

# Integration: Reaching Consensus in Low-Diameter Wireless Networks

Stephan Olariu  
Old Dominion University  
olariu@cs.odu.edu

Jeffrey Nickerson  
Stevens Institute of Technology  
jnickerson@stevens.edu

## Abstract

*In a centrally controlled system consensus is often reached by decree: the central entity in charge of the system dictates the “consensus” to the rank-and-file. The situation is vastly different in a truly decentralized distributed system where the various entities in the system must agree on a common view – the consensus. Consensus is, in this regard, an exercise in integration, for it is the local, often parochial, views of the various participants that are being integrated in the process. We study the consensus problem in an eminently decentralized distributed system populated by anonymous participants communicating by radio. Our main contribution is to show that consensus can be reached in four deterministic communication steps in systems whose underlying graph has diameter two, even if the topology of the network is completely unknown to the participants. This result is relevant to all situations where a consensus must be reached by anonymous participants (who perhaps do not wish to reveal their identities) provided that the underlying graph has low diameter.*

## 1. Introduction

Wireless sensor networks can be used to build situation awareness, but the information from such networks needs to be integrated in order to be useful. As sensors have become less expensive, it is possible to create sensor networks in which hundreds or thousands of nodes can communicate locally; these local communications need to be synthesized in order to define the context of a monitored area. In some networks, mobility is an added component; not only do nodes traveling with radios need to integrate, mobility changes their local context as they attempt to communicate.

Gupta and Kumar [4] showed that mobile ad hoc networks grow in such a way that each node will

become a congestion point. Others afterwards have tried using hierarchical localization as a way of overcoming the limitation in growth. Sensor networks face similar problems of integration, with the added constraint of power consumption [1]. For, if all sensors are constantly in communication, they deplete their power too rapidly to perform useful work.

There is another aspect of these networks which is of interest - their similarity to human organizational structures: humans also partition information to avoid congestion. In certain circumstances, technical and social network issues fuse. For example, in emergency situations, responders often use wireless communication technology. They rely on sensors to inform them. And they are in enough danger that they need to organize in a robust way. This has been noticed by researchers applying insights about abstract network structures to human communication [3].

As part of an overall program of research to understand these new integration problems, the authors have recently studied issues related to wireless communication, and sensor networks [6-7]. Here, the authors build on previous work in looking at a particular key aspect of integration, the building of consensus. Consensus is important in technical systems in that it can be used to smooth over malfunctions or reduce noise. Consensus also is a way of integrating information. In human communication, it performs a similar integration function, and is a crucial part of political processes, as it often plays a role in representation and governance.

Wireless environments often have the characteristic of being highly distributed, complex, with sometimes mobile participants, who, through contracts, auctions, or other mechanisms want to reach some form of agreement. We look at ways of achieving quick consensus in abstract sensor

networks; future research may extend these results to other domains such as electronic markets.

Numerous algorithms are built around the resolution of *decision problems* where each of the participants in a distributed system makes a local decision reflected by specific conditions arising in its neighborhood (the *local view*). The problem is to integrate in a suitable manner the various local views into a coherent *global view*, representative of the entire system at a certain moment in time.

Our goal is to study the consensus problem in a decentralized distributed system populated by *anonymous* nodes communicating by radio. This problem has been studied with respect to mobile ad hoc networks (e.g [2]) and sensor networks (e. g. [8]), but we start from different assumptions about clustering. Our main contribution is to show that consensus can be reached in four deterministic communication steps in systems whose underlying graph has diameter two, even if the topology of the network is completely unknown to the nodes. This result is relevant to situations where a consensus must be reached by anonymous nodes that either lack identities, as nodes in a sensor network do, or do not wish to reveal their identities (as in a voting or bidding system), provided that the underlying graph has low diameter.

The remainder of the work is organized as follows: Section 2 discusses the consensus problem. Section 3 presents our consensus protocol in anonymous diameter-2 graphs. Finally, Section 4 offers concluding remarks.

