Wireless Video Sensor Network and Its Applications in Digital Zoo

Johannes Karlsson

Doctoral Thesis, December 2010
Department of Applied Physics and Electronics
Umeå University, Sweden

Digital Media Lab
Department of Applied Physics and Electronics
Umeå University
SE-901 87 Umeå, Sweden

Thesis submitted to Umeå University for the degree of Doctor of Philosophy (Ph.D.) in Applied Electronics.

Akademisk avhandling för avläggande av teknologie doktorsexamen (Tekn. Dr.) vid Umeå universitet i tillämpad elektronik.

ISSN 1652-6295:13
ISBN 978-91-7459-116-3

Author's email: johannes.karlsson@tfe.umu.se
Typeset in LaTeX by Johannes Karlsson
Printed by Arkitektkopia AB, Umeå University, Umeå, December 2010.

Wireless Video Sensor Network and Its Applications in Digital Zoo

Johannes Karlsson
Department of Applied Physics and Electronics, Umeå University, Sweden.

ABSTRACT

Most computing and communicating devices have been personal computers that were connected to Internet through a fixed network connection. It is believed that future communication devices will not be of this type. Instead the intelligence and communication capability will move into various objects that surround us. This is often referred to as the "Internet of Things" or "Wireless Embedded Internet". This thesis deals with video processing and communication in these types of systems.

One application scenario that is dealt with in this thesis is real-time video transmission over wireless ad-hoc networks. Here a set of devices automatically form a network and start to communicate without the need for any previous infrastructure. These devices act as both hosts and routers and can build up large networks where they forward information for each other. We have identified two major problems when sending real-time video over wireless ad-hoc networks. One is the reactive design used by most ad-hoc routing protocols. When nodes move some links that are used in the communication path between the sender and the receiver may disappear. The reactive routing protocols wait until some links on the path breaks and then start to search for a new path. This will lead to long interruptions in packet delivery and does not work well for real-time video transmission. Instead we propose an approach where we identify when a route is about to break and start to search for new routes before this happen. This is called a proactive approach. Another problem is that video codecs are very sensitive for packet losses and at the same time the wireless ad-hoc network is very error prone. The most common way to handle lost packets in video codecs is to periodically insert frames that are not predictively coded. This method periodically corrects errors regardless there has been an error or not. The method we propose is to insert frames that are not predictively coded directly after a packet has been lost, and only if a packet has been lost.

Another area that is dealt with in this thesis is video sensor networks. These are small devices that have communication and computational capacity, they are equipped with an image sensor so that they can capture video. Since these devices in general have very limited resources in terms of energy, computation, communication and memory they demand a lot of the video compression algorithms used. In standard video compression algorithms the complexity is high for the encoder while the decoder has low complexity and is just passively controlled by the encoder. We propose video compression algorithms for wireless video sensor networks where complexity is reduced in the encoder by moving some of the image analysis to the decoder side. We have implemented our approach on actual low-power sensor nodes to test our developed algorithms.

Finally we have built a "Digital Zoo" that is a complete system including a large scale outdoor video sensor network. The goal is to use the collected data from the video sensor network to create new experiences for physical visitors in the zoo, or "cyber" visitors from home. Here several topics that relate to practical deployments of sensor networks are addressed.

Keywords: Wireless Sensor Networks, Wireless ad-hoc Networks, Digital Zoo, Video Compression, Real-Time Video Communication, Object Tracking, Sensor Fusion.

SAMMANFATTNING

Hittills har vi mest använt persondatorer anslutna till Internet för att behandla och skicka information. I framtiden kommer dock inte de flesta kommunicerande enheter vara av denna typ. I stället kommer intelligens och kommunicerande egenskaper att flytta in i olika objekt som omger oss och vi kommer att få ett "Prylarnas Internet" eller som det kallas på engelska ett "Internet of things". Den här avhandlingen handlar om behandling och kommunikation av videosignaler i denna typ av system.

Ett område som behandlas i avhandlingen är överföring av realtidsvideo i trådlösa ad-hoc nätverk. Den här typen av nät behöver inte någon tidigare installerad infrastruktur för att kommunicera utan enheterna bygger själva upp det nät som behövs och hjälper varandra att vidarebefordra information. Vi har identifierat två huvudproblem för videoöverföring i trådlösa ad-hoc nätverk. Det första är att de routingprotokoll som oftast används väntar till dess att en länk på den rutt som används försvinner och först då börjar att söker efter en ny länk. Vi förslår i stället att systemet identifierar länkar som håller på att brytas redan innan de bryts och att systemet därigenom tidigare kan börja söka efter nya rutter. Det andra problemet hänger ihop med att överföringen av komprimerade videosignaler är mycket känslig för dataförluster. Samtidigt har trådlösa ad-hoc nätverk fler paketförluster vilket tillsammans med videosignalens känslighet ställer till problem. En metod som nu används för att hantera dataförsluser i videoöverföring är att periodiskt skicka bildrutor som ännu inte har komprimerats med hänsyn till tidigare bildrutor. Med denna metod lagas fel periodiskt, oavsett om det har varit något fel eller inte. Vi föreslår i stället att man bara ska skicka denna typ av bildrutor direkt efter att ha tappat datapaket, och endast i dessa fall.

Ett annat område som behandlas i avhandlingen är trådlösa videosensornätverk. De är utrustade med en bildsensor så att de kan filma video och kan genomföra beräkningar och kommunicera. Eftersom enheterna har mycket begränsade resurser sett till energi, beräkningskapacitet, kommunikationskapacitet, och minne så ställs stora krav på algoritmen som används för videokomprimering. I de komprimeringsalgoritmer som idag används i de vanligaste typerna av videokodningar så krävs mycket mer resurser för att komprimera en video än för att spela upp den. Vi föreslår en ny typ av videokomprimeringsalgoritm där vi reducerar komplexiteten för komprimering genom att flytta en del av bildanalysen till mottagaren. Vi har implementerat och testat algoritmen på verkliga sensorenheter i ett trådlöst nät.

Slutligen har vi byggt en "Digital Djurpark" som är ett komplett system där det ingår ett storskaligt trådlöst videosensornätverk i utomhusmiljö. Målet är att använda den insamlade informationen från det trådlösa videosensornätverket för att skapa nya upplevelser både för fysiska besökare i djurparken och för "cyber-besökare" från hemmen. I arbetet behandlas många ämnen som berör praktiska installationer av trådlösa sensornätverk.

Contents

Preface

This thesis consists of the following articles:

I. Johannes Karlsson, Jerry Eriksson and Haibo Li, "**Real-time video over wireless ad-hoc networks**," *in Proceedings of the fourteenth International Conference on Computer Communications and Networks (ICCCN 2005)*, 17-19 October 2005, San Diego, California, USA.

II. Johannes Karlsson, Jerry Eriksson and Haibo Li, "**Two hop connectivity for uniformed randomly distributed points in the unit square**," *in Proceedings of the First International Conference on Communications and Networking in China (CHINACOM 2006)*, 25-27 October 2006, Beijing, China.

III. Johannes Karlsson, Mikael Israelsson, Jerry Eriksson and Haibo Li, "**Efficient P2P Mobile Service for Live Media Streaming**," *in Proceedings of the Australian Telecommunication Networks and Applications Conference (ATNAC 2006)*, 4-6 December 2006, Melbourne, Australia.

IV. Johannes Karlsson, Adi Anani and Haibo Li, "**Enabling Real-Time Video Services Over Ad-Hoc Networks Opens the Gates for E-learning in Areas Lacking Infrastructure**," *International Journal of Interactive Mobile Technologies*, 2009.

V. Tim Wark, Peter Corke, Johannes Karlsson, Pavan Sikka and Philip Valencia, "**Real-time Image Streaming over a Low-Bandwidth Wireless Camera Network**," *in Proceedings of the Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2007)*, 3-6 December 2007, Melbourne, Australia.

VI. Johannes Karlsson, Jean-Paul Kouma, Haibo Li, Tim Wark and Peter Corke, "**Demonstration of Wyner-Ziv Video Compression in a Wireless Camera Sensor Network**," *in Proceedings of the 9th Scandinavian Workshop on Wireless Ad-hoc &*

Sensor Networks (ADHOC'09), 4-5 May 2009, Uppsala, Sweden.

VII. Johannes Karlsson, and Haibo Li, "**Moving Motion Estimation from Encoder for Low-Complexity Video Sensors**," *manuscript to be submitted.*

VIII. Karin Fahlquist, Johannes Karlsson, Keni Ren, Li Liu, Haibo Li, Shafiq ur Réhman and Tim Wark, "**Human Animal Machine Interaction: Animal Behavior Awareness and Digital Experience**," *in Proceedings of ACM Multimedia 2010 - Brave New Ideas*, 25-29 October 2010, Firenze, Italy.

IX. Johannes Karlsson, Tim Wark, Keni Ren, Karin Fahlquist and Haibo Li, "**Applications of Wireless Visual Sensor Networks - the Digital Zoo**," *in Visual Information Processing in Wireless Sensor Networks: Technology, Trends and Applications*, IGI Global, in press

X. Johannes Karlsson, Keni Ren and Haibo Li, "**Tracking and Identification of Animals for a Digital Zoo**," *in Proceedings of The 1st IEEE/ACM Internet of Things Symposium*, 18-20 December 2010, Hangzhou, China.

All the contributions that have been published during the PhD program are listed here:

1. Johannes Karlsson, Shafiq ur Réhman and Haibo Li, "**Augmented Reality to Enhance Vistors Experience in a Digital Zoo**," *in Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia (ACM MUM 2010)*, 1-3 December, 2010, Limassol, Cyprus.

2. Johannes Karlsson, Tim Wark, Keni Ren, Karin Fahlquist and Haibo Li, "**Applications of Wireless Visual Sensor Networks - the Digital Zoo**," *in Visual Information Processing in Wireless Sensor Networks: Technology, Trends and Applications*, IGI Global, in press

3. Johannes Karlsson, Keni Ren and Haibo Li, "**Tracking and Identification of Animals for a Digital Zoo**," *in Proceedings of The 1st IEEE/ACM Internet of Things Symposium*, 18-20 December 2010, Hangzhou, China.

4. Karin Fahlquist, Johannes Karlsson, Keni Ren, Li Liu, Haibo Li, Shafiq ur Réhman and Tim Wark, "**Human Animal Machine Interaction: Animal Behavior Awareness and Digital Experience**," *in Proceedings of ACM Multimedia 2010 - Brave New Ideas*, 25-29 October 2010, Firenze, Italy.

5. Johannes Karlsson, Adi Anani and Haibo Li, "**Enabling Real-Time Video Services Over Ad-Hoc Networks Opens the Gates for E-learning in Areas Lacking Infrastructure**," *International Journal of Interactive Mobile Technologies*, 2009.

6. Johannes Karlsson and Adi Anani, "**Ad-hoc Networks as a tool for E-Learning**," *in Proceedings of the International Workshop on New Achievements in the e-Learning's Domain (WAEL'09)*, 16-18 June 2009, Umeå, Sweden.

7. Johannes Karlsson, Jean-Paul Kouma, Haibo Li, Tim Wark and Peter Corke, "**Demonstration of Wyner-Ziv Video Compression in a Wireless Camera Sensor Network**," *in Proceedings of the 9th Scandinavian Workshop on Wireless Ad-hoc & Sensor Networks (ADHOC'09)*, 4-5 May 2009, Uppsala, Sweden.

8. Johannes Karlsson, Jean-Paul Kouma, Haibo Li, Tim Wark and Peter Corke, "**Poster Abstract: Distributed Video Coding for a Low-Bandwidth Wireless Camera Network**," *in Proceedings of the 5th European conference on Wireless Sensor Networks (EWSN 2008)*, 30 January - 1 February 2008, Bologna, Italy.

9. Tim Wark, Peter Corke, Johannes Karlsson, Pavan Sikka and Philip Valencia, "**Real-time Image Streaming over a Low-Bandwidth Wireless Camera Network**," *in Proceedings of the Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2007)*, 3-6 December 2007, Melbourne, Australia.

