Brief Announcement: Asynchronous Bounded Expected Delay Networks

Rena Bakhshi[1] Jörg Endrullis[1] Wan Fokkink[1] Jun Pang[2]

[1]Department of Computer Science
Vrije Universiteit Amsterdam
1081HV Amsterdam, Netherlands
[rbakhshi,joerg,wanf]@few.vu.nl

[2]Computer Science and Communications
Université du Luxembourg
L-1359 Luxembourg, Luxembourg
jun.pang@uni.lu

ABSTRACT

We propose a natural generalisation of asynchronous bounded delay (ABD) network models. The commonly used ABD models assume a known bound on message delay. This assumption is often too strict for real-life applications. To this end we introduce a novel probabilistic network model, called asynchronous bounded expected delay (ABE), which requires a known bound on the expected message delay. While the conditions of ABD networks restrict the set of possible executions, in ABE networks all asynchronous executions are possible, but executions with extremely long delays are less probable. The ABE model captures asynchrony that occurs in sensor networks and ad-hoc networks.

Categories and Subject Descriptors: C.2.1 Computer-communication Networks: Network Architecture and Design, D.4.7 Organization and Design: Distributed systems.

General Terms: Algorithms, Performance, Theory.

Keywords: ABD networks, probabilistic algorithms, anonymous networks.

1. INTRODUCTION

The two main network models are synchronous and asynchronous. In synchronous network all nodes proceed simultaneously in global rounds. While this model allows for efficient algorithms, the assumptions are typically too strict for practical applications. The asynchronous network model, on the other hand, requires only that every message will eventually be delivered. The assumptions of this model are generally too weak to study the time complexity of algorithms.

For the development of practically usable, efficient algorithms we need to find a golden mean between synchronous and asynchronous networks. A possible approach is asynchronous bounded delay (ABD) networks [3, 5], where a fixed bound on the message delivery time is assumed. While this assumption allows for efficient synchronisation, it brings ABD networks closer to synchronous than to asynchronous networks. The ABD model is a nice theoretical framework, but the assumption of a bounded message delay is often hard to satisfy in real-life networks.

We propose a probabilistic model, that we call *asynchronous bounded expected delay (ABE) networks*. The ABE network model extends asynchronous networks with the assumption that a bound on the expected message delay is known. Thus, we strengthen the asynchronous networks model with a minimal requirement for analysing the time complexity of algorithms. Surprisingly, this assumption suffices for the development of efficient algorithms. For asynchronous rings, the lower bound on the message complexity for leader election is known to be $\Omega(n \cdot \log n)$. In [2], a leader election algorithm for anonymous, unidirectional ABE rings of known size n has been developed, having both (average) linear time and message complexity. So its efficiency is comparable to the most optimal leader election algorithms known for anonymous, synchronous rings [4].

We briefly elaborate on the benefits of the ABE network model. In practice, the message delay is often unbounded. The reasons for an unbounded delay are multifarious; to name a few:

(i) message queueing due to limited network bandwidth and peaks in the network load,
(ii) dynamic message routing,
(iii) message retransmission due to loss or corruption of messages transmitted via physical channels.

We shed a bit more light on the case (iii). Messages sent via a physical channel may get lost or corrupted, for example, due to material imperfections or signal inferences. Since message transmission is unreliable, all we can settle for is a probability p of successful transmission. To ensure that a message arrives at its destination, it may need to be retransmitted (possibly multiple times) until the transmission has been successful. The number of necessary retransmissions for a message cannot be bounded: with probability $(1-p)^k$ a message requires more than k retransmissions, and thus the message delay is unbounded. While the message delay cannot be bounded, from the probability p we can derive the average number of needed retransmissions and thereby the average message delay. In particular, the average number of transmissions is $k_{avg} = \sum_{k=0}^{\infty}(k+1) \cdot (1-p)^k \cdot p = \frac{1}{p}$. If a successful transmission takes one time unit, the average message delay is $\frac{1}{p}$ as well.

2. ABE NETWORKS

We introduce the model of ABE networks, which strengthens asynchronous networks with the assumption of a known bound on the expected message delay. This strengthening allows one to analyse the (average) time complexity of algorithms.

PODC'10, July 25–28, 2010, Zurich, Switzerland.
ACM 978-1-60558-888-9/10/07.

At first glance it may appear superfluous to consider a bound on the expected delay, instead of the expected delay itself. We briefly motivate our choice. First, when determining the expected delay for real-world networks, one needs to take into account parameters such as material properties, environmental radiation, electromagnetic waves, etc. Frequently, these values change over time, or cannot be calculated precisely. Thus we have to cope with ranges for each of these parameters, and consequently, the best we can deduce is an upper bound on the expected message delay. Second, the links in a network are typically not homogeneous and often have different expected delays. Then the maximum of these delays can be chosen as an upper bound, instead of having to deal with different delays for ever link.

