

Fail-Aware Publish/Subscribe

NCA 2007

Zbigniew Jerzak, Christof Fetzer, Robert Fach

Dresden University of Technology

Tuesday 10th July, 2007

{Zbigniew.Jerzak, Christof.Fetzer, Robert.Fach}@inf.tu-dresden.de

State-of-the-art

- ▶ Problems:
 - ▶ we have **conversation**
 - ▶ we need **dissemination**
- ▶ Dissemination:
 - ▶ you are interested in data (not source)
- ▶ Hence loose-coupling:
 - ▶ anonymous
 - ▶ asynchronous

State-of-the-art

- ▶ Problems:
 - ▶ we have **conversation**
 - ▶ we need **dissemination**
- ▶ Dissemination:
 - ▶ you are interested in data (not source)
- ▶ Hence loose-coupling:
 - ▶ anonymous
 - ▶ asynchronous

State-of-the-art

- ▶ Problems:
 - ▶ we have **conversation**
 - ▶ we need **dissemination**
- ▶ Dissemination:
 - ▶ you are interested in data (not source)
- ▶ Hence loose-coupling:
 - ▶ anonymous
 - ▶ asynchronous

Content-Based Systems

- ▶ Peer-to-Peer, Publish/Subscribe, CBN...

- ▶ Content-/Name-driven
- ▶ Loosely-coupled and distributed
- ▶ Failures are imminent [MPHD06]

Content-Based Systems

- ▶ Peer-to-Peer, Publish/Subscribe, CBN...

- ▶ Content-/Name-driven
- ▶ Loosely-coupled and distributed
- ▶ Failures are imminent [MPHD06]

Fail-Awareness

...allows an application to [...] detect when it cannot guarantee all its safety properties and in particular, to detect when to switch to a fail-safe mode [FC03]

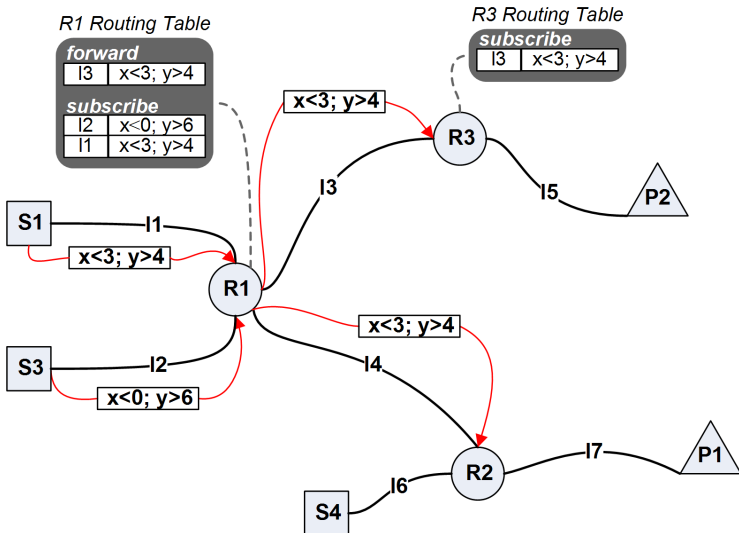
- ▶ Did we receive all messages?
- ▶ Was the received message timely?

Fail-Awareness

...allows an application to [...] detect when it cannot guarantee all its safety properties and in particular, to detect when to switch to a fail-safe mode [FC03]

- ▶ Did we receive all messages?
- ▶ Was the received message timely?

Publish/Subscribe Systems [CRW01]



Time in Publish/Subscribe

- ▶ no clock synchronisation
- ▶ no violation of loose-coupling
- ▶ many-to-many communication

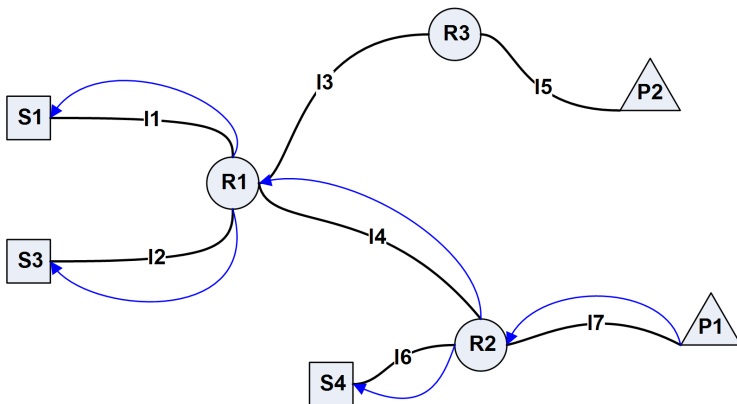
How long did it take for a message to be delivered from p to q ?

Time in Publish/Subscribe

- ▶ no clock synchronisation
- ▶ no violation of loose-coupling
- ▶ many-to-many communication

How long did it take for a message to be delivered from p to q ?

Time in Publish/Subscribe



Idea

- ▶ do not calculate the transmission time
- ▶ calculate upper bound on transmission time

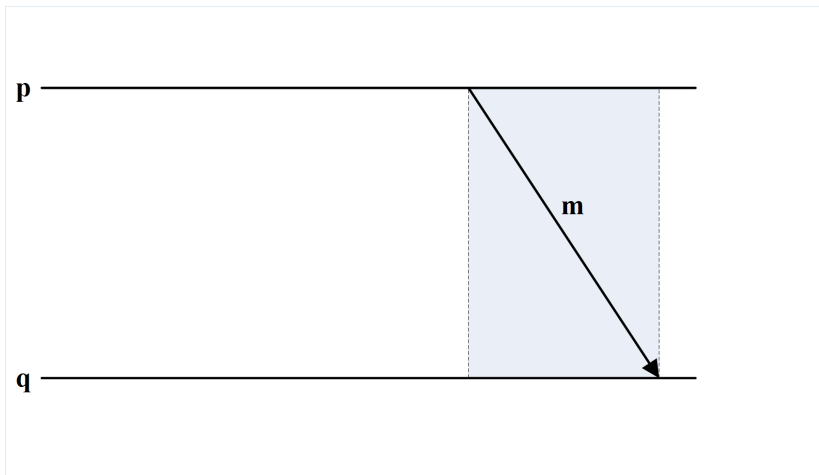
Message transmission time is no longer than the upper bound