10. Johannes Karlsson, Tim Wark, Philip Valencia, Michael Ung and Peter Corke, "**Demonstration of Image Compression in a Low-Bandwidth Wireless Camera Network**," *in Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN 2007)*, 24-27 April 2007, Cambridge (MIT Campus), Massachusetts, USA.

11. Johannes Karlsson, Mikael Israelsson, Jerry Eriksson and Haibo Li, "**Efficient P2P Mobile Service for Live Media Streaming**," *in Proceedings of the Australian Telecommunication Networks and Applications Conference (ATNAC 2006)*, 4-6 December 2006, Melbourne, Australia.

12. Johannes Karlsson, Jerry Eriksson and Haibo Li, "**Real-Time Video Performance Using Preemptive Routing**," *in Proceedings of The Australian Telecommunication Networks and Applications Conference (ATNAC 2006)*, 4-6 December 2006, Melbourne, Australia.

13. Johannes Karlsson, Jerry Eriksson and Haibo Li, "**Two hop connectivity for uniformed randomly distributed points in the unit square**," *in Proceedings of the First International Conference on Communications and Networking in China (CHINACOM 2006)*, 25-27 October 2006, Beijing, China.

14. Johannes Karlsson, Jerry Eriksson and Haibo Li, "**P2P video multicast for wireless mobile clients**," *in Proceedings of the Fourth International Conference on Mobile Systems, Applications, and Services (MobiSys 2006)*, 19-22 June 2006, Uppsala, Sweden.

15. Johannes Karlsson and Haibo Li, "**P2P video multicast for wireless mobile clients**," *in Proceedings of Swedish Symposium on Image Analysis*, 16-17 March 2006, Umeå, Sweden.

16. Johannes Karlsson, Jerry Eriksson and Haibo Li, "**Real-time video over wireless ad-hoc networks**," *in Proceedings of the fourteenth International Conference on Computer Communications and Networks (ICCCN 2005)*, 17-19 October 2005, San Diego, California, USA.

17. Johannes Karlsson, Mikael Israelsson and Haibo Li, "**Real-time video over wireless ad-hoc networks**," *in Proceedings of Swedish Symposium on Image Analysis*, 11-12 March, 2004, Uppsala, Sweden.

18. Lakshmi Venkatraman, Falk Herrmann, Johannes Karlsson, "**BTNet: A New Topology Establishment Scheme for Bluetooth Scatternets**," *in Proceedings of 5th European Personal Mobile Communications Conference (EPMCC)*, 22-25 April 2003, Glasgow, Scotland.

Patents

19. Falk Herrmann, Andreas Hensel, Arati Manjeshwar, Mikael Israelsson, Johannes Karlsson, Jason Hill, "**Self-organizing hierarchical wireless network for surveillance and control**" US 2003/0151513

Acknowledgements

First of all I would like to thank my supervisor Prof. Haibo Li for giving me the opportunity to pursue research in the Digital Media Lab (DML) at the department of Applied Physics and Electronics (TFE). I am grateful for the valuable guidance from him in the world of research and for all the support and encouragement.

Thanks to everyone that has been associated with Digital Media Lab, both in the past and present, for creating an enjoyable and interesting working environment.

Thanks to all members of the Digital Djurpark project group. I also thank the staff at Lycksele zoo for letting me use the zoo as a test site for wireless video sensor networks.

Thanks to all the staff at Robert Bosch Corporation Research and Technology Center North America (RTC) in California. It was during my time there in 2001 I started my research in the field of wireless sensor networks.

Thanks to Tim Wark and all the other staff at CSIRO ICT Centre in Brisbane, Australia. It was during my time there I started my research on environmental sensor networks, that later lead to the Digital Zoo project. I am looking forward to a very fruit-full collaboration in the future.

For help during thesis writing, I would like to thank Håkan Gulliksson for proofreading and Keni Ren for illustrations and cover design.

Many thanks to my parents Ingrid and Roland, all of my brother and sisters: Petter, Tomas, Kristina, Markus, Anna and Viktor and my grand parents Ulla and Gunnar Lundbäck and Anna Karlsson.

Finally but most importantly, I express my warm thanks to my wife Caroline Karlsson and our twin boys Tim and Isak.

Financial Support

The research in my PhD program was financially supported by ProcessIT, the VITAL project and the Digital Djurpark project.

The Digital Djurpark project is funded by EU Mål 2, Region Västerbotten, Lycksele kommun, Lycksele djurpark, and CSIRO. I am very grateful for the financial support to this project, it gives me resources to build up large-scale sensor networks. Without this support, much of the research in this thesis would not have been possible.

Johannes Karlsson
Umeå, December 2010

1 Introduction

1.1 Motivation

Most computing and communicating devices have been personal computers that were connected to Internet through a fixed network connection. It is believed that future communication devices will not be of this type. Instead the intelligence and communication capability will move into various objects that surround us. The objects of this type are often called "Smart Objects". Wireless sensor networks are a specific class of these devices that have sensing capability and that communicate over wireless networks. It is predicted that "Smart Objects" will be counted in billions in the near future. The new type of Internet that will emerge out of this technology is often referred to as the "Internet of Things" or the "Wireless Embedded Internet".

The advances in the area of image sensors have made it possible to create high-resolution image sensors to a low-cost. This will make it possible to create smart objects that have the possibility to capture images and video. The trend for Internet applications today is that multi-media content has become more important. Video communication in wireless sensor networks is therefore an important topic, and many research issues are left to be solved.

Mainly two types of video communicating "Smart Objects" are considered in this thesis. One is battery powered handheld devices that have display, wireless communication interface, and computational capacity. For these devices we consider a wireless ad-hoc network. This is a special type of network where all nodes can act as both hosts and routers and are capable of operating without support of any central control. The nodes can forward data for each other in multiple hops and the path between two nodes is set up dynamically depending on the connectivity between the nodes and without the support of any central coordinator. In this thesis real-time video communication for this type of devices is considered. The ad-hoc network can be deployed rapidly since it does not rely on any infrastructure. Application scenarios for this type of system can be for example disaster areas where the existing communication infrastructure has been destroyed and a new one must be set up.

The other type of "Smart Objects" considered in this thesis is wireless video sensor networks. These are small, low-cost, and low-complexity devices that are equipped with an image sensor and can communicate the captured images over a wireless channel. These devices are very limited in terms of memory, energy, computation, and communication capacity. Handling video in a network built by this kind of devices is a challenging task due to the limited resources on the sensor node and the high amount of data that is generated from the captured video signal. One application scenario for wireless video sensor networks is the "Digital Zoo" that has been developed as a part of this thesis. Here a wireless video sensor network is deployed at a zoo. The goal is to capture information about the animals and their environments to enable new experiences for physical visitors in the zoo, and for "cyber" visitors from home.

1.2 Research goals and methods

The first, and most important step, is to identify the main problems for the research area. Often way to little emphasis is put on this step.

Research in this particular area can be performed at three different levels. First it is possible to use theoretical models for the video and wireless networks. This will require a rather abstract model of the real world system where the network is applied and of the network itself. The network is often described as a graph and the quality of the link can be modelled as a markov process.

Most research on wireless ad-hoc networks is performed using network simulators. Two of the most commonly used network simulators are GloMoSim [1] and NS2 [2]. With them it is possible to use a more realistic model of the original system, including radio, link-layer, routing and mobility. It is also possible to use a real video codec in the simulations to simulate the effects of the interaction between the video codec and the network. When this method is used it possible to reproduce the experiments in a controlled way.

The third level is to perform experiments on a real system. The difference between the real world and the models used in the simulators can often lead to significantly different behaviour. It is however difficult to reproduce the experiments performed in a real system. The cost of building such system can also be high.

In this thesis all of these levels are considered and used. First a theoretical solution can be given when for example new routing protocols and video codecs are developed. These can be tested in simulators or experimental results can be given. On the next level these solutions should be tested on real hardware in small prototype system. Here more system aspects of the proposed solution are addressed. On the final level a real large-scale system is considered. Here we can really test if our theoretical solutions also work in the real world. Often field tests in large-scale systems are also used to identify important research problems. If the research is only based on theoretical models, or network simulations, it is hard to identify the real research problems. It is also easy to adjust the problem to fit your solution, which will not lead to good research.

2 Wireless Sensor Networks

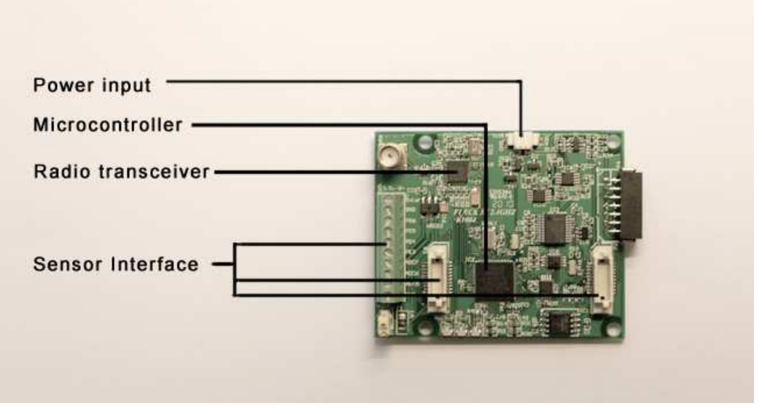


Fig. 2.1: *Picture of a typical mote-class wireless sensor node. It consists of a microcontroller, a wireless transceiver. To form a complete sensor node, energy sources and sensors should be added to the node.*

Wireless sensor networks (WSN) are new tools to capture information from the environment at a scale, both in time and space, previously not possible. A wireless sensor network is built up by a set of nodes where each node typically consists of a microcontroller, sensor/actuator, power supply and a wireless communication device.

The sensors or actuators are used as interfaces from the node to the physical world. A sensor is used to observe the physical environment around the sensor and the actuators are used to change the environment. The type of sensors can range from simple temperature sensors to high resolution image sensors.

The microcontroller is the central unit in the node. It will control the sensors or actuators and decide how and when to observe or control the environment. It is also responsible for processing the collected information. The microcontroller usually also have internal or external memory where it can store the collected and/or processed information. Finally the microcontoller is controlling the wireless communication device and decides when and where to send information. It can also receive data from other nodes and decide what to do based on the content of the received data.

To operate the sensor node needs a power-supply. This can be pre-stored energy from batteries or harvested energy from the environment stored in a rechargeable battery. The most common way to harvest energy is to use solar panels. Other sources for energy harvesting can be vibration, temperature changes and water/air flow. For sensor nodes energy is usually a very limited resource. The microcontroller should therefore duty-cycle activities as a way to maximize the operation of the sensor node.

Finally a wireless sensor node also includes a wireless transceiver. It can range from simple transceivers that just send and receive individual bits of information to the network to more advanced transceivers that handle many of the link layer tasks. This can include to package information into packets, medium access control and encrypt and decrypt the data. The radio transceiver is usually the most power-consuming component in a wireless sensor node. Most transceivers used in wireless sensor networks are operating on the industrial, scientific and medical (ISM) radio bands. Some of the most common frequencies are 433 MHz (Europe), 868 MHz (Europe), 915 MHz (US and Australia), and 2.45 GHz (worldwide).

2.1 History of WSN

In the end of the 1990s MEMS researchers at US Berkeley university coined the term "smart dust" [3]. The vision was to have cubic millimetre wireless sensor node containing sensor, microcontroller, solar cell, battery, and wireless transceiver. These "dust sized" nodes can be distributed in the environment to collect, compute and communicate information. The term "smart dust" is still used when talking about miniature sized wireless sensor nodes.

