Ahoy: A Proximity-Based Discovery Protocol

Robbert Haarman

University of Twente
The Netherlands

University of Twente
The Netherlands

Contents

Part 1

Introduction to Ahoy

University of Twente
The Netherlands

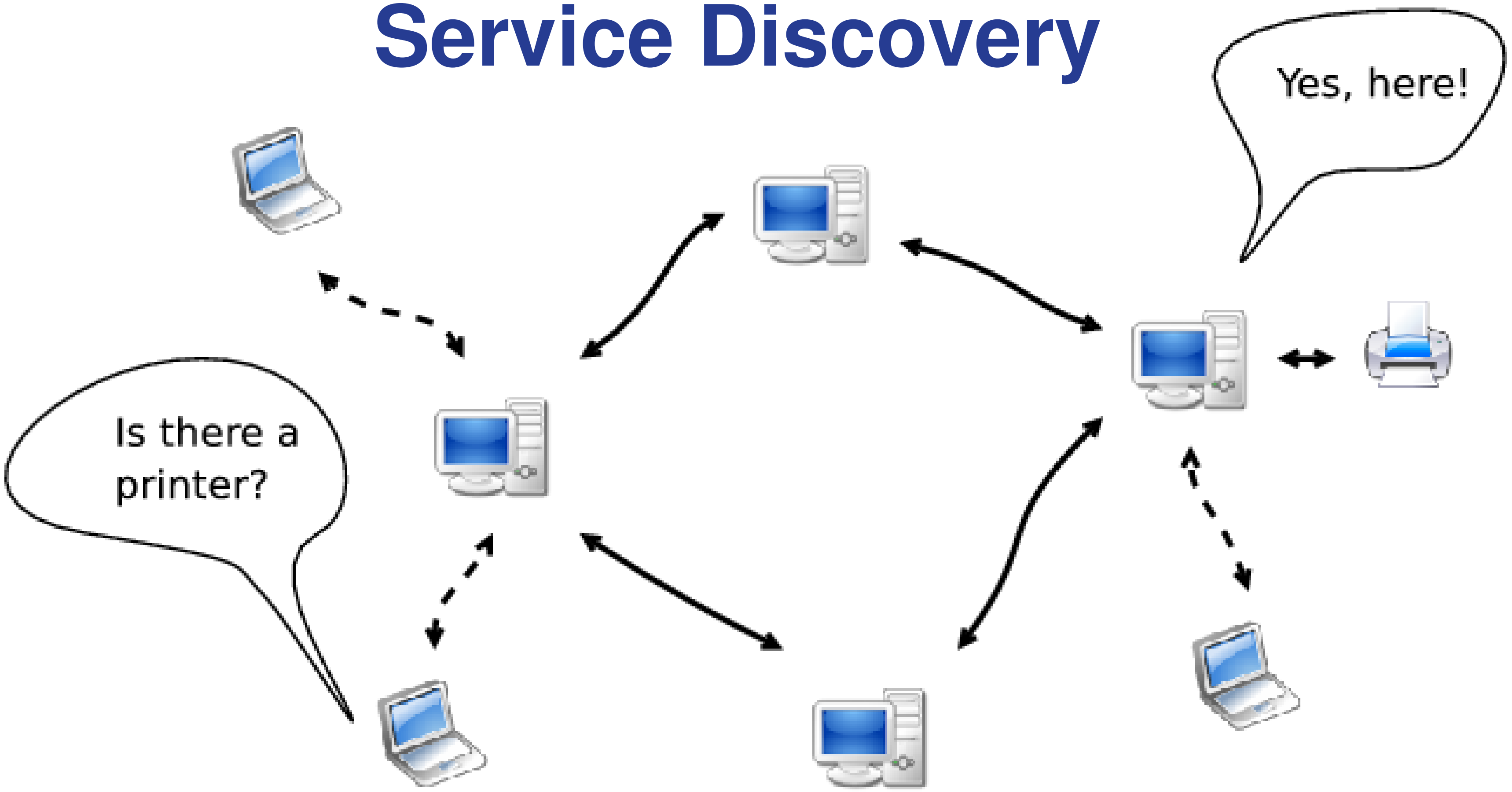
What is Ahoy?

1. Ahoy is a *service discovery* protocol.
2. Ahoy is designed for *mobile ad-hoc networks.*
3. Ahoy is *decentralized.*
4. Ahoy is *efficient.*

What is Ahoy?

1. **Ahoy is a *service discovery* protocol.**
2. Ahoy is designed for *mobile ad-hoc networks.*
3. Ahoy is *decentralized.*
4. Ahoy is *efficient.*

Service Discovery

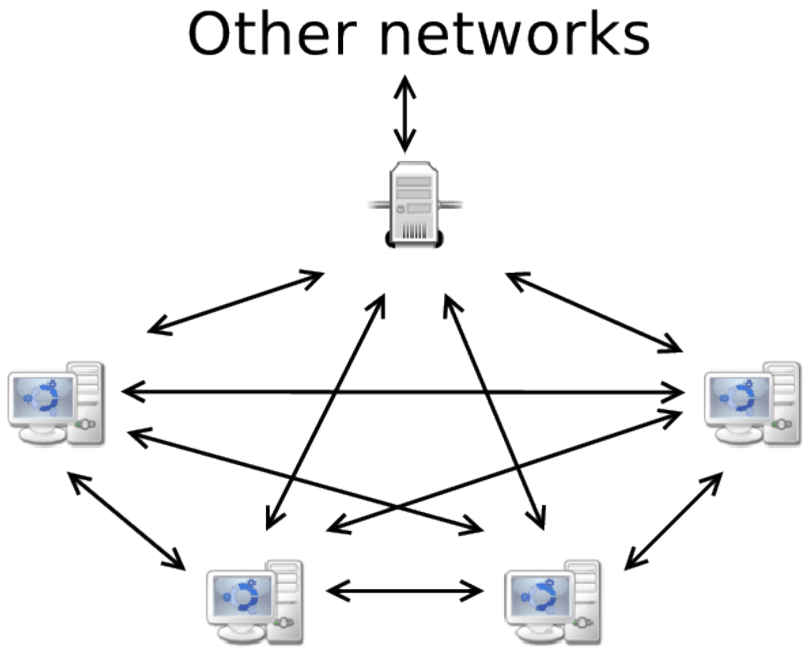


What is Ahoy?

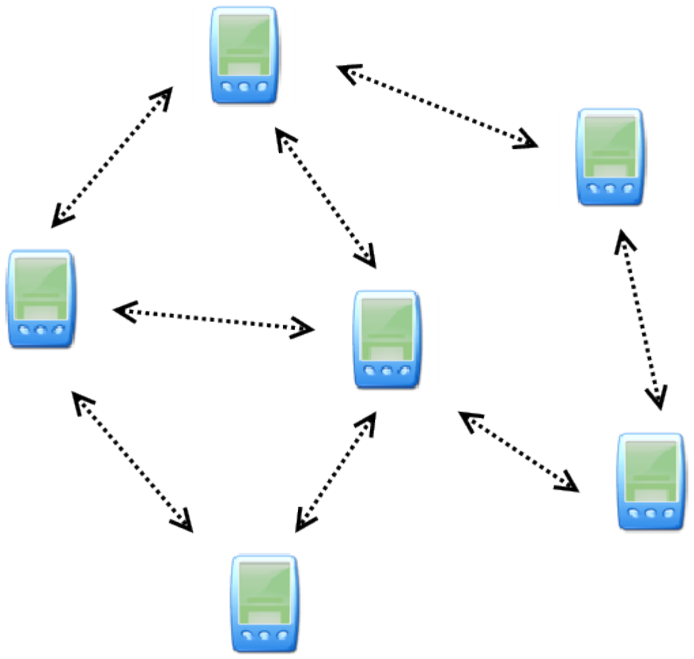
1. Ahoy is a *service discovery* protocol.
2. **Ahoy is designed for *mobile ad-hoc networks.***
3. Ahoy is *decentralized.*
4. Ahoy is *efficient.*

University of Twente
The Netherlands

Traditional Infrastructure Network



Mobile Ad-Hoc Network



Challenges

- Services may be located multiple hops away
- Connectivity may change
- Limited resources

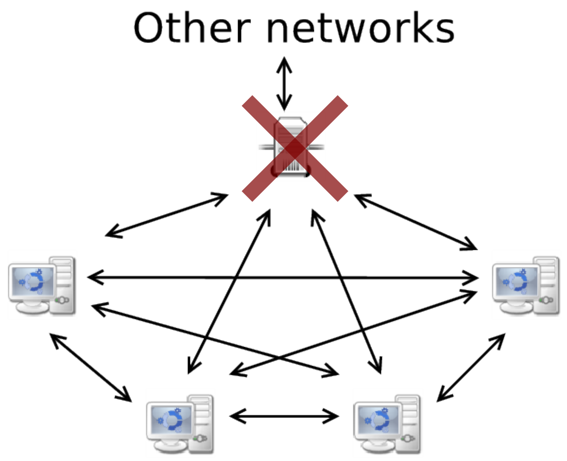
University of Twente
The Netherlands

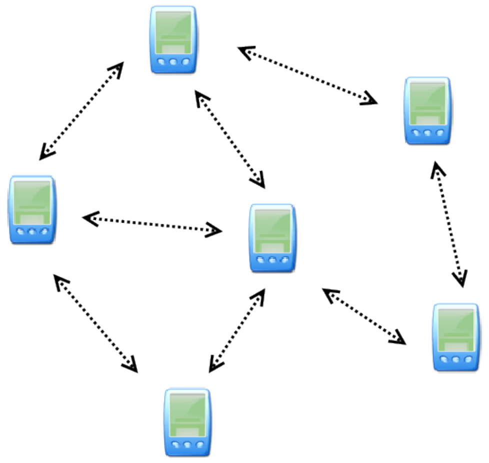
What is Ahoy?