Idea

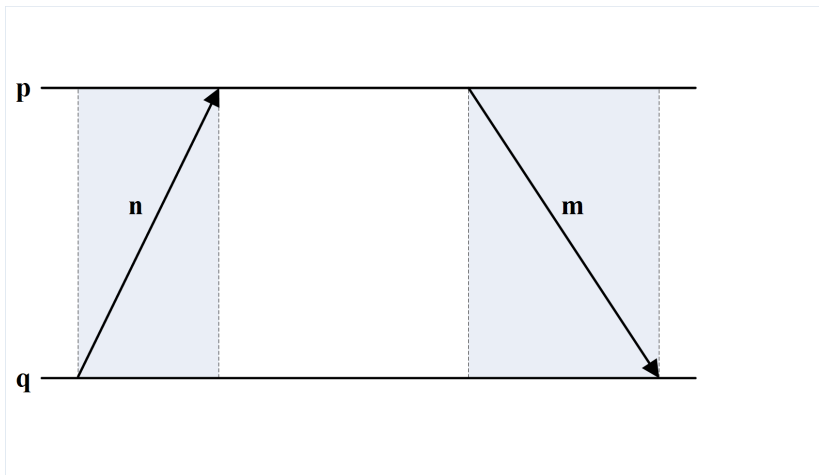
- ▶ do not calculate the transmission time
- ▶ calculate upper bound on transmission time

