Demo Abstract:
Cascades: An Extensible Heterogeneous Sensor Networking Framework

Phillip Sitbon, Nirupama Bulusu, Wu-Chi Feng
Portland State University
{sitbon, nbulusu, wuchi}@cs.pdx.edu

ABSTRACT
This demonstration shows a powerful high-level, heterogeneous sensor networking framework, *Cascades*. We intend to demonstrate how, with this framework, application designers have great control over implementation designs without the requirement of in-depth development. Several key components and example applications will be demonstrated, along with some important concepts used in this Python-based framework. The emphasis will be on video sensing and application control in heterogeneous sensor networks consisting of Crossbow Stargate (PC-class) and MicaZ (mote-class) devices.

Categories and Subject Descriptors
D.0 [**Software**]: General.
J.0 [Computer Applications]: General.

General Terms
Management, Design.

Keywords
Frameworks, Heterogeneous Sensor Networking

1. CASCADES OVERVIEW
Heterogeneous, or hybrid, sensor networks are becoming more common for the purpose of easing constraints on homogeneous sensor networks while increasing scalability. More capable devices, such as the Stargate Gateway[2] for example, can be used as fixed and reliable communication, storage and processing points in a sensor network in order to expand its abilities. More capable sensors such as video cameras add ability to high-level sensor network devices, especially in enabling robust event detection.

The availability of an extensible and intuitive set of tools and concepts has the potential to fulfill a wide variety of application and system needs in hybrid sensor networks. Some of the key challenges to network organization and management due to increased scale and heterogeneity include deployability, programmability, management, and retaskability. These challenges are especially unique in hybrid sensor networks because devices can no longer be treated the same, meaning that applications will only be effective when exploiting specific differences of a system.

There are inherent tradeoffs in meeting these challenges. For example, programmability is achieved through abstraction, which inevitably compromises performance. Designing a system for performance hinders deployability and retaskability.

Cascades, a modular framework in the scripted language *Python*[1], gives application designers great control over implementation designs without the requirement of in-depth development.

Cascades combines highly optimized code segments together in the form of Python modules. This architecture has a number of useful properties. First, it allows us to create highly-optimized, efficient code segments for computationally expensive tasks such as JPEG compression, MPEG compression, or image processing. Second, the Python scripting language allows us to seamlessly stitch code segments together. Third, because the system is scripted and modular, modifying or retasking the sensor system requires us to simply update the parts of the system that have changed, reducing network bandwidth required for management. Finally, the scripting language is at a high enough layer such that it allows us to hide many computer science specific optimizations (e.g. video buffering and adaptation or basic networking) from non-computer scientists.

We consider hierarchical sensor organization in the context of event detection in order to introduce the concept of *Logical Disassociation (LD)*: the process of creating automatic disassociations between different types of sensors by generating a known event and observing differences in their responses. This allows sensor devices that would normally use network connectivity for determining association to refine the conditions under which events should be considered related. This concept is implemented as an integrated semi-transparent component. Figure 1 gives an example scenario that will be used in the demonstration.

Mote-class devices such as the MicaZ[3] typically have lower wireless communication power as a result of their low-power design. To augment their limited range, we provide another semi-transparent component called the *Mote Abstraction Layer (MAL)*, which relays received messages received from motes among networks of more capable devices. Due to the inherent complexity of routing messages within a (possibly mobile) network of wireless and/or wired relay nodes, MAL was constructed in such a way that it can run with or without the help of lower-layer routing algorithms, i.e. AODV[4], in a set of simple configurations that trade overhead for simplicity. An example usage of MAL can be seen in Figure 2.

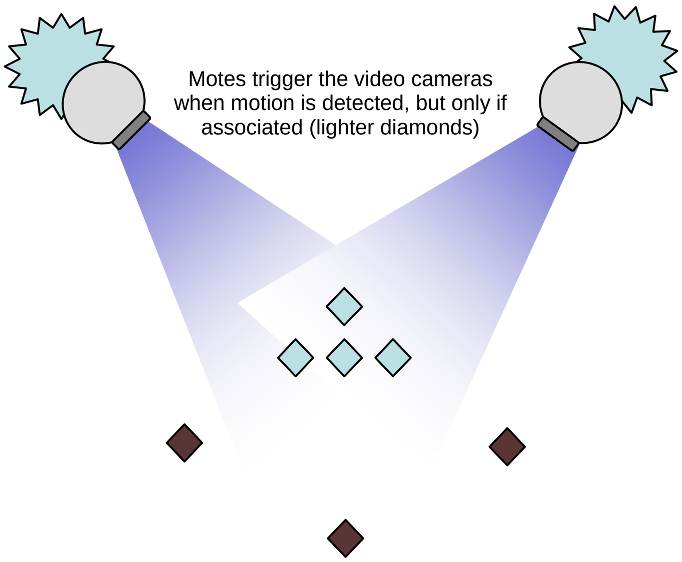

Figure 1. Using LD for motion/light triggered video.

We have built several applications using Cascades and evaluated them. Our results show that while the script-based architecture and abstraction of motes in Cascades provides easy to program interfaces and allows us to combine multimodal sensing in flexible ways, the system maintains acceptable levels of performance.

2. DEMONSTRATION OVERVIEW

In this demonstration, we will show the flexibility of the Cascades framework through the selection, deployment, and utilization of various components and end-user applications.

A subset of the demonstration will be dedicated to showing how the framework abstracts communication among network nodes and sensor devices, including our implementation of the TinyOS framed packet protocol. Furthermore, we will demonstrate the interoperability of such applications by running the same scripts on multiple platforms (Stargate, PC, Mac) as well.

Also, we will be demonstrating the concept of *Logical Disassociation*, primarily coordinating change-in-light events on MicaZ motes with video motion events on Stargate devices. Audio will be used in a smaller example where audio events are temporally correlated to determine association. Communication of mote-level events between Stargates, motes and PC's will be handled by the *Mote Abstraction Layer (MAL).*

Finally, we will demonstrate our management interface by using it to deploy, monitor, and update applications running at multiple network locations. This includes dynamic updating of active applications without interruption.

The goal of this demonstration is to convey the flexibility and usability of Cascades for both researchers and end-users alike. The components of the framework will be covered in detail and customized during the demonstration in order to convey the customizability of the framework.

3. REFERENCES

[1] Python. 12 Jun. 2005 <http://python.org>.

[2] "Stargate Gateway (SPB400)." Crossbow Technology. 12 Jun. 2005 <http://www.xbow.com/Products/productsdetails.aspx?sid=85>.

[3] "MICAz ZigBee Series (MPR2400)." Crossbow Technology. 12 Jun. 2005 <http://www.xbow.com/Products/productsdetails.aspx?sid=101>.

[4] C. E. Perkins and E. M. Royer. "Ad-hoc On-Demand Distance Vector Routing". Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.

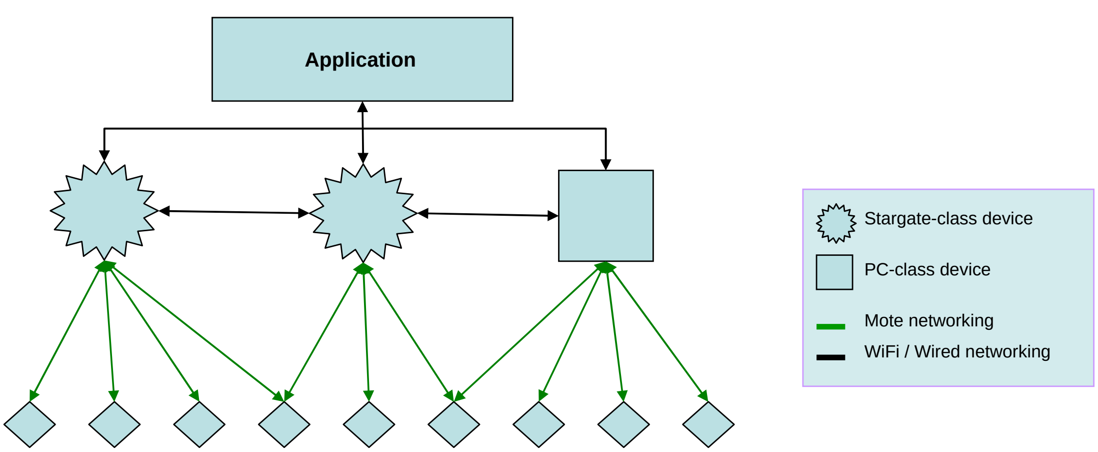

Figure 2. Example MAL Network.