In the beginning of 2000s a series of wireless sensor platforms were built by a team of researchers at the UC Berkeley university. The platform was built using standard microcontrollers and radio tranceivers. The sensor network platforms are generally referred to as Berekely motes. The team at Berkeley also started the development of an operating system for wireless sensor networks. The name of the operating system was TinyOS, and it is still one of the most widely used operating systems for wireless sensor networks, especially in the academia. Another common opertating system for wireless sensor networks is the Contiki operating system, developed by researchers at the Swedish Institute of Computer Science (SICS). It was the first operating system for sensor networks that provided IP communication and it is an open source operating system written in C that was first released in 2003.

In 2003 the Institute of Electrical and Electronics Engineers (IEEE) released the 802.15.4 standard. This standard defines the physical layer (PHY) and the medium access control (MAC) for wireless sensor network transceivers. This was an major milestone for the acceptance of wireless sensor networks in the industry. Up to this point no global standard for low-power radio transceivers existed. This standard has been very successful and in recent years most sensor network platforms are using IEEE 802.15.4 transceivers.

Recently the trend is towards using the Internet Protocol (IP) as the networking protocol for wireless sensor networks and other smart objects. The terms "Internet of things" and "wireless embedded Internet" are widely used. The Internet Engineering Task Force

(IETF) 6LoWPAN working group has defined a set of standards to enable IPv6 over low-power, low-rate wireless networks. This work was started 2005 and in 2007 the first 6LoWPAN specifications were released.

2.2 Applications for WSN

The applications for wireless sensor networks are numerous, a few of them will be described in this section.

The first applications for WSN were military. In the early days of WSN the military was a major supplier of research funding in the field. One example of a military application is battlefield surveillance to detect and classify enemy vehicle movements.

Another major field for WSN research is environmental monitoring. Some examples of applications for this field are habitat monitoring, precision farming, bush fire warning system, water quality monitoring, and applications where the sensor networks are used to understand the effects of climate change.

Recently the interest of WSN for industrial applications has increased. Machine health monitoring is one application in this area. Here a sensor node attached to a machine can detect when a machine is about to break by detecting vibration patterns. Another application is logistics where smart wireless sensors can be attached to items and not only track their position, but also use various sensors to detect different events, like temperature change, shock, or high moisture levels.

2.3 Wireless video sensor networks

The recent availability of low-cost CMOS image sensors has enabled the field of wireless video sensor networks. These sensors can capture pictures or video from the environment and enable new applications for wireless sensor networks. Some example of new applications are wireless sensor networks for surveillance and security where a network of nodes can identify and track objects from their visual information. The wireless video sensor networks will also greatly enhance the application area of environmental monitoring. Visual information from the environment is important in applications like precision farming or habitat monitoring. The usage of wireless video sensor networks will also enable new entertainment applications like the Digital Zoo where visual information can be provided at a high scale in real-time from a location.

The wireless video sensor networks will also introduce new research challenges to the field of WSN. First of all the amount of data generated by an image sensor is much higher than for many other types of sensors. A low-resolution image of QVGA (320x240) at 12 bits per pixel will generate 115 200 bytes of data. This is much more than the available RAM memory on a typical wireless sensor node.

Since a video contains both temporal and spatial redundancy compression of the signal is possible. The compression algorithm used on a sensor node should have low complexity since the node has limited computational resources. At the same time the algorithm should have high compression efficiency since the nodes are limited in communication capacity, and there is a high energy cost for communication. These two requirements are contradictory since a more complex encoder usually produce higher compression rate.

One promising approach approach to handle video is to use compressive sampling (CS). When traditional image and video compression techniques are used the media signal is first sampled at twice the media bandwidth, according to the Nyquist-Shannon sampling theorem. After this step the signal is compressed. Even though the compressed image and video is small, this means that the sensor node still needs a large amount of memory to store the uncompressed image. The CS principle instead seeks to minimize the collected data already at the sampling step. The theory claims that it is possible to perfectly reconstruct a signal sampled at a rate lower than the Nyquist rate, given that the signal is sparse. Using this principle will greatly reduce the constraints on memory, energy, communication speed, and computational resources for the sensor node.

Transportation of images and videos over WSN is also a challenging task. Even a compressed image or video contains much data compared to other sensor network applications. The network load will therefore be relatively high. If the image capturing nodes are scheduled on and off then the type of traffic in the network will also be bursty. Another problem is that video codecs are usually sensitive for lost packets, and a multi-hopping wireless sensor can have rather high packet loss rate compared to other types of networks.

3 Mobile wireless ad-hoc networks

In traditionally cell based systems the communication relies on infrastructure. All communication between nodes is handled by a central base station. In a wireless ad-hoc network the nodes are capable of operating without support of any infrastructure. The nodes can forward data for each other in multiple hops and the path between two nodes is set up dynamically depending on the connectivity between the nodes and without the support of any central coordinator. A wireless ad-hoc network consists of a set of mobile nodes that are connected by wireless links, for example WLAN or Bluetooth. The nodes are free to move randomly and the topology of the network may therefore also change randomly. This network can be deployed rapidly since it does not rely on any infrastructure. It can be useful in for example disaster areas where existing infrastructure has been destroyed and new communication must be set up.

Routing protocols developed for traditional fixed networks will not work efficiently in ad-hoc networks due to the special characteristics of theses networks. The earliest wireless ad-hoc networks were developed by the Defense Advanced Research Projects Agency (DARPA) in the early 1970s. Since then several routing protocols for ad-hoc networks has been proposed [4]. Such protocols must deal with the typical characteristics of these networks. The nodes in this network are mobile and links between nodes can break and new links can be established. This will lead to a highly dynamic topology. The nodes usually have very limited resources when it comes to energy, bandwidth and processing power. The wireless links are error-prone and have time-varying characteristics. The broadcasting nature of wireless links also cause special problems like the hidden and exposed terminal problem. These routing protocols must also be fully distributed since they have no fixed infrastructure.

Some of the design goals for a wireless ad-hoc networking protocol are to have low routing load, high packet delivery ratio and low end-to-end delay. Many of the most common ad-hoc routing protocols are also designed to scale well for large number of nodes [5]. There is often a tradeoff between routing load and routing performance. If a node has updated information about the network topology it can make good routing decisions. The ad-hoc routing protocols can be classified in two major classes, reactive and proactive, depending on the routing information update mechanism.

In the proactive or table driven protocols each node try to maintain up-to-date information about the network topology. When a node needs a path to a destination it runs a path finding algorithm on the topology information. This requires each node to maintain one or more tables to store information about the network topology. Routing information is generally flooded in the network. This information can be updated periodically or when a change in the network topology is recognized. The method used to exchange the information about changes in the network topology differs between different proactive routing protocols. The number of tables needed to store topology information also differs between different protocols. Some examples of proactive routing protocols are Highly Dynamic Destination-Sequenced Distance Vector routing protocol (DSDV) [6], Optimized Link State Routing Protocol (OLSR) [7] and Wireless Routing Protocol (WRP) [8].

The reactive or on-demand routing protocols do not maintain any network topology information. When a source node requires a route to a destination it initiates a route discovery process. This process is completed when one or a set of possible routes has been found between the source and the destination. In this class of routing protocols no periodically route information is exchanged between nodes. Once a route has been established a route maintenance mechanism is used to maintain information about the route. The route maintenance is performed until no route is available or the route is not needed anymore. Some of the routing protocols that belongs to this category are Ad hoc On-Demand Distance Vector (AODV) [9] and Dynamic Source Routing (DSR) [10].

3.1 AODV

The Ad hoc On-Demand Distance Vector (AODV) routing protocol is designed for mobile ad hoc networks having from tens to thousands of mobile nodes. AODV can handle from low to relatively high mobility rates. It has been developed for a network where all nodes can trust each other. AODV is designed to reduce the routing load in order to improve scalability and performance. As the name indicates it is a reactive distance vector protocol. It is reactive because the nodes that are not active in the communication does not have to maintain routing information. Distance vector means that each node will only know the next hop to reach a target destination.

In AODV three different message types are used. These are Route Request (RREQ), Route Reply (RREP) and Route Error (RERR). They are all transmitted via the Unigram Data Protocol (UDP).

To ensure loop freedom a destination sequence number is used. Every node must have a table containing the latest received sequence number for each destination node it maintains. This sequence number is called "destination sequence number". This is updated every time a new RREQ, RREP or RERR message related to a given destination is received. Each node also has to maintain its own sequence number. This number is increased each time the node originates a RREQ or RREP packet.

When a node needs a route to a destination and does not have one available it will broadcast a RREQ packet to find a route to the destination. This can happen if the destination is previously unknown to the node, or if the route has expired or has been marked as invalid. When a node receives a RREQ packet it will first determine if this route request has been received before. If this is the case the packet will be discarded. Otherwise the

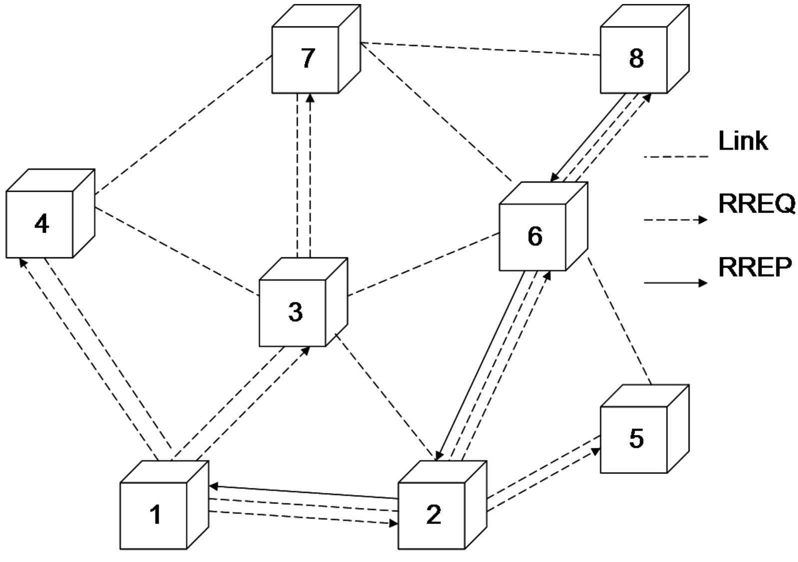


Fig. 3.1: *The route discovery process for AODV.*

route back to the originating node will be updated. This is called the reverse route and is needed for sending a RREP back to the originating node. If the node is the destination or if it has an active route to the destination it will discard the RREQ packet and respond with a RREP packet. Otherwise it will update and rebroadcast the RREQ packet.

When a RREP packet is created it will be sent by using unicast to the next hop towards the originator of the RREQ. When the RREP packet is received the routing table will be updated and the RREP packet will then be rebroadcasted towards the originator of the RREQ. During this phase a forward route will be built up from the originator of the RREQ to the originator of the RREP.

If a node detects a link break for an active route it will build a list of all unreachable destinations that uses that link as next hop. These destinations are then marked as invalid in the routing table and a RRER packet containing the unreachable destinations is then transmitted. If a node gets a data packet destinated for a node for which it does not have an active route it will also mark that destination as invalid and a RERR packet is transmitted. When a RERR packet is received by a node it will build a list of all destinations in the RERR packet for which it uses the transmitter of the received RERR packet as next hop. These destinations will then be marked as invalid and a RERR will be transmitted.

To detect connectivity between nodes a HELLO packet may be used. A node should only transmit HELLO packets if it is part of an active route. This packet is transmitted periodically and a node may detect connectivity by listening to this message. If a node has not received any packet from its neighbor for more than a given time interval it will consider the link to that neighbor lost.

3.2 DSR

The DSR routing protocol is designed for mobile ad-hoc networks having up to about two hundred nodes. It is designed to work well for even high mobility. It assumes that there is a co-operative environment where all nodes are willing to forward information for each other. Some of the optimizations in this routing protocol require that the network interface can operate in promiscuous mode. This is when the network interface delivers all packets to higher layer without filtering on the link-layer destination address. It is an on-demand routing protocol and routes are only established when needed. DSR does not require any periodically packets to be transmitted. Since it is a source routing protocol each data packet will carry in its header information about the entire route. This makes it possible for the sender to control the route for its own packets.

