RVS Retreat 2007 at Quarten

Torsten Braun, Ulrich Ultes-Nitsche, Marc Brogle, Dragan Milic, Patrick Lauer, Thomas Staub, Gerald Wagenknecht, Markus Anwander, Markus Waelchli, Markus Wulff, Carolin Latze, Michael Hayoz, Christoph Ehret, Thierry Nicola

IAM-07-004

December 2007

RVS Retreat 2007 at Quarten

Torsten Braun, Ulrich Ultes-Nitsche, Marc Brogle, Dragan Milic, Patrick Lauer, Thomas Staub, Gerald Wagenknecht, Markus Anwander, Markus Waelchli, Markus Wulff, Carolin Latze, Michael Hayoz, Christoph Ehret, Thierry Nicola

Abstract

The research group "Computer Networks and Distributed System" of the Institute of Computer Science and Applied Mathematics at the University of Bern, headed by Prof. Torsten Braun, organized an internal retreat from July 2-4, 2007 at Quarten. The reserach group "Telecommunications, Networks, Security" of the Department of Computer Science at the University of Fribourg, headed by Prof. Ulrich Ultes-Nitsche also participated in the event. The focus of this retreat was to present and discuss recent research results and currently ongoing research activities of the members of both research groups. The research group members gave fourteen presentations, form the areas of overlay networks, wireless mesh and sensor networks, network security, distributed systems and automata theory. Extensive time (typically 60 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 This technical report summarizes the various talks and describes mostly unpublished work that is currently in progress.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: C.2.1 [Computer-Communication Networks]: Network Architecture and Design; C.2.2 [Computer-Communication Networks]: Network Protocols; C.2.3 [Computer-Communication Networks]: Network Operations; C.2.4 [Computer-Communication Networks]: Distributed Systems

General Terms: Design, Management, Measurement, Performance, Reliability, Security, Verification

Additional Key Words: peer-to-peer, wireless mesh networks, wireless sensor networks, overlay multicast, network security, automata theory

Contents

Overlay and Peer-to-Peer Networks

OM-QoS: Overlay Multicast Quality of Service

Marc Brogle, University of Bern
brogle@iam.unibe.ch

Introduction

Peer-to-Peer (P2P) [1, 2] networks have recently become very popular. Application Level Multicast (ALM) building on-top of P2P mechanisms is a possible solution to overcome the limited Internet-wide availability of IP Multicast. End-users can use ALM on-top of P2P and overlay networks to efficiently disseminate data using the multicast paradigm. Unfortunately there is no standard for ALM and also no general QoS schemes for ALMs exist. To support QoS for ALMs, the multicast trees should be built in such a manner that all paths from the root to the leafs have monotonically decreasing QoS requirements or capabilities. This is presented in Fig. 1. It is important to note that the QoS classes used need to have a natural order. This can be for example the bandwidth, the jitter, or any combination of QoS parameters, as long as they can be naturally ordered. Delay would not work because it is an additive QoS metric. OM-QoS as presented in [3] aims to be a general framework or ruleset that enables QoS for different P2P/ALM schemes by ensuring the construction of the QoS capable multicast trees as explained above.

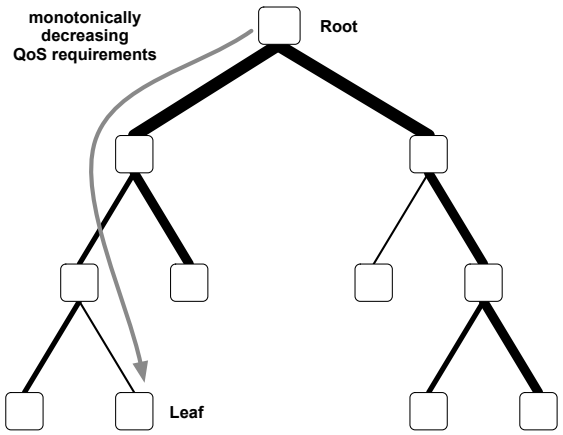

Figure 1: QoS capable multicast tree

Investigated P2P/ALM networks

The following ALM/P2P networks and combinations have been investigated so far: Scribe/Pastry, Bayeux/Tapestry, NICE, CAN (Content Addressable Networks), and Compass Routing. They are all presented in Fig. 2.

Pastry [4] is a P2P network using Plaxton's routing mechanism [5]. The P2P network can be visualized as a ring. The peers select a randomly chosen ID. A host in the P2P network has a limited view and only knows about a few other peers. To route a message to another peer, the host selects a peer from its routing table as next hop, which matches at least one more digits of the destination's ID. If no peer "lives" at the destination address, the peer numerically closest to the destination is responsible for handling the message. Scribe [6, 7] runs on top of Pastry and uses its routing mechanism to build the multicast trees. Each hop on the way from

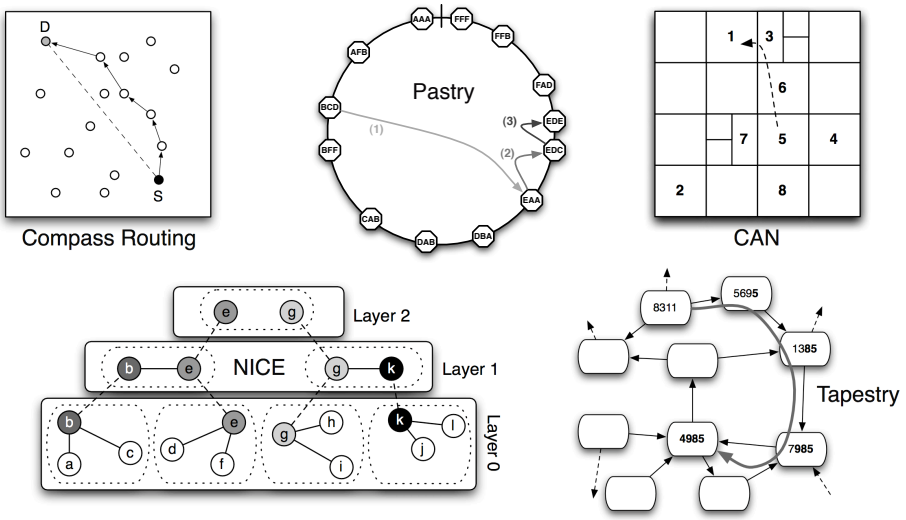

Figure 2: Investigated P2P/ALM networks

the joining host to the destination host (which is the root of the multicast tree) remembers the previous hop. To disseminate the data the reverse path from a subscriber to the root will be used.

Tapestry [8] is similar to Pastry. The prefix matching is though done right-to-left instead of left-to-right. Bayeux [9] provides multicast functionality for Tapestry. A host joining a multicast group sends a JOIN message to the root node using Tapestry's routing mechanism. The route node sends then a TREE message to the joining host using Tapestry. The path of the TREE message will also be path used to disseminate multicast data, which does not correspond to the reverse-path as it is the case with Scribe/Pastry.

NICE [10] is a cluster based P2P/ALM system. Hosts joining the P2P network are clustered together according to their relative proximity in terms of round-trip-time (RTT) delay. The clusters are limited by size and each cluster has a cluster head. The cluster head is determined by comparing all RTTs among the cluster members. The host having the lowest and the highest value of the RTTs is selected as the cluster head for this cluster. All clusters are bundled together in a layer. Now all cluster heads from that layer are forming a new layer, in which again clusters will be built and new cluster heads will be determined. This is continued until there is just one layer with one cluster at the top left. Data is multicasted by the cluster heads, which are forwarding the data to all members of their different clusters. This could lead to a high fan-out for a cluster-head, which would be selected in all or most of the layers.

