A Self-Aware Clock
and its possible applications
i.e.

Proper measurements as the basis for the Experimental Evaluation and for the Dynamic Adaptation of Critical Systems and Infrastructures

Andrea Bondavalli

Resilient Computing Group

Università degli Studi di Firenze

Work done with Lorenzo Falai and Andrea Ceccarelli

Distributed systems and ICT infrastructures(!)

Pervasive

- Composed by a large population of connected and cooperating nodes,
- highly dynamic (especially in wireless context), Frequent and sudden join or leave the distributed system.
- Highly heterogeneous. Nodes may be very different from one another,
- Use of different hw&sw
- different synchronization mechanisms,
- different devices (e.g. GPS receivers)
- different connections and....

Critical applications and services

Still they are required to

Run critical applications

Deliver reliable services

Be trusted

Survive failures and attacks

An important requirement which derives from the above is the ability to **adapt at runtime** to handle

-user mobility,

-changing user needs,

-varying resources availability and

- system faults.

Time and timeliness

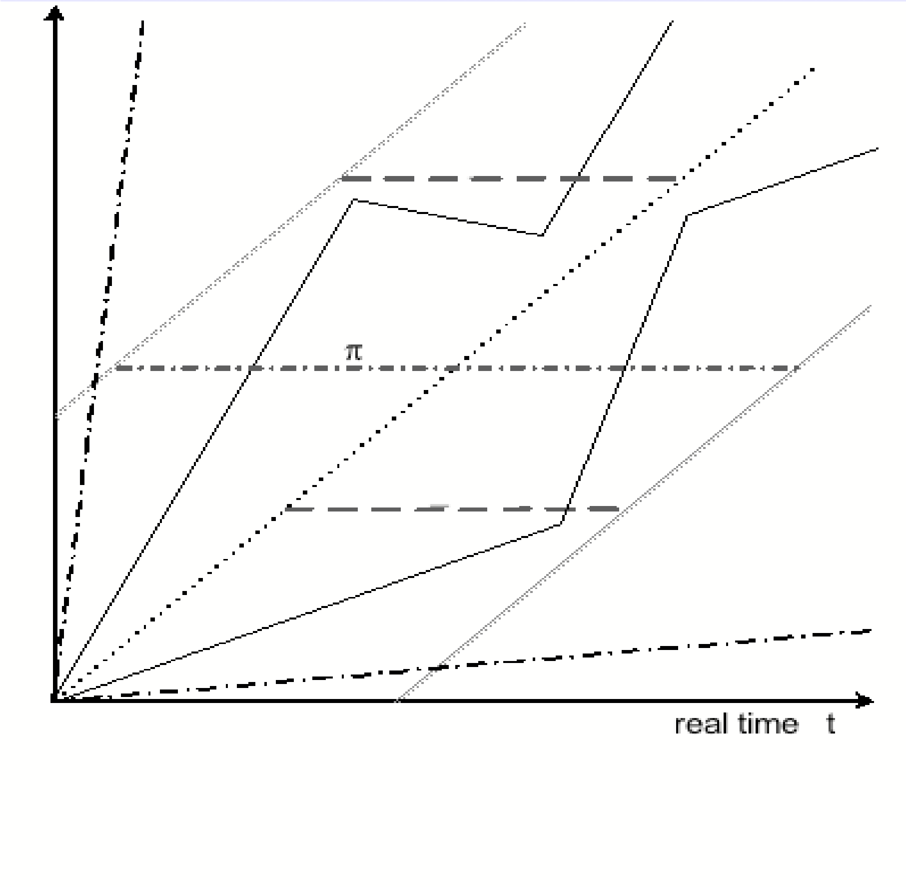
Very likely – if not always- such applications are

-**running in unpredictable or unreliable environments (e.g. Internet-based pervasive systems, or mobile ad-hoc network) and**

-**-have timeliness requirements and make timeliness assumptions which** may be violated, producing

timing failures. **Some applications may occasionally miss some deadlines.**

However the environment does not allow to evaluate a worst-case scenario (with sufficient coverage of the assumptions), or the worst-case obtained is very pessimistic and far from the normal case.