## 2. Consensus in a distributed system

The task of reaching a consensus is fundamental to distributed systems [5]. Consider a distributed system  $D$  modeled by a graph  $G=(V,E)$  where  $V$  is the set of nodes in the system, with two nodes connected by an edge whenever they can communicate directly with each other. Furthermore, communication with its neighbors is the only means for a node to acquire new information.

In this regimen of strictly local communications, nodes develop a *local view* of the network. As a rule, the local views of nodes, especially those that cannot communicate directly, may vary. The task of building a *consensus* calls for all the nodes in  $D$  to agree on a *global view* of the system. It goes without saying that reaching a consensus in a distributed system is a form of integration that is, per force, short lived and needs to be performed periodically. Consensus is relatively easy in centralized system, where it is simply

imposed by the central entity. On the other hand, reaching a consensus in a general distributed system is a difficult task.

Clustering was proposed in large-scale distributed systems as a means of achieving scalability through a hierarchical approach to consensus. Essentially, a cluster is a subset of the nodes of the underlying graph that satisfies a certain predicate  $P$ . The precise definition of  $P$  varies in different contexts, but is typically chosen in a way that makes local consensus in each cluster easy. Consequently, it provides a way of organizing a large graph into an aggregate of smaller ones.

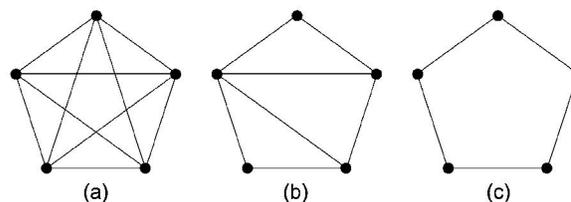


Figure 1. Various instances of predicate  $P$ .

Since the difficulty of the task of reaching a consensus increases with the diameter of the underlying graph, small-diameter clusters are generally preferred. As illustrated in Figure 1(a), some authors define predicate  $P$  such that every node in the cluster is one hop away from every other node, that is, each cluster is a diameter-1 graph (a.k.a. a clique). A less restrictive and widely-accepted definition of  $P$ , illustrated in Figure 1(b), insists on the existence of a central node adjacent to all the remaining nodes in the cluster. Such a central node is typically referred to as *cluster-head* and the corresponding predicate  $P$  is also known as the *dominance* property.

As mentioned before, in the presence of a central node consensus is reached trivially: indeed, the cluster-head dictates the consensus. However, the cluster-head may also easily become the traffic bottleneck and single point of failure of the cluster. Additionally, in wireless distributed systems where the nodes move, these two predicates are rather hard to enforce requiring frequent re-clustering, a considerable overhead.

To overcome these difficulties, one is lead naturally to defining clusters simply as a diameter-2 graphs, as illustrated in Figure 1(c). Only when the cluster is no longer a diameter-2 graph, will the cluster maintenance algorithm be invoked. This latter definition of a cluster is less restrictive and results in

more robust clusters and in significantly lower overhead for cluster maintenance. In fact, this reasoning can be carried further. Specifically, one can adopt the diameter- $k$  property, ( $k > 0$ ), as the sole defining predicate of a cluster. The trade-off is clear: one is trading ease of reaching a consensus for stability and lower maintenance costs. A large value of  $k$  results in very stable clusters but reaching a consensus becomes hard. In the limit, when  $k = \text{diam}(g)$  the entire system is just one cluster that is trivial to maintain but reaching a consensus becomes a nightmare. This implies that there must exist some value of  $k$  that strikes a most desirable balance between the two tasks. The optimal value of  $k$  is application dependent: what exactly one means by consensus varies across the spectrum of networks and applications.