CAN (Content Addressable Networks) [11, 12] uses a virtual space, which is divided among the participating peers. Each joining hosts selects some random coordinates in that space. It then finds the host responsible for that partition of the space, in which the selected coordinates lay. The responsible host for that partition of the space then has to divide its space equally and has to assign one half as responsibility to the new host. To send data to another host, the neighbor closest to the destination's area is selected as the next hop. Multicasting is done by sending data to all neighbors using a duplicate suppressing mechanism to reduce redundancy.

Compass Routing with Delaunay Triangulations as presented in [13] also uses a virtual n-dimensional space. Hosts have knowledge about their immediate sur-

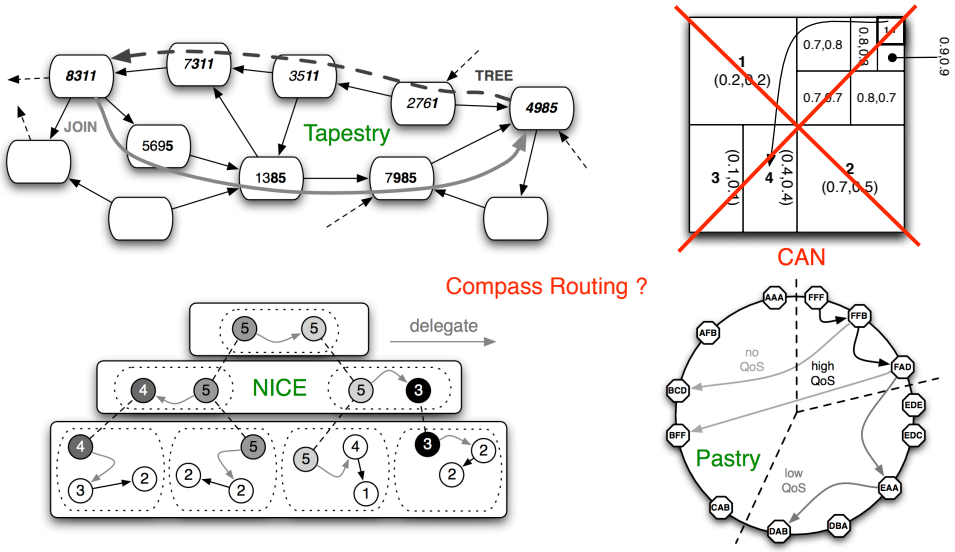

Figure 3: P2P/ALM QoS enabling protocol dependent approach

roundings but don't have a full view of the P2P network. To send a message to another host, the forwarder is selected among its neighbors, which has the lowest angle towards the destination's coordinates compared to the direct path. Multicasting can be done by using reverse-paths and duplicate avoiding mechanisms.

Introducing QoS to the investigated P2P/ALM networks

Introducing QoS to the different ALM/P2P networks to build QoS capable multicast trees as described in the introduction (see also Fig. 1) can be achieved as presented in Fig. 3 for Pastry, Tapestry and NICE.

Scribe/Pastry can be extended with QoS by changing the ID assignment method. Instead of randomly assigning an ID when entering the P2P network, the hosts assign themselves IDs depending on their QoS requirements. The higher the QoS requirements are, the higher the ID is. The root itself will have the highest ID, and for each multicast group we will have a dedicated Pastry network, where only peers join that are interested in the multicast data. This leads to a partitioning of the ID space as shown in Fig. 3. When hosts join to the multicast group (FFF...), they will always pass through a node with higher or same QoS requirements. The reverse paths (which is used to build the multicast tree) therefore automatically generate a QoS capable multicast tree.

Tapestry/Bayeux can use the a slightly modified ID assignment mechanism as for Scribe/Pastry. For higher QoS requirements of a peer, more digits of the ID (from right to left) have to match the root's ID.

NICE can be made QoS enabled by changing the way the cluster head is determined, which is shown in Fig. 3. Instead of using the delay as a metric to determine the cluster head, the QoS requirements can be used. The cluster member with the highest QoS requirements will be the cluster head. To reduce the fan-out of the cluster heads, a delegate is introduced (a member of the same cluster with same or next lower QoS requirements), which will then on behalf of the cluster head disseminate the data in its cluster.

CAN on the other hand cannot be made QoS enabled by using the same approaches as described before. If the QoS requirements would determine the initial coordinates (the higher the QoS the higher the coordinates) and the root would

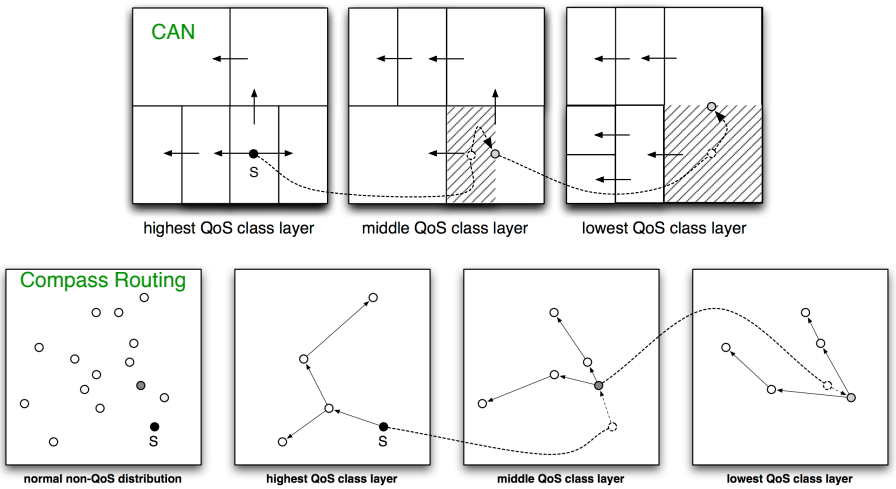

Figure 4: Layered QoS enabling approach

get the highest coordinates, then due to the quite rigid movement of the responsible zones during the joins and leaves, paths from the root to any node would not necessarily lead to QoS capable multicast trees as described before. In Fig. 3 an example is shown, which would have a path that does not follow the principle of monotonically decreasing QoS requirements. The number denotes the order the peers joined the P2P network, while the values in brackets denote the initial coordinates, which depend on the QoS requirements (high QoS requirement results in high coordinates).

The same problem occurs with Compass Routing. There is no evident way how to make this P2P scheme QoS enabled by modifying the routing mechanism or ID / coordinate assignment method according to the QoS requirements of a host.

To enable QoS for CAN and Compass Routing, a different approach has to be chosen, which is presented in Fig. 4. The layered approach basically works as follows: (1) each QoS class has one dedicated layer or P2P network assigned; (2) the sender is responsible to disseminate data (using the ALM specific method) in its own layer as well as to send it to one peer in the next higher and next lower QoS layer; (3) a peer receiving data from another layer has to disseminate it in its own layer as well as to send it to the next layer in the same direction (up or down the QoS hierarchy). Using these three mechanisms we can again ensure that the multicast trees are QoS capable and that all paths from the root to leafs have monotonically decreasing QoS requirements.

The layered approach seems to be a more general solution to ensure QoS capable multicast trees. Further investigation is required to see if it can be applied to all so far analyzed P2P/ALM protocols. One of the challenges of this approach is how to efficiently find the right forwarder in the next layer for the different P2P/ALM protocols.

Conclusion and Outlook

All the ALM/P2P schemes that have been investigated so far can be enhanced with QoS functionalities by using the presented OM-QoS mechanisms. The layered OM-QoS approach as used with Compass Routing and CAN seems to be a general solution to enable QoS for P2P/ALM networks. This approach has to be further investigated with the other ALM/P2P schemes, for which OM-QoS so far can offer only protocol specific solutions to enable QoS.