A set of two clocks:
precision *π*,
accuracy *α*,
drift and
clock synchronization;

the outside thick dashed lines represent the bound in the rate of drift,
a fundamental assumption for deterministic clock synchronization, since it allows to predict the maximum deviation after a given time interval.

Clocks and synchronization

Need for time-dependent protocols → clock synchronization algorithms:

from a local virtual clock at each node of the system, communication between nodes allows to execute internal (e.g. TTA), external (e.g. NTP) or hybrid (e.g. CesiumSpray) clock synchronization protocols.

In large scale systems, e.g. Internet, hierarchical or master-based algorithms are preferred, (cooperative protocols difficult due to high number of nodes and distance among them).

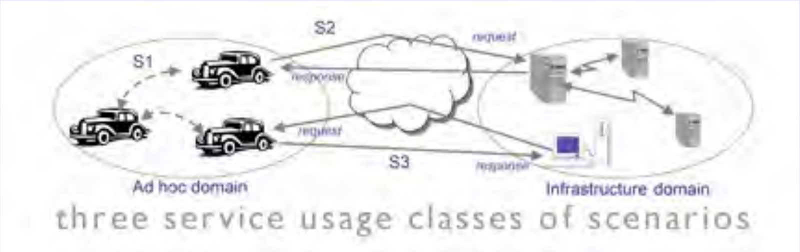
In Wireless Sensor Networks requirements of

- energy efficiency (minimal energy spent synchronizing clocks),
- scalability (large population of sensor nodes),
- robustness (continuous adaptation to network conditions, partitioning)
- and ad hoc deployment (time synchronization must work with no a priori configuration, and no infrastructure available).

Make global time-scales and continuous synchronization not the right choice.

Clocks in HIDENETS - I

HIDENETS: HIghly DEpendable ip-based NETworks and Services www.HIDENETS.aau.dk



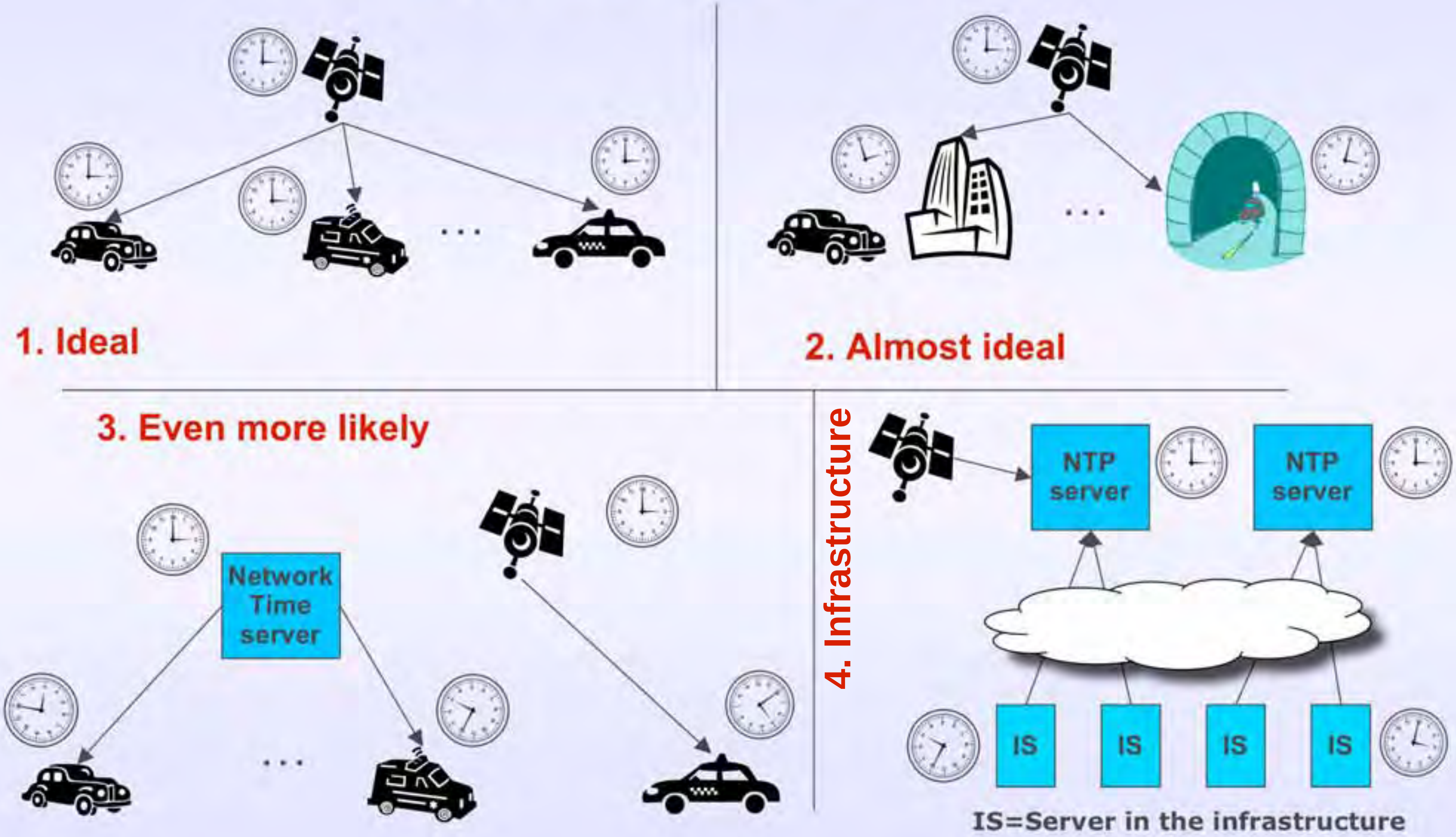
HIDENETS: example of *open* and *mobile* system

