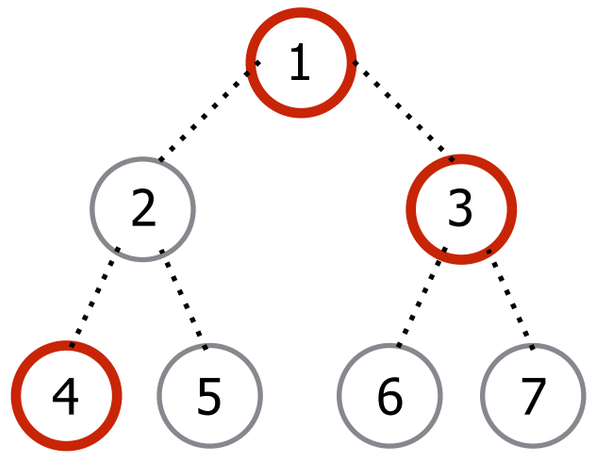
```
onBoot():
   initCache();
   startAMControl();

onAMControlDone():
   nodeID = TOS_NODE_ID;
   bcastPeriodic( PERIOD );

onBcastFired():
   if ( sendIsBusy ), back-off;
   pkt = createPacket();
   if ( bcast( pkt )  == SUCCESS )
       sendIsBusy = TRUE;
   if ( maxBcastsReached )
       stopBcastPeriodic();
```

```
onPktReceived():
   if ( pkt.getSourceID == nodeID ), return;
   if ( pkt.version < cache.version ), return;

   addCache( pkt );
   schedForward( PERIOD );

onForwardFired():
   if ( sendIsBusy ), schedForward( PERIOD );
   else
       pkt = getPktFromCache();
       if ( bcast( pkt )  == SUCCESS )
           sendIsBusy = TRUE;
```

# Structure (Wiring) of the Algorithm



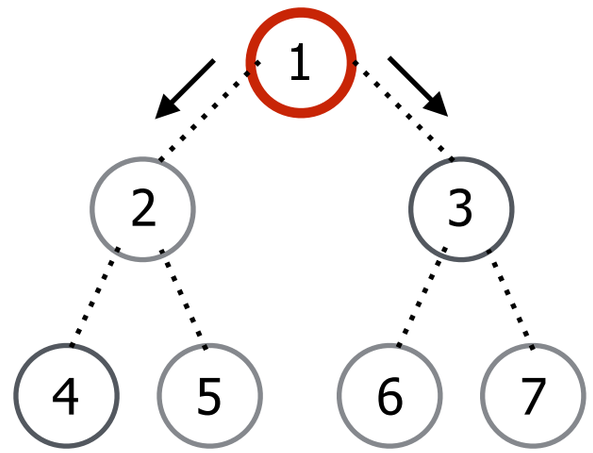
# Demo




(*) The sources are indicated with red border

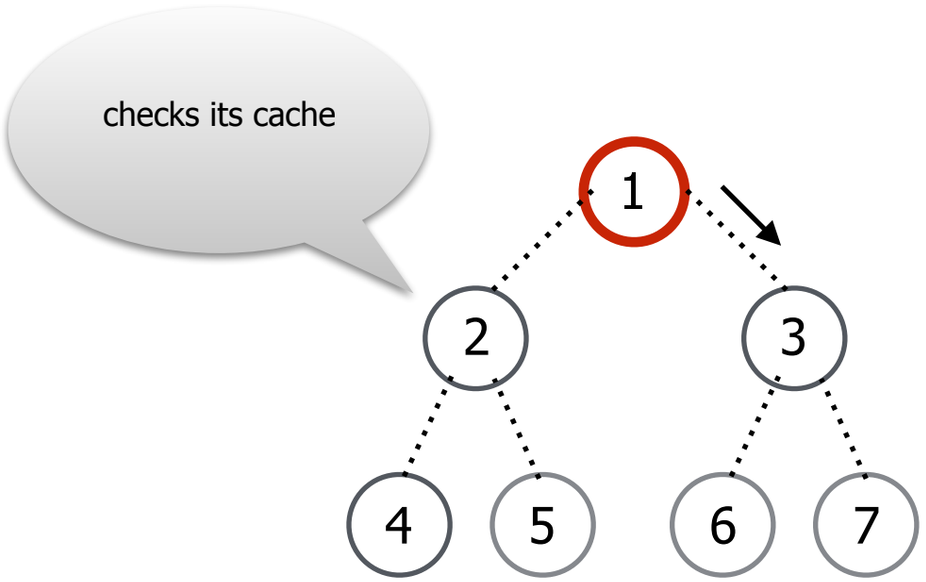
(**) The dotted lines indicate which nodes are in range

