The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
12-14 September 2013, Berlin, Germany

Ad-hoc Media Façade

Pavlo Ilin[1] , Jürgen Sieck[2] , Volodymyr Brovkov[1]
[1] Odessa National Polytechnic University, Ukraine
[2] University of Applied Sciences Berlin, Germany

***Abstract* – Present paper addresses the use of wireless sensor-actor networks (WSAN) with application to the task of design ad-hoc media facade system. Discussed the requirements for system architecture. Proposed architecture was evaluated using prototype.**

***Keywords* – media facade; wireless networks: sensor systems; ad-hoc systems**

I. Introduction

Goal of our research is to build the ad-hoc media facade system (AMF). AMF is a system, which consists of small wireless devices showing the image. The devices can be stationary (for example, fixed on the wall) or movable (for example, users move them to change the image). New devices can be introduced or existing devices can be removed at any time.

The description of such system can be reformulated in terms of Wireless Sensor-Actor Networks (WSAN). WSAN is is relatively new and emerging field of researches.

WSAN is a group of the devices, which play two different roles - sensors and actors. They are interconnected by a one wireless media. Sensors gather information about physical world and pass it to the actors. Actors gather and process the information from sensors and, using it, take decisions and perform actions on a physical world. Sensing and acting tasks are performed in distributed manner. The operation of a WSAN can be thought of as a loop of event-sensing, communication, decision, and acting.

Research tasks for WSAN are addressed in [1]. Among them are: system architecture choice, effective sensor-actor and actor-actor interaction coordination, task assignment, networking stack protocols selection.

Usually, real-time requirement is applied to WSAN. This requirement to the network is closely related to the network Quality of Service (QoS). Network QoS can be accepted as a measure of the service quality that the network offers to the applications/users [2].

Xia [3] gives an overview of the problems of QoS management in WSAN. For the real-time WSAN four QoS parameters can be considered as fundamental: throughput, packet loss ratio, delay and jitter.

- Throughput (or bandwidth) is the effective number of data flow transported within a certain period of time.
- Packet loss ratio is the percentage of data packets that are lost during the transmission.
- Delay is the time between a packet departure from source node and packet arrival at the destination node.
- Jitter is the variations in delay.

During WSAN system design, next challenges can prevent QoS provision: resource constraints, platform heterogeneity, dynamic network topology, mixed traffic.

WSAN technology is a descendant of Wireless Sensor Networks (WSN), so some research problems are the same for WSN and WSAN. But, heterogeneous WSAN with multi-source multi-sink topology model appears to be quite different from a traditional data collection WSN model with single static sink [4]. Thus, the solutions to the research problems can be different.

II. System architecture

Xia *et al.* [5] show two possible basic architectures of WSAN: with the dedicated controller or without.

For our system, we will use architecture with dedicated controller. Thereby, the system (Fig. 1) consists of the next components:

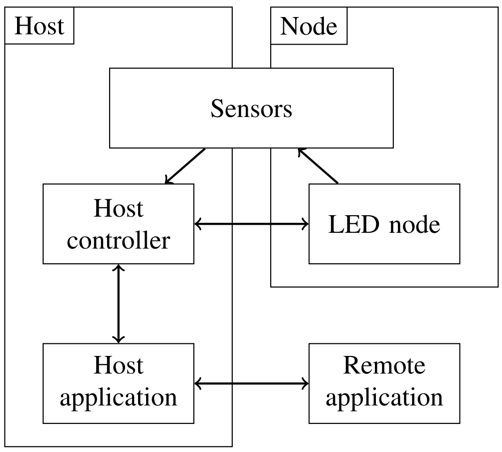

Figure 1. Overview of a system architecture

- LED node - the actor, placed in event area; it forms the image;
- sensors - they detect the position of LED nodes, using video processing and pattern recognition or any

other kind of position detection technics. They can be installed on host and LED node sides.
- host-controller - a system, which decides, how to split an image between actors (task assignment). To be able to do that, it should know, where LED nodes are located.
- host-application - application, which generates images to be shown using LED nodes;
- remote-applications - remote clients, which provide interactive control.