New elements can join or exit during execution of the applications (e.g. approaching of new cars, change of the point of connection with infrastructure, …)

Internal Synchronization is not feasible for such open system

We need External Synchronization, that allows to construct a *common*, *external*, *absolute time reference* to which all subsystems must be synchronized → Universal Time Coordinated

Clocks in HIDENETS - II



Synchronization Quality

Consensus on the fact that

The quality of clock synchronization is a variable factor, very hard to predict.

Many causes may be responsible such as

- varying communication delays,
- propagation delays,
- inaccessibility of reference nodes,
- network partitioning,
- node failures,
- or any possible kind of failure in the clock synchronization algorithm or in the clocks itself.

Applications requiring synchronization and having timeliness requirements can take advantage from awareness of the quality of synchronization.

Awareness of the quality of synchronization

A light, highly-portable, low intrusive oracle of the quality of clock synchronization.

It allows to keep the nodes of a network aware about the quality of synchronization node-to-node and node-to-time reference.

Such self-aware clock (called SAClock) can help in many contexts such as WSN, Vehicular Ad hoc NETworks (VANETs), Car-to-Car (C2C) applications, etc. both for

i) elastic/adaptive applications (real-time mission critical applications, soft real-time applications), which are able to relax timeliness requirements and continue to work properly in a degraded situation, and for

ii) non time-elastic applications

SA Clock - definition

Self-Aware Clock

- Abstraction of the local clock
- Main task: monitoring the synchronization level of the local clock with respect to a global time reference (UTC, Universal Time Coordinated)
- It hides to applications the existence of the synchronization mechanism that it is monitoring: it just shows to the applications the current time value (*LocalClockTime*) and the current accuracy of the provided information

When asked, SA Clock provides <localCT, minUTC, maxUTC>

- This information is such that the actual real-time (*ActualUTC*) is within the interval:
  - **[*minUTC*; *maxUTC*]**
- and *localCT* is the most probable value of the actual real-time