If a node wants to transmit data it will first look in its route cache for a route to the destination. If there are no routes available the node will initiate the route discovery process by broadcasting a Route Request (RREQ) packet. This packet will contain the initiator and the target of the RREQ and a record containing all intermediate nodes through which this packet will pass on its way from the originator to the target. It will also contain a unique request identification. When a node receives a RREQ it will discard the packet if it has already received this RREQ before. If it is the target of the RREQ it will send back a Route Reply (RREP) packet to the initiator of the RREQ packet. This packet will contain the list of all intermediate nodes from the RREQ packet. If the node receiving the RREQ packet is not the target node and have not heard the packet before it will add its address to the list of intermediate nodes and then rebroadcast the RREQ packet.

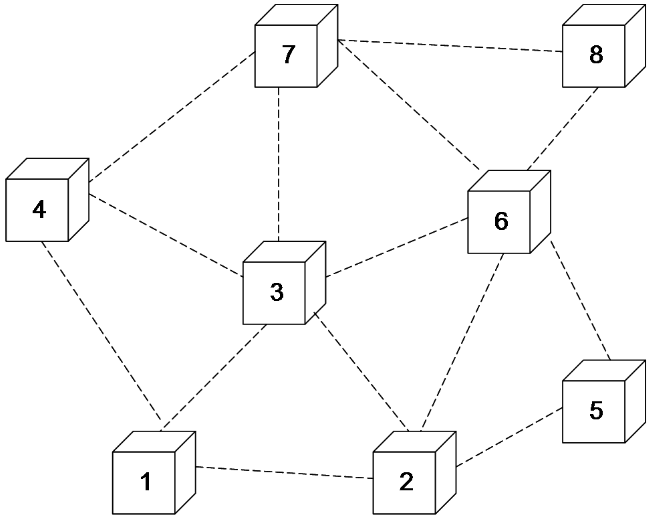
When the RREP is sent back the reversed route from the RREQ packet is used to source route the packet. This will require that all discovered links are bidirectional. When the initiator of the RREQ packet receives the RREP packet it will add the route to its route cache.

When a node forwards a packet it is responsible for confirming that the packet can be transmitted over the link. This can be done by link layer acknowledgement or by some explicit acknowledgment from DSR. If a node detects a broken link it will send a Route Error (RERR) packet back to the source of the packet using the broken link. When a node receives a RERR packet it will remove all routes in its route cache that uses the broken link.

There are several optimizations available for DSR. All nodes can add information from overheard and forwarded packets to its route cache. If an intermediate node has a route in its route cache to the destination it can reply to a RREQ. If a node in a route detects that it can directly receive a packet without the need for the previous node in the route to relay the packet it can reply with a gratuitous RREP to the sender of the data packet. This is called automatic route shortening.

3.3 OLSR

The optimized link state routing protocol is an optimization of the classical link state routing protocol. This is a proactive table driven routing protocol. It means that it will all the



ID	MPRs
1	2,3
2	1,6
3	6
4	1,7
5	2,6
6	3
7	3,6
8	6,7

Fig. 3.2: *A table of MPRs for all nodes when OLSR routing protocol is used.*

time exchange topology information with other nodes in the network. This routing protocol is well suited for application needing low latency and for applications communicating with many different other nodes since a route is always available to all other nodes in the network. In this protocol each node selects some of its neighbors as multipoint relays (MPRs). Only nodes selected as a MPR of a neighbor will forward control information from that neighbor. The idea with MPRs is to reduce the number of redundant transmission of a control message over a region. This protocol is therefore especially well suited for dense ad-hoc networks.

The MPR of a node, N, is a subset of all 1-hop neighbors of that node. This subset must be selected in such way so all 2-hop neighbors of node N have a 1-hop connection to a MPR of node N. This means that a control message transmitted by a node will be received by all of its 2-hop neighbors. The neighbors having both a 1-hop and a 2-hop connection is not considered as a 2-hop neighbor here. The set of MPRs can be any subset of a nodes 1-hop neighbors, but the smallest subset will give least routing overhead. Each node maintains information about the nodes that have selected it as a MPR. If a node receives a control packet from a node and is one of that nodes MPR it is expected to rebroadcast the control packet.

Link sensing and neighbor detection is done by HELLO packets. The signaling of MPR selection is also exchanged in the HELLO packets. A HELLO packet contains information about all neighbors and if they have been selected as MPR or not. This packet must be transmitted within a certain time interval.

OLSR is a link state protocol and all nodes must have knowledge about the network topology. All nodes that are selected as a MPR by some other node will broadcast topology control information. This topology information must contain at least the set of nodes that are using this node as a MPR. This topology information is rebroadcasted by the MPR nodes to the entire network. Each node will run a shortest path algorithm on the topology information to generate the routing table. It contains information about all destinations and for each of them the next hop and number of hops to reach this destination. The routing table is updated every time new topology information is received.

3.4 Implementations

There exist several implementations of wireless ad-hoc network protocols. Most of the implementations are for the Linux platform, but there also exists implementations for Windows. In this section some of the available implementations both for wireless devices and for network simulators are listed.

- AODV-UU from Uppsala University is RFC3561 compliant and runs on both Linux and in the NS-2 simulator. In Linux it will run as a user-space daemon and it requires no kernel modifications. It is possible to crosscompile for ARM based devices like IPAQ.
- Kernel-AODV is a Linux implementation of AODV from National Institute of Standards and Technology (NIST). It is RFC3561 compliant.
- DSR-UU is a DSR implementation available from Uppsala University. It runs on both Linux and in the NS-2 network simulator.
- The olsr.org is a OLSR daemon available for both Windows and Linux. The implementation is RFC3626 compliant.
- Mesh Connectivity Layer (MCL) from Microsoft is a implementation of a modified version of DSR called Link Quality Source Routing (LQSR). Both binaries and source code for Windows are available.
- The GloMoSim network simulator has support for several ad-hoc routing protocols for example AODV, DSR and WRP.
- The NS-2 network simulator has many ad-hoc routing protocols available.

3.5 Challenges

Sending real-time video over wireless ad-hoc networks is a challenging task. The ad-hoc network is error-prone due to the wireless links and the random topology. At the same time video is very sensitive for packet losses because of the predicitive coding. The often reactive nature of the ad-hoc routing protocols is also causing problems. When a route is lost and a new has to be established there will often be an unacceptable delay for the real-time video transmission.

4 Video Coding

Applications for wireless video range from remote expert help for rescue personnel in a disaster area to live TV streaming on a mobile phone. The real-time requirement differs a lot depending on the application. For example an interactive video communication may require a very low delay while a live-TV streaming can tolerate a relatively high delay. The wireless devices have very limited resources in terms of bandwidth, computation and energy. A video coding system for wireless video must be carefully designed to meet these constraints.

In digital video the video is sampled both in the spatial and temporal domain and it can be described as $\psi(m, n, k)$, where k is the frame number and m and n the pixel within a frame. Video data contains spatial and temporal redundancy making uncompressed video highly inefficient. An uncompressed video having resolution 176x144 and a frame rate of 10 fps will produce a data stream having a bitrate of more than 6 Mbit/s. In this section we will describe some of the most commonly used video compression standards and the basic principle behind them.

4.1 Block-based hybrid video coding

In block based hybrid coding the image is first divided into a number of non overlapping blocks. Usually two different blocks are used where the typically block size is 8x8 pixels and a macroblock of 16x16 pixels. The image contains one luminance component and two color components. The color component is usually sub sampled 2:1, so one 8x8 block for the color components corresponds to 16x16 pixels in the original image.

Each macro block is first motion estimated (ME) against the previous decoded frame, called the reference frame. The motion vector specifies the displacement between the current block and the best matching block in the reference frame. The motion vectors are used to create a motion compensated (MC) block from the reference frame. The difference between the motion compensated block and the original block is then discrete cosine transformed (DCT). This is a transformation to the frequency plane and is used to exploit the correlation between error pixels. The quantized DCT coefficients and the motion vectors are then coded and transmitted.

A frame that is coded by using the previous frame as reference is called an inter frame. A frame can also be coded without using a previous frame as reference and these frames are called intra frames. The first frame in a video sequence does not have any previous

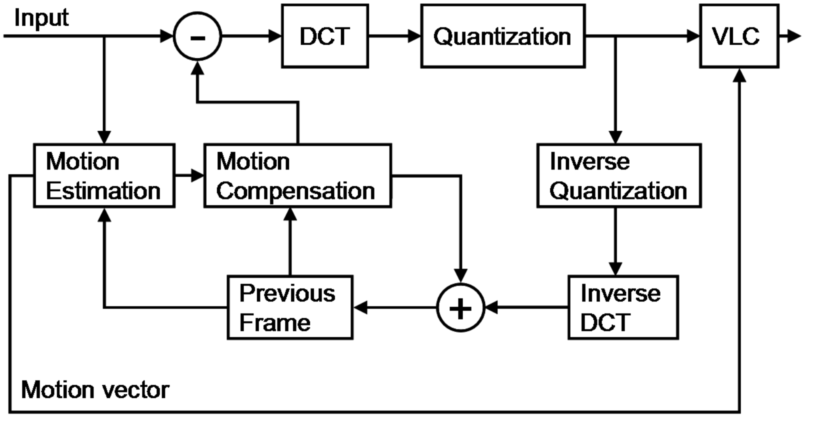


Fig. 4.1: *Block based hybrid codec.*

frames and must be coded as an intra frame. In many video codecs an intra frame is also inserted periodically.

The encoder and the decoder must use the same reference picture. The encoder must therefore include a decoder to create this reference picture. In the decoder part the DCT components are first inverse quantized, and then inverse DCT transformed. Each error block is then added to corresponding motion compensated block to create the decoded picture. This will be the reference picture for the next frame.

4.2 MPEG-4

MPEG-4 [11] was defined by the Moving Picture Experts Group (MPEG). This is a working group within the International Organization for Standardization (ISO). It consists of a collection of several standards. In this section the MPEG-4 part 2 video coding standard will be discussed. The MPEG-4 part 10 standard is also called H.264/AVC and will be discussed in the next section. The standard covers a large set of tools for audio and video coding. To make it possible to create implementations of the standard several profiles were defined. A "profile" is a subset of all the MPEG-4 tools that a decoder has to implement. The most common profiles are simple profile and advanced simple profile. They are basically the same but the advanced simple profile has some extra tools to increase the coding efficiency, for example quarter pixel motion estimation. To set constraints on parameters in the bitstream "levels" are used. This can be for example constraints on the resolution and bitrate.

There exist several implementations of MPEG-4 part 2. XviD [12] is an open source implementation supporting the simple and advanced simple profile.

4.3 H.264/AVC

H.264/AVC [13] was developed in a cooperation, known as the Joint Video Team (JVT), between the ITU Telecommunication Standardization Sector (ITU-T) and the MPEG. The name H.264 comes from the ITU part and the name AVC from the MPEG part. The goal of the project was to develop a video coding standard that could produce good video quality at bitrates half of what previous standards would need.

The codec is based on the block-based hybrid coding approach. It contains several improvements compared to old standards in different steps of the coding process. None of these improvements contains the majority of the improved coding efficiency. Instead the relatively small improvement in each step will add up to the significantly improved coding efficiency. The H.264 can improve the rate-distortion efficiency with a factor of two in bitrate savings when compared to for example MPEG-2. The complexity for H.264 is however much higher compared to old video coding standards. The encoder part typically has 4-5 times higher complexity and the decoder part 2-3 times higher complexity than MPEG-2.

The macroblock is 16x16 pixels and can be segmented in 16x16, 16x8, 8x16 or 8x8 pixels. If the segmentation is 8x8 pixels each block can be further segmented in 8x8, 8x4, 4x8 or 4x4 pixels. For each of the segments a motion vector will be transmitted. The maximum number of motion vectors per macroblock is therefore sixteen. The precision of the motion vector is a quarter of a pixel. In H.264 more than one prior coded picture can be used as reference. Both the decoder and the encoder must store a multi-picture buffer and the index of the picture used as reference must be transmitted for each motion vector.

