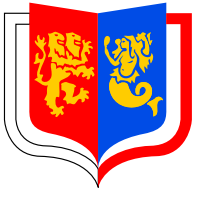
Rendezvous for Bluetooth devices

Marie Duflot



THE UNIVERSITY
OF BIRMINGHAM

Outline

- Introduction
- Presentation of the lower layer
- How to connect Bluetooth devices
- Verification

Introduction

What is Bluetooth ?

- Protocol for short range wireless communication
- Voice and Data
- Open specification
- Special Interest Group (Ericsson, IBM, Intel, Microsoft, Motorola, Nokia, Toshiba,...)

Bluetooth *vs* Wi-Fi, what is different ?

Wi-Fi

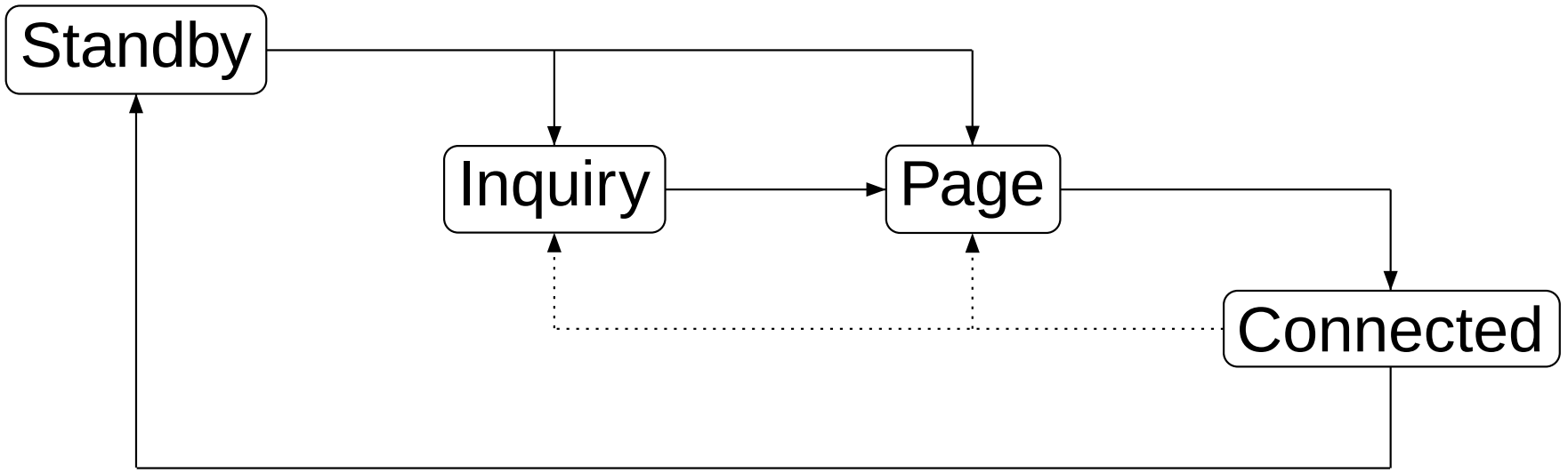
- Replacement for ethernet
- More powerful: 100/400m, 11Mbps
- Higher power consumption

Bluetooth

- Peer to peer communication
- Shorter range and rate: 10m, 1Mbps
- Lower power consumption