Bibliography

[1] S. A. Theotokis and D. Spinellisa, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys (CSUR)*, vol. 36, pp. 335–371, December 2004.

[2] K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *Communications Surveys & Tutorials, IEEE*, pp. 72–93, 2005.

[3] M. Brogle, D. Milic, and T. Braun, "QoS enabled multicast for structured P2P networks," in *Workshop on Peer-to-Peer Multicasting at the 4th IEEE Consumer Communications and Networking Conference*, IEEE, January 2007.

[4] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, (London, UK), pp. 329–350, Springer-Verlag, 2001.

[5] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environ ment," in *SPAA '97: Proceedings of the ninth annual ACM symposium on Paralle l algorithms and architectures*, (New York, NY, USA), pp. 311–320, ACM Press, 1997.

[6] A. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel, "Scribe: The design of a large-scale event notification infrastructure," pp. 30+, 2001.

[7] M. Castro, P. Druschel, A. M. Kermarrec, and A. I. T. Rowstron, "Scribe: a large-scale and decentralized application-level multicast infrastructure," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, pp. 1489–1499, 2002.

[8] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and," tech. rep., 2001.

[9] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz, "Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination," in *NOSSDAV '01: Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video*, (New York, NY, USA), pp. 11–20, ACM Press, 2001.

[10] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 32, (New York, NY, USA), pp. 205–217, ACM Press, October 2002.

[11] S. Ratnasamy, M. Handley, R. M. Karp, and S. Shenker, "Application-level multicast using content-addressable networks," in *NGC '01: Proceedings of the Third International COST264 Workshop on Networked Group Communication*, (London, UK), pp. 14–29, Springer-Verlag, 2001.

[12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, vol. 31, pp. 161–172, ACM Press, October 2001.

[13] J. Liebeherr, M. Nahas, and W. Si, "Application-layer multicasting with delaunay triangulation overlays," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, pp. 1472–1488, 2002.

Optimizing Dimensionality and Accelerating Landmark Positioning for Coordinates Based RTT Predictions

Dragan Milic, University of Bern
milic@iam.unibe.ch

Introduction

It has beeen shown that RTT is one of the limiting factors for the effective bandwidth of a TCP connection [1]. With the dramatic increase of bandwidth available to the end users, the RTT becomes the limiting factor of effective bandwidth available trough a TCP connection. Thus, taking RTT into account for constructing a peer-to-peer (P2P) network or choosing a FTP service mirror leads to signifficant performance gains in terms of available bandwidth and minimizing overall use of network resources.

Measuring RTTs between all possible communication partners, and storing them is unfortinately not practicable, since both number of measurements needed and the storage requirement grow quadratically ($O(n^2)$) with the number (n) of potential communication partners.

Overview

There are numerous proposals, how this scalability isse can be addressed. One of the pioneering works on this field is IDMaps [2]. Authors of IDMaps propose using special infrastructure nodes, so called "tracers". All tracers periodically measure and store RTT information to other tracers. As a RTT prediction between two hosts in the Internet, a simple sum of distances between the hosts and the tracers nearest to them and the distance between the corresponding tracers. The main issue of IDMaps approach is that there is a need for infrastructure nodes (tracers) strategically positioned in the Internet. Consequenlty, practical use of IDMaps approach is quite limited, since a deployment of such an infrastructure in the Internet is highly unlikely.

Another, more promising, approach has been proposed by Ng. and Zhang in the scheme they named Global Network Positioning (GNP) [3]. In their work, the authors propose embedding RTT information as distances in an n-dimensional metric space. To overcome the scalability issue, the authors propose that the complete RTT information is measured only between few designated end-systems called "landmarks". The complete RTT information measured between the landmarks is then used to compute the positions of landmarks in the virtual space. All other hosts, determine their position in the virtual space, by measuring RTTs to the landmarks and performing multilateration relative to landmarks positions. In such a system, RTT prediction is the distance in the virtual space between their corresponding positions in the virtual space. The evaluation performed by the authors of GNP, shows that although only partial RTT information is used, the achieved precision of RTT predictions is more than satisfactory. Although GNP has numerous advantages, it also has its drawbacks. The most severe drawback of GNP is its lack of scalability: each host in the system measures RTTs to landmarks, which eventually leads to overload of the network infrastructure in the networks, where the landmarks are located. Another issue is the use of function minimization and the computational overhead introduced by it, to determine both host and landmarks positions in the virtual space. Also, the set of landmarks is fixed, meaning that

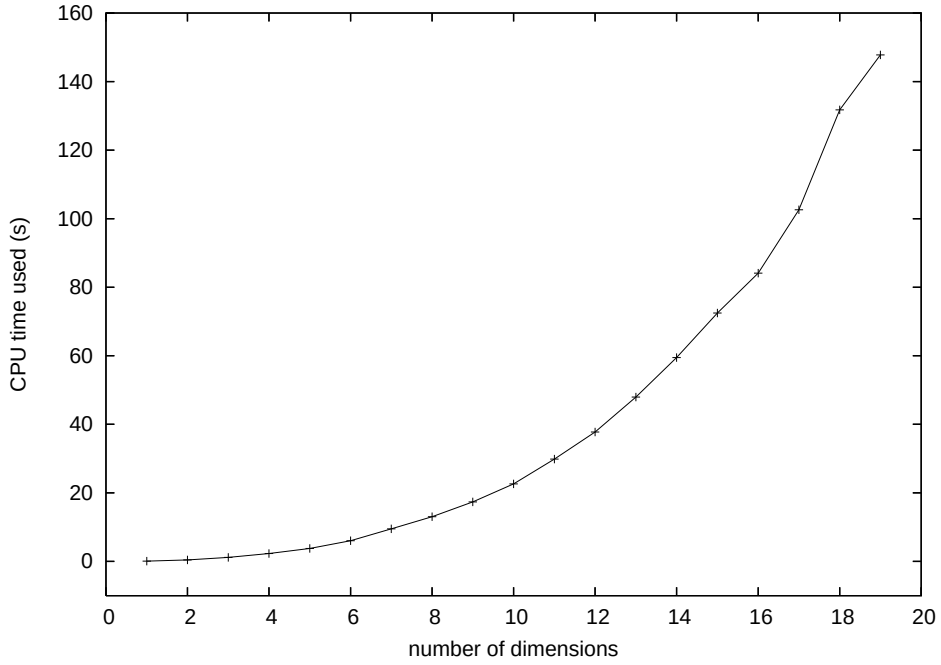

Figure 5: CPU time for minimizing objective function used to position landmarks

landmark must always be reachable and providing the service, which makes GNP not practicable in a P2P network.

Computational Overhead of GNP

Although quite simple and ellegant, GNP can still be optimized. For example, calculating positions of the landmarks tends to be a very computation intensive. The factors with largest impact on the computational overhead of landmark and hosts positioning in GNP are the number of landmarks, number of dimensions of the virtual space and the choice of the starting point for the function minimization.

In our work, we have focused on finding the optimal number of dimensions of the virtual space and a good starting point for the function minimization. As an illustration of the dependency of the computational overhead on the number of dimensions we have performed the following experiment. For "distances" we took the complete RTT informations of 20 randomly chosen PlanetLab Sites obtained from PlanetLab all-sites-ping experiment [4]. Fig. 5 shows the CPU time on a 3GHz Pentium D processor needed to perform the function minimization for positioning landmarks, depending the number of dimension of the virtual space. Further research showed that the computational overhead grows cubically ($O(d^3)$) with the number of dimensions d. This shows us, that finding the "optimal" number of dimensions can bring signifficant saving of computational resources.

