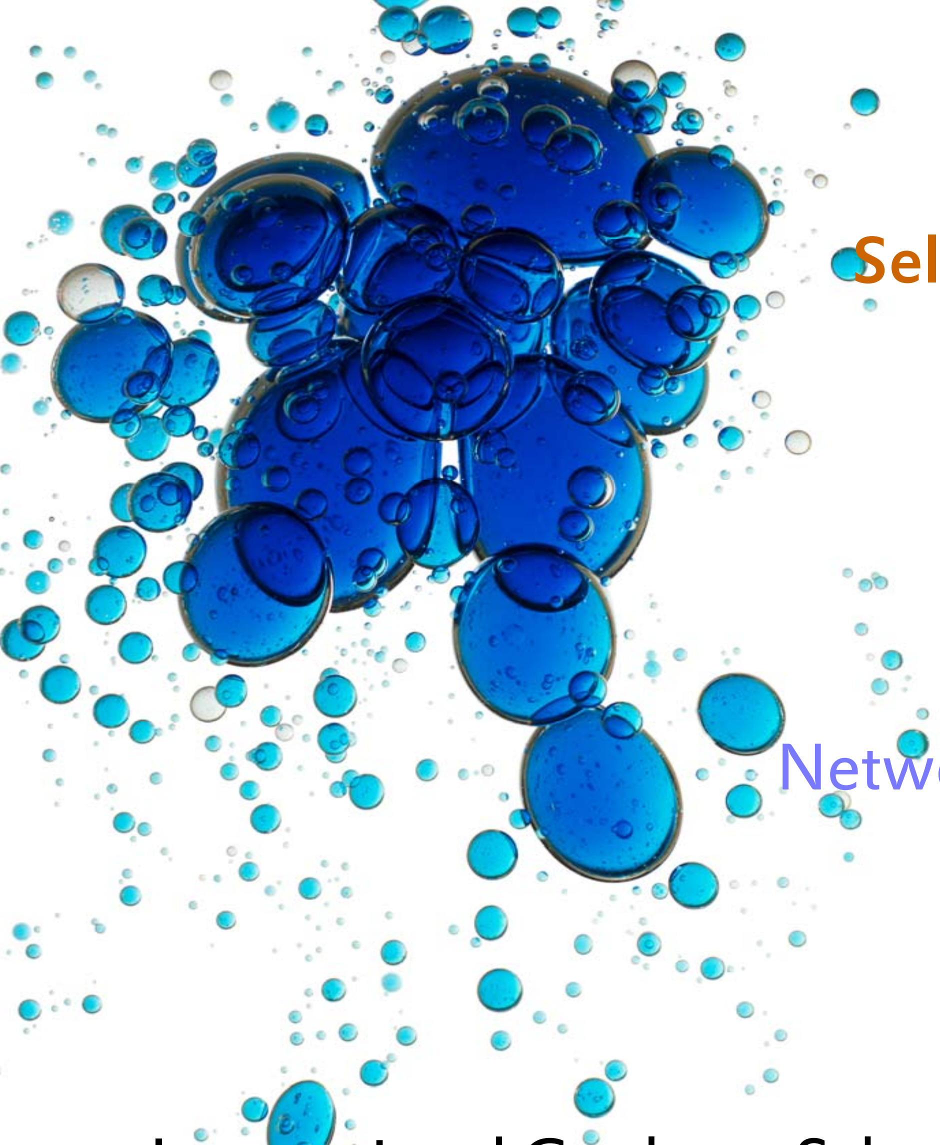


Self-Configuration and Self-Organized Synchronization in Communication Networks

Christian Bettstetter

University of Klagenfurt
Networked and Embedded Systems

Lakeside Labs

Guest Lecture at TU Ilmenau
International Graduate School on Mobile Communications
January 25, 2010

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT

Lakeside Labs

What will be discussed in this lecture?

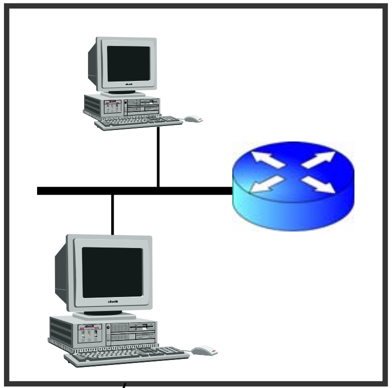
- Why is there a trend toward a higher level of **self-organization** in computer and communication networks?
- What is the meaning of self-organization in this context?
- What is the state-of-the-art? What are research issues?

Discussion using two network functions:

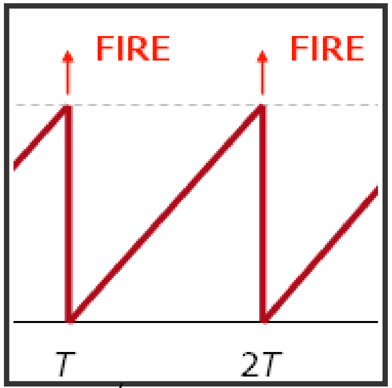
- Self-configuration —— Internet —— state-of-the-art
- Self-synchronization — wireless networks — research

Outline

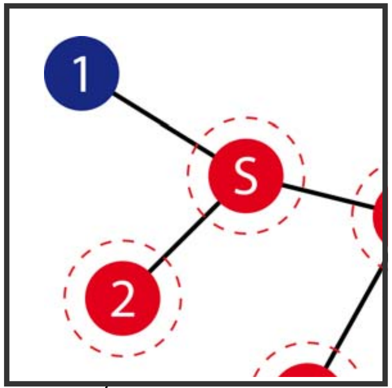
Motivation



Self-Configuration in the Internet



Synchronization in Wireless Networks



Conclusions and Open Issues

Visions from the past …

The Wireless Century

The citizens of the wireless decade will walk around with wireless transceivers attached at hats or something else.

The transceiver will react to myriads of vibrations trying to find connections.

Robert Sloß: Das drahtlose Jahrhundert.
In: *Die Welt in hundert Jahren*, Berlin, 1908.

… became reality.

Mobile Telephony → Mobile Internet → Mobile Multimedia → **Higher data rates More functionality**

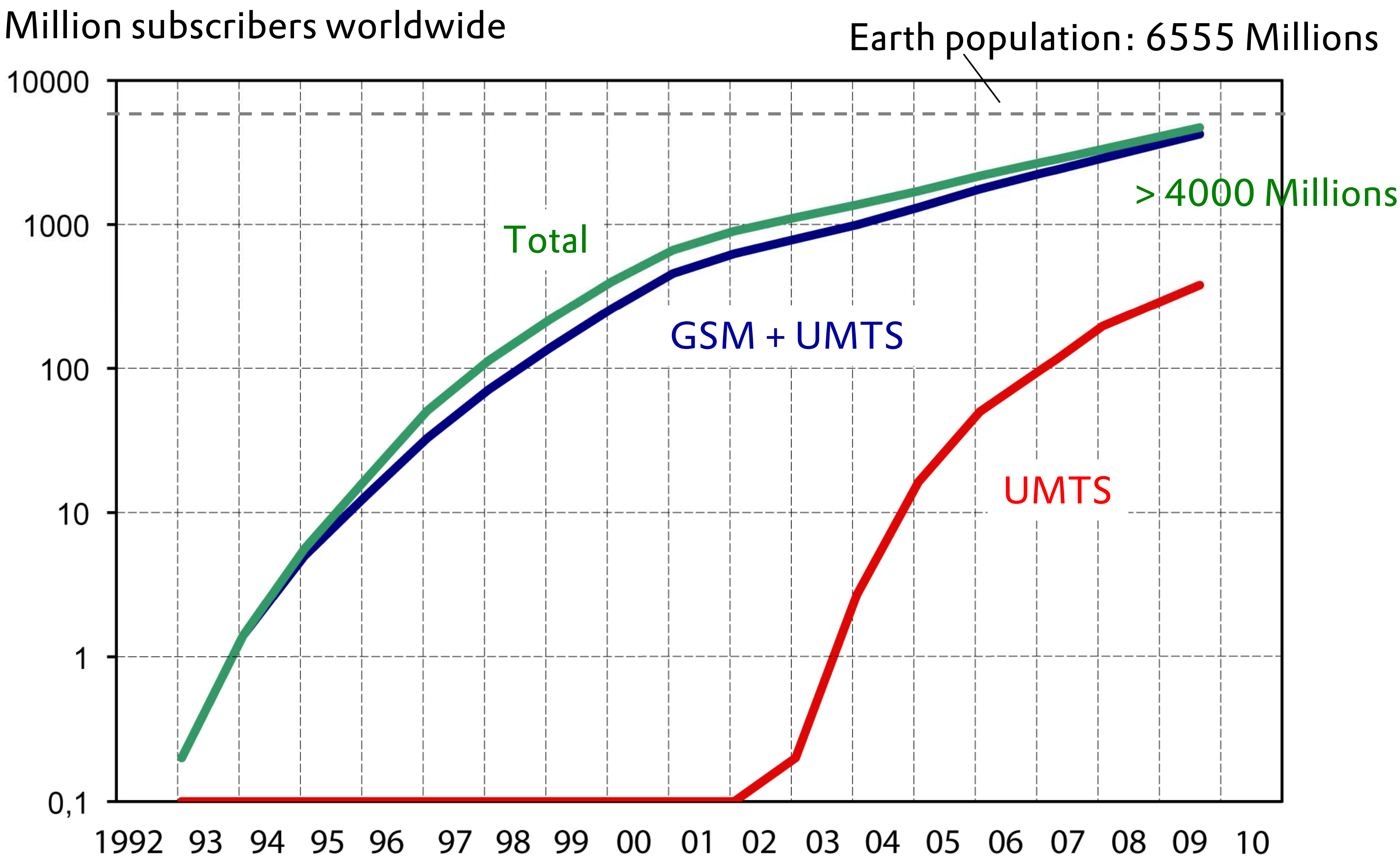
GSM

GPRS, WLAN

UMTS, WLAN, HSPA

GSM	Global System for Mobile Communication
GPRS	General Packet Radio Service
WLAN	Wireless Local Area Network
UMTS	Universal Mobile Telecommunication Network
HSPA	High-Speed Packet Access

GSM: A European Success Story

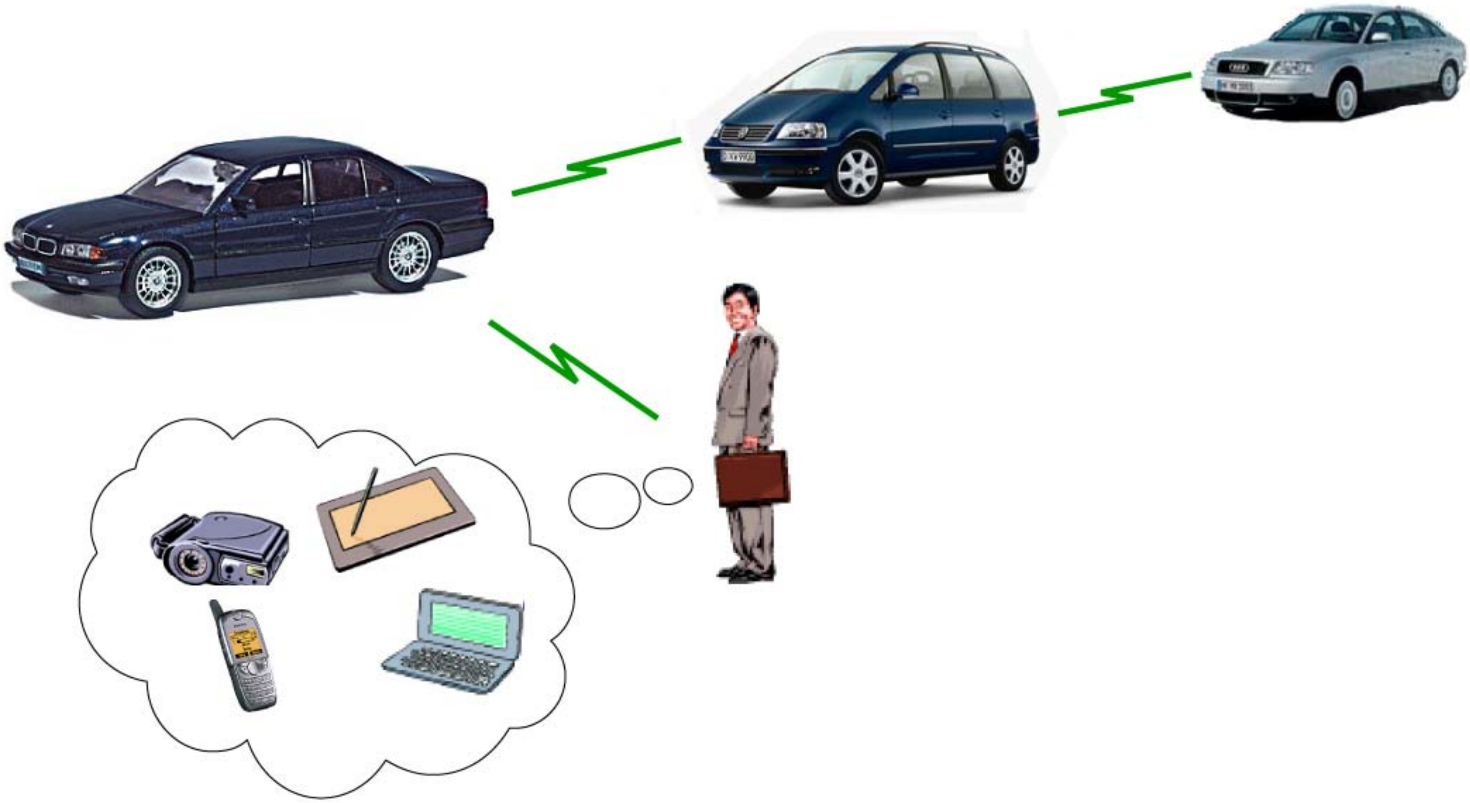


Source: GSM Association (www.gsmworld.com)

New Era: Pervasive Computing

- Computers become **embedded** into everyday objects. They become **invisible** to us.
- Objects are being networked. The **Internet of things** is evolving.
- **Sensors** are an important interface: they link the real world to the virtual one.
- Completely **new applications** arise.

Application Example #1 : Vehicular Networking



Application Example #2: Networks of Wearable Computers

Researchers of a European project on wearable computing for fire fighters, doctors, and plane and car manufacturers.

What is the Problem, if “Every Thing” is Networked?

Increased dynamics and complexity

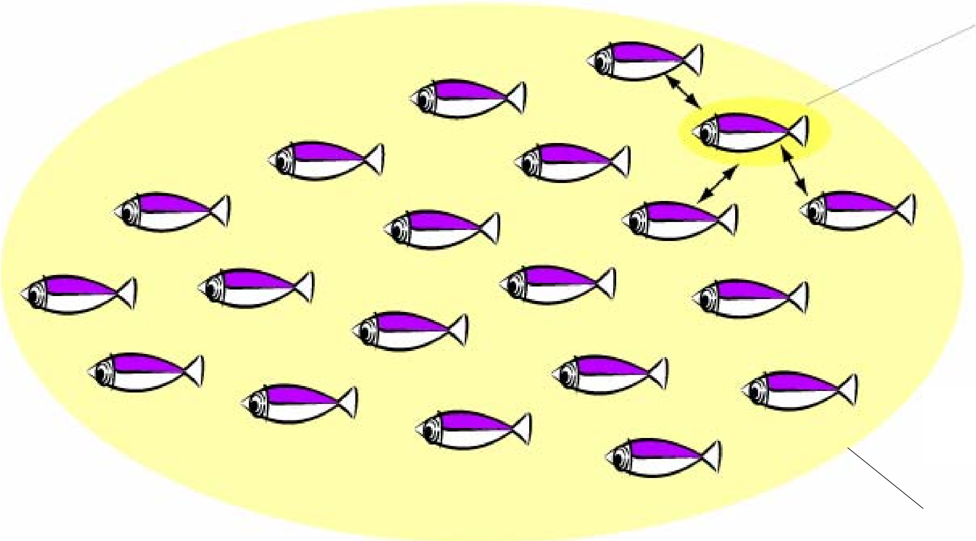
Raises requirements:

- Adaptability
- Distributed operation
- Autoconfiguration

Trend toward self-organization in communication networks:

- Self-configuration in the Internet
- Infrastructureless wireless networks (ad hoc networks)
- Peer-to-peer overlay networks
- Web 2.0, Wikis, social online networks

What is Self-Organization?



