Computer Networks and Distributed Systems

IAM-05-002

June 05

Computer Networks and Distributed Systems

Retreat of the
”Computer Networks and Distributed Systems” research group
Institute of Computer Science and Applied Mathematics
University of Bern

June 27-30
Griesalp, Kiental, Switzerland
http://www.iam.unibe.ch/~rvs/events/

Abstract

The research group ”Computer Networks and Distributed Systems” of the Institute of Computer Science and Applied Mathematics at the University of Bern, headed by Prof. Torsten Braun, organized an internal retreat from June 27-30, 2005 at Griesalp / Kiental. The focus of this retreat was to present and discuss recent research results and currently ongoing research activities of the research group members. The research group members gave eleven presentations, most of them in the areas overlay networks, wireless mesh and ad hoc networks as well as sensor networks. Extensive time (typically 90 minutes per talk) has been allocated to allow detailed presentations and discussions. This technical report summarizes the various talks and describes mostly unpublished work that is currently in progress.

CR Categories and Subject Descriptors: C2.1 [Computer-Communication Networks]: Network Architecture and Design; **C2.2 [Computer-Communication Networks]:** Network Protocols; **C2.5 [Computer-Communication Networks]:** Local and Wide-Area Networks; **C2.6 [Computer-Communication Networks]:** Internetworking.

General Terms: Algorithms, Design, Performance, Sensors, Mesh.

List of Presentations

Overlay and Peer-to-Peer Networks

1. MC-FTP: A Multicast File Transfer Protocol for Efficient Data Dissemination
   Dipl. phil. nat. Marc Brogle and Dragan Milić, University of Bern
2. EuQoS Multicast Middleware: Basic Architecture Overview and Concepts
   Dipl. phil. nat. Marc Brogle and Dragan Milić, University of Bern
3. Landmark Positions in an n-Dimensional, Virtual Space
   Dipl. phil. nat. Dragan Milic, University of Bern
4. New Results in the XBAC Project
   Dipl. phil. nat. Matthias Scheidegger, University of Bern

Wireless Mesh and Ad hoc Networks

5. Wireless Mesh Networks
   Dipl. phil. nat. Thomas Staub, University of Bern
6. Setup of Simulations on Heterogeneous Networking with CAHN
   Dipl. phil. nat. Marc Danzeisen, University of Bern
7. Cooperation in Multi-hop Wireless Networks
   Dipl. phil. nat. Attila Weyland, University of Bern

Sensor Networks

8. TCP Support for Sensor Nodes,
   Prof. Dr. Torsten Braun:, University of Bern
9. Distributed Event Detection in Wireless Sensor Networks
   Dipl. phil. nat. Markus Waelchli, University of Bern
10. Positioning in Wireless Sensor Networks
    Dipl. phil. nat. Thomas Bernoulli, University of Bern

Fund Raising with EU Projects

11. Background Information to EU Projects
    Dr. Marc-Alain Steinemann, University of Bern

Acknowledgements: This event has been mainly supported by Mittelbauvereinigung der Universität Bern (MVUB).

MC-FTP: A Multicast File Transfer Protocol for Efficient Data Dissemination

Marc Brogle and Dragan Milić

brogle|milic@iam.unibe.ch

1 Introduction

Peer-to-peer (P2P) and file-sharing applications like BitTorrent [1] and many others have become very popular. There are different approaches, which use P2P networks for efficient data dissemination:
Slurpie [2] builds an overlay network between downloading clients only if better performance can be gained that way. It supports HTTP and FTP downloads from servers. Files are split into blocks and block lists are represented as bit vectors. Each peer stores information about n (const.) other peers and they update each other periodically. Downloads occur normally between peers, the server is visited only if no peer has the needed block.
Bullet [3] splits the data into disjoint object sets, which are disseminated disjointly to peers considering bandwidth. Peers locate and retrieve missing data from other peers. Summary tickets, which are summarizing so called working sets representing progress on peers, get exchanged periodically between peers. Bullet builds a mesh on top of an existing overlay tree.
FastReplica [4] replicates large files among n nodes, where n is typically between 10-30 nodes. The distribution step sends a part of the original file called *subfile* and a distribution list of nodes to which the subfile has to be sent to. In the collection step all the nodes send their subfile to the remaining nodes in the group.
FTP-M [5] extends the classical FTP protocol (client + server) and uses TCP-M allowing TCP-like reliable multicast. TCP-M splits TCP connections and fuses ACKs. FTP-M offers a convenient API based on BSD sockets and has two modes, the normal and the one-to-many mode. Data transfer occurs in the following two phases: the *negotiation phase* opens the command connections for FTP and the *data transfer phase* makes the file upload also called FTP push.
Our proposal MC-FTP is a protocol for efficient data dissemination using multicast, either native IP multicast or overlay multicast, and enables anonymity under certain preconditions. It expects users to cooperate on forwarding data for others. MC-FTP allows efficient usage of network resources and reduces the load at the server regarding network and CPU usage. It allows classical file sharing between users and can convert classical server based 1:n data dissemination for e.g. patches and new ISO-images to a P2P file distribution. MC-FTP has a completely de-centralized architecture and does not need any infrastructure support.

2 MC-FTP Simple Action Flow Overview

To understand action flow in MC-FTP a simple scenario will be used. The details of the involved parties like File Leaders, senders and receivers will be explained in the next section.
First hosts that are interested in a certain file for sending and/or receiving join the File Management Group for this file. Then the File Leader assigns the sending hosts and starts to send keep-alive messages, which include which chunks will be sent on which groups at what rate (see also Figure 1).

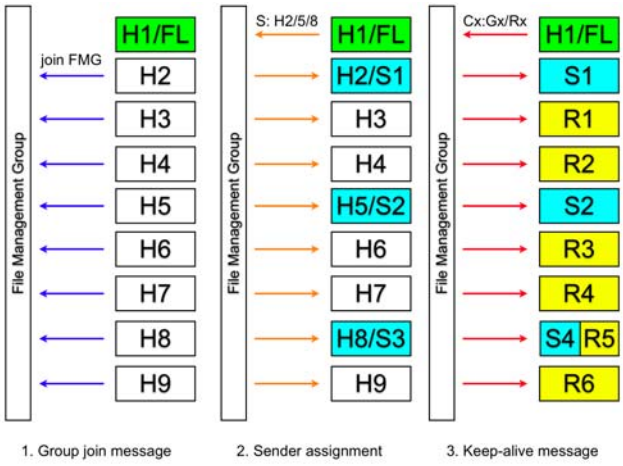


Fig. 1. Host joining, sender assignment and keep-alive message sending

After the involved nodes learn on which groups what chunks will be send, senders and receivers join the corresponding groups (see also Figure 2). Then the senders can start to send the chunks with the specified rate defined by the File Leader in the keep-alive message that are periodically resent. The receivers subscribed to the corresponding group will then receive these chunks (see also Figure 3). Senders can also be receivers for other chunks they don't have locally available yet.

3 MC-FTP Architecture and Components

The MC-FTP protocol is multicast technology and addressing scheme independent. It uses native IP multicast in local networks or operates on top of an overlay network using ID based multicast groups. The overlay approach allows good scalability concerning the group address space issue and also overcomes the limitations of multicast support in the Internet. Anonymity for all peers is granted when the underlying overlay network framework / technology also supports multicast anonymity.

Information about a file that has to be disseminated is stored in a so-called File Descriptor (torrent-like) containing different information about the file. This information includes a MD5-hash of the file (= file ID), a list of File Chunks holding redundant data and a Circular Pairwise Checksum (MD5) of chunks in-order (see also Figure 4).

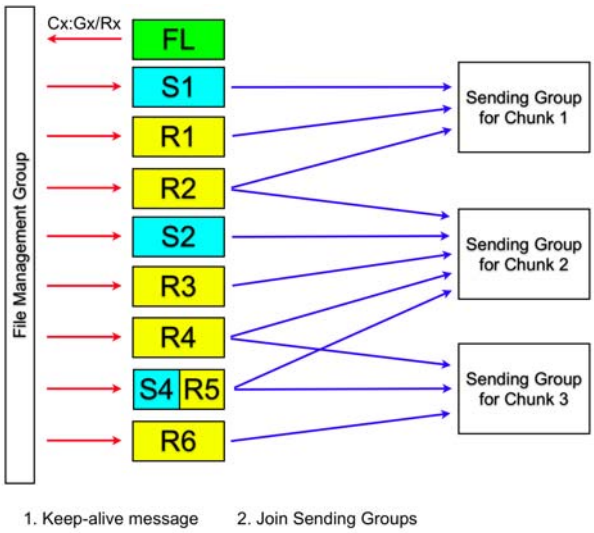


Fig. 2. Joining of senders and receivers to the chunk sending groups

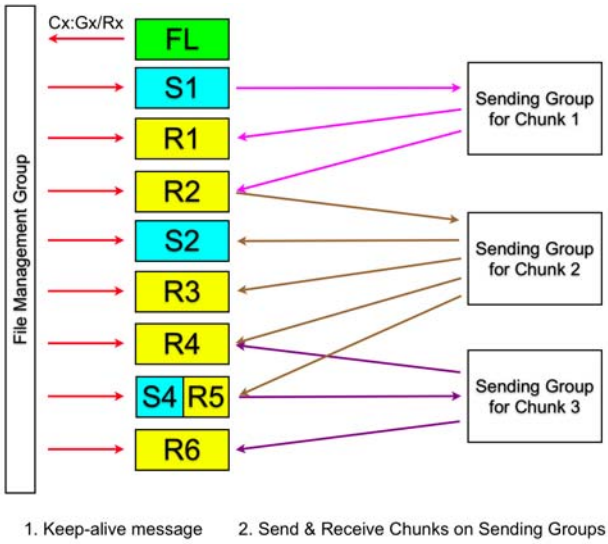


Fig. 3. Sending and receiving of chunks

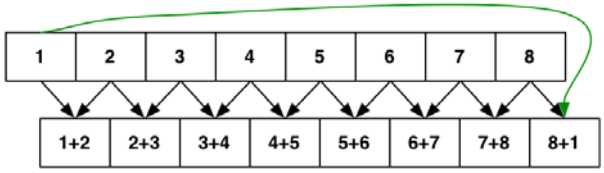


Fig. 4. Circular Pairwise Checksum

The architecture of MC-FTP also includes two DHTs (distributed hash table). One is used for group reservation management and the other is used for mapping a File Descriptor to a File Management Group.

One File Management Group is assigned to each file. Files are defined by the corresponding File Descriptors. Each File Management Group has one File Leader. The File Management Group is reserved by the File Leader and stored in the corresponding DHT (key: File Descriptor, value: group address). All peers interested in a certain file lookup the File Management Group address in the corresponding DHT and then join this group. All peers involved in a file distribution use the File Management Group for message exchange.

A new Downloading peer tries to locate the address of the existing file management group via a lookup in the according DHT. If there is no group defined in according DHT it retries to find the group after a certain back-off time. Otherwise it joins the file management group.

A file-providing peer, even if it has only partially downloaded the file, tries to locate the address of the existing file management group via a lookup in the according DHT. If no existing file management group address is found, it tries to become the File Leader for the corresponding file, which could raise concurrency issues.

Each peer involved in a file transfer creates for itself per file a unique ID and a public / private key pair for itself to use for encryption and signing of the communication data.

The File Leader manages all peers interested in a certain file like downloaders, pure senders and mixed peers. File leaders periodically send keep alive messages containing the File Leader's unique ID, the current groups for streaming data and for each data-streaming group the rate and order of the chunks. If no keep-alive messages received after a certain time a new File Leader will be chosen by negotiation. File leader negotiation is done in a distributed manner with a voting or challenging algorithm, which still has to be determined. Clustering of File Leaderships on a peer should be possible. Good connected peers having fast and stable connections should accumulate leaderships up to a certain degree. The optimal number of accumulations for different scenarios still has to be determined.

A File Leader manages the sending peers, determines which chunks should be sent to which groups and defines the sending rate for the chunks. Periodically the File Leader asks all members in its group to report their status using a status message containing the unique ID and the public key of the peer, their locally available chunks and the available bandwidth. With the status request, the File Leader also sends its public key, which is used by group members to encrypt their status messages. File leaders manage the leasing of multicast groups. They reserve multicast addresses for file

management groups and for the groups on which the file chunks are sent. They also release the multicast group addresses that are not in use.