### 3. Reaching consensus in an anonymous diameter-two graph

We place our discussion in the context of a wireless network populated by nodes equipped with radio transceivers. The nodes have a transmission range that is small relative to the diameter of the network. We assume that the time is slotted and the nodes are synchronous. They know that the cluster to which they belong induces a diameter-2 graph. The nodes are anonymous (very much like individual sensors in a wireless sensor network) and have no topological information: they can receive radio signals from their neighbors, nothing else.

Communication between nodes is omnidirectional and symmetric. As a consequence, a message transmitted by a node is received (in the same slot in which it is transmitted) by all the nodes within its transmission range. If a node receives packets transmitted by two or more nodes in the same time slot, then they collide and are garbled beyond recognition. Accordingly, for a node the *local status* of a channel in a time slot is:

- $N$  (null) if no node in its range transmitted on the channel,
- $S$  (single) if exactly one node in its range on the channel, and
- $C$  (collision) if two or more stations in its range transmitted on the channel.

By definition a transmitted packet is successfully received only if the status of the channel is  $S$ . We further assume that in a given slot a node can both

transmit and receive: this is not unreasonable as it assumes the existence of a transmitted and a received at each node. (This assumption may impact the energy consumption of nodes. However, in this paper this is of no immediate concern.)

Consider a diameter-2 cluster where each of the nodes has generated either a 1 or a 0. For example, in case of a wireless sensor network this is a value generated by each sensor as a result of engaging in a sensing operation. Notice that having generated its local value, each node is only aware of its local status: namely, the value it had generated. Assume that in some application it is important for the nodes to agree on exactly one of the following outcomes:

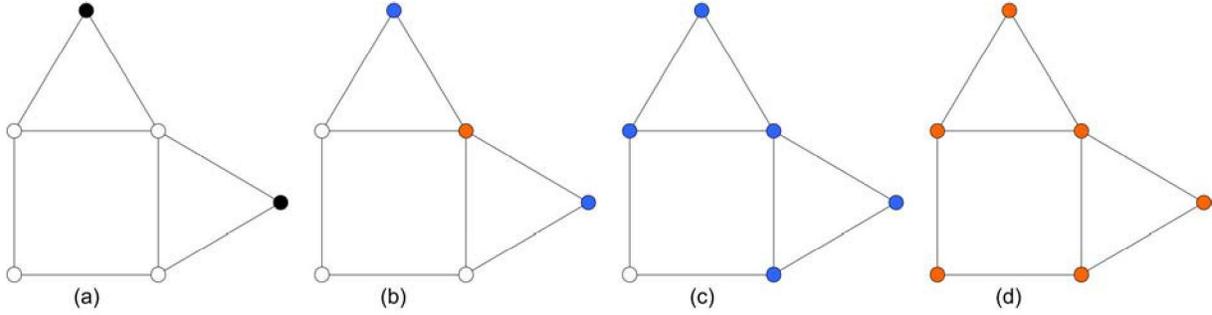
- A. No value of 1 was generated in the cluster
- B. Exactly one value 1 was generated in the cluster
- C. Two or more values of 1 were generated in the cluster.

The consensus problem, then, is for the nodes to agree on a global view, that is, on one of the outcomes A or B or C above. This task is non-trivial because of the lack of individual IDs, the lack of topological information and the strict communication regimen. For example, referring to Figure 2(a) the two dark nodes have generated values of 1, the others 0. Thus, the appropriate consensus for the cluster is outcome C.

The main goal of this subsection is to propose an efficient consensus algorithm for the scenario described above. To be rigorous, we need to ensure that the consensus reached by the nodes correctly reflects the situation in the cluster.

The idea of the algorithm is simple. To begin, the set  $T$  of nodes that store a value of 1 transmit in the same slot. Let  $|T|$  stand for the cardinality of  $T$ . In order for the nodes to reach a consensus it suffices to determine whether  $|T|=0$ , or  $|T|=1$ , or  $|T| > 1$ . We shall proceed in the following two stages: in the first state we discriminate the case  $|T|=0$  from  $|T| > 0$ . In the next stage, we discriminate between the cases  $|T|=1$  and  $|T| > 1$ .