# Flooding



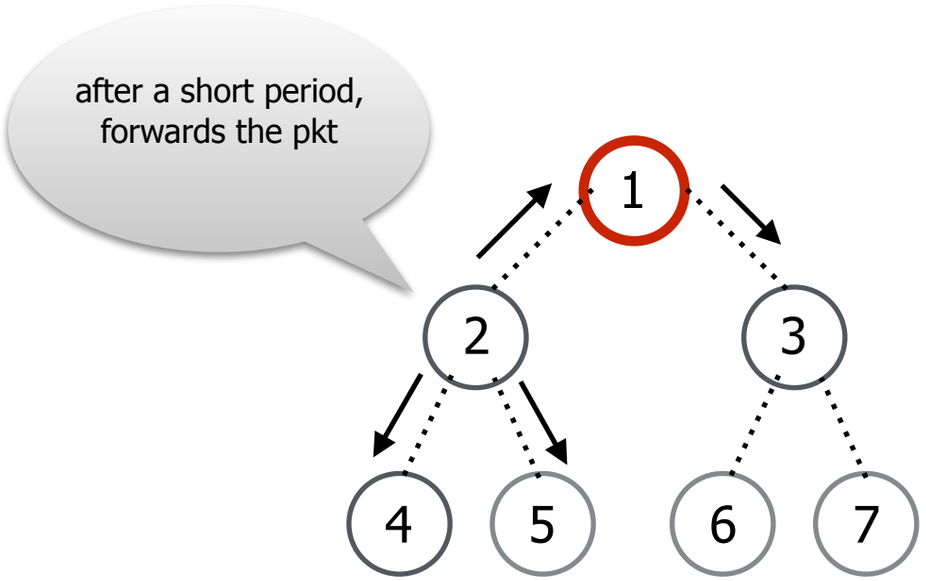

Node 1 broadcasts a packet

# Flooding



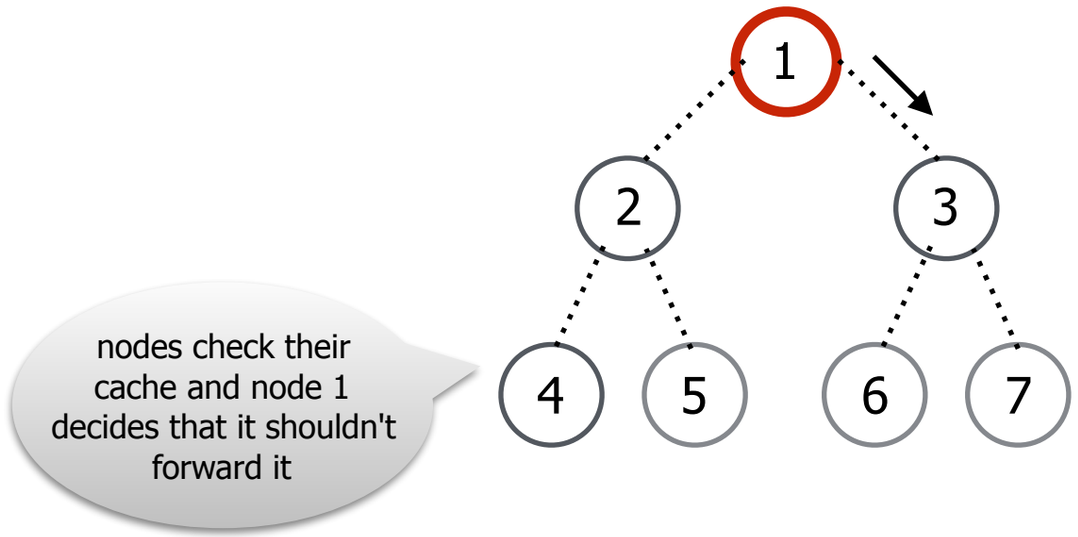

Node 2 receives the packet

# Flooding



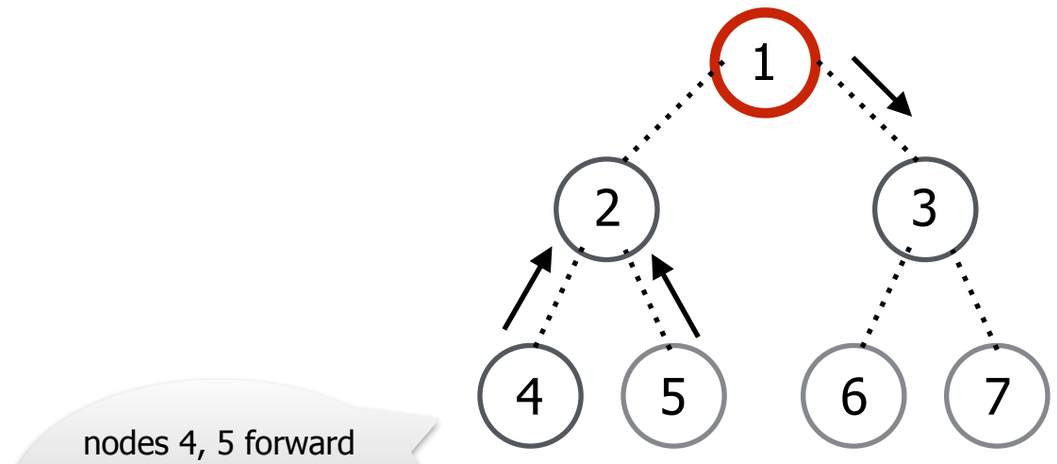

Node 2 forwards the packet

# Flooding




Nodes 1,4,5 receive the packet

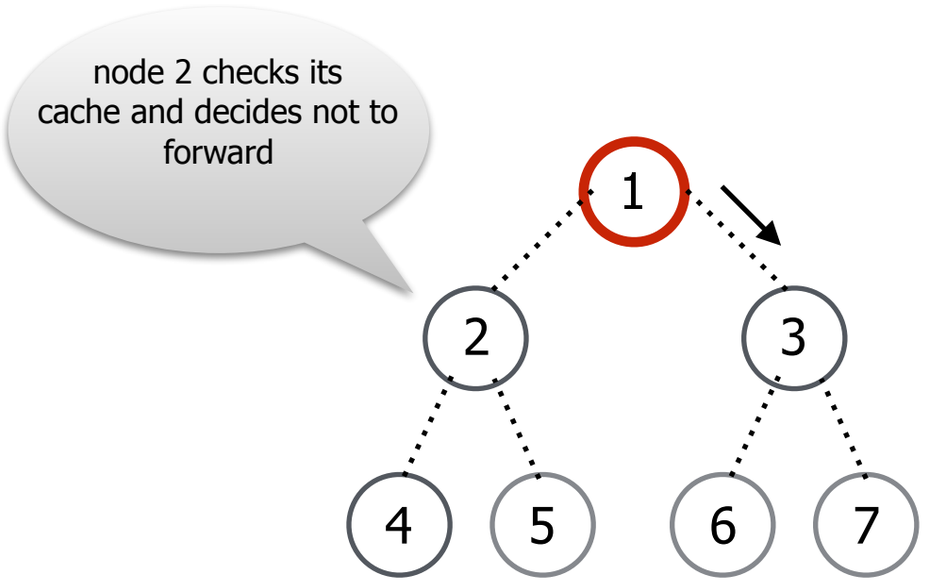
# Flooding




nodes 4, 5 forward the packet after a small period

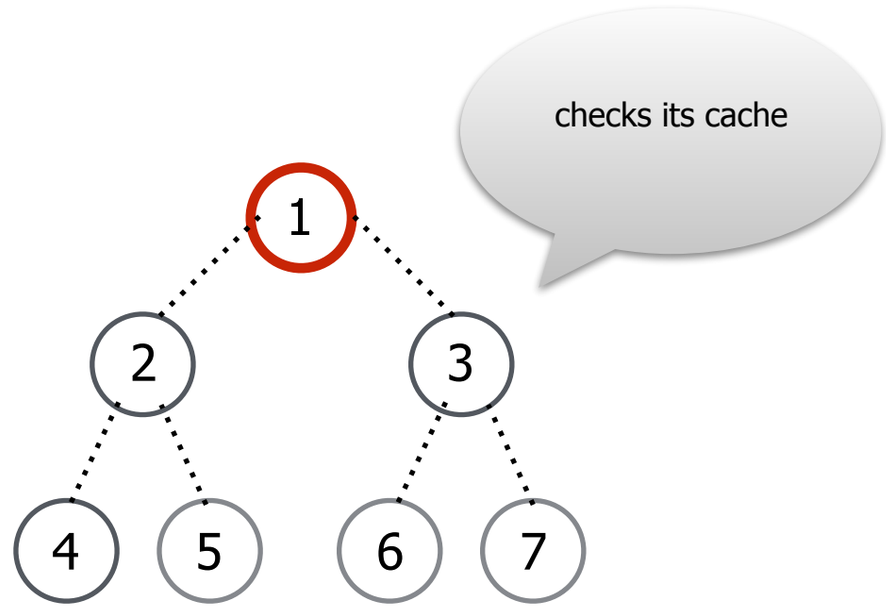
Node **4,5** forward the packet

# Flooding



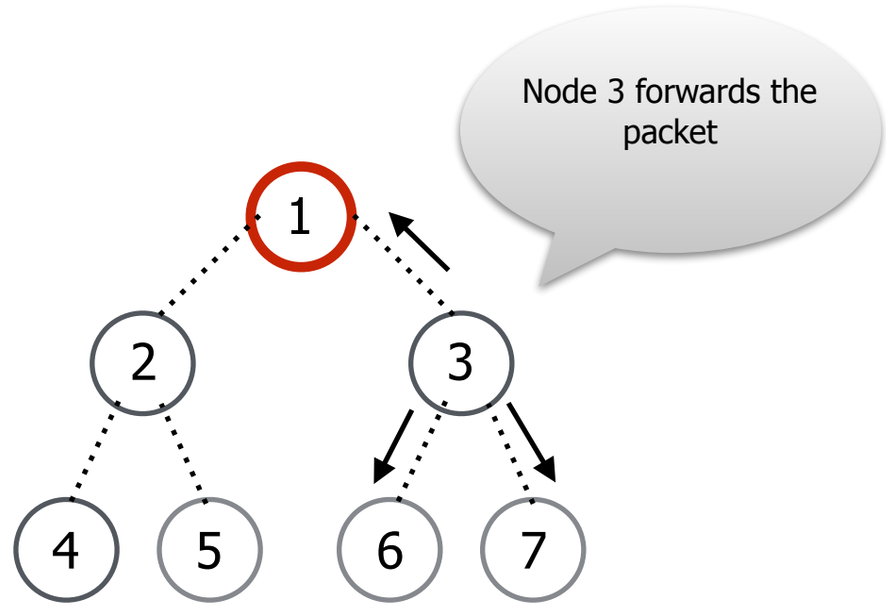

Node 2 receives the packet

# Flooding



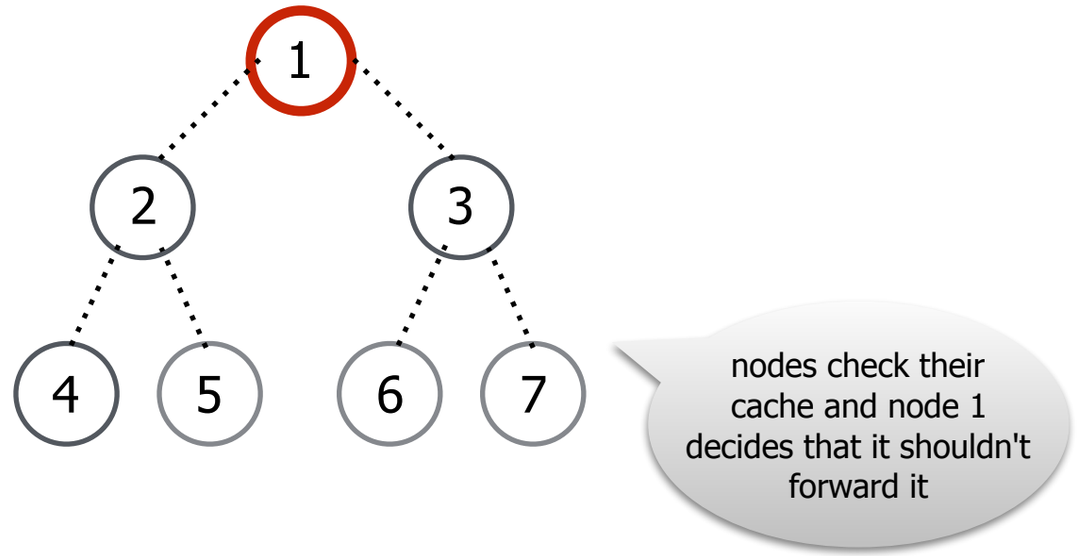

Nodes 3 receives the packet

# Flooding



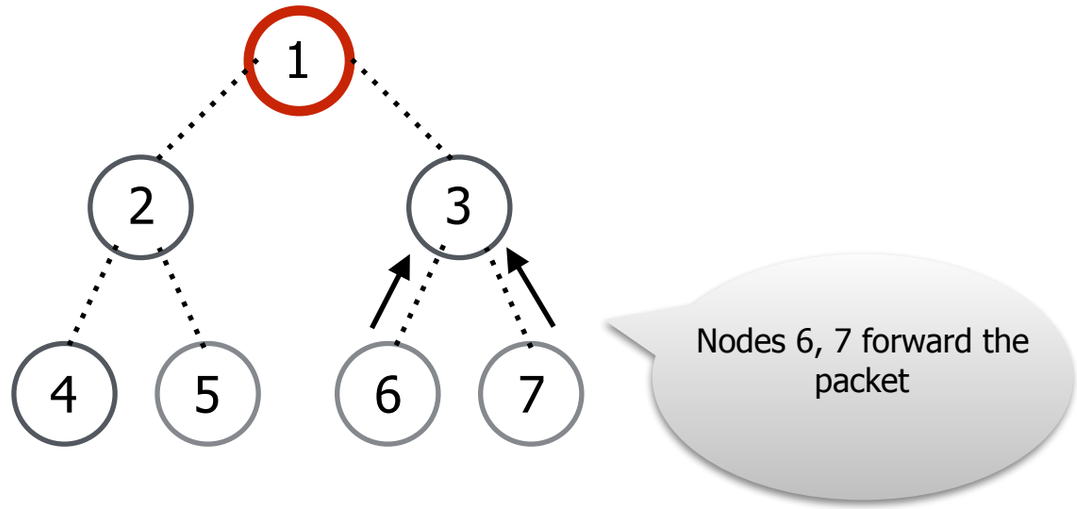

Nodes 3 forwards the packet

# Flooding



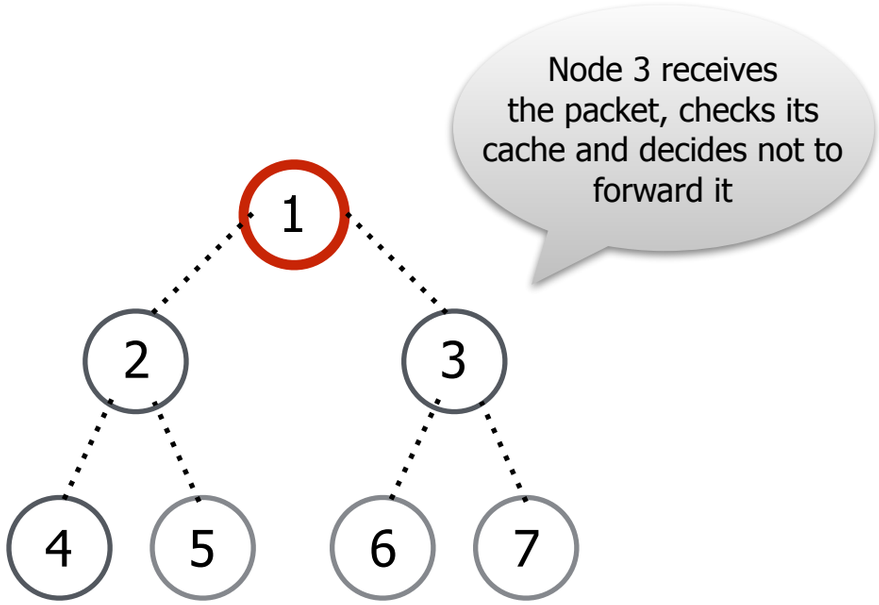

Nodes 1,6,7 receive the packet

# Flooding



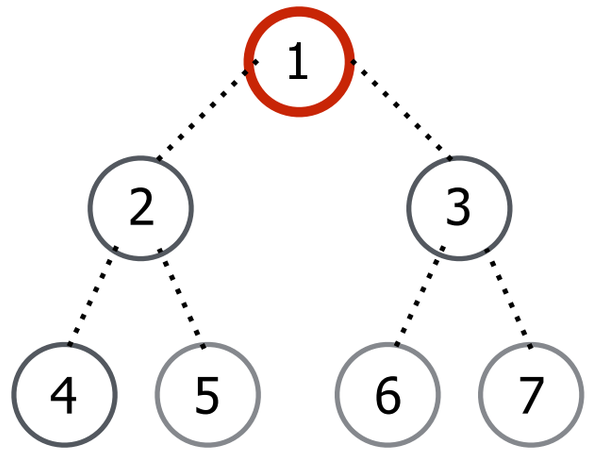

Nodes **6,7** forward the packet

# Flooding




Nodes 3 receives the packet

# Flooding




System reaches a silent state

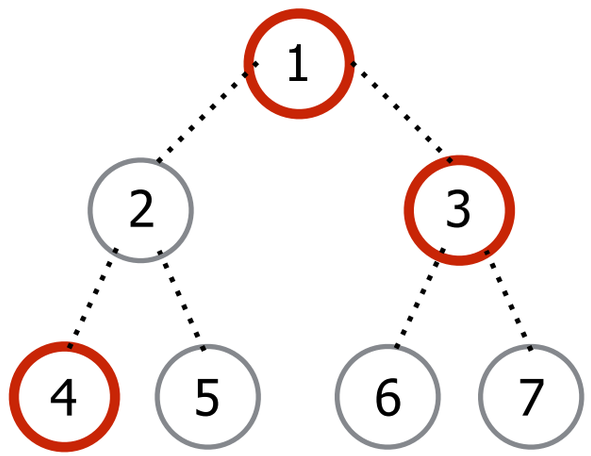
# Cache functionality

Packets ( SrcID / SeqNo )