Message transmission time is no longer than the upper bound

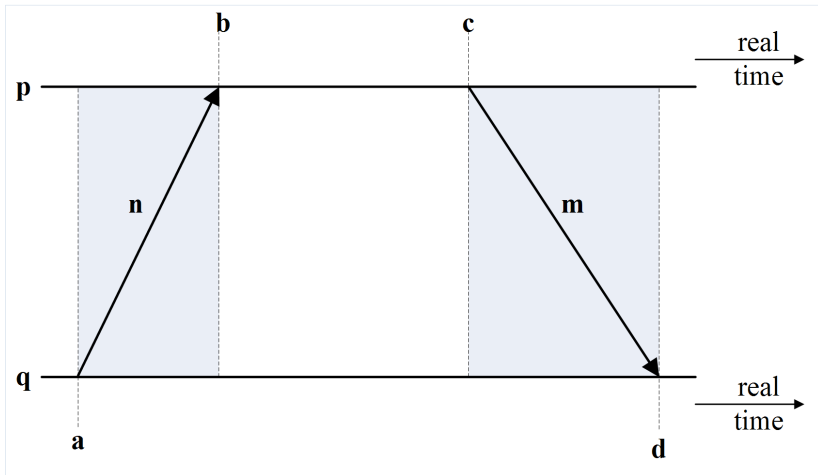
Upper Bound – Point to Point



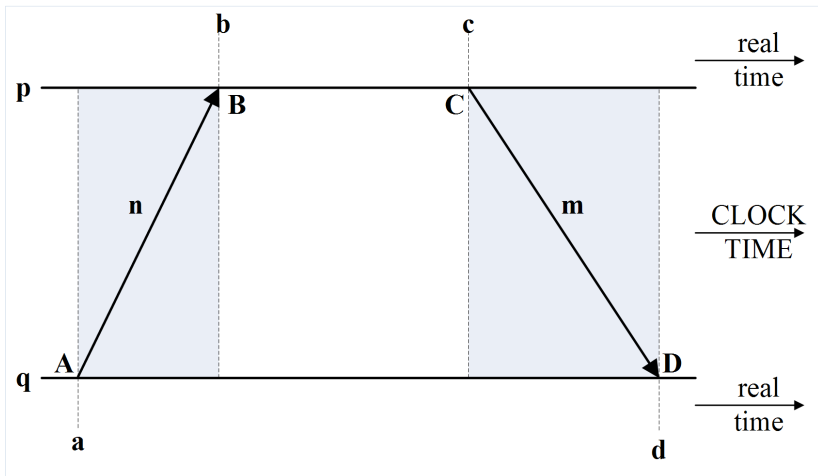
Upper Bound – Point to Point



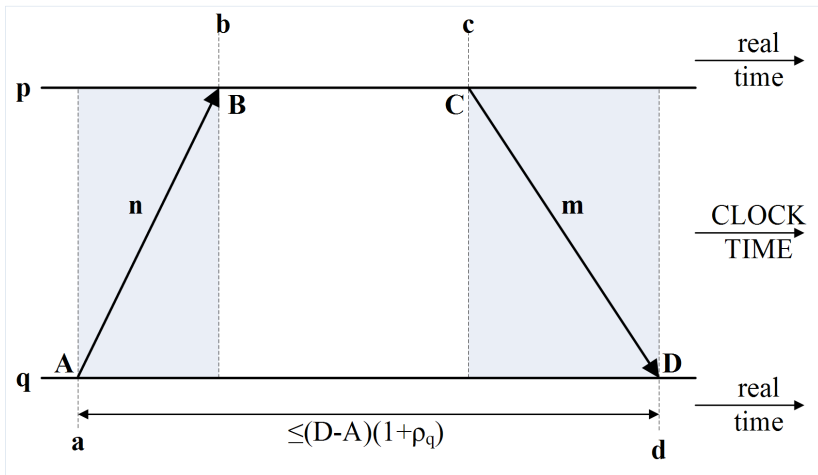
Upper Bound – Point to Point



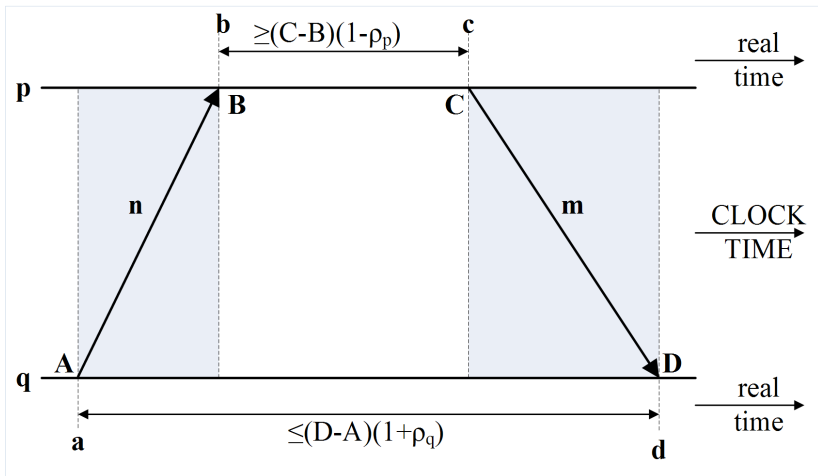
Upper Bound – Point to Point



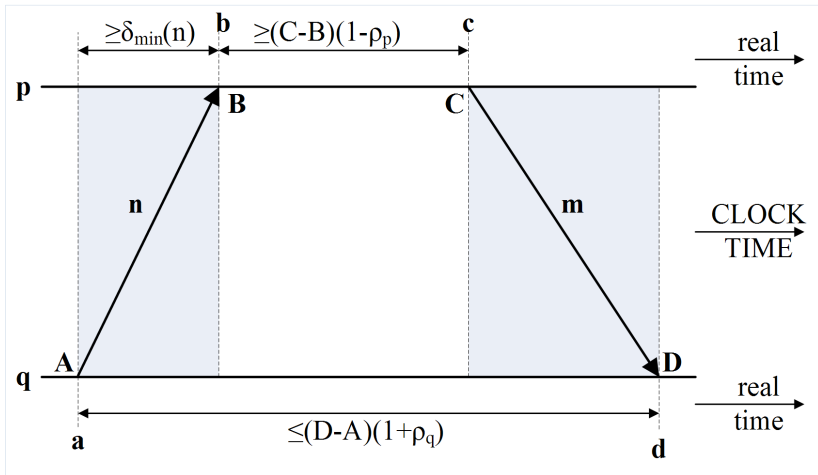
Upper Bound – Point to Point



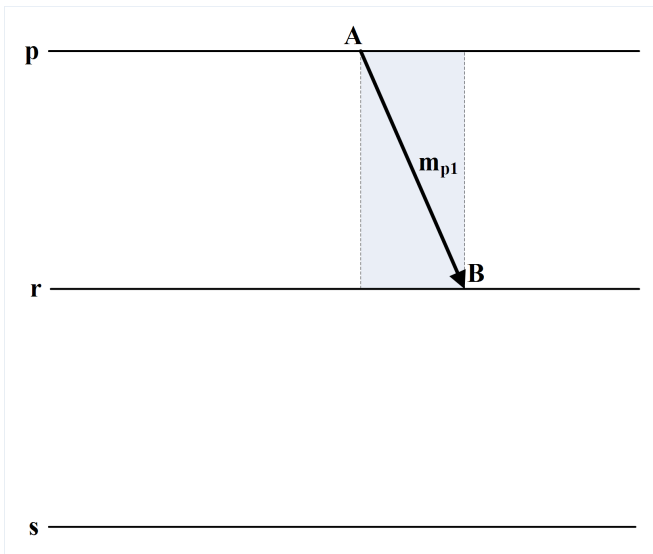
Upper Bound – Point to Point



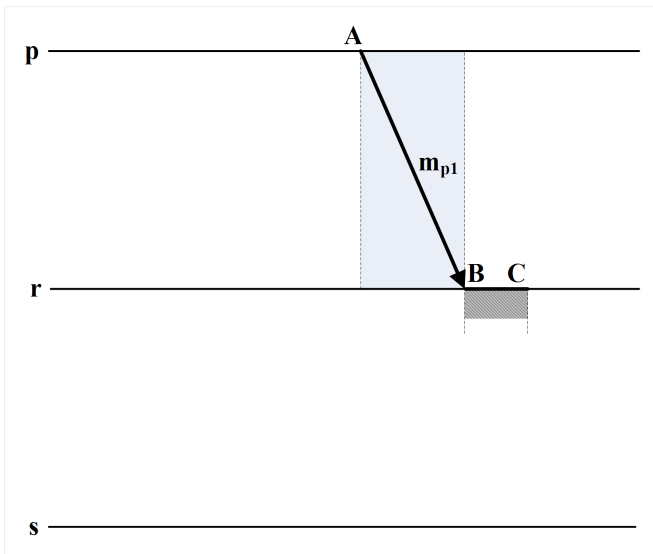