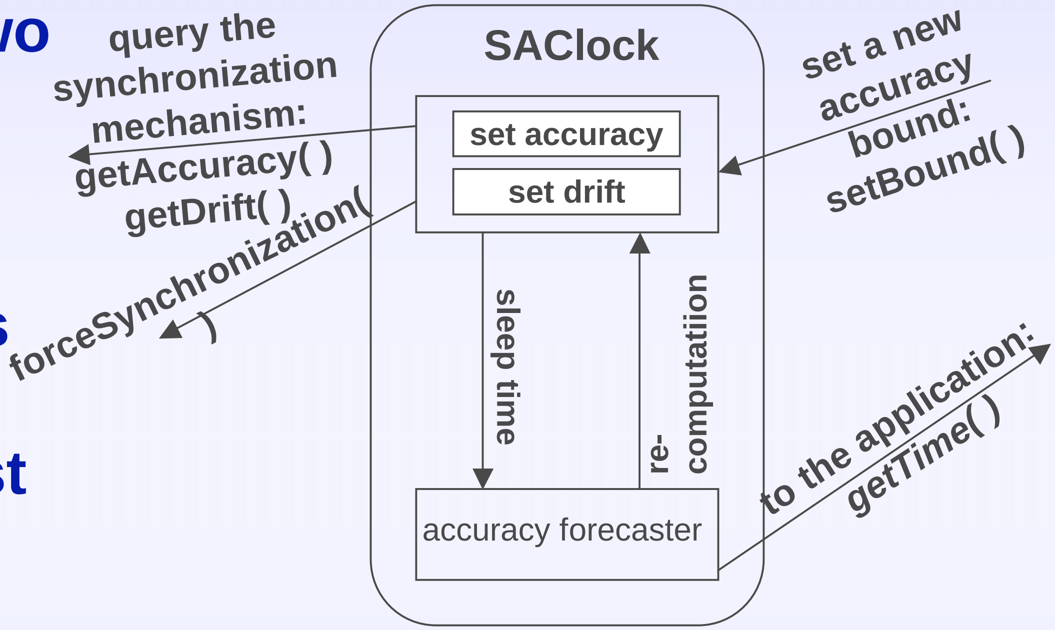
minUTC and maxUTC are usually related to the current estimation of the actual *accuracy* of the local clock with regards to the global time reference, as follows:

- minUTC = localCT - accuracy
- maxUTC = localCT + accuracy

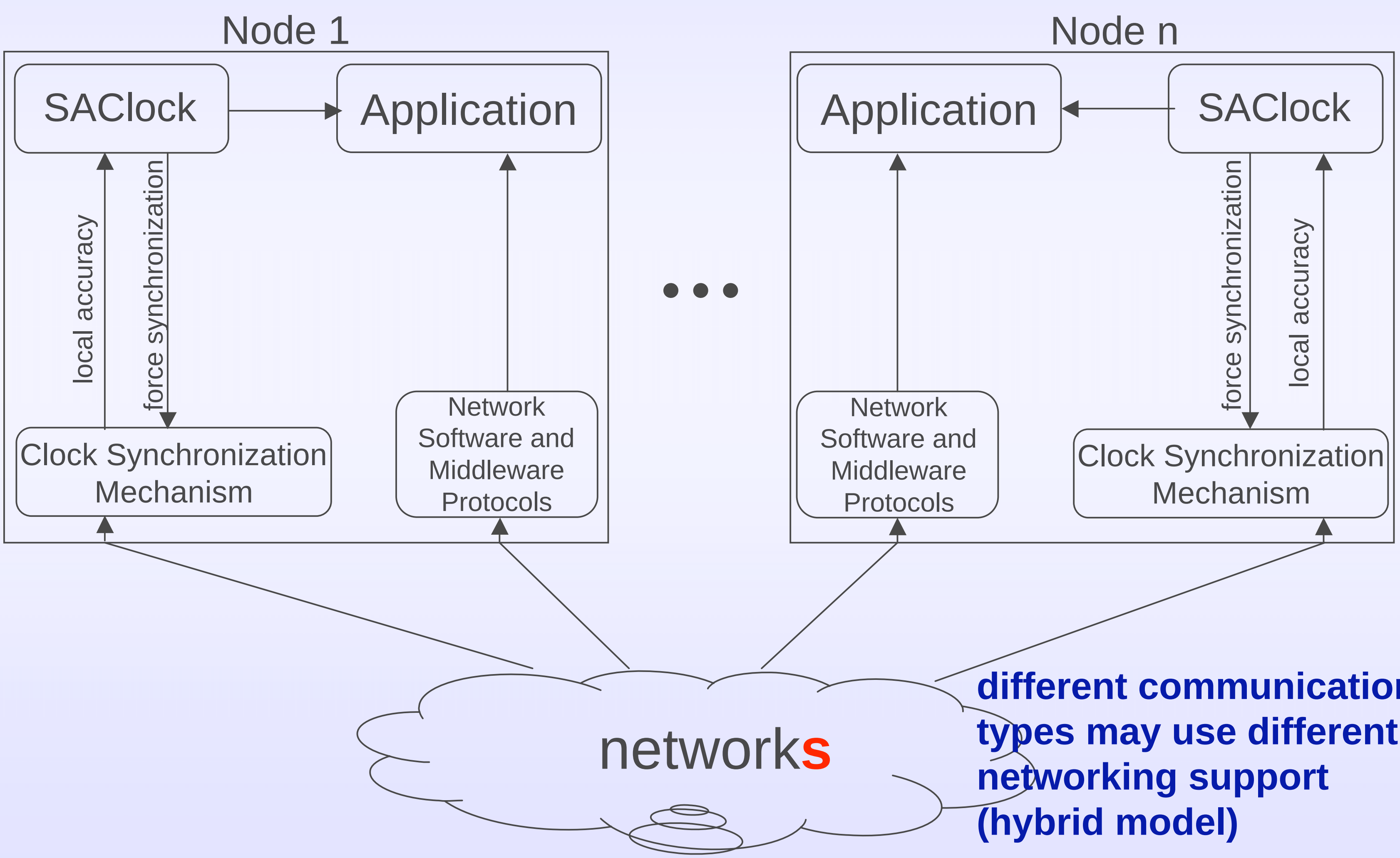
An high level overview of SAClock

SAclock is composed of two parts:

i) an interface, which is common to all applications and processes
to guarantee portability just allowing to interact
with the SAClock, and

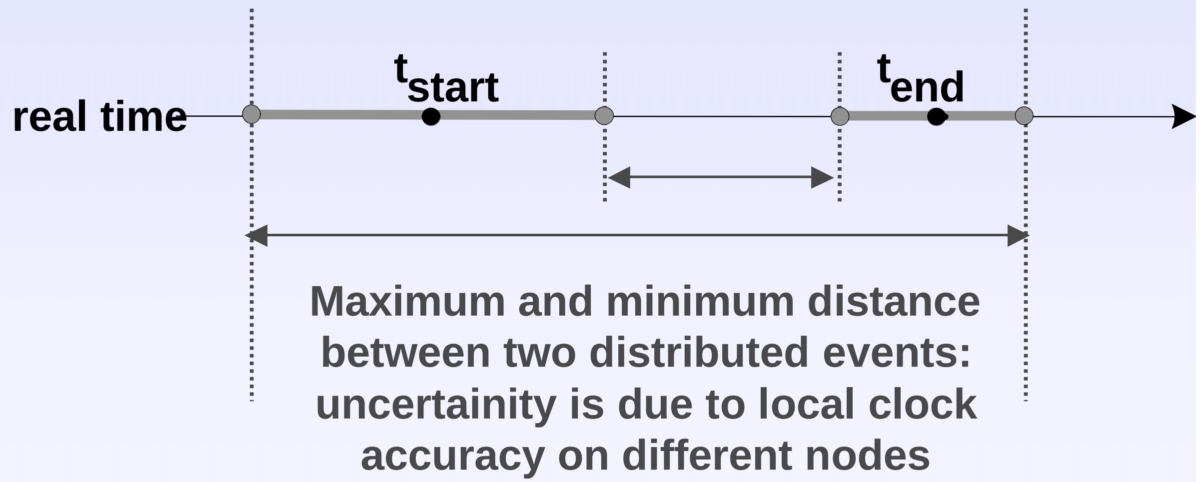


ii) An internal level, which contains the implementation of the SAClock. This second part of the service may need to be re-implemented to adapt to different synchronization mechanisms, hardware, operating systems and networks.
(we have an implementation for Linux and NTP)



different communication types may use different networking support (hybrid model)