Estimating Optimal Number of Dimensions

To estimate the optimal number of dimensions, we propose examining the RTT measurements matrix, by determining the simplex with maximal number of dimensions that can be constructed from the given distances. The constraint, which prevents us from embedding any matrix of $d+1$ distances into an d-dimensional space is the

triangle inequaltiy or in general, the d-simplex inequality. One way to detect if the d-simplex inequality is violated, is to calculate the Cayley-Menger determinant [5] of the distance matrix. The value of the Calyley-Menger determinant determines the square volume of a parallelipiped with given distances as edges. If this value is positive, one should be able to reconstruct a simplex with given edges.

We propose using Cayley-Menger determinant as a base for an algorithm, which starts with single landmarks (0-simplices) and tries adding more landmarks to each of them, until the maximal simplex is found.

Reconstructing the Simplex

To find the starting point of the function minimization, we propose to reconstruct the vertices of the maximal simplex, and then position all other landmarks relative to them. Reconstructing the maximal simplex is equivalent to finding one solution for the following equation system:

$$\sum_{i=1}^{m-1} (\mathcal{C}^i_{\mathcal{L}_j} - \mathcal{C}^i_{\mathcal{L}_k})^2 = (\hat{d}_{\mathcal{L}_j\mathcal{L}_k})^2 \tag{1}$$
$$\text{where } j \in \{1, \ldots, k\} \text{ and } k \in \{1, \ldots, m-1\}$$

There is an infinite number of possible solutions for landmark positioning (i.e. obtained trough rotation or translation of one solution), which means that there is an infinite number of solutions for this equation system. To obtain an equation system, which has a finite number of equation, we have to apply constraints to (1), which eliminate translations and rotations. This can be achieved by restraining the first landmark to the origin of the metric space, allowing second landmark to be only on the first axis etc.

Transforming (1) considering theese constraints yields the following equation system:

$$\begin{aligned}
(\mathcal{C}^1_{\mathcal{L}_2})^2 &= (\hat{d}_{\mathcal{L}_1\mathcal{L}_2})^2 \\
(\mathcal{C}^1_{\mathcal{L}_3})^2 + (\mathcal{C}^2_{\mathcal{L}_3})^2 &= (\hat{d}_{\mathcal{L}_1\mathcal{L}_3})^2 \\
&\vdots \\
\sum_{i=1}^{m-1} (\mathcal{C}^i_{\mathcal{L}_m})^2 &= (\hat{d}_{\mathcal{L}_1\mathcal{L}_m})^2 \\
(\mathcal{C}^1_{\mathcal{L}_2} - \mathcal{C}^1_{\mathcal{L}_3})^2 + (\mathcal{C}^2_{\mathcal{L}_3})^2 &= (\hat{d}_{\mathcal{L}_2\mathcal{L}_3})^2 \\
(\mathcal{C}^1_{\mathcal{L}_2} - \mathcal{C}^1_{\mathcal{L}_4})^2 + (\mathcal{C}^2_{\mathcal{L}_4})^2 + (\mathcal{C}^3_{\mathcal{L}_4})^2 &= (\hat{d}_{\mathcal{L}_2\mathcal{L}_4})^2 \\
&\vdots \\
\sum_{i=1}^{m-1} (\mathcal{C}^i_{\mathcal{L}_{m-1}} - \mathcal{C}^i_{\mathcal{L}_m})^2 &= (\hat{d}_{\mathcal{L}_{m-1}\mathcal{L}_m})^2
\end{aligned}$$

After using standard techniques for solving such an equation system (variable substitution) we obtain the following (recursive) solution for computing coordinates of

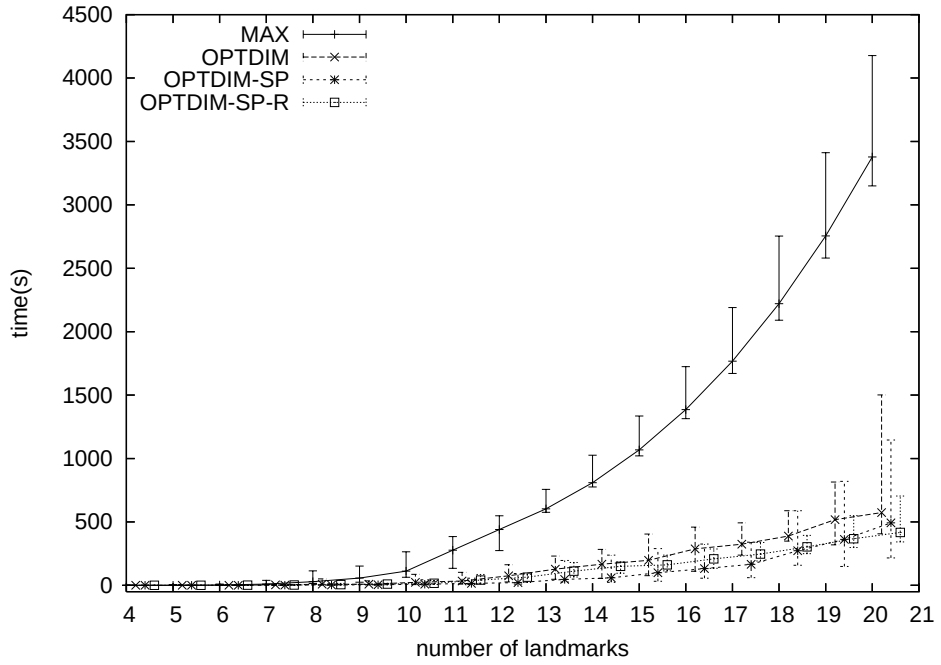

Figure 6: CPU time comparison

the landmarks:

$$\mathcal{C}^j_{\mathcal{L}_i} = \begin{cases} 0 & , j \geq i \\ (2\mathcal{C}^j_{\mathcal{L}_{j+1}})^{-1}\big[(\hat{d}_{\mathcal{L}_1\mathcal{L}_i})^2 - \\ -(\hat{d}_{\mathcal{L}_{j+1}\mathcal{L}_i})^2 - \\ -\sum_{k=1}^{j-1} 2(\mathcal{C}^k_{\mathcal{L}_{j+1}})(\mathcal{C}^k_{\mathcal{L}_i}) + & , j < i-1 \\ +\sum_{k=1}^{j} (\mathcal{C}^k_{\mathcal{L}_{j+1}})^2\big] \\ \pm\sqrt{(\hat{d}_{\mathcal{L}_1\mathcal{L}_i})^2 - \sum_{k=1}^{j-1} (\mathcal{C}^k_{\mathcal{L}_i})^2} & , j = i-1 \end{cases} \tag{2}$$

Evaluation

To compare the gains in the computation time, we have compared the CPU time needed by our algorithm for finding optimal number of dimensions and the starting point(OPTDIM-SP), with plain GNP using the number of dimensions calculated using our algorithm (OPTDIM) and the GNP using the theoretical maximal number of dimensions (MAX). As shown in the Fig. 6, our proposed algorithm signifficantly reduces the time needed to find the position of the landmarks. We also performed an statistical test, to determine if the coordinates determined by our algorithm are worse choice (have larger square error) than ones determined by GNP. All our tests indicate that there is no signifficant difference in the square error of embedding obtained using our algorithm and GNP.

