Chapter 31
Cache Control Issues in Pub–Sub Networks and Wireless Sensor Networks

Dimitrios Katsaros

31.1 Introduction

Applications that exploit the publish–subscribe (pub–sub) paradigm are organized as a collection of clients which interact by publishing events and by subscribing to the events they are interested in. In a pub–sub network, any message is guaranteed to reach all interested active clients whose subscriptions are known at publish time. However, in a dynamic distributed environment, clients join and leave the network during time, and it is possible that a client joins the network after the publishing of an interesting message. In current pub–sub systems, it is not possible for a new subscriber to retrieve previously published messages that match her subscription. Therefore, enabling the retrieval of previously published content by means of storing is one of the most challenging problems in pub–sub networks. Wireless sensor networks (WSNs) consist of wirelessly interconnected devices that can interact with their environment by controlling and sensing "physical" parameters. Although there is no single realization of a WSN to support all applications, there are some common characteristics of these networks that need to be efficiently addressed in all these applications: (a) lifetime, (b) scalability, and (c) data-centric networking (whereas the target of a conventional communication network is to move bits from one machine to another, the purpose of a sensor network is to provide information and answers). Therefore, techniques of temporary caching of information at various places in the sensor network is a challenging issue that can achieve all three requirements. The caching decisions are strongly dependent on the network topology; therefore, analysis of the topology by discovering which nodes are located in "central" positions of the network can improve the caching algorithms.

D. Katsaros (✉)
University of Thessaly, Volos, Greece
e-mail: dkatsar@inf.uth.gr

J.H. van Schuppen and T. Villa (eds.), *Coordination Control of Distributed Systems*,
Lecture Notes in Control and Information Sciences 456,
DOI 10.1007/978-3-319-10407-2_31

31.2 Motivation

To support archival retrieval in a pub–sub network, data storage servers replicate the whole content of a given server. When a client is interested in the content of that server, his/her request is redirected to one of the existing storages (i.e., the closest one). Since storages serve only a portion of the total requests and are placed closer to the client, clients are served faster. Therefore, the objective function is to minimize client's response latency subject to installing the minimum number of storages.

On the other hand, the vast majority of applications running over WSNs require the optimization of the communication among the sensors so as to serve the requested data in short latency and with minimal energy consumption. The battery lifetime can be extended if the "amount" of communication is reduced, which in turn can be done by caching useful data for each sensor either in its local store or in the near neighborhood. Additionally, caching can be very effective in reducing the need for network-wide transmissions, thus reducing the interference and overcoming the variable channel conditions. The cooperative data caching is an effective and efficient technique to achieve these goals. The fundamental aspect in every cooperative caching schemes for sensor networks is the identification of the nodes which will implement the aspects of the cooperation concerning the caching decisions. Therefore, we need to define nodes that will run control decisions usually without complete knowledge of the state of neighboring nodes, and also to define a cache admission/replacement policy for the contents of each sensor node cache.

31.3 Examples

Several industrial and academic pub–sub systems such as IBM's Gryphon, Siena, Elvin, and REDS develop an overlay event notification service. An event notification service is an infrastructure that facilitates the construction of event-based systems, whereby producers of events publish event notifications to the infrastructure, and consumers of events subscribe with the infrastructure to receive relevant notifications. The two primary services that should be provided by the infrastructure are the determination of which notifications match which subscriptions, and routing notifications from producers to consumers. However, these systems are lacking support of archival retrieval.

In the area of WSNs, the necessity of cooperative caching can be exemplified via the following example scenario. Consider, a sensor network deployed in a modern battlefield, with sensor nodes dispersed in a large area; each sensor node is equipped with a micro-camera that can take a photograph of a very narrow angle around its position. The sensors update the photographs they take (storing a prespecified number of the immediate past images, so as to be able to respond to historic queries), and share (on demand) with each other the new photographs, in order to built a more complete view of the region that is being monitored. The sharing is necessary because every micro-camera can capture a limited view of the whole region, either due to the sensor node's position or because of the obstacles that exist nearby the sensor node. Therefore, every sensor may request and receive a large number of photographs taken by some other sensor(s) through multihop communication. Afterward, each sensor is able to respond to queries about "high-level" events, e.g., enemy presence. Apparently, sensor battery recharging might be infeasible or difficult due to the limited access to the field. Also, the location of the sensors has not been decided by some placement algorithm (the sensors were dropped by an unmanned aircraft), and the communication is strictly multihop.

31.4 Problem Formulation

Even though the caching problems in pub–sub and WSNs are not identical, they bear many similarities, and thus, we will only provide the formulation of the online cooperative caching problem in WSNs. This is a control problem combining the cache admission and the cache replacement control policies, which continuously try to optimize the cache contents in a way that optimizes a performance measure, e.g., the percentage of requests serviced by each cache.