4 Open Issues and Outlook

Some open questions remain:
How are File Descriptors located? They can be downloaded, which is easy to implement (BitTorrent-like) or a distributed hash table search engine could be used.
How to perform integrity checks, signing and trustworthiness of a File Descriptor?
How to detect malicious peers and solve the problem of fake new File Leaders? Malicious peers could declare themselves as new File Leaders, existing members of the file share group would detect this but new members cannot know which File Leader to trust. The correction is not easy for new members joining an existing compromised File Management Group.
What is the tradeoff between anonymity and overall complexity?
How to determine the maximum number of file management groups to subscribe to?
How many files should a peer offer for downloading and if it has a lot of files how would it be to cycle through them offering only a still to be determined number of files in a certain time frame?
For the future the message protocol should be defined in more detail. Different DHTs like Pastry or Chord should be evaluated in the context of group address management and for file-to-management-group-address mapping. Also the possibilities for a File Descriptor search engine based on DHTs or other mechanisms should be evaluated. Different erasure encodings that could be used when splitting a file into chunks should be analyzed to optimize data recovery from partial downloads. Also a way should be found to efficiently assign sending of chunks and the according groups among the senders to minimize duplication of the received data on downloaders. Finally cooperation with existing protocols (HTTP, FTP, etc.) has to be evaluated.

References

1 Incentives Build Robustness in BitTorrent, Bram Cohen, May 2003
2 Slurpie: A Cooperative Bulk Data Transfer Protocol, Proceedings of IEEE INFOCOM, Rob Sherwood, Ryan Braud, Bobby Bhattacharjee, March 2004
3 Bullet: high bandwidth data dissemination using an overlay mesh, SOSP ’03: Proceedings of the nineteenth ACM symposium on Operating systems principles, Dejan Kostic, Adolfo Rodriguez, Jeannie Albrecht, Amin Vahdat, 2003
4 FastReplica: Efficient Large File Distribution within Content Delivery Networks, Proceedings of USITS ’03: 4th USENIX Symposium on Internet Technologies and Systems, Ludmila Cherkasova, Jangwon Lee, 2003
5 FTP-M: An FTP-like Multicast File Transfer Application, Manamohan Mysore, George Varghese, 2001

EuQoS Multicast Middleware: Basic Architecture Overview and Concepts

Marc Brogle and Dragan Milic

brogle|milic@iam.unibe.ch

1 Introduction

The *EuQoS* project [1] aims to resolve the outstanding design issues presently associated with the delivery of end to end QoS service across heterogeneous networks. With the help of EuQoS these issues should be solved and the infrastructures should be upgraded so that new applications can be supported by the Internet and new service packages can be offered by operators, ISP and other service providers.
Support for QoS in IP multicast is difficult to achieve due to lack of wide deployment of IP multicast in the Internet and it seems that this will probably not change in the near future, even with adoption of IPv6. The *Multicast Middleware* is a so called feature of the EuQoS project, aiming to solve the present difficulties of multicast communication. This is achieved by introducing an overlay network between the involved end systems in order to distribute multicast data. Overlay communication will be based on unicast TCP and UDP communication to ease the use of available QoS mechanisms. These mechanisms include measurement-based ("Best-Effort") QoS or QoS mechanisms provided by other modules of the EuQoS project. Existing applications do not need to be modified to use the provided overlay multicast capabilities, since the Multicast Middleware is completely transparent to the end-system.
The difference between *native multicast* and *overlay multicast* can be seen on Figure 1. While native multicast is the most efficient way of distributing data from one sender to multiple receivers, overlay multicast introduces some redundancy in terms of amount of sent data over the involved links. The goal is to minimize the additional data transfer volume introduced by using an overlay network to distribute multicast data.

2 Multicast Middleware Concepts

The Multicast Middleware can bee seen as a special variant of end system multicast [2], where the applications of the end system do not need to be aware of the introduced multicast supporting framework.
Multicast traffic on the end system is captured and re-routed through the overlay network that has been built between the end systems. The concept of intercepting multicast traffic and routing it through an overlay network was also presented in [3].

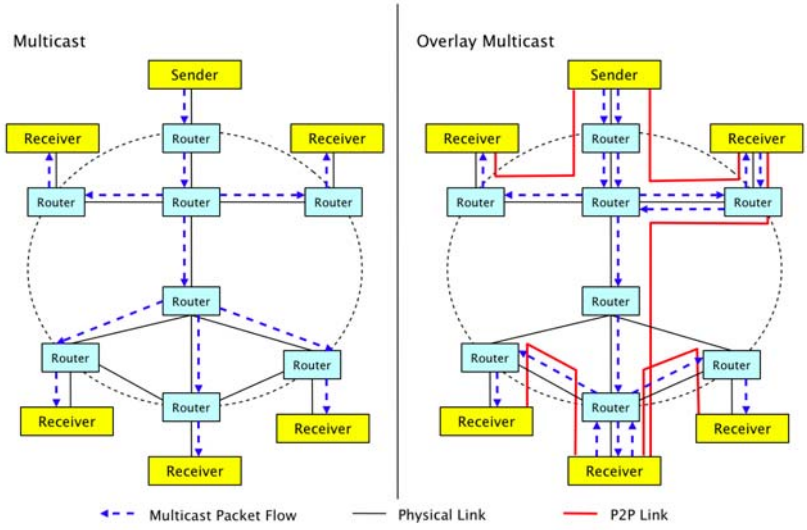


Fig. 1. Traditional Multicast vs. Overlay Multicast

The overlay links using TCP and/or UDP can be setup regarding QoS requirements signaled directly by the application. This can be done either using existing protocols (like RSVP) or using the provided framework by requesting QoS services via the provided mechanism (Webservice). The preferential use of TCP helps to overcome some connectivity restrictions introduced by firewalls and NAT routers [4].
The Multicast Middleware aims to be independent of the underlying QoS mechanisms. It can either use the EuQoS QoS signaling introduced in the EuQoS project or it can use measurement-based ("Best-Effort") multicast to bridge gaps where no (EuQoS) QoS support is provided by the underlying network. To determine the links for establishing the connections between peers to build the overlay network, the Multicast Middleware has to combine QoS information received by the QoS supporting underlying network architecture (like EuQoS) and the measured information about existing non-QoS links. The inter-corporation of both mechanisms in a simple network environment is shown in Figure 2.

3 Multicast Middleware Basic Architecture

The following example explains the basic architecture of the Multicast Middleware. It shows the flow of the data and the message exchange between a sender and a receiver in the context of video streaming use case. First an application on the receiver side sends a multicast join message to announce that it is interested in a certain multicast data transfer. On the sender side data (in the example a video stream) is sent as UDP multicast traffic, which is then routed through the overlay (using TCP) accordingly to the setup routes after the join message has been processed.

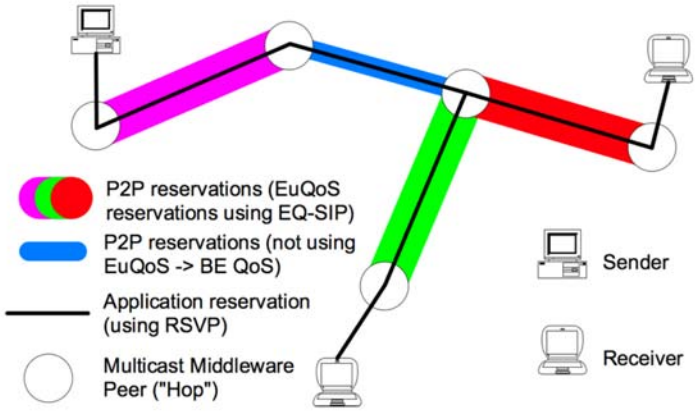


Fig. 2. QoS Reservations using BE-QoS or EuQoS-Signaling

In Figure 3, the VLC [5] (video LAN client, a freely available video playback and streaming software) application is sending the group join message (IGMP) which is then intercepted by the Multicast Middleware and sent through the overlay network as a P2P join message to setup the routes accordingly. The interception of the multicast UDP and IGMP messages done by TAP [6] (a virtual ethernet network device), which then forwards the packets to the Multicast Middleware process on the end system.

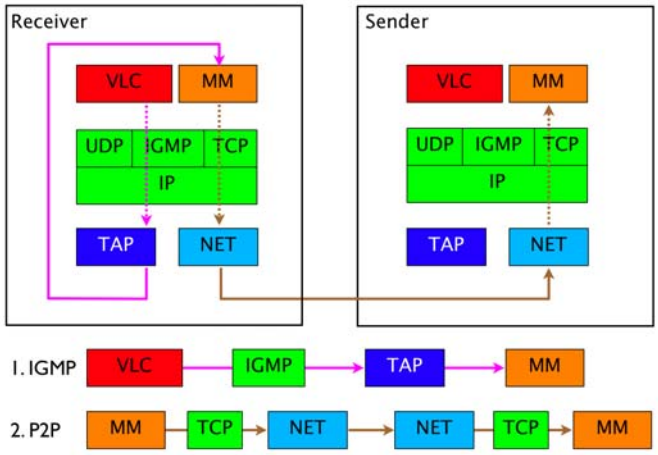


Fig. 3. Receiver initiated Multicast Group join

Figure 4 shows how the application (VLC) on the sender side is sending standard multicast UDP packets, which get again intercepted by the TAP interface. The packets are processed by the Multicast Middleware, which sends them over the

Internet using the TCP overlay links. At the receiver's side, the Multicast Middleware processes the received packets and sends them trough the TAP interface as standard UDP Multicast packets, which get then received by the VLC application.

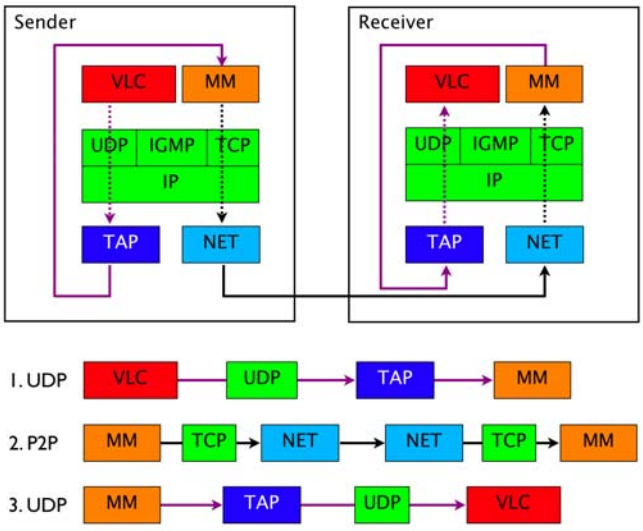


Fig. 4. Sender controlled Broadcasting of UDP Multicast Traffic

References

1 EuQoS (End-to-end Quality of service support over heterogeneous networks), official website at http://www.euqos.org
2 A case for end system multicast, Yang-Hua Chu, Sanjay Rao, Hui Zhang, 2000
3 Quality-of-Service for Internet Multicast, Prof. Dr. T. Braun, 2003
4 Connectivity restrictions in overlay multicast, Aditya Ganjam, Hui Zhang, 2004
5 VLC (video LAN client), official webpage at http://www.videolan.org,
6 Universal tun/tap driver, official webpage at http://vtun.sourceforge.net/tun,

Landmark Positions in an n-Dimensional Virtual Space

Dragan Milić

milic@iam.unibe.ch

1 Introduction

The communication latency prediction in the Internet is becoming more important due to the increasing number of Internet applications that attempt to optimize their network communication. This is done by considering the network distance across which data is transferred. Such applications range from peer-to-peer networks, which can exploit the host network proximity information to optimize the choice of neighboring peers to data downloading agents, which can choose the mirror with the lowest RTT to maximize the TCP throughput for the file download.
To be able to determine a one-way communication latency between two hosts in the Internet, it is necessary to have synchronized clocks. Due to this limitation, the communication latency measurement and prediction in the Internet is usually limited to measuring and predicting the round trip time (RTT). Since not all paths in the Internet are symmetric (mostly due to the policy based routing of BGP) it cannot be assumed that the one-way latency is the half of the RTT.
For a given group of n hosts, measuring the RTT between all host pairs would result in an optimal result but does not scale ($O(n^2)$!). The alternative is to predict the RTT between two hosts. In the case of the RTT prediction the complexity of the procedure is leveraged with the precision.