Let  $Status(t)$  denote the local status of the channel that a given node perceives in time slot  $t$ . For proper discrimination, each node stores the *local history* of the channel, as captured by the sequence  $Status(1)$ ,  $Status(2)$ ,  $Status(3)$ ,... At the end of the algorithm, when consensus has been reached, each node stores the corresponding values in register  $Global(t)$ . Prior to time slot  $t$ ,  $Global(t)$  is undefined.



**Figure 2.** Illustrating the consensus algorithm.

The details of the algorithm for the first stage are spelled out as follows.

STAGE I:

**Slot 1:** Every node in  $T$  transmits

**Slot 2:** Every node in the cluster with  $Status(1) \neq N$  transmits. Every node such that  $Status(2) = N$  sets  $Global(2) = A$  and terminates the algorithm.

Some of the results of correctness in this section rely on the following technical result.

**Lemma 0.** Every pair of non-adjacent nodes in a diameter-2 graph has a common neighbor.

**Proof.** Let  $u$  and  $v$  be two non-adjacent nodes in a diameter-2 graph and let  $\pi(u, v)$  be the shortest path connecting them. Since the diameter of the graph is at most two, it follows that the length  $|\pi(u, v)|$  of  $\pi(u, v)$  satisfies  $|\pi(u, v)| \leq 2$ . Since the nodes were assumed to be non-adjacent, it follows that  $|\pi(u, v)| = 2$ , confirming that  $u$  and  $v$  must have a common neighbor.  $\square$

**Theorem 1.**  $|T| > 0$  if and only if for an arbitrary node in the cluster  $Status(2) \neq N$ .

**Proof.** First, if  $|T| = 0$  then no station has transmitted in Slot 1 and  $Status(1) = N$  for all the nodes in the cluster. As a consequence, no node transmits in Slot 2 and so  $Status(2) = N$  for all nodes in the cluster.

Conversely, suppose that the status  $Status(2)$  of a node at the end of Slot 2 is either  $S$  or  $C$ . We claim that no node in the cluster can have  $Status(2) = N$ . Indeed, suppose that some node would have  $Status(2) = N$ . This implies that neither the node nor any of its neighbors has transmitted in Slot 2. In turn, this implies that neither the node nor any of its neighbors has received a message at the end of Slot 1 (in other words that all of them had  $Status(1) = N$ ). But now we have reached a contradiction. By Lemma 0, there could have been no transmission in the

cluster, and yet the status  $Status(2)$  of some of the nodes was distinct from  $N$ . Thus, all the nodes in the cluster must have had  $Status(2) \neq N$ . In turn, this means that some node must have transmitted in Slot 1, confirming that  $|T| > 0$ . This completes the proof of the theorem.  $\square$

Theorem 1 has the following important consequence.

**Corollary 2.** The algorithm corresponding the Stage I correctly discriminates the cases  $|T| = 0$  and  $|T| > 0$ .

By Corollary 2, at the end of Stage I, the consensus was reached only if no node generated a 1, i.e. the correct consensus was A. Stage I is not sufficiently powerful to distinguish between outcomes B and C. For that purpose, we need the algorithm specific to Stage II detailed below.

STAGE II:

**Slot 3:** Every node in the cluster with  $Status(1) = C$  transmits.

**Slot 4:** Every node in the cluster with  $Status(3) \neq N$  transmits. Every node in the cluster sets  $Global(4) = B$  or  $Global(4) = C$  depending on whether or not its own  $Status(4) = N$ .

For an illustration refer to Figure 2. Figure 2(a) is depicting the initial configuration, where two of the nodes have generated values of 1 and the others 0. Figure 2(b) illustrates the local status of various nodes in the cluster. The common neighbor of the transmitting nodes has recorded a collision (red) and the other a single transmission (blue). In Slot 3 the red node transmits. The status at the end of Slot 3 is depicted in Figure 2(c). In Slot 4 all the blue nodes transmit. The final status is shown in Figure 2(d). At this point the consensus has been reached.