NodeID

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | 1/2 | 1/3 | 4/2 | 4/3 |

Cache




(*) Green arrow indicates the latest saved packet

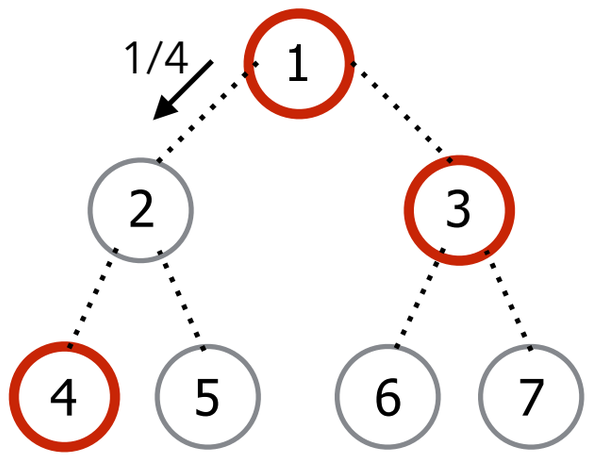
(*) Purple arrow indicates the next position for forwarding

# Cache functionality

Packets ( SrcID / SeqNo )

NodeID

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **2** | 1/2 | 1/3 | **1/4** | 4/3 |



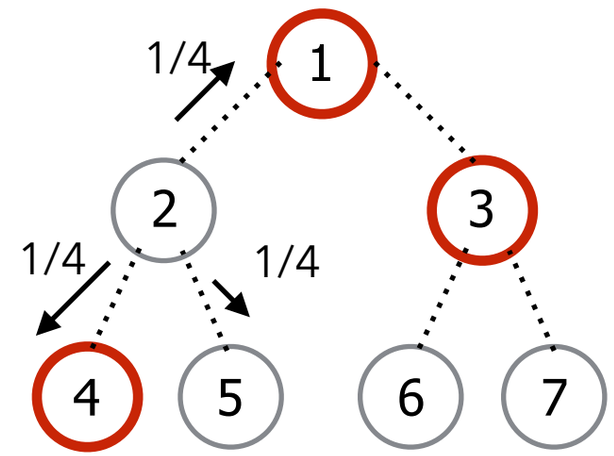

Node **2** receives a packet with tag **1/4**. Then checks it cache and since it doesn't find a packet from the same source with latter seqNo, it saves the packet, advances the green arrow and schedules it for forwarding

# Cache functionality

Packets ( SrcID / SeqNo )

NodeID

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | 1/2 | 1/3 | 1/4 | 4/3 |

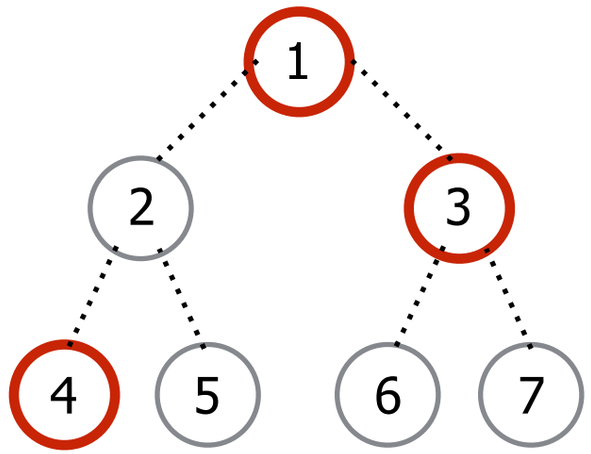



Node 2 forwards the scheduled packet and the purple arrow advances

# Algorithms weak point

Packets ( SrcID / SeqNo )

| NodeID | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | 3/4 | 1/3 | 1/4 | 4/3 |