Brief Overview

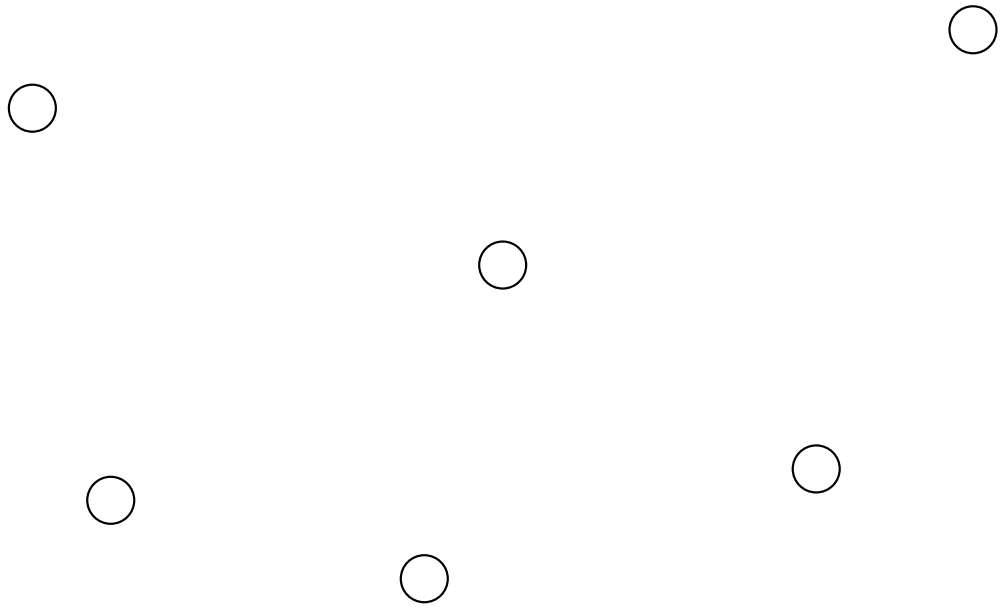
States of a Bluetooth device



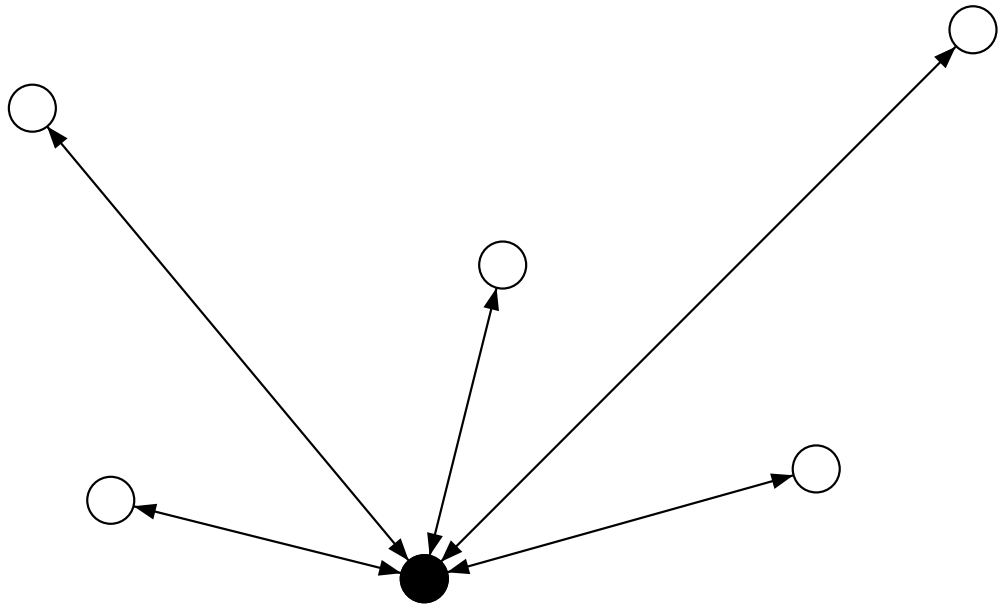
Standby: default operational state

Connected: device ready to communicate in a piconet

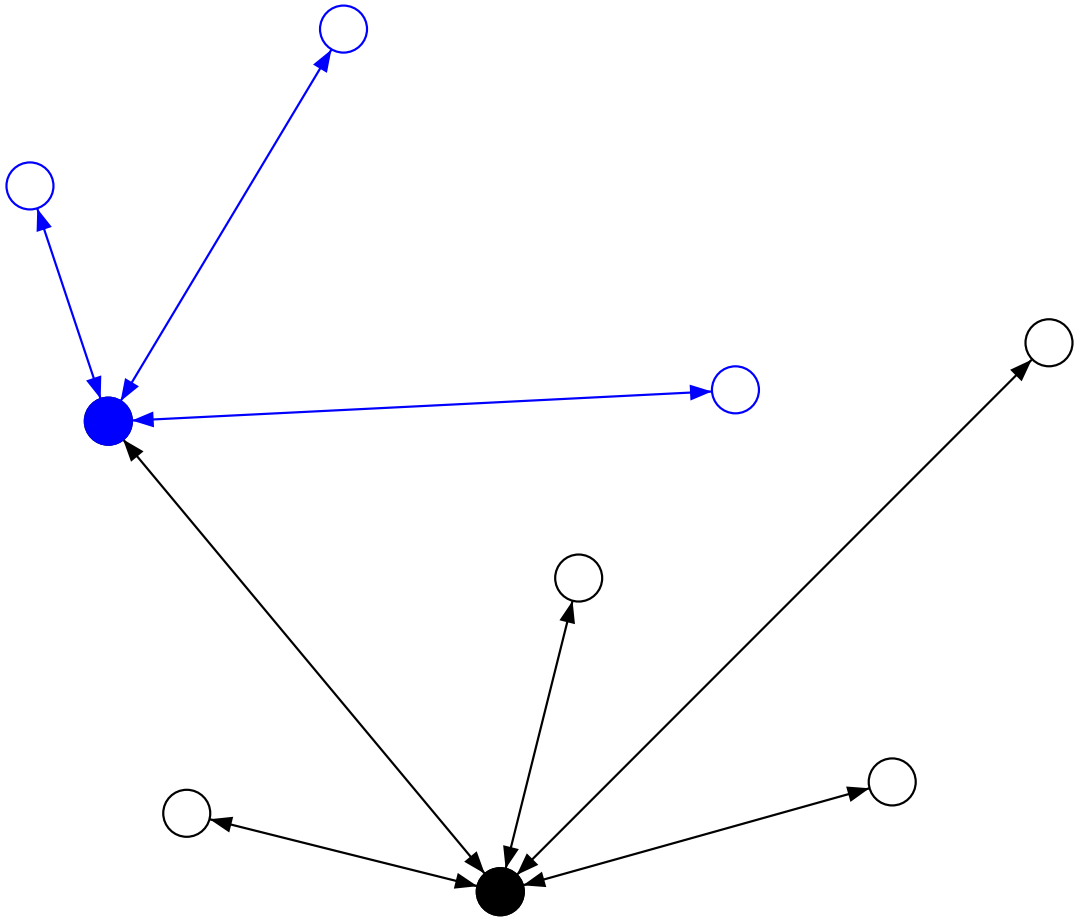
Piconets



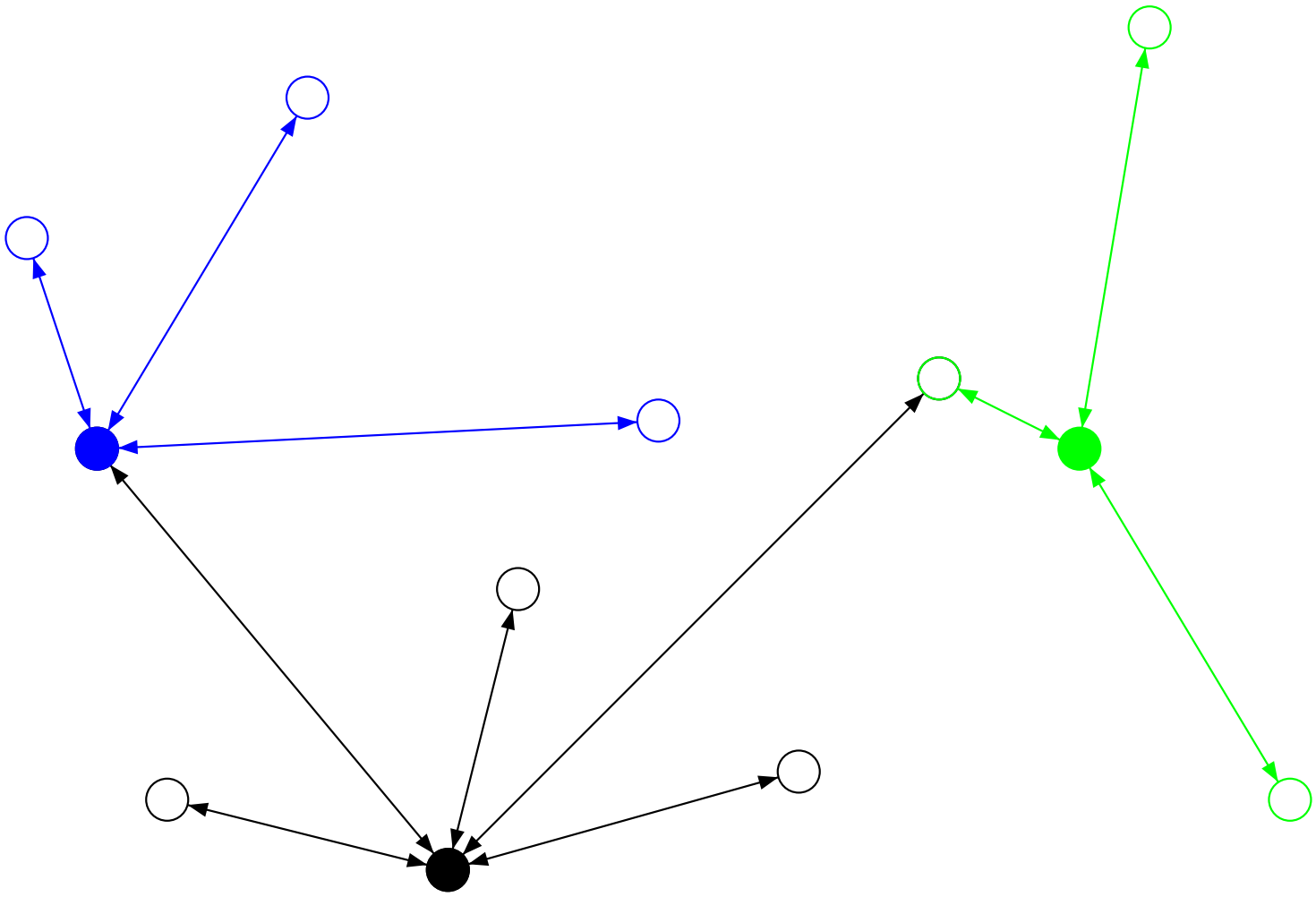
Piconets



Piconets and Scatternets



Piconets and Scatternets



Frequencies and Hopping

- Industrial Scientific Medical band
  - ➤ free
  - ➤ interference (microwaves, etc.)
- Solution : hop through sequence of frequencies
  - ➤ "common"
    to discover/be discovered by other devices
  - ➤ "unpredictable"
    to communicate in a piconet

Access Code

- Devices need to know if a message is intended for them
- An Access Code is prefixed to each message
  - Inquiry: general or dedicated,
  - Page: based on the address of the receiving device,
  - Connected: based on the address of the master.
- When receiving a message, a device “reads” it only if it has a correct Access Code

Clocks

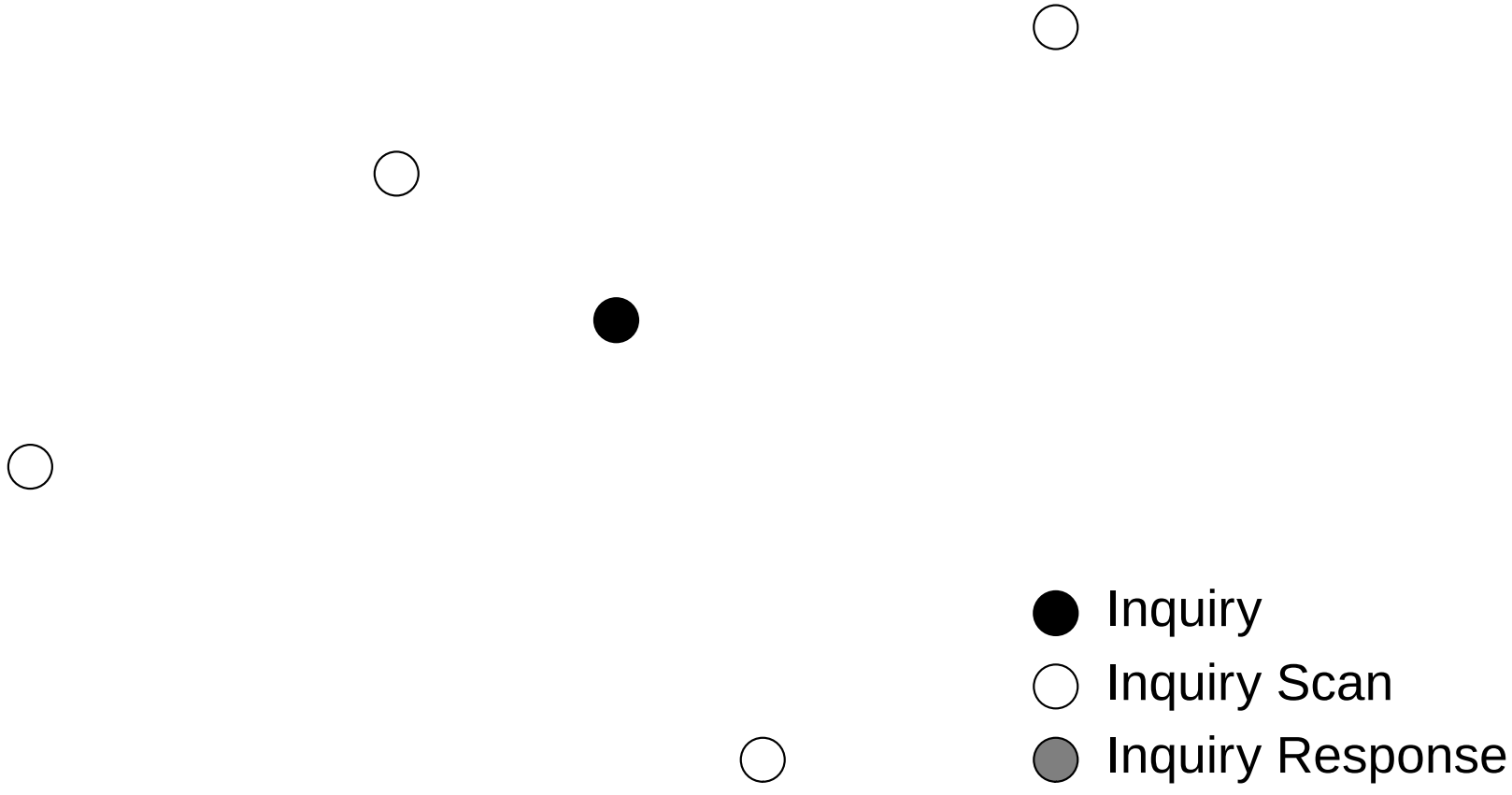
- Independent free running clocks
  - ➤ synchronisation via an offset
- Rates are not exactly the same
  - ➤ need to readjust their estimations
- The role of the clock is to determine
  - ➤ when a device can/cannot transmit/receive a message
  - ➤ at which frequency