1.1 RTT Prediction using Virtual Spaces

There are numerous RTT prediction schemes already proposed like IDMaps[1] GNP[2], Vivaldi[3] and ICS[4]. The most promising proposals try to assign to each host a set of coordinates (a point in an n-dimensional virtual space) in such way, that the RTT between any host pair in the virtual space is predicted by the Euclidean distance function between the points assigned to the hosts. Since the Internet cannot be represented as an ideal Euclidean metric space, all of those proposals try to determine the best coordinates for each host (to embed a host into a virtual space) by minimizing the square error between the predicted and measured RTT. The methodology for embedding the hosts range from using the simplex downhill [5] method used in [2, 6] and principal component analysis used in [4, 7] to physical model simulations [3, 8, 9].

1.2 Different Approaches to Embed Hosts in Virtual-Spaces

There are two general approaches for embedding hosts in a virtual space: landmark-based and landmark less approaches.

1.2.1 Landmark-Based Approaches

The landmark-based approaches determine the host coordinates in a virtual space by measuring the RTT between a host and a set of predetermined hosts – the landmarks. The coordinates of a host are usually determined by minimizing the square error between measured and predicted RTT between the host and the landmarks [2, 6] or by applying the PCA transformation [4, 7] to the measured values. The disadvantages of landmark-based approaches are that they usually require some kind of infrastructure (a predetermined set of landmarks), which are used by all hosts in the system (scalability issue!). The disadvantages of the landmark based approaches have been addressed in [6] and [10].

1.2.2 Landmark-Less Approaches

Unlike the landmark-based approaches, the landmark-less approaches require no infrastructure to achieve embedding of hosts in a virtual space. The landmark-less approaches usually represent a distributed simulation of a dynamic physical model (such as the simulation of spring systems [3, 8]) which adapts to the measured RTTs which between the hosts involved in a communication (e.g. RPC) and converges towards a stable configuration in which the overall RTT prediction error of the system is minimized. Although such physical models work very well under ideal conditions, they seem to be very instable when they are applied to data collected from the Internet. Usually, the whole system starts oscillating instead of reaching a stable state, which leads to constantly changing coordinates for all hosts in the system. The stability issue of the physical models is usually addressed by introducing a dampening physical factor such as the friction, which damps the oscillation of the system [8].

1.3 Problems with Host-Embedding

All mentioned positioning systems suffer from different kind of problems. The most severe problems are triangular inequality, number of dimensions, stability and non-determinism.

1.3.1 Triangular Inequality

The triangular inequality is the essential property of all metric spaces. It states that for each three points in an n-dimensional space (a,b,c) and a distance function d, the inequality $d(a,b)+d(b,c)\geq d(a,c)$ must hold. The triangular inequality holds for every network if ideal routing is assumed, which is not the case in the Internet, where the policy-based BGP routing leads to violation of the triangular inequality.

1.3.2 Number of Dimensions

Since there is no "natural" number of dimensions for the Internet, each approach must either use a predefined number of dimensions or have a scheme for calculating the minimal number of dimensions needed. The number of dimensions influences the computational complexity and the minimal number of landmarks needed to locate a host (at least n+1).

1.3.3 Stability

Some proposals [9, 3] use a (distributed) physical model simulation to determine the coordinates of hosts in the Internet. The premise for the success of such approaches is that the simulation converges towards a stable state, which is not always the case, since the measured data from the Internet often violates the prerequisites for the stability of those systems (e.g. trough violation of the triangular inequality).

1.3.4 Non-Determinism

The approaches [2, 6], which use the downhill simplex method for the function minimization, suffer from the non-deterministic property of the downhill simplex: the result of the minimization depends on the initial simplex, which is usually (pseudo-) randomly chosen.

2 Calculating Landmark Positions

In [2] the landmark positions are calculated by minimizing the distance function between the measured and calculated RTTs. The drawbacks of the GNP approach are the non-determinism of the coordinate calculation (the downhill simplex method uses a starting simplex which is usually randomly determined) and the lack of the linear-dependency check (the GNP approach has no mechanism to determine if the chosen landmarks are appropriate for the given number of dimensions).
In this paper we propose an algorithm for determining the coordinates of the landmarks, which is able to determine the minimal number of, coordinates that are needed to represent the landmark information. The proposed algorithm also considers the contribution of each landmark to the dimensionality of the system.

Description of the Algorithm

The input of the algorithm is a $n \times n$ matrix D_{start} representing the distances between the hosts which are the potential landmarks. The output of the algorithm is a set of landmarks (L, a subset of the input hosts), the number of dimension d and the coordinates of the landmarks.
At the beginning, the maximal subset T_{max} of the input hosts is determined in which the triangular inequality holds for all triplets,. All hosts, which are not in this subset, are not considered to be used as landmarks. One host from T_{max} is chosen and

designated as the first landmark in the landmark set *L*. This host is used as the origin of the coordinate system.

For each host $l \in T_{\max}$ the "height" of the simplex h_l in the higher dimension is calculated using the n-dimensional version of Heron/Tartaglia formula for simplex volume computation. h_l is the distance between the new point p_l of the simplex representing this host and its projection p'_l in the hyper plane spanned by other landmarks (P_l).

If h_l is greater than a predefined threshold $h_{l_{min}}$ then the number of dimensions of the system is increased by one and the landmark is added to *L*. The coordinates of the landmark are calculated using following formula: $p_l = p'_l + v \times h_l$, where *v* is an orthogonal vector to the hyper plane spanned by the all other landmarks $l_x \in L$ and calculated as an n-dimensional cross product of those vectors. The projection of *p* in the hyper plane P_l is calculated using the n-dimensional equivalent of barycentric coordinates.

If $h_l < h_{l_{min}}$ then the host is added to the set of the correction hosts H_{corr} which are used later to adjust the landmark positions.

After the initial simplex has been defined and if H_{corr} is not empty, then the coordinates for each correction host are calculated using multilateration (Newton iteration [11]). After this step, the set of correction hosts H_{corr} are added to *L* and the Newton Iteration is used to minimize the square error of the landmark positions.

3 Outlook and Open Questions

The described approach has to be implemented and its performance has to be compared to other approaches (such as GNP, ICS and Vivaldi) using simulations with generated topologies and RTT measurement data collected from the Internet (e.g. Planet Lab). The open question remains how to handle the hosts, which do not satisfy the triangular inequality with landmarks. One of the possibilities is to treat them separately as a kind of a "parallel universe" which is isolated from the rest of the system. Another open issue is the optimal value for the threshold $h_{l_{min}}$ for the minimal height in the new Dimension. This value could be determined by considering the RTT measurement deviation.

References

1 P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "Idmaps: a global internet host distance estimation service," *IEEE/ACM Trans. Netw.*, vol. 9, no. 5, pp. 525–540, 2001.

2 T. E. Ng and H. Zhang, “Predicting internet network distance with coordiantes-based approaches,” in *IEEE Infocom02*, (New York / USA), June 23-27 2002.
3 F. Dabek, R. Cox, F. Kaashoek, and R. Morris, “Vivaldi: a decentralized network coordinate system,” in *SIGCOMM ’04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 15–26, ACM Press, 2004.
4 L. Tang and M. Crovella, “Virtual landmarks for the internet,” in *Internet Measurement Confgerence 03*, October 2003.
5 J. Nelder and R. Mead, “A simplex method for function minimization,” *Computer Journal*, vol. 7, pp. 308–313, 1965.
6 M. Costa, M. Castro, A. Rowstron, and P. Key, “Pic: Practical internet coordinates for distance estimation,” in *International Conference on Distributed Systems*, (Tokyo, Japan), March 2004.
7 H. Lim, J. C. Hou, and C.-H. Choi, “Constructing internet coordinate system based on delay measurement,” in *Internet Measurement Confgerence 03*, October 2003.
8 C. de Launois, “A stable and distributed network coordinate systems,” tech. rep., Université catholique de Louvain, December 2004.
9 Y. Shavitt and T. Tankel, “Big-bang simulation for embedding network distances in euclidean space,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 993–1006, 2004.
10 T. E. Ng and H. Zhang, “A network positioning system for the internet,” in *USENIX 2004*, (Boston MA, USA), pp. 14–15, June 2004.
11 Åke Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 1996.

New Results in the XBAC Project

Matthias Scheidegger

mscheid@iam.unibe.ch

1 Introduction

The XBAC project is based on the idea to create a distributed Internet distance estimation service for overlay networks based on groups of nodes located close to each other. Nodes in these groups exchange measurement data gathered from active probing or from passive traffic monitoring. The resulting pool of available data enables robust distance estimation, and even prediction, to peer groups.
In order to build this architecture an adequate way to group nodes is needed. To this end, we introduce the concept of a node cluster. In terms of QoS behavior over time, nodes inside a cluster appear practically equivalent to nodes outside the cluster. Accordingly, nodes in a group can directly use measurements made by another node inside the group. More formally, given a distance function $d(n,m) \geq 0$ and a set of nodes N, we call the set $C \subseteq N$ a cluster if

$$d(o,n) \approx d(o,m), \quad \forall n,m \in C, \quad \forall o \notin C \qquad (1)$$

Two aspects of this definition are useful. First, the above-mentioned requirements for the nodes in an organizational group a satisfied if the group is also a cluster.
Second, the above definition divides the network into a system of hierarchical groups, which greatly helps reducing the architecture's complexity. From the viewpoint of a given organizational group, the network divides into a set of clusters. Due to their characteristics, it is sufficient to only store measurement data per cluster instead of storing it per peer node. This reduces the complexity of the system by several orders of magnitude. However, even given this significantly lower complexity, probing and storing distance information for every group pair in the network couldn't scale. Accordingly, the nodes in a group only store information about peer groups that are "of interest," i.e. the group has, or has had, active connections to nodes in these groups. Communicating between groups can solve distance estimation requests concerning clusters to which too little data is available within a single group.

The concept of grouping endpoints by proximity has been used in other work. IDMaps [1] is another architecture for end-to-end distance estimation. The approach assigns endpoints to IP address ranges. Special nodes, called tracers, are deployed throughout the network backbone. Network delays are then estimated by taking the sum of the delays from the endpoints' IP ranges to their respective nearest tracer, plus the delay between the tracers. Thus the approach becomes very scalable, at the cost of introducing additional infrastructure to the network. MULTI+ overlay multicast tree construction [2] is another approach using IP ranges to group endpoints by proximity. A general API for distance estimation services called SONAR has been proposed in

[3]. It does not include specifics about how distance estimation should be done, however. mOverlay [4] uses a different approach to identify clusters: Joining nodes measure their distance to a set of candidate clusters. Based on these measurements they iteratively select other clusters until they find a suitable one to join.

2 Structure of the Distance Estimation Service

An overlay network structure lends itself to realize a distance estimation service as proposed above, due to its advantages in terms of scalability, robustness, and deployability. The network must be structured on two levels: first, data exchange and management within organizational groups, and second, group discovery and exchange of distance estimation requests on the inter-group level.
Ring-based distributed hash tables (DHTs) allow for robust organization and storage of information while providing a relatively efficient lookup mechanism. In fact, DHTs such as Chord or Pastry may solve the design issues of both, the inter-group level as well as the intra-group level. Accordingly, we base our design on the concept of a ring of rings: The nodes of each group maintain a set of DHT rings to organize themselves as well as store and retrieve measurement data and information about the network. On the inter-group level, the organizational groups take the role of nodes in global DHT rings. The global rings serve several purposes, for example identifying groups with information about a third group of interest, or the bootstrapping procedure where the system has to find a suitable group for a newly joining node.

3 Cluster Identification

We cannot expect all nodes in the network to be part of the XBAC architecture. However, we still want to be able to provide distance estimates for nodes that are not part of the XBAC overlay network. In order to do this efficiently, we require a way to cluster nodes without their direct cooperation. We have developed an approach based on comparing time series of round trip time or TCP throughput. If two nodes are close to each other their respective time tend to be similar. Distance difference functions are used to detect such commonalities in time series of RTT and TCP throughput distance measures.
The cluster identification procedure begins with a preprocessing step. The measurement data must be normalized to get well-formed time series. In the case of TCP throughput we also apply the natural logarithm to each value of the time series to obtain additive characteristics. We obtain a well-formed, additive time series a_i.

