Middleware Solutions for the Internet of Things

Chinwei "Vic" Hu
Department of Electrical and Computer Engineering
The University of Texas at Austin

Abstract

In this paper, we briefly talk about what the Internet of Things (IoT) is, why do we care, and a few selection of the state-of-the-art middleware solutions to address some of the common problems IoT faces, such as heterogeneous coordination, context-awareness communication, resource-constrained and efficient computation. In particular, we look at Grapevine [6] and Gander [11] developed at The University of Texas at Austin.

1. Introduction

Big Data [9] and deep analytics are becoming prevalent in almost every industry, thanks to the invention of the world-wide web [3] and ubiquitous web services. To make better decisions in both our personal and professional life, we need to somehow weave together the information fabric across the virtual and physical domains, which introduces an extremely complex collection of sensors, protocols, context-awareness, constrained resources, real-time responsiveness, and distributed computing problems. All of these challenges converge to the notion of the Internet of Things paradigm [2].

The Internet of Things can be used to monitor the behavior and conditions of persons or machines, enhance real-time situational awareness of physical environment, assist better decision making based on data analytics and visualization, automatically and optimally control self-contained complex systems, and efficiently distribute resources in a grid network [5]. Nodes interacting in such grid network could be mobile phones, tablets, sensors, actuators, or essentially anything with some kind of communication beacon such as QR code or RFID tags [16](Radio-Frequency Identification). Due to its pervasive presence and limitless potentials, the US National Intelligence Council includes IoT in a list of "Disruptive Civil Technologies" with potential impacts on US national power [12].

The main purpose of this paper is to study a collection of middleware solutions in addressing the aforementioned challenges in the IoT, including Grapevine [6], Gander [11], and DAIS [7], all done at The Department of Electrical and Computer Engineering at The University of Texas at Austin. In particular, we will look at motivations and intuitions, design patterns, system architectures, and how they can be improved to be practically integrated into our works and life.

2   System Overview

One challenge in IoT is to address and aggregate context-awareness in surrounding neighborhood, such as traffic condition of nearby intersections, skill level and interests of players in a

group, or formulating a dynamic mobile network based on communication quality and available resources. Specifically in a dynamic pervasive network, we need to identify the notion of groups based on context predicates, and to define entity and group context to assess and share them in their surrounding computing environments. Motivated by this fundamental problem, Grapevine [6] addresses exactly this problem by introducing a mechanism to efficiently define and share context and group memberships in a network.

2.1 Grapevine

To define context summary, Grapevine uses the Bloomier filter data structure [4], which consists of a Bloom filter cascade with $O(rn)$ space complexity and false positive rate $\epsilon \propto 2^{-r}$, where n is the number of elements and r is the number o bits. On the other hand, a notion of groups is defined by a constraint function f_G of four types: *Labeled, Asymmetric, Symmetric*, and *Context-Defined*, which is combined with the context summary of entities to formulate expressive group membership relations.

As shown in Fig. 1, the *Context Handler* computes the context summaries with Bloomier filters to formulate groups based on the group definitions, while the *Context Shim* attaches the updated context summary to the outgoing packets and detaches the incoming context summary of the incoming packets, which uses the *interceptor* design pattern to work with the *Context Handler*. Furthermore, a static notion of τ is used to address the tradeoff between dissemination cost and the scope of shared context.

In implementation, Grapevine uses Java to gain transparency in deployment platform and fine-grained control over the network layer. The information flow of Grapevine, as shown in Fig. 2, illustrates how the *Context Handler* interacts with the application to compute group membership and context, to store a map of entity context, and to create Bloomier filters for the *Context Shim* to access the context summary. The performance and effectiveness of Grapevine are evaluated through a simulated network of 10-by-10 grid and the Pharos multi-robot testbed[1].

Figure 1. Grapevine System Architecture [6]
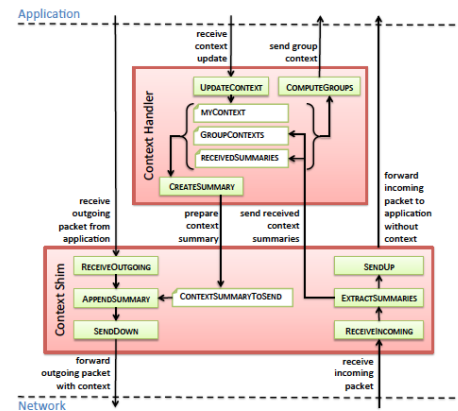
Figure 2. Information Flow in the Context Shim and Context Handler of Grapevine [6]

2.2 Gander

In contrast to the traditional indexing and ranking methods that require a global view of context connectivity and chronicle archives such as PageRank[13], the Gander search engine [11] provides an IoT perspective to the spatiotemporal queries in the *Personalized Networked Spaces* (PNetS). Challenges of searching in the here and now include the privacy and security of personal data, rapidly changing context and conditions, and the impracticality of centrally indexing a constantly growing amount of
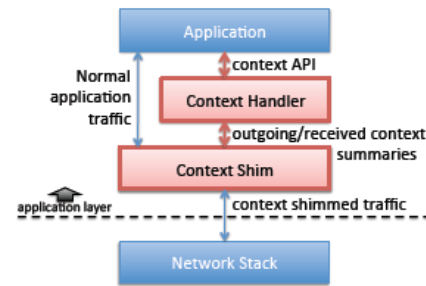
data. Moreover, by limiting a search within constrained neighborhood and spatiotemporal context, more efficient network and computing resources could be attained.

In order to match the *query resolution*, *query constraints* and *reachability* need to be defined to measure the *relevance metric*. Furthermore, a *tuple space* is used to provide the *global virtual data structure* [14] distributed in a dynamic network. Since it's impractical to obtain global view in PNetS, query protocols have to be operated within the surrounding data. To find the most relevant query results within a constrained neighborhood of nodes, Gander *samples* result candidates in five styles, which are illustrated in Fig. 3.
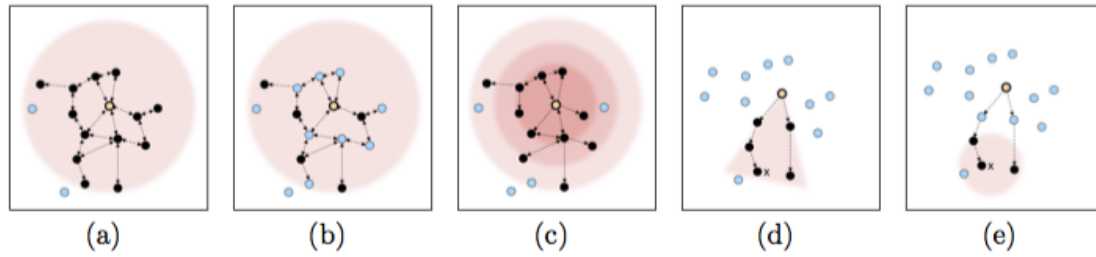

Figure 3. Query sampling styles[11]. Dashed lines represent sent messages, and darkened nodes are those who respond to a given query. (a) Flooding (b) Random (c) Probabilistic (d) Directional (e) Regional

Each of the five sampling styles demonstrates the tradeoff between quality and cost in network overhead and latency. *Flooding* attempts to reach every node within the network hops, *Random* parameterizes the likelihood of responses, *Probabilistic* forwards received messages based on a locality probabilistic distribution, *Directional* constraints the sampling within a certain heading and angle, and *Regional* aims at a given target area. Moreover, a secondary contextual queries are carried out from potential result's perspective, which collects local conditions without adding network overhead and delays.

The implementation of Gander, *myGander*[10], follows the model-view-controller architecture pattern as shown in Fig. 4. The model consists of a tuple space implementing the data structure, a routing table that maintains the connectivity in the network, a query processor that handles incoming queries to the tuple space, performs secondary contextual queries on the collected data and meta-data, and routes the results according to its sampling style.

To evaluate sampling protocols in a real world large scale network is extremely challenging. In practice, benchmark simulation for *myGander* can be done with OMNet++[15] for networking and SUMO[8] for node mobility, in which each node uses its own global virtual data structure to process queries and connects to other mobile node through a proxy.
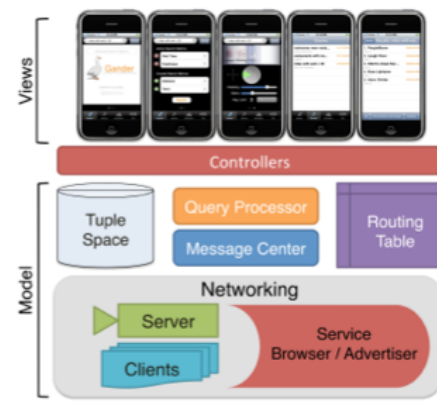

Figure 4. *myGander* **Architecture [11]**

2.3 DAIS

3 Architecture Design & Patterns

3.1 Gander: The Search of Here and Now

With the rise of sensor-enhanced objects and computationally-powerful mobile devices, the need for a dynamic search engine that is adaptive to the surrounding spaces and query context is inevitable.

Gander [10]

4 Current State, Drawbacks, and Future Improvements

Dynamic network hop τ in Grapevine.

5. Conclusion

Acknowledgment

References

[1] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2339–2345. IEEE, 2008.

[2] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[3] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The world-wide web. *Communications of the ACM*, 37(8):76–82, 1994.

[4] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal. The bloomier filter: an efficient data structure for static support lookup tables. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 30–39. Society for Industrial and Applied Mathematics, 2004.

[5] M. Chui, M. Löffler, and R. Roberts. The internet of things. *McKinsey Quarterly*, 2:1–9, 2010.

[6] E. Grim, C.-L. Fok, and C. Julien. Grapevine: Efficient situational awareness in pervasive computing environments. In *Submitted for consideration in an upcoming IEEE conference*, 2012.

[7] S. Kabadayi, C. Julien, and A. Pridgen. Dais: Enabling declarative applications in immersive sensor networks. Technical Report TR-UTEDGE-2006-000, The Center for Excellence in Distributed Global Environments, The University of Texas at Austin, 2006.

[8] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner. Sumo (simulation of urban mobility). In *Proc. of the 4th Middle East Symposium on Simulation and Modelling*, pages 183–187, 2002.

[9] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[10] J. Michel, C. Julien, J. Payton, and G.-C. Roman. mygander: A mobile interface and distributed search engine for pervasive computing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 495–497. IEEE, 2012.

[11] J. R. Michel. The gander search engine for personalized networked spaces. 2013.

[12] U. OFFICIAL. Disruptive civil technologies.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[14] G. P. Picco, A. L. Murphy, and G.-C. Roman. On global virtual data structures. *Process Coordination and Ubiquitous Computing*, pages 11–29, 2002.

[15] A. Varga et al. The omnet++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM?2001)*, volume 9, page 185. sn, 2001.

[16] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer, M. Balazinska, and G. Borriello. Building the internet of things using rfid: the rfid ecosystem experience. *Internet computing, IEEE*, 13(3):48–55, 2009.