Sensors are, usually, assumed to be highly resource-constrained devices. They are designed as autonomous nodes, equipped with a battery. They can be installed in the places, which are prevent or complicate a maintenance. To decrease power consumption, sensor nodes can be implemented using low-performance CPUs. Actors assumed to be less constrained. They are bigger, have more powerful CPU and are connected to the power supply or equipped with better batteries as they have to interact with physical world.

In our system, the situation is different. As we are going to use computation-hungry image recognition algorithms, sensor is going to be powerful system (for example, ordinary PC). We can assume, that the sensors are powerful devices without energy constrains. On the other hand, actors (LED nodes) have to be small devices build on embedded platform. Embedded platform means, that actors have constrained computational power. If actors have to be movable, they will be powered by a battery. This requirement imposes strict energy constraints, which have to be considered by network routing and task assignment algorithms.

Our system should perform a real-time control of a LED-nodes. It means, system should update overall image in syncronized manner and with sufficient for application speed. Network protocol should be QoS-aware.

An overview of radio-frequency wireless technologies is provided in [6], [7]. Among other wireless technologies, such as WiFi, Bluetooth and Bluetooth LE, ZWave, ANT, most promising to us appears to be IEEE 802.15.4.

IEEE 802.15.4 is an open standard, which describes PHY and MAC network layers. It is used in the higher level technologies such as ZigBee, 6LowPAN. IEEE 802.15.4-based solutions are widely used in WSN systems. There is big community of researches and engineers, who use IEEE 802.15.4. Thus, it have to be easier to use previous experience and community support. Closest competitors to IEEE 802.15.4 are ZWave and ANT. Both provide full network stack up to application layer. But they are proprietary and haven't got such a wide use.

Remote applications provide interactive control of the AMF. They can be an applications on mobile devices, clients to web-page (if host application provides an access using web-server), etc. Remote applications communicate with host application using standard LAN/WLAN and Internet.

III. Prototype description

The system prototype (Fig. 2, Fig. 3) has been built to study the system use cases and evaluate different technologies.

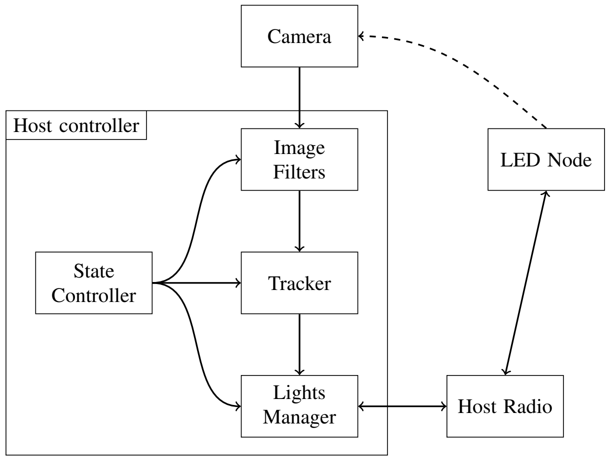

Figure 2. Prototype architecture

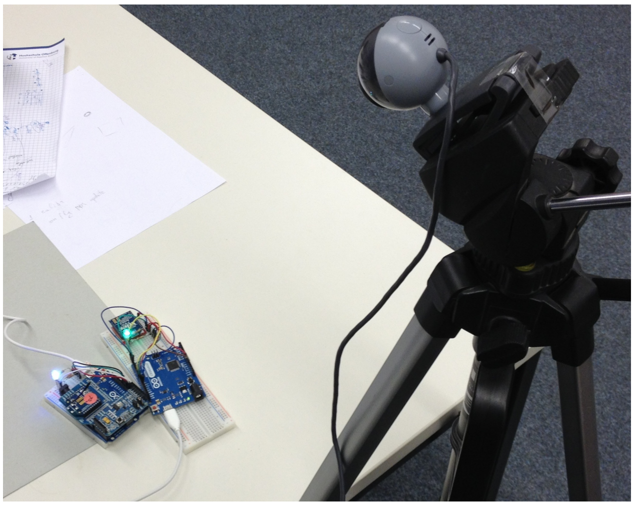

Figure 3. Prototype installation: 2 LED-nodes and video camera

From the hardware point of view, system consists of 2 components - host and LED-nodes.

