Zeroties: a Publish/Subscribe Service for Applications in Local Ad-Hoc Networks

Chris Satterfield Felix Grund
University of British Columbia
Vancouver, BC

ABSTRACT

1 INTRODUCTION

Throughout the last decade we have seen the Internet become the most commonly used infrastructure for communication: regardless of physical location, people and their devices communicate via messengers and VoIP, and collaborate via live editing tools like, for example, Google Docs, Sheets, and, Slides. At the same time, we have seen some applications communicating over local area networks become increasingly common for certain scenarios like home entertainment (e.g. Google Chromecast, Apple Bonjour, Spotify Connect) or Wi-Fi printers. With the advancements of “smart” devices and the “Internet of Things” (IoT) it is likely that this trend will grow beyond these currently still largely hardened application domains. The adoption of standards that eliminate the burden of manual configuration of network devices further contributes to this movement.

One such standard that has received widespread usage is Zero-configuration networking [4] and its protocols mDNS [2] for service advertisement and DNS-SD for service discovery [1]. Using the *Zeroconf* protocols, devices can publish named services in the local network and discover such services automatically in an ad-hoc fashion. While most applications for Zeroconf networks are shipped with specific hardware (e.g. Google Chromecast dongle) there have recently been attempts to provide software environments for developers to enable them write their own applications on already existing hardware infrastructure. One such application was Mozilla FlyWeb[1], an addon for the Firefox browser that made it possible to advertise and discover services from within Web applications through a JavaScript API. In a previous project, we created *Successorships* [2], a JavaScript library exposing an easy-to-use API to build fault-tolerant Zeroconf web applications. Successorships was built on Mozilla Flyweb as one main part of its architecture. This decision confronted us with two significant problems:

- Successorships had a hard dependency on Mozilla FlyWeb. However, FlyWeb was declared abandoned by Mozilla even before we finished our work on our library.
- The implementation of the Zeroconf protocols was slow to a degree that made our library fairly unusable in practice.
- On top of the dependency on FlyWeb and a past developer version Firefox, FlyWeb itself contained a bug that made our library only available on MacOS.

In this paper, we introduce Zeroties, a platform-independent asynchronous publish/subscribe service for Zeroconf advertisement and discovery. We carefully reviewed different publish/subscribe designs [3] and implemented a communication scheme based on *asynchronous notifications*. In this sense, the operations exposed by Zeroties are as follows:

- **Publish**: publish a Zeroties service and advertise it in the local network
- **Subscribe**: listen for updates on the list of available Zeroties services

Zeroties is implemented as two parts: (1) a standalone OS-level application, and (2) addons for Chrome and Firefox that connect to this application. With our addon implementations for Chrome and Firefox we aim to show that our approach translates well between different browsers and does not share the restrictions of FlyWeb. Our browser addons expose an API to web applications that comprises the full service/discovery functionality of Zeroconf. As a result, we have successfully eliminated the ties that prevented Successorships from usage in practice.

To evaluate Zeroties, we first created the webapp-based presentation for this project using Successorships in combination with the Zeroties app and the addon for Google Chrome. Running the presentation on Chrome proved that we successfully broke ties with FlyWeb and Mozilla Firefox. Furthermore, we could see in this example application that recovery from server failures was significantly faster than with the previous version based on FlyWeb. We finally repeated the performance measurements from the Successorships project, in which we simulated 29 server failures and subsequent recoveries in a small mobile network. We found that the system recovered in average about three times as fast with Zeroties than with FlyWeb.

In summary, we make the following contributions:

- Zeroties, a asynchronous standalone publish/subscribe service for Zeroconf applications.
- Addons for Chrome and Firefox that make this service available to web applications.
- An evaluation based on a Zeroties sample application and a reiteration of the Successorship measurements indicating significantly improved performance.

The remainder of this paper is organized as follows: Section 2 outlines the problems with the implementation of Successorships and its dependency on FlyWeb that motivated this work. Section 3 presents the system model and design goals of Zeroties, before Section 4 describes its implementation. We evaluate Zeroties in Section 5 and suggest limitations and future work in Section 6. Section 7 situates our work in the context of related research and Section 8 concludes the paper.

[1] https://wiki.mozilla.org/FlyWeb
[2] https://github.com/ataraxie/successorships

2 PROBLEMS WITH SUCCESSORSHIPS

3 APPROACH

4 IMPLEMENTATION

5 EVALUATION

6 LIMITATIONS AND FUTURE WORK

7 RELATED WORK

[5]

8 CONCLUSION

REFERENCES

[1] S. Cheshire and M. Krochmal. Dns-based service discovery. RFC 6763, RFC Editor, February 2013.

[2] S. Cheshire and M. Krochmal. Multicast dns. Technical report, 2013.

[3] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, June 2003.

[4] E. Guttman. Autoconfiguration for ip networking: enabling local communication. *IEEE Internet Computing*, 5(3):81–86, May 2001.

[5] S. G. Hong, S. Srinivasan, and H. Schulzrinne. Accelerating service discovery in ad-hoc zero configuration networking. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 961–965. IEEE, 2007.