Theoretical Computer Science 726 (2018) 41–77

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Gracefully degrading consensus and k-set agreement in directed dynamic networks [☆]

Martin Biely [a], Peter Robinson [b], Ulrich Schmid [c], Manfred Schwarz [c], Kyrill Winkler [c]

[a] EPFL IC IIF LSR, Station 14, CH-1015 Lausanne, Switzerland
[b] Department of Computing & Software, McMaster University, Canada
[c] Embedded Computing Systems Group, TU Wien, Vienna, Austria

ARTICLE INFO

Article history:
Received 24 August 2015
Received in revised form 19 February 2018
Accepted 21 February 2018
Available online 6 March 2018
Communicated by D. Peleg

Keywords:
Directed dynamic networks
Consensus
k-Set agreement
Message adversaries
Impossibility results
Lower bounds
Failure detectors

ABSTRACT

We study distributed agreement in synchronous directed dynamic networks, where an omniscient message adversary controls the presence/absence of communication links. We prove that consensus is impossible under a message adversary that guarantees weak connectivity only, and introduce eventually vertex-stable source components (VSSCs) as a means for circumventing this impossibility: A VSSC(k, d) message adversary guarantees that, eventually, there is an interval of d consecutive rounds where every communication graph contains at most k strongly connected components consisting of the same processes (with possibly varying interconnect topology), which have no incoming links from outside processes. We present a consensus algorithm that works correctly under a VSSC$(1, 4E + 2)$ message adversary, where E is the dynamic network depth. Our algorithm maintains local estimates of the communication graphs, and applies techniques for detecting network stability and univalent system configurations. Several related impossibility results and lower bounds, in particular, that neither a VSSC$(1, E - 1)$ message adversary nor a VSSC$(2, \infty)$ one allow to solve consensus, reveal that there is not much hope to deal with (much) stronger message adversaries here.

However, we show that gracefully degrading consensus, which degrades to general k-set agreement in case of unfavorable network conditions, allows to cope with stronger message adversaries: We provide a k-universal k-set agreement algorithm, where the number of system-wide decision values k is not encoded in the algorithm, but rather determined by the actual power of the message adversary in a run: Our algorithm guarantees at most k decision values under a VSSC(n, d) + MAJINF(k) message adversary, which combines VSSC(n, d) (with some small value of d, ensuring termination) with some information flow guarantee MAJINF(k) between certain VSSCs (ensuring k-agreement). Since related impossibility results reveal that a VSSC(k, d) message adversary is too strong for solving k-set agreement and that some information flow between VSSCs is mandatory for this purpose as well, our results provide a significant step towards the exact solvability/impossibility border of general k-set agreement in directed dynamic networks.

Finally, we relate (the eventually-forever-variants of) our message adversaries to failure detectors. It turns out that even though VSSC$(1, \infty)$ allows to solve consensus and to implement the Ω failure detector, it does not allow to implement Σ. This contrasts the fact that, in asynchronous message-passing systems with a majority of process crashes, (Σ, Ω) is a weakest failure detector for solving consensus. Similarly, although the message

☆ This work has been supported the Austrian Science Fund (FWF) projects P20529 (PSRTS), P28182 (ADynNet), S11405 (RiSE) and P26436 (SIC).

E-mail addresses: martin.biely@biely.eu (M. Biely), peter.robinson@mcmaster.ca (P. Robinson), s@ecs.tuwien.ac.at (U. Schmid), mschwarz@ecs.tuwien.ac.at (M. Schwarz), kwinkler@ecs.tuwien.ac.at (K. Winkler).

https://doi.org/10.1016/j.tcs.2018.02.019

adversary VSSC(n, d) + MAJINF(k) allows to solve k-set agreement, it does not allow to implement the failure detector Σ_k, which is known to be necessary for k-set agreement in asynchronous message-passing systems with a majority of process crashes. Consequently, it is not possible to adapt failure-detector-based algorithms to work in conjunction with our message adversaries.

1. Introduction

Dynamic networks such as wireless sensor networks, mobile ad-hoc networks and vehicle area networks, are becoming ubiquitous nowadays. The primary properties of such networks are sets of participants (called processes in the sequel) that are a priori unknown and potentially changing, time-varying connectivity between processes, and the absence of a central control. Dynamic networks is an important and very active area of research [1].