LED-node platform is build using Arduino Leonardo board, one RGB LED and an IEEE 802.15.4 transceiver. RGB LED is used to produce a "pixel" of the image. During the tests, we have tried to use two different tranceivers:

- Digi XBee S2, which implements ZigBee protocol stack;

- Microchip MRF24J40MA, which implements plain IEEE 802.15.4 standard.

Host consists of PC, video camera and host-transceiver. Video camera is used as a sensor.

Host-transceiver is build from Arduino board with radio transceiver. It is used to establish a network between Host and LED-nodes. Installed transceiver is the same as on the LED-node.

Software is divided into 3 main parts: host-controller, host-transceiver firmware and LED-node firmware.

Host-controller consists of next components:

- sensor data processing:
  - image filter;
  - lights tracker.
- controller:
  - state controller;
  - lights manager.

Host-tranceiver firmware works as a repeater. All traffic from host-controller is sent to the LED nodes using radio tranceiver and vice versa.

LED-node firmware is capable to:

- control LED color and change it depending on a node's position;
- response to host beacon requests during the node discovery.

Host application and remote applications aren't introduced in this prototype.

IV. Algorithmic basis

During the process of operation, the system performs next tasks:

- image filtering;
- lights labeling;
- lights tracking;
- light calibration;
- light position update.

Image filter receives a frame from video camera, performs previous processing of the image, filters-out the noises and distinguishes possible lights. Image filter is build using OpenCV library. It select bright areas on the image, using threshold filtration on Value channel in HSV color space. Then it applies morphological operation erosion to remove little areas left from noise. Left areas (blobs) are assumed to be possible lights.

Lights tracker assigns IDs to lights (labeling), tracks them in video sequence (tracking) and detects their position. ID is assigned on the stage of a new light calibration. Afterwards, tracker maintains actual information about light position and passes this information to lights manager. Tracker is build using cvBlob library. For labeling it uses algorithm based on contour tracing technique [8].

Controller is event-driven. For event-driven programming boost::asio library has been used. It allows to process I/O operations and schedule deferred actions.

State controller react to I/O and timer events and issues commands to other components of host. State controller is modeled using Finite State Machine. FSM is coded using boost::msm library. This library allows to easily map UML representation of FSM to the source code.

New light calibration (Fig. 4) is performed by controller together with lights tracker. When the LED-node is discovered, state controller switches to the calibration state. New node is commanded to enable light and then to disable. Lights tracker remembers all vanished blobs. After specific amount of time (specific amount of frame updates), node is commanded to enable light back. It is assumed, that node is not moving during calibration stage. Lights tracker marks all blobs which have appeared back. If there are only one such blob, light is identified and labeled. Otherwise, calibration is performed again or discarded.

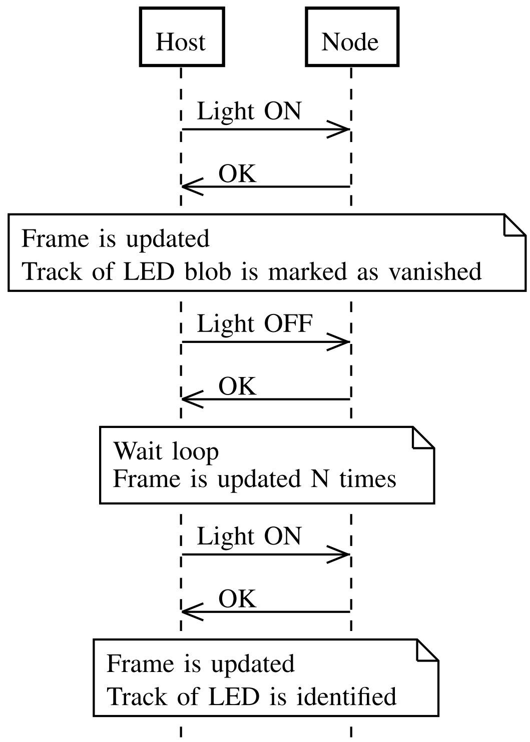

Figure 4. Sequence diagram of calibration stage

V. Experiments with prototype

Evaluation network of 4 LED-nodes and a host were used for the next experiments:

- new LED-node registration and calibration;
- LED-node position update.