It is quite surprising that this simple algorithm should allow the nodes in a general diameter-2 graph to correctly reach consensus. That this is, indeed, the case will be proved by the following technical result.

**Theorem 4.**  $|T|=1$  if and only if for an arbitrary node in the cluster  $Status(4)=N$ .

**Proof.** Since Stage II was executed, we must have  $|T|\geq 1$ . In order to distinguish between the case  $|T|=1$  and  $|T|>1$ , we rely, again, on Lemma 0. Suppose that  $|T|=1$ . Then, no node in the cluster can have  $Status(1)=C$ . Consequently, there is no transmission in Slot 3 and every node in the cluster has  $Status(3)=N$ . This further implies that no node in the cluster transmits in Slot 4 and thus, for every node,  $Status(4)=N$ .

Conversely, assume that  $|T|>1$ . By Lemma 0, it must be the case that for some nodes in the cluster,  $Status(1)=C$ . To see that this is the case, we distinguish between two cases:

All the nodes that transmitted in Slot 1 were pairwise adjacent. In this case, trivially, all of them will have  $Status(1)=C$ .

Some two nodes that transmitted in Slot 1 were non-adjacent. By Lemma 0, they must have a common neighbor that receives both transmissions, recording a collision. The common neighbor with have  $Status(1)=C$ .

Thus, some node(s) must have transmitted in Slot 3 and, therefore, some node must have had  $Status(3)\neq N$ . Such nodes have transmitted in Slot 4. In order to complete the proof of the theorem, we need show that no node in the cluster can have  $Status(4)=N$ .

For, consider such a node. It is clear that neither the node, nor any of its neighbors has transmitted or heard a transmission in Slot 3. By Lemma 0, this means that no node has heard a collision in Slot 1, contradicting that  $|T|>1$ . Thus, every node in the cluster has had  $Status(4)\neq N$ , as claimed.

Theorem 1 and Theorem 4, combined, prove that in all cases the nodes in the cluster have reached the correct global consensus. Notice the generality of this result: correctness does not depend on nodes knowing the topology of the cluster (or parts thereof) and the algorithm works even if the nodes lack individual IDs, making it eminently applicable to sensor networks.

## 4. Concluding remarks

Highly distributed networks need ways of quickly integrating their information; consensus is one recognized method. We have shown in this paper that network clusters of a certain characteristic, having a diameter of 2, can be integrated effectively with no need to know the locations or even the identities of others. Future research will explore the hierarchical implications of this technique; once consensus has been reached, a node might represent its cluster to neighboring clusters. It is possible the algorithm discussed may be useful as an integration technique in a variety of applications in which mobility and anonymity play important roles.

## Acknowledgements

This research was supported in part by the Office of Naval Research, grant #N00014-05-1-00632.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. Wireless sensor networks: A survey, *Computer Networks*, 38(4), 2002, 393-422.
- [2] G. Chockler, M. Demirbas, S. Gilbert, C. Newport, and T. Nolte. Consensus and collision detectors in wireless Ad Hoc networks. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 197-206.
- [3] P. S. Dodds and C. F. Sabel. Information exchange and the robustness of organizational networks, *PNAS* (100:21) 2003, 12516-12521.
- [4] P. Gupta and P. R. Kumar. "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. IT-46, no. 2, March 2000. 388-404.
- [5] N. Lynch. *Distributed Algorithms*, Morgan Kaufmann, San Francisco, CA, 1996.
- [6] J. V. Nickerson and S. Olariu. "A Measure for Integration and its Application to Sensor Network," Workshop on Information Technology and Systems (WITS), 2005.
- [7] S. Olariu and J. V. Nickerson. "Protecting with Sensor Networks: Perimeters and Axes," SIMA Workshop, MILCOM, 2005.
- [8] L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti. Localized techniques for broadcasting in wireless sensor networks. In *Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, 41-50.