Conclusion

Our work shows that it is possible to have an "educated guess" about the optimal number dimensions d for embedding landmarks in a Euclidean space. We also showed that it is possible to find a good starting point for the function minimization. Based on this, we have defined two algorithms. The first algorithm is able to find the optimal number of dimensions d for embedding landmarks in a Euclidean space and all sets of landmarks that can be used as a base for that embedding. The second algorithm finds a good starting point for the function minimization used in GNP to find the landmark positions. The results of our evaluation show that both algorithms we are proposing, are correct and able to accelerate the computation of the landmark positions of GNP.

Bibliography

[1] S. Floyd and K. Fall, "Router mechanisms to support end-to-end congestion control," tech. rep., Lawrence Berkeley National Laboratory, 1997.

[2] 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 Transactions on Networking*, vol. 9, pp. 525–540, October 2001.

[3] T. S. E. Ng and H. Zhang, "Predicting internet network distance with coordiantes-based approaches," in *IEEE Infocom02*, (New York / USA), June 23-27 2002.

[4] C. Yoshikawa, "Planetlab all-sites-pings experiment, available online: http://ping.ececs.uc.edu/ping/," 2006.

[5] D. M. Y. Sommerville, *An Introduction to the Geometry of n Dimensions*, p. 124. New York: Dover Publications, 1958.

Exploiting P2P Networks

Patrick Lauer, University of Bern
lauer@iam.unibe.ch

Introduction

As P2P networks gain in popularity they become a more interesting target for attacks. There are many types of attacks conceivable, ranging from small tweaks that improve performance for some users at the cost of all others to simple denial of service attacks. To be able to defend the potential attackers need to be characterized and their attack vectors need to be understood. Anonymizing networks like Tor offer new functions relating to privacy and anonymity, but at the same time these features enable different types of attacks. Two specific attacks are described and some possible countermeasures shown.

Classification of Attackers

Knowing the attackers capabilities is the first step for planning and building defenses. Defending against the nosy neighbor will have a different impact on design and infrastructure than trying to defend against the institutional hackers

We offer a rough hierarchical classification of attackers, from weak to strong:

- Random Hacker Guy: this attacker cannot do much damage if the data does not go through a machine he controls.
- Local Admin: this attacker can only block or spoof (local) connections and unencrypted data. As soon as encryption is used it is beyond the reach of this attacker, he can only block all encrypted traffic then.
- ISP: with more resources at its disposal this attacker can do much more subtle manipulation, including spoofing large amounts of P2P clients. Still encryption and smart routing protocols can limit the effect of this attacker
- Institutional Hackers: With dedicated resources and legal backing these hackers are very powerful. With wiretapping and physical extraction of hardware they can completely take over smaller targets and in some cases even compromise encryption.
- Industrial Espionage: Like the institutionalized hackers they have access to many resources, but they are not constrained to strictly legal things. Even brute-force cracking of encryption might not be beyond the powers of such attackers, but as they don't operate in the open it's hard to assess their capabilities realistically.

The "normal" user will most likely never have to consider the last two types of attackers, but depending on the paranoia and sensitivity of data one may wish to make it at least very hard for those to interfere. The classification into passive and active attackers could also be made, passive attackers only sniffing data where active attackers manipulate data whenever they can. Especially in the context of privacy and security the passive attackers can cause lots of damage and are hard to detect. For normal P2P networks they are mostly irrelevant, only active manipulation of data is seen as problematic.

Attack Strategies on normal P2P Networks

Once the attacker's capabilities are known it is of course sensible to consider his potential strategies. These attacks are all active in the sense that some data somewhere is manipulated. To list the more obvious attack methods:

- Spamming: By adding false and misleading data the performance of most P2P networks can be severely affected. For example filesharing networks may suffer from fake search results, misleading users to download bad or wrong files. Alternatively injecting false data into downloads will cause corruption and if detected repeat downloads.
- Flooding: Most P2P networks have some kind of hierarchy, for example index servers, "supernodes" and many other specialized functions. Flooding those with requests is a very cheap method to prevent users from accessing and using them. Especially routing infrastructure may suffer from such attacks.
- Knocking out single items: An attacker may wish to limit access to certain items. Repeated searches for this item, followed by DoS attacks on all users providing them, will severely limit the availability.
- False routing advertisements: Most decentral P2P networks use a Distributed Hashtable structure like Kademlia[5]. Returning wrong routing information may damage the P2P topology, worst case splitting the network into disjunct islands.

Most of these attacks can be done with a moderate investment, requiring only a large enough number of P2P clients.

Anonymizing Networks

While normal P2P networks offer a large surface for attacks already they are usually not designed to resist attacks. Designing for anonymity demands a high resistance to these attacks. Added to the "normal" issues there is a new layer of attacks as the origin of most data should be cloaked. It may be easier for an attacker to disrupt services when using these cloaking features, thus naive filtering methods are bound to fail. There is a conflict between security and anonymity.

At the same time, the larger a P2P network gets the more anonymity it can provide. This is obvious if the anonymity set is considered, that is the set of potential senders of a message. More users automatically enlarge this set if the underlying algorithms and methods have been designed well. The added features of anonymizing networks decrease the performance, which in turn reduces the rate of adoption

The most well-known anonymizing networks at the moment are most likely tor[1], freenet[2], I2P[3] and the email mixes like MixMinion[4].

Attacks on Anonymizing Networks

tor

As an example for the Onion Router-type networks we'll discuss attacks on tor. It is one of the most used and well-known networks, but it suffers from some interesting design limitations. They are most likely artifacts of the design, but they also make it very easy for attackers to cause damage.

Tor has an estimated 20-50.000 users, but obviously the anonymous part of it makes exact statistics very hard if not impossible. The network is split into client nodes, which provide access to the end-user, and servers which do the hard work of

routing the packets over multiple hops to obfuscate the origin. The big advantage of this design is the relative simplicity and the rather good performance of the dedicated servers.

There is a set of about 500-1000 servers which provide the backbone of the network. These provide excellent targets for all kinds of denial of service attacks. Also, these nodes are trusted more than strictly needed. For example the announced bandwidth of a server was a criterion for selection. While this has been fixed it was obviously easy to have a server announce infinite bandwidth so that as many connections as possible ran over it, subverting privacy and performance at no cost.

Another issue with the server/client design is that only dedicated servers provide gateways for traffic to go outside the network, for example to web servers to allow "anonymized browsing". These servers can inject tracking cookies, javascript and many other potentially dangerous bits into non-encrypted traffic. The implicit trust in the exit nodes makes all data that goes over them suspect.

Infinite Path Attack

While the mentioned attacks abuse implementation-specific issue of tor there are also attacks that are applicable to most Onion Routing networks. As data gets routed between multiple nodes in the network it also gets encrypted at each hop and end-to-end. This prevent any eavesdropping or correlation of ingoing and outgoing data on every node. Because of this a node cannot decide where a stream originated or where it terminates, it can only name the next hop forward and backwards. If an attacker uses this anonymity feature to build extremely long tunnels over as many servers as possible injecting one packet of data will cause each server to have to process at least one packet. Obviously this can easily be abused to overload the network with traffic amplification. A client with as much bandwidth as the slowest servers can take them out easily while causing a good amount of traffic on all others. The connection will break down when the first servers overload, but constructing multiple tunnels is quite trivial. And because the servers have no way to locate the source on their own they cannot defend against it easily. Collaboration would make that possible at the cost of anonymity, but that is not desirable.

The server-centric design of tor amplifies this attack as an attacker only has to take the rather small number of servers into consideration. We are not aware of a practical implementation of this attack.

Infinite Loop Attack

This is a variation of the Infinite Path attack, concentrating on a local denial of service. One needs a small number of victim servers, depending on the exact setup of the network three may be enough. Then one constructs a tunnel over those victims. But instead of leading to an endpoint this tunnel is looped back over the victim servers, causing each packet sent to pass each server multiple times. For example looping five times over the servers will cause them at least five times the traffic load one injects, thus making it trivial to knock out single servers or groups of servers for a short time. Again the anonymity features work against the defenders, the attacker will be very hard to locate.