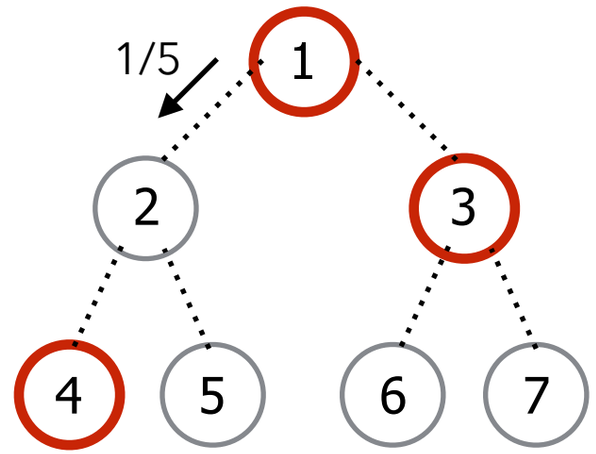

Initial scenario

# Algorithms weak point

Packets ( SrcID / SeqNo )

NodeID

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | 3/4 | 1/3 | 1/4 | **1/5** |

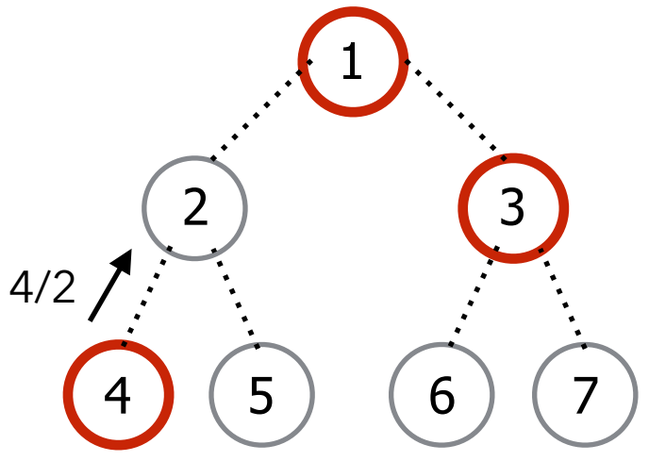



Node 2 received packet **1/5,** saves the packet and schedules it for forwarding

# Algorithms weak point

Packets ( SrcID / SeqNo )

| NodeID | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | **4/2** | 1/3 | 1/4 | 1/5 |



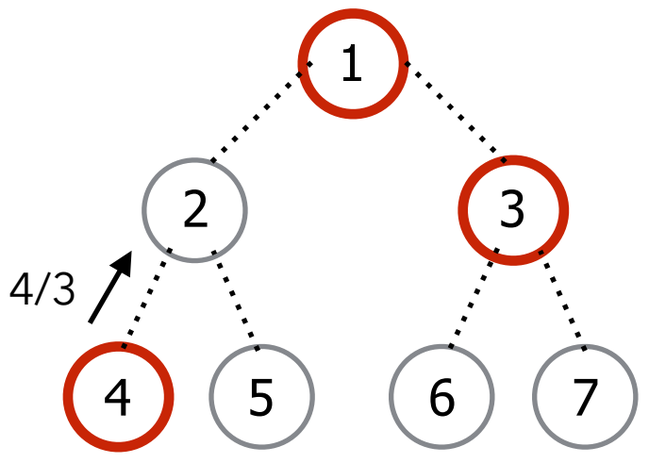

Node 2 before forwarding the packet with tag 1/5, receives another packet

# Algorithms weak point

Packets ( SrcID / SeqNo )

NodeID

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | 4/2 | **4/3** | 1/4 | 1/5 |



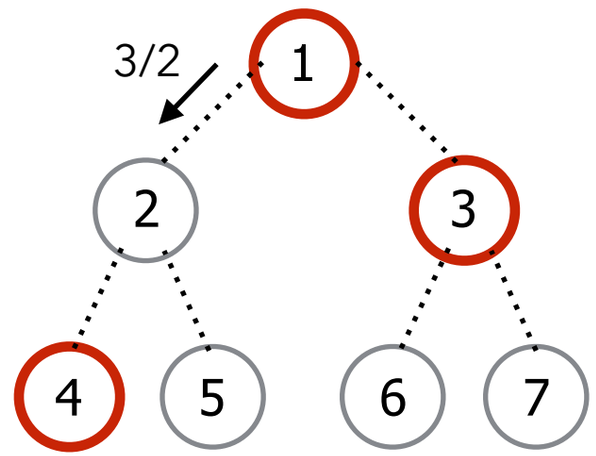

Node 2 before forwarding the packet with tag 1/5, receives yet another packet

# Algorithms weak point

Packets ( SrcID / SeqNo )

NodeID

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | 4/2 | 4/3 | **3/2** | 1/5 |



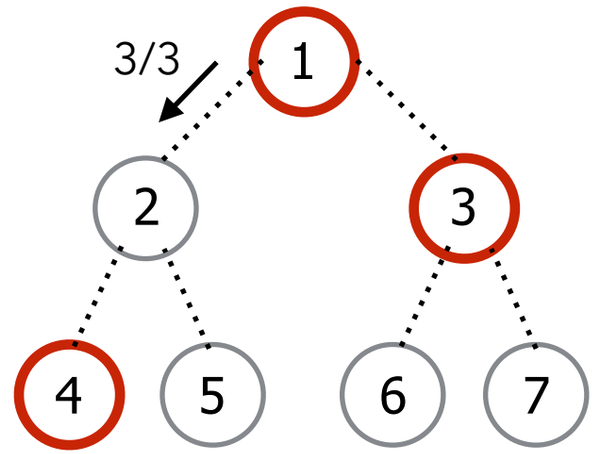

Node 2 before forwarding the packet with tag 1/5, receives yet another packet

# Algorithms weak point

Packets ( SrcID / SeqNo )

NodeID

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 2 | 4/2 | 4/3 | 3/2 | **3/3** |



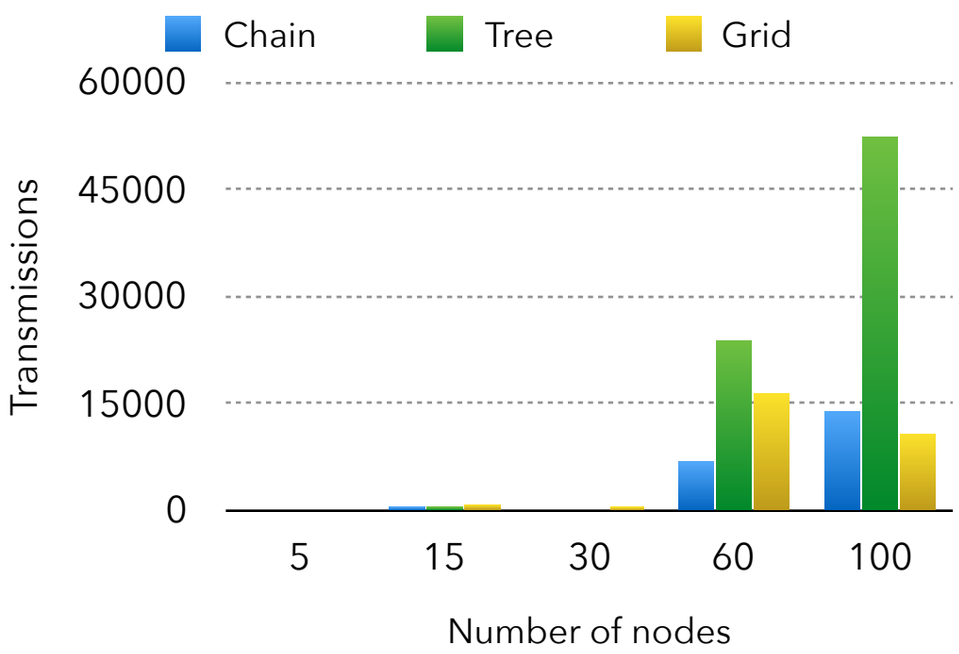

Node 2 before forwarding the packet with tag 1/5, receives yet another packet.

It saves the received the packet and overwrites the 1/5 packet.