Upper Bound – Point to Point



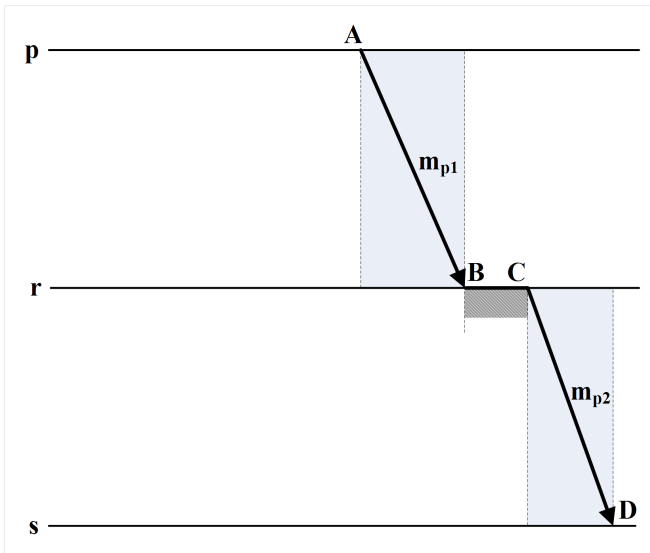
Upper Bound – Publish/Subscribe



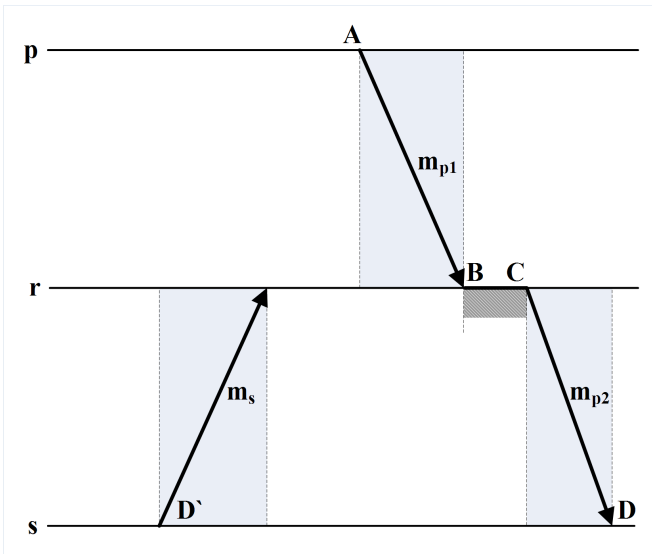
Upper Bound – Publish/Subscribe



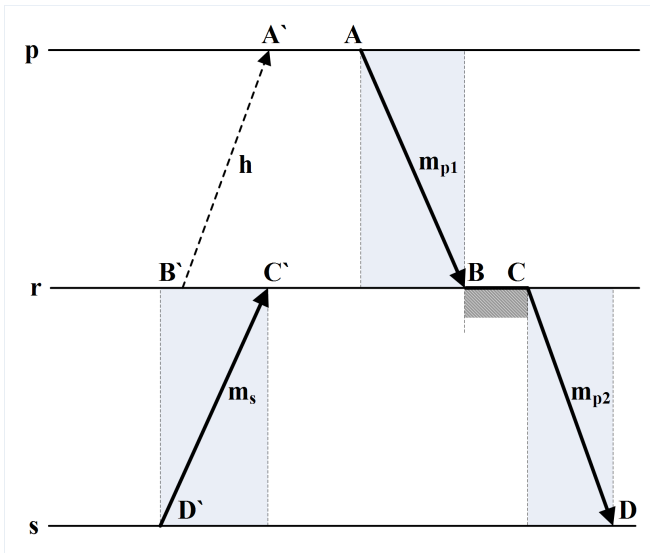
Upper Bound – Publish/Subscribe



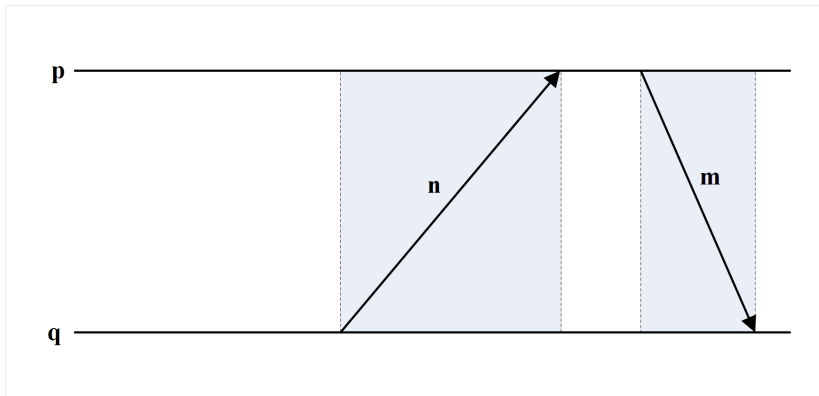
Upper Bound – Publish/Subscribe



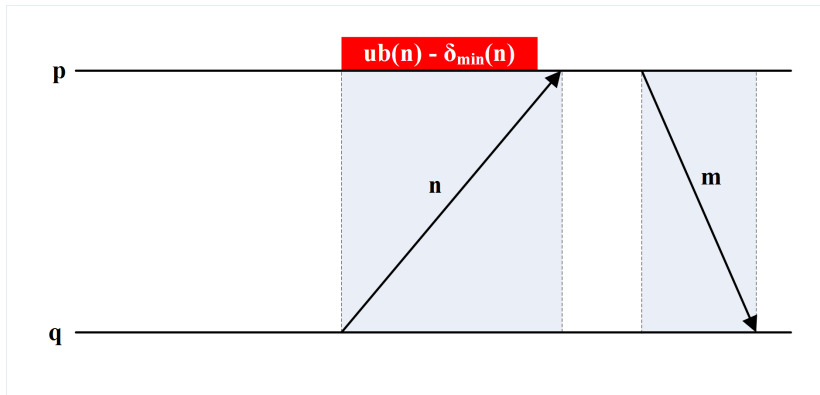
Upper Bound – Publish/Subscribe



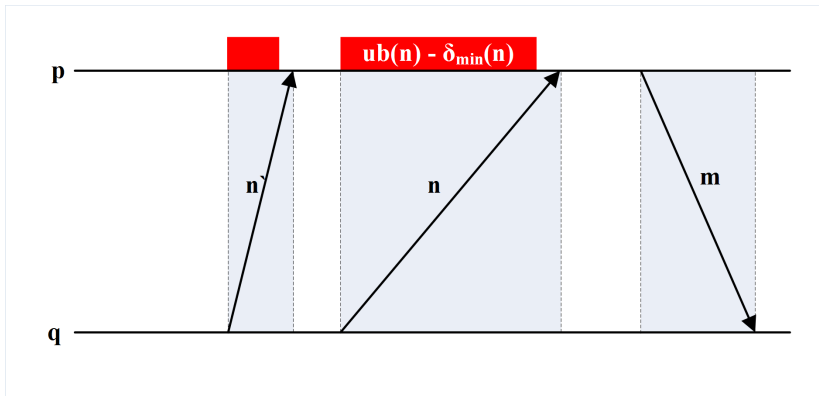
Upper Bound – Optimizations



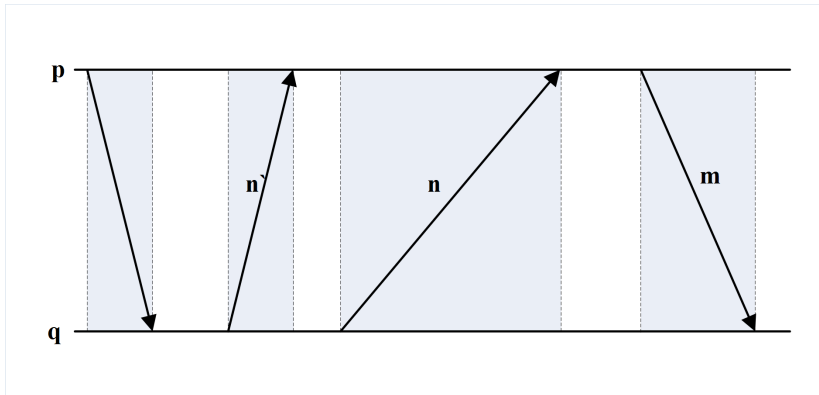