Individual Entity („Fish“) *many*

- Simple behavior rules
- Local view
- Distributed operation

Emergence

Entire System (“Shoal”)

- Solves a complex task
- Is adaptive to changes
- Is very scalable

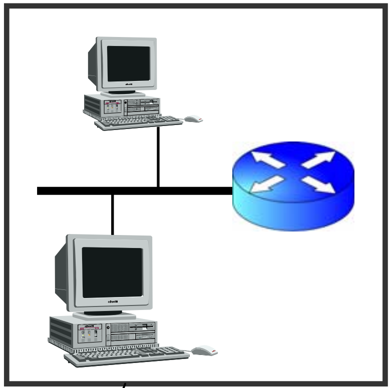
C. Prehofer, C. Bettstetter: Self-Organization in Communication Networks: Principles and Design Paradigms. *IEEE Communications Magazine*, Feature on Advances in Self-Organizing Networks, July 2005.

Bird Swarming

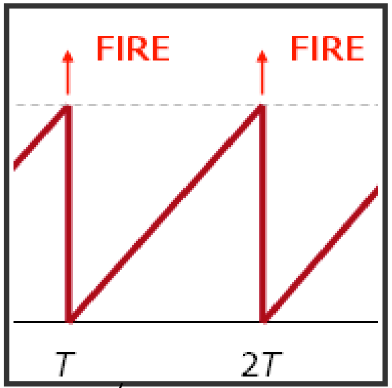
A video is shown here.

Outline

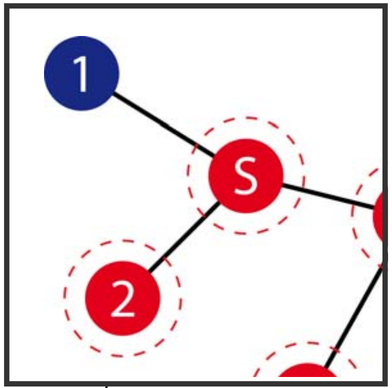
Motivation



Self-Configuration in the Internet

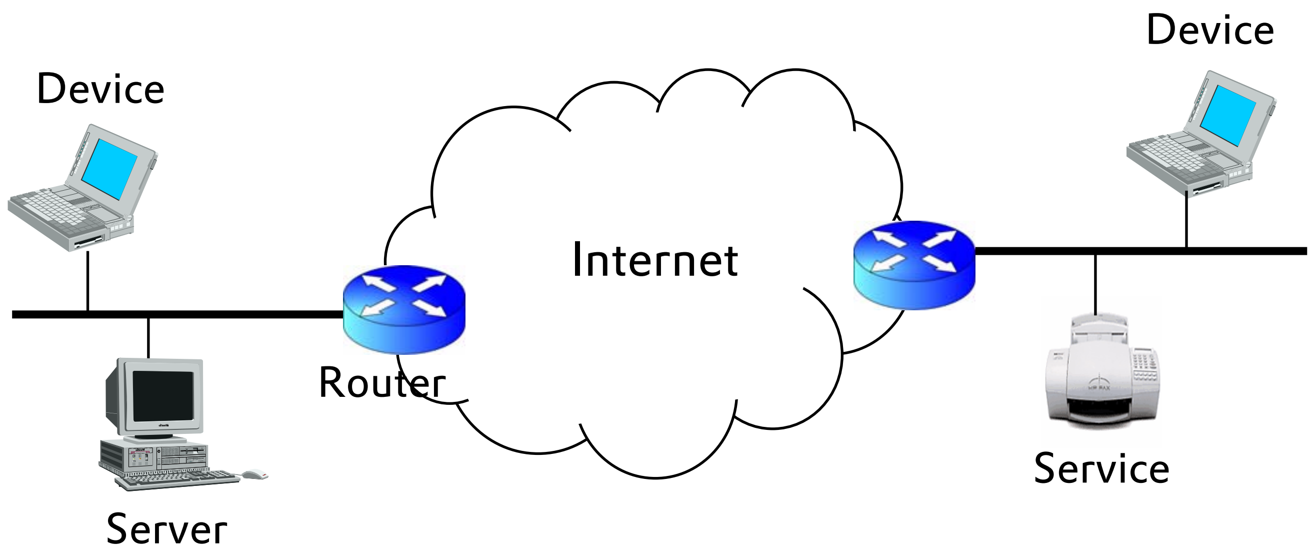


Synchronization in Wireless Networks



Conclusions and Open Issues

Internet Hosts and Network Infrastructure



- Historically, configuration and management has been difficult
- Highly trained network administrators are needed
- Manual configuration is a handicap for mobile devices

Self-Configuration in the Internet

Goal: Enable networking in the absence of manual configuration and human administration.

Aspects:

- Device autoconfiguration
- Service discovery
- Router autoconfiguration

Routers

Device

Services

Self-Configuration in the Internet

Device

- Configure an address to be reachable for others
- Discover and use services offered in the network

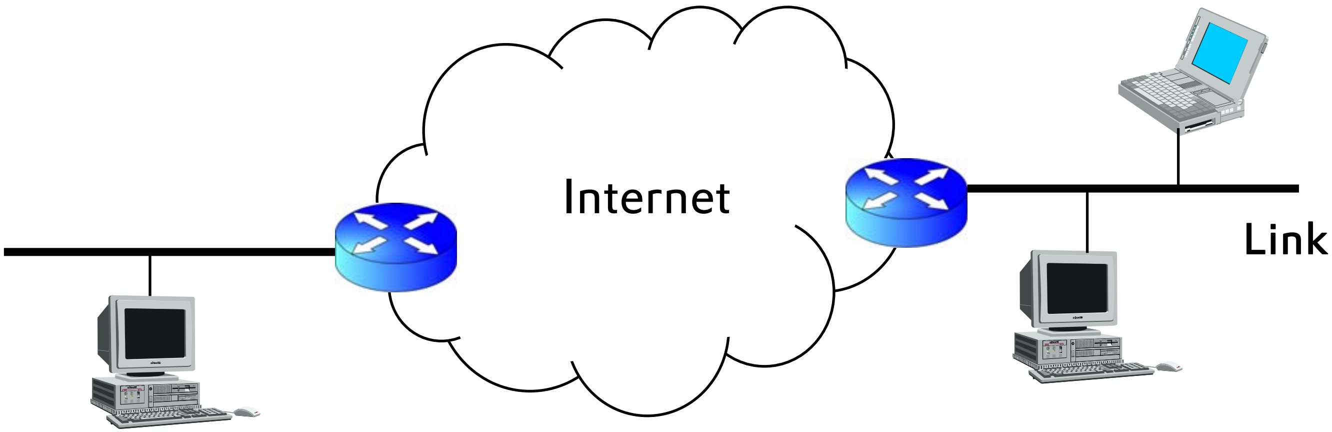
Services

- Advertise the service to network devices
- Enable use of the service

Routers

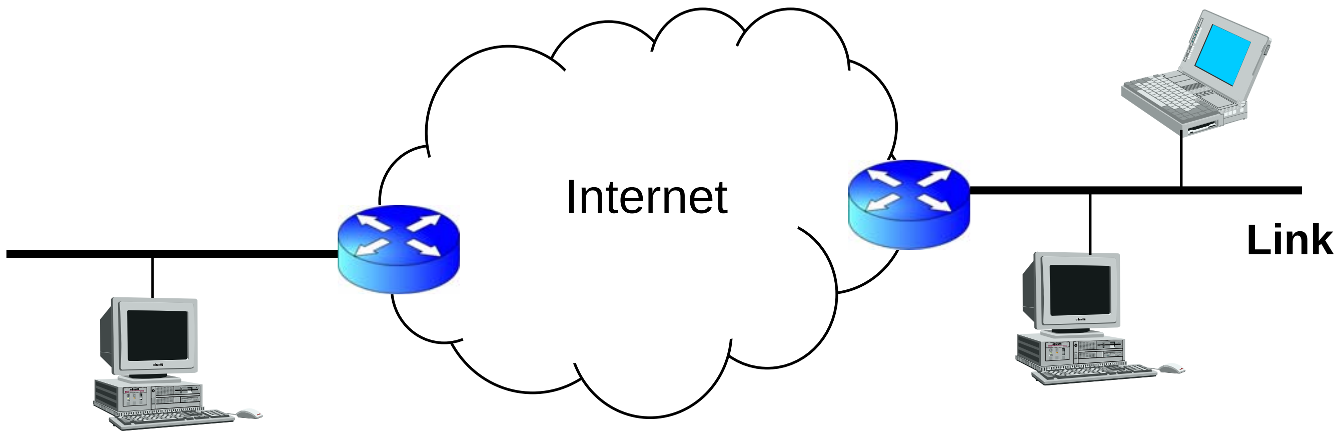
- Configure a network prefix

Addressing in the Internet



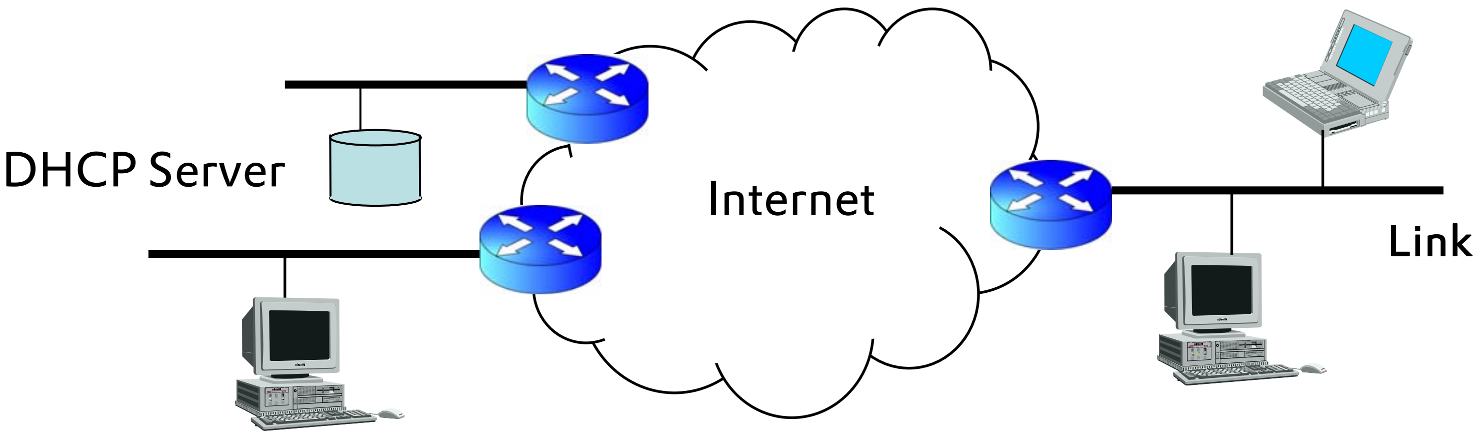
- A device that wants to participate in the Internet needs an **IP address**.
- It is used to **identify** the device and to **route** packets to it.
- How does a device obtain an IP address?

Manual Configuration of IP Addresses



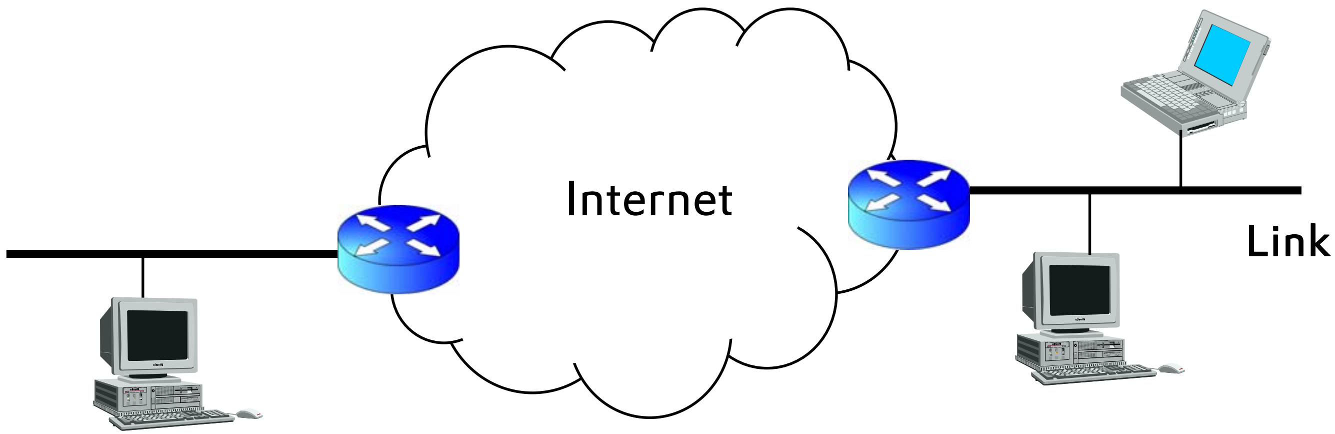
- A system **administrator** configures each device **manually** with an IP address from a specific address space.
- No elements of self-organization; requires **significant human intervention** and creates a very **stiff** address structure

Dynamic, Server-Based Configuration of IP Addresses



- A **server** installed by the administrator in his or her domain **manages** the available IP address pool.
- Using the **Dynamic Host Configuration Protocol (DHCP),** devices are able to automatically obtain an IP address from this server.
- This enables devices to **adapt to changes** in their environment (e.g., to obtain a new IP address when they **move** to a different network).
- This concept is called stateful autoconfiguration.

Self-Configuration of IP Addresses



- A device configures its address **itself** with help of a local router.
- Neither human intervention nor a dedicated server are needed.
- This concept is called stateless autoconfiguration.

- How is it done? How to create a **globally unique** address? → Next slides

Self-Configuration of IP Addresses

Step #1 : Generate unique link-local address (here IPv6)

a. A booting device creates a link-local address:

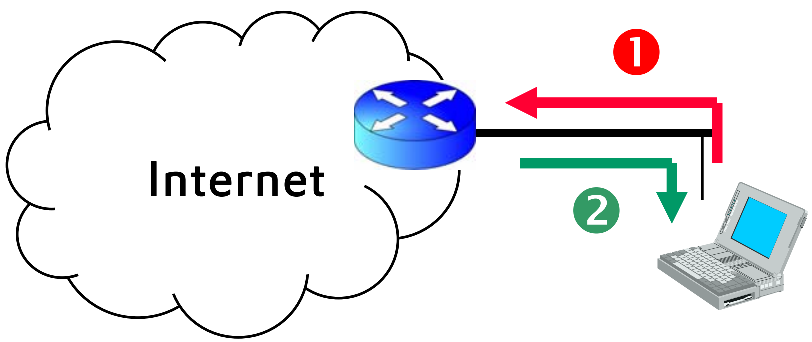
FE 80 00 00 00 00 00 00	Interface identifier	
64 bit	64 bit	IEEE EUI-64

b. The device performs **duplicate address detection** to check if the link-local address is already used by another device on the same link:

- It sends a Neighbor Solicitation message asking for the link-layer address of the generated link-local address
- If another device is already using the link-local address, that device will respond.
- If no respond occurs, it is assumed that the link-local address is unique on this link and can be used.