Especially on anonymizing networks this attack is bad because it can be used to disable servers for a short time, thus allowing an attacker to locate the origin of a connection by watching the latency - if it goes up during an attack the connection is routed over the victim servers. The privacy implications are of course pretty bad, but luckily this attack is quite limited in what it can do. For example the attacker must control the data source, otherwise measuring the end-to-end latency becomes

impossible. Still it is moderately scary to consider what an attacker could do with such traffic amplification attacks.

A strategy to prevent these latency measurements are sg-mixes[8], but those cause a higher latency in general and are thus usually not desirable.

Using laws to hack networks

A special type of attack was seen on the JAP (JAP Anon Proxy) network [7]. The german BKA (FBI equivalent) managed to use some legal structures to compromise the (centrally managed) JAP services. This hack was quite elegant because the JAP operators were legally bound not to mention the compromise so that enough data could be logged to locate one evildoer. This compromised all participants in the network and removed any trust the users may have had. This attack is surprisingly elegant, but can only be done by government agencies. It is extremely hard to detect as all persons involved are unable to speak of it unless they wish to risk to spend some time in jail.

DoS mitigation

When faced with the threat of DoS attacks one will try to find some ways to at least minimize the impact. One method originally designed to fight email spam is hashcash[9]. It is based on partial hash collisions, i.e. finding data so that its hash has the top or bottom n bits equal to a given string. This is easy to verify (hash that data and compare), but hard to generate (for an n-bit collision one will usually need 2^n attempts). The big problem with such methods is the inequality in processing power - it is trivial for an attacker to acquire enough hardware to still flood while many other users will have a restricted level of service when they cannot generate hashcash fast enough.

It seems that this is not suitable for large-scale deployment.

Another strategy would be Trust Metrics. The basic idea is to assign a trust value based on for example historical data. Then one can evaluate wether one wishes to communicate with another entity or how many resources to dedicate based on that trust value. The biggest problem with all these systems is to find a valid metric that is hard to abuse. Local metrics that every client generates himself lack historical information for most communications, global systems are either extremely slow and complex or trivial to abuse. Only a few rudimentary attempts at trust metrics are in use, for example the credit system of the emule network [6]

Outlook

Exploiting most P2P networks is rather easy. Most networks were not designed with security or resilience in mind. This gives attackers a large surface to attack and manipulate P2P networks. Retrofitting features onto existing P2P networks may not be possible, but when designing a new network these problems should be taken into account. Resilience against attackers may also prove to be resilience against other failure modes, evaluating these defensive methods may offer insights for other P2P networks.

The damage to anonymizing networks from exploits is even larger than to other networks. We have enumerated many common attacks and hope to be able to implement the specific attacks on tor at some point in the future. At the same time designing a resilient P2P network may be an interesting challenge.

The DoS mitigation strategies we mentioned are neither complex nor reliable. Trust metrics and hashcash-like DoS mitigation tools offer many research opportunities and may yield tools and strategies that can be applied in many other places.

Bibliography

[1] "The TOR Onion router, available online: http://www.torproject.org/," 2007.

[2] "freenet," available online: http://freenetproject.org/

[3] "I2P, available online: http://www.i2p.net/, " 2007.

[4] "MixMinion, available online: http://mixminion.net/, " 2007.

[5] P. Maymounkov, and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proceedings of IPTPS02*, (Cambridge, USA), March 2002.

[6] "emule, available online: http://www.emule-project.net/, " 2007.

[7] "JAP, website and law enforcement information (german), official website at http://anon.inf.tu-dresden.de/strafverfolgung/index_de.html," 2007.

[8] D. Kesdogan, J. Egner, and R. Büschkes, "Stop-and-Go-MIXes Providing Probabilistic Anonymity in an Open System," Springer LNCS 1525, 1998.

[9] "Hashcash, available online: http://hashcash.org/," 2007.

Wireless Mesh- and Sensor-Networks

Wireless Mesh Networks

Thomas Staub, University of Bern
staub@iam.unibe.ch

Introduction

Wireless mesh networks (WMN) are becoming more and more popular. They offer a cost-efficient extension of network coverage, which is easy to deploy and maintain. They are based on wireless multi-hop communication and are compatible as well as interoperable with existing networks. Several deployments of WMNs devoted to research (MIT Roofnet [1, 2], Orbit [3], MSR [4]) exist. In addition, multiple cities offer already Internet broadband access as a community service by the means of WMNs [5], other cities are planning to deploy WMNs.

As WMNs may even provide communication possibilities in particular situations where certain systems (e.g. GSM) are overloaded and unavailable, our research focuses in exploiting WMNs as a robust and redundant carrier network for realtime communication such as IP telephony and video conferences during critical situations (e.g. emergency communication). The main drawback of a WMN is the unpredictable variation of the route quality caused by the unreliable and erroneous wireless medium. Node failures or link breaks result in high transmission delays, which makes the deployment of real-time applications a challenging task.

Our research investigates the possibilities of path diversity and multi-stream coding for improving the perceived quality of realtime applications. In a WMN usually multiple paths from the source to the destination exist. The characteristics of multiple paths are normally uncorrelated, i.e. the delay, jitter and loss rate of the paths differ significantly between the paths. Moreover, multiple uncorrelated paths provide redundancy in the transmission, offering possibilities to improve the quality of the transmission by different multi-stream coding schemes. The realtime communication is encoded in multiple streams, which are then transmitted separately on different paths. The individual streams are merged during the decoding at the destination. The received quality of the realtime communication is then depending on the used coding and the streams received in time. New algorithms have to be developed for the mapping of appropriate coded streams on the individual paths dependent on the current network conditions. Multi-path routing, multi-stream coding schemes and mapping algorithms form the architecture for "Adaptive Transport Over Multipaths (ATOM)".

Multi-path Routing and Routing Metrics

Existing multi-path routing protocols (AODVM [6], AOMDV [7], NDMR [8], SNDMR [9], AODV-BR [10], MP-AOMDV [11], SMR [12], MP-DSR [13], MSR [14], CHAMP [15], and MPABR [16]) do not support multi-channel communication. The hop-count metric is used for the routing decision. However, besides the missing support of multiple channels, the hop-count metric is further based on the wrong assumption that links either work or not. Neither packet loss, bandwidth, (self-)interference nor varying delays for the different links are considered.