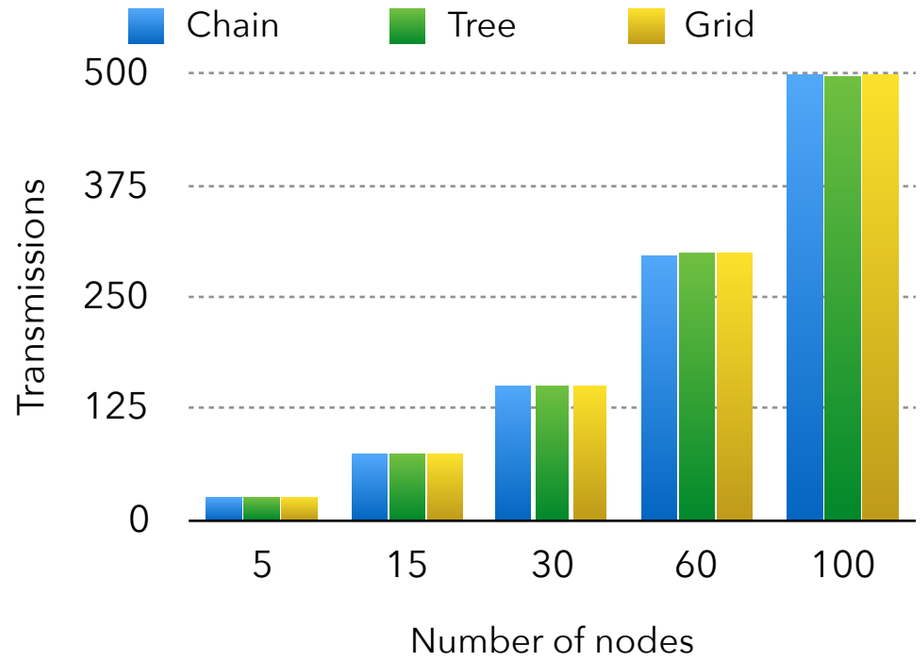
So then nodes 4,5 never receive that overwrited packet