1. Ahoy is a *service discovery* protocol.
2. Ahoy is designed for *mobile ad-hoc networks.*
3. **Ahoy is *decentralized.***
4. Ahoy is *efficient.*

Decentralized

- No reliance on a central authority





- Helps deal with connectivity changes

What is Ahoy?

1. Ahoy is a *service discovery* protocol.
2. Ahoy is designed for *mobile ad-hoc networks.*
3. Ahoy is *decentralized.*
4. **Ahoy is *efficient.***

Efficiency

- Ahoy sends few messages
- Ahoy sends small messages

How is Efficiency Accomplished?

- Do not send all information to everyone
- Do not send all information

University of Twente
The Netherlands

Part 2

Protocol Overview

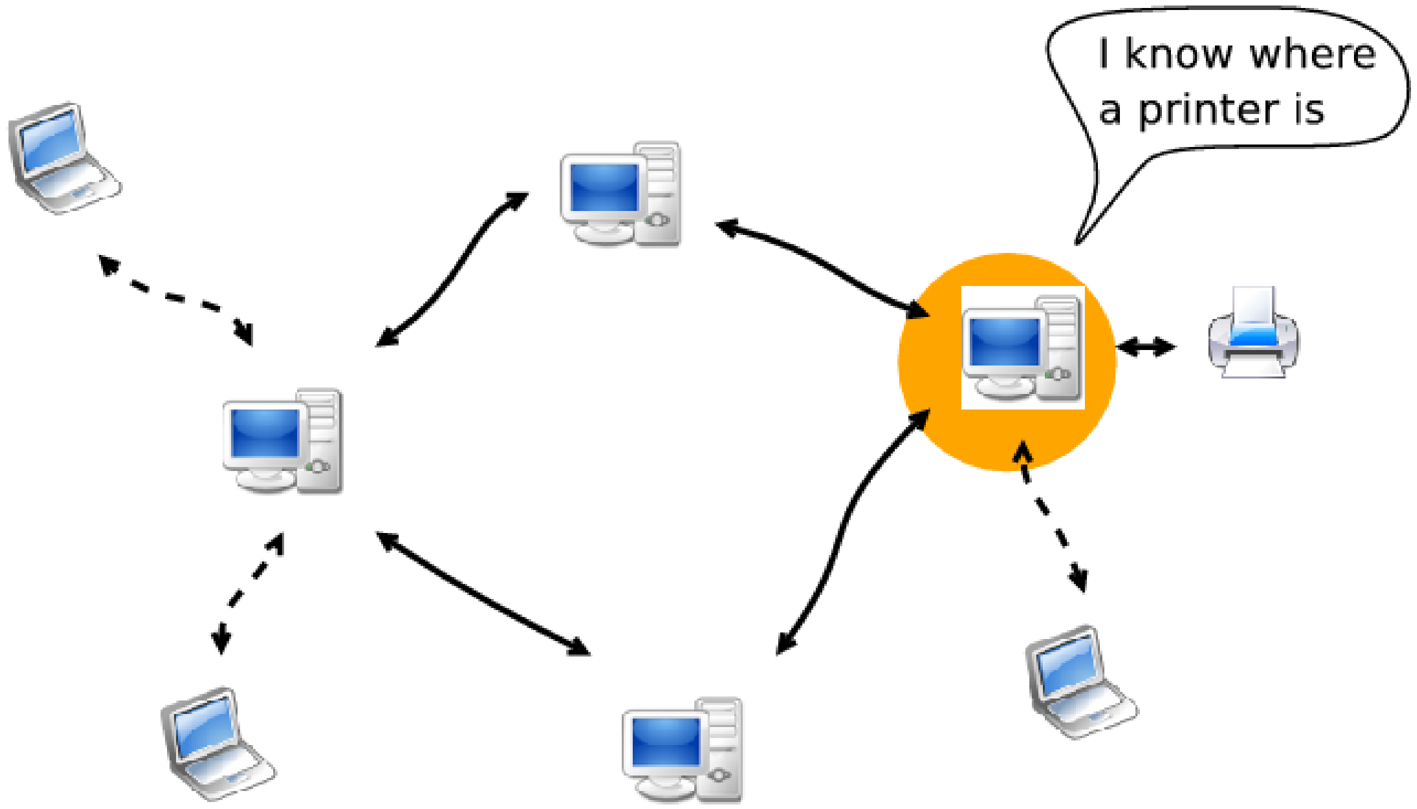
University of Twente
The Netherlands

Protocol Overview

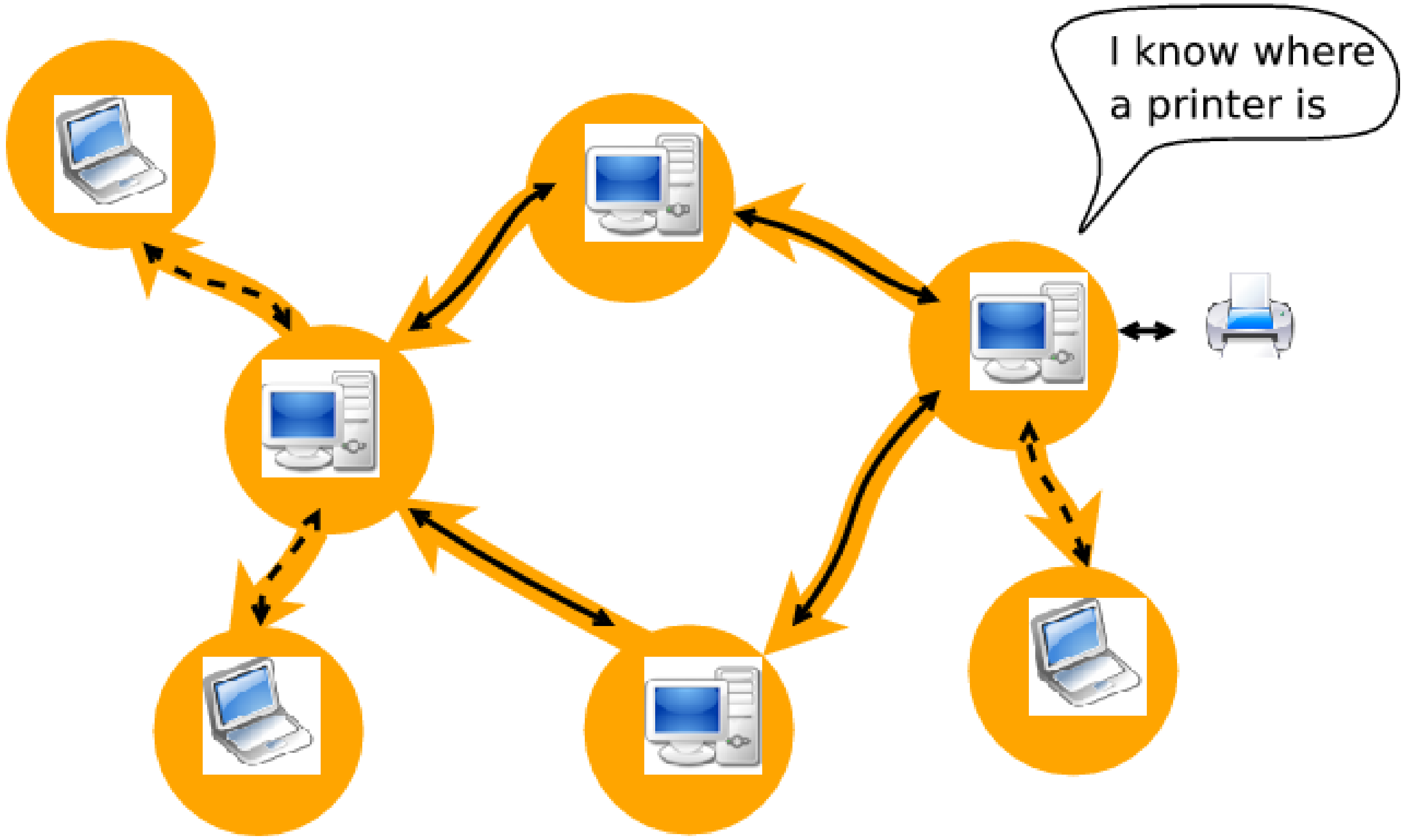
- Tell where information can be found
- When information is needed, find it