Several enhanced routing metrics already exist. Expected Transmission Count (ETX, eq. 1) [17] is a single channel metric that considers forward and reverse link loss ratios (d_f, d_r). Expected Transmission Time (ETT, eq. 2) [4] is an enhancement of ETX and further takes the link bandwidth (b_f) into account. As no intra-flow and inter-flow interference is incorporated, ETX is only suited for single channel communication. Metric of Interference and Channel-Switching (MIC, eq. 5) [18]

is a metric that captures inter-flow as well as intra-flow interference. Inter-flow interference is taken into consideration as aggregated channel time of neighbouring nodes (N_l) that the link consumes (Interference aware Resource Usage, IRU). The signal strength of the interfering nodes and their produced amount of traffic are completely ignored which is a major flaw of MIC. The iAWARE (eq. 8) metric [19] incorporates these facts by the derivation of the interference ratio from SNR (Signal to Noise Ratio) and SINR (Signal to Interference and Noise Ratio). But none of these approaches take the route stability into account. An ideal metric for our use case should consider this robustness of the paths (stability of the route in the last period of time) and should further allow the selection of paths according to QoS requirements for appropriate mapping of paths, coding and streams. It is currently an open issue, on which we are working.

$$ETX = \frac{1}{d_f \times d_r} \tag{1}$$

$$ETT_l = ETX_l \frac{s}{b_l} \tag{2}$$

$$WCETT(p) = (1-\beta) \sum_{link\ l \in p} ETT_l + \beta \max_{1 \le j \le k} X_j \tag{3}$$

$$X_j = \sum_{\text{hop i on channel j}} ETT_i \qquad 1 \le j \le \#\text{channels} \tag{4}$$

$$MIC(p) = \frac{1}{N \times \min(ETT)} \sum_{link\ l \in p} IRU_l + \sum_{node\ i \in p} CSC_i \tag{5}$$

$$IRU_l = ETT_l \times N_l \tag{6}$$

$$CSC_i = \begin{cases} w_1 & \text{if } CH(prev(i)) \neq CH(i) \\ w_2 & \text{if } CH(prev(i)) = CH(i) \end{cases}, \qquad 0 \le w_1 \le w_2 \tag{7}$$

$$iAWARE(p) = (1-\alpha) \sum_{link\ l \in p} \frac{ETT_l}{IR_l} + \alpha \max_{1 \le j \le k} X_j \tag{8}$$

$$IR_i = \min IR_i(u), IR_i(v) \tag{9}$$

$$IR_i(u) = \frac{SINR_i(u)}{SNR_i(u)} \tag{10}$$

$$SINR_i(u) = \frac{P_u(v)}{N + \sum_{\omega \in \eta(u) - v} \tau(\omega) P_u(w)} \tag{11}$$

$$SNR_i(u) = \frac{P_u(v)}{N} \tag{12}$$

Multi Description Coding and Mapping

The realtime communication is encoded at the source to multiple streams, which are then transmitted independently on different paths. The decoder at the destination merges the individual streams to one stream. The final perceived quality depends on the type and number of the received streams. Whereas in Multi Description Coding (MDC) the streams are equivalent and any combination of received streams may be used for decoding, Layered Coding (LC) requires the faultless reception of the base layer stream, otherwise the enhancement streams become worthless. As LC normally introduces less redundancy and therefore uses less bandwidth, it may be the better choice if one rock-solid stable path and multiple varying paths exist. But if all available paths are varying, MC may outperform LC in the perceived quality.

The selection of coding scheme, number of streams, number of paths searched by

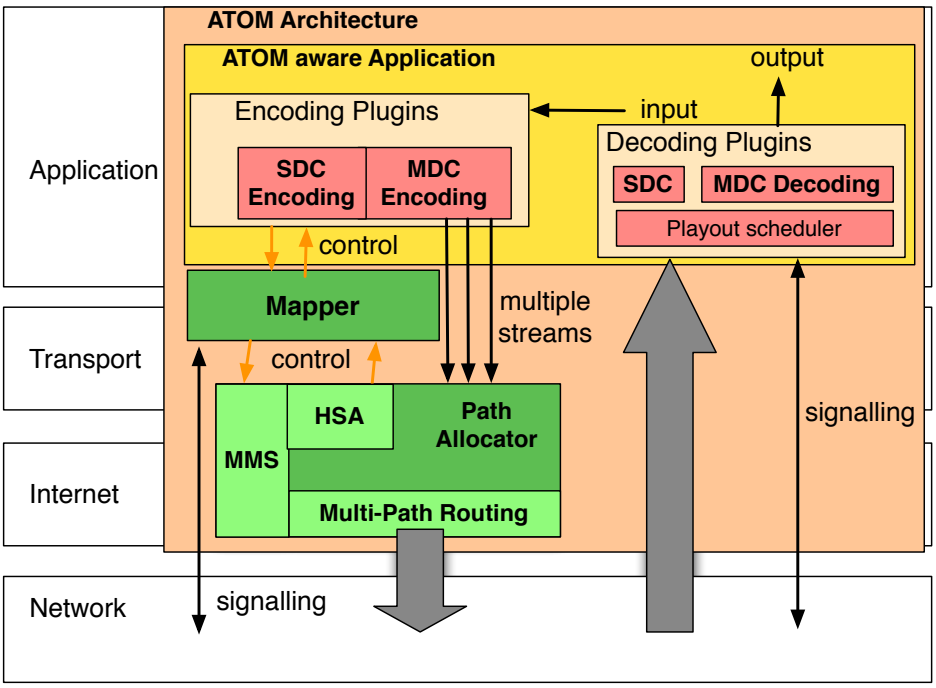

Figure 7: ATOM architecture

the routing scheme, and the mapping of the individual streams on the appropriate path strongly depend on the network conditions. Moreover, there are interactions between the tasks. Therefore, they represent a complex optimisation problem.

Adaptive Transmission Over Multipaths (ATOM)

The architecture "Adaptive Transimission Over Multipaths (ATOM)" combines path diversity, multi-stream coding, and path mapping and allocation (see Fig. 7). The system consists of ATOM aware application, Mapper, Path Allocator (PA) including Monitoring and Measurement System (MMS), History and Statistical Analyser (HSA), and Multi-Path Routing (MPR). An ATOM aware application signals the available encoding plugins to the Mapper component. The Mapper receives the actual network conditions from MMS, available routes from MPR, and route stability from HSA. It decides about the coding scheme to be used, the number of streams, the number of paths, and the mapping between the individual streams and the available paths. The decision is then communicated to the application as well as to PA. Further, the mapper signals the choice to the application at the destination. The source application now encodes its realtime data according to the configuration received from the Mapper and pass the individual streams to PA, which transmits them to the network. The PA is further responsible to keep the multi-path routing up-to-date. If the receiving application observes significant changes in the received quality of the realtime data or variations of the network are measured by MMS, the Mapper is notified in order to adapt encoding, mapping and path allocation.

Conclusions and Outlook

In order to exploit WMNs as a robust and redundant communication infrastructure for IP telephony and video conferencing, the architecture "Adaptive Transmission

Over Multipaths (ATOM)" combines path diversity, multi-stream coding, network monitoring, and mapping algorithms for enhancing the perceived quality. ATOM requires the development of an enhanced routing metric including values like route stability, statistical analysis of periodical outages besides normal QoS parameters. Further, the optimisation problem concerning selection of coding schemes, number of streams, number of paths, and mapping of the stream on the appropriate paths dependent on the network conditions has to be solved.

We plan to implement the whole architecture in the network simulator Omnet++. Some parts are already done. The metric iAWARE has to be enhanced and tested to support route stability information. Further, a prototype Linux implementation of ATOM on embedded hardware (PCEngines WRAP.2E, ALIX.3 boards) is envisioned. Some supporting actions for real world experimentation are finished. An embedded Linux image as well as a management architecture for WMNs [20] has been developed. In addition, works on a multi-channel prototype and an emulated wireless mesh network (VirtualMesh) are on-going.

Bibliography

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *International Conferences on Broadband Networks (BroadNets)*, (San José, CA, USA), October 2004.

[2] J. C. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *11th Annual International Conference on Mobile Computing and Networking (MOBICOM 2005)*, (Cologne, Germany), pp. 31–42, August–September 2005.

[3] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, vol. 3, pp. 1664 – 1669, March 2005.

[4] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 114–128, ACM Press, 2004.

[5] R. Karrer, A. Sabharwal, and E. Knightly, "Enabling large-scale wireless broadband: The case for taps," in *2nd Workshop on Hot Topics in Networks (HotNets II)*, (Cambridge, MA, USA), November 2003.

[6] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," *IEEE Infocom 2003 (INFOCOM)*, (San Francisco, CA, USA), May–April 2003.