Self-Configuration of IP Addresses

Step #2: Acquire network prefix from neighboring router



1. Device sends Router Solicitation using link-local address as source address
2. Each router on the link responds using a Router Advertisement, which contains the network prefix

Step #3: Generate globally valid IP address

Network prefix	Interface identifier

Self-Configuration in the Internet

Goal: Enable networking in the absence of manual configuration and human administration.

Aspects:

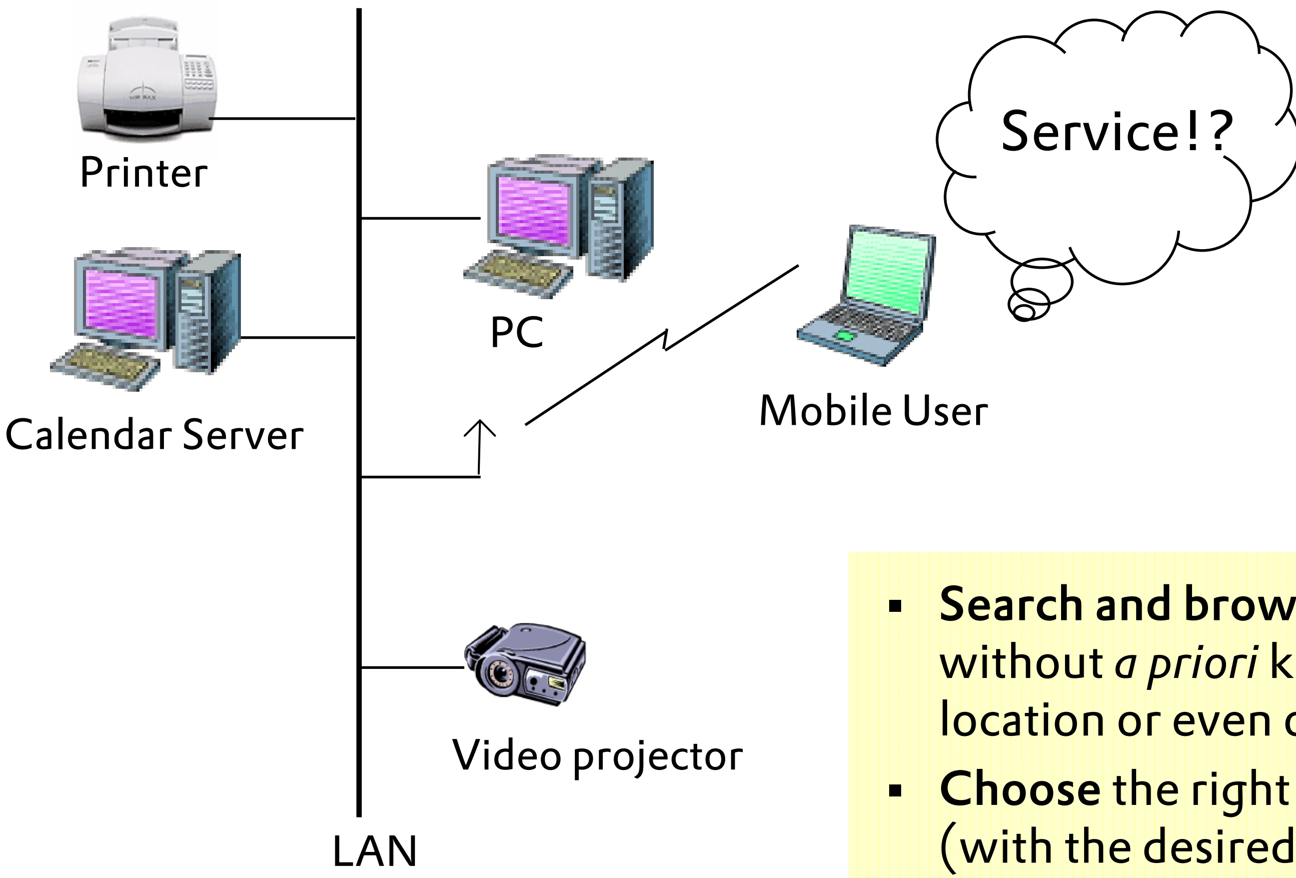
- Device autoconfiguration
- **Service discovery**
- Router autoconfiguration

Routers

Device

Services

Service Discovery: What is it?



- **Search and browse** for services, without *a priori* knowledge of their location or even of their existence
- **Choose** the right service (with the desired characteristics)
- **Utilize** the service

Service Discovery Protocols

Java Intelligent Network Infrastructure

Universal Plug and Play (UPnP)

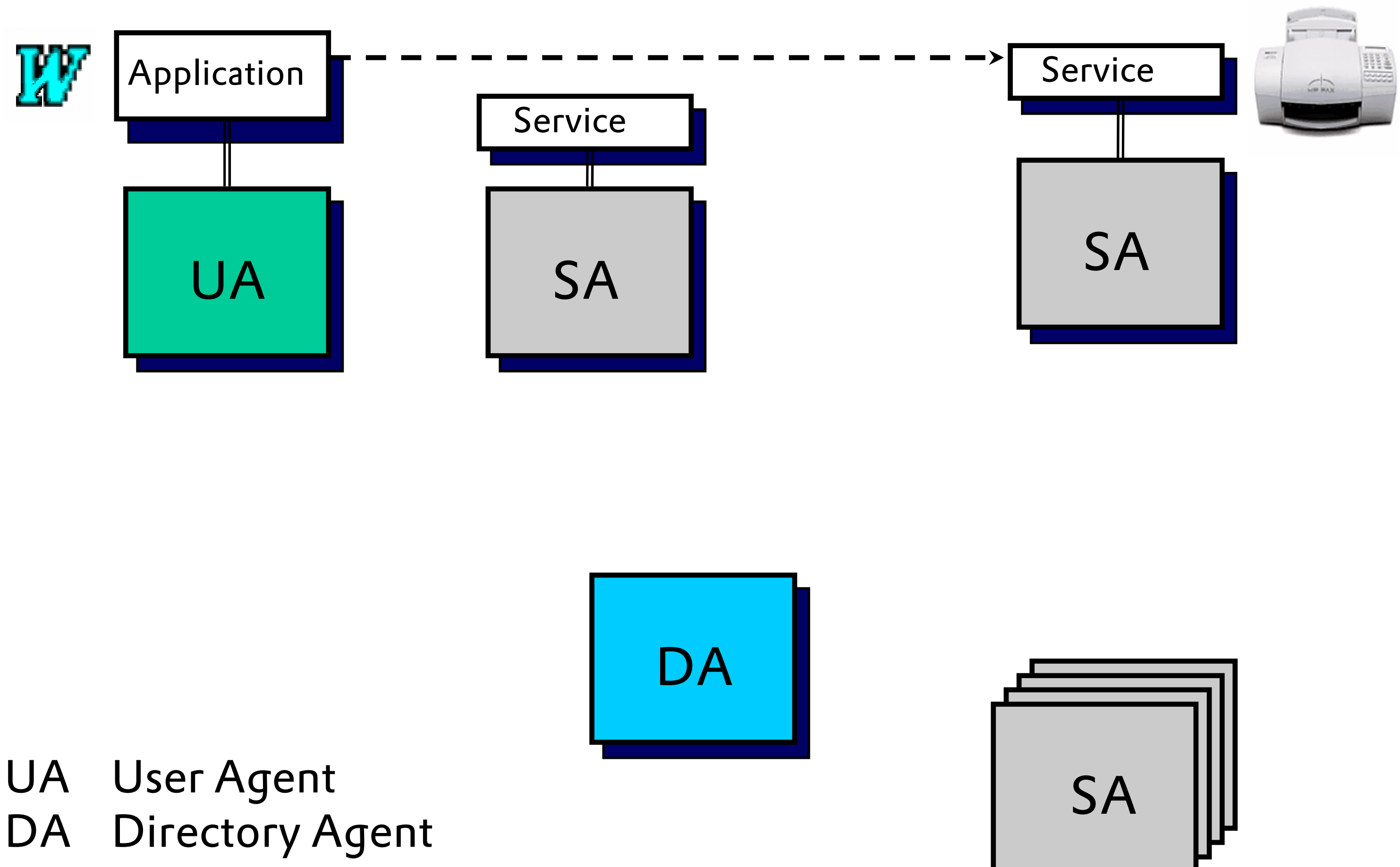
SLP

Service Location Protocol

ZeroConf („Bonjour“)

Bluetooth SDP

SLP Architecture



UA	User Agent
DA	Directory Agent
SA	Service Agent

SLP Architecture: Three Main Components

SA

Service Agent
advertises and registers a service and its characteristics
on behalf of the service

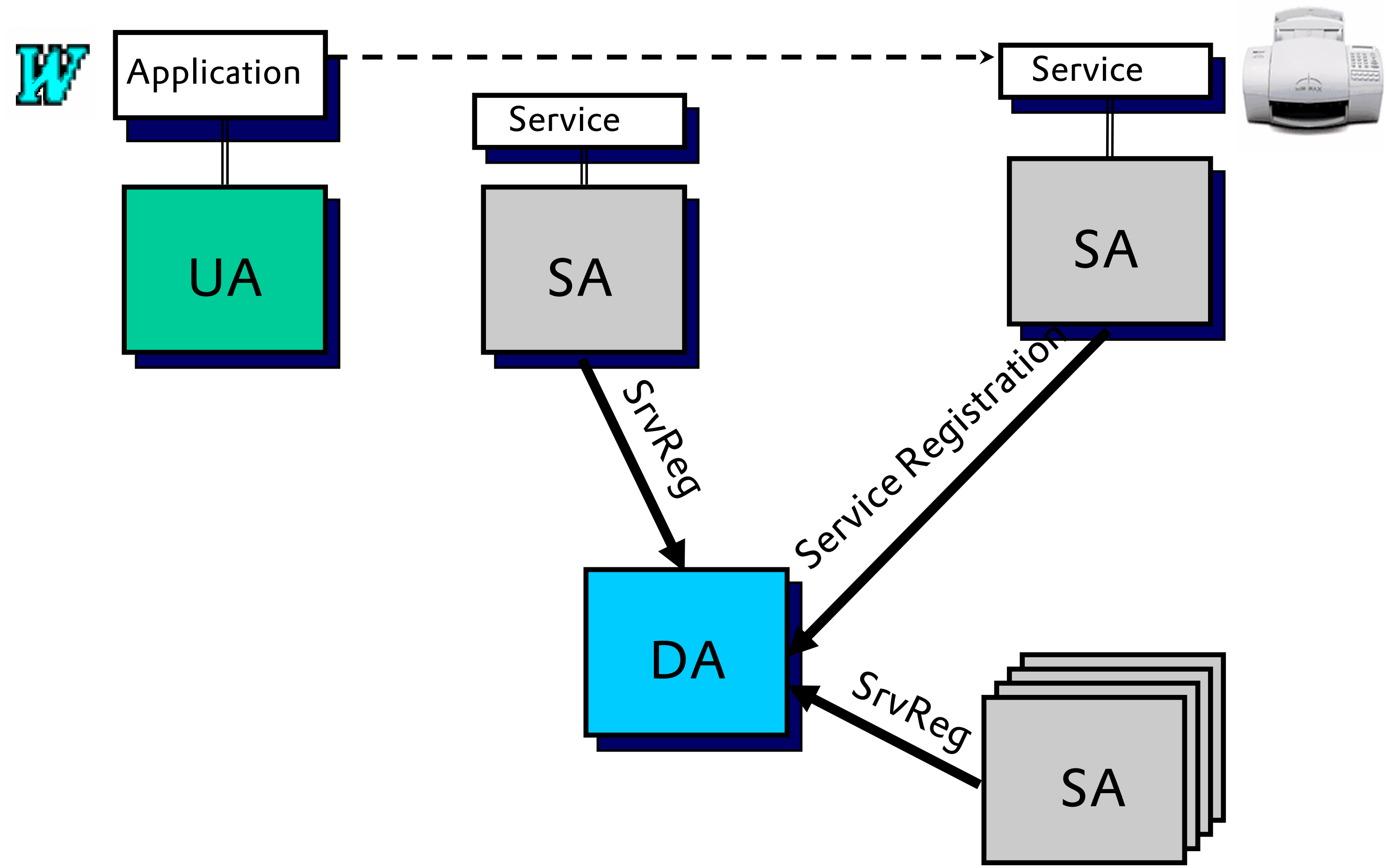
UA

User Agent
performs service discovery
on behalf of a client (application or user)

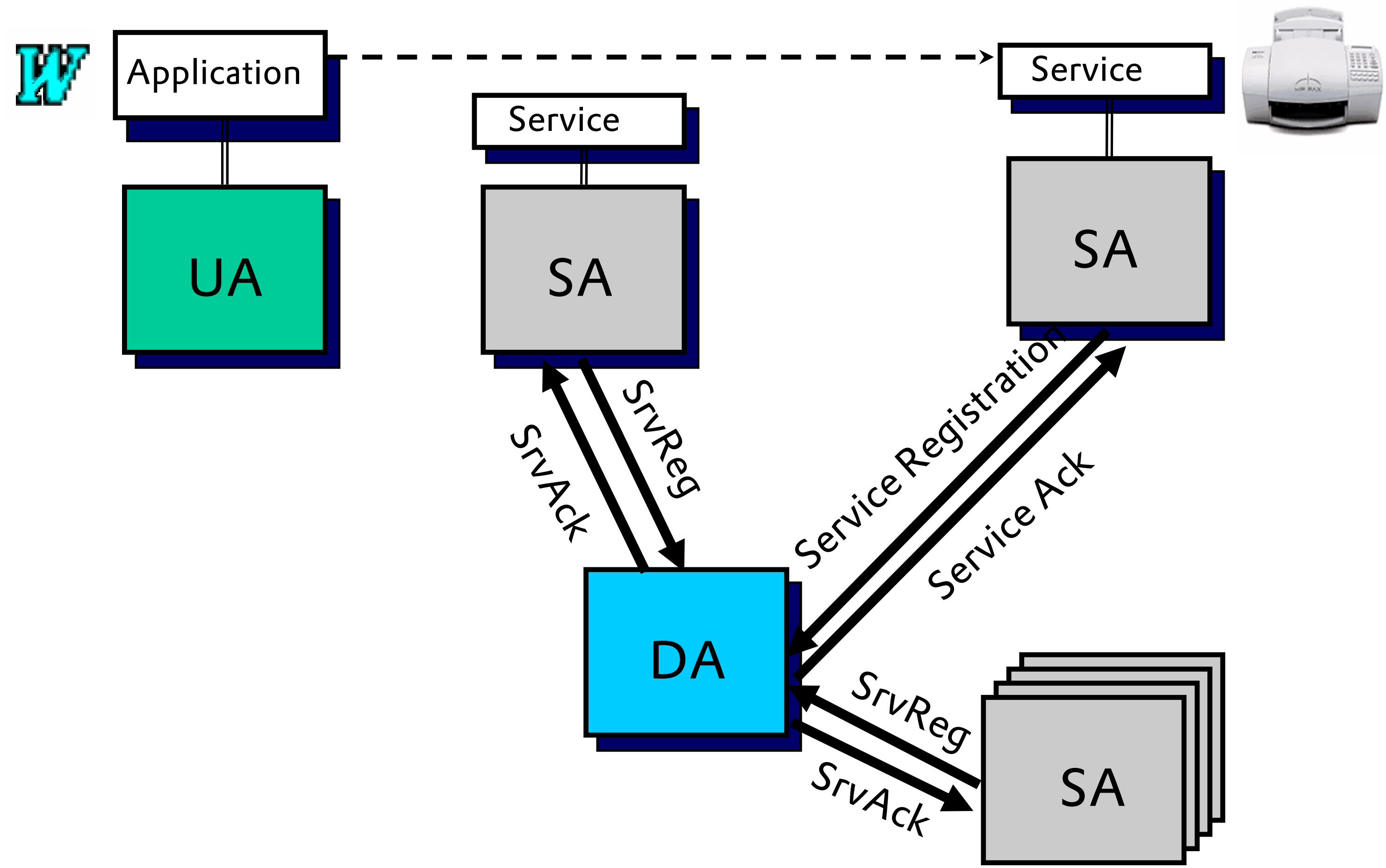
DA

Directory Agent
collects service registrations from SAs and
handles service requests from UAs