University of Twente
The Netherlands

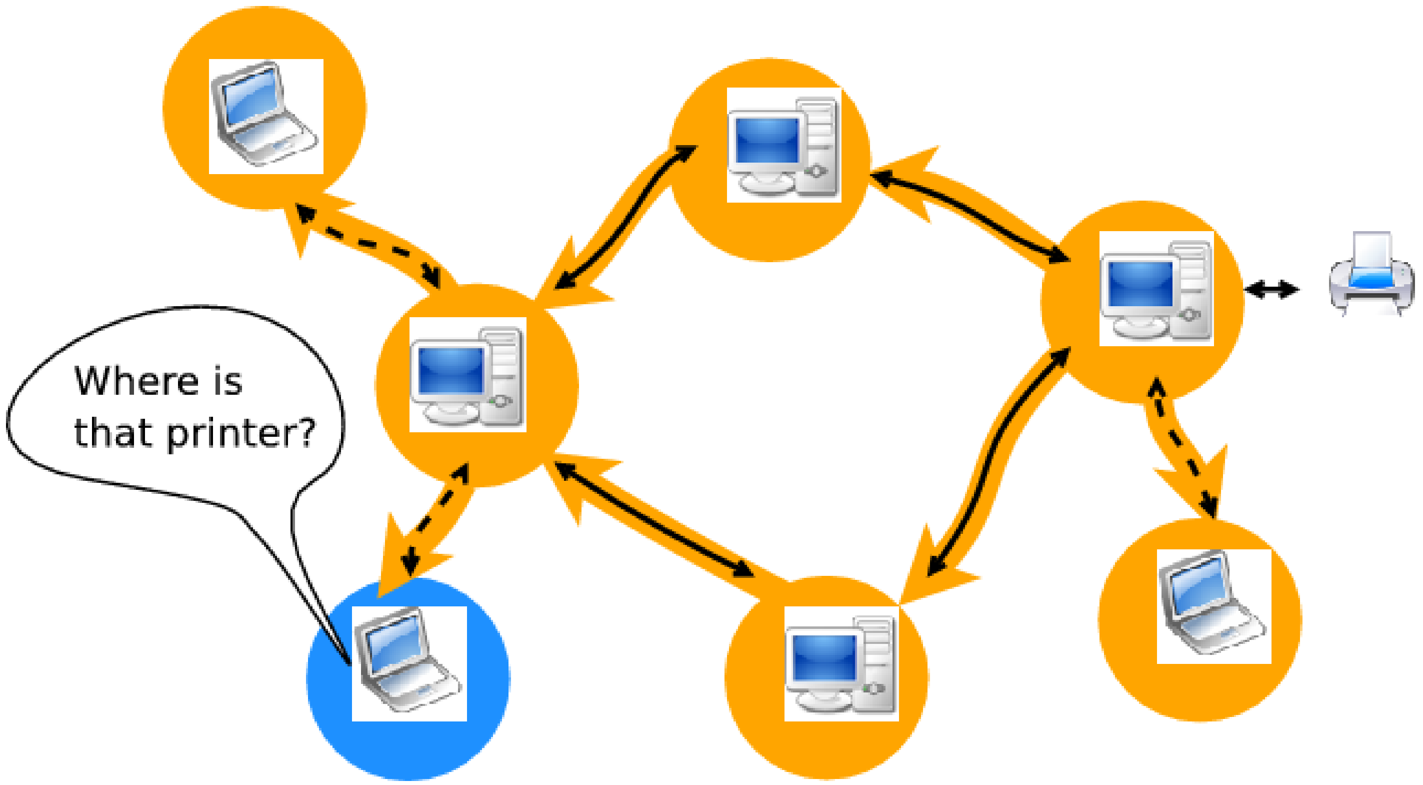
Ahoy



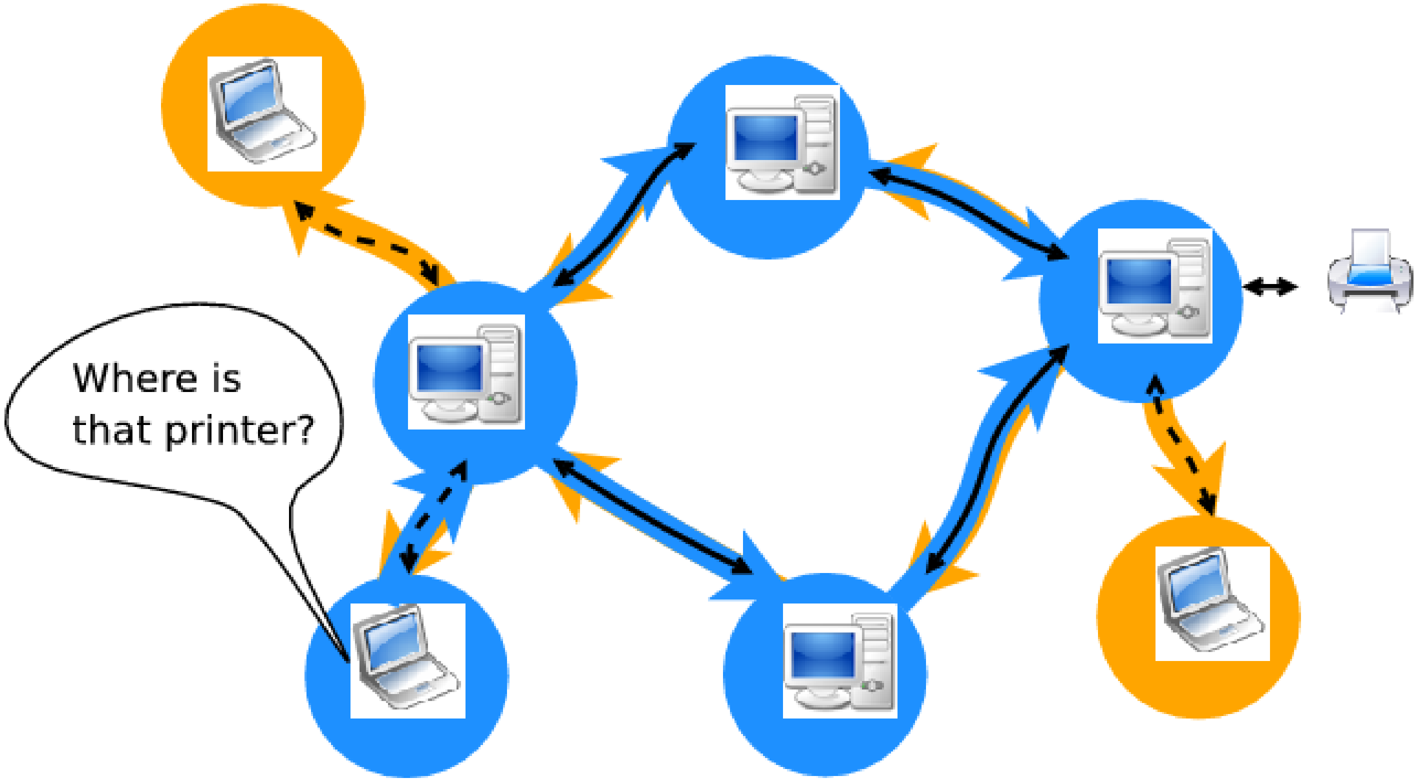
Ahoy



Ahoy



Ahoy



Part 3

Message Types

Five Message Types

- Announcements
- Queries
- Responses
- Keep-Alive Messages
- Update Requests

Five Message Types

- **Announcements**
- Queries
- Responses
- Keep-Alive Messages
- Update Requests