Upper Bound – Optimizations



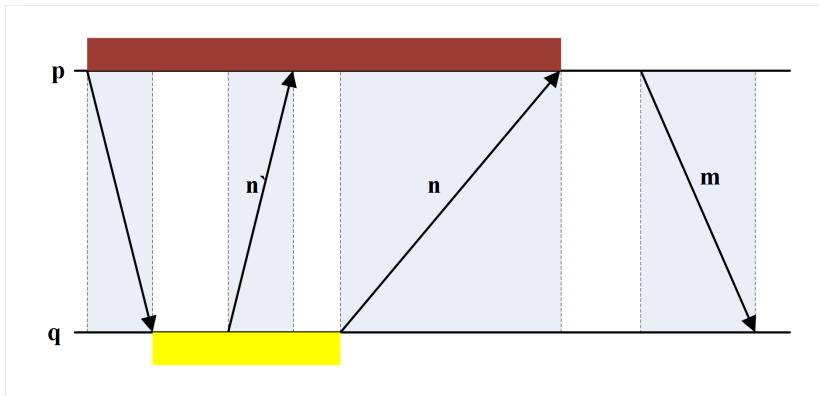
Upper Bound – Optimizations



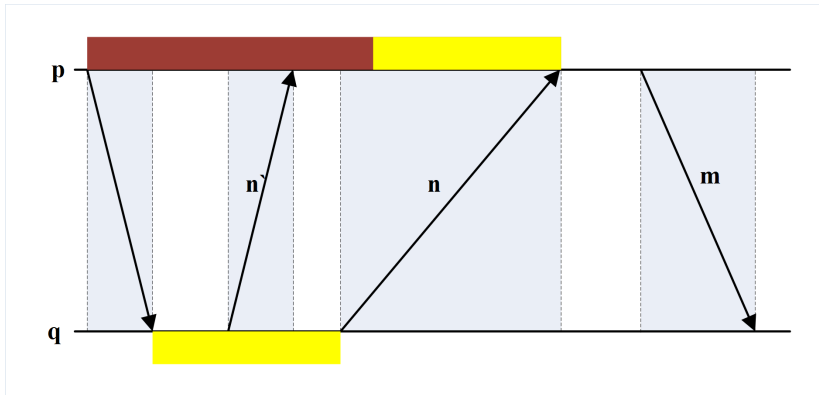
Upper Bound – Optimizations



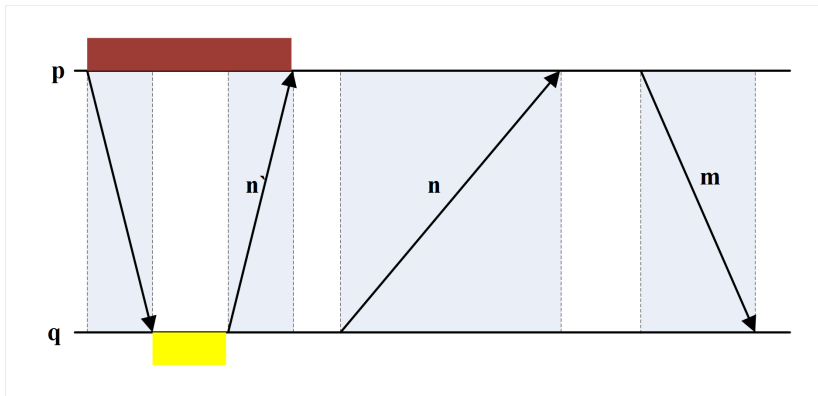
Upper Bound – Optimizations



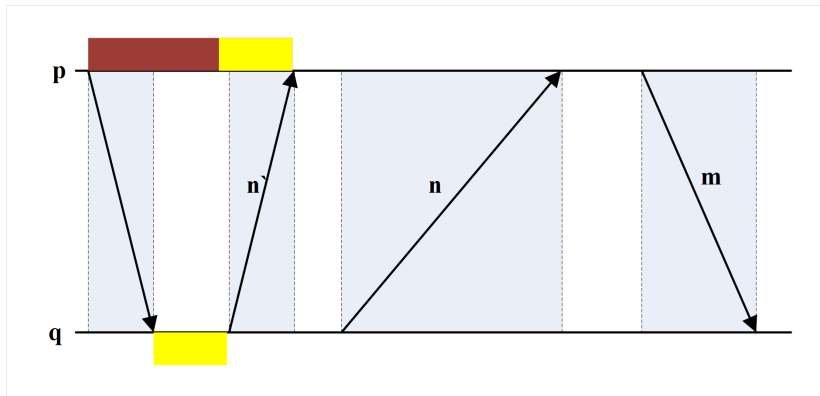
Upper Bound – Optimizations



Upper Bound – Optimizations



Upper Bound – Optimizations



Upper Bound vs NTP

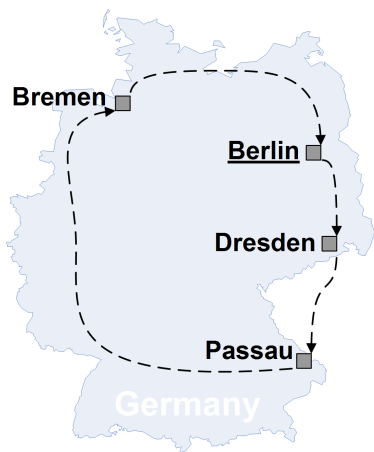
NTP

- ▶ no guarantees whatsoever
- ▶ $2 * \text{max_error}$ gives estimated upper bound