Usage: Precision estimation

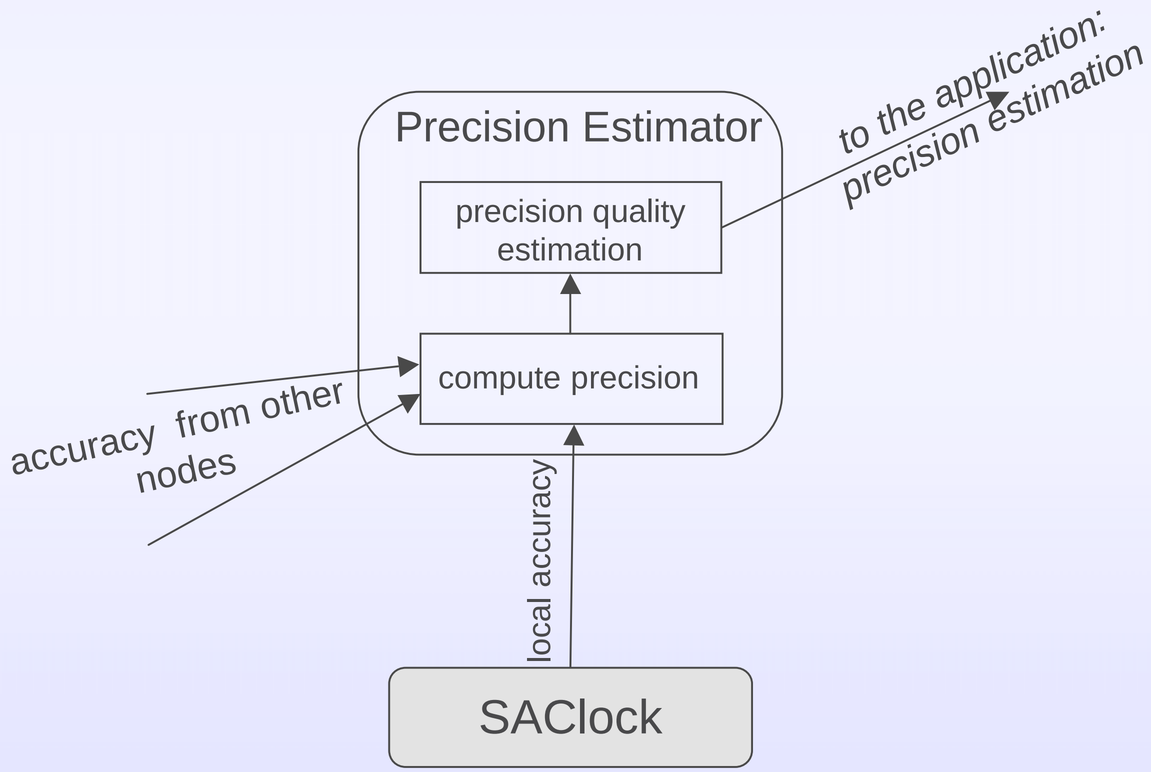


Two events (on two different nodes) happen: uncertainty in inserting the events on the time line strictly related to the synchronization quality of the local clocks.
Thus, the accuracy of the clocks of the involved nodes is the fundamental parameter to decide about the uncertainty in the time-stamped events.

If every node implements a SAClock, the systems precision (varying over time) can be estimated using the accuracy of the nodes. E.G., system precision can be estimated as the sum of the two worst accuracies of the nodes of the distributed system.

Precision Monitor

Precision monitor: a support mechanism for applications willing to be aware of current system precision



Applications of R&SA Clock In HIDENETS

Directly on Applications

Floating Car Data application

- Timestamp associated to an information obtained through the SA Clock
- In this way, remote hosts that receive this information can estimate an interval of *UTC real-time* in which the information was generated.