A second time series s_i is computed by calculating a moving standard deviation of the time series. In the case of TCP throughput measurements, this standard deviation is calculated relative to the low-pass filtered series using a box filter with a 1-hour window. In the case of round trip time measurements, we use a moving minimum filter, also with a 1-hour window. The choice of moving minimum filter is due to the

characteristics of round trip time on the Internet, almost constant minimum with very variable peak values.
Based on these time series we can compute the distance differences. The function

$$M(p,q) = \frac{1}{T}\sum_{i=1}^{T} a_{p,i} - a_{q,i} \qquad (2)$$

indicates cases where two time series show similar values at similar times. In the case of time series with strong variance we need to compensate using the function

$$SD(p,q) = \sqrt{\frac{1}{T}\sum_{i=1}^{T} (a_{p,i} - a_{q,i} - M(p,q))^2} \qquad (3)$$

Combined, these functions result in the first distance difference function

$$\delta_{MSDD}(p,q) = \begin{cases} |M(p,q) - SD(p,q)|, & p \neq q \\ 0, & p = q \end{cases} \qquad (4)$$

While δ_{MSDD} detects similar values at similar times, this is not enough. Two nodes may be totally unrelated but still have similar values at similar times. Only if the changes in values in both time series happen at the same time can we consider both nodes to be in the same cluster. Conversely, if both time series show common changes but do not show common values, we cannot consider the nodes to be in the same cluster. We detect synchronous changes using the distance difference function

$$\delta_{CSD}(p,q) = 1 - cor(s_p, s_q) \qquad (5)$$

We consider two nodes to belong to the same cluster only if both distance difference functions yield values close to zero for the respective time series.
Evaluation has shown that this approach is in fact able to remotely detect clusters given time series of round trip time or of TCP throughput.

4 QoS Prediction

Most distance estimation services only provide momentary or stationary values for a given node pair. Using the exchanged and stored information provided by the XBAC architecture we are able to enhance the distance estimation process by computing QoS predictions. A client application should be able to send requests like “what is the minimal value of round trip time that will not be exceeded in the next 10 minutes, with 95% confidence?” This approach will result in more robust connections, which is especially useful for video conferences and overlay networks in general.

More than short-time predictions (in the order of seconds) are difficult to make on the Internet. We developed an approach to RTT prediction based on creating a discrete and finite state space and computing a Markovian transition model to predict transitions from one class to the other. We divide a time series of round trip times into slices of 10 minutes length and create a 5-dimensional vector by computing the 0-, 25-, 50-, 75-, and 100-percentiles of the slice. See Figure 1 for illustration.

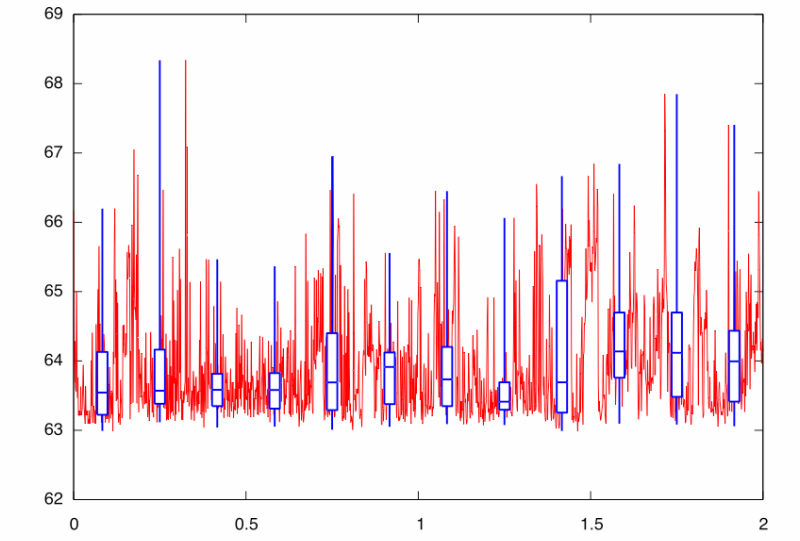


Fig. 1. A time series of round trip times, and the corresponding percentile vectors.

Given a sufficiently long time series (e.g. a week) we obtain an adequate number of vectors. Using a hierarchical clustering algorithm we identify 32 "typical" vectors for the time series. This results in a sequence of class identifiers in the range 0–31.
This sequence can then be converted into a transition matrix T, where each element t_{ij} describes the probability that the next observed class will be j provided the last class was i. This approach can easily be extended by making the next class depend on the last n classes, in which case the matrix will become an n+1-dimensional array.
Given a model as described above, the next class can be predicted as follows: Create percentile vectors based on the last $n \cdot 10$ minutes of observations and assign them to one of the 32 classes using a classification algorithm. Then, extract the probability vector $T_{i_1 \ldots i_n}$ from the array and select the class with the maximum probability. The predicted percentile vector can be further improved by computing a weighted mean percentile vector based on the respective probabilities $T_{i_1 \ldots i_n j}$ of the classes $j = 0 \ldots 31$.

5 Conclusion and Outlook

This article only gives a very short overview of the work done in the XBAC project. Many details were left out due to space restrictions. Future work will mainly concentrate on two areas: One the one hand, the overlay design presented in Section 2 will be further refined. Among other things, messages and data formats will be defined. On the other hand, the QoS prediction approach from Section 4 will be improved by adding further indicators (e.g. slope and curve of the sample) to the prediction process.

References

1 P. Francis, S. Jamin, J. Cheng, Y. Jin, D. Raz, and Y. S. IDMaps: a global internet host distance estimation service. In IEEE/ACM Transactions on Networking, 9(5):525-540, October 2001.

2 L. Garcés-Erice, W. Biersack, and P. A. Felber. MULTI+: Building topology-aware overlay multicast trees. In *5th International Workshop on Quality of Future Internet Services (QofIS'04)*, September 2004.

3 *SONAR: A network proximity service*. http://www.netlib.org/utk/projects/sonar.

4 X. Y. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu. A construction of locality-aware overlay network: mOverlay and its performance. IEEE Journal on Selected Areas in Communications, 22(1):18– 28, January 2004.

Wireless Mesh Networks

Staub Thomas

staub@iam.unibe.ch

Today, various wireless network technologies are deployed in isolated networks. In order to combine these networks and enhance the overall coverage with network services a key technology, wireless mesh networks (WMN), has appeared. Last but not least this technology makes a big step in the direction of being always-on-line anywhere anytime.

The authors of [1] provide a good overview of wireless mesh networks, state-of-the-art protocols and open research issues in this area. WMNs are wireless ad-hoc networks. They consist of two types of nodes: mesh clients and mesh routers. Both support multi-hop communication and can therefore act as routers. Additionally, a mesh router is equipped with multiple radio interfaces based on the same or different wireless access technology. The mesh routers are rather static than mobile and can build a wireless mesh backbone. They contain gateway and bridge functionalities to other networks. Mesh clients are mobile or static devices, which connects over multi-hop communication to a mesh router. They can be more sensitive in power consumption than mesh router that are static and can therefore be directly connected to the electricity network. Classification of WMNs leads to three main types. Infrastructure WMNs build a wireless backbone for conventional clients. Community and neighborhood networks can be built using this infrastructure meshing. Client WMNs offers peer-to-peer networks among client devices. They are practically the same as a MANETs. The third category is hybrid WMNs that provides a combination of the other two types and will be the most applicable.

Currently the research and development of WMNs is driven by the following application scenarios [1]: broadband home networking (cost efficient “last mile”), community and neighborhood networking, enterprise networking, metropolitan area networks, transportation systems, building automation, health and medical systems, surveillance systems, Emergency / Disaster, and P2P.

The authors of [1] identify different open issues across all communication layers. They are caused by the already shown specialties of WMNs, e.g. various power constraints, and by multiple types of network access technologies. Further, new antenna types such as smart antennas, directional antennas and MIMO technologies introduce new flexibility in communication diversity at expense of new hidden and exposed node problems. MAC protocols as well as routing protocols have to care about multi-channel or multi-radio interfaces, switching times between the radio systems, and power consumption. On the MAC layer scheme the focus is changing from capacity, throughput, and fairness to the support of QoS metrics. Routing protocols should take the differences of the nodes into account. Multi-radio and multi-channel interfaces add a new dimension to routing protocols. The path as well as the appropriate channel or radio has to be selected and the routing decision has to include channel and radio switching times, possible interferences between different channel and power consumption. This leads to a strong need of cross layer design.

The performance degradation of TCP in multi-hop wireless networks leads to new transport protocols (e.g. ATP) which interoperability problems with existing TCP/IP networks and to optimizations and variations of TCP. Open issues on the application layer are adaptation of existing Internet applications, P2P information sharing applications optimized for WMNs and new services made available through WMNs (e.g. distributed backup in neighborhood networks). Furthermore, there are several network management and security problems.

In my opinion WMNs offer a more robust and redundant communication infrastructure than the wireless networks deployed today. They supply communication possibilities even in special situations where certain systems (e.g. GSM network) are overloaded. With the integration of QoS abilities in WMNs, namely the priorization, they are a key technology for an emergency signaling system.

Context-awareness in WMNs builds another interesting topic. The individual nodes should be aware of their context. They know their position, movement, display resolution, processing power, remaining energy and have a local view of the network topology with the different available access technologies. This awareness could be used to control access technology or channel hand-over. It offers new possibilities for application content adaptation. For example (see Figure 1), the start of a video communication in a high resolution is not reasonable, if the user leaves the high bandwidth area rapidly and will have only the bandwidth for a low-resolution video stream in the next minutes. It would be better to have a non-interrupted low-resolution stream during the whole communication.

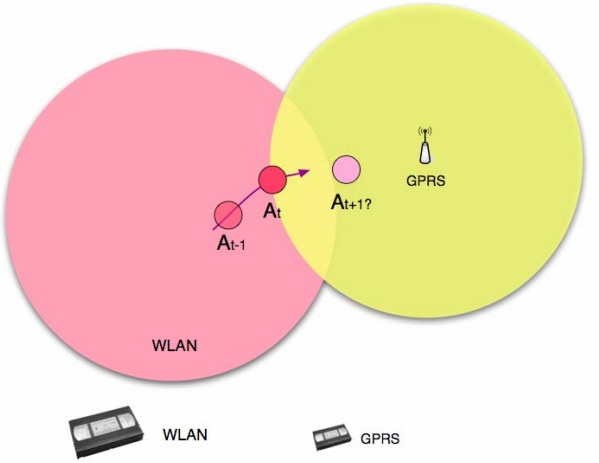


Fig. 1.

An web application can offer only text content in low bandwidth areas, in areas with little more bandwidth the content is enhanced with low resolution images and with even more bandwidth it could offer high quality images and videos (see Figure 2).

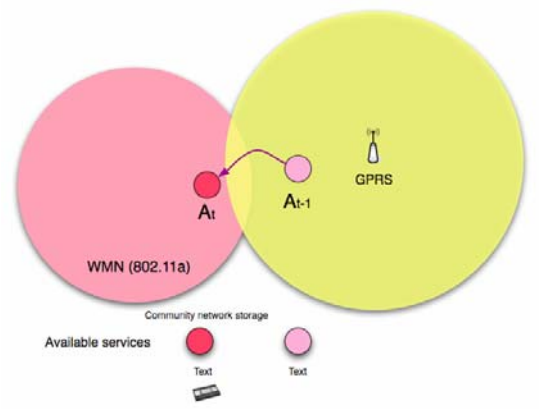


Fig. 2.

To support context-aware content conditioning, i.e. adaptation to the device's possibilities and its current situation, for existing applications like a web browser we thought of a context-aware middleware (CAM) on the client and a proxy system (see Figure 3). CAM signals the context information from the client as well as some user defined adaptation profile to the proxy which prepares the normal web content for the client and delivers a conditioned view of the content to the client device.

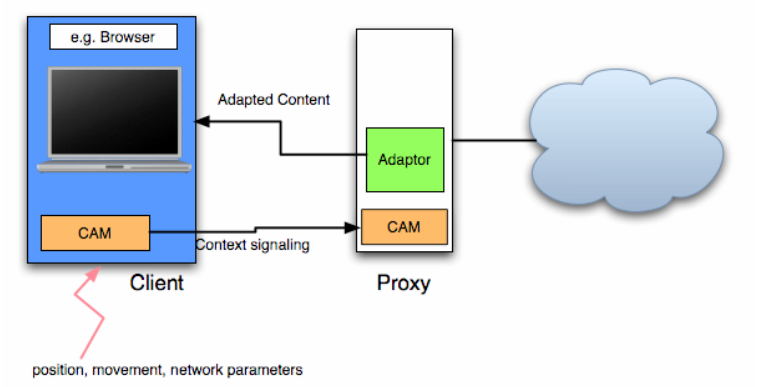


Fig. 3.