# Total transmissions

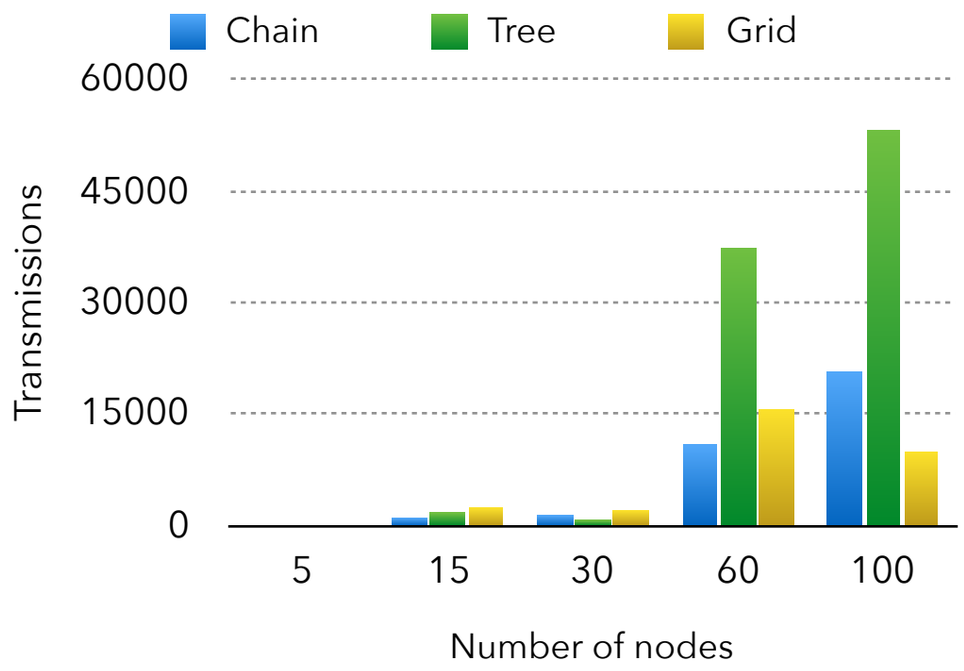
## Odd senders



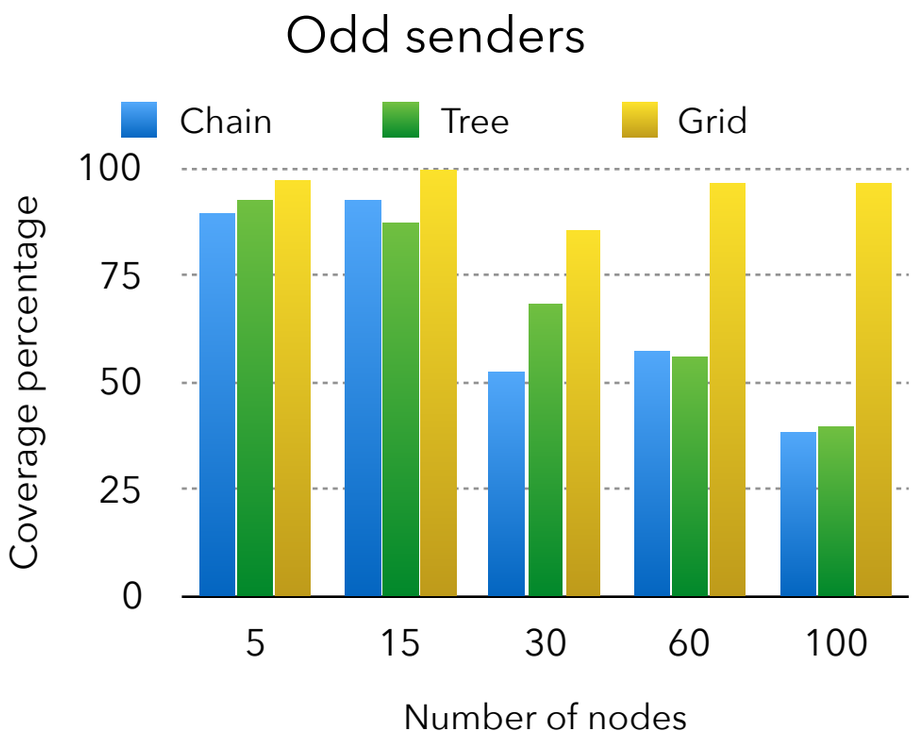

## One sender

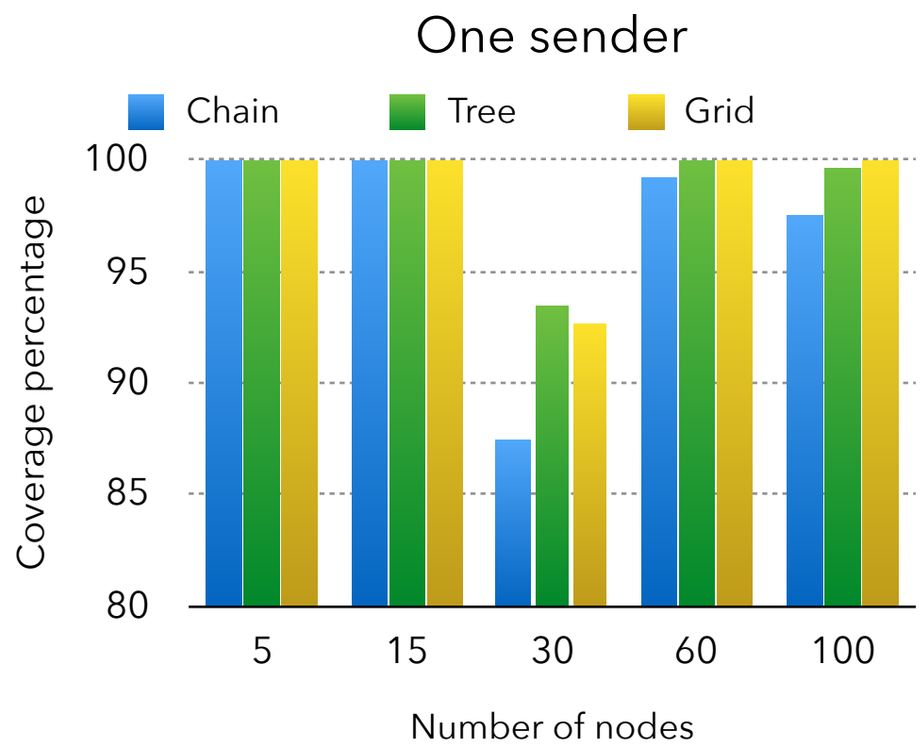


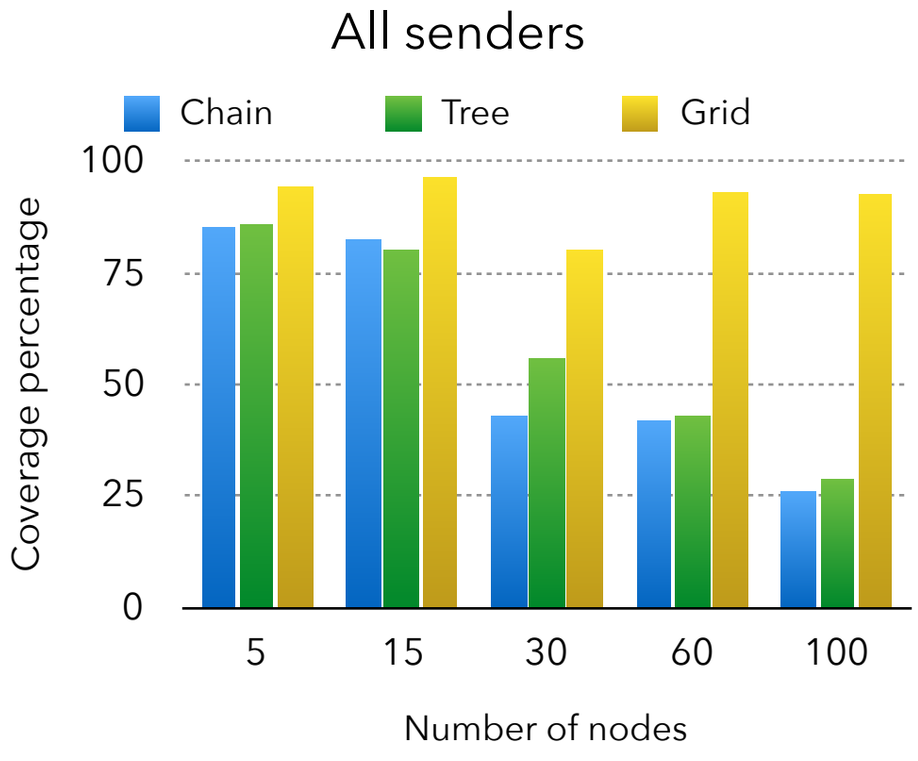

## All senders



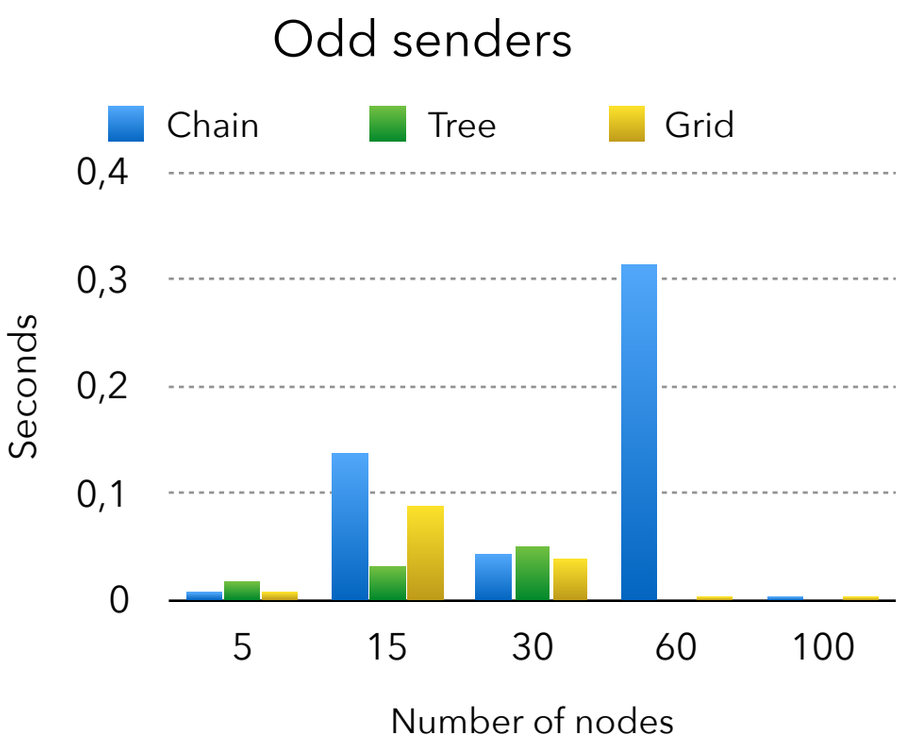
# Coverage

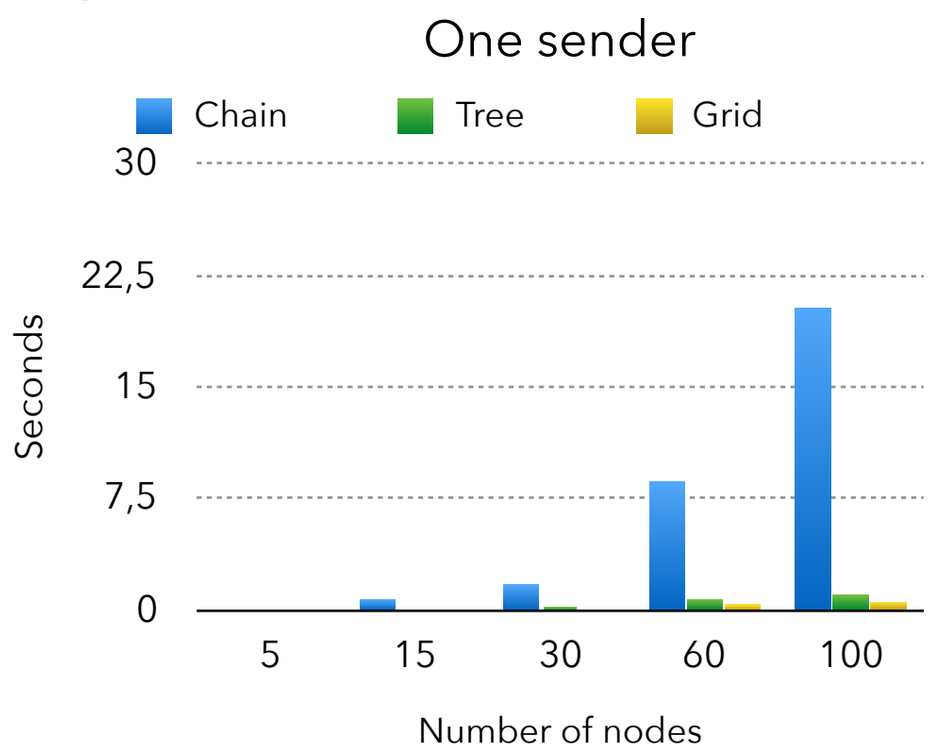

Odd senders
Chain
Tree
Grid
100
75
50
25
0
Coverage percentage
5
15
30
60
100
Number of nodes

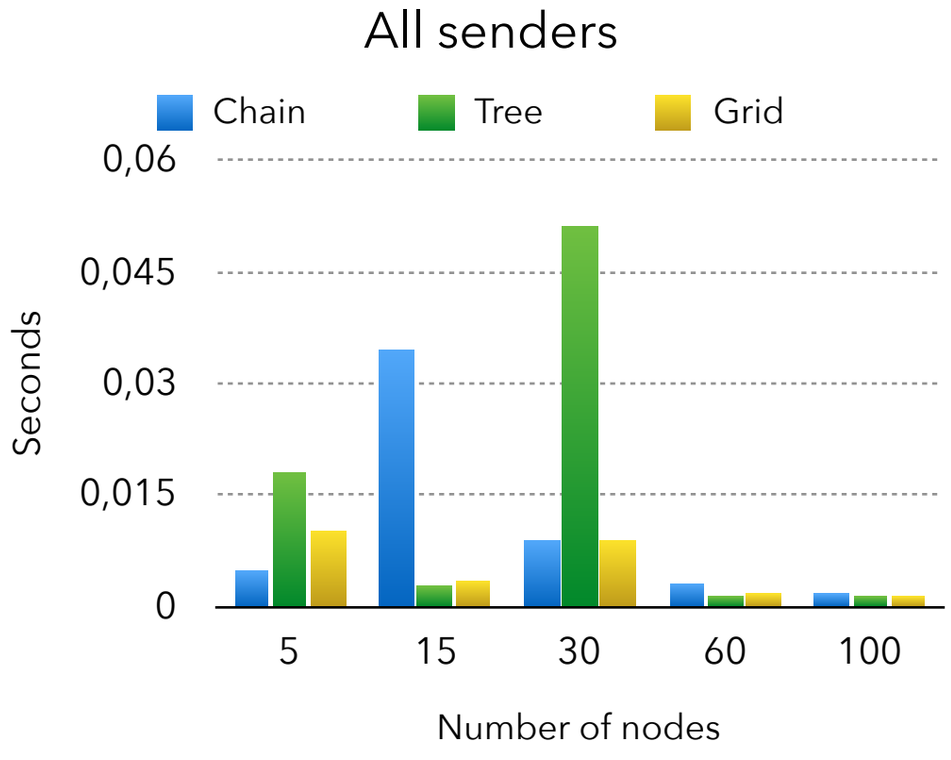

One sender
Chain
Tree
Grid
100
95
90
85
80
Coverage percentage
5
15
30
60
100
Number of nodes