The rendezvous layer

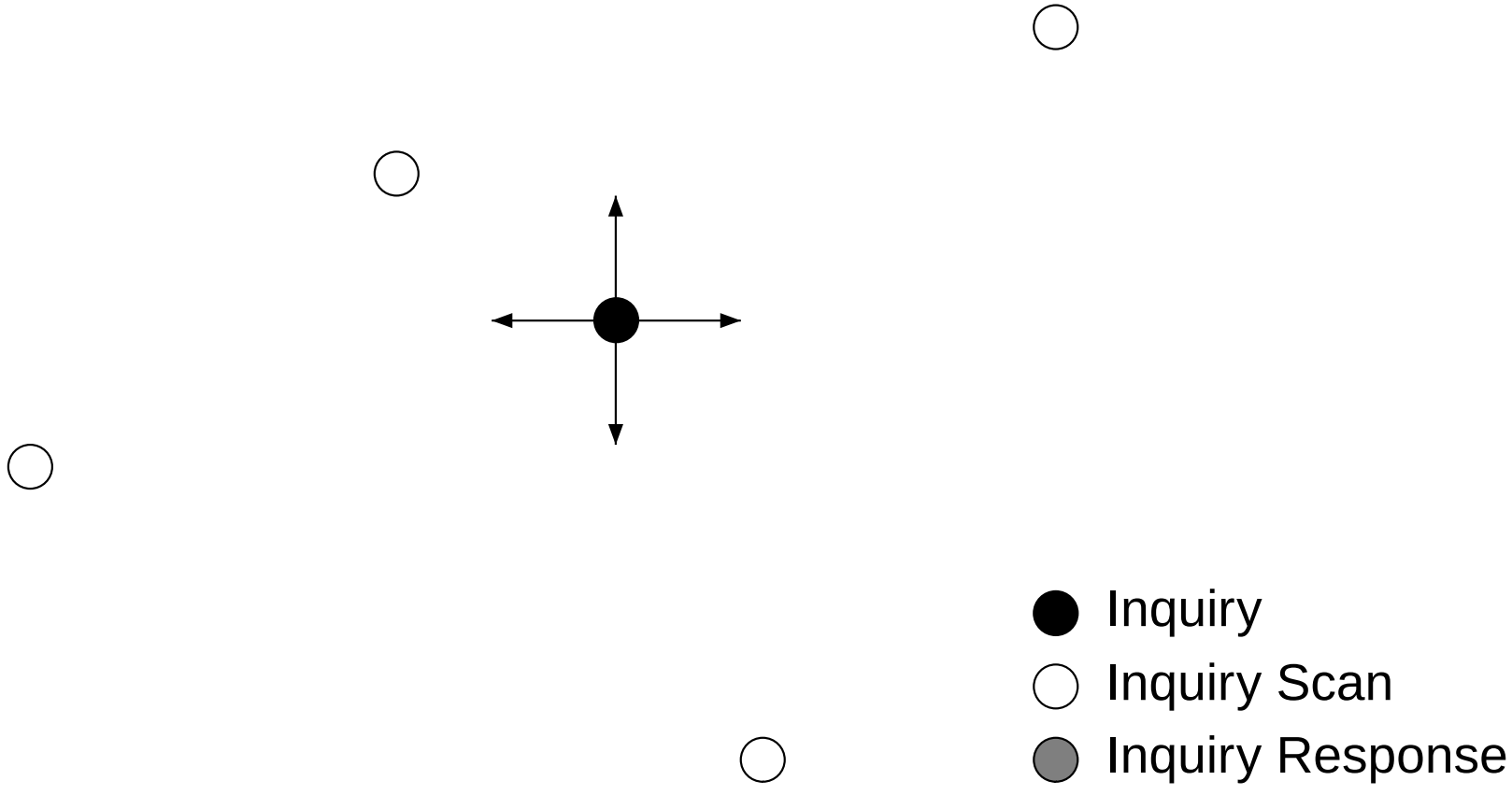
Inquiry

- Inquiry frequency hopping sequence : common to all devices
  - ➤ Inquiring device : hops every $312.5\mu s$
  - ➤ Inquired device : hops every $1.28s$
- Inquiry Access Code :
  - ➤ either general (look for all devices)
  - ➤ or dedicated (look for a particular type of device)
- Inquiry response packet : contains address & clock of the inquired device
- Randomized aspect to avoid collisions