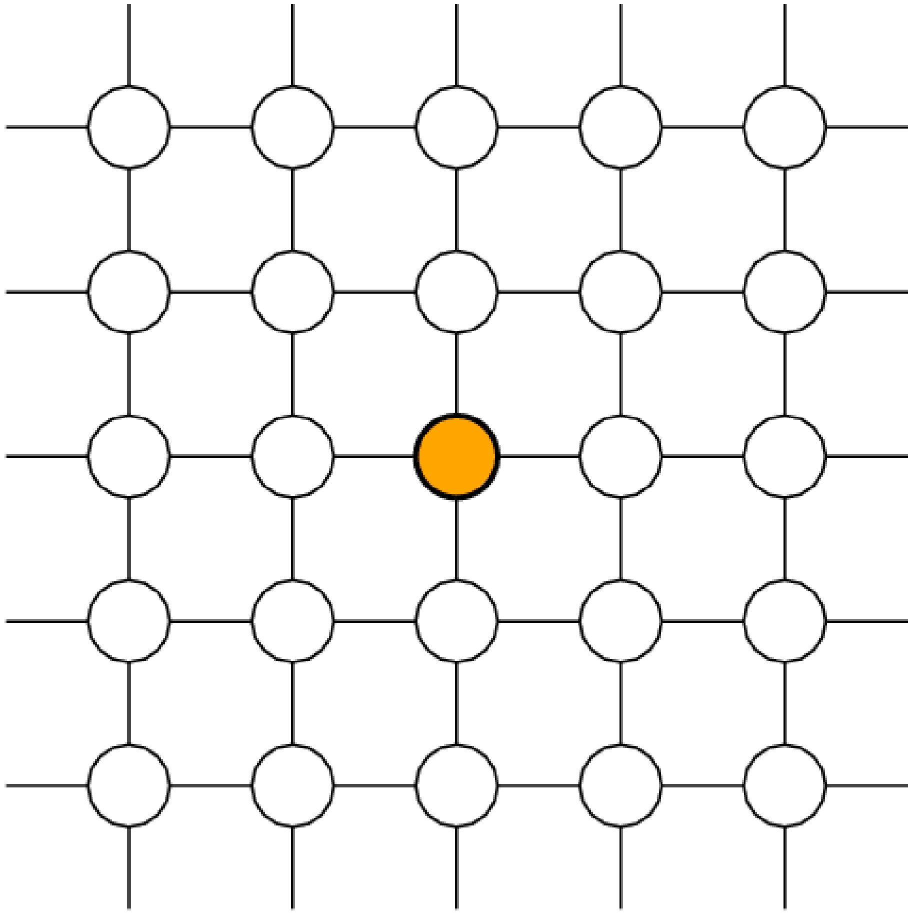
University of Twente
The Netherlands

Announcements

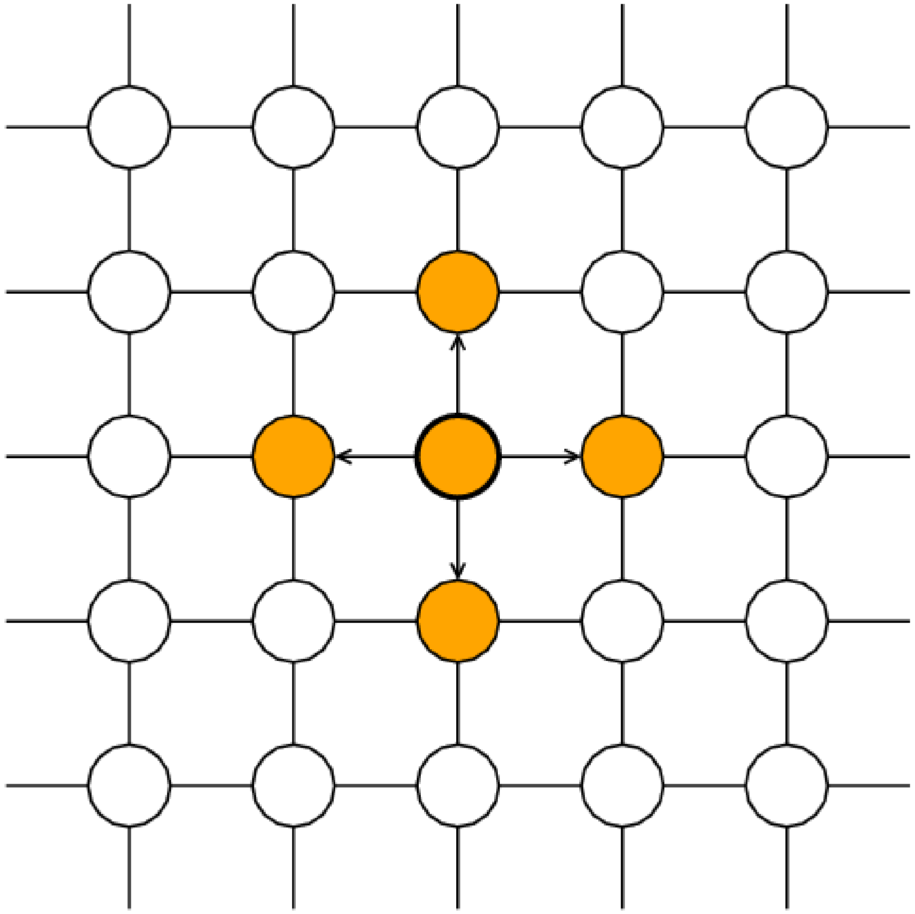
- Tell where to find service information
- Multiple layers
  - First layer: services of sending node
  - Second layer: services of sender's neighbors
  - And so on, up to configurable limit
- Sent to all direct neighbors