Usage in the middleware

Local and Distributed Duration Measurement service

- This service uses the timestamps obtained through SA Clock
- Usage of special timestamp <LCT,MinUTC,MaxUTC> allows to obtain *multiple* estimation of a distributed measure (mean, minimum, maximum values for the measure)

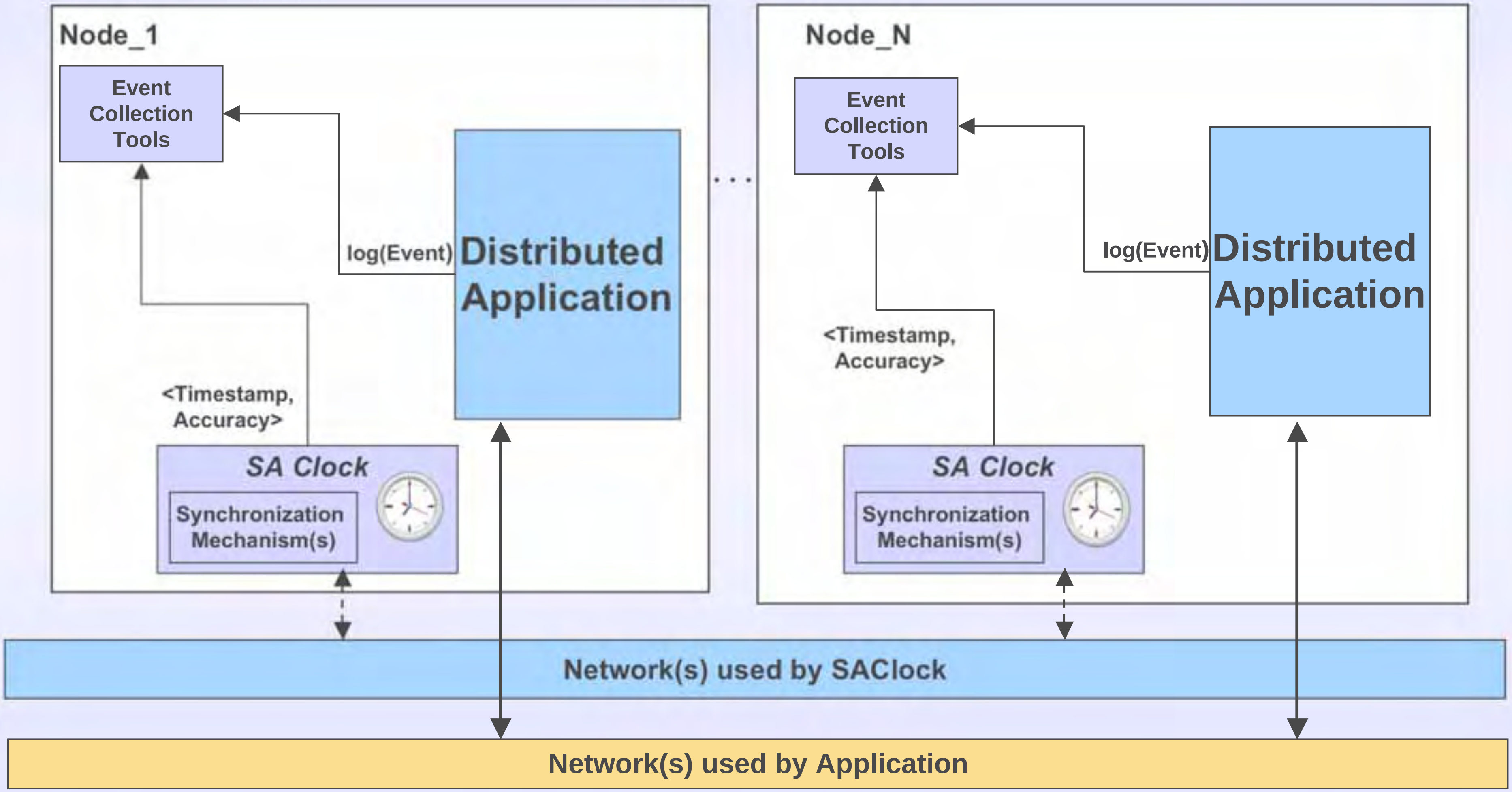
More General Usefulness

The dynamic on-line estimation of the synchronization quality (and its realization through SAClock) is very important and provides the right thing in two different and equally important activities related to distributed systems and pervasive infrastructures;

-A correct and precise vision of the environment behaviour which allows to provide valid data for on line monitoring and dynamic recovery (for e.g. time elastic applications or survivability in hostile unpredicted environments)

A proper and correct evaluation and quantitative validation of QoS and dependability metrics since it is a key element for the definition and set up of appropriate MEASUREMENT instruments (according to metrology foundations uncertainty should always be evaluated to quantitatively state the quality of the measurement performed).