Future work consists of modeling the context information in a general way, defining the context signaling, and the middleware and the proxy.

Reference

1 I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445–487, March 2005.

Setup of Simulations on Heterogeneous Networking with CAHN

Marc Danzeisen

danzeis@iam.unibe.ch

Abstract. Communication networks become more and more heterogeneous. This has mainly two reasons: first, the convergence of IT and telecom has pushed the integration of short range communication technologies like WLAN and Bluetooth with existing cellular networks like GPRS and UMTS. Second, the high penetration of mobile devices having multiple communication interfaces integrated enable users to be always connected to the Internet or to other nodes. However, to successfully use these different communication technologies a certain level of knowledge is required, which turned out to be a hurdle for normal end users. In earlier work an architecture including a dedicated protocol has been developed to facilitate heterogeneous networking. The introduced system eases the setup of heterogeneous connections between nodes and offers hence the possibility to profit from the advantages of heterogeneous networking environments. To evaluate the gain in terms of throughput and energy saving potential a dedicated simulator has been built.

Keywords. Heterogeneous Networks, Ad-hoc, On-demand, Energy Saving, Simulations.

1 Introduction

With the help of our introduced architecture to enable cellular assisted heterogeneous networking (CAHN) nodes can be connected using always the best available link. Whenever communicating nodes come close enough to each other, they switch to direct links or ad-hoc communication. Due to the fact, that short range communication technologies like WLAN, Bluetooth or UltraWideBand (UWB) are offering much more bandwidth at a much lower power level, the seamless integration of these short range technologies is noticeably increasing the average throughput and decreasing the energy consumption of data sessions. The integration of infrastructure based access networks with the ability to switch to direct short range communication whenever possible is not only promising efficiency increase in terms of battery lifetime and session duration, but also in terms of network resource management. Operators can save network capacity by supporting the direct link between nodes. Due to the seamless handover to infrastructure based networks in case of loss of the direct link, the user experience is not affected. Especially, when considering the trends towards flat rate based billing for mobile data services, this network resource savings become economically interesting for operators.

2 Simulation Setup

To better understand the impact of CAHN and quantify the potential benefit of our architecture we decided to do some simulations. A thorough evaluation of existing network simulation tools showed that there is no simulator available for heterogeneous networks. All of the existing simulators (academic and commercial) do not provide support for the simulation of heterogeneous networks with dynamic vertical handovers during runtime, end-to-end communication between nodes using different wireless technologies simultaneously, and switching between infrastructure and ad-hoc mode of operation. Furthermore, these simulators either do not yet implement certain wireless technologies, e.g., GPRS in Qualnet, or implement different technologies for different incompatible versions, e.g., UMTS for ns-2.26 and GPRS for ns-2b7a. Therefore, we implemented our own network simulator that allows the modeling of heterogeneous networks at a simplified level. The simulator does not account for any physical propagation or MAC layer functionality and simulates sessions between peer mobile nodes at the application layer, i.e., no packet transmission are simulated. We are mainly interested to estimate the potential benefit of enabling seamless switching between infrastructure and ad-hoc based links like illustrated in Figure 1:

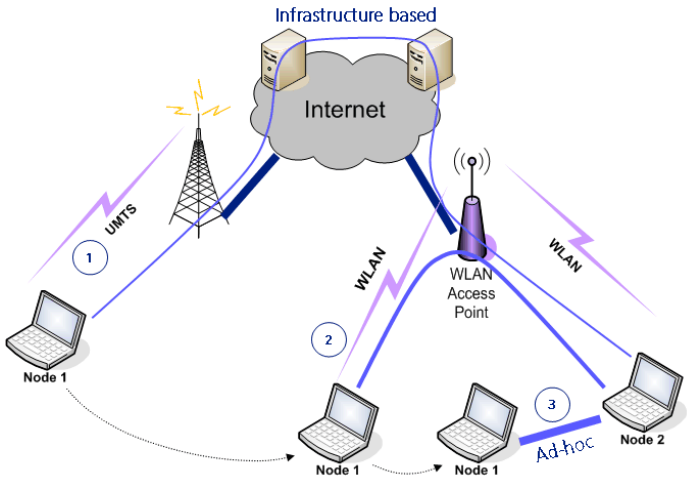


Fig. 1. Seamless Switching between Infrastructure and Ad-hoc Mode

The second feature of interest is the ability of CAHN to signal incoming session requests using the low power cellular signaling system. Power demanding broadband communication interfaces can therefore be kept in sleep mode if no data session is ongoing, without loosing the advantage of being always reachable. This feature is referred to as On-demand broadband connectivity and illustrated in Figure 2.

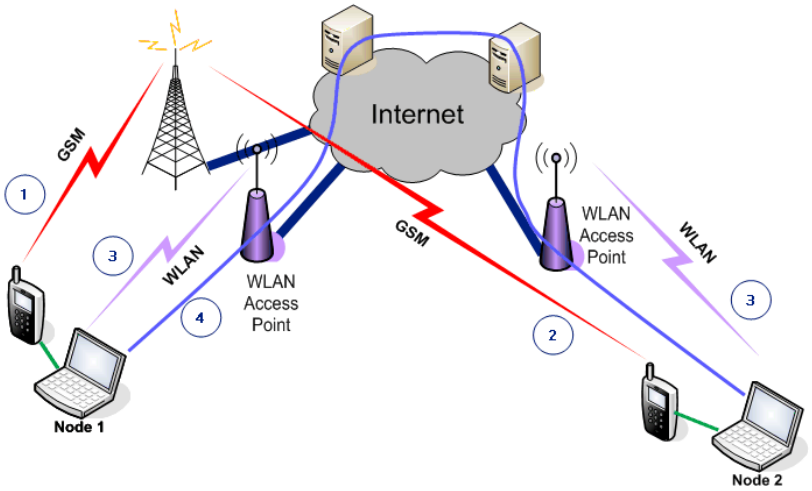


Fig. 2. On demand establishment of end-to-end communication sessions

We plan to conduct the simulations in three different test series. The first one will vary the simulation area. We will analyze four different area sizes, namely 1km x 1km, 2km x 2km, 5km x 5km and 10km x 10km. The second will address the impact of varying the session density. Therefore, we will simulate between 100 and 1000 data sessions between the nodes. With the third simulation the impact of the different technologies will be evaluated by changing the coverage areas of the different technologies, namely GPRS, UMTS and WLAN.

3 Preliminary Results

First simulation runs showed that the average throughput can be increased by up to a factor of 4 and the energy consumption reduced about 80% in certain scenarios. To confirm these preliminary results we will runs further simulations and add new scenarios by varying the data rates offered by the different networking technologies.

References

1 M. Inoue et. al., “Novel out-of-band signaling for seamless interworking between heterogeneous networks,” IEEE Wireless Communications Magazine, vol. 11, no. 2, pp. 56–63, april 2004.
2 (2005) 802.21. Media Independent Handover Interoperability. [Online]. Available: http://www.ieee802.org/21/
3 M. Danzeisen et. al., “On the benefits of heterogeneous networking and how cellular mobile operators can help,” in Proceedings of IEEE WSNET, 2005.

4 M. Danzeisen et. al., “Heterogeneous networking establishment assisted by cellular operators,” in Proceedings of MWCN, 2003.
5 T. Camp et. al., “A survey of mobility models for ad hoc network research,” Wireless Communications and Mobile Computing (WCMC), vol. 2, no. 5, pp. 483–502, 2002.
6 L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in INFOCOM 2001, Anchorage, AK, USA, Apr. 2001, pp. 1548–1557.
7 (2005) Merlin U630 Wireless PC Card Modem. Lucent Technologies. [Online]. Available: http://www.lucent.com/livelink/

Cooperation in Multi-hop Wireless Networks

Attila Weyland

weyland@iam.unibe.ch

Abstract. This short paper is separated into three sections. In the first section, we give an overview of existing approaches which effectuate cooperation in multi-hop wireless networks. In the second section, we propose improvements to our CASHnet algorithm and show some preliminarily simulation results. In the last section, we conclude and give an outlook about future work.

1 Cooperation Schemes in Multi-Hop Wireless Networks

Cooperation among nodes is vital in multi-hop networks. Without nodes forwarding other nodes packets, communication over multiple hops is impossible and the nodes remain unconnected. Thus, a constant contribution from all participants of a multi-hop network is necessary to keep the nodes connected and thereby the network operational. Figure 1 and 2 illustrate the problem.

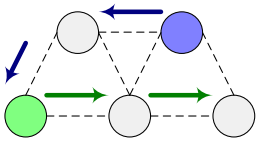


Fig. 5. A and D require C and B's cooperation to reach their communication partners respectively.

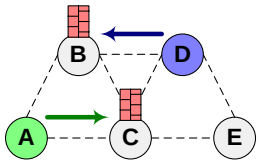


Fig. 6. Without cooperation only single-hop connections are possible and the network falls apart.

Considering the military origin of multi-hop networks, cooperation among nodes is not an issue in the corresponding application scenarios. This is true for all scenarios, where nodes are under control of a single authority and the multi-hop network is established for the purpose of the application. Example scenarios include military operations and disaster recovery.
In scenarios without single authority, cooperation among nodes is not obvious. When each user of a node is her own authority, she can decide by herself what to do. This individual freedom of each user leads to selfishness. Helping other users by forwarding their packets results in the consumption of the own node's limited resources, such as processing and transmission time as well as battery power. Regarding the resource consumption, a node's owner is better off when being uncooperative, because he can save the resources for her own transmissions. When applying this attitude to all nodes in a multi-hop network, no forwarding takes place

and communication over multiple hops becomes impossible. And although a common goal in connectivity among the nodes might exist, the necessity of cooperation to achieve that goal is difficult to comprehend by individual users. Especially, when the communication partner is located outside the current multi-hop cellular network, e.g. in the Internet, the benefit of helping neighbors is not apparent.
Therefore, the cooperation in non-single authority application scenarios must be effectuated by additional measures. The challenge of achieving cooperation in multi-hop networks lies in the management of cross-layer information flows and the coordination of actions on different layers. Cooperation clearly requires cross-layer protocol design and is tightly connected to security.

1.2 Approaches to Cooperation

Cooperation in multi-hop networks can be looked at from two sides, the network and the user/node perspective. From the network perspective, the nodes have to cooperate because they act as the backbone infrastructure. If they do not cooperate, the network ceases to exist. Thus, any uncooperative node harms the network and poses a threat to the network's correct functioning. Often, an uncooperative node is considered as a security threat. The consequence is that cooperation must be enforced by all possible means. From the user perspective, cooperation is costly, because it consumes resources such as processing and transmission time as well as battery power. It is not obvious for a user, to allow her node to forward other users' packets. To make up for this loss in resources caused by cooperation, some kind of reward is distributed. Thus, cooperation must be encouraged by giving an incentive to the user.

1.2.1 Enforcement

In the cooperation enforcement schemes, uncooperative nodes get punished so severely, that they have no choice but to cooperate. Figures 3 - 5 show the typical operation of an enforcement scheme. The underlying assumption is that all nodes are always able to cooperate. So, uncooperativeness is just a sign of bad behavior and must be corrected using appropriate measures.

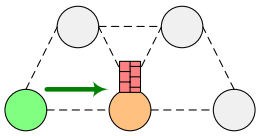


Fig. 7. C drops A's packets destined for E

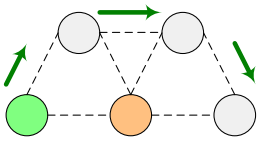


Fig. 8. A propagates C's behavior to B, D and E

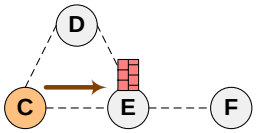


Fig. 9. E does not forward C's packets destined for F

However, this assumption ignores situations, where a node may not be able to cooperate after all, even if it wants to. This includes nodes running on very low battery power, nodes located at border areas with few packets to forward or nodes with a full buffer. Another problem arises in the determination of the cooperativeness of a node. In enforcement approaches it is common to perform some kind of neighborhood watch, which means each node is monitored and evaluated by its

neighbors. Therefore, the enforcement approaches are also called *detection-based* schemes. The surveillance results are then used to optimize the operation of the network.