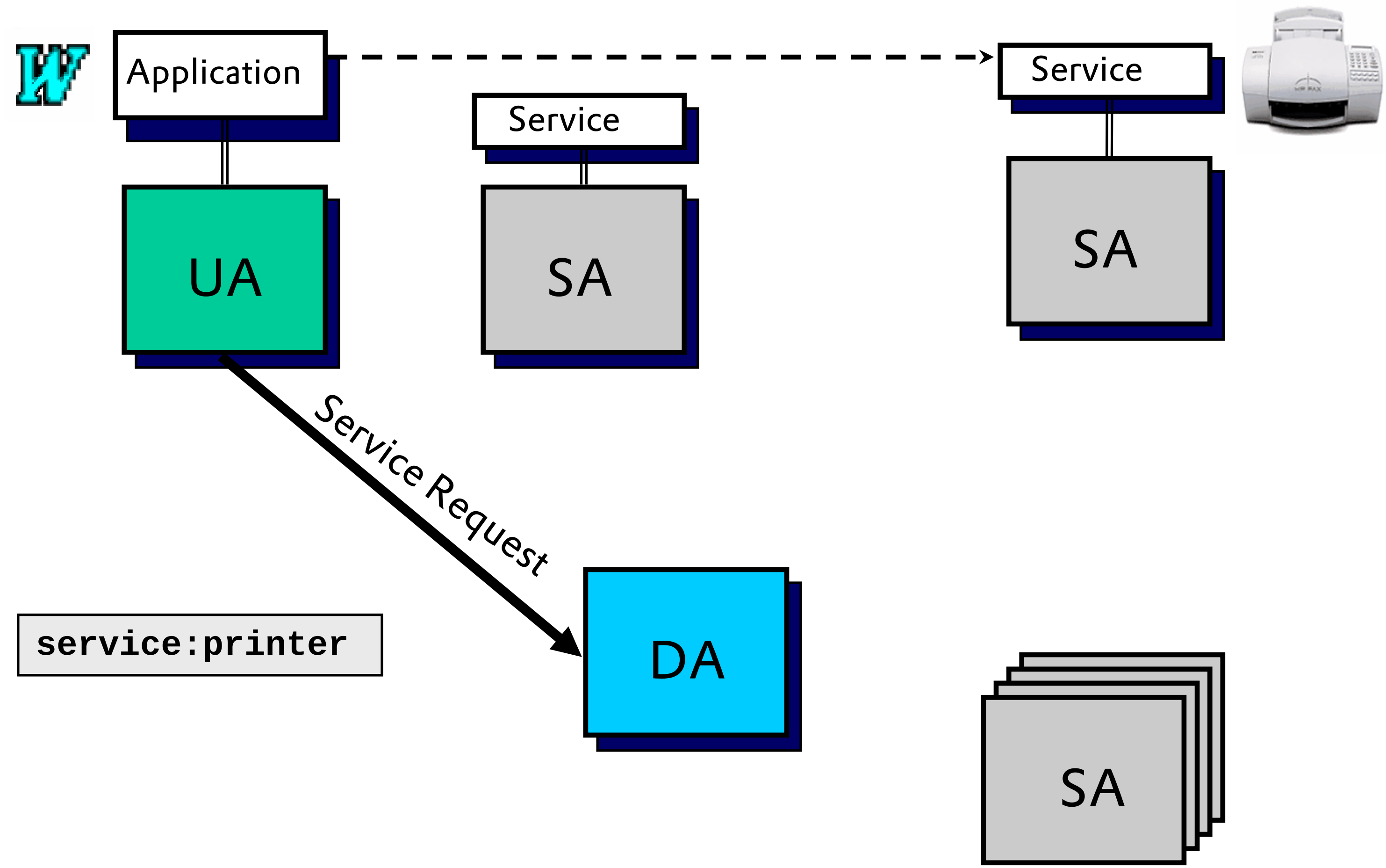
Service Registration and Update



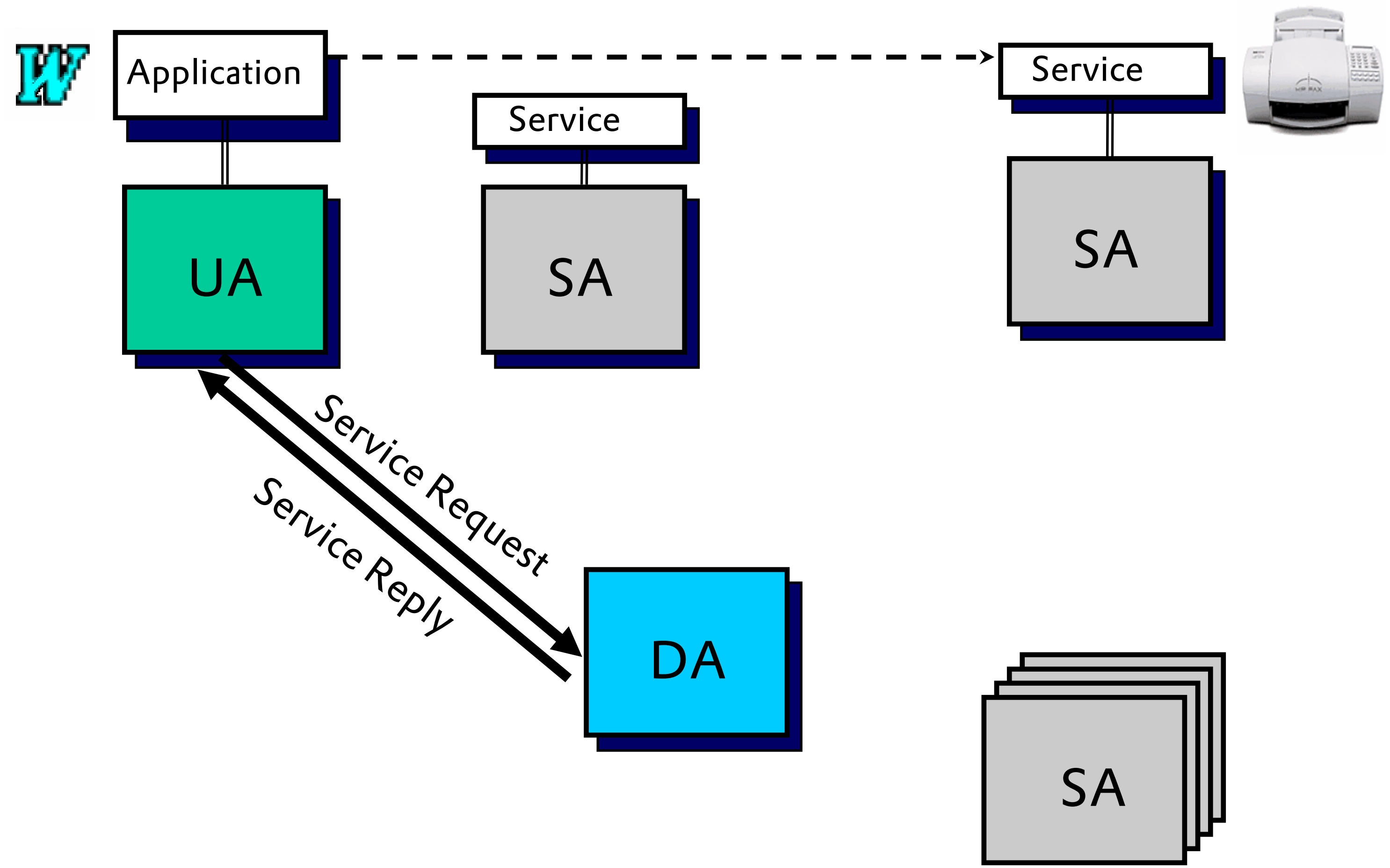
Service Registration and Update



Service Discovery



Service Discovery



`service:printer://lj4050.uni-klu.ac.at:1020/queue1`

Service URLs and Templates

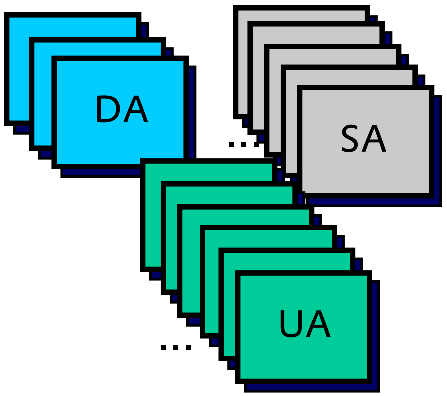
```
service:printer://lj4050.tum.de:1020/queue1

scopes = uni-klu,itec,administrator
printer-name = lj4050
printer-model = HP LJ4050 N
printer-location = Room L02.01.09
color-supported = true
pages-per-minute = 9
sides-supported = one-sided, two-sided
```

Service Templates

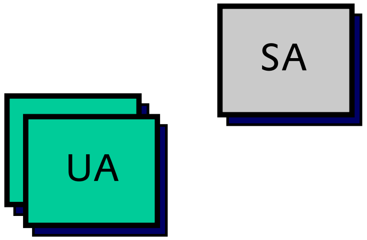
- Define attributes and default values
- Are registered with IANA (Internet Assigned Numbers Authority)
- Enable interoperability between different vendors

Different Types of Operation

- **Large network environments:**
  - More DAs for load sharing
  - Scopes to group resources/services.
    E.g. all devices in one room are one scope
  - Access policies for scopes



- **Small network environments**
  - SLP also works without DA
  - without DHCP, without DNS, without routing



SLP Standardization and Implementation

Standardization

- Internet Engineering Task Force (IEFT)
- Reached status „proposed standard“
- Three versions: SLP, SLPv2 und SLPv3

Major Implementations

- Solaris
- Linux
- Mac OS X (up to version 10.1)
- Novell NetWare

Zero Configuration Networking (ZeroConf)

Functionality

- Address autoconfiguration:
  Configure an IP address without a DHCP server
- Service Discovery:
  Discover and advertise services without a directory server
- Address translation:
  Translate between IP addresses and names without a DNS server

Development

- IETF Zeroconf working group
- Apple's Rendezvous, later Bonjour
- Other open source implementations

Major Implementations

- Mac OS X (used e.g. for iTunes, AirPort, AirTunes, Apple TV)
- Linux and BSDs

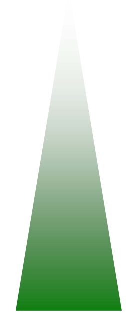
Self-Configuration in the Internet: Summary

Network function:

- Address autoconfiguration
- Service discovery
- Address translation
  (http://www.uni-klu.ac.at → 143.205.180.80)

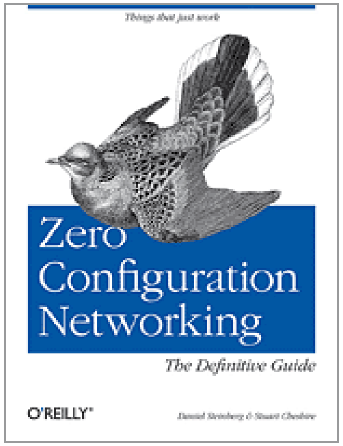
Degree of self-configuration:

- Manual, static configuration
- Dynamic configuration with help of a server
- Automatic configuration



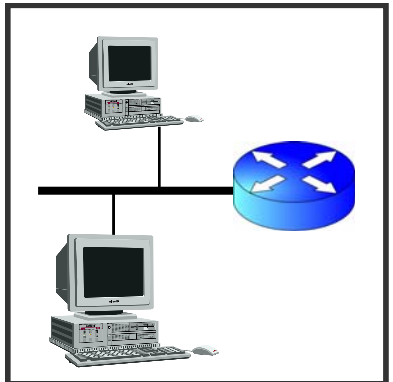
Literature: Self-Configuration in the Internet

- T. Narten: Neighbor discovery and stateless autoconfiguration in IPv6. *IEEE Internet Computing*, July/August 1999.
- E. Guttman: Service location protocol: Automatic discovery of IP network services, *IEEE Internet Computing*, July/August 1999.
- E. Guttman: Autoconfiguration for IP Networking, *IEEE Internet Computing*, May/June 2001.
- D. Steinberg, S. Cheshire: *Zero Configuration Networking: The Definitive Guide*, O'Reilly, Dec 2005.

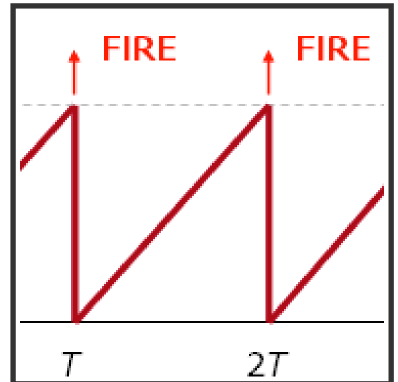


Outline

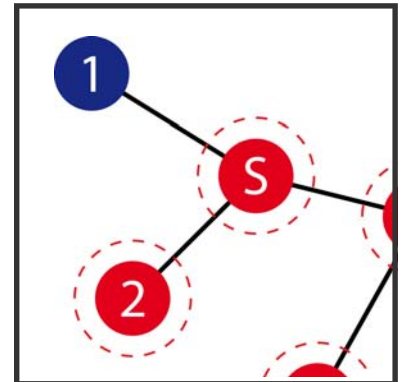
Motivation



Self-Configuration in the Internet



Synchronization in Wireless Networks



Conclusions and Open Issues

Synchronization

Definition

- Greek: *sýn* meaning *together* and *chrónos* meaning *time*
- Synchronous events: Events occur at the same time
- Synchronization: Adjustment that causes events to occur at the same time

Occurring in a variety of fields

- People arriving at a meeting at the same time
- Sports: synchronized swimming
- Music: orchestra with a conductor (next slide)
- Technology: computer clock synchronization

In the context of communications and computing, the term "synchronization" is used very broadly.

What is Synchronization? Experiment with Metronomes

Examples of Synchronization in Communications

- Network Time Protocol (NTP):
  Used to synchronize clocks of Internet routers and hosts via a hierarchy of time servers and clients.
- Time and frequency synchronization in cellular networks: A mobile station synchronized to its base station
- Global Positioning System (GPS)

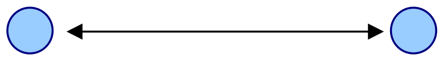
Synchronization in Communications and Computing

- Carrier synchronization
- Symbol or bit synchronization
- Slot synchronization
- Frame synchronization
- Packet synchronization

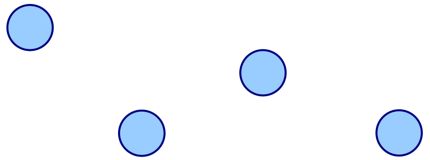
- Clock synchronization
- Data or file synchronization
- Multimedia synchronization

- ...