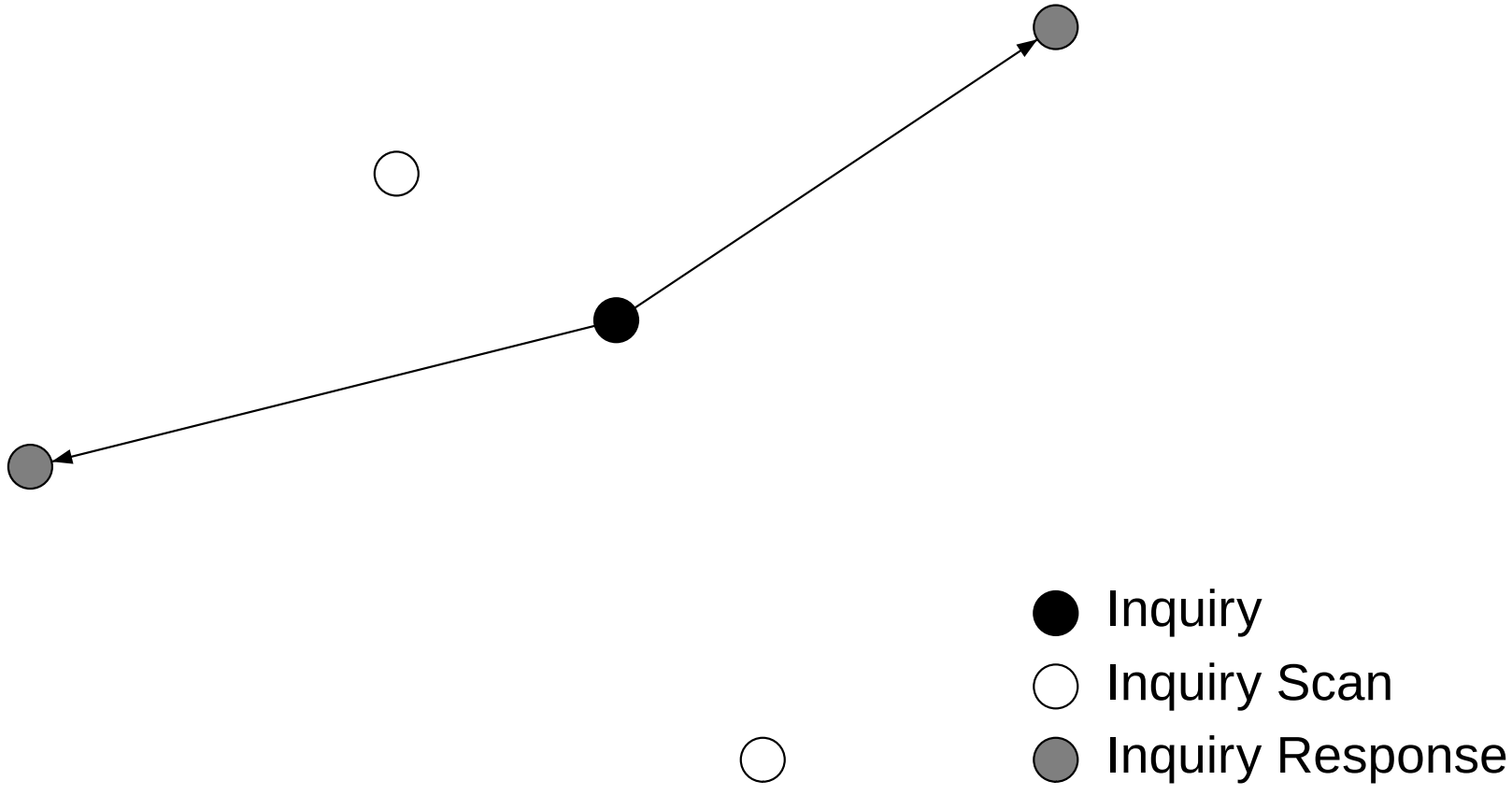
Inquiry



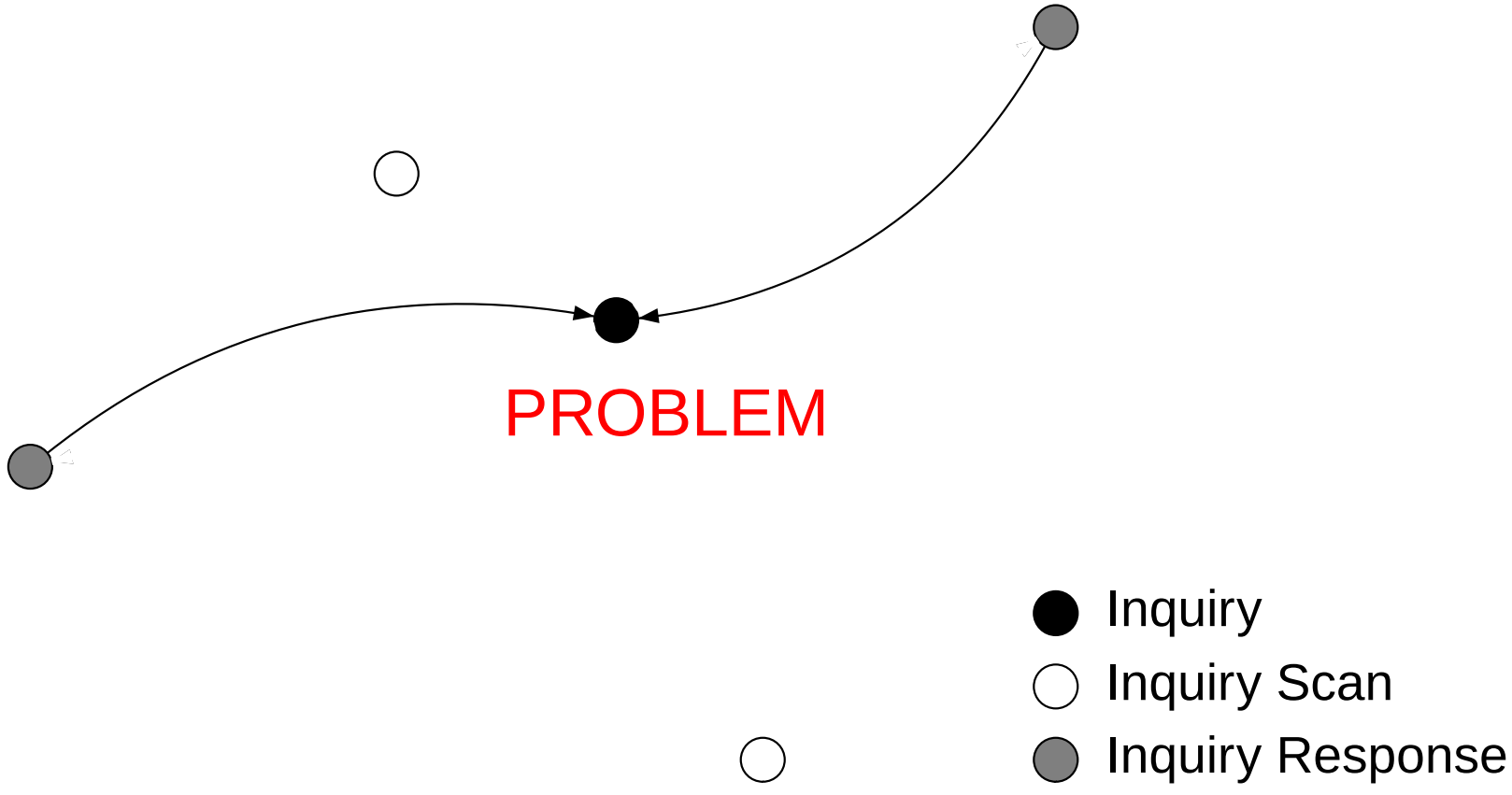
Inquiry



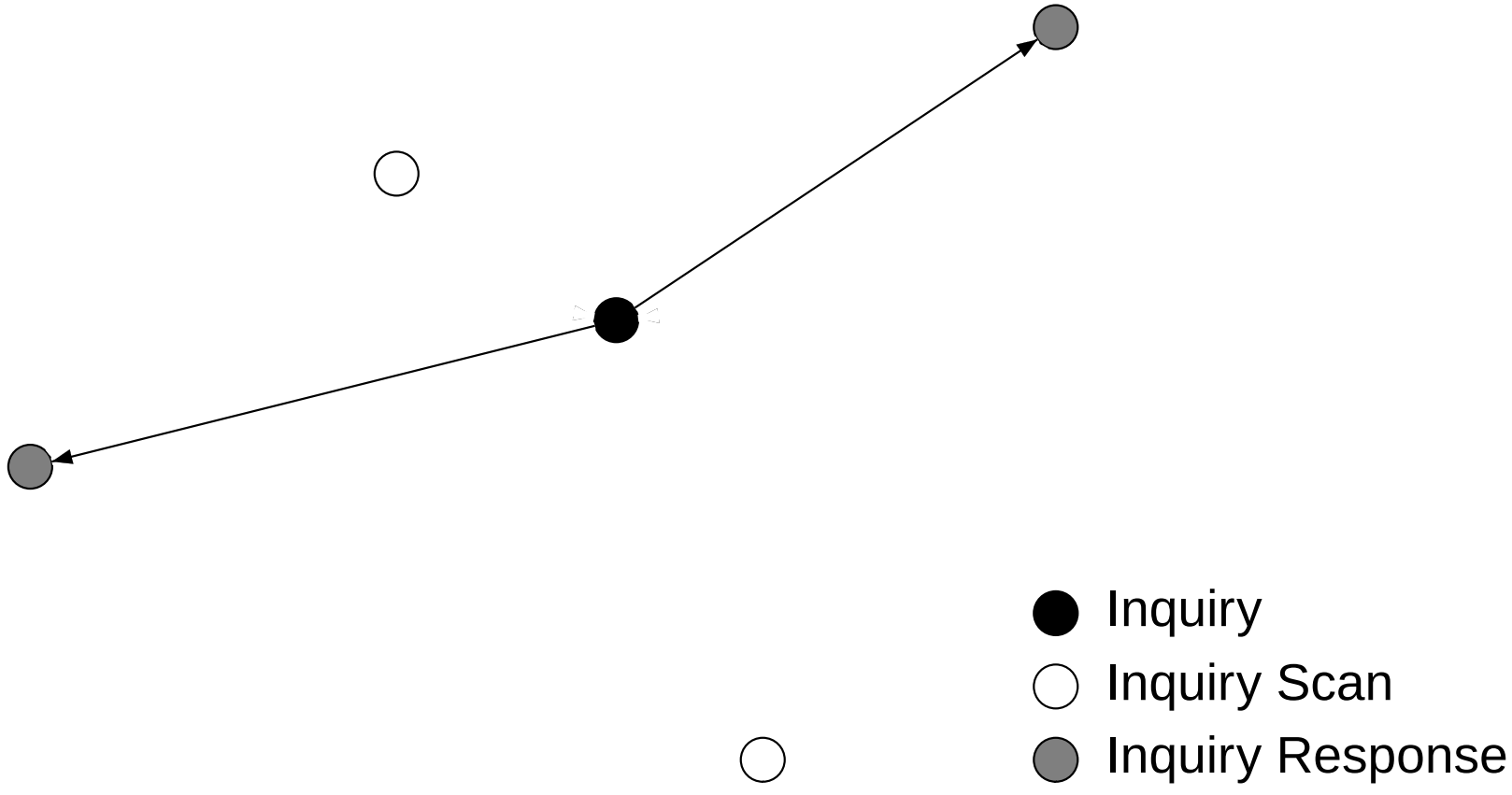
Inquiry



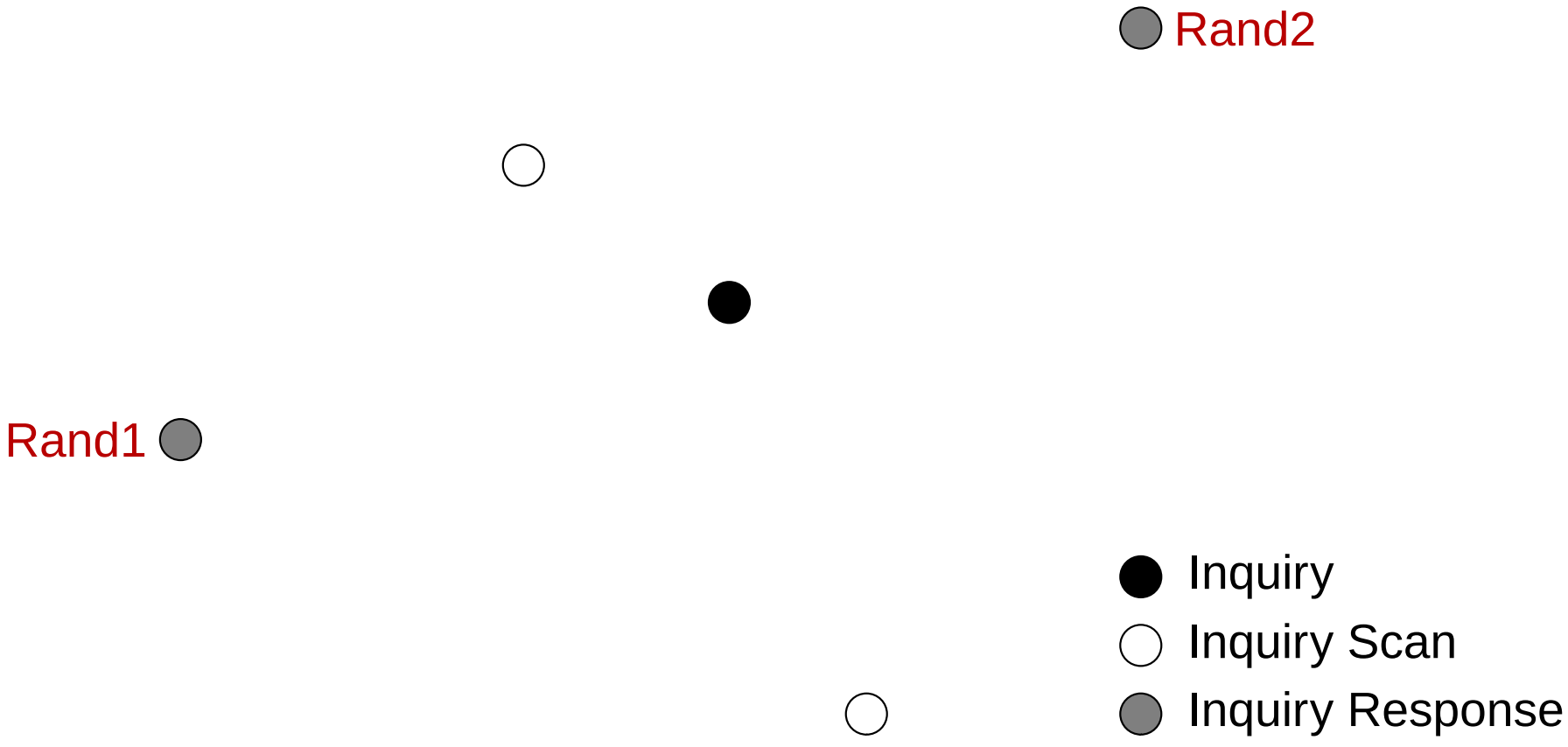
Inquiry



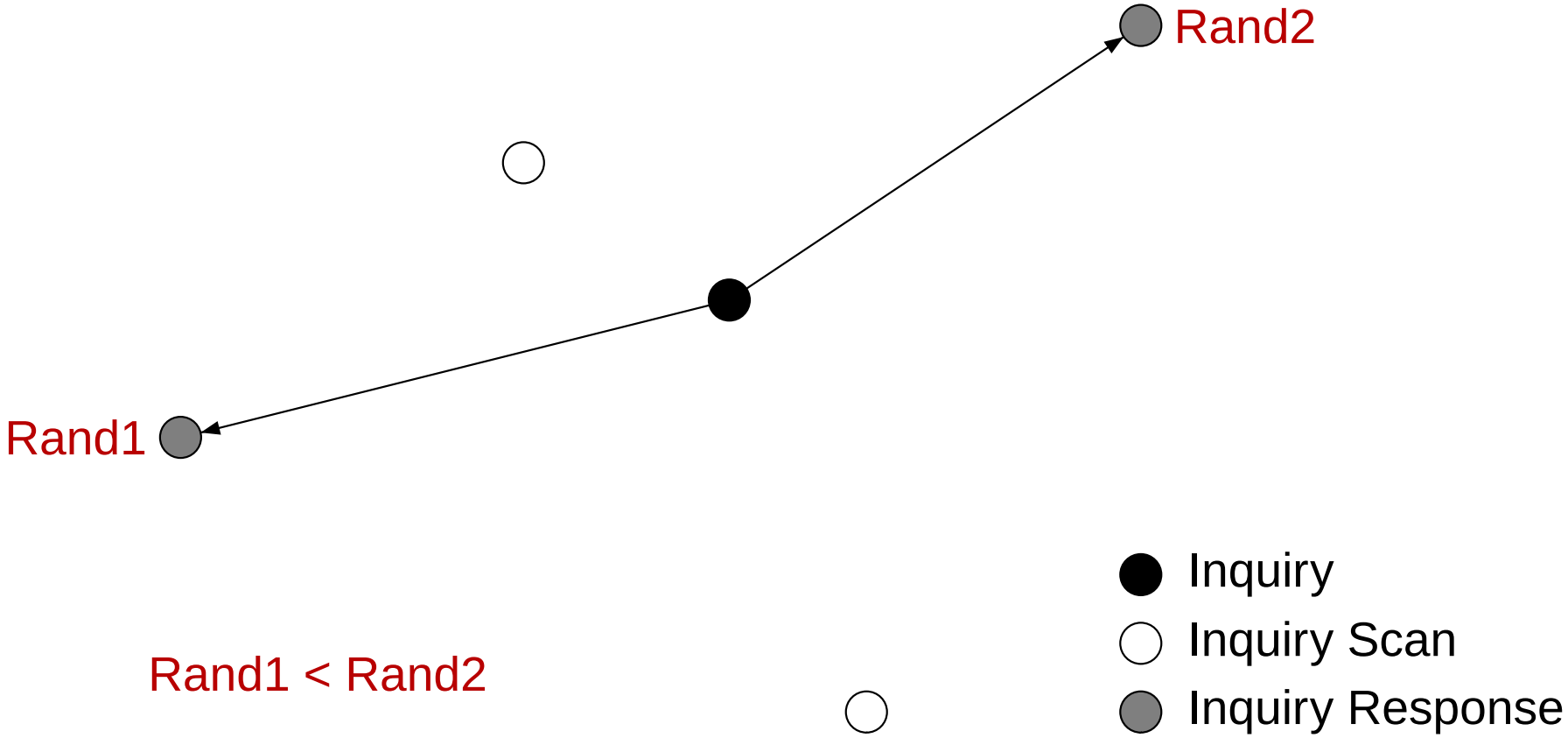
Inquiry