1.2.2 Encouragement

Encouraging cooperation in multi-hop networks is based on the assumption that nodes may be reluctant or unable to cooperate. Reasons for uncooperative behavior include the avoidance of additional costs imposed on a user/node or the inability caused by the state of the node or the network, e.g. congestion. To make up for the additional costs of cooperation, the user should be compensated. This compensation should be high enough to overcome the user's reluctance and make cooperation attractive. Also, in case of the inability to cooperate nodes do not get punished. Figure 6 and 7 depict the operation of an exemplary encouragement scheme.

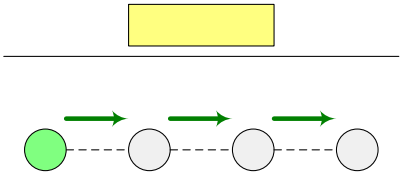


Fig. 10. O transmits a packet to R and each forwarding node keeps a receipt for the transmitted packet.

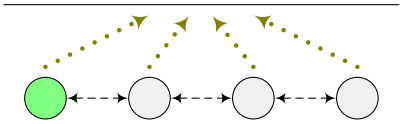


Fig. 11. O, A, B and R periodically transmit their receipt to the accounting center.

Due to the usage of incentives to encourage cooperation, an additional valuable good is introduced into the architecture. Therefore, the encouragement approaches are also called *motivation-based* schemes. Besides the connectivity, the chosen incentives must be protected from misuse. This requires security measures beyond trust relations.

2. Improvements to CASHnet

CASHnet [1] is our cooperation and accounting strategy for hybrid networks. It encourages cooperation by giving rewards to cooperative nodes and applying charges to the transmission of own packets. CASHnet works as follows: Every time a node (customer) wants to transmit a self-generated packet, it has to pay with *Traffic Credits*. The amount is related to the distance in hop counts to the gateway. Every time a node forwards a packet, it gets *Helper Credits*. Traffic Credits can be bought for real money or traded for Helper Credits at Service Stations. A Service Station is similar to a low-cost terminal for loading prepaid cards and has a secure, low-bandwidth connection to the provider, which is used for authentication and payment operations. Figure 8 illustrates a typical scenario for CASHnet.

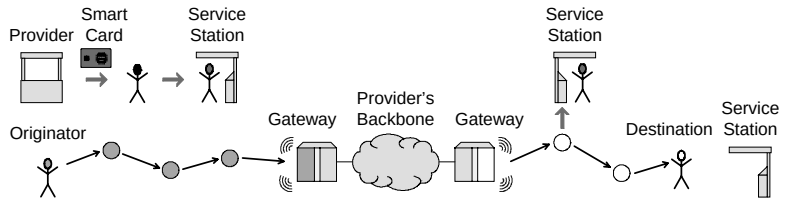


Fig. 12. CASHnet example scenario, where a customer obtains a Smart Card and initializes it at the Service Station to participate in the network.

We presented our initial simulation results in [2] and a comparison with the Nuglet [3] scheme in [4]. The results showed that although CASHnet outperformed Nuglet with two or more Service Stations, the performance was not satisfactory. We analyzed the simulation scenarios and found the node density to be too low and increased it accordingly. We also optimized the per-hop per-packet rewarding in our scheme, by introducing a packet counter threshold to each node. Thus, not every data packet gets acknowledged, but every nth. Figure 9 and 10 show the results for the old and the improved CASHnet scheme. We vary the packet counter threshold between 1, 5 and 10.

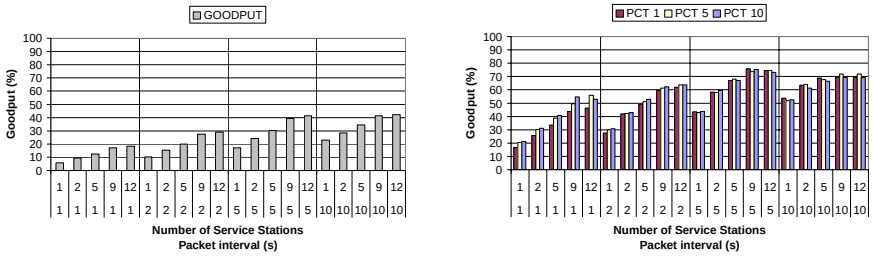


Fig. 13. Goodput of the old and the improved CASHnet scheme for different packet counter thresholds, PCT.

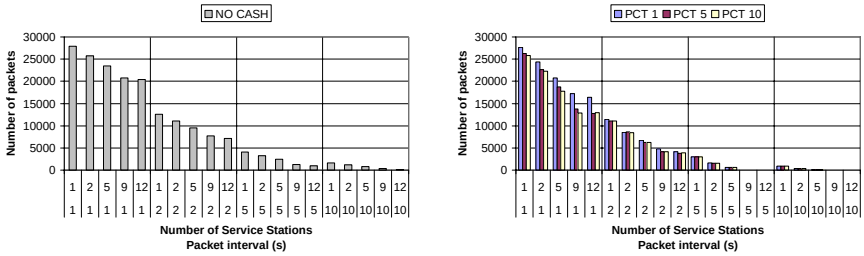


Fig. 14. Number of dropped packets due lack of Traffic Credits of the old and the improved CASHnet scheme for different packet counter thresholds, PCT.

With an increasing packet counter threshold, the value of the acknowledgement increases and so does the probability of the recipient of an acknowledgement to become unreachable. We see this effect in Figure 9, which contains the goodput. We also find the efficiency of the packet counter threshold to greater under high network load. This can be also seen in Figure 10, where the highest improvements are achieved in scenarios with a high number of service stations.

3 Summary and Outlook

In this short paper we gave an overview and a description of the two different approaches to cooperation. We also presented some improvements to our CASHnet scheme and showed promising, preliminary results.
We are currently implementing a prototype of the CASHnet architecture under Linux. We also analyze further improvements, such as reseller nodes and mobile service stations, via simulations.

References

1 A. Weyland, and T. Braun. Cooperation and Accounting Strategy for Multi-hop Cellular Networks. In *Proceedings of 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2004)*, Mill Valley, CA, USA, April 2004.
2 A. Weyland, T. Staub, and T. Braun. Liveliness Evaluation of a Cooperation and Accounting Strategy in Hybrid Networks. In *Proceedings of 4th Workshop on Applications and Services in Wireless Networks (ASWN 2004)*, Boston, MA, USA, August 2004.
3 A. Weyland, T. Staub, and T. Braun. Comparison of Incentive-based Cooperation Strategies for Hybrid Networks. In *Proceedings of 3rd International Conference on Wired/Wireless Internet Communications (WWIC 2005)*, Xanthi, Greece, May 2005.
4 L. Butty´an and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *ACM Mobile Networks & Applications*, 8(5), October 2003.

TCP Support for Sensor Nodes

Torsten Braun

braun@iam.unibe.ch

1 Introduction

Wireless sensor networks are composed of a large number of radio-equipped sensor devices that autonomously form networks through which sensor data is transported. Typically, devices are severely resource-constrained in terms of energy, processing power, memory, and communication bandwidth. Many wireless sensor network applications require an external connection to monitoring and controlling entities (sinks) that consume sensor data and interact with the sensor devices. Running TCP/IP in the sensor network allows connecting the sensor network directly to IP-based network infrastructures without proxies or middle-boxes. TCP/IP should be used for administrative tasks that require reliability and compatibility with existing application protocols. Examples of such tasks are configuration and monitoring of individual sensor nodes as well as download of binary code and scripts to sensor nodes.

2 Challenges with TCP/IP in Wireless Sensor Networks

Recent work showed that TCP/IP can be implemented on sensor nodes with limited processing power and memory [1]. On the other hand, TCP/IP may result in relatively large headers that may add significant overhead in case of short payload. However, TCP may be mainly used for configuration and programming tasks, where a rather high amount of data is transferred and payload is rather large. A TCP/IP header compression scheme for sensor networks can overcome this problem. Due to the stateful approach proposed below such a compression scheme should be feasible. Other problems to be solved for TCP/IP in sensor networks are related to addressing. While in traditional IP networks IP addresses are assigned to each network interface based on the network topology, IP-based sensor networks may use spatial IP address assignment based on node locations, which might be relative to a base station location [2]. While in traditional IP networks, packets are transparently routed through the network based on the network topology, data centric routing mechanisms are often preferable in wireless sensor networks [3]. To implement data centric routing in IP-based sensor networks, application overlay networks might be used. It is also well known that TCP has serious performance problems in wireless networks [4]. One problem is that TCP, which has been designed for wired networks with low bit error rates, interprets packet loss as an indication of congestion and decreases its transmission rate in case of a lost packet. This results in low throughput. The main

problem for sensor networks operating autonomously with constrained power supply is the energy-inefficiency of TCP. This is caused by TCP's end-to-end retransmission scheme requiring that lost packets are retransmitted by the original sender of the packet. In a multi-hop network, retransmitted packets must be forwarded by all intermediate nodes from sender to receiver, thus consuming valuable energy at every hop. In general, end-to-end error recovery is not a good approach for reliable transport in sensor networks, because the per-hop packet loss rate may be in the range of 5% to 10% or even higher [5].

3 TCP Support for Sensor Nodes

TCP Support for Sensor nodes (TSS) aims to support energy-efficient sensor network operation and forms a layer between TCP and the routing layer in a communication protocol stack of sensor nodes. TSS should ideally be implemented in TCP sensor nodes with senders and receivers as well as in intermediate sensor nodes that relay TCP (data) segments and acknowledgements of a TCP connection. TSS tries to reduce the number of transmissions by several mechanisms discussed below. The TSS mechanisms do not require explicit link or MAC level acknowledgements, but TCP segments and acknowledgements are the only packets that are needed, if nodes can overhear the neighbour's transmissions. Performance evaluations of the discussed mechanisms showed significant performance improvements in terms of the number of transmitted data packets as well as throughput [6].

Caching

An intermediate node caches a segment until it is sure that the successor node towards the destination has received the segment. A node knows this when it detects that the successor node has forwarded the segment (implicit acknowledgement) or when it spoofs a TCP acknowledgement that has been sent from the destination toward the source of the TCP segment. Nodes are assumed to listen to packet transmissions of their neighbour nodes in order to be able to detect whether the neighbour nodes have forwarded TCP segments. A packet that is known to be received by the successor node will be removed from the cache. In addition to the cache, TSS requires another packet buffer for temporarily storing the next packet that is waiting to be forwarded to the successor node. One might argue that forcing sensor nodes to overhear packets does not support energy efficient operation. However, typically, a packet will be forwarded immediately by the successor node and only in case of packet loss a node must overhear for the whole retransmission timeout interval. An alternative would be explicit link level acknowledgements. This would not only require the node to listen and receive but also the successor node to transmit an additional acknowledgement packet. Explicit acknowledgements will therefore be more expensive than overhearing, which introduces an overhead lower than 20 %.

Local Retransmissions of TCP Segments

All intermediate nodes are able to perform local retransmissions, when they assume that a cached segment has not been received by the successor node towards the destination. Retransmissions are triggered by timeouts, which requires intelligent setting of timeout values. The retransmission timeout is set to 1.5 * rtt and allows repairing multiple packet losses before an end-to-end retransmission timeout is triggered. It might happen that a node's retransmission timeout expires, if it has received an overheard packet header with an error and dropped that implicit acknowledgement. Then, the node retransmits a TCP data segment although that one has already been received and forwarded by the successor node. However, the already forwarded TCP segment should not be forwarded again. Forwarding can be prevented by a small history list consisting of the last few forwarded packets to filter out all segments that have been forwarded previously. Retransmitted TCP segments can be uniquely identified by the source address and the IP identification field. End-to-end retransmissions should not be filtered in order to support end-to-end recovery in serious error situations.