To avoid visible block structures H.264 uses deblocking filter. This will significantly improve the subjective video quality, but the objective video quality will also be increased. For the same objective video quality the bitrate is typically reduced by 5-10% when deblocking filter is used [14].

4.4 VP8

The VP8 codec was released by Google in May 2010. At the same time the WebM multimedia container format was released for use together with VP8 for HTML5 video.

The VP8 has similar features as the H.264 video codec. Some of the major differences are that VP8 does not use the sub 8x8 partitions of H.264. It also uses less reference frames compared to H.264. In VP8 three reference frames are used, H.264 can use up to 16 reference frames. The three different type of reference frames that can be used in VP8 are the last frame, a "golden" frame that is a good reference frame from the past, and an "alternate reference frame", that is a special reference frame that is only used as reference and never displayed at the decoder. The encoder can therefore use this reference frame to create any type of special reference frame that can help to improve compression. The VP8 encoder does not use the B-frames that are commonly used in the H.264 codec.

Fig. 4.2: *Error propagation when the frames 101-103 are lost. Picture (a) is frame number 104 and the following pictures are taken at 4 frames interval.*

4.5 Real-time video transmission

Sending real-time video over wireless networks is a challenging task. The wireless devices have very limited resources in terms of bandwidth, computation and energy. The complexity of the video codec can often be an issue, especially in interactive video communication where the video must be encoded in real-time on the wireless device.

In block-based hybrid video coding a new frame is first predicted from the last decoded video frame. The encoder must therefore also include a decoder that is synchronized with the decoder at the receiver. If data is lost in the network the encoder and decoder will lose synchronization. This will lead to the fact that a lost packet will not only cause error in the current frame but also in the following frames. This is called error propagation, see Fig. 4.2. A video codec developed for wireless communication must be able to handle these problems in an efficient way.

4.6 Error control

The error control mechanism for video transmission can be implemented at several different levels. To reduce the risk of lost data during transmission it is possible to use for example forward error correction, retransmissions and multi-path transport. At the source coding level methods can be used to make the video stream more robust to lost data. If there are errors in the transmission the decoder can minimize the effect of this by using error concealment methods. Finally, it is possible to have methods using interaction between the encoder and the network to handle lost data.

One simple method to stop error propagation is to periodically insert intra frames. These frames are not predictively coded and will therefore stop the error propagation. This is a very common method used in video codecs to handle lost packets.

4.7 Video quality metrics

The quality of the coded and transmitted video will be reduced because of two reasons. First distortion is caused by the compression of the video. The quality will also be decreased because of lost data in the transmission.

It is important to have an objective method to measure the quality of the decoded video compared to the original. This method should correspond as good as possible to the subjective video quality. The objective video quality is usually measured as peak signal to noise ratio (PSNR). This is basically the normalized average difference between each pixel in the original video and the decoded video. This is the most commonly used method to measure video quality.

$$\begin{aligned} MSE = \sigma_e^2 &= \frac{1}{N}\sum_k \sum_{m,n} (\psi_1(m,n,k) - \psi_2(m,n,k))^2 \\ PSNR &= 10\log_{10}\frac{\psi_{max}^2}{\sigma_e^2}, \end{aligned} \tag{4.1}$$

where ψ_1 and ψ_2 are the two video sequences and ψ_{max}^2 is the peak intensity value of the video.

This measurement does however has a non-linear relation with the subjective video quality. It is therefore important to use a utility function that describes the subjective video quality as a function of the PSNR. A video having a PSNR below 30 dB is usually considered bad and a PSNR above 40 dB is considered as good quality.

4.8 Scalable video coding

In scalable video coding it is possible for a decoder to reconstruct the video at different quality by using different number of layers. This is useful if users having different bandwidth or computational resources available wants to access the same video stream. Scalable video coding is typically performed by providing multiple versions of a video stream in terms of quality, spatial resolution, temporal resolution or a combination of those.

In quality scalability the video is represented in different layers with increasingly finer quantization steps. For the first layer, often called the base layer, coarse quantization steps are used. For the following layers, often called enhancement layers, the difference between the previous layer and the original image is quantized by using finer quantization steps.

For spatial scalability the video is represented in varying resolutions. To encode a frame using spatial scalability multiple resolutions of the frame is first created. This is done by spatial downsampling. The fist layer is then created by encoding the image having lowest resolution. The decoded image of this layer is then interpolated to the resolution of the image having second lowest resolution. The difference between this interpolated image and the original image is then coded. The decoded version of this image is then used as reference for the third layer and so on.

In temporal scalability the video is represented at different frame rates. This is very similar to the spatial scalability. Here temporally upsampled pictures from a lower layer are used as reference for a higher layer.

4.9 Codec comparison

To compare the performance of a video codec the quality, measured as PSNR, as a function of the bitrate of the encoded video is used. Here a comparison between H.261, MPEG4, VP8 and H.264 is presented. For encoding of the H.261 and MPEG4 video codecs the FFMPEG version 0.6 was used. For encoding of the H.264 video codec the program x264 r1724 was used. Finally for encoding of the VP8 video codec the ivfenc tool from libvpx v0.9.2-47 was used.

A total of 14 different encodings were run for each type of video codec. To vary the quality and bitrate, the quantization step was changed between each encoding. The key frame interval was forced to infinity for all different video codecs. For x264 the psycho-visual optimizations were disabled since they may worsen the PSNR. Each experiment was repeated for a number of different video clips.

The first video clip used was the "Akiyo" video clip at CIF (352x288) resolution. This clip contains 300 frames and is encoded at 30 frames per second. We can see in Figure 4.3 that H.264 was by far the most efficient video codec followed by VP8 and MPEG4. The H.261 encoder was by far the least efficient encoder for this video clip.

The next video clip used was the "Foreman" video clip at CIF resolution. This clip also contains 300 frames and is encoded at 30 frames per second. In Figure 4.3 we can se that for this clip H.264 also had the best compression efficiency. Here it was however less difference to VP8 that was also the second best for this clip. It was followed by MPEG4, and the worst efficiency was shown by the H.261 codec.

The newer type of video codecs did show improved coding efficiency compared to the older types of codecs in the previous tests. The improved encoding efficiency does however come to the cost of higher complexity of the encoder. In this test we compare the encoding time for different same types of codecs that was used in the previous tests. We use the "Foreman" video clip that is 10 seconds long. The CPU used for all encodings were a Core2 Duo E7300 running at 2.66 GHz. In Figure 4.5 we can see that both

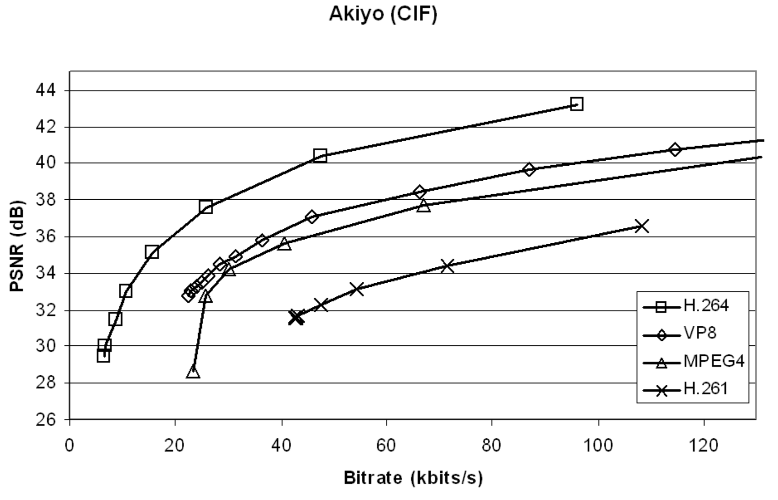


Fig. 4.3: *Rate distortion curves for different video codecs. The Akiyo standard video clip is used in the CIF resolution.*

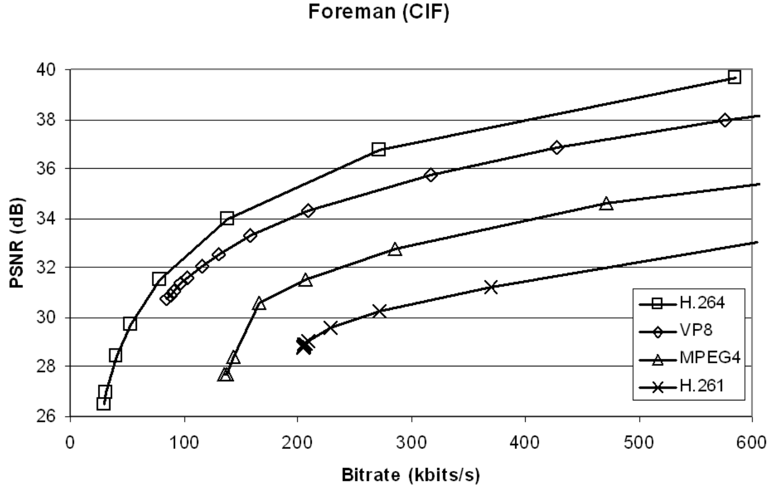


Fig. 4.4: *Rate distortion curves for different video codecs. The Foreman standard video clip is used in CIF resolution.*

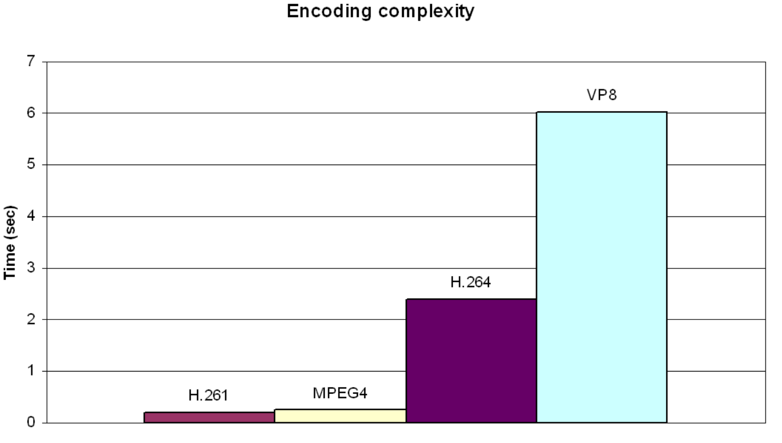


Fig. 4.5: *Encoding complexity for different video codecs. The presented time is the encoding time for the Foreman video clip in CIF resolution.*

H.261 and MPEG4 had a very low encoding complexity. The encoding time was 0.23 and 0.25 seconds for the respectively codec. The H.264 had a much higher complexity and the encoding time for this codec was 2.39 seconds. The encoder having the highest complexity was the VP8 encoder that needed 6 seconds to encode the video clip. There can however be differences in how well the implementation of each codec is optimized. Both FFMPEG and x264 are widely used and has been developed for a long time. The libvpx is recently released and is considered more as a reference implementation for the VP8 codec.

It is important to note that modern video codecs have a wide range of available settings. It is fairly easy to claim that encoder x is more efficient than encoder y, if one is allowed to set any types of settings for the two encoders, and use a video clip that is well suited for encoder x. It is not to uncommon that companies are using this approach when marketing their own codecs.

To show this we compared the H.264 and VP8 video codecs and used three different presets for each codec. The presets used for H.264 was "veryslow", "normal", and "veryfast". The presets used for VP8 was "CPU=0", "CPU=3", and "CPU=5". In Figure 4.6 we can se that for the setting "veryslow" and "normal" the H.264 performed better than any type of setting on the VP8 encoder. The "CPU=0" setting of the VP8 encoder did however perform better than the "veryfast" setting of the H.264 encoder.

In Figure 4.7 we can see that the encoding time did vary a lot depending on the settings used for both the VP8 and H.264 codec. We can note that for the most advanced settings for both encoders they were not able to encode this low-resolution video clip in real-time.