In the online cooperative caching problem, there are several goals that need to be optimized, such as energy consumption and access latency. These goals are often conflicting. Therefore, it is unfeasible to formulate the online cooperative caching problem using a single equation that would encompass all these factors. We express it here as an optimization problem with the goal of optimizing one of these metrics only, i.e., access latency. Thus, we provide the following formulation for the cooperative caching problem.

Given an ad hoc network of sensor nodes $G(V, E)$ with p equisized data items $D_1, D_2, \ldots, D_p$, where data item D_j can be served by a sensor SN_j, a sensor may act as a server of multiple data items. Each sensor SN_i has a capacity of m_i units of storage, e.g., bits. We use a_{ij} to denote the access frequency with which a sensor SN_i requests the data item D_j and d_{il} to denote the distance (in hops) between sensors i and l. The *cooperative caching problem* is an online problem, with the goal being the selection of a set of sets $M = \{M_1, M_2, \ldots, M_p\}$, where M_j is a set of sensors that store a copy of D_j, to minimize the total access cost:

$$\tau(G, M) = \sum_{i \in V} \sum_{j=1}^{p} a_{ij} \times min_{l \in (\{SN_j\} \cup M_j)} d_{il} \tag{31.1}$$

and fulfilling the memory capacity constraint that:

$$|\{M_j | i \in M_j\}| \leq m_i \qquad \text{for all } i \in V,$$

which means that each network node SN_i appears in at most m_i sets of M.

31.5 Theory and Concepts

Since earlier work [4] suggested that the smart selection of the so-called “mediator” nodes is a crucial factor in addressing energy and latency considerations, one of the aims was to design centrality measures [8] to help us in the selection of the mediator nodes which will be robust and easy to compute (without the need of complex calculations or many rounds of message exchanges). The “central” nodes are able to control the communication among others: for instance, (a) in routing protocols for sensor networks, such nodes can be selected to forward the packets because, due to their position, they will succeed in reducing the routing latency, (b) in disconnection-tolerant mobile sensor networks, such nodes can be selected as data mules to carry messages, until they find the chance to pass these messages to sensors which are closer to the packets’ final destination, and so no. Therefore, the significance of such central sensors varies depending on the application and the protocol, and thus, we use the word “influence” to depict the ability of the central nodes to affect (usually for optimization purposes) the communication among other sensors.

31.6 Overview of Research Contributions

The sensor degree, i.e., the number of its 1-hop neighbors, has been used as a measure of centrality. Looking at Fig. 31.1, we see that the nodes 3, 4, 7, 6 are equally central with respect to their degree. If we compute the betweenness centrality for each sensor—the percentage of shortest paths among all pairs of sensors that pass via this sensor—in the whole graph, then node 7 is the most “central," followed by nodes 3, 4 and then comes node 6. This is somehow counter-intuitive, since node 6 has *all network nodes* at its vicinity.

Starting from this observation, we propose a new centrality measure, called the μ-*Power Community Index*, defined as follows [12]:

Definition 31.1 The μ-*Power Community Index* of a sensor v is equal to k, such that there are up to $\mu \times k$ sensors in the μ-hop neighborhood of v with degree greater than or equal to k, and the rest of the sensors with in that neighborhood have a degree less than or equal to k.

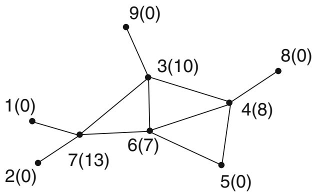

Fig. 31.1 Betweenness centrality (the numbers in parentheses) for a graph

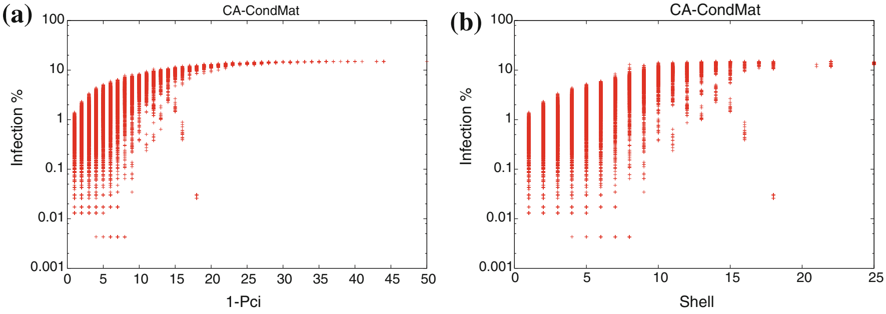

Fig. 31.2 Spreading capability of nodes in the ca-CondMat network with a single original spreader according to **a** 1-PCI and **b** k-shell index. There are nodes with high k-shell indices, some of which infect a large portion of the network, as well as nodes with the same k-shell index (16) that infect a significantly smaller part of the network. On the other hand, only nodes with very small 1-PCI exhibit such behavior

The calculation of this measure is completely local involving only communication among neighboring nodes without knowledge of the complete network topology. Having defined such "controller" sensors, the task of providing a solution to the online cooperative caching problem can be done along the lines proposed in [4] and [5].