Regeneration and Recovery of TCP Acknowledgements

TCP acknowledgements are extremely important for TSS, since several mechanisms such as round-trip-time estimation, retransmission, and caching depend on it. Experiments have shown that loss of acknowledgements may have a severe impact on the amount of TCP segment transmissions. TSS deploys two mechanisms for retransmissions of TCP acknowledgements that help to decrease the number of TCP segment transmissions significantly. First, the local acknowledgement regeneration mechanism becomes active when a node receives a TCP data segment, which has already been acknowledged by the destination. In that case, the TCP segment is dropped and a TCP acknowledgement with the highest acknowledgement number is regenerated and transmitted toward the source. Second, an aggressive recovery mechanism recovers TCP acknowledgements, if a node has not discovered the forwarding of the TCP acknowledgements by the successor node. Since TCP acknowledgements should usually be forwarded without significant delay towards the sender of TCP segments, each node measures the time between its own TCP acknowledgement transmission to the successor node and the overhearing of the TCP acknowledgement transmission from the successor node towards the TCP segment sender (source). Similar as for the rtt estimation we use exponential averaging. We set the TCP acknowledgement retransmission timeout to the double average value. After timeout expiration, a TCP acknowledgement is recovered using the highest acknowledgement number.

Backpressure Mechanism

If the successor of a node has not forwarded all received packets, the network might be congested or packet forwarding does not make progress, because a previous TCP

segment with bit error needs to be recovered first. If a node would continue with packet forwarding in these cases, the risk of unnecessary transmissions would be rather high. In a congestion situation, a forwarded segment might easily get lost then. The same is true in case of a lost packet due to bit errors. In such a situation all caches on subsequent nodes are occupied and the transmission of a new packet would not be protected by caching. For that reason, a TSS node stops any forwarding of subsequent packets until it knows that all earlier packets have been received and forwarded by its successor. Successful forwarding can be detected by overhearing the forwarded packet or by detecting a TCP acknowledgment for that TCP segment. If packet forwarding stops at some point, all other nodes in the chain behind the stopping node will also stop their transmissions until progress is detected at their respective successor nodes. In case of a lost packet (due to congestion or bit errors) packet loss should be recovered by the node that forwarded the packet at last. In that case, we have to avoid that retransmissions are triggered by nodes closer to the sender. This can be achieved by increasing the retransmission timeouts at the nodes closer to the sender. This backpressure mechanism should also be implemented at the sender end point. We propose to not increase the TCP congestion window as long as there are a few packets, e.g., three, waiting at the sender for transmission. This mechanism is quite effective and keeps the number of packets in flight within the limits as proposed by [7].

4 References

1 A. Dunkels: Full TCP/IP for 8-bit Architectures, ACM MobiSys, pp. 85-98, San Francisco, May 2003

2 A. Dunkels, T. Voigt, J. Alonso: Making TCP/IP Viable for Wireless Sensor Networks, Work in Progress Session at 1st European Workshop on Wireless Sensor Networks (EWSN 2004), Berlin, January 2004

3 D. Estrin, R. Govidan, J. Heidemann and S. Kumar. Next century challenges: scalable coordination in sensor networks, Mobile Computing and Networking, pp. 263-270, 1999

4 H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving TCP/IP performance over wireless networks. ACM MobiCom, pp. 2-11, November 1995

5 C.-Y. Wan, A. Campbell, L. Krishnamurthy: PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks, 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, September 28, 2002

6 T. Braun, T. Voigt, and A. Dunkels: Energy-Efficient TCP Operation in Wireless Sensor Networks, Praxis der Informationsverarbeitung und Kommunikation (PIK), special issue on Wireless Sensor Networks, No. 2, 2005

7 Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, M. Gerla: The Impact of Multihop Wireless Channel on TCP Throughput and Loss, IEEE Infocom, San Francisco, March 30 - April 3, 2003

Distributed Event Detection in Wireless Sensor Networks

Markus Waelchli

waelchli@iam.unibe.ch

Abstract. Distributed event detection and data aggregation are inherent tasks of many wireless sensor network applications. The inaccuracy of the sensed data and the difficulty to determine event properties such as location, area diameter, etc. make event detection and data aggregation challenging problems. In current approaches, uncertainty is only barely considered and the detection and localization of events has not yet been done in a fully distributed manner. In our work we aim to fill these gaps and provide appropriate techniques.

Motivation

The main objective of the proposal is the design and implementation of a distributed event detection, event localization and data aggregation framework. As uncertainty is an intrinsic property of event detection, event localization, and data aggregation appropriate techniques like fuzzy logic or probability theory could be useful. Furthermore, fuzzy logic seems to be a promising approach due to its computational efficiency, its simple description language, and its tolerance towards measurement inaccuracies. We think that including sensor and event locations into a fuzzy logic system enables accurate event localization. In addition, we aim to provide estimated coordinates in order to support the localization of events in a fully distributed manner. Consequently, our approach enables for example not only the detection of a fire, but also the localization of the origin of that fire. Moreover, the use of fuzzy logic is not limited to event detection, but is also appropriate for data aggregation and data filtering in order to support efficient event processing, reduce network traffic, and identify unusual events, e.g. false alarms. Many existing systems do not take these issues into account. A possible application of fuzzy rule bases could be the detection and avoidance of outliers, e.g. the detection of a significant temperature rise at a specific sensor due to some malfunction of the device. Such outliers could thus be avoided in the aggregation process and would not tamper the aggregated result. The projected architecture aims to increase the significance of distributed event detection, event localization, and data aggregation by employing a novel distributed inference method. Furthermore, an intuitive and easy to use framework for sensor network monitoring and observing shall be provided.

Related work

Our projected approach falls in the topics of event detection and localization, middleware approaches, and data aggregation for sensor networks. The scientific community has already contributed some work to these areas. The authors of [1] propose a sniper detection system where sensors estimate the position of an event with the help of muzzle blasts and acoustic shockwaves measured by TDOA mechanisms. These measurements are then delivered to a base station where the position of the sniper is calculated by searching the maximum of a four dimensional consistency function that is derived from the gathered data measurements. In Sextant [2], positions of nodes and events are estimated from connectivity constraints that are gathered from the physical network layer. Positions of nodes are described with Bézier polygons, whereas the location of an event is estimated as a probability distribution over an area derived from the relevant Bézier polygons. Landmarks are needed in order to estimate the positions of the sensors. The authors of [3] propose a sensor deployment and event localization framework. After deployment, the sensors build local clusters where the cluster head elects a subset of the cluster participants to provide detailed information about an event. This can be done, as after an event occurrence all sensors send a small message, indicating that they sensed an event, to their cluster head. The authors of [4] propose a distributed deviation, or outlier, detection by investigating real-time streaming data. The approach supports powerful cluster heads that have a large communication range and are able to calculate powerful operations. TinyDB [5, 6] is a distributed database approach that is enhanced with stochastic techniques to support spatio-temporal data gathering. Unlike our approach TinyDB focuses more on periodic query processing and event localization is not considered. GADT [7] supports probabilistic ADTs in order compensate the deviation of sensor measurements. GADT is provided to avoid the inaccuracy of sensed data. The work of [8] proposes an ad-hoc group formation algorithm based on quorum techniques to gather data from predefined areas. The localization of events is not supported by this approach. DSWare [9] enables applications to specify a compound event in order to collect relevant information from a certain geographical area. DSWare focuses only on event detection, the localization and the area diameter have to be provided by the application. With iSpheres [10], unanticipated situations are described as normal conditions with an anomaly as a significant deviation from normality. iSpheres is not tailored for sensor networks with its own requirements and limitations.

Key ideas for a distributed event detection and localization framework

Event detection and localization are intrinsic features of wireless sensor networks. Some work in this context has already be done by the scientific community. A common feature of these approaches [1, 2, 3] is their dependence on a central instance, e.g. a sink node with more computational power, where the measurements from the sensors in the field are collected and the event localization is computed.

Other approaches [7, 8] are mainly concerned in enabling and establishing group communication and data aggregation in a predefined area that has to be observed. In contrast to these approaches, we intend to provide a fully distributed event detection framework that avoids the drawbacks of increased data traffic between the sensed area and the base station. Some initial concepts and ideas are outlined in this section.
We suggest that the detection of an event at a node is observed as a set of deviated values simultaneously sensed by that node (e.g. increased temperature, significant shockwave, etc.). Furthermore, such an event is only significantly observable in a restricted region and the event is decreasingly observable the farther away a node is. We propose that, considering these requirements, all nodes in the relevant region can derive the significance with which they sensed a certain event. Moreover, this significance can be inferred as the barycenter of the set of deviated values sensed by the node. Thereby, each sensed value can be weighted and therefore satisfies a certain membership function, e.g. 80ºC could have a membership degree of 0.8 in relation to the predicate 'hot'. A key idea of our approach is to use fuzzy logic mechanisms to classify the deviated values on the one hand and to infer the significance of the event from these values on the other. The significance of an event is thereby represented as a value in the basic interval [0, 1].
Using these derived values we propose to use a distributed election algorithm that determines the relevant subset of sensors which are responsible for handling the event further, e.g. sending their information to a base station, or aggregation their information among each other. In Figure 1, two possible approaches are outlined.

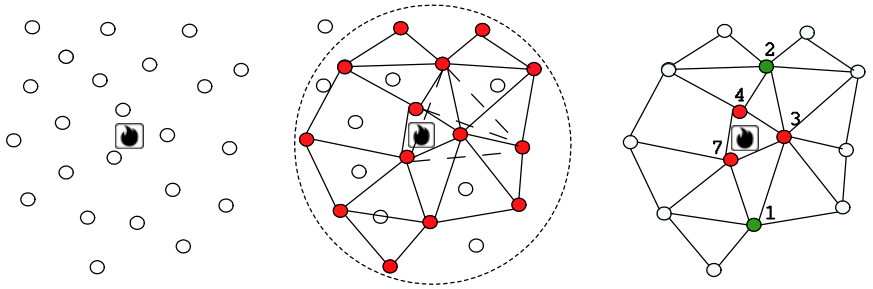


Figure 1a: DDB-based election scheme

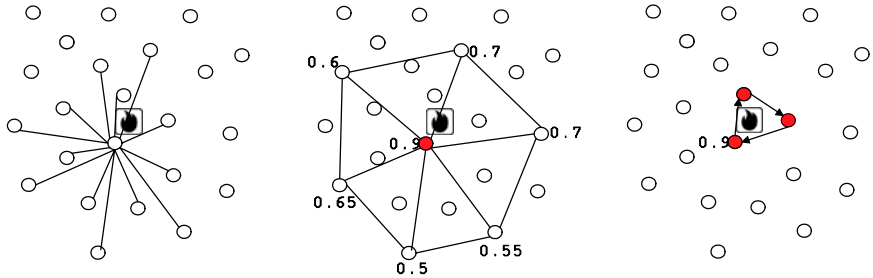


Figure 1b: Planar Graph based election scheme

In subfigure 1a, the first approach is presented. In a first step, a subset of nodes is determined. This can be done with an adapted version of the dynamic delayed

broadcasting algorithm (DDB) proposed by [11]. In order to enable the election of all the sensors surrounding the event, the DDB algorithm has to be adapted in the way that sensors with a high significance value broadcast their value anyway and thus are ensured to take part in the election round. When this step is complete, the nodes closest to the significance border (farthest away from the event) start with the election of their most significant neighbors. This procedure is then repeated until the sensors closest to the event are reached (the red ones in subfigure 1.a). The node closest to the event is not able to chose a better positioned node and thus determines itself as 'winner' and is responsible to further process the event detection. An advantage of this scheme is that position information is not needed.
In subfigure 1b, the second approach is presented. Initially all sensors in the relevant region broadcast their significance and their position to their neighbors. With this information each node computes its local planar graph and decides if it is the most relevant node or not. In subfigure 1b, again the red node with a significance value of 0.9 will determine itself as winner. In this approach it is possible that more than one node determines itself as being the winner. Therefore, some communication among the winners has to take place. The winner could for example surround the event on its border (see subfigure 1b). An advantage of this approach is that it is tolerant against wrong measurements. If for example a node has two opposite neighbors both with higher values than itself, then there occurred either two events within the relevance radius of this node or one of both neighbors has a wrong value. This can be derived from the estimated event locations computed from the data gathered in the vicinity. The significance of an node is thereby interpreted as an approximation of its distance to the event. Consequently, an event may be located in a two-dimensional field by the positions of three neighbors and their respective significances.