Point-to-point synchronization:

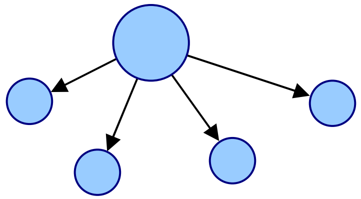


Network synchronization:

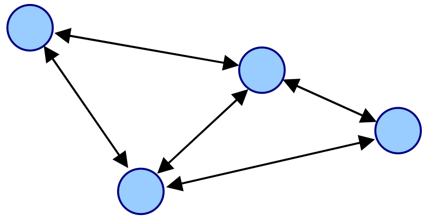


- Internal synchronization
- External synchronization

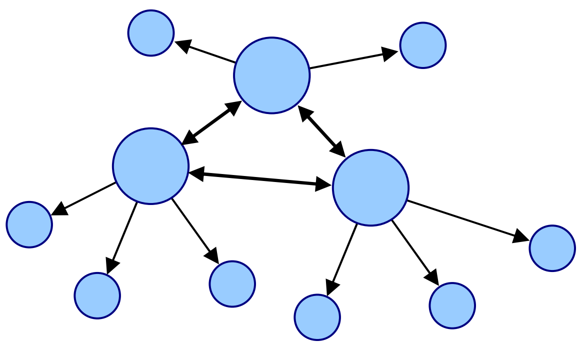
Network Synchronization Strategies



Master-slave synchronization
(monarchy)

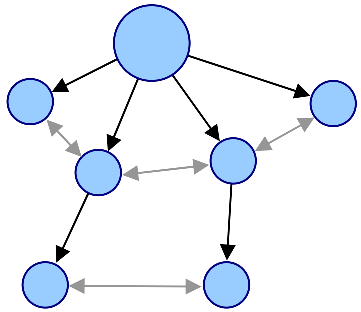


Mutual synchronization
(democracy)

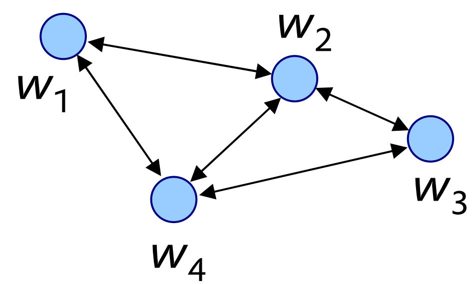


Mix of master-slave and mutual
synchronization (oligarchy)

Network Synchronization Strategies

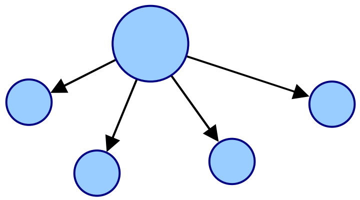


Hierarchical
master-slave synchronization
(hierarchical monarchy)

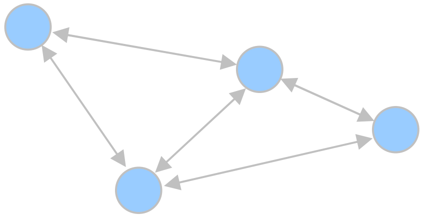


Hierarchical
mutual synchronization
(hierarchical democracy)

Master-slave synchronization

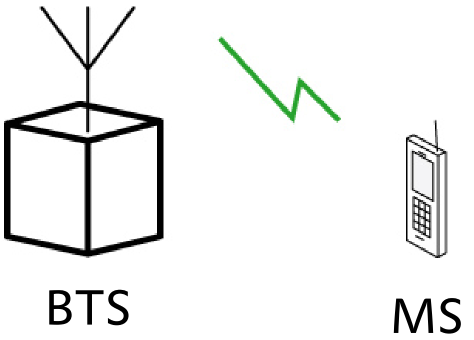


Master-slave synchronization
(monarchy)



Mutual synchronization
(democracy)

Synchronization in GSM (between BTS and MS)



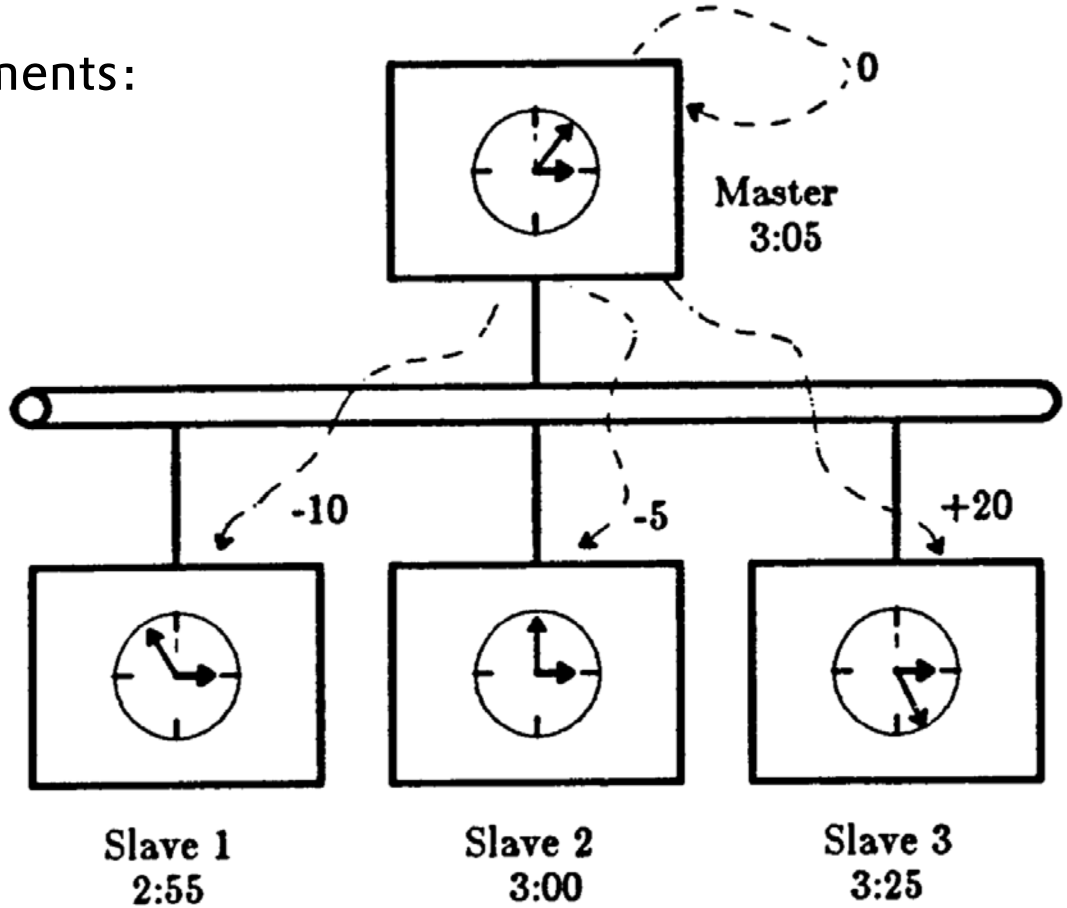
The BTS sends signals on the *Broadcast Control Channel (BCCH)* to enable the MS to synchronize itself to the BTS.

- Carrier frequency synchronization (*frequency correction bursts*): Adjustment of the sending and receiving frequencies of a MS to the frequencies of the BTS
- Time synchronization *(frequency correction, synchronization bursts*):
  - Frame synchronization: Adjustment of the start of a periodically repeating transmit frame
  - Bit synchronization

Synchronization in LANs: Berkeley algorithm (1980s)

Step #1:

Measurements:

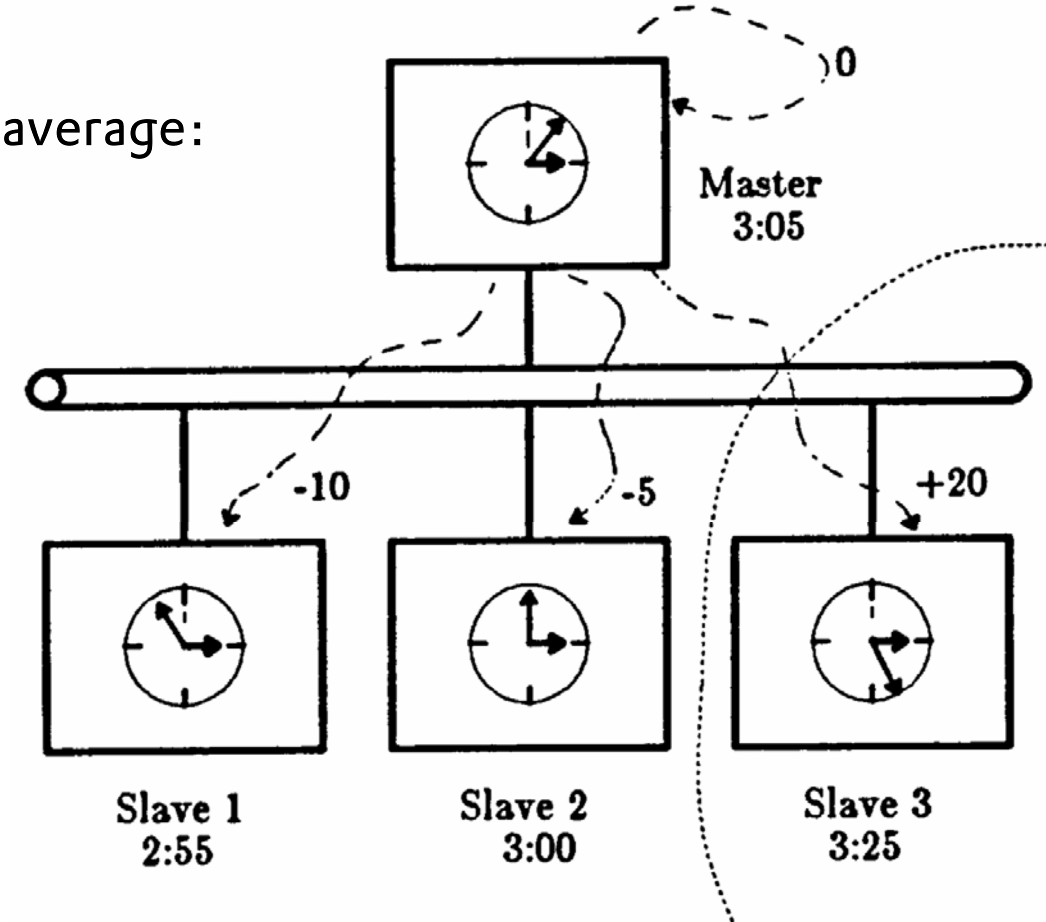


R. Gusella and S. Zatti: The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD, *IEEE Trans. on Software Engineering*, 15 (7): 847–853, 1989. © 1989 IEEE. Figure used with kind permission from the IEEE IPR Office.

Berkeley algorithm (1980s)

Step #2:

Compute average:



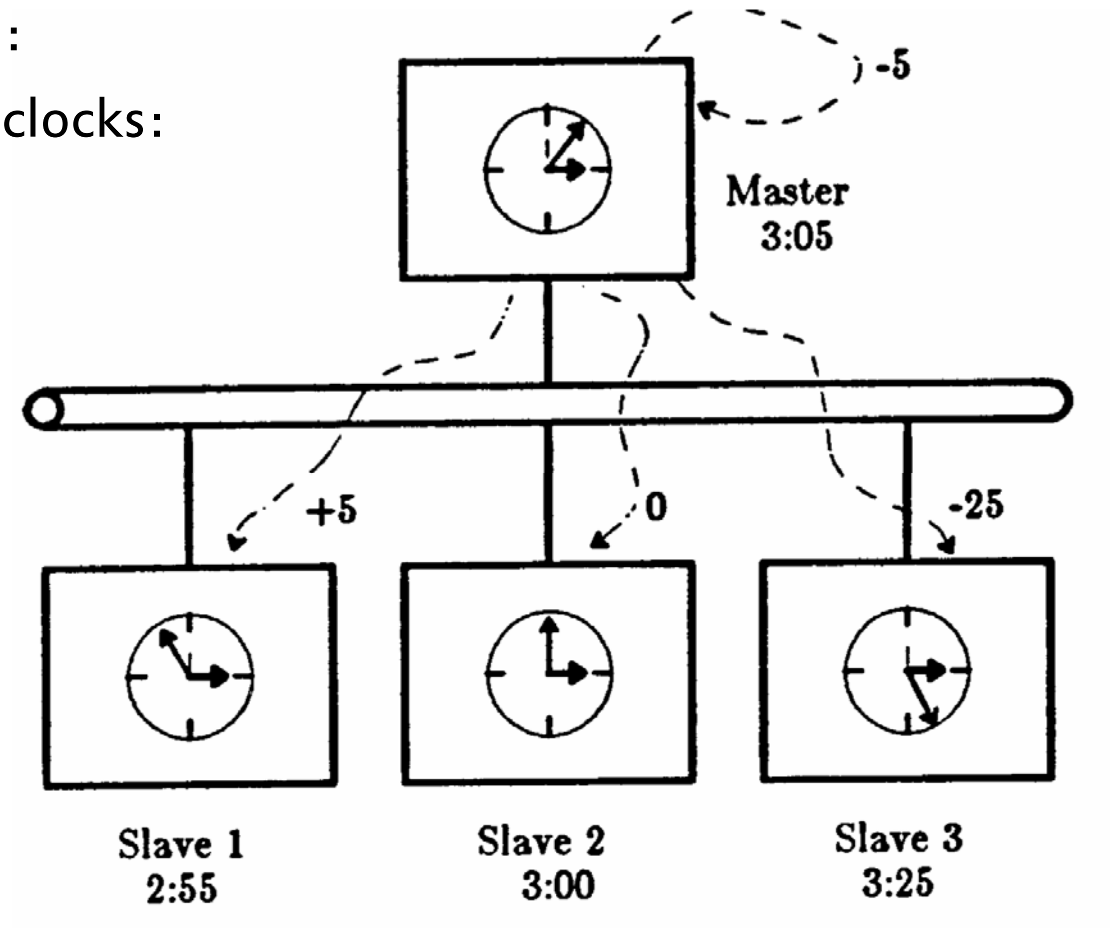
$$Av = \frac{0 - 10 - 5}{3} = -5$$

R. Gusella and S. Zatti: The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD, *IEEE Trans. on Software Engineering*, 15 (7): 847–853, 1989.