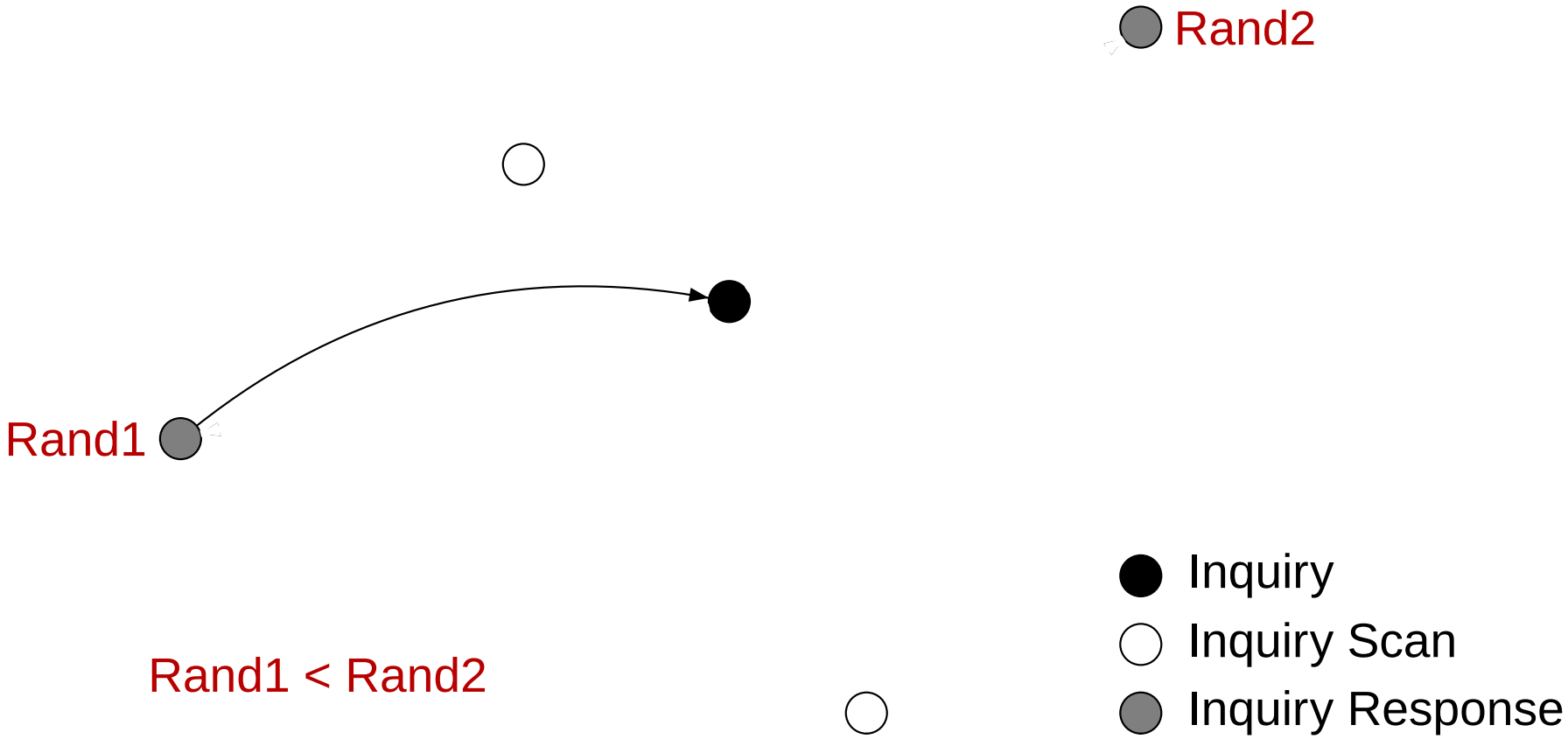
Inquiry



Inquiry


Rand2
Rand1
Rand1 < Rand2
Inquiry
Inquiry Scan
Inquiry Response

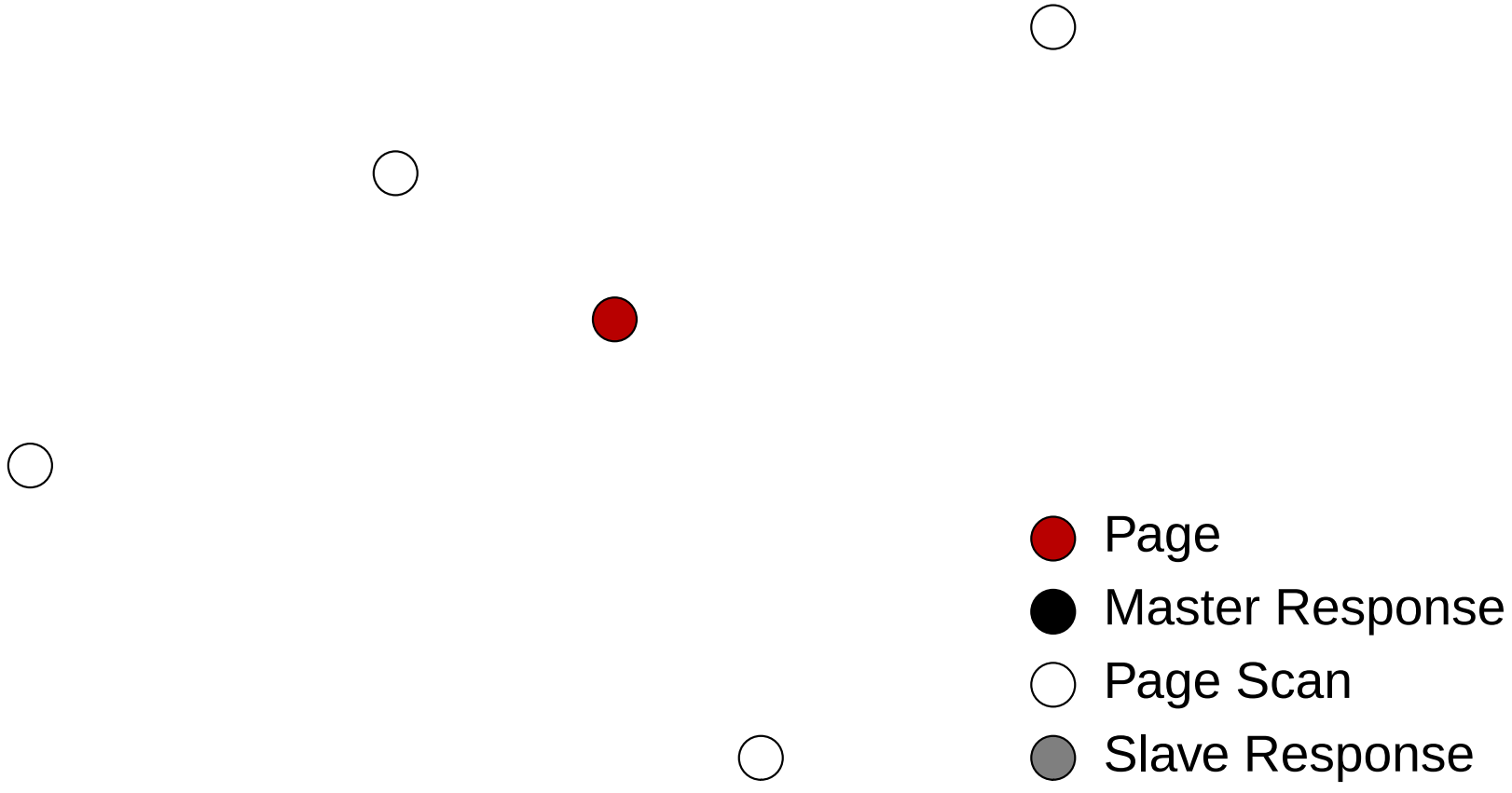
Inquiry


Rand2
Rand1
Rand1 < Rand2
Inquiry
Inquiry Scan
Inquiry Response

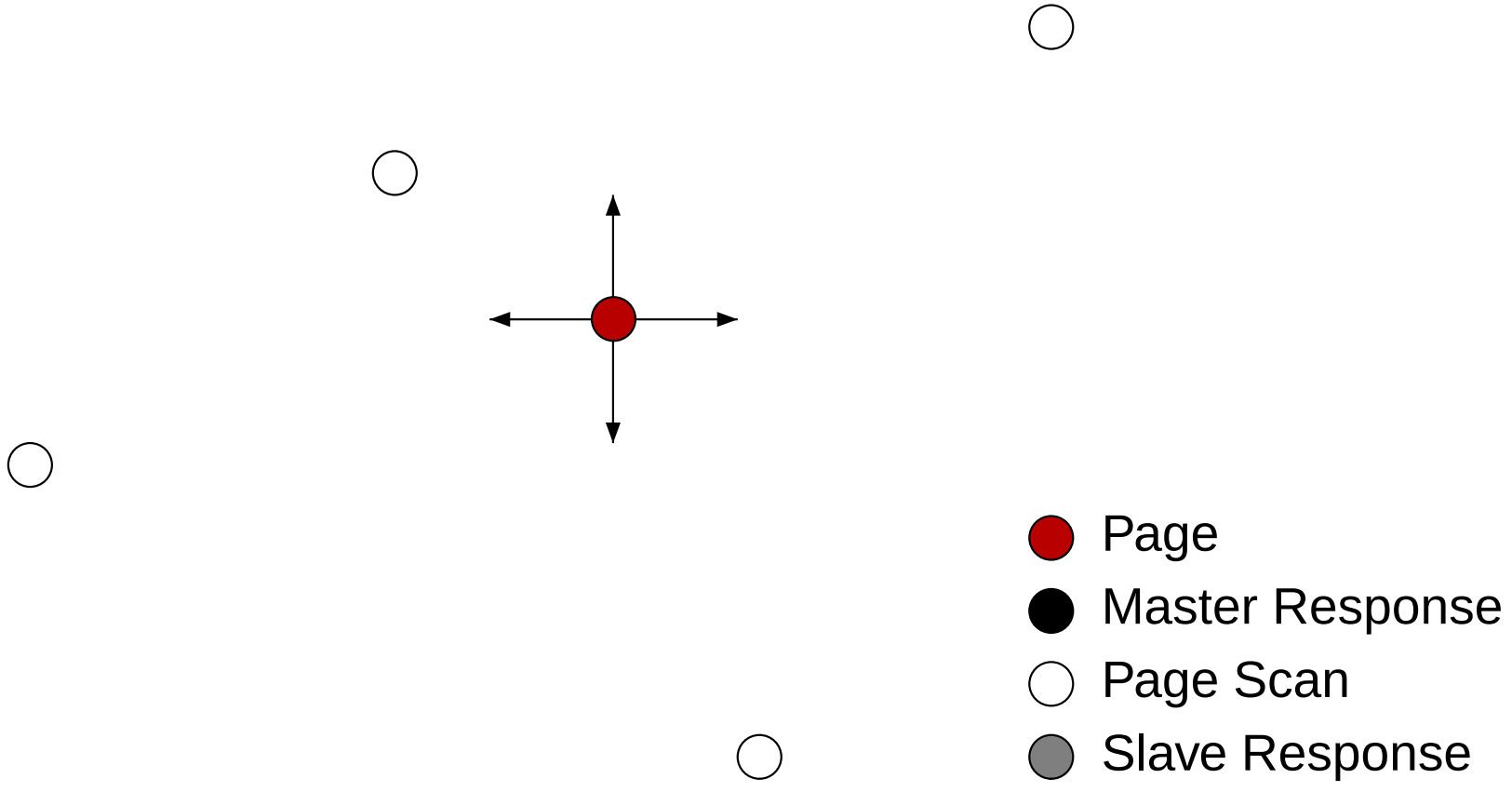
Page

- Page frequency hopping sequence : common to all devices
  - ➤ Paging device : hops every $312.5\mu s$
  - ➤ Paged device : hops every $1.28s$
- Access Code : uses the address of the paged device
- Master response packet : contains address & clock of the master