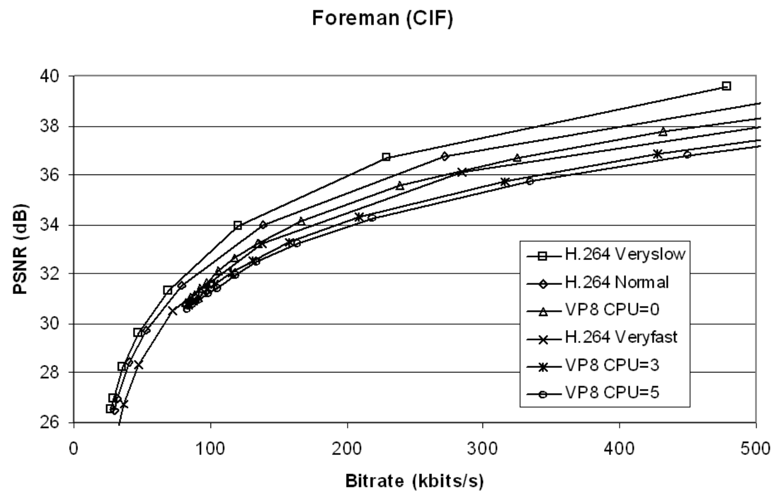


Fig. 4.6: *Rate distortion curves for different settings of the VP8 and H.264 video codecs. The Foreman standard video clip is used in format CIF.*

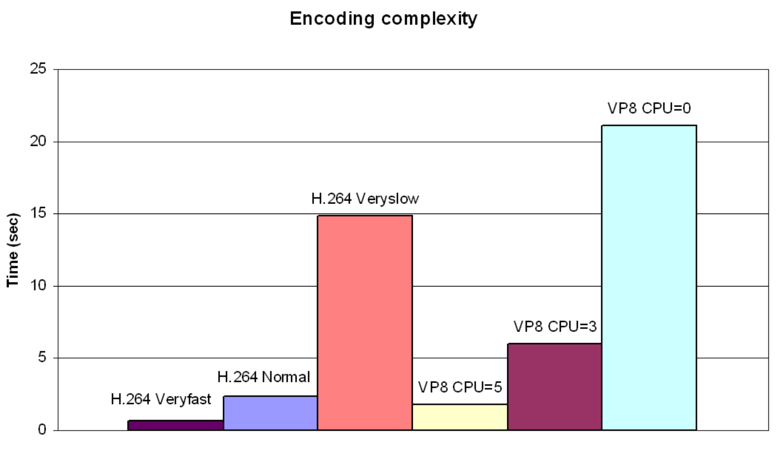


Fig. 4.7: *Encoding complexity for different settings of the VP8 and H.264 video codecs.*

5 The Digital Zoo project

The Swedish zoo "Lycksele djurpark" was founded in 1959 and has since then been one of the largest tourist attraction in northern Sweden. It is the northernmost zoo in the country and is specialized in housing nordic animals like bear, elk, musk-ox, and wild boar to mention some. The zoo covers an area of about 47 hectares and is currently home to more than 400 animals, divided into 30 species. The opening season is from end of May to the end of September, with some additional openings around holidays like Christmas and Eastern. Although the zoo is closed the animals life continues as usual and they are outdoors.

The goal of the project "Digital Zoo" funded by the EU regional development fund is to make the zoo digital and more attractive. The approach is to install a wireless sensor network covering the entire zoo. These sensors will collect information about the animals and their surrounding. The type of information collected can be in the form of video, pictures, sound, and temperature. The collected data will be processed and semantic information extracted. This will enable new applications for zoo visitors, both physical visitors at the zoo, and "cyber" visitors from home. Through these new applications the zoo will become both more attractive and more available.

5.1 System overview

In the Lycksele zoo a wireless sensor network has been deployed. The nodes in this network are totally autonomous and can organize themselves into a working network. Most nodes are equipped with a camera to capture images and video from the animals. The network has been operating in the zoo for over one year. The current deployment consists of 25 camera equipped nodes that are using WLAN as wireless links. The routing protocol mostly used during this deployment is OLSR. The network has so far captured more than 5 000 hours of video from the zoo.

The information collected is transmitted over the wireless network to a single sink node. This is a computer connected both to the wireless sensor network and to the Internet through a fibre connection. On this computer the collected information will be further processed to enable new applications for users in the zoo, or from home. The collected media information is also processed at the nodes collecting the information. From a system perspective it is a good approach to move much of the information processing to

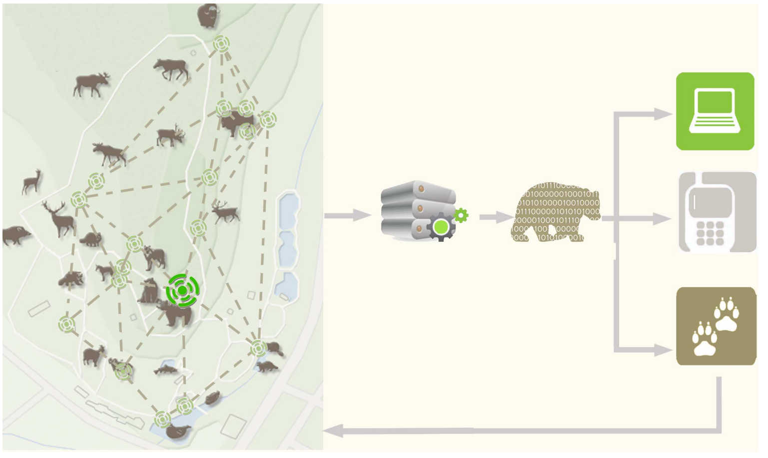


Fig. 5.1: *An overview of the Digital Zoo system. In the zoo a set of video transmitting nodes are deployed. These nodes are configuring themselves into a wireless mesh network. A Internet connected sink node is located at a central location in the zoo, this is marked as a larger node in the network. This node will process the collected information to enable new applications for users in zoo and from home.*

the sensor nodes. If all information first is transmitted to sink node, much communication capacity and energy will be wasted to transmit data that will not be used later anyway.

The location of the zoo gives us a hard environment for real-world wireless sensor network deployments. The location is close to the Arctic circle, and during winter it is virtually no sunshine for long periods. The temperature is often around -30 degrees Celsius and can occasionally drop below -40 C. The ground is also covered by snow during the entire winter, putting special requirement on the housing. On the other hand, during summer there is a lot of sunshine. This will give us a scenario where the deployed system must be highly adaptive to the varying conditions.

5.2 Hardware

To collect information about the animals in the zoo a heterogeneous set of wireless sensors are used. At some location we have access to the main power supply and more powerful cameras can be used that have the capability to advance image processing and deliver high-quality live video. At other locations we need to use low-power cameras that can only operate on harvested solar energy. To save energy these nodes will not be fully operating at all time. Instead they will be triggered on special events and will only process the collected media information when some interesting events are detected.

Fig. 5.2: *Illustration of camera stack formed by a Fleck$^{\mathrm{TM}}$-3 sensor node, AD Blackfin board and CMOS camera.*

5.2.1 CSIRO Fleck$^{\mathrm{TM}}$-3 platform

For solar powered cameras in the zoo we will use the CSIRO-developed Fleck$^{\mathrm{TM}}$-3 platform. This is a robust platform family designed for outdoor applications in environmental monitoring and agriculture. The Fleck$^{\mathrm{TM}}$-3, consists of an Atmega1281 micro-controller with 8Kbyte of RAM running at 8 MHz, a Nordic NRF905 radio transceiver, a real-time clock, 1Mbyte of flash memory and a temperature sensor. The nRF905 radio was chosen due to its sensitive receiver and operation in the 433 and 915 MHz ISM bands, which combined give good radio communications up to 1000 m over flat ground and in some situations well over 2000 m. It provides a packet-oriented interface to the host, and acts as an SPI bus slave for both configuration and data. The chip handles preamble and CRC generation, address and CRC checking and performs Manchester encoding with a real data transmission rate of 50 kbits/s and a typical throughput of 125 packets/second. The latest version of Fleck$^{\mathrm{TM}}$-3 is using an IEEE 802.15.4 radio. This will enable a higher data rate and also better interoperability to other sensor network platforms.

The Fleck$^{\mathrm{TM}}$-3 platform is also designed for easy expansion via add-on daughter-boards which communicate via digital I/O pins and the SPI bus, nearly 20 interfaces for various applications [15] has been developed. The Fleck runs the CSIRO developed FOS, a cooperative threaded operating system that supports multi-hop routing, over-the-air programming, remote procedure calls and Java. It is also possible to run TinyOS or Contiki operating system on this platform.

A complete camera stack using the Fleck$^{\mathrm{TM}}$-3 platform consists of a Fleck$^{\mathrm{TM}}$-3 sensor board, a DSP board and a camera board. To save energy the DSP board and the camera board can be controlled on and off from the Fleck$^{\mathrm{TM}}$-3.

The DSP board in our camera stack contains the Analog-Devices 600MHz ADSP-BF537 Blackfin processor. The Blackfin belongs to the family of 32-bit RISC processors and has no memory management. We use the BlueTechnix BF537E board which comprises the CPU with 132 KB SRAM, 32MB external SDRAM and 4MB flash. The Blackfin has an SPI bus that is connected to the Fleck for interprocessor communications. The implementation is quite compact with the Blackfin board being fitted to the back of the camera board which then plugs onto the Fleck, see Figure 5.2.

Two software environments exist for the Blackfin: uCLinux and Analog Devices Visual DSP++. uCLinux is a fairly complete Linux port that supports a flash filesystem, shell and networking. However it does not support SPI bus slave mode communications, so we use a serial port to connect the Fleck to the Blackfin which is effective, but slower than SPI.

The Camera Daughterboard contains the sensor chip, lens holder and two switchable ultra-bright LEDs for illumination. The sensor is an OV7640 Color CMOS from OmniVision Technologies Inc, which outputs VGA (640 x 480) or QVGA (320 x 240) images at a maximum rate of 30fps or 60fps respectively. It uses a Bayer pattern filter to achieve color, has progressive scan and supports windowed and sub-sampled images. The camera parameters such as exposure time and gain can be set by the host over the local I^2C bus.

5.2.2 Single-chip low-power platform

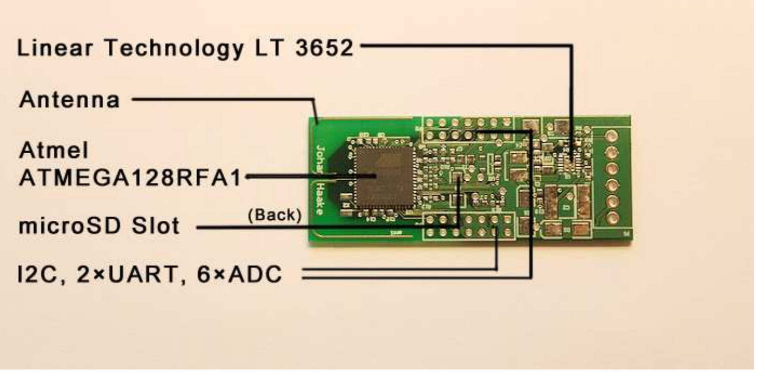


Fig. 5.3: *Picture of a low-power sensor node developed within the Digital Zoo platform. This platform is built on the ATMEGA128RFA1 single chip 802.15.4 microcontroller and radio transceiver. It has a pin header to access most of the I/O from the microcontroller. For energy harvesting Linear Technology LT3652 is used. On the back the sensor node has a microSD card slot.*

A new low-power, low-cost sensor platform has been developed within the Digital Zoo project. The size of this board measures 19mm x 50mm. This is a general purpose platform that can be connected to a various set of sensors. Since it is very limited in terms of memory and computational resources it is not mainly targeted for video sensor networks. If a JPEG enabled camera module is used it could however be possible to use low-resolution images on this platform.