Berkeley algorithm (1980s)

Step #3:

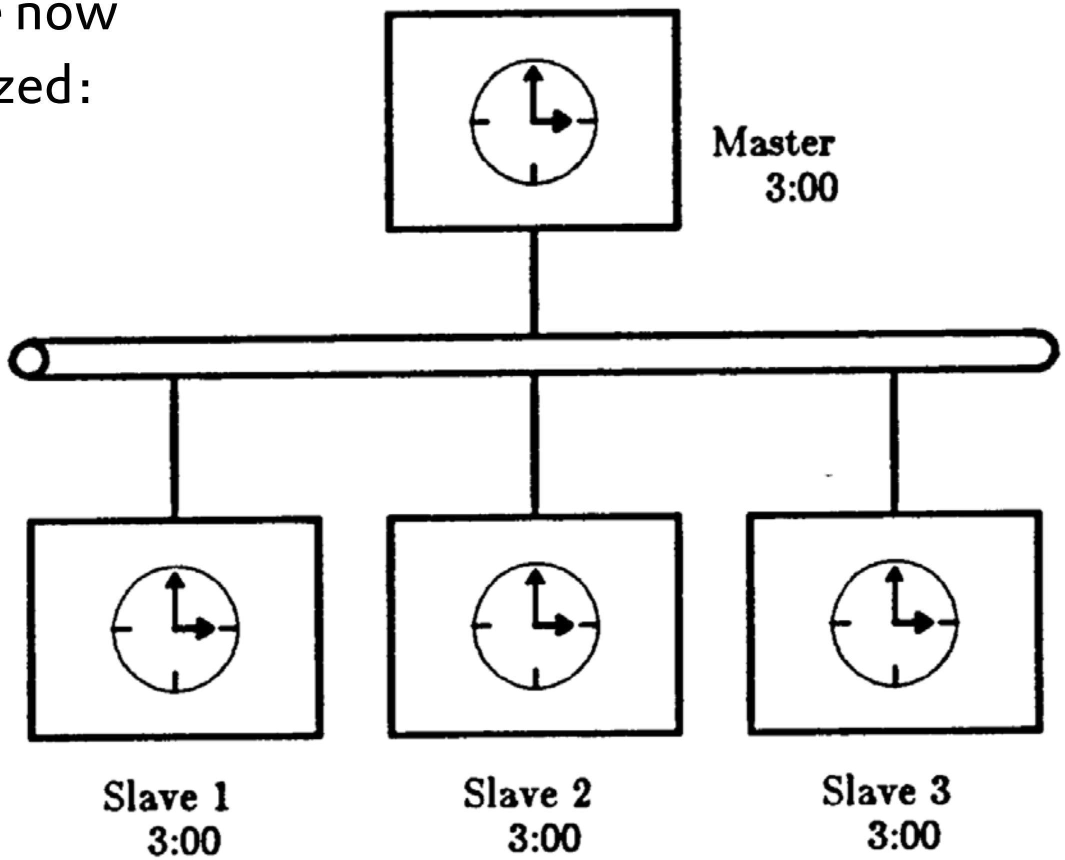
Correct clocks:



R. Gusella and S. Zatti: The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD, *IEEE Trans. on Software Engineering*, 15 (7): 847–853, 1989. © 1989 IEEE. Figure used with kind permission from the IEEE IPR Office.

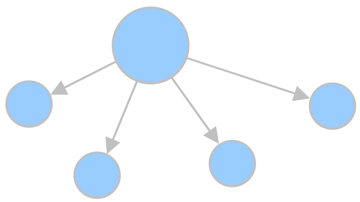
Berkeley algorithm (1980s)

Clocks are now synchronized:

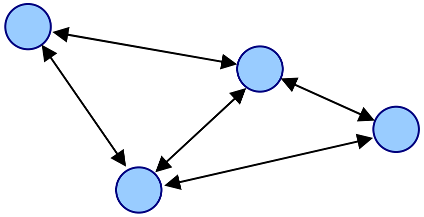


R. Gusella and S. Zatti: The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD, *IEEE Trans. on Software Engineering*, 15 (7): 847–853, 1989. © 1989 IEEE. Figure used with kind permission from the IEEE IPR Office.

Mutual synchronization (completely distributed synchronization)



Master-slave synchronization
(monarchy)



Mutual synchronization
(democracy)

Synchronous Flashing of Fireflies in South-East Asia

A video is shown here.

Synchronous Flashing of Fireflies in South-East Asia

A video is shown here.

Synchronous Flashing of Fireflies in South-East Asia

Early hypotheses of the mechanism

- Environment (e.g. wind, thunder) triggers the synchronization
- Some “leader” firefly controls the synchronized flashing

Experimental work (1960s to 80s)

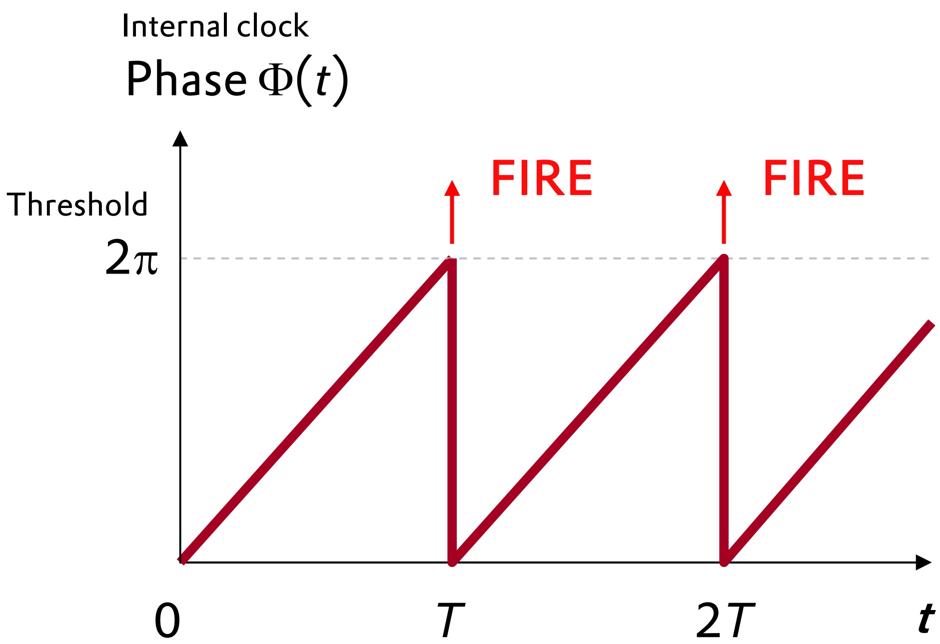
- Firefly in a dark room flashes with quite constant frequency
- Exposed to generated light flashes, it responds to these stimuli

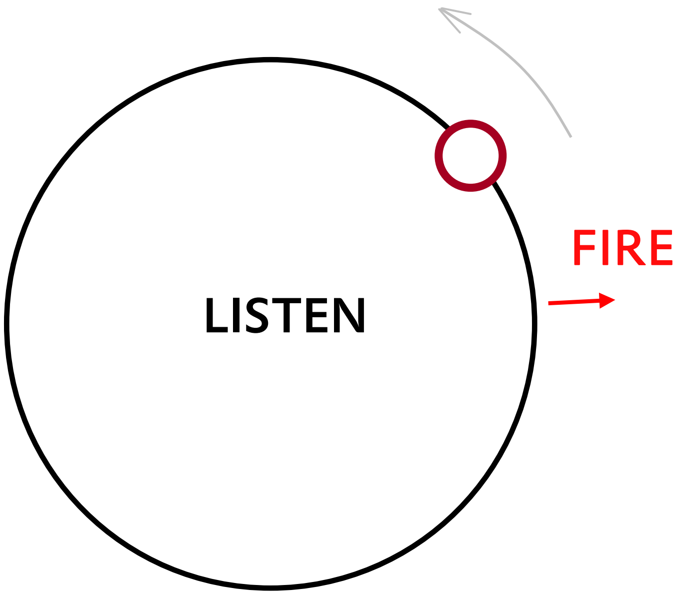
Values in ms

A: no influence
B: delay in flashing
C: earlier 2nd next flash

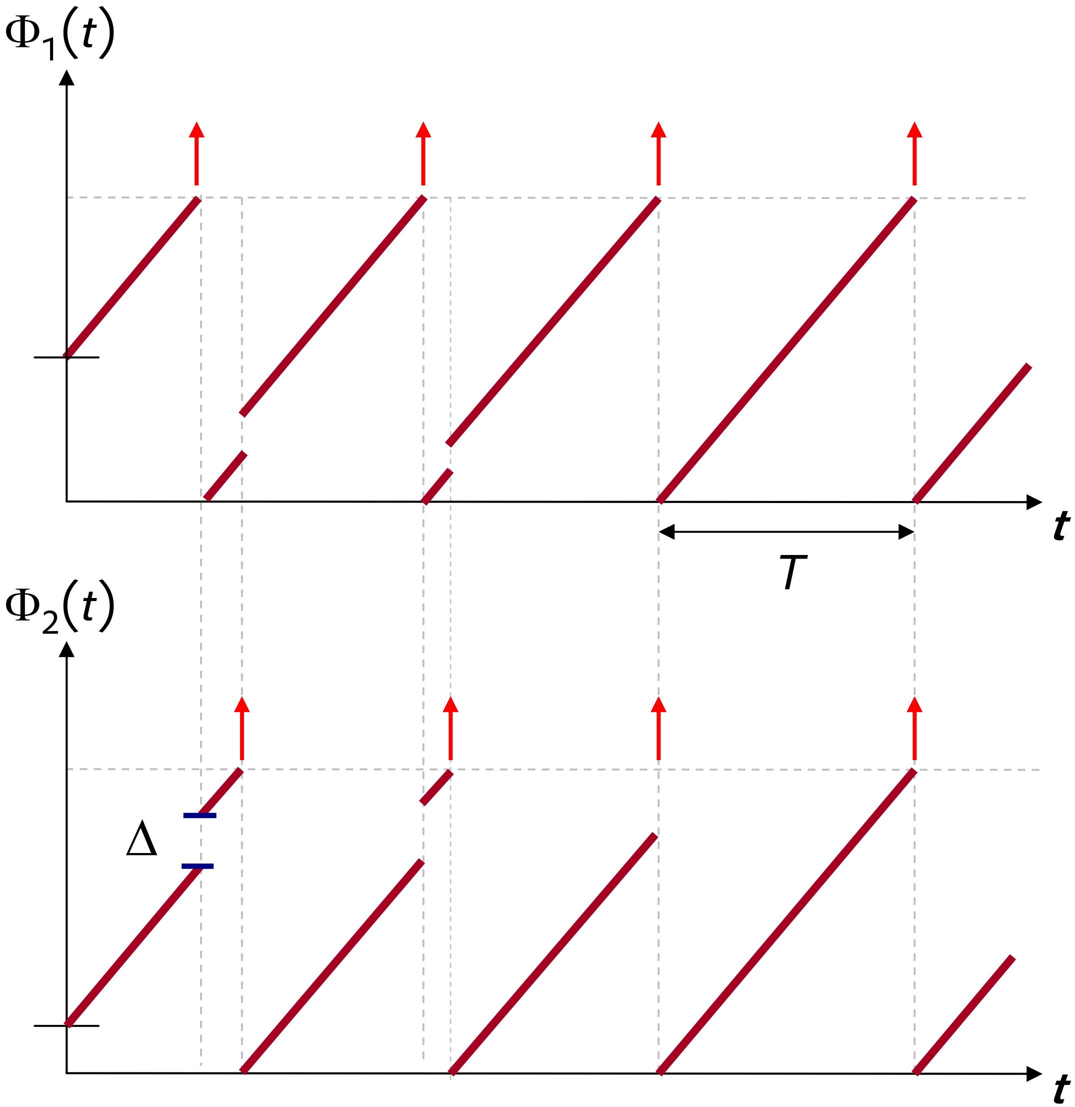
J. Buck *et al.*: Control of Flashing in Fireflies V. *Journal of Comparative Physiology A*, 144:630–633, 1981.

Modeling One Firefly: Integrate-and-Fire Oscillator



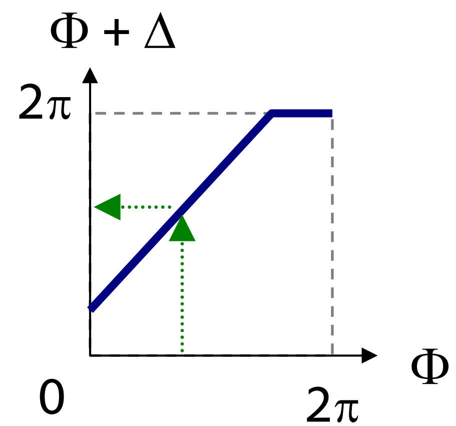


Modeling Two Fireflies: Coupled Integrate-and-Fire Oscillators



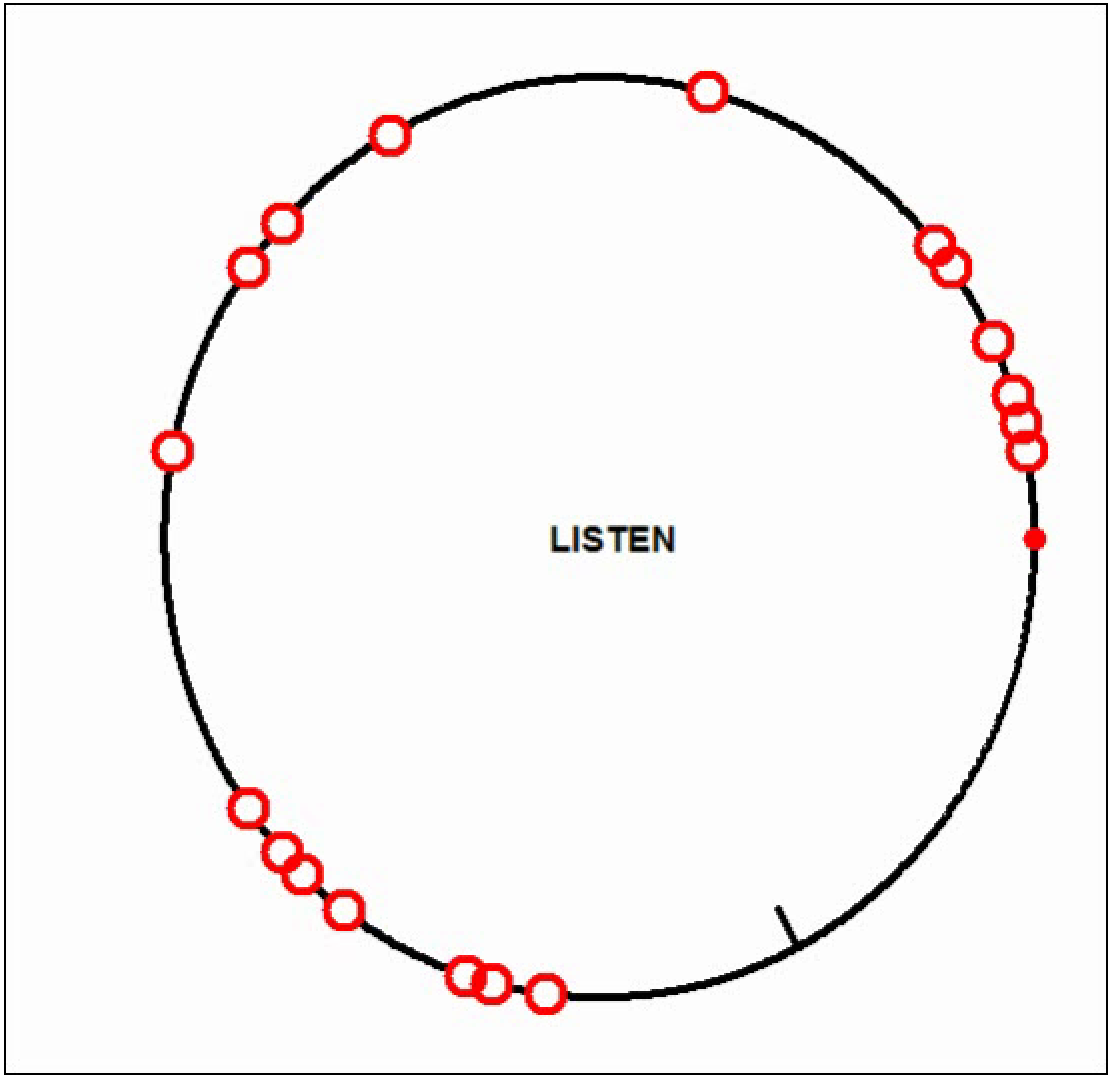
Phase jump upon reception of a pulse:

$$\Phi \to \Phi + \Delta$$



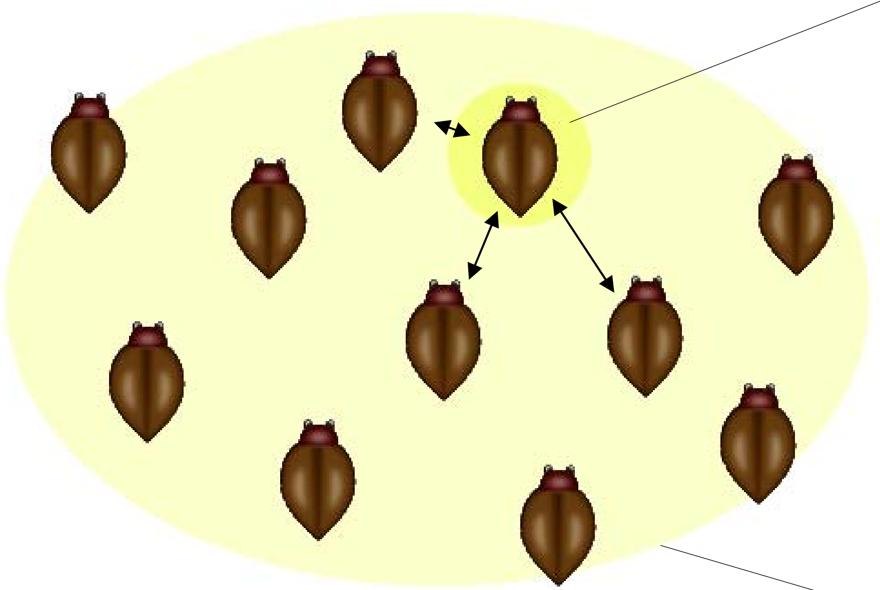
Proven to lead to synchrony

Several Coupled Integrate-and-Fire Oscillators



FIRE

Why is this algorithm appealing?