University of Twente
The Netherlands

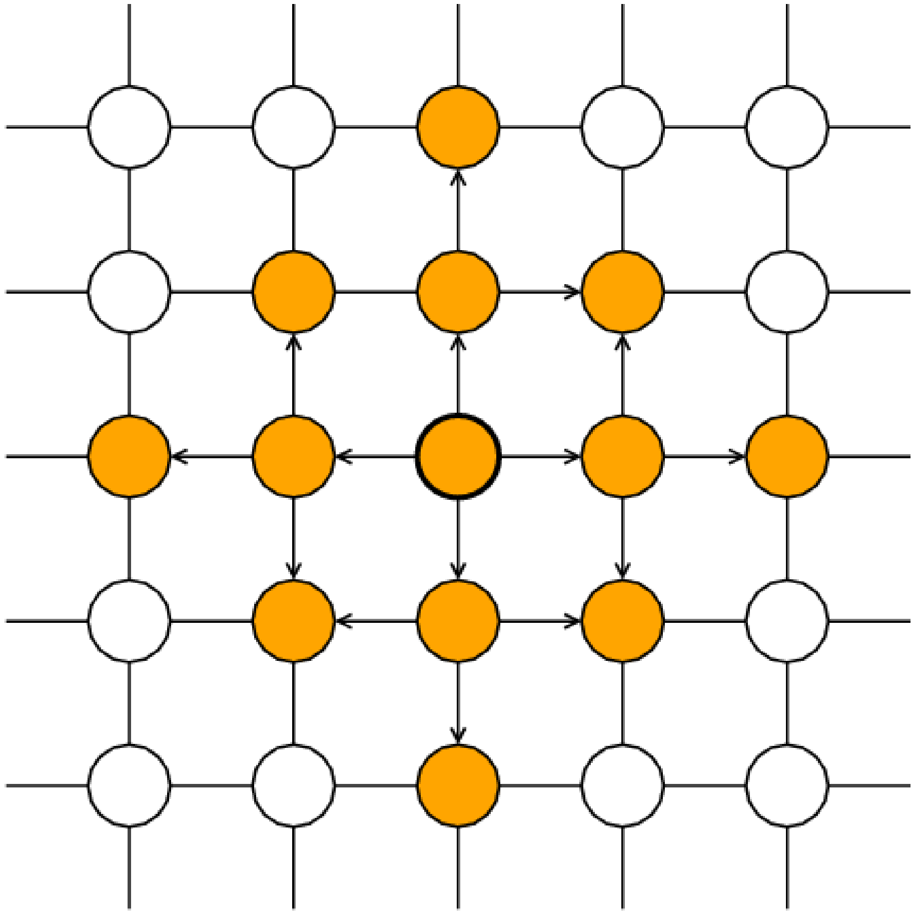
Distribution of Announcements



Distribution of Announcements



Distribution of Announcements



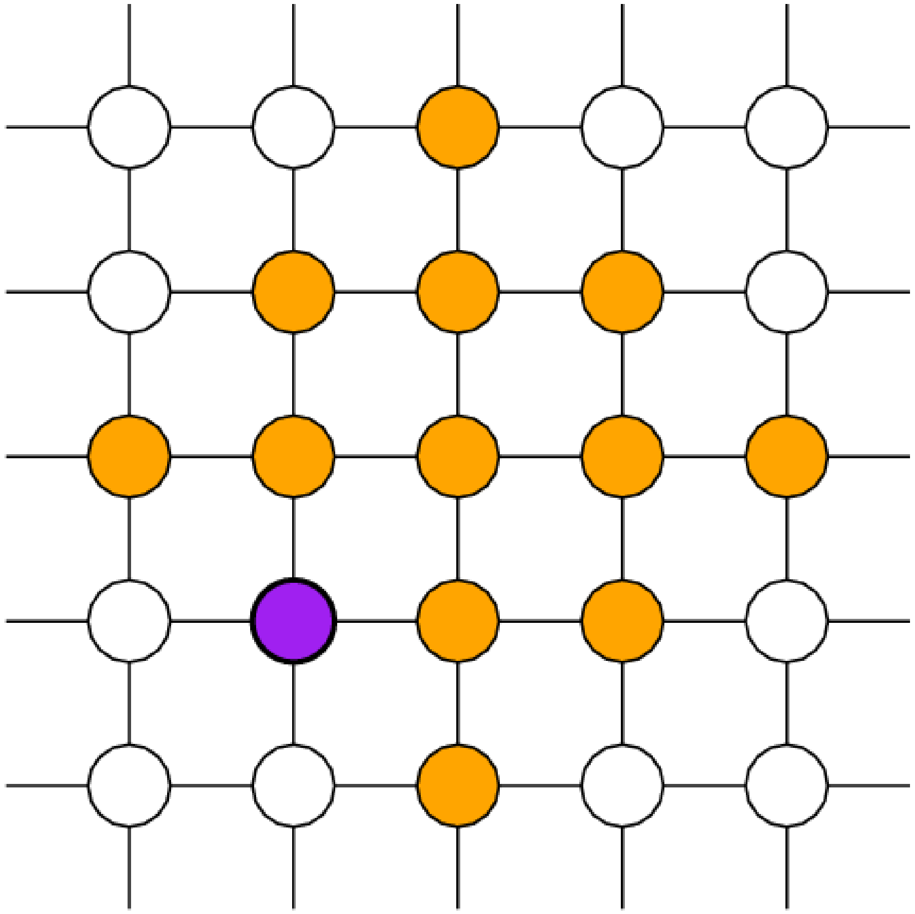
Five Message Types

- Announcements
- **Queries**
- Responses
- Keep-Alive Messages
- Update Requests

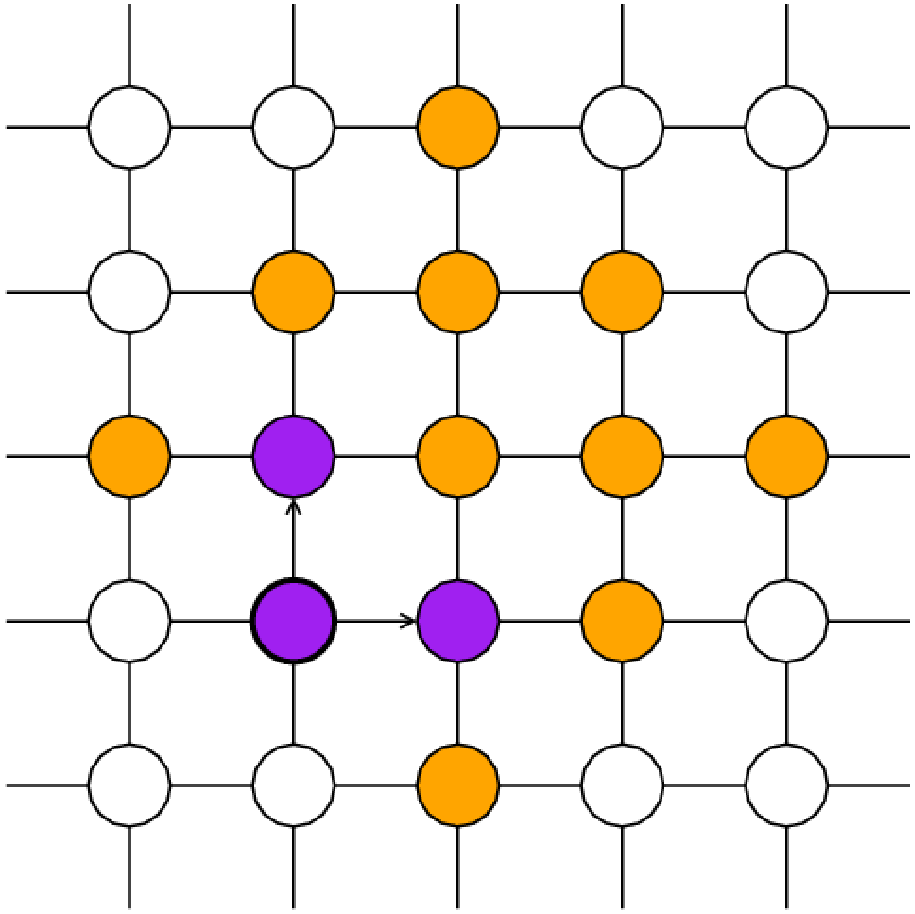
Queries

- Request service details
- Contain service name
- Are sent from neighbor to neighbor
- Only to neighbors who know about the service
- Only to neighbors who are close to the service

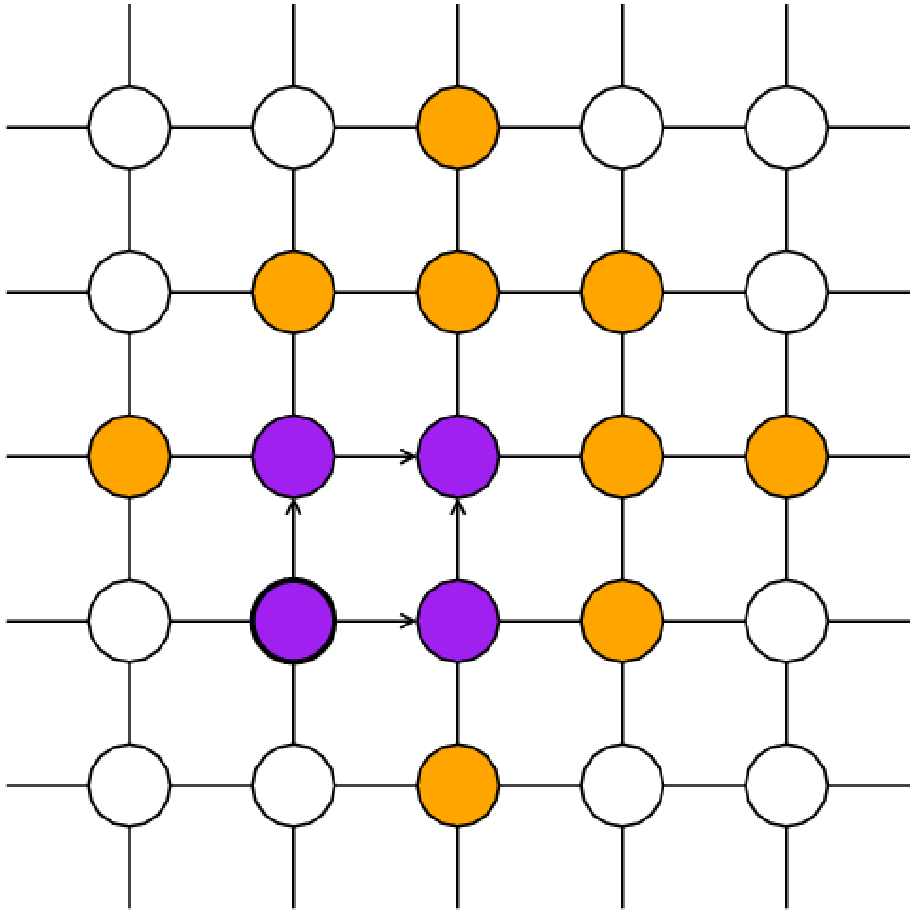
Distribution of Queries



Distribution of Queries



Distribution of Queries



Five Message Types

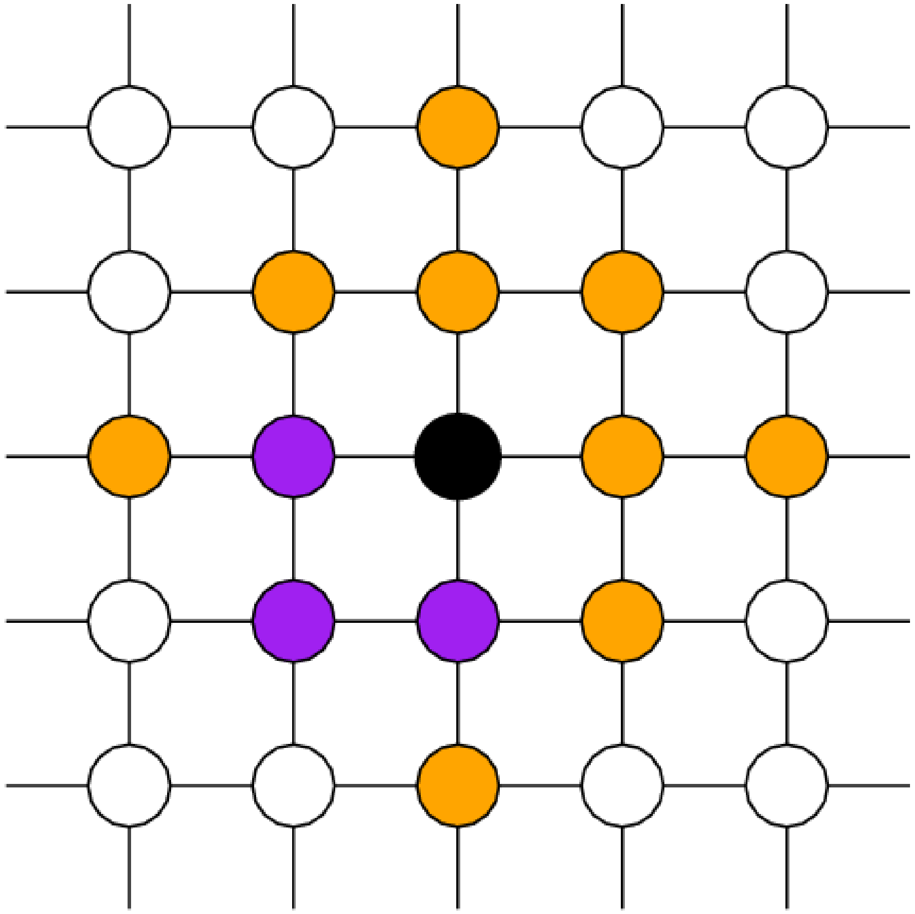
- Announcements
- Queries
- **Responses**
- Keep-Alive Messages
- Update Requests