References

1 G. Simon, G. Balogh, G. Bap, M. Maróti, B. Kusy, J. Sallai, Á. Lédeczi, A. Nádas and K. Frampton, Sensor Network-Based Countersniper System, *SenSys'04*, Baltimore, Maryland, USA, November 2004

2 S. Guha, R.N. Murty and E.G. Sirer, Sextant: A Unified Node and Event Localization Framework Using Non-Convex Constraints, *MobiHoc'05*, pp. 205-216, Urbana-Champaign, Illinois, USA, May 2005

3 Y. Zou and K. Chakrabarty, Sensor Deployment and Target Localization in Distributed Sensor Networks, *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 3, Nr. 1, pp. 61-91, February 2004

4 T. Palpanas, D. Papadopoulos, V. Kalogeraki and D. Gunopulos, Distributed Deviation Detection in Sensor Networks, *SIGMOD Record*, Vol. 32, No. 4, pp. 77-82, December 2003

5 S.R. Madden, M.J. Franklin, J.M. Hellerstein and W. Jong, The Design of an Acquisitional Query Processor for Sensor Networks, *SIGMOD'03,* San Diego, California, June 2003

6 A. Deshpande, C. Guestrin, S.R. Madden, J.M. Hellerstein and W. Hong, Model-Driven Data Acquisition in Sensor Networks, In *Proceedings of the 30th VLDB conference,* Toronto, Canada, 2004

7 A. Faradjian and J. Gehrke and P. Bonnet, Gadt: A probability space adt for representing and querying the physical world, In *International Conference on Data Engineering (ICDE)*, pp. 201-212, 2002

8 M. Kumar, L. Schwiebert and M. Brockmeyer, Efficient data aggregation middleware for wireless sensor networks, In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 1579-1581, Fort Lauderdale, Florida, USA, October 2004
9 S. Li, S. H. Son and J. A. Stankovic, Event Detection Services Using Data Service Middleware in Distributed Sensor Networks, *IPSN'03*, pp. 502-517, Palo Alto, USA, April 2003
10 E. Albek, E. Bax, G. Billock, K.M. Chandy and I. Swett, An Event Processing Language (EPL) for Building Sense and Response Applications, In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, Denver, Colorado, April 2005
11 M. Heissenbüttel, T. Braun, M. Wälchli and T. Bernoulli, Broadcasting in Wireless Multihop Networks with the Dynamic Forwarding Delay Concept, *Technical Report, IAM-04-010*, University of Bern, Switzerland, December 2004

Positioning in Wireless Sensor Networks

Thomas Bernoulli

bernoull@iam.unibe.ch

Wireless sensor networks consist of devices with at least one sensing facility for a physical phenomenon. Depending on the application, the sensed data is collected, processed, and aggregated within the network before it is sent to at least one base station. Base stations are the gateways between the sensor network and other networks transporting the data to the user.
Data reported to the users is normally attributed with the geographical position where it has been sensed. Taking this information into account users are not only briefed that an event has taken place in the covered area, but also where it happened.
To attribute sensed data with the geographical position of the triggering event is a difficult task. It consists of two main units:
All nodes must be aware of their position
If the event has been sensed by more than one node, the information has to be merged to estimate the event's position (this can be done within the network or at the base station)

The second item is not wider discussed in this chapter but is subject of the chapter 'Distributed Event Detection in Wireless Sensor Networks' from Markus Wälchli. In the context of wireless sensor networks assuming a node is aware of its position is quite a strong assumption. The well known GPS system can not be used for senor networks, as the sensor nodes should save battery power and GPS consumes a lot of energy (and is not cheap anyway). In such networks it is only feasible to assume that a very small fraction of the nodes have GPS. The other nodes must figure out their position by some algorithms which take for example neighborhood of the GPS equipped nodes into account.
This chapter will only consider mechanisms that do not need more than some nodes being equipped with GPS. Infrastructure based mechanisms or those which need some special equipped nodes (e.g. [1]) seem to be unfeasible for wireless sensor networks. The remaining mechanisms can be divided into two groups. The first group covers algorithms which only take topology information into account. Algorithms based on some additional measurement values, e.g. distance estimations to neighbors (from signal strength or with additional hardware), form the second group of mechanisms. What both groups have in common is that at least 3 nodes (called landmarks) must be aware of their geographical position before the algorithm starts.
The first representative of the first group is called DV-Hop [2]. As the name already states it is based on a Distance-Vector mechanism and hop counting. It consists of three relatively simple steps:

- Every landmarks floods the network with a packet containing its position. The packet is flooded using a distance vector mechanism and every node in the network stores its distance (in hops) to the landmark.

- A landmark receiving such a packet calculates the average distance a packet travels by one hop and floods this information.
- Every node receiving a hop distance packet can estimate its position based on this distance and the collected distances to landmarks using multiliteration.

This algorithm is easy to implement and provides good positioning of the networks has a regular shape and the distances between nodes are not varying to much. As soon as the distances are varying a lot and the network is spread over an area of irregular shape (e.g. in the shape of a 'C'), the algorithm does not provide accurate positioning.
The second representative of the first group is called MDS-MAP [3]. It is based on Multi-Dimensional-Scaling. Out of all distances between the nodes it calculates a relative map using MDS. Using some landmarks this map is normalized and transformed. Although the algorithm is described in its centralized operation mode, it can easily be distributed as it is shown, that calculating the relative map out of all distances among nodes in a 2-hop neighborhood leads to quite accurate maps if the networks is dense enough. 3-hop neighborhood is really enough, even for sparse networks.
Different mechanisms of the second group exist [2], but they all have similar weaknesses: The distance or angle estimations to neighbors have to achieve a certain level accuracy. Good news first: If the accuracy is high enough, this mechanisms can outperform the topology based ones. Bad news follow: To achieve this level of accuracy additional hardware is required, something that is objectionable for sensor nodes.
When it comes to evaluation of positioning mechanisms, this is normally done by simulation. Typical scenarios consist of a random network in the 2D-plane. If some distance or angle measurements are needed, their error is modeled as a Gaussian distributed variable with some reasonable boundaries. This leads to open questions and future work that could be done:

- When talking about sensor networks, a common use case is deployment of sensors in buildings. Positioning of sensor nodes in buildings has to be done in the three dimensional space. Adaptation, evaluation, and improvement of positioning algorithms for 3D opens a wide field for future work.
- Up to now evaluations model no errors (topology based mechanisms) or model them Gaussian distributed. But signal propagation in real world probably leads to systematic and biased errors. Especially if we do not only consider scenarios in the plain field but also those where sensor nodes are deployed in areas with obstacles like cities, forests, and many more. Future work could try to model these influences more accurately. Because such work would lead to more complex simulators, future research does not only go in the direction of better physical models of signal propagation in 3D environments, but also to distributed simulation.

The issue of positioning of nodes with limited resources is a interesting topic. Future work could try to get a increased match between the evaluation scenarios and the real world use cases. Two possibilities to reach that goal are the step from 2D to 3D and the improvement of the used simulator's physical model.

References

1 T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks", *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003.

2 D. Niculescu, and B. Nath, "DV Based Positioning in Ad Hoc Networks", *Telecommunication Systems*, 2003.

3 Y. Shang, H. Shi, and A. A. Ahmed, "Performance study of localization methods for ad-hoc sensor networks", *Proceedings of the 1st International Conference on Mobile Ad-hoc and Sensor System*, 2004.

Background Information to EU Projects

Marc-Alain Steinemann

steine@iam.unibe.ch

Abstract. This presentation describes the networks, backgrounds and procedures, necessary for writing an EU proposal. The presentation is mainly based on McCarthy's Proposal Writing course.

1 Introduction

EU projects are mostly huge projects with many partners from many countries. EU projects help to establish new contacts und make it possible to perform research across the borders. For those that would like to lead an EU project, time consuming preparations are required. These preliminary tasks are described in a brief report, which is mainly based on Sean McCarthy's [1] proposal writing course.
To the time this report is written, the sixth EU framework program is running, which is replaced by the seventh program from 2007 on. Each framework is devoted to a main topic and consists of different work parts. In these parts, one to several calls are published. Figure 1 presents the structure of a framework.

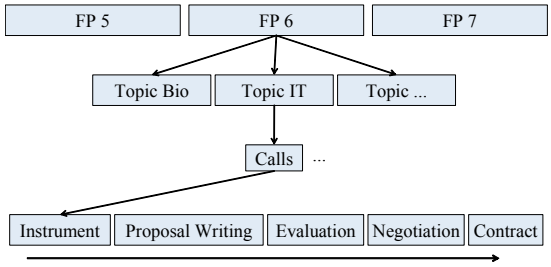


Fig.1. Structure of a frame work

2 History of Origins of a Call

The EU has principally two procedures to reach their political goals: EU and national policies and support programs, presented in Fig.2.

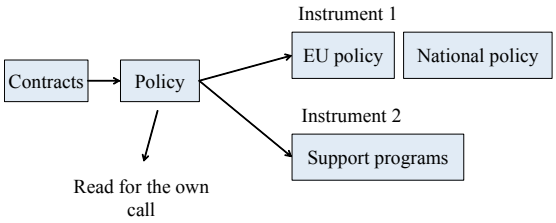


Fig. 2. Activities in the EU

The origins of the support programs lie in the politics. The background papers are listed on europa.eu.int. The EU commission maps the politics into support programs and makes the calls. Thus, it is necessary to start lobbying early. Figure 3 shows the history of a call.

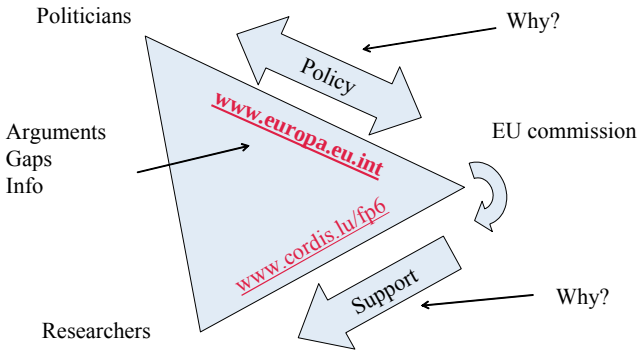
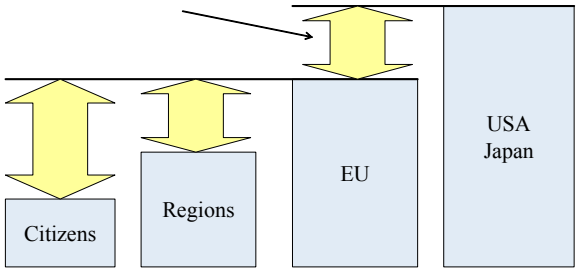


Fig. 3. History of a call

The goal of a support program is, to close existing gaps within the habitants of the EU, the regions of the EU and the EU and other countries, like presented in Figure 4.



Proposals must fill gaps. Identify gaps!

Fig. 4. Proposal must fill gaps

3 Preparations

Before we can start writing a proposal, time consuming preparations are necessary. For a better understanding of the Brussels politics, it is recommended to actively work on green and white papers or at least to know them very well.
At least nine months before the call, a partner network should be constituted in the form of a consortium and the project vision be fixed.
Partners should meet based on existing contacts. If not possible, partner DB can be used. IT is recommended to join Advisory Boards and Networks of Excellence. Additionally, there exist European Research Associations, topic networks and COST. Members of EU Assessments/Monitoring/Evaluation panels can be contacted and preferably visited personally.
The active and passive participation at EU conferences and seminars as well as contact to authors from EU studies and reports is very recommended.
The partner network should consist of the best of the best European scientists and also include SME, which can develop results to prototypes.
Three months before the call, the ideas should be evaluated by external experts from research and politics.
To the time the call is published, the project should register at Cordis as the evaluators are assigned in the order the projects are registered.

4 Writing of the Proposal

Additionally to the selection of the correct instrument, which fits to the own project if a good lobby work has been done before, the proposal must resolve a from Brussels recognized problem. It is necessary to educate the evaluator with pictures tables and facts with references to official documents.
The documentation to the proposals has to be respected in any case. Different but similar questions have to be replied based in their history of origins. It is recommended to read to guide for evaluators and to write in bold text the required points.
It is important to build new projects upon existing ones and to show the benefit of the results. The projects should exceed other powers in the world in some way.
The current version of the proposal can always be loaded on Cordis. It is important to exactly respect the delivery time.
For the writing of the proposal you should calculate with six months. A team normally consists of:

- A scientific coordinator, responsible for Part B and scientific plan.
- Scientists, who provide the content of the work packages of Part B.
- A consortium manager, who integrates the single parts.
- A financial administrator, who writes Part A and the budget.
- A lawyer, who treats with the consortium contracts.
- A result utilization manager.

References

1 Sean McCarthy, www.hyperion.ie