Individual Entity („Firefly“) *many*

- Simple behavior rules
- Local view
- Distributed operation

Emergence

Entire System (“Swarm”)

- Solves a complex task
- Is adaptive to changes
- Is very scalable

C. Prehofer, C. Bettstetter: Self-Organization in Communication Networks: Principles and Design Paradigms. *IEEE Communications Magazine*, Feature on Advances in Self-Organizing Networks, July 2005.

Applications of coupled oscillator synchronization

- Synchronization of heart cells (Peskin)
- Synchronous firing of neurons
- Formation of earthquakes (Hopfield)
- Forest fires
- Mass extinctions
- Sleep cycles
- Bridge vibrations

Millenium Bridge (London)

Source: Wikimedia Commons

Our Research: Application to Wireless Networks

Problem statement:

Can we apply this algorithm to achieve slot synchronization in ad hoc networks?

Why do we need slot synchronization?

Essential building block for various functions in communication and control systems, e.g.:

- medium access,
- distributed sensing, and
- scheduling of sleep phases.

Joint work with Alexander Tyrrell and Gunther Auer

NTT docomo

DOCOMO Euro-Labs

Can Firefly Synchronization be Applied to Wireless Systems?

Firefly algorithm assumes:

- Synchronization pulses are infinitely short
- No delays
- Nodes listen and transmit at the same time
- All nodes form a fully meshed network

Removing one or more of these assumptions makes synchronization unstable. → Direct transfer to wireless systems is **infeasible.**

Example: With delays, nodes may receive "echos" of their own fire pulse.

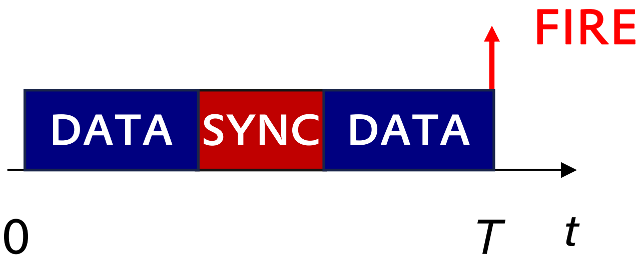
A. Tyrrell, G. Auer, C. Bettstetter: Biologically Inspired Synchronization for Wireless Networks. In *Advances in Biologically Inspired Information Systems: Models, Methods, and Tools*, Series: *Studies in Computational Intelligence*, Springer, 2007.

Meshed Emergent Firefly Synchronization (MEMFIS) (1/4)

Solution taking into account the **technological constraints** of wireless systems while maintaining nice properties of firefly sync.

Key design characteristics:

- A common **synchronization word** is **embedded** in each payload packet.
- This synchronization word is detected at the receiver using a cross-correlator.
- **Delays** are handled by enhancing the synchronization algorithm.

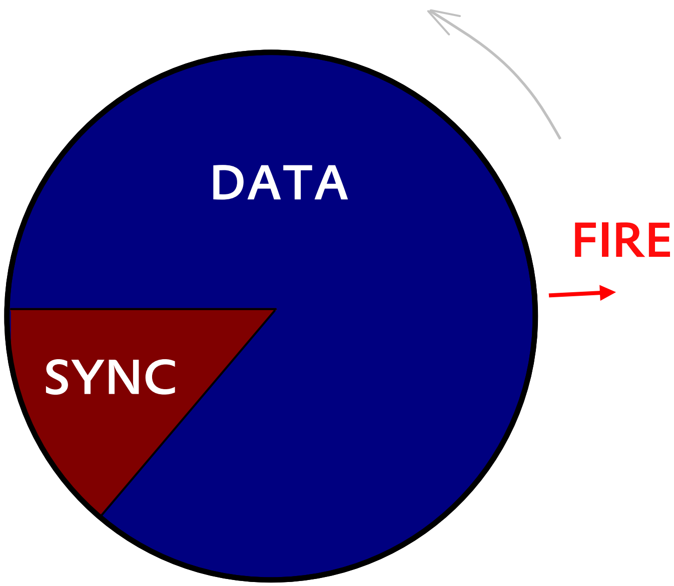


Result: Synchronization **emerges** gradually as nodes exchange packets randomly. No dedicated synchronization phase needed.

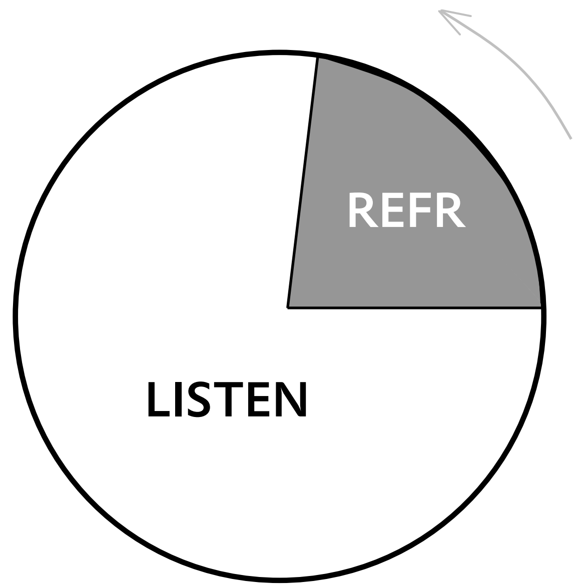
A. Tyrrell, G. Auer, C. Bettstetter: Emergent Slot Synchronization in Wireless Networks. *IEEE Transactions on Mobile Computing*, to appear.

Meshed Emergent Firefly Synchronization (MEMFIS) (2/4)

Node in transmit modus:



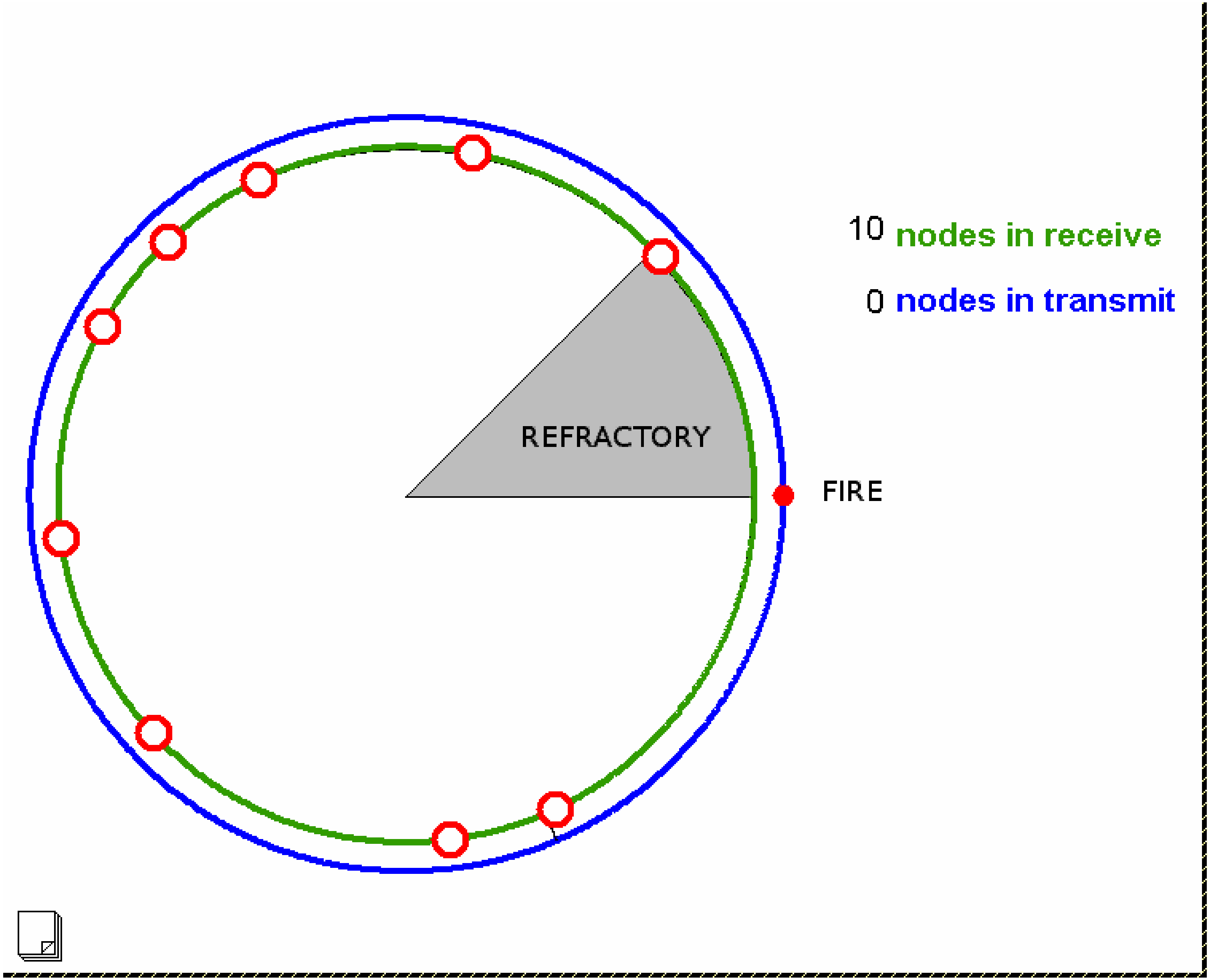
Node in receive modus:



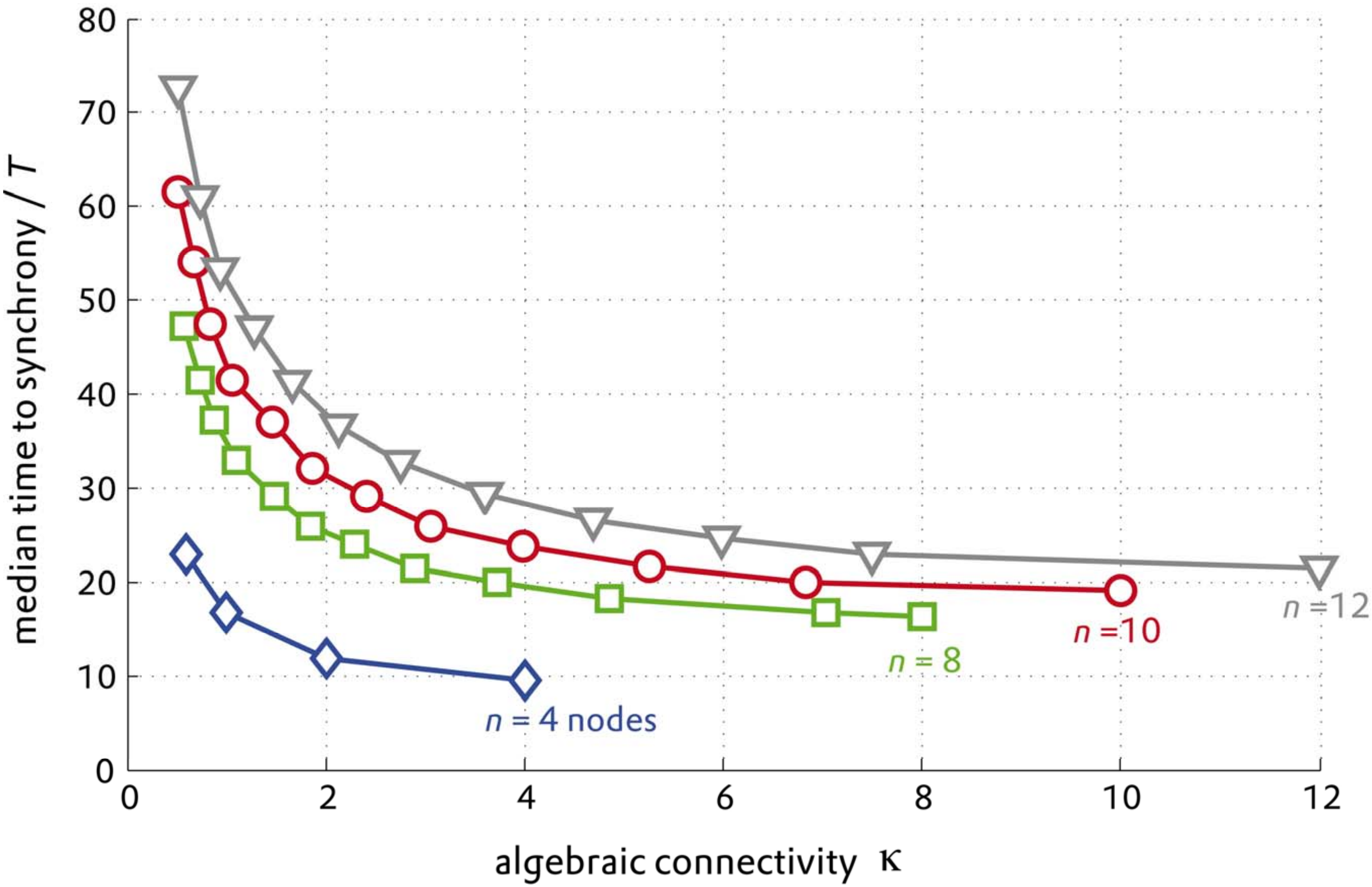
During REFRACTORY period received SYNC words are ignored.

A. Tyrrell, G. Auer, C. Bettstetter: Emergent Slot Synchronization in Wireless Networks. *IEEE Transactions on Mobile Computing*, to appear.