The microcontroller and radio transceiver used on this sensor node is the new single chip solution from Atmel, ATMEGA128RFA1. This chip contains an 8-bit microcontroller running at 16 MHz. It has 128K Bytes of flash memory and 16K Bytes of RAM. It has several peripheral features like A/D converter, SPI and UART interface, and several I/O lines. The chip also contains a fully integrated low-power IEEE 802.15.4 transceiver for the 2.4 GHz ISM band. It supports a data rate up to 2 Mb/s. This platform can run both TinyOS and the Contiki operating system. Since the same type of radio transceiver as the one used in the latest Fleck™-3 it will enable good interoperability between the two platforms.

In deep sleep mode the chip consumes less than 250 nA. When CPU is active at 16 MHz the consumption is 4.1 mA. The radio transceiver consumes 12.5 mA in RX mode and 14.5 mA in TX mode.

To attach different types of sensors to the board most of the I/O lines and the peripheral interfaces from the microcontroller are available on a connector on the board. On the back of the board a micro SD card slot is also available to enable the possibility to add a large amount of Flash memory to the node.

For energy harvesting we use Linear Technology LT3652. It is a complete monolithic step-down battery charger that operates over a 4.95V to 32V input voltage range.

5.2.3 Netbook camera nodes

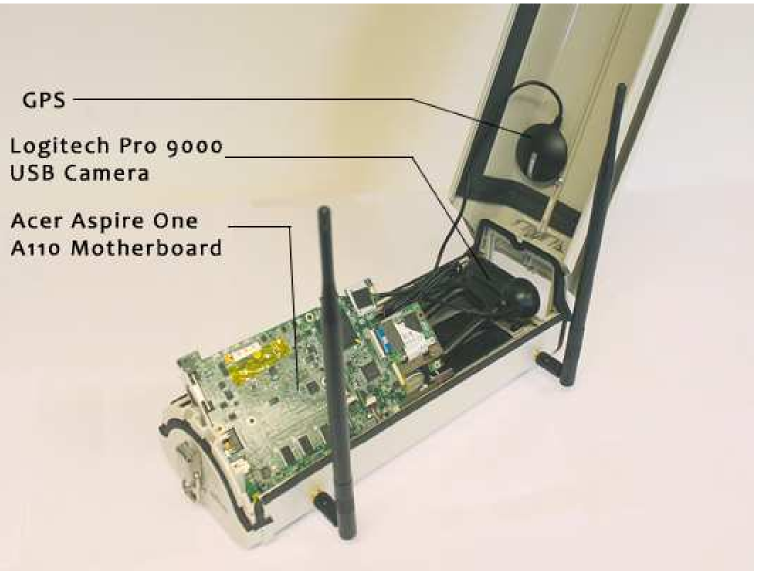


Fig. 5.4: *Picture of a wireless ad-hoc camera node built on the motherboard from Acer Aspire One A110 netbooks. The node is also equipped with a Logitech Pro 9000 camera and a GPS device.*

The first platform built for the Digital Zoo project was based on the motherboard from Acer Aspire One A110 netbooks. This motherboard consists of an Intel Atom N270 single core 1.6 GHz CPU, 512 MB RAM, 8 GB flash memory and an Atheros 802.11bg wireless card. We use a Logitech Pro 9000 USB camera that has a maximum resolution of 1600x1200 pixels and can provide 800x600 at 30 fps. To get the location of all cameras they are also equipped with a GPS device. The largest component in this camera node is the netbook motherboard. It has a size of 215x134 mm and to make it fit into a standard outdoor camera housing we have to tilt it 45 degrees. See figure 5.4 for a picture of the camera system.

The Logitech Pro 9000 USB camera can handle rather low-light conditions. For some cameras we are however using IR-lighting and a modified camera where the IR-filter is removed in order to get video during night in the winter. In the summer it is no problem to get video 24 hours per day from the standard Logitech camera.

A total of 30 nodes of this type has been built and deployed in the zoo. Numerous minor hardware and software errors have been discovered and fixed during the more than one year the deployment has been active. The most serious problem so far has been Linux driver issues for the wireless network card used on this board. So far there has not been any major hardware failure and most nodes are still operating in the zoo.

5.2.4 Gumstix camera

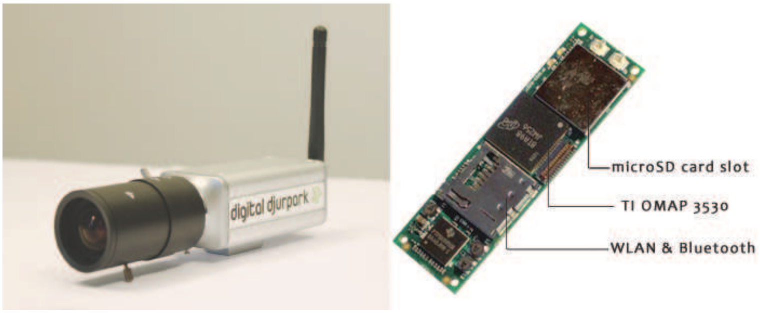


Fig. 5.5: *Our high-performance camera platform based on the Gumstix Overo™ Fire computer-on-module.*

The second generation of the high-performance WLAN based camera platform has been built using the Gumstix Overo™ Fire computer-on-module and a custom image sensor board, as shown in Figure 5.5. The Gumstix Overo™ Fire consists of a Texas Instrument Omap 3530 application processor with ARM Cortex-A8 CPU and a C64x+ digital signal processor (DSP) core, both running at 600 MHz. The board has 256 MB of RAM and 256 MB of flash memory. For communication an IEEE 802.11g wireless LAN is used as a wireless mesh network. The board measures only 17 mm x 58 mm x 4.2 mm which enabled us to build a really small and powerful camera platform.

The camera board contains an OmniVision OV9710 image sensor supporting 1280x800 at 30 fps. The image sensor also has a good low light sensitivity of 3.3V per lux-second.

For this platform a multi-hop wireless ad-hoc network on 802.11 wireless links is used. Since this platform has high computational capacity, as well as high communication capacity, it provides the ability to send high-quality real-time video. This platform has however a rather high power consumption, typically a few watts, and therefore will in general be connected to the mains power supply.

5.2.5 Radio frequency identification

To identify the animals in the zoo we are using radio-frequency identification (RFID). These type of transponders are passive. They don't need any batteries, instead they receive the energy used to transmit back its message from the reader. Most animals in the zoo are carrying a RFID chip, either as an implant or as an ear tag. The implant are usually small glass encapsulated transponders that are implanted below the skin at the back of the neck of the animal. The size of this transponder typically has a length ranging from 10 mm to 20 mm and has a diameter of about 2 mm. For ear tag RFID, larger transponders can be used. Here a typical device has a diameter of 20 mm to 30 mm and a weight of 3 grams to 10 grams. This type of transponder is mainly used for livestock in farming applications.

The RFID transponders used for animals are following the ISO 11785 standard [16]. Readers for this standard have been evaluated from different manufacturers like Texas Instrument, Allflex and also low-cost readers based on the Atmel U2270B reader IC. These readers can be interfaced to all type of sensor nodes that are used in the zoo.

5.3 Applications

Different applications are built based on the multi-media information that is captured from the animals in the zoo. The goal of these applications is to make the zoo more attractive for the visitor, but also to make the zoo more available. Here we present some of the applications that has been implemented, or that are under development, in the Digital Zoo project.

5.3.1 Live video streaming

Since we can provide live video feeds from the cameras deployed in the zoo, the first and most obvious application for a digital zoo was to stream the live content to the webb. Here a Flash streaming server was used to make it possible to provide live video feeds the same way most newspapers and TV networks use for WebbTV on the Internet. To get a large user base, the live TV from the zoo was provided in collaboration with local and national newspapers. This application has been very popular, for long periods the number of connections to the live video streams have been over 2 000 per day. There have been some occasional peaks of more than 6 000 connections per hours.

5.3.2 Animal identification and tracking

Fig. 5.6: *Single individual tracking.*

In the previous application it is only possible to follow a group of animals based on the location of the camera that captures the video. In many areas there can be more than 10 animals of the same type, and it is very difficult for the user to follow individual animals. To enable applications where it is possible to follow individual animals a system to identify and track individual animals is developed. This system will use computer vision techniques at the cameras to track the animals moving in its field of view. See Figure 5.6 for a picture of a tracked animal.

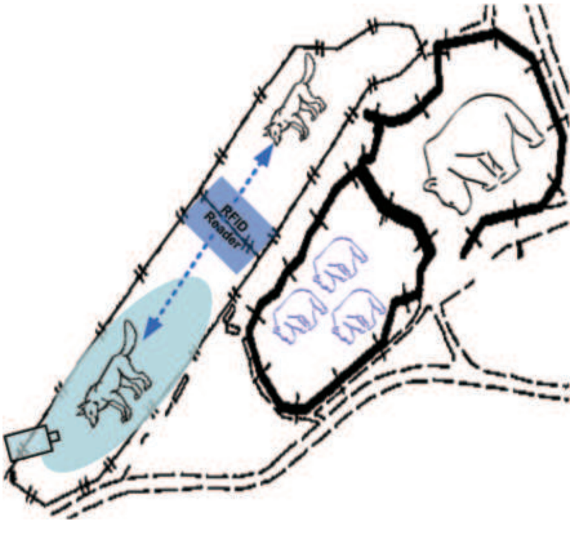


Fig. 5.7: *Wolf Cage Area: The wolves are crossing the tunnel equipped with RFID reader. From which the wolves are identified and tracked by the camera using animal tracking algorithm.*

To identify the animals in the zoo, RFID readers are used. Since reading distance is very limited, from a few decimetres up to maximum 1 meter, the readers are placed at strategic locations where it is possible to get close to the animal. It should also be at places where the animal spends much of its time. This can be for example places where the animal eats and drink, or tunnel or gates that the animal often passes. Once a RFID reading is available the information can be fused with the computer vision system from the cameras to track the position of the animal. The tracking of animal will continue until it leaves the field of view or the tracking is lost for other reasons. If real-time tracking is not required it is also possible to track the animals position back in time from the RFID reading event. See Figure 5.7 for an example of a deployed tracking system in zoo.

When tracking system is used it is possible to overlay information about individual animals on the Live WebbTV stream. It is also possible to connect recorded video sequences to individual animals making it possible to watch pre-recorded content about animal of interest.

5.3.3 Augmented reality

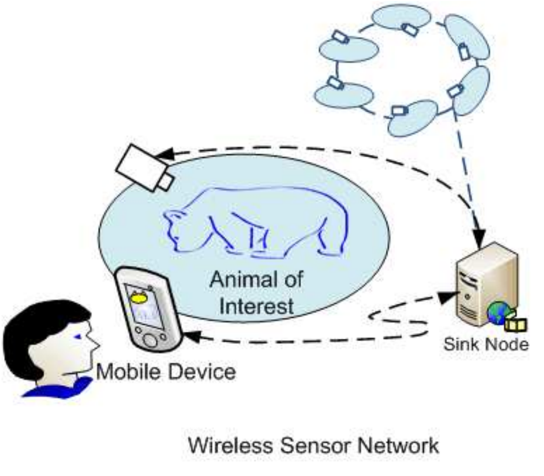


Fig. 5.8: *Systematic overview: The augmented reality information of an animal of interest is rendered on the handheld device based on the sink node data-set in WSN.*

Based on the tracking of individual animals it is possible to create an Augmented Reality (AR) based application to enhance the experience for the physical visitors in the zoo. In this application an AR visual representation of an animal of interest is overlaid on display screen to enable the users to view more information for a specific animal and/or site of interest. The software modules required for this application can be split into two components. One application is the AR client running on the handheld device. The second software component is the server application running on the sink node in the wireless network.

The server application running on the sink node will maintain a database of two kinds of data. One is the more static information that is added by the zoo staff or automatically transferred from the zoo keeping database. This is general information about the animals, like name, breed, gender, date of birth, offsprings, etc. The other type of data is the dynamically updated data from the wireless sensor network. This information is for example animals current position and collected media clips from of the animal.