All senders
Chain
Tree
Grid
100
75
50
25
0
Coverage percentage
5
15
30
60
100
Number of nodes

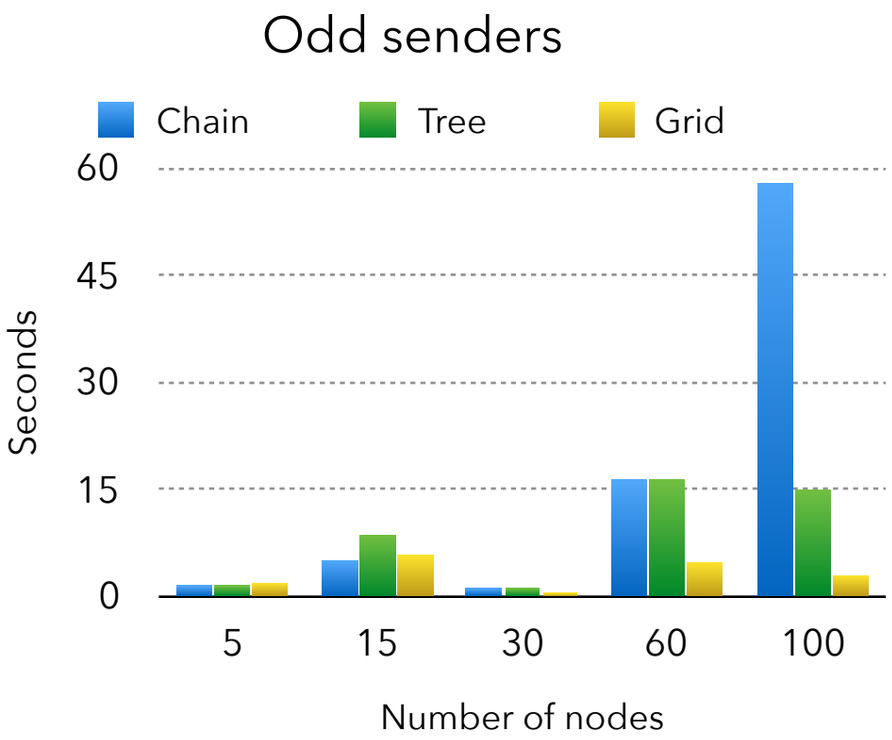
# Min. Latency

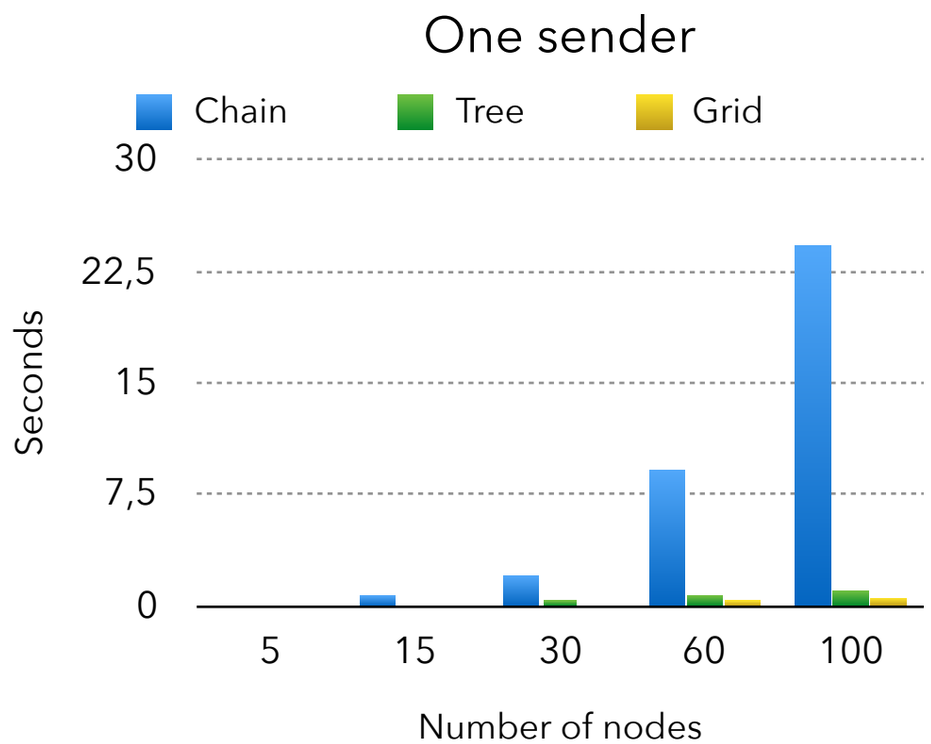

Odd senders
Chain
Tree
Grid
0,4
0,3
0,2
0,1
0
Seconds
5
15
30
60
100
Number of nodes

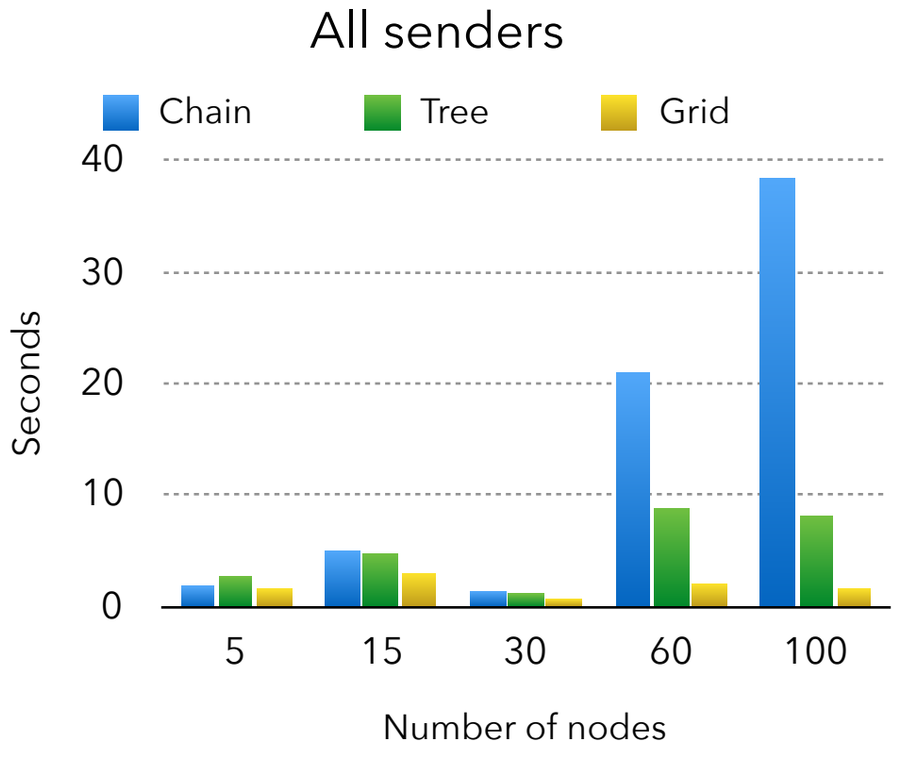

One sender
Chain
Tree
Grid
30
22,5
15
7,5
0
Seconds
5
15
30
60
100
Number of nodes


All senders
Chain
Tree
Grid
0,06
0,045
0,03
0,015
0
Seconds
5
15
30
60
100
Number of nodes

# Avg. Latency

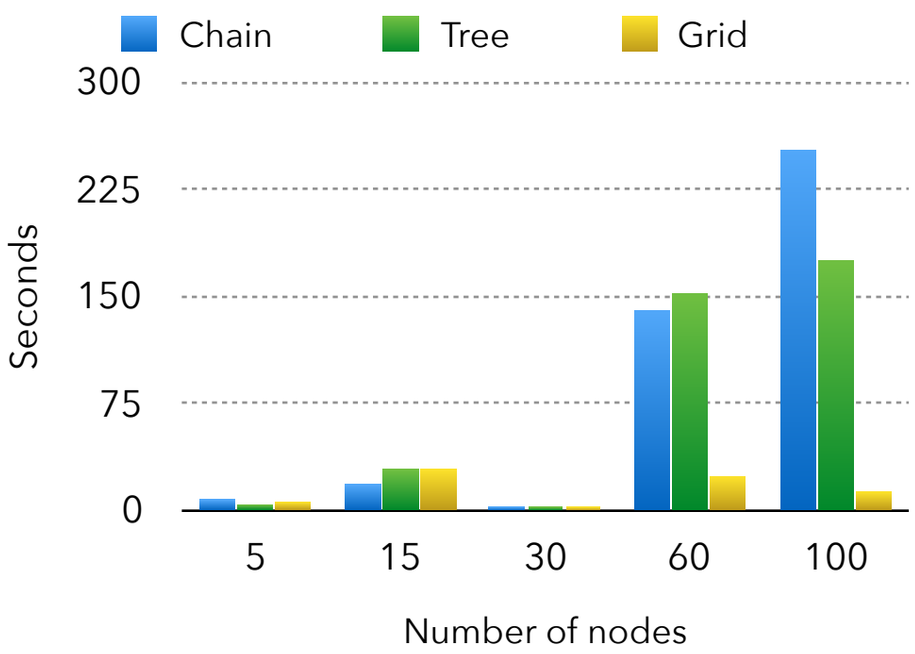

Odd senders
Chain
Tree
Grid
60
45
30
15
0
Seconds
5
15
30
60
100
Number of nodes


One sender
Chain
Tree
Grid
30
22,5
15
7,5
0
Seconds
5
15
30
60
100
Number of nodes