Page



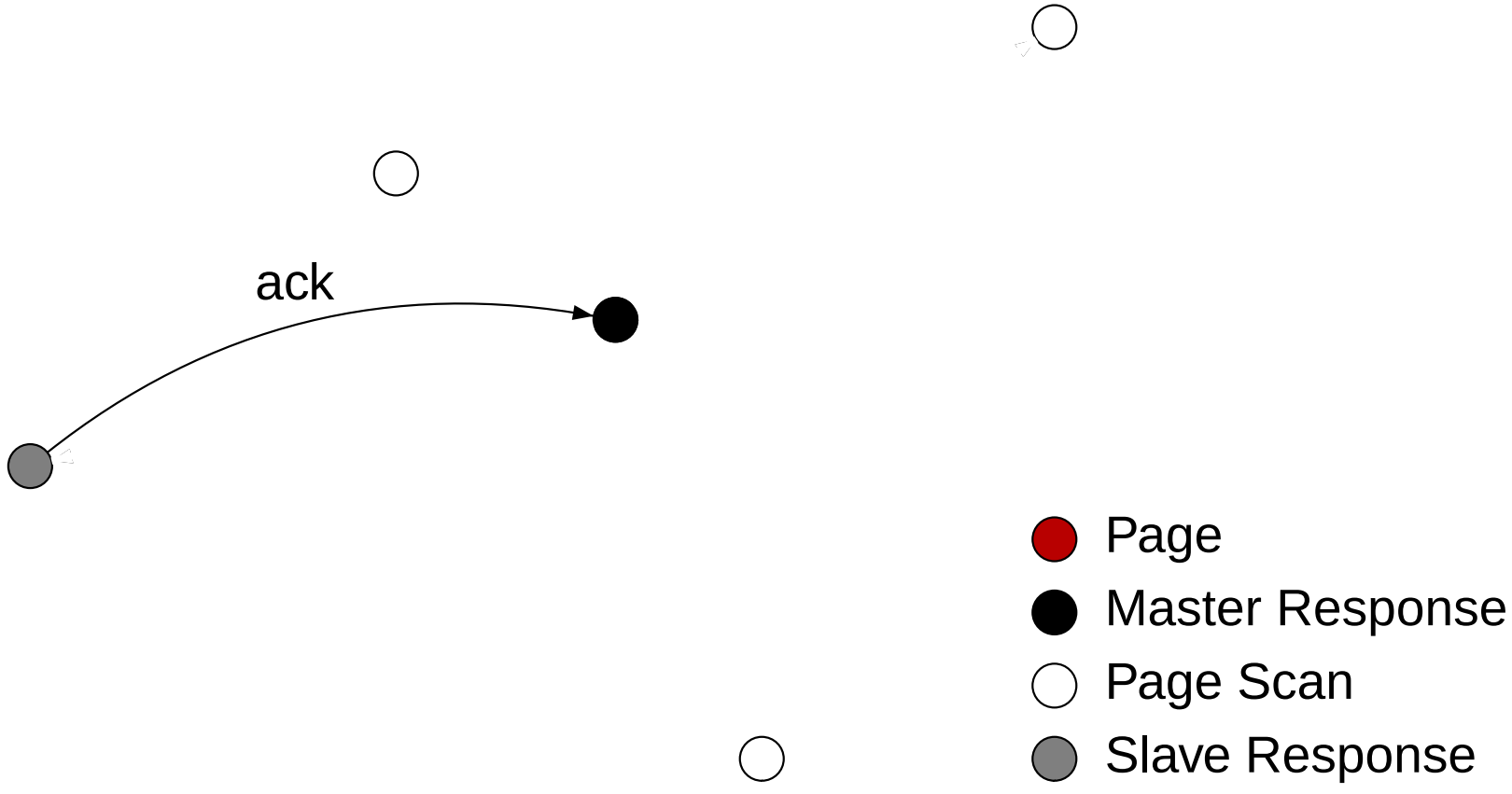
Page



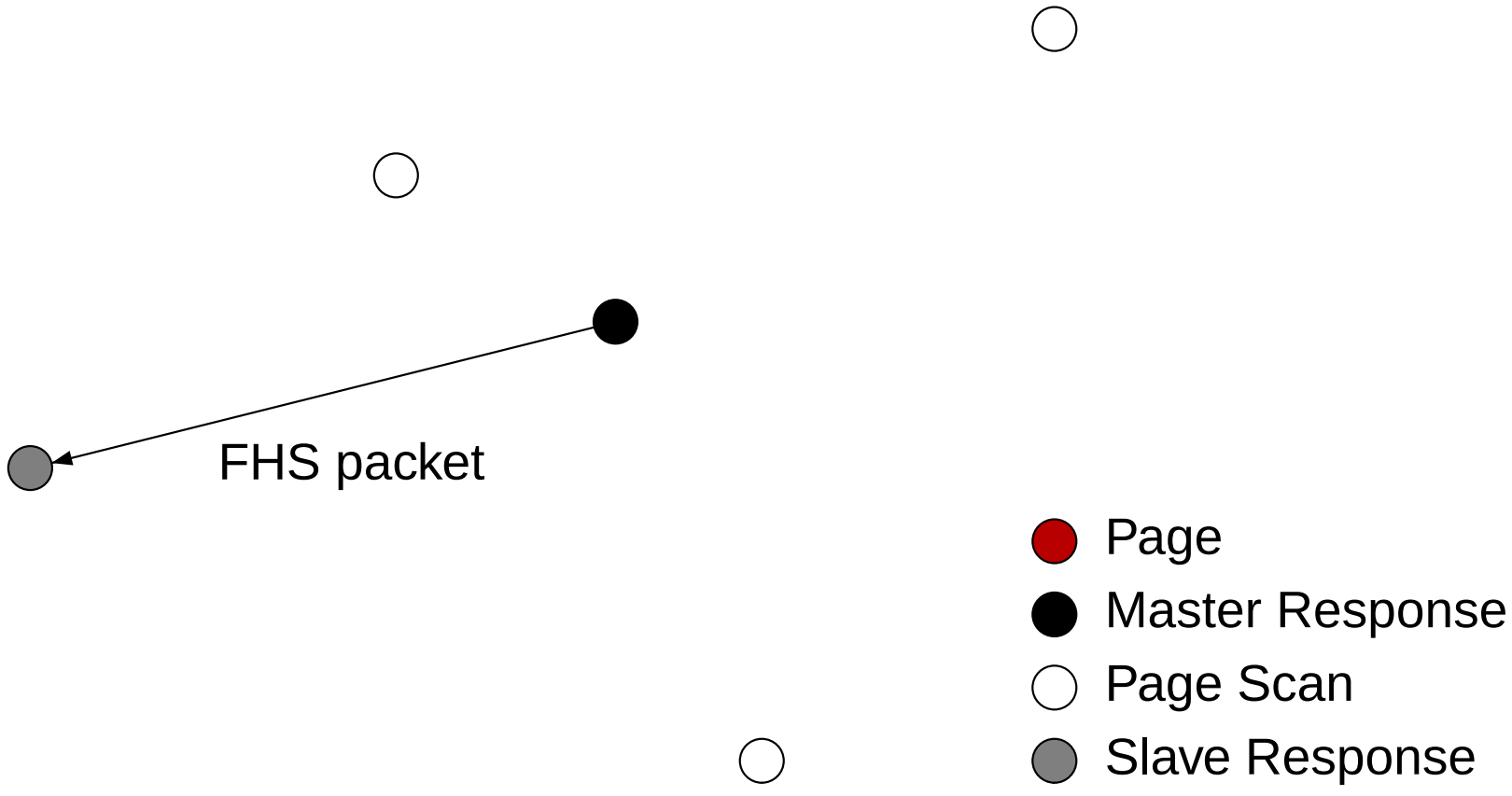
Page


slave ID
slave ID
Page
Master Response
Page Scan
Slave Response

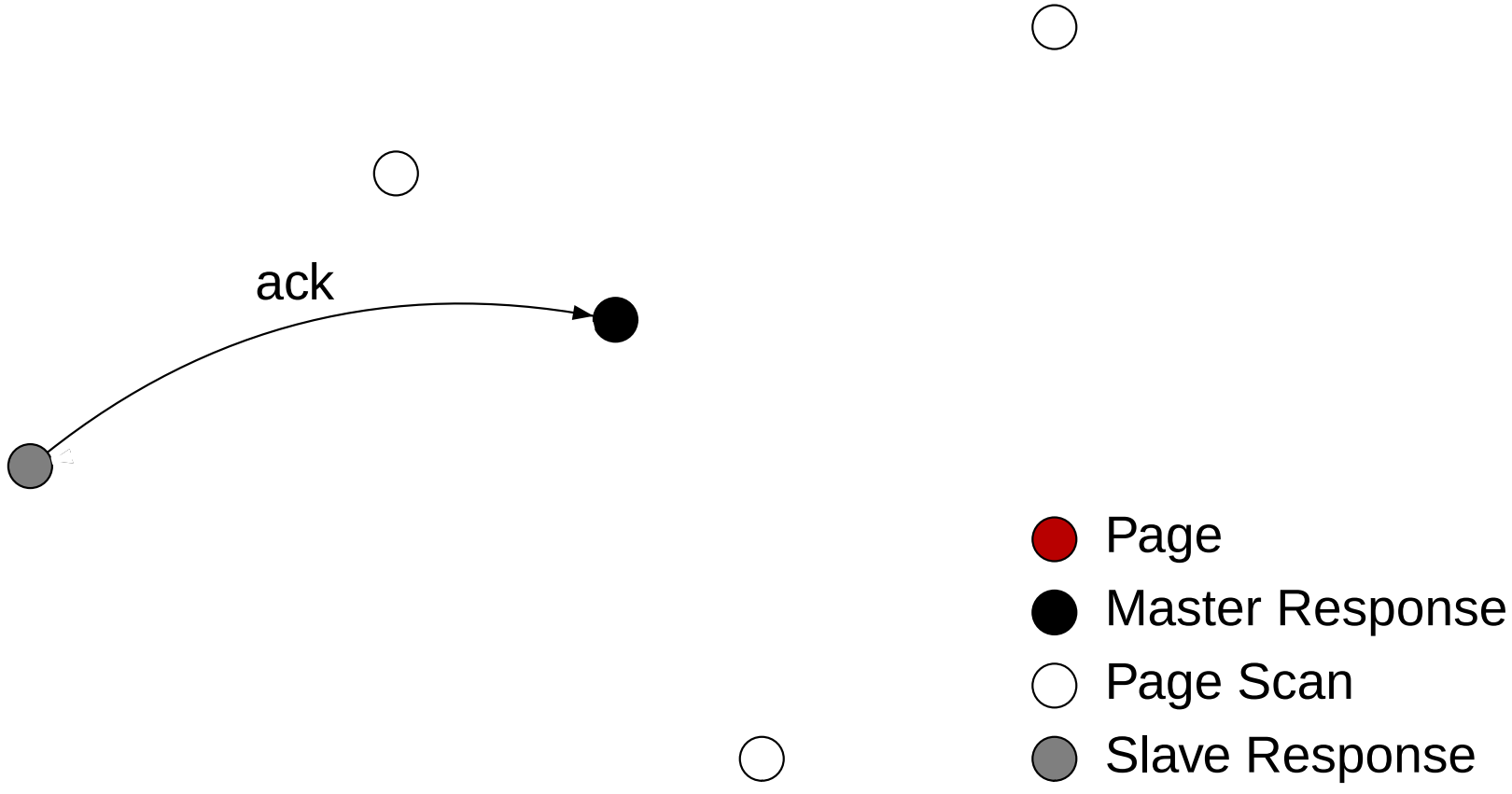
Page



Page



Page



Verifi cation of the protocol

Model-checking

Does the system *satisfy* the property?

Model-checking

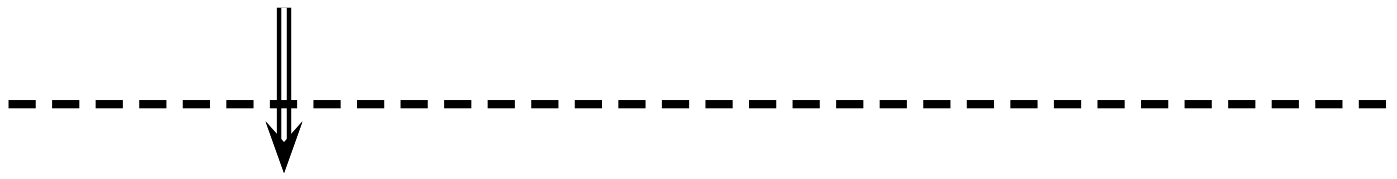
Does the system *satisfy* the property?