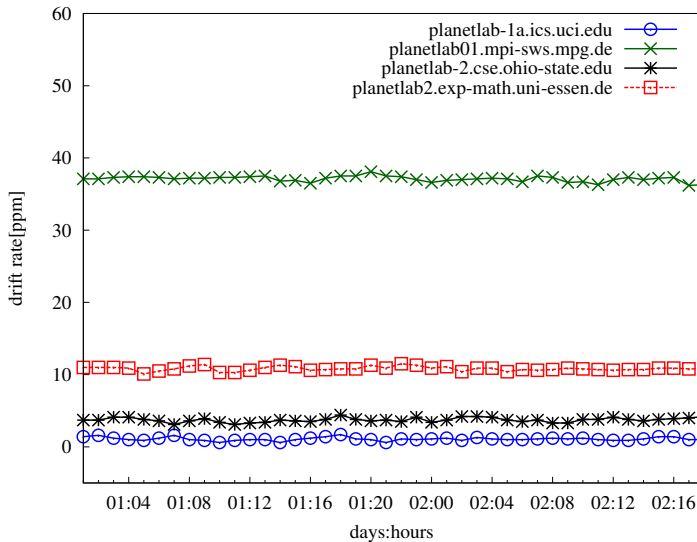
Upper Bound

- ▶ precise
- ▶ more accurate

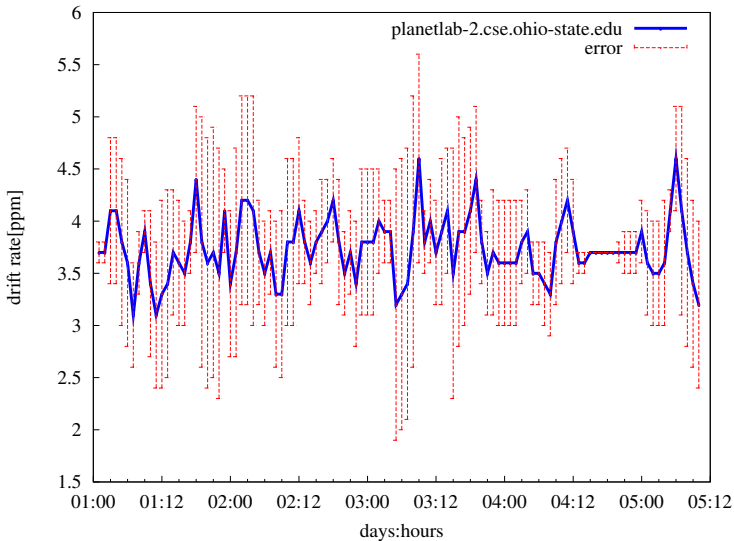
Experiment Setup



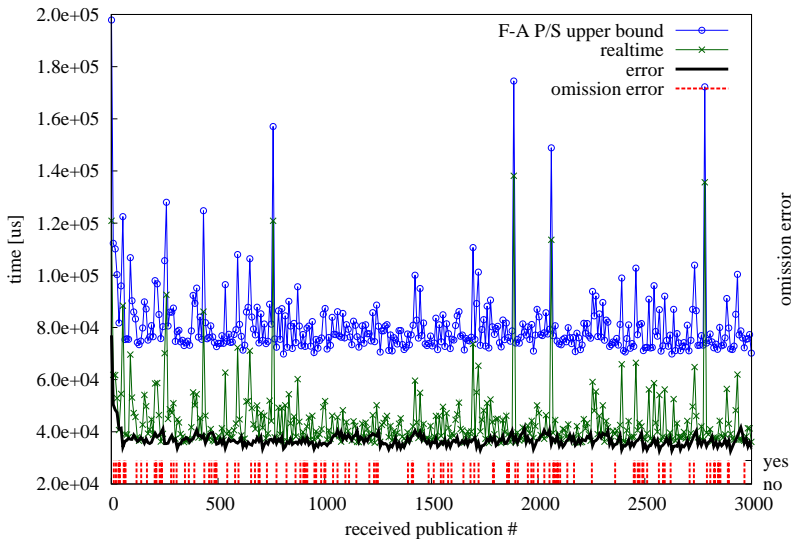
TSC as Time Source



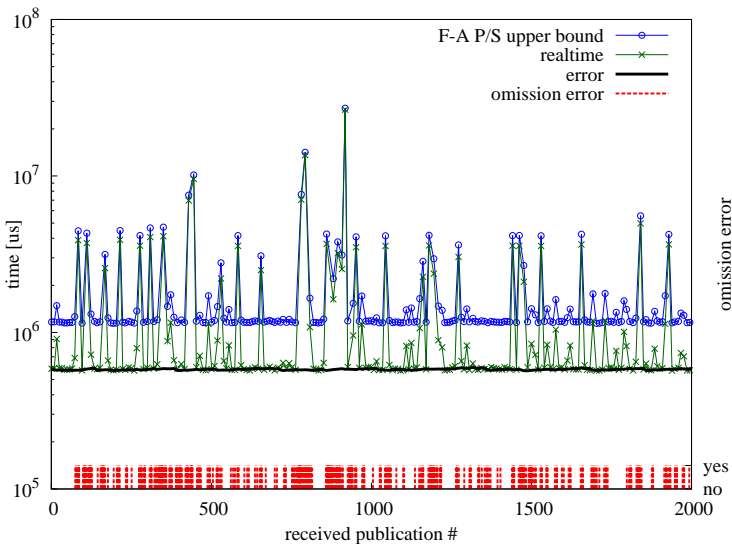
TSC as Time Source – Single Host



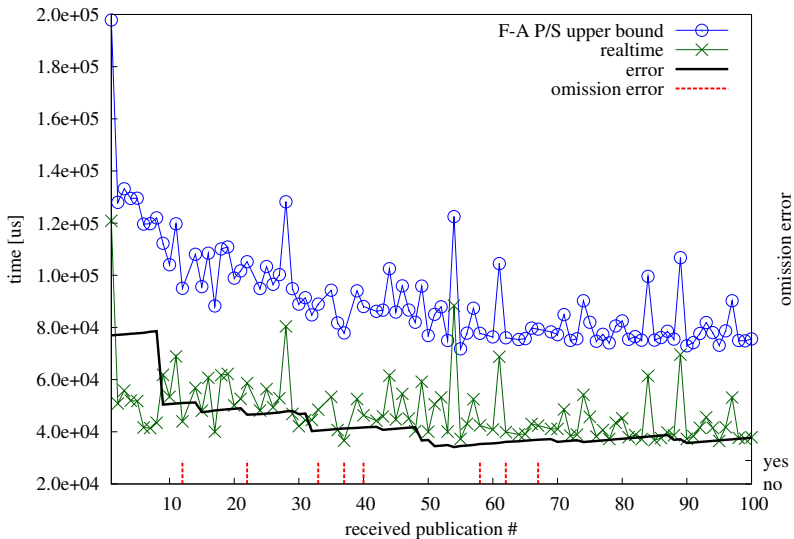