The μ*-Power Community Index* has been used also to address the problem of influential spreader identification in complex networks [2], which is yet another network control problem that aims at finding the nodes in complex networks that can spread a message rapidly among other nodes or finding nodes that can control the rest of the nodes. So far, the k-shell decomposition was the champion method; if from a given graph, we recursively delete all vertices and edges incident with them of degree less than k, the remaining graph is the k-shell.

Figure 31.2 shows all nodes' spreading capability according to their 1-PCIs and k-shell indices for the ca-CondMat collaboration networks from the e-print arXiv covering condensed matter physics. The 1-PCI method results in a more monotonic distribution than k-shell decomposition, providing a clearer ranking of spreading capabilities. It converges to an approximately straight line, where maximum influence lies, more steeply than the k-shell method in all the studied cases. Choosing a spreader with, say, 1-PCI $>$ 23 will yield the maximum influence, whereas choosing one from the core or from the high shells might not be optimal because in some cases nodes within the same shell have different spreading capabilities.

31.7 Further Reading

Relevant research on the use of social network analysis for improving the performance of networks includes the analysis of time-varying networks [1, 16], the detection of communities [7, 13, 15], the proposal of new centrality measures

[6, 11, 14, 17], the discovery of influential spreaders [2], and the social-based routing [3, 9, 10, 18].

References

1. Acer UG, Drineas P, Abouzeid A (2010) Random walks in time-graphs. In: Proceedings of MOBIOPP, pp 93–100
2. Basaras P, Katsaros D, Tassiulas L (2013) Detecting influential spreaders in complex, dynamic networks. IEEE Comput mag 46(4):26–31
3. Daly EM, Haahr M (2007) Social network analysis for routing in disconnected delay-tolerant MANETs. In: Proceedings of ACM MOBIHOC, pp 32–40
4. Dimokas N, Katsaros D, Manolopoulos Y (2008) Cooperative caching in wireless multimedia sensor networks. ACM Mob Netw Appl 13(3–4):337–356
5. Dimokas N, Katsaros D, Tassiulas L, Manolopoulos Y (2011) High performance, low complexity cooperative caching for wireless sensor networks. ACM Wireless Netw 17(3):717–737
6. Dolev S, Elovici Y, Puzis R (2010) Routing betweenness centrality. J ACM 57
7. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174
8. Freeman LC (1977) A set of measures of centrality based on betweenness. Sociometry 40(1):35–41
9. Gao W, Li Q, Zhao B, Cao G (2009) Multicasting in delay tolerant networks: a social network perspective. In: Proceedings of ACM MOBIHOC, pp 299–308
10. Hui P, Crowcroft J, Yoneki E (2008) Bubble rap: social-based forwarding in delay tolerant networks. In: Proceedings of ACM MOBIHOC, pp 241–250
11. Hwang W, Kim T, Ramanathan M, Zhang A (2008) Bridging centrality: graph mining from element level to group level. In: Proceedings of ACM SIGKDD, pp 336–344
12. Katsaros D, Dimokas N, Tassiulas L (2010) Social network analysis concepts in the design of wireless ad hoc network protocols. IEEE Netw mag 24(6):23–29
13. Li F, Wu J (2009) LocalCom: a community-based epidemic forwarding scheme in disruption-tolerant networks. In: Proceedings of IEEE SECON
14. Nanda S, Kotz D (2008) Localized bridging centrality for distributed network analysis. In: Proceedings of 17th international conference on IEEE ICCCN, pp 1–6
15. Newman NEJ (2006) Modularity and community structure in networks. Proc National Acad Sci 103(23):8577–8582
16. Pallis G, Katsaros D, Dikaiakos MD, Loulloudes N, Tassiulas L (2009) On the structure and evolution of vehicular networks. In: Proceedings of IEEE MASCOTS, pp 502–511
17. Perra N, Fortunato S (2008) Spectral centrality measures in complex networks. Phys Rev E 78(036107):036107-1-03107-10
18. Pujol JM, Toledo AL, Rodriguez P (2009) Fair routing in delay tolerant networks. In: Proceedings of IEEE INFOCOM, pp 837–845