Accurately modeling dynamic networks is challenging, for several reasons: First, process mobility, process crashes/recoveries, deliberate joins/leaves, and peculiarities in the low-level system design like duty-cycling (used to save energy in wireless sensor networks) make static communication topologies, as typically used in classic network models, inadequate for dynamic networks. Certain instances of dynamic networks, in particular, peer-to-peer networks [2] and inter-vehicle area networks [3], even suffer from significant churn, i.e., a large number of processes that can appear/disappear over time, possibly in the presence of faulty processes [4], and hence consist of a potentially unbounded total number of participants over time. More classic applications like *mobile ad-hoc networks* (MANETS) [5], wireless sensor networks [6,7] and disaster relief applications [8] typically consist of a *bounded* (but typically unknown) total number of processes.

Second, communication in many dynamic networks, in particular, in wireless networks like MANETS, is inherently broadcast: When a process transmits, then every other process within its transmission range will observe this transmission — either by legitimately receiving the message or as some form of interference. This creates quite irregular communication behavior, such as capture effects and near-far problems [9], where certain (nearby) transmitters may "lock" a receiver and thus prohibit the reception of messages from other senders. Consequently, the "health" of a wireless link between two processes may vary heavily over time [10]. For low-bandwidth wireless transceivers, an acceptable link quality usually even requires communication scheduling [11] (e.g., time-slotted communication) for reducing the mutual interference. Overall, this results in a frequently changing spatial distribution of pairs of nodes that can communicate at a given point in time.

As a consequence, many dynamic networks, in particular, wireless ones [12], are not adequately modeled by means of bidirectional links. Fading and interference phenomenons [13,14], including capture effects and near-far problems, are *local* effects that affect only the receiver of a wireless link: Given that the sender, which is also the receiver of the reverse link, resides at a different location, the two receivers are likely to experience very different levels of fading and interference [15]. This effect is even more pronounced in the case of time-slotted communication, where forward and backward links are used at different times. Consequently, the existence of asymmetric communication links cannot be ruled out in practice: According to [16], 80% of the links in a typical wireless network are sometimes asymmetric.

Despite these facts, most of the dynamic network research we are aware of assumes bidirectional links [17,18]. The obvious advantage of this abstraction is simplicity of the algorithm design, as strong communication guarantees obviously make this task easier. Moreover, it allows the re-use of existing techniques for wireline networks, which naturally support bidirectional communication. However, there are also major disadvantages of this convenient abstraction: First, for dynamic networks that operate in environments with unfavorable communication conditions, like in disaster relief applications or, more generally, in settings with various interferers and obstacles that severely inhibit communication, bidirectional links may simply not be achievable. Implementing distributed services in such settings thus require algorithms that do not need bidirectional links right from the outset. Second, the entire system needs to be engineered in such a way that bidirectional single-hop communication can be provided within bounded time. This typically requires relatively dense networks and/or processes that are equipped with powerful communication interfaces, which incur significant cost when compared to sparser networks or/and cheaper or more energy-saving communication devices. And last but not least, if directed single-hop communication was already sufficient to reach some desired goal (say, reaching some destination process) via multi-hop messages, waiting for guaranteed single-hop bidirectional communication would incur a potentially significant, unnecessary delay. Obviously, in such settings, algorithmic solutions that do not need bidirectional single-hop communication could be significantly faster.

In this paper, we thus restrict our attention to dynamic networks consisting of an *unknown but bounded* total number of processes, which are interconnected by *directed* communication links. The system is assumed to be synchronous,[1] hence time is measured in discrete *rounds* that allow the processes to exchange at most one message. Time-varying communication

[1] As synchronized clocks are typically required for basic communication in wireless systems anyway, e.g., for transmission scheduling and sender/receiver synchronization, this is not an unrealistic assumption: Global synchrony can be implemented directly at low system levels, e.g., via IEEE 1588 network time synchronization or GPS receivers, or at higher levels via time synchronization protocols like FTSP [19] or even synchronizers [20].