Meshed Emergent Firefly Synchronization (MEMFIS) (3/4)



Meshed Emergent Firefly Synchronization (MEMFIS) (4/4)



Selected Other Contributions

MEMFIS

- Impact of **false alarm and missed detections** on synchronization
- Analysis of the algorithm **convergence**

General topics

- Impact of **frequency drifts** on self-organizing synchronization
- Synchronization with **negative phase jumps**
  (achieved & proved convergence to global synchrony)

Max-Planck-Institut
für Dynamik und Selbstorganisation

A. Tyrrell, G. Auer, C. Bettstetter: A Synchronization Metric for Meshed Networks of Pulse-Coupled Oscillators.
In *Proc. Intern. Conf. Bio-Inspired Models of Network, Information, and Comp. Sys. (BIONETICS),* Hyogo, Japan, Nov 2008.

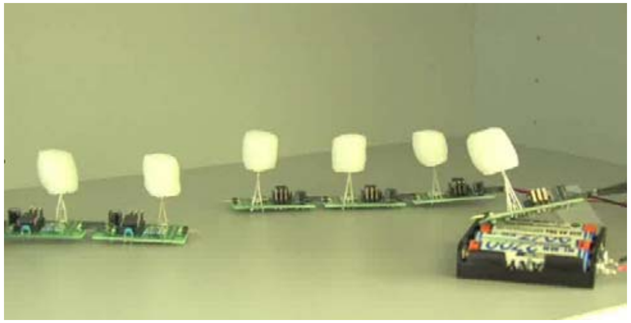
A. Tyrrell, G. Auer, C. Bettstetter: On the Accuracy of Firefly Synchronization with Delays. Best paper award.
In *Proc. Intern. Symp. on Applied Sciences in Biomed. Commun. Techn. (ISABEL),* Aalborg, Denmark, Oct 2008.

J. Klinglmayr, C. Bettstetter, M. Timme. Globally Stable Synchronization by Inhibitory Pulse Coupling. Invited paper.
In *Proc. Intern. Symp. on Applied Sciences in Biomed. Commun. Techn. (ISABEL),* Bratislava, Slovak Republic, Nov 2009.

Ongoing and Future Work

Research

- Robustness of synchronization against faulty and malicious nodes
- Implementation onto a programmable hardware platform

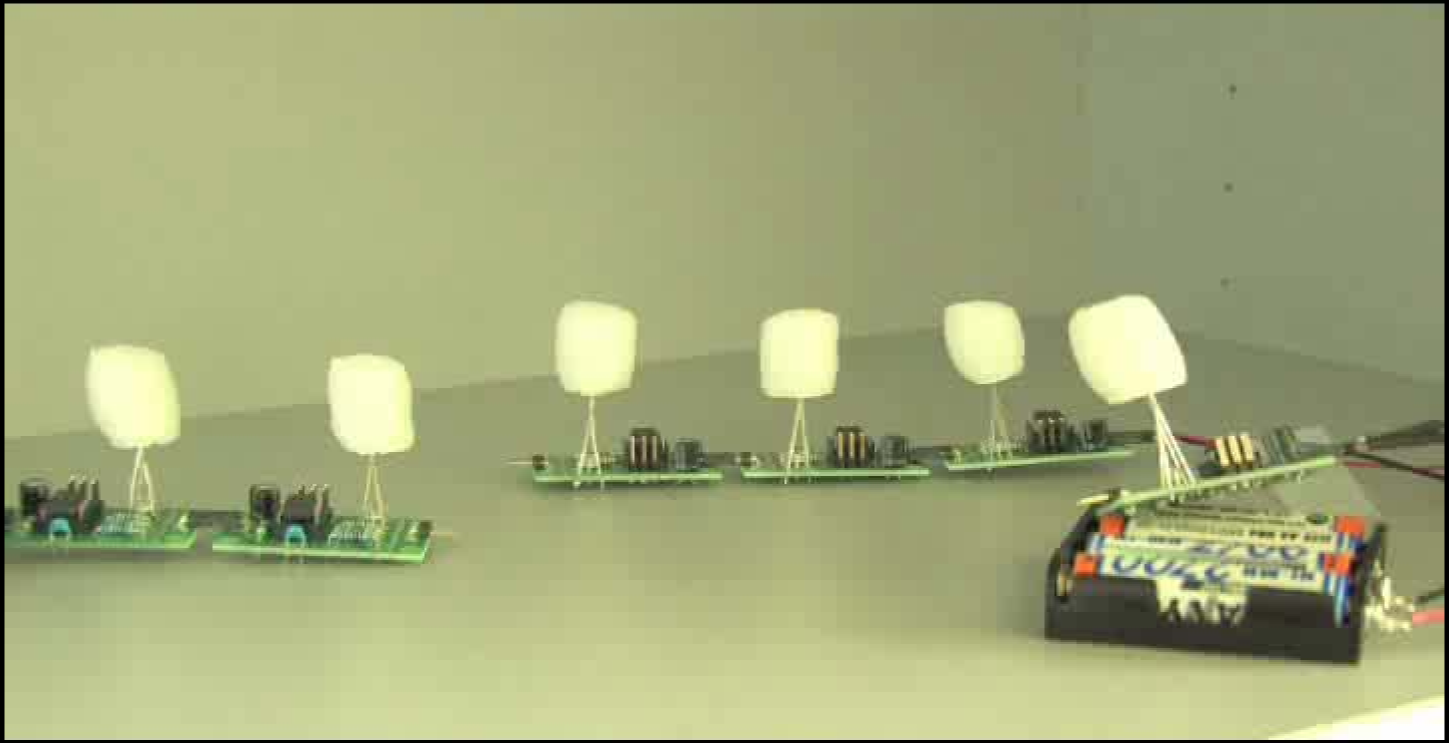


Demo applications

- With light signals (electroflies)
- With audio signals (iPhone app)

A. Tyrrell, G. Auer, C. Bettstetter, R. Naripella: How Does a Faulty Node Disturb Decentralized Slot Synchronization over Wireless Networks? Accepted for *Proc. IEEE Intern. Conf. on Communications (ICC),* Cape Town, South Africa, May 23-27, 2010.

Electroflies



iPhone App “BUZZflies”

iPhone App "BUZZflies"

App Store › Unterhaltung › Cam Lai Ngo

BUZZflies

Beschreibung

This application detects sounds from other iPhone devices and synchronizes them utilizing an synchronization algorithm inspired from the nature: the firefly synchronization algorithm.

Website von Cam Lai Ngo › **BUZZflies Support ›**

...Mehr

Gratis-App

Kategorie: Unterhaltung
Erschienen: 06. Januar 2010
Version: 1.0
4.8 MB
Sprachen: Englisch
Verkäufer: Cam Lai Ngo
© Cam Lai Ngo

Kennzeichnung: 4+

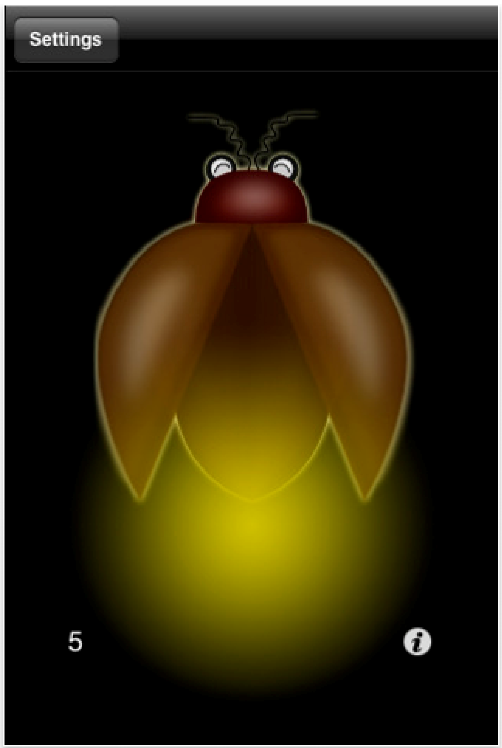
Voraussetzungen: Kompatibel mit iPhone. Erfordert iPhone OS 3.0 oder neuer.

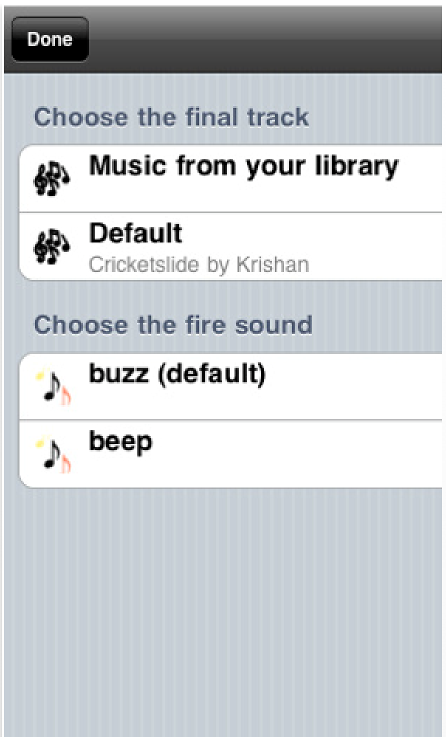
Kundenbewertun...

Bewerten

Wir haben noch nicht genügend Bewertungen erhalten, um einen Durchschnittswert für die aktuelle Version von diesem/dieser application anzeigen zu können.

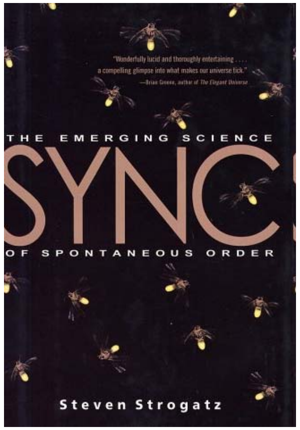
Screenshots

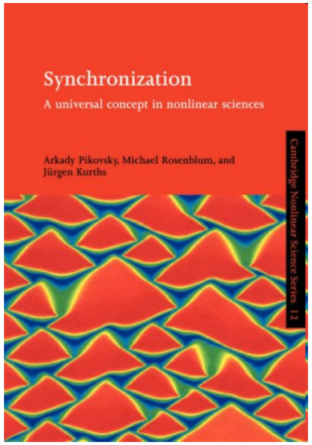




Literature: Synchronization

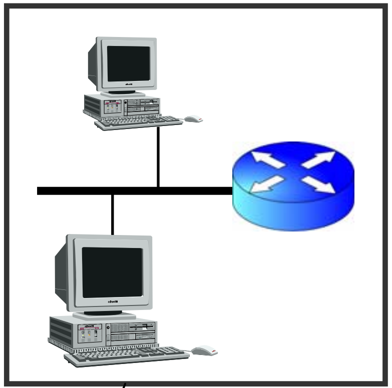
- J. Buck, E. Buck: Synchronous Fireflies. *Scientific American*, May 1976.
- S. H. Strogatz, I. Stewart: Coupled Oscillators and Biological Synchronization. *Scientific American*, Dec 1993.
- S. H. Strogatz: *SYNC: The emerging science of spontaneous order*, Hyperion, 2003.
- A. Pikovsky, M Rosenblum, J. Kurths: *Synchronization: A Universal Concept in Nonlinear Sciences*, Cambridge University Press, 2001.
- S. Bregni: *Synchronization of Digital Telecommunication Networks*, Wiley, 2002.
- A. Tyrrell, G. Auer, C. Bettstetter. Emergent Slot Synchronization in Wireless Networks. To appear in *IEEE Transactions on Mobile Computing*. Preprint available.



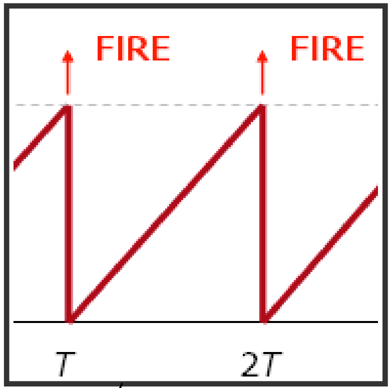


Outline

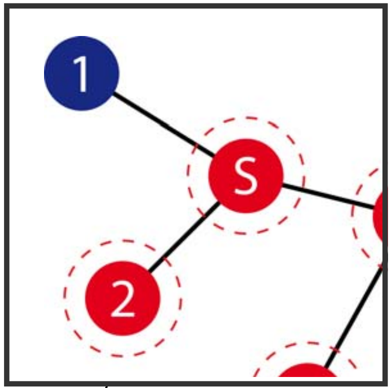
Motivation



Self-Configuration in the Internet



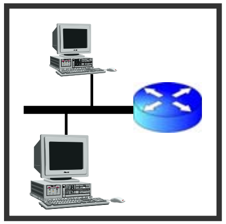
Synchronization in Wireless Networks

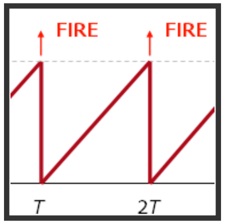


Conclusions and Open Issues

Summary and Conclusions

- Trend toward **self-organization** at various layers of communication networks.
- Several state-of-the-art technologies for **self-configuration** in Internet-based networks are available, e.g.:
  - Self-configuration of addresses
  - Service discovery
- Self-organizing **synchronization** in wireless networks is a research topic.
  - Solution based on the theory of pulse-coupled oscillators
  - Taking into account characteristics and capabilities of radio communications





Discussion Issues

- To what extend can **today's systems** be **replaced or complemented** by self-organizing systems, taking into account
  - constraints and acceptance of the technology and
  - risks for users?

- How to **design and engineer** technical self-organizing systems?
  - Are traditional approaches for system and software engineering suited?
  - What are building blocks or paradigms for the design?

These are difficult questions ...

Some Design Paradigms for Self-Organization in Networks

1. Find local behavior rules that lead to a desired global behavior
   - Adopt a reference design, e.g., from biology
   - Use trial-and-error
   - Employ evolutionary algorithms and heuristic search

2. Minimize long-lived state information
3. Design protocols that adapt to changes
4. Exploit implicit coordination

C. Prehofer, C. Bettstetter: Self-Organization in Communication Networks: Principles and Design Paradigms. *IEEE Communications Magazine*, Feature on Advances in Self-Organizing Networks, July 2005.

Towards a Theory of Self-Organization

- Which **conditions** are necessary for a system to exhibit self-organization? Which conditions prevent it from doing so?
- How can we **quantify** fundamental features of self-organizing systems, such as emergence, adaptability, and robustness?
- Can we **design** components and their local interactions in such a way that a desired global system behavior emerges? If so how?

Literature: SO in Computer and Communication Networks

- A. Sarma, C. Bettstetter, S. Dixit (Eds.): Self-Organization in Communication Networks. In *Technologies for the Wireless Future*, *Vol. 2,* Wiley, June 2006.
- C. Prehofer, C. Bettstetter: Self-Organization in Communication Networks: Principles and Design Paradigms. *IEEE Communications Mag.*, July 2005.
- S. Staab (Ed.): Neurons, Viscose Fluids, Freshwater Polyp Hydra—and Self-Organizing Information Systems, *IEEE Intelligent Systems*, July 2003.
- J. P. Hubaux *et al.*: Towards Self-Organizing Mobile Adhoc Networks: the Terminodes Project," *IEEE Communications Mag.*, January 2001.
- F. Dressler: *Self-Organization in Sensor and Actor Networks*, Wiley, 2007.
- Q. H. Mahmoud (Ed.): *Cognitive Networks: Towards Self-Aware Networks*, Wiley, 2007.