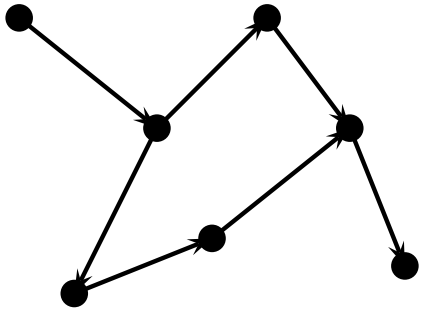
Modelling ----------

Model-checking

Does the system *satisfy* the property?

Modelling

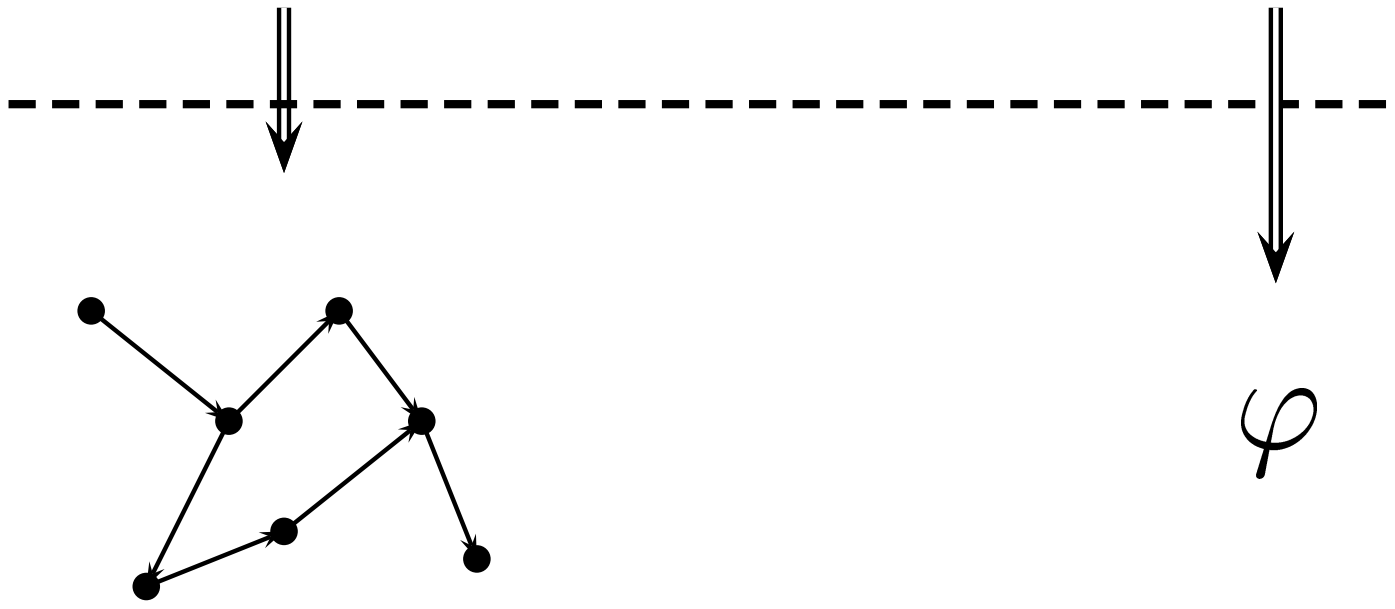




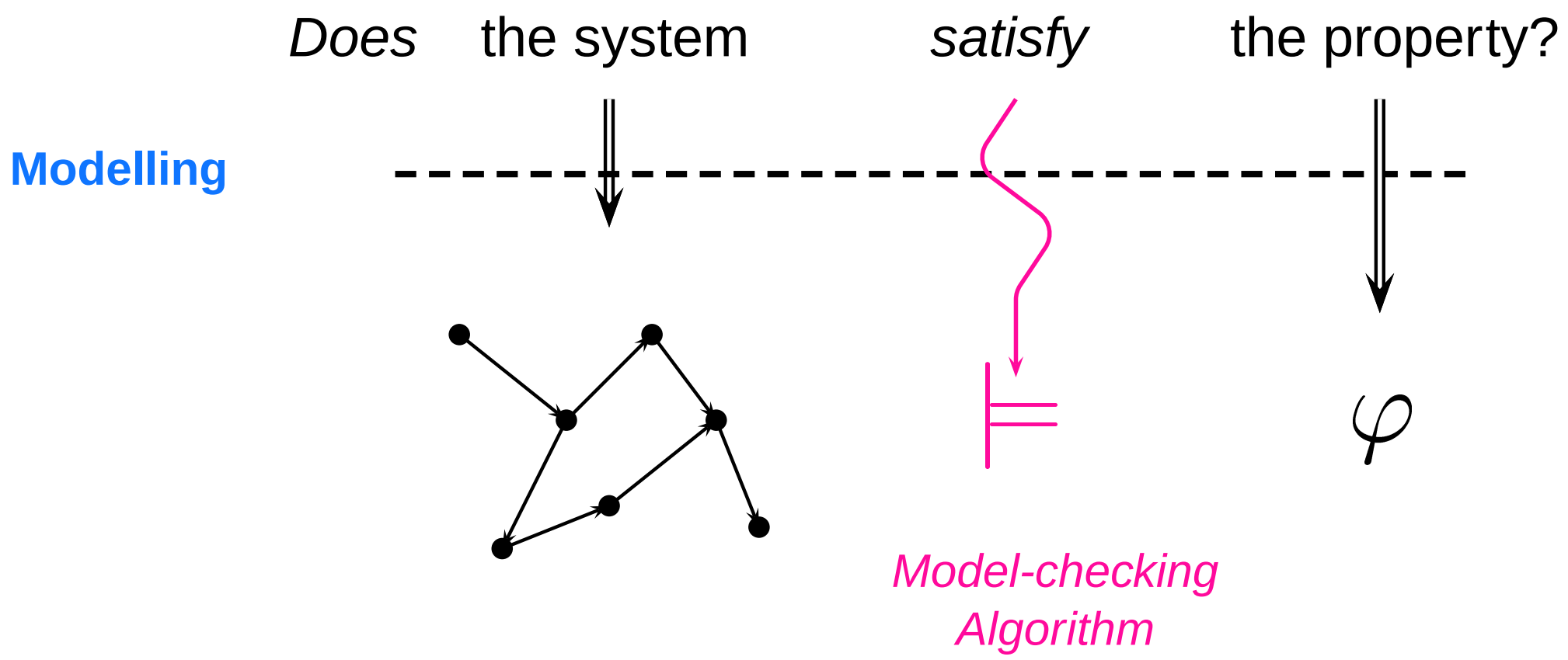
Model-checking

Does the system *satisfy* the property?

Modelling



Model-checking



Abstraction using UPPAAL

The problems:

- Constants are too large.
- Too many states.

The solution: abstract the model

$\Rightarrow$ UPPAAL used to verify that the new model is really an abstraction

- Abstraction relation : trace refinement
- Transformation of probabilistic timed automata into “formerly probabilistic” ones

Abstraction using UPPAAL (2)

The aim : show that $Sys \preceq AbsSys$

Problem : need to use Sys which is too big.

Solution : decompose the system

$$Sys$$

Abstraction using UPPAAL (2)

The aim : show that $Sys \preceq AbsSys$

Problem : need to use Sys which is too big.

Solution : decompose the system

$$Sys \quad = \quad Send \parallel Rec \parallel Rec$$

Abstraction using UPPAAL (2)

The aim : show that $Sys \preceq AbsSys$

Problem : need to use Sys which is too big.

Solution : decompose the system

$$Sys = Send \parallel Rec \parallel Rec$$

$$\upharpoonleft\!\downharpoonright \quad \upharpoonleft\!\downharpoonright \quad \upharpoonleft\!\downharpoonright$$

$$AbsSend \parallel AbsRec \parallel AbsRec$$

Abstraction using UPPAAL (2)

The aim : show that $Sys \preceq AbsSys$

Problem : need to use Sys which is too big.

Solution : decompose the system

$$Sys = Send \parallel Rec \parallel Rec$$

$$\shortmid\!\curlywedge \quad \shortmid\!\curlywedge \quad \shortmid\!\curlywedge$$

$$AbsSys = AbsSend \parallel AbsRec \parallel AbsRec$$

Abstraction using UPPAAL (2)

The aim : show that $Sys \preceq AbsSys$

Problem : need to use Sys which is too big.

Solution : decompose the system

$$\begin{array}{ccc} Sys & = & Send \parallel Rec \parallel Rec \\ \shortmid\!\!\downarrow & & \shortmid\!\!\downarrow \quad \shortmid\!\!\downarrow \quad \shortmid\!\!\downarrow \\ AbsSys & = & AbsSend \parallel AbsRec \parallel AbsRec \end{array}$$

Verification using PRISM

Why use PRISM ?

- The system contains probabilities
- Only digital clocks
- Simple translation from UPPAAL to PRISM

PRISM will be used to verify quantitative properties like

"The proba. of establishing a connection within t time units is at least p"

“The average power consumption before establishing a communication is n”

Challenge : PRISM needs to verify the complete abstract model

Summary

- Need a lot of abstraction
  - Large constants
  - Different time scales
- Other difficulties
  - Specification not precise enough
  - Broadcast synchronisation