Publication Transmission Time (*national*)



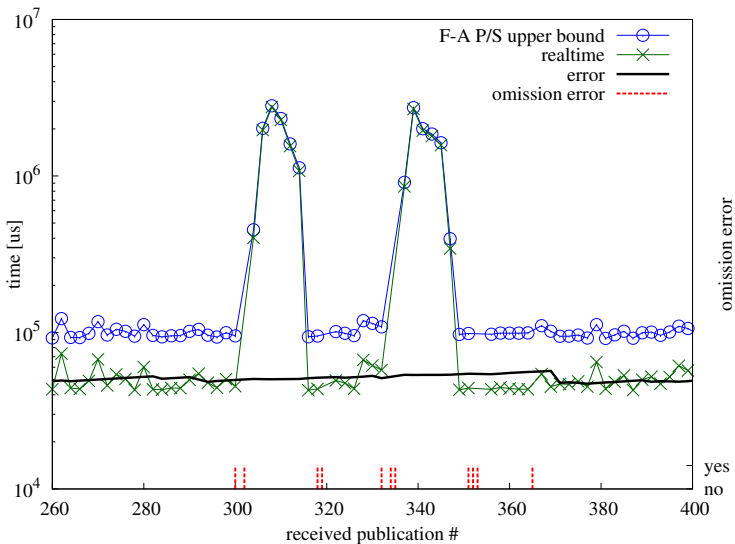
Publication Transmission Time (*global*)



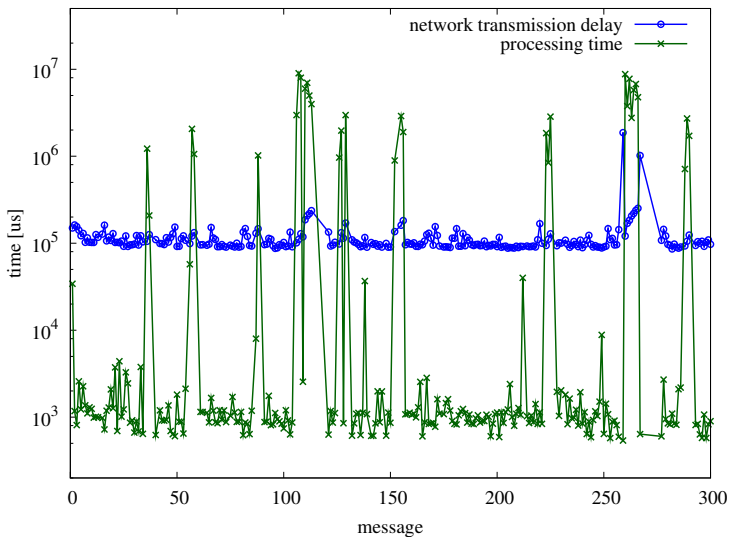
Publication Transmission Time – Bootup (*national*)



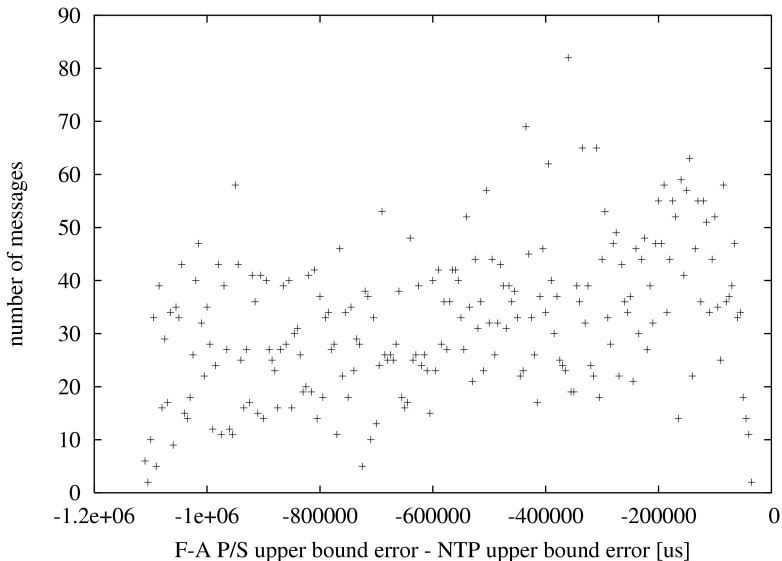
Publication Transmission Time – Error (*national*)