Responses

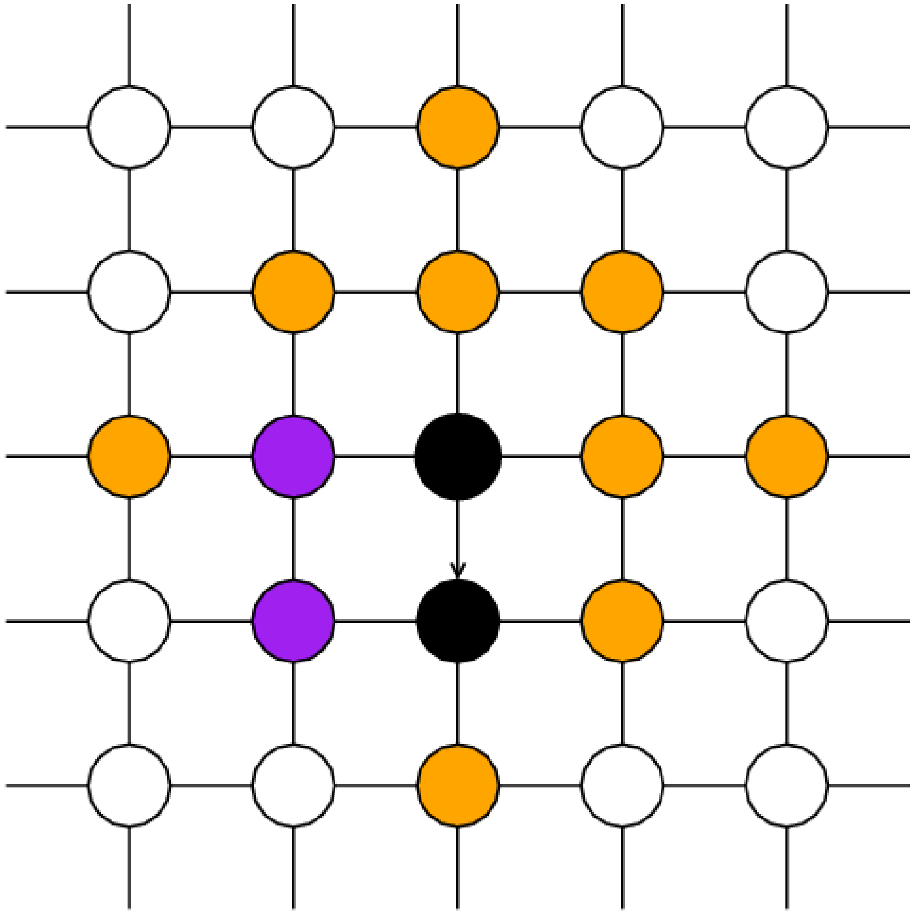
- Tell service details
  - Specifically: IP address and port number
- Sent from node offering service to node sending query

University of Twente
The Netherlands

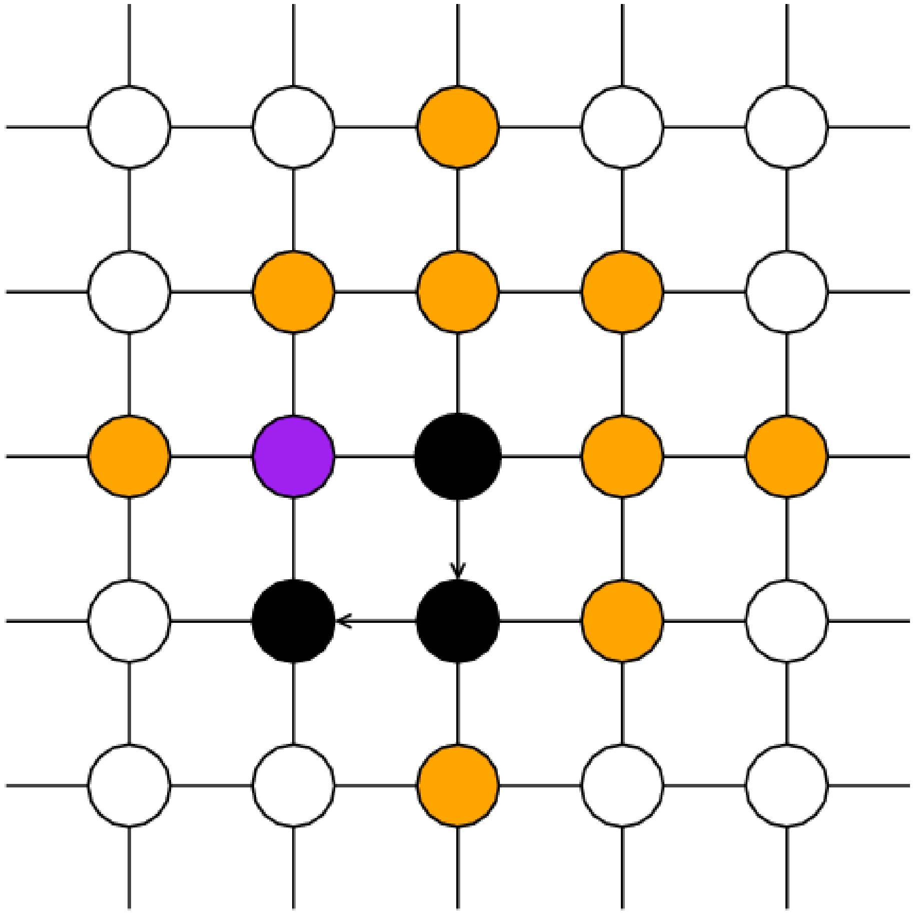
Response Distribution



Response Distribution



Response Distribution



Five Message Types

- Announcements
- Queries
- Responses
- **Keep-Alive Messages**
- Update Requests

Keep-Alive Messages

- “I'm still here!”
- Sent to direct neighbors
- Detect changes in topology
- Detect missed announcements
  - Through announcement id
- No keep-alive received for a while:
  - Information from neighbor is discarded

University of Twente
The Netherlands

Five Message Types

- Announcements
- Queries
- Responses
- Keep-Alive Messages
- **Update Requests**

Update Requests

- “Could you repeat that?”
- Sent when an announcement has been missed
- Causes announcement to be sent again

Part 4

Attenuated Bloom Filters

How To Tell Where Information Is

- Announcements do not contain service names
- But they do tell where information can be found
- How?
- Answer: Attenuated Bloom Filters

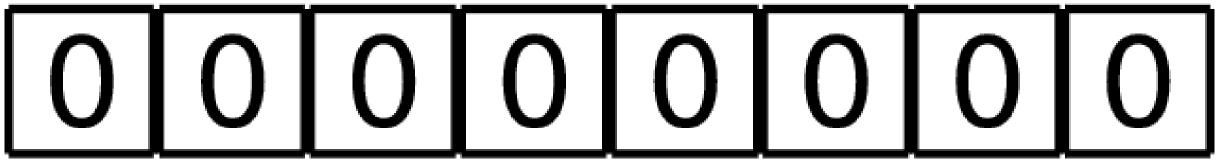
University of Twente
The Netherlands

Bloom Filters

- Two operations:
  - Adding an item
  - Testing if an item is present
- Compact representation
- Small chance of false positives

Bloom Filter Implementation

- Array of bits (initially all 0)



- Set of hash functions
- Each hash function maps a service name to a bit in the array

Bloom Filters: Adding Items

- Apply each hash function to item
- Set corresponding bits to 1

Adding Service “printer”

- Two hash functions
  - First returns 1
  - Second returns 5
- Resulting Bloom filter:

0	1	0	0	0	1	0	0

Bloom Filters: Item Present?