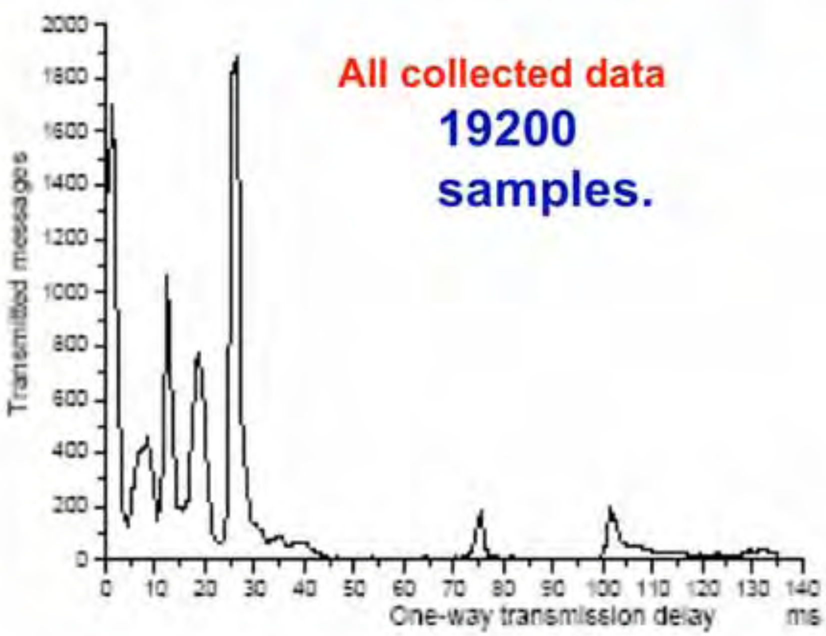
SAClock supporting the experimental evaluation of distributed systems



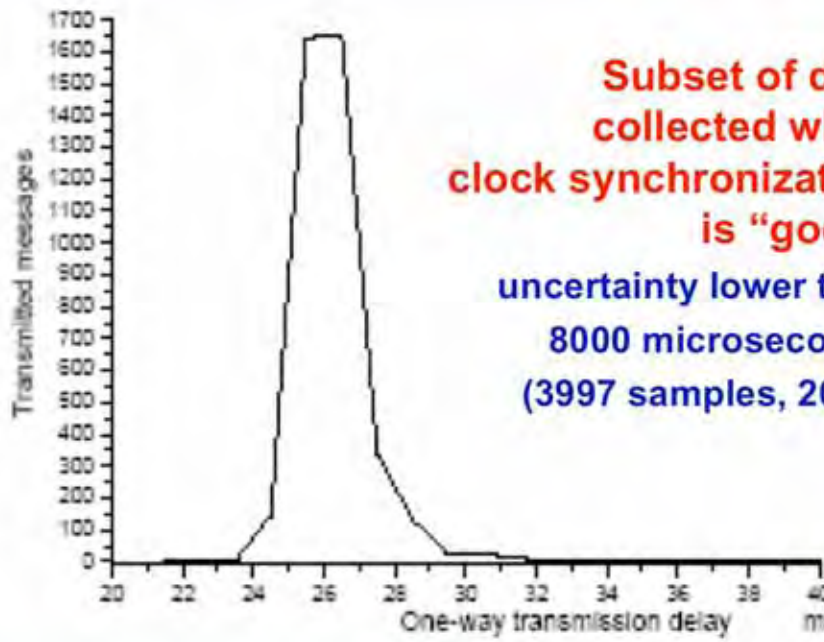
A simple but representative result

Used the previous structure for the
Evaluation of One Way End-to-End transmission delay on a LAN.

A lot of samples collected each with associated accuracy of clock synch.



All collected data
19200 samples.



Subset of data collected while clock synchronization is "good"
uncertainty lower than 8000 microseconds (3997 samples, 20%).