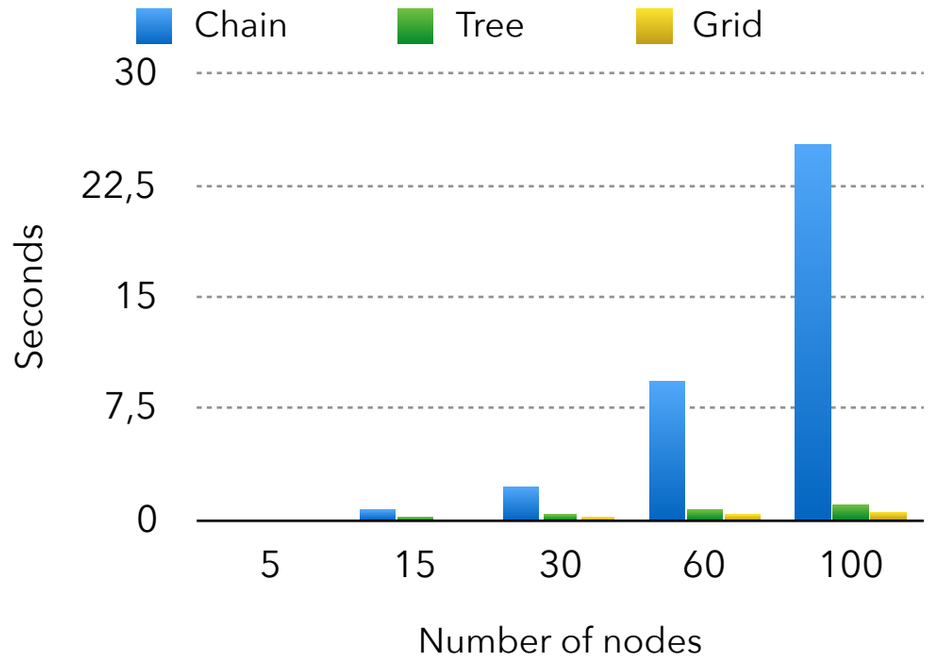

All senders
Chain
Tree
Grid
40
30
20
10
0
Seconds
5
15
30
60
100
Number of nodes

# Max. Latency

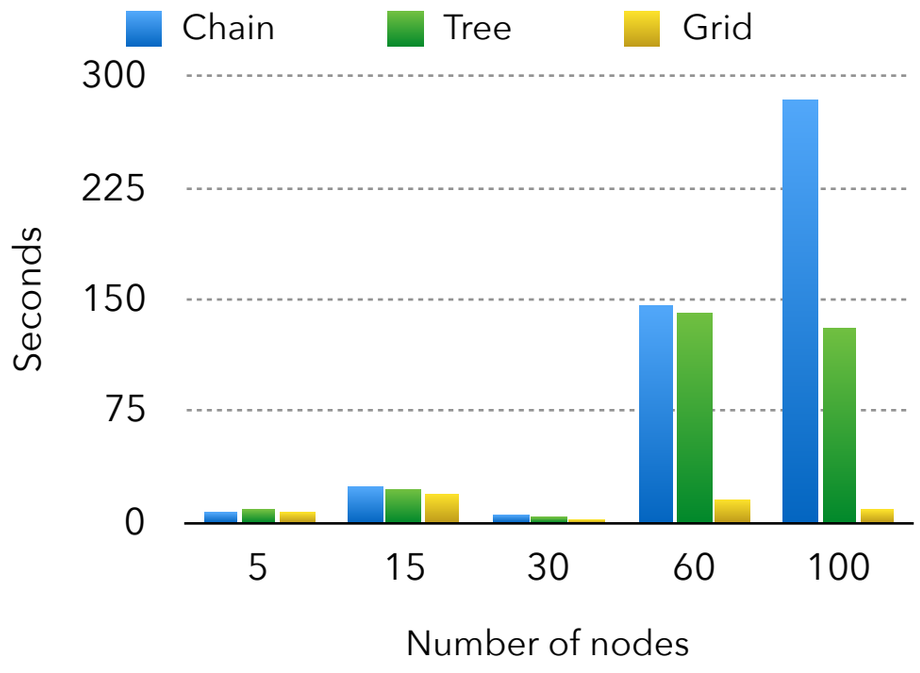
## Odd senders



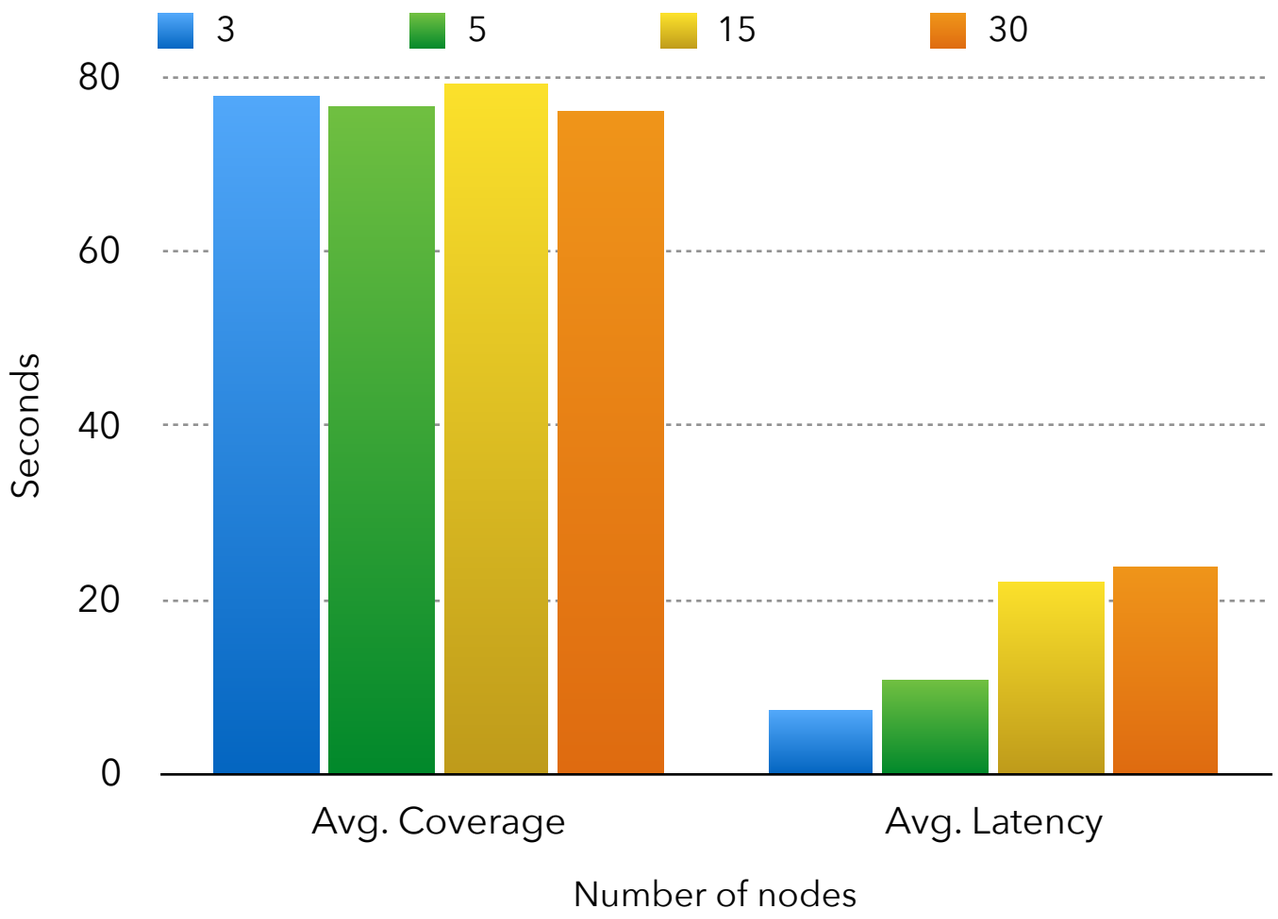

## One sender




## All senders



# Cache size


3
5
15
30
80
60
40
20
0
Seconds
Avg. Coverage
Avg. Latency
Number of nodes