- Apply each hash function to item
- Test if corresponding bits are 1
- If not all are 1, item is absent
- If all are 1, item is probably present
- Bits might be 1 because of other items
- This is called a false positive

Bloom Filters: Item Present

- Bloom filter:

0	1	0	0	0	1	0	0

- Service name “printer”
- Two hash functions
  – First returns 1
  – Second returns 5

Bloom Filters: Item Not Present

- Bloom filter:

0	1	0	0	0	1	0	0

- Service name “thermometer”
- Two hash functions
  - First returns 2
  - Second returns 7

Bloom Filters: False Positive

- Bloom filter:

0	1	0	0	0	1	0	0

- Service name “mail server”
- Two hash functions
  - First returns 5
  - Second returns 1

University of Twente
The Netherlands

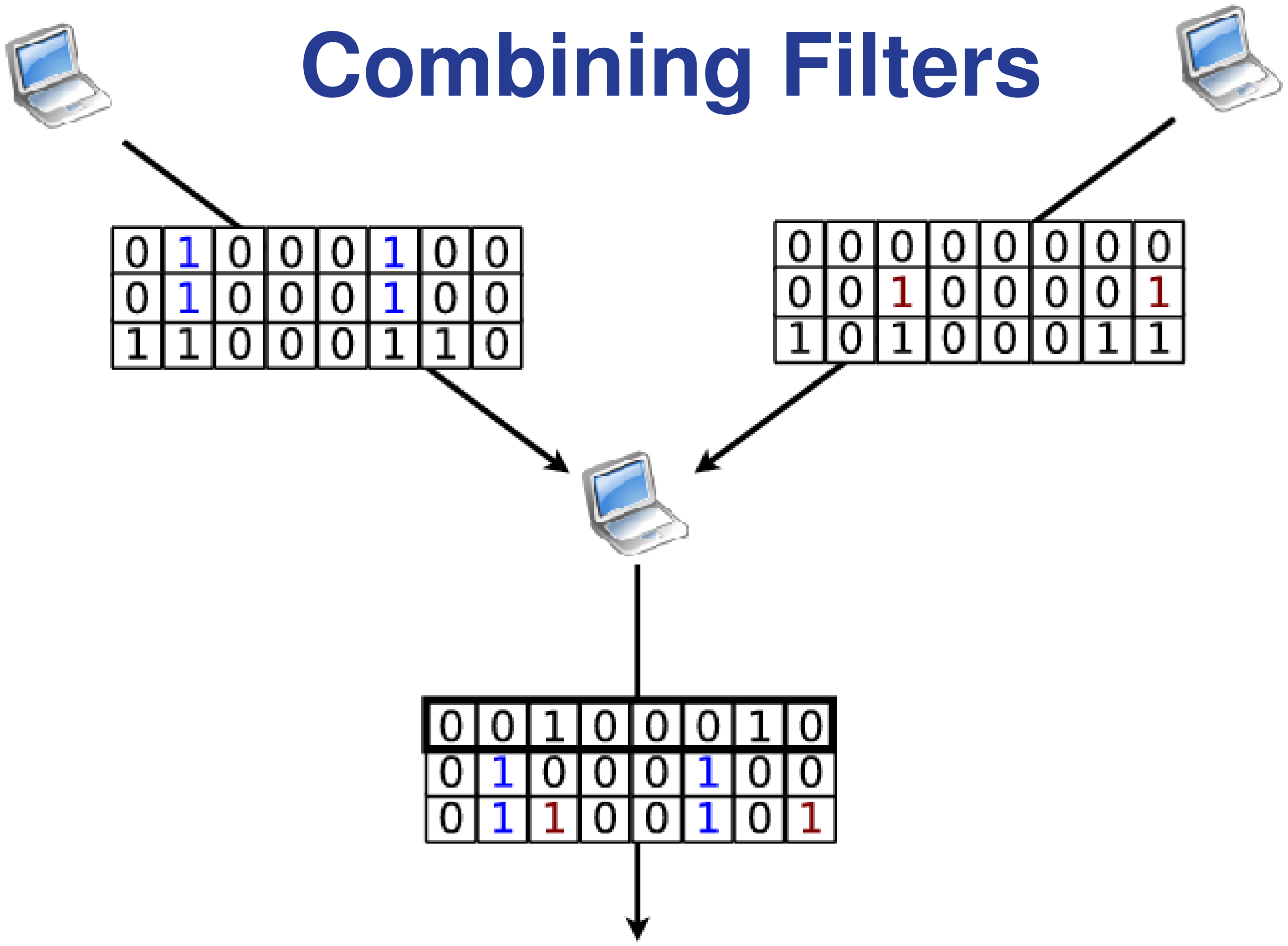
False Positives

- False positives are bad
- They cause queries to be sent
- But no information will be found
- Thus, resources are wasted

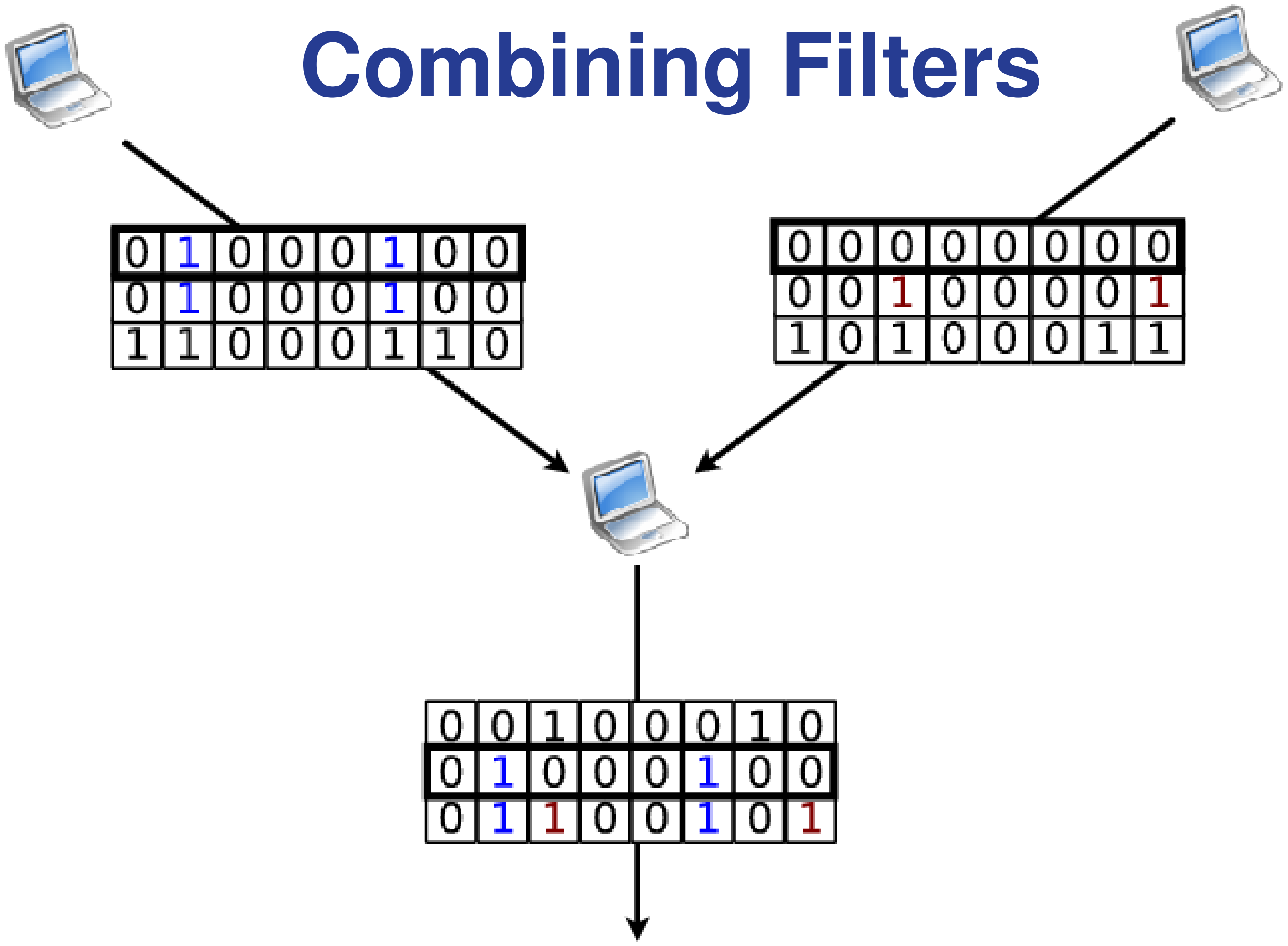
Attenuated Bloom Filters

- Multiple layers of Bloom filters
- One layer for services of the sender
- One layer for services of its neighbors
- One layer for services of their neighbors
- Etc.