Definition 1 We call a network *asynchronous bounded expected delay (ABE)* if the following holds:

1. A bound δ on the expected message delay (network latency) is known. (We assume that the delays of different messages are stochastically independent.)
2. Let t be a real time. We assume that bounds $0 < s_{low} \leq s_{high}$ on the speed of the local clocks are known. That is, for every node A the following holds for the local clock $\mathcal{C}_A$:
   $s_{low} \cdot (t_2 - t_1) \leq |\mathcal{C}_A(t_2) - \mathcal{C}_A(t_1)| \leq s_{high} \cdot (t_2 - t_1)$.
3. A bound γ on the expected time to process a local event is known.

In comparison with ABD networks, the ABE network model relieves the assumption of a strict bound on the message delay. The assumption is weakened to a bound on the expected message delay. Thereby we obtain a probabilistic network model which, as discussed above, covers a wide range of real-world networks to which the ABD network model is not applicable. Thus, we advocate the model of ABE networks as a natural and useful extension of the asynchronous network model.

To conclude this section, we discuss synchronisers for ABE networks. A synchroniser is an algorithm to simulate a synchronous network on another network model. A well-known impossibility result [1] states that asynchronous networks cannot be synchronised with fewer than n messages per round (ever node needs to send a message every round). This of course destroys the message complexity when running synchronous algorithms in an asynchronous network. The more efficient ABD synchroniser by Tel et al. [6] relies on knowledge of the bounded message delay. As in asynchronous networks the message delay in ABE networks is unbounded (although we assume a bound on the expected delay). In a slogan: every execution of an asynchronous network is also an execution of an ABE network. The difference is that huge message delays in ABE networks are less probable. Hence, the impossibility result [1] for asynchronous networks carries over to ABE networks, and we obtain the following theorem:

Theorem 1 *ABE networks of size n cannot be synchronised with fewer than n messages per round.* □

Hence, we cannot run synchronous algorithms in ABE networks without losing the message complexity.

Although ABE networks are very close to asynchronous networks, it turns out the model allows for the development of efficient algorithms.

3. AN ELECTION ALGORITHM FOR ANONYMOUS RINGS

We present an election algorithm for anonymous, unidirectional ABE rings. The algorithm is parameterised by a *base activation parameter* $\mathcal{A}_0 \in (0, 1)$. The order of messages is arbitrary between any pair of nodes.

During execution of the algorithm every node is in one of the following states: idle, active, passive or leader; in the initial configuration all nodes are idle. Moreover, every node A stores a number $d(A)$, initially 1. The messages sent between the nodes are of the form $\langle \mathsf{hop} \rangle$, where $\mathsf{hop} \in \{1, \ldots, n\}$ is the hop-counter of the message. Every node A executes the following algorithm:

- If A is idle, then at every clock tick, with probability $1 - (1 - \mathcal{A}_0)^{d(A)}$, A becomes active, and in this case sends the message $\langle 1 \rangle$.
- If A receives a message $\langle \mathsf{hop} \rangle$, it sets $d(A) = \max(d(A), \mathsf{hop})$. In addition, depending on its current state, the following actions are taken:
  (i) If A is idle, then it becomes passive and sends the message $\langle d(A) + 1 \rangle$.
  (ii) If A is passive, it sends the message $\langle d(A) + 1 \rangle$.
  (iii) If A is active, then it becomes leader if $\mathsf{hop} = n$, and otherwise it becomes idle, purging the message in both cases.

In other words, messages travel along the ring and 'knock out' all idle nodes on their way. That is, idle and passive nodes *forward* messages; by forwarding a message, idle nodes are turned passive. If a message has knocked out an idle node (at any point during its lifetime), we refer to the message as *knockout message*. If a message hits an active node, then it is purged, and the active node becomes idle, or is elected leader if $\mathsf{hop} = n$ (that is, if the node itself is originator of the message).

The value $d(A)$ stores the highest received hop-count for every node. It indicates that $d(A) - 1$ predecessors are passive. A higher value of $d(A)$ increases the probability that a node A becomes active. By taking $1 - (1 - \mathcal{A}_0)^{d(A)}$ as wake-up probability for nodes A, we achieve that the overall wake-up probability for all nodes stays constant over time. This ensures that the algorithm has linear time and message complexity. For details the reader is referred to [2].

4. REFERENCES

[1] B. Awerbuch. Complexity of network synchronization. *J. ACM*, 32(4):804–823, 1985.

[2] R. Bakhshi, J. Endrullis, W. Fokkink, and J. Pang. Asynchronous bounded expected delay networks. *CoRR*, (1003.2084), 2010. http://arxiv.org/abs/1003.2084.

[3] C. T. Chou, I. Cidon, I. S. Gopal, and S. Zaks. Synchronizing asynchronous bounded delay networks. *IEEE Trans. on Communications*, 38(2):144–147, 1990.

[4] A. Itai and M. Rodeh. Symmetry breaking in distributed networks. *Inf. Comput*, 88(1):60–87, 1990.

[5] G. Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 2000. 2nd edition.

[6] G. Tel, E. Korach, and S. Zaks. Synchronizing ABD networks. *IEEE/ACM Trans. Netw.*, 2(1):66–69, 1994.