The client application, on the user mobile phone, will be connected through the wireless network to the server application. Through this connection the client will have updated information about the animals current position and information related to each animal. To render this information the client must also know the position and orientation of the handheld device. This information is provided by the GPS, accelerometer and digital compass on the device. Once the geographical position and orientation of the handheld device is calculated the additional animal information can be rendered on the display. Since quite much information is available for each animal, the AR application should only overlay some information, like animals name and possible also the breed. The user could then click on a specific animal to get more detailed information and also previously recorded video clips.

6 Contributions and future research directions

6.1 Summary of contributed papers

Paper I
Johannes Karlsson, Jerry Eriksson and Haibo Li, "**Real-time video over wireless ad-hoc networks**," *in Proceedings of the fourteenth International Conference on Computer Communications and Networks (ICCCN 2005),* 17-19 October 2005, San Diego, California, USA.

This paper deals with real-time video transmission over wireless ad-hoc networks that are relatively limited in size, about 10-20 nodes. First, the main challenges when standard video codecs and standard routing protocols are used for this type of application are identified. This problem is looked at from several layers, from application to link layer. Two main issues are identified. One is at the routing layer where the proactive design of most ad-hoc routing protocols leads to an interruption in packet delivery when routes are changed. The fact that many ad-hoc routing protocols do not consider link quality when routes are selected is also causing problems. The other problem identified was that using intra frames for error control in video transmission was not an efficient method for this type of application. Secondly we proposed a dynamic reference picture selection method for error control of video transmitted over wireless ad-hoc networks. The work in this paper was based on network simulation.

Paper II
Johannes Karlsson, Jerry Eriksson and Haibo Li, "**Two hop connectivity for uniformed randomly distributed points in the unit square**," *in Proceedings of the First International Conference on Communications and Networking in China (CHINACOM 2006),* 25-27 October 2006, Beijing, China.

In this paper we present analytical solutions to the network topology as a function of communication range when two or three nodes are uniformly and independently distributed in a unit square. For the one-hop case, when two nodes are distributed, we gave the exact solution. For the two-hop case, when three nodes are distributed, we gave the upper and lower bounds. For this case we compared the results with simulated results. The work in this paper was based on analytical methods.

Paper III
Johannes Karlsson, Mikael Israelsson, Jerry Eriksson and Haibo Li, "**Efficient P2P Mobile Service for Live Media Streaming**," *in Proceedings of the Australian Telecommunication Networks and Applications Conference (ATNAC 2006)*, 4-6 December 2006, Melbourne, Australia.

In this paper we present a peer-to-peer method to stream multi-media content to users mobile phones. The application scenario is that a group of people watch the same real-time video over a GPRS connection. We also assume that each mobile phone is equipped with some local wireless connection, for example Bluetooth or WLAN. In our method each user only receives a substream of the total video stream and then all users share the received substreams with each other over the local wireless network. The complete system was implemented in a network simulator. A demo has also been implemented on real mobile phones. When this work was done in 2006 the data rate for mobile phones was still very low and the price for using data communication was very high. Even tough data rate has increased lately this method could still be used to reduce the load in the mobile phone network.

Paper IV
Johannes Karlsson, Adi Anani and Haibo Li, "**Enabling Real-Time Video Services Over Ad-Hoc Networks Opens the Gates for E-learning in Areas Lacking Infrastructure**," *International Journal of Interactive Mobile Technologies*, 2009.

In this paper a system is presented that can enable e-learning in many areas in the world that don't have good communication infrastructure. The solution is based on real-time video streaming over wireless ad-hoc networks. Two classes of hardware platforms that can be used to enable this system are presented. To improve the performance of real-time video transmission over wireless ad-hoc networks an extension to the DSR routing protocol was proposed. We call this extension real-time DSR (RT-DSR). We compared the performance of real-time video transmission when using this routing protocol to using standard DSR and AODV. The work in this paper was based on network simulations.

Paper V
Tim Wark, Peter Corke, Johannes Karlsson, Pavan Sikka and Philip Valencia, "**Real-time Image Streaming over a Low-Bandwidth Wireless Camera Network**," *in Proceedings of the Third International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2007)*, 3-6 December 2007, Melbourne, Australia.

In this paper a complete system that has been implemented for real-time image streaming in wireless sensor networks is presented. The compression algorithm used in this system is similar to the block-based hybrid class of video codecs. To reduce the complexity of the encoder no motion estimation is used. An energy analysis where the total energy usage of our compression algorithm is compared to sending uncompressed images is also presented.

Paper VI
Johannes Karlsson, Jean-Paul Kouma, Haibo Li, Tim Wark and Peter Corke, "**Demonstration of Wyner-Ziv Video Compression in a Wireless Camera Sensor Network**," *in Proceedings of the 9th Scandinavian Workshop on Wireless Ad-hoc & Sensor Networks (ADHOC'09)*, 4-5 May 2009, Uppsala, Sweden.

In this paper we presented a real implementation of wyner-ziv based encoding of video on a wireless sensor network platform. This approach is based on a method where motion estimation is performed in the receiver. Two types of frames are sent from the encoder at the sensor node. One is regular intra frames using conventional compression scheme. On the other type, the Wyner-Ziv frames, block wise Discrete Cosine Transform (DCT) is applied and the coefficients are then coarsely quantized. The Wyner-Ziv frame is used at the decoder to perform motion estimation from the previously decoded frame. In the paper we also give a presentation of the sensor network hardware platform we have used for this implementation.

Paper VII
Johannes Karlsson, and Haibo Li, "**Moving Motion Estimation from Encoder for Low-Complexity Video Sensors**," *manuscript to be submitted.*

In this paper a method to reduce complexity at the encoder for wireless video sensor networks is presented. The approach is based on removing the motion estimation step at the encoder. Instead motion prediction is performed at the decoder side. This information is sent back from the decoder to the encoder where it is used for motion compensation. Since this method is sensitive for network delay we present the performance of our method for different network delays. We also present experimental results of network delay from a deployed sensor network.

Paper VIII
Karin Fahlquist, Johannes Karlsson, Keni Ren, Li Liu, Haibo Li, Shafiq ur Réhman and Tim Wark, "**Human Animal Machine Interaction: Animal Behavior Awareness and Digital Experience**," *in Proceedings of ACM Multimedia 2010 - Brave New Ideas*, 25-29 October 2010, Firenze, Italy.

In this paper we present a brave idea for human-animal interaction. The system can be divided into two parts where one is the animal to human communication and the other is human to animal communication. For the animal to human communication we propose the use of wireless video sensor networks that capture and analyse information about the animal. For the human to animal communication we propose the use of vibrotactile collars. The work in this paper was based on the large scale video sensor network deployment in the Digital Zoo project.

Paper IX
Johannes Karlsson, Tim Wark, Keni Ren, Karin Fahlquist and Haibo Li, "**Applications of Wireless Visual Sensor Networks - the Digital Zoo**," *in Visual Information Processing in Wireless Sensor Networks: Technology, Trends and Applications,* IGI Global, in press

In this book chapter we present the complete Digital Zoo system. We first present our hardware platforms that are used to build the video sensor network. We then describe our large scale network deployment in the zoo. We give energy models for how the solar-powered sensor nodes will be duty-cycled for different times of year. We also briefly present our computer vision system used for tracking of animals. Finally we present a number of applications for the digital zoo.

Paper X
Johannes Karlsson, Keni Ren and Haibo Li, "**Tracking and Identification of Animals for a Digital Zoo**," *in Proceedings of The 1st IEEE/ACM Internet of Things Symposium,* 18-20 December 2010, Hangzhou, China.

In this paper a system for identification and tracking of animals is presented. This system is based on two different components. One is a visual sensor network where animals position are being tracked. The other part is RFID readings to identify animals at fixed location. By fusing this information it is possible to track individual animals.

6.2 Summary of contribution

The contributions from this thesis can be divided into three areas. The key contributions to each area are presented here. Real-time video over wireless ad-hoc networks. The contributions to this area are from the papers I, II, III and IV.

- Dynamic reference picture selection protocol to handle error control for video transmission over wireless ad-hoc networks.
- Proactive extension for the DSR routing protocol to improve performance when sending real-time video.
- Analytical solutions to the network topology as a function of communication range when two or three nodes are uniformly and independently distributed in a unit square.
- A solution based on wireless ad-hoc networks to come over the problems of delivering e-learning to areas that don't have good communication infrastructure.

- Protocols for P2P multicast of real-time video in hybrid GPRS / wireless ad-hoc networks.

Low-complexity video encoding for wireless video sensor networks. The contributions to this area are from the papers V, VI, and VII.

- A low-complexity video compression algorithm for video sensor networks based on block based hybrid coding were no motion estimation is used.
- Protocol design for streaming of images in multi-camera sensor network. This was implemented and evaluated on a set of wireless video sensor nodes.
- Design and implementation of A wyner-ziv based video compression on actual wireless sensor network hardware.
- A low-complexity video compression algorithm based on removing motion estimation from encoder and instead perform motion prediction at decoder.

Applications and system aspects of wireless video sensor networks. The contributions to this area are from the papers VIII, IX, and X.

- A concept based on wireless sensor networks for human animal machine interaction. This was selected to the "Brave New Ideas" session on the ACM Multimedia 2010 conference.
- Design,implementation, and deployment of a large scale video sensor network for a "Digital Zoo".
- Animal identification and tracking system based on information fusion from visual and RFID data.

6.3 Future research directions

The deployed large-scale video sensor network in the "Digital Zoo" project gives us a unique basis for future research in this field. One interesting research direction is to continue the work on energy models for solar-powered video sensor nodes for Arctic conditions. Here the theoretical models will be compared to collected data from a long-term deployment of video sensor nodes from the test site. Another interesting area is to extend the identification and tracking of animals to work on a multi-camera tracking basis. This will introduce several new research challenges, but will also enable many new applications. One application based on the animal tracking is Augmented Reality for physical visitors in zoo. There are still many open research issues within this area before this application can be implemented. For low-complexity compression of video in wireless video sensor networks the framework of compressive sensing promises many possibilities. This is an area that will definitely be explored for future research in this area.

Papers

Bibliography

Bibliography

[1] http://pcl.cs.ucla.edu/projects/glomosim/.

[2] "The network simulator ns-2." http://www.isi.edu/nsnam/ns/.

[3] B. Warneke, M. Last, B. Liebowitz, and K. S. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Computer*, vol. 34, pp. 44–51, 2001.

[4] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Personal Communications*, pp. 46–55, April 1999.

[5] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina, "Performance comparation of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications Special Issue on Advances in Mobile Ad Hoc Networking*, vol. 8, February 2001.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *ACM Conference on Communications Architectures, Protocols and Applications, SIGCOMM '94 London, UK*, pp. 234–244, ACM, ACM, August 1994.

[7] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," RFC 3626, Internet Engineering Task Force, October 2003.

[8] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in *The First International Conference on Mobile Computing and Networking, ACM MOBICOM'95, November 13-15, 1995, Berkeley, California, USA*, pp. 86–95, ACM, ACM, November 1995.

[9] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing." RFC 3561, July 2003.

[10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, vol. 353, Kluwer Academic Publishers, 1996.

[11] "Iso/iec 14496-2 information technology - coding of audio-visual objects." International Organization for Standardization, December 2001. Second edition.

[12] "Xvid open source mpeg-4 implementation." www.xvid.org.

[13] "Iso/iec 14496-10:2005 information technology – coding of audio-visual objects – part 10: Advanced video coding." International Organization for Standardization, 2005.

[14] H. S. Ralf Schäfer, Thomas Wiegand, "The emerging h.264/avc standard," tech. rep., EBU Technical Review, January 2003.

[15] P. Corke, S. Sen, P. Sikka, and T. Wark, "Wireless sensor network: two-year progress report," Tech. Rep. TR 06/249, CSIRO ICT Centre, August 2006. http://www.csiro.au/files/files/p8zh.pdf.

[16] "Iso 11785:1996. radio frequency identification of animals – technical concept." International Organization for Standardization, Geneva, Switzerland.