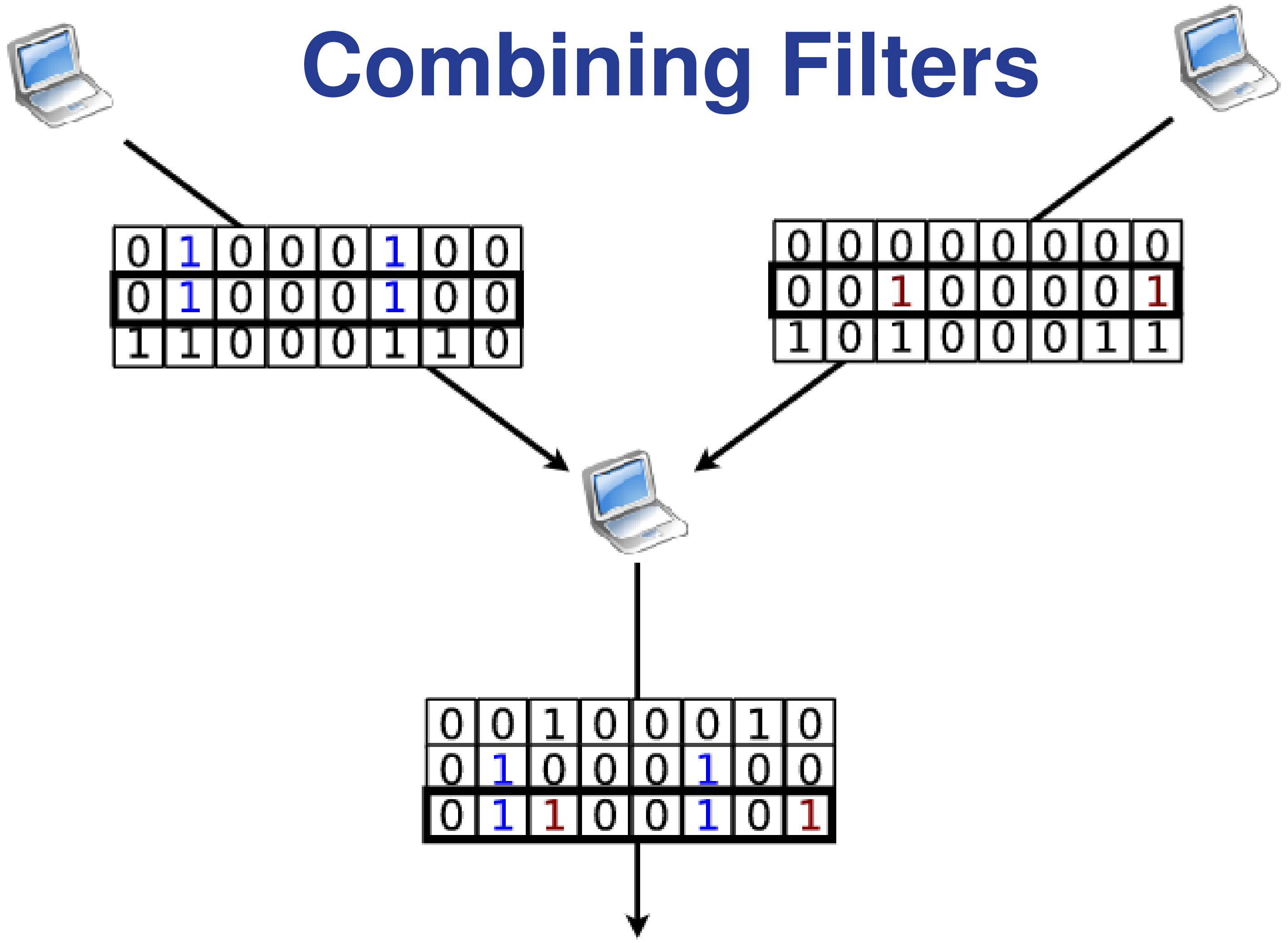
Combining Filters



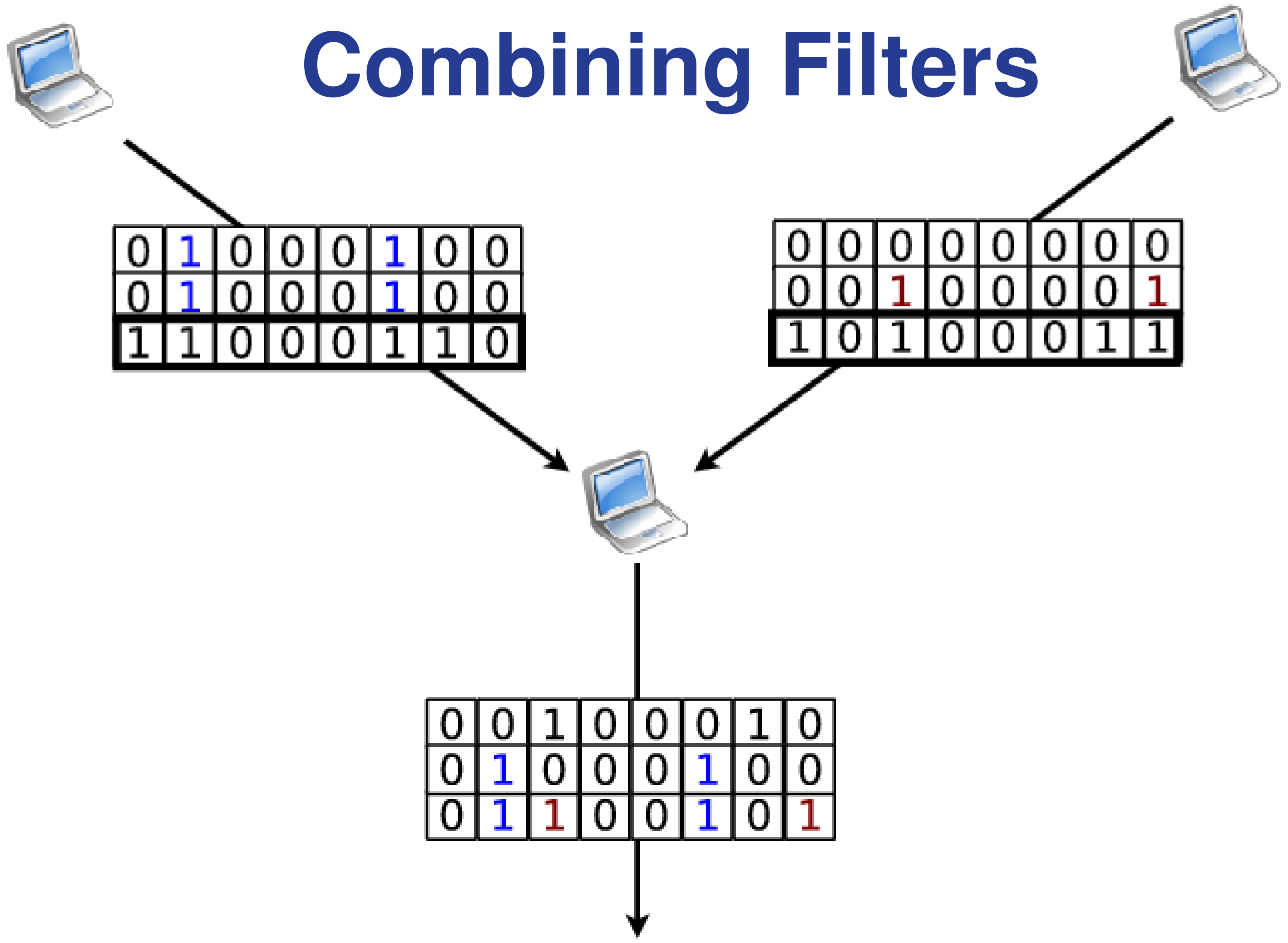
Combining Filters



Combining Filters



Combining Filters



Attenuated Bloom Filters

- Are small
- Do not contain service names
- But do tell which neighbor is likely to know about a service

Part 5

My Contributions

My Contributions

- Original idea from Geert, Fei, and Patrick
- I implemented a prototype
- I decided protocol details
- I contributed some protocol enhancements
- I came up with the name

University of Twente
The Netherlands

The Prototype

- Shows that Ahoy works
- Forced details to be decided
- Serves as a platform for further experimentation

University of Twente
The Netherlands

Protocol Details

- What message types exist?
- What exactly do we put in them?
- When do we send messages?
- What technology do we build on top of?

Protocol Enhancements

- Original protocol sent announcements periodically
- I added keep-alive messages and update requests
- This saves resources:
  - Keep-alive messages are 5 bytes
  - Announcements can be 100s or 1000s

The Name

- Ahoy is a pun on “Bonjour”, the service discovery mechanism used by Apple
- It also alludes to my fondness of sailing

Part 6

Summary

University of Twente
The Netherlands

What is Ahoy?

- Ahoy is an efficient, decentralized service discovery protocol for mobile ad-hoc networks

What Have I Contributed?

- Prototype
- Protocol Details
- Protocol Enhancements
- The Name

What Have We Gained?

- We know Ahoy works
- We have an implementation
- Design alternatives have been documented
- Other alternatives can be tried

Part 7

Questions

University of Twente
The Netherlands

Thanks

- Geert, Patrick, Fei, and Hartmut for their ideas and feedback
- Everyone for attending