Experiments were done twice with different radio tranceivers - XBee S2 and MRF24J40MA.

For the new LED-node registration and calibration, next algorithm have been used. Host sends periodically a beacon-package to all nodes in the network using broadcast. This beacon-package is only package sent by broadcast - all other communication are done using unicast messages. All nodes obtain host address from received beacon-package and answer to host. Host checks all answers, selects nodes with unknown addresses and performs calibration of them. During the calibration procedure, host sends control requests to the LED-node and awaits responses-acknowledgments.

Experiment with XBee S2 had shown, that broadcast messages are highly unreliable and can lead to network unresponsiveness. Sometimes, sent broadcast messages block all other send/receive operations on host. Host drops received answers. Thus, new nodes can stay undetected for a few detection rounds. Moreover, this problem breaks the communication with known nodes.

Experiment with MRF24J40MA had shown no such problems. Broadcasts show no issues with the blocked communiccation.

For LED-node position update, host sends messages with new position coordinates to the LED-node. Messages are sent, only if node have been moved on a distance, bigger than the specific threshold. LED-node sends response-acknowledgment. To make an experiment more visual, whole coordinate space was divided to the predefined color areas. LED node changes LED color, when it is moved to another color area.

Experiment with XBee S2 had shown, that the level of jitter is high. A single message can be delivered in specified time constraints. But, in case of a frequent messages issue, delay increases nonlineary and can reach the values over 1 second. Such situation appears when LED-node is moved fast and its position is updated frequently. Problem is due to the acknowledgments both on application and ZigBee levels.

Experiment with MRF24J40MA had shown better results. We have not observed avalanche-like increase of delivery time.

XBee S2 uses the same PHY and MAC layers as the MRF24J40MA. So we can explain poorer performance results only because of the use of ZigBee.

VI. Summary and future work

This work addresses the task of creation ad-hoc media facade system based on WSAN. The task imposes certain QoS requirements on network hardware and protocol characteristics. The hardware characteristics are different from those, which are usually considered in related works.

Simplified system prototype has been created for use cases and problems study.

We are currently studying a few related problems:

- network protocols such as ZigBee and its competitors;
- use of complex network topologies and their influence on QoS;
- algorithms of delay compensation such as delay prediction on host or control command prediction on actor nodes.

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351 – 367, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870504000319

[2] D. Chen and P. K. Varshney, "Qos support in wireless sensor networks: A survey," in *International Conference on Wireless Networks*, H. R. Arabnia, L. T. Yang, and C.-H. Yeh, Eds. CSREA Press, 2004, pp. 227–233.

[3] F. Xia, "Qos challenges and opportunities in wireless sensor/actuator networks," *Sensors*, vol. 8, no. 2, pp. 1099–1110, 2008. [Online]. Available: http://www.mdpi.com/1424-8220/8/2/1099

[4] Y. Li, C. S. Chen, Y.-Q. Song, and Z. Wang, "Real-time QoS support in wireless sensor networks: a survey," in *7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems - FeT'2007*, Toulouse, France, 2007. [Online]. Available: http://hal.inria.fr/inria-00188265

[5] F. Xia, Y.-C. Tian, Y. Li, and Y. Sung, "Wireless sensor/actuator network design for mobile control applications," *Sensors*, vol. 7, no. 10, pp. 2157–2173, 2007. [Online]. Available: http://www.mdpi.com/1424-8220/7/10/2157

[6] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, no. 9, pp. 6869–6896, 2009. [Online]. Available: http://www.mdpi.com/1424-8220/9/9/6869

[7] A. Garcia-Hernando, *Problem Solving for Wireless Sensor Networks*, ser. Computer Communications and Networks. Springer-Verlag London, 2008. [Online]. Available: http://books.google.com.ua/books?id=kr3sLEJKrpwC

[8] C. C. Liñán, "cvblob frequently asked questions," 2012. [Online]. Available: http://code.google.com/p/cvblob/wiki/FAQ