[7] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," in *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, pp. 92–93, July 2002.

[8] X. Li and L. Cuthbert, "On-demand node-disjoint multipath routing in wireless ad hoc networks," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, (Tampa, Florida, USA), pp. 419–420, November 2004.

[9] W. Xu, P. Yan, and D. Xia, "Similar node-disjoint multi-paths routing in wireless ad hoc networks," *International Conference on Wireless Communications,*

Networking and Mobile Computing (WiMob'05), (Montreal, Quebec, Canada), vol. 2, pp. 731–734, August 2005.

[10] S. J. Lee and M. Gerla, "Aodv-br: backup routing in ad hoc networks," *IEEE Wireless Communications and Networking Conference (WCNC 2000)*, (Chicago, IL, USA), vol. 3, pp. 1311–1316, September 2000.

[11] P. Sambasivam, A. Murthy, and E. M. Belding-Royer, "Dynamically adaptive multipath routing based on aodv," *Third Annual Mediterranean Ad Hoc Networking Worksho (MED-HOC-NET 04)*, (Bodrum, Turkey), June 2004.

[12] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," *IEEE International Conference on Communications (ICC)*, (Helsinki, Finland), vol. 10, pp. 3201–3205, June 2001.

[13] R. Leung, J. Liu, E. Poon, A. L. C. Chan, and B. Li, "Mp-dsr: a qos-aware multi-path dynamic source routing protocol for wireless ad-hoc networks ," *26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, (Tampa, Florida, USA), pp. 132–141, November 2001.

[14] L. Wang, Y. Shu, M. Dong, L. Zhang, and O. W. W. Yang, "Adaptive multipath source routing in ad hoc networks," *IEEE International Conference on Communications (ICC 2001)*, (St. Petersburg, Russia), pp. 867–871, vol.3, June 2001.

[15] A. Valera, W. K. Seah, and S. Rao, "Cooperative packet caching and shortest multipath routing in mobile ad hoc networks," *IEEE Infocom 2003 (INFOCOM)*, (San Francisco, CA, USA), May–April 2003.

[16] P. McCarthy and D. Grigoras, "Multipath associativity based routing," *2nd International Conference on Wireless on Demand Network Systems and Service (WONS 2005)*, (St. Moritz, Switzerland), vol. 00, pp. 60–69, IEEE Computer Society, January 2005.

[17] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing ," in *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom'03)*, (San Diego, California), September 2003.

[18] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," *First IEEE Workshop on Wireless Mesh Networks (WiMesh)*, (Santa Clara, CA, USA), September 2005.

[19] A. P. Subramanian, M. M. Buddhikot, and S. Miller, "Interference aware routing in multi-radio wireless mesh networks," *2nd IEEE Workshop on Wireless Mesh Networks (WiMesh 2006)*, (Reston, Virginia, USA), pp. 55–63, September 2006.

[20] T. Staub, D. Balsiger, M. Lustenberger, and T. Braun "Secure remote management and software distribution for wireless mesh networks," *7th International Workshop on Applications and Services in Wireless Networks (ASWN 2007)*, (Santander, Spain), pp. 47–54, May 2007.

Energy-efficient Management of Heterogeneous Wireless Sensor Networks

Gerald Wagenknecht and Markus Anwander, University of Bern
wagen@iam.unibe.ch and anwander@iam.unibe.ch

Introduction

In this report we introduce a management architecture for heterogeneous wireless sensor networks (WSN). Further, we present the used sensor wireless node, which build the heterogeneous WSN and Contiki, the used operating system, which is running on the nodes. At last some selected transport protocols and the possibility to adapt TCP for WSNs is presented.

Management Scenario and Tasks

In a wireless sensor network (WSN) different types of sensor nodes, which might measure different sensor values and perform different tasks, are running. Possible tasks are for example event detection, localization, tracking, monitoring and many more. Existing sensor node platforms in general have different radio modules, which can not communicate with other types of sensor nodes. Sensor nodes of the same type build a sensor subnetwork (SSN). A heterogeneous WSN is built from several sensor subnetworks. To interconnect such a heterogeneous WSN we propose wireless mesh nodes as gateways between these SSNs. The mesh node works as gateway by using a sensor node gateway which is plugged into the mesh node via USB, RS232, or another serial interface. The wireless mesh nodes communicate among each other via IEEE 802.11. For the communication within a heterogeneous WSN a uniform communication protocol is required. Therefore, we propose to adapt and enhance TCP/IP for its use in WSNs as a general communication protocol suite. A possible scenario is shown in Fig. 8. We have evaluated a number of sensor nodes and selected four of them to build a heterogeneous sensor network: ESB nodes [1], tmote SKY [2], BTnodes [3], and MICAz [4]. For the management backbone a wireless mesh network (WMN) consisting of Wireless Router Application Platform boards (WRAP) [4] nodes have been chosen. A possible scenario of a heterogeneous wireless sensor network is shown in Figure 8.

The use of such an architecture with a wireless mesh network as backbone has various advantages. In addition to the communication gateway functions between different SSNs mesh nodes also perform management tasks for heterogeneous WSNs. Moreover, the use of a WMN has advantages by dividing a huge WSN into smaller SSNs. Thus, the number of hops can be decreased significantly. These results in a better communication performance with a lower packet delay and packet delay variance as well as a lower packet loss and energy consumption per packet sent.

Among the task concerning the communication, the mesh nodes also provide management functionality for heterogeneous WSNs. In a complex heterogeneous WSN with a large number of different sensor nodes there is a need for a comprehensive management architecture. It might be possible that the sensor nodes and the running applications do not work properly, their configuration has to be changed and their software has to be updated during their lifetime. From the management point of view there are several tasks required to manage a heterogeneous WSN and its sensor nodes:

- monitoring the WSN and the sensor nodes
- (re)configuring the WSN and the sensor nodes

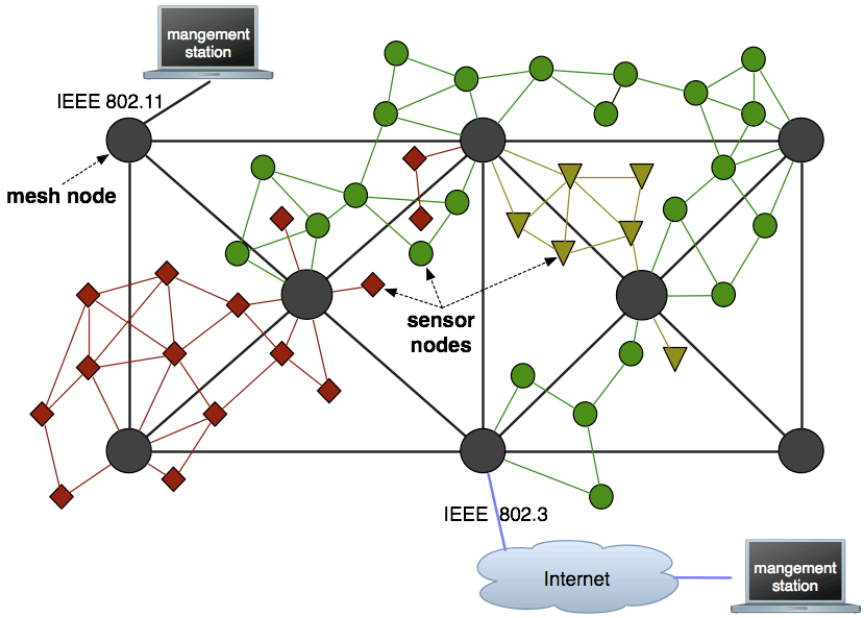

Figure 8: A possible scenario of a heