Wiseman - A Management and Deployment Approach for WSN Testbed Software

Torsten Teubler and Horst Hellbrück
Lübeck University of Applied Sciences
Electrical Engineering and Computer Science
Email: {teubler,hellbrueck}@fh-luebeck.de

I. Introduction

Wireless Sensor Networks (WSN) are setup by small battery powered microcontroller devices (nodes) intended to record real world data with attached sensors. WSN nodes are constrained in computational power, memory, and energy. Therefore, research focuses on algorithms fulfilling their task under these given constraints. WSN algorithms are developed with the help of network simulators and recently by using WSN testbeds for more realistic results. A testbed is an installation of a real WSN allowing testing and evaluation of algorithms on a target platform. Compared to an ad hoc deployment of nodes for an experiment nodes in a testbed are already deployed and can be programmed remotely and controlled in parallel by the testbed software.

Deployment of testbed software is a task which occurs more often than expected. For example configuration files need to be adapted and deployed to the testbed. Also the testbed software itself needs to be adapted to fit the needs of a testbed operator, more precisely the administrator of a testbed.

Our experience shows that a manual deployment is tedious and error prone. Therefore, we developed an automatic deployment approach and implement a solution for our testbed software Wisebed [1]. Our approach is generic and we expect it to work for other testbed software as well.

II. Related Work

Automated, centralized deployment of software is a part of Continuous Integration (CI)/Continuous Deployment (CD) philosophy. With CI software changes are checked in often and early in a production repository. CD additionally includes deployment of these changes as often and early. CI and CD are coupled with an automated test suite to ensure that faulty software isn't deployed. An example for highly adaptable CI and CD system is Jenkins [2].

We consider that a deployment approach for WSN testbed software can be built up on Jenkins as well but special plugins have to be developed. A management mechanism for installed software is not foreseen by Jenkins and there are no plugins for this implemented yet.

Beyond CI/CD systems there are management/administration systems for huge networks also supporting machine management and software deployment like LANDesk [3]. Here the same holds as for CI/CD systems. They are heavyweight (even more than CI/CD systems) and tailored to needs of business applications. LANDesk for example allows deployment of operating systems or software license monitoring but has no support for fine grained service management.

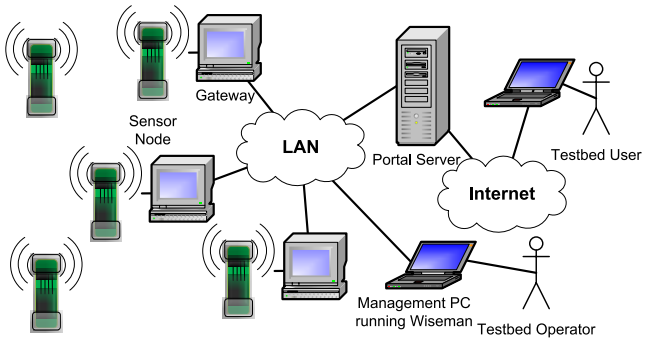

Fig. 1. Testbed Architecture with Wiseman

In our case a testbed operator performs modifications to the configuration and only minor modifications to the testbed software. A main task of a testbed operator is to deploy testbed software to testbed hardware. Therefore, we developed Wiseman as a slim system of our own tailored to the needs of WSN testbeds.

III. Concept

In the following we identified tasks for a deployment and management approach that we implemented in Wiseman. We additionally present configuration and design of Wiseman.

An Overview of a WSN testbed architecture with Wiseman is depicted in Figure 1. Wisebed's concept foresees a "portal server" as a central control unit and "gateways" as a WSN node control unit at a testbed site. In the following, we will refer to both kind as *testbed machines* if a distinction is not necessary.

A. Tasks

Deployment of files and folders: for first installation and update of testbed software an automatic and fast deployment of executables and configuration files is necessary. Extension of the testbed by adding more gateways and sensor nodes needs configuration changes to be deployed.

Backup/Restore mechanism: If a testbed update is deployed it is necessary to perform a backup of the actual state. If the

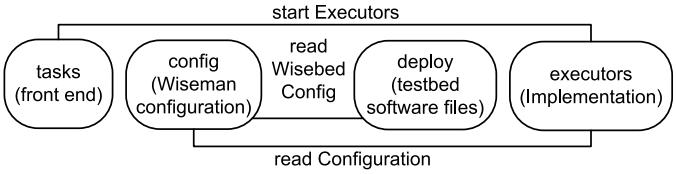

Fig. 2. Activities of Wiseman

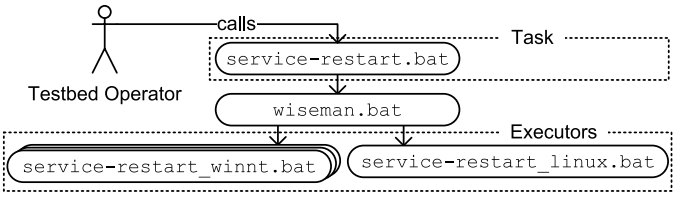

Fig. 3. Wiseman Usage Example

update is not working properly the previous state of the testbed can be restored easily.

Service management: Services of testbed software need management. For example in case of a testbed extension an automatic restart of services is necessary in Wisebed.

Testbed machine monitoring: Monitoring testbed machines health state is required including restart/shutdown testbed machines.

"Personalized deployment": Some configuration files are tailored to individual machines for example. Wiseman currently supports find and replace in configuration files for particular variables in configuration files.

Mounting network shares: For debugging purposes it is useful to mount network shares of testbed machines quickly.

In Wiseman it is possible to apply all tasks to a single, to a group, or to all testbed machines. This is useful for example if a service hung up on a single machine it can be restarted on this machine exclusively.

B. Configuration

We target to keep configuration of Wiseman simple. As Wisebed comes with a considerable amount of configuration files we reuse these configuration files in Wiseman. For example the list of testbed machines and their logon credentials are read by Wiseman.

Additional information for Wiseman is a network share which points to the location the testbed software resides on testbed machines. This network share needs to be created on all machines for Wiseman during first installation.

C. Design

Wiseman's activities are depicted in Figure 2. The four building blocks *tasks*, *config*, *deploy*, and *executors* represent the folder structure of Wiseman. Tasks represent the frontend or user interface of Wiseman. Config provides access for the executors to Wisebed's configuration and stores Wiseman's configuration. Deploy is an assembly of files of the testbed software (executables and configuration files). Executors are the backend of Wiseman performing the work. Executors are started by tasks and are configured by files placed in config.

IV. Implementation

As stated in Section I Wiseman is implemented as a collection of batch files according to our conceptual design in Figure 2.

In our testbed we run a heterogeneous environment. Gateways run Windows because software for programming our WSN nodes is available for that OS exclusively. In contrast, the portal server runs Linux. Therefore, a requirement for Wiseman is to handle this heterogeneous OS environment. To the best of our knowledge Windows machines can be controlled only by other Windows machines with acceptable effort. Therefore, we decided to use a Windows machine running Wiseman using SysInternals Suite [4] to manage our gateways running Windows. Linux machines can be controlled easily by a Windows machine using Cygwin [5] and tools like ssh, scp etc.

Configuration on the testbed machines is kept to a minimum to achieve a low configuration and setup overhead.

A task is a batch file called by the testbed operator. It follows a certain pattern which is the same to all tasks to enable simple creation of new tasks. The second component of Wiseman is a batch file simply called *wiseman* which calls an executor for each testbed machine. An exemplary call is depicted in Figure 3. Like tasks all executors follow a certain pattern and contain a part where custom code is added (e.g. for restarting services).

V. Conclusion

In this work we presented Wiseman a slim management and deployment approach targeted for WSN testbeds. Before we implemented Wiseman, deployment with Remote Desktop on each Windows machine or with ssh on Linux machines was a tedious task even for a small testbed of several machines. Now we operate a larger testbed with Wisemans deployment and management features. A deployment takes now only a few minutes compared to one or two hours before.

In future work will cover advanced logging functionality which is not fully implemented yet.

Acknowledgment

This work was funded by the Federal Ministry of Education & Research of the Federal Republic of Germany (01BK0906, GLab) and (17PNT017, DataCast). The authors alone are responsible for the content of this work.

References

[1] H. Hellbrück, M. Pagel, A. Kröller, D. Bimschas, D. Pfisterer, and S. Fischer, "Using and operating wireless sensor network testbeds with wisebed," in *Proceedings of the 10th IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop*, Favignana Island, Sicily, Italy, Jun. 2011.
[2] K. Kawaguchi et al., "http://jenkins-ci.org/," Website, January 2013.
[3] LANDesk Software, "http://www.landesk.com/," Website, January 2013.
[4] M. Russinovich and B. Cogswell, "http://technet.microsoft.com/en-us/sysinternals/," Website, January 2013.
[5] Red Hat Inc., "http://www.cygwin.com/," Website, January 2013.