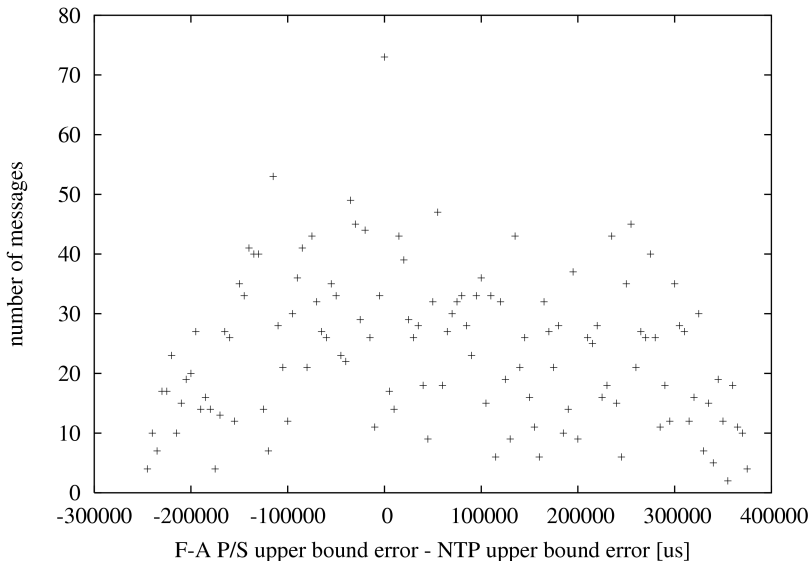
Processing Time vs Transmission Time (*national*)



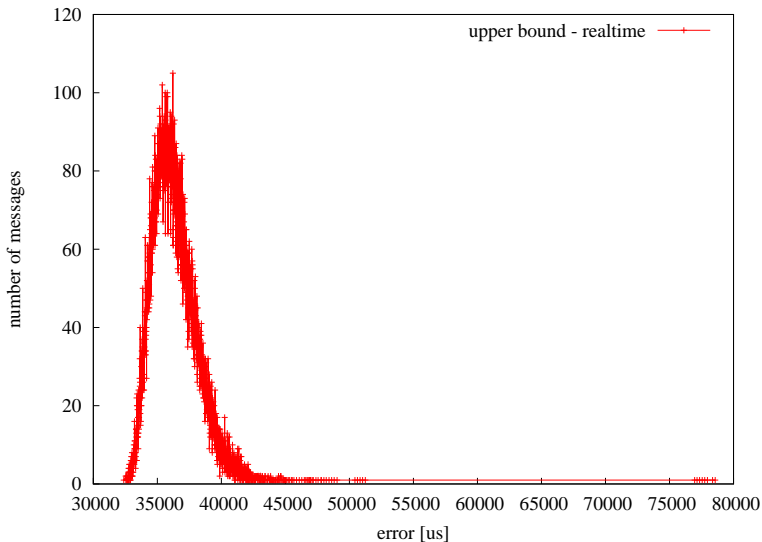
Upper Bound vs NTP (*national*)



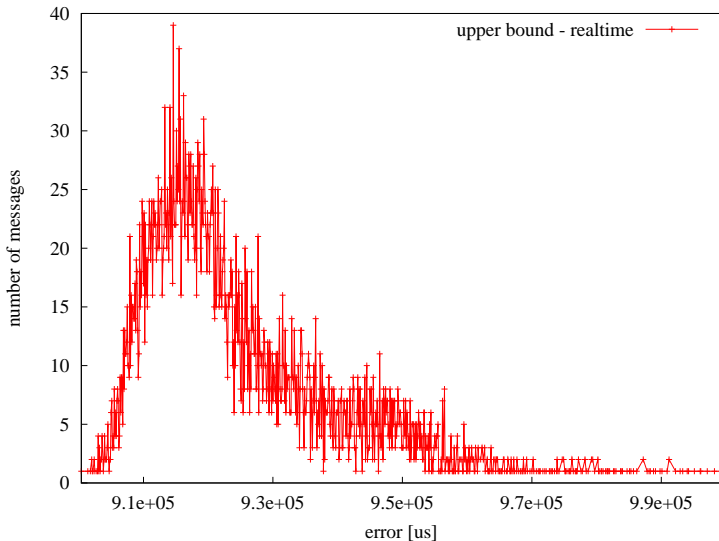
Upper Bound vs NTP (*global*)



Upper Bound vs Real Time (*national*)



Upper Bound vs Real Time (*global*)



Summary & Outlook

- ▶ Fail-Awareness in Publish/Subscribe
- ▶ Better or comparable with NTP
- ▶ No constraints on the system

Outlook

- ▶ Take a closer look at omissions

Summary & Outlook

- ▶ Fail-Awareness in Publish/Subscribe
- ▶ Better or comparable with NTP
- ▶ No constraints on the system

Outlook

- ▶ Take a closer look at omissions

Thank You!

<http://wwwse.inf.tu-dresden.de/zib/>

References



Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf.
Design and evaluation of a wide-area event notification service.
ACM Trans. Comput. Syst., 19(3):332–383, 2001.



Christof Fetzer and Flaviu Cristian.
Fail-awareness: An approach to construct fail-safe applications.
Journal of Real-Time Systems, pages 203–238, March 2003.



Alan Mislove, Ansley Post, Andreas Haeberlen, and Peter Druschely.
Experiences in building and operating ePOST, a reliable peer-to-peer application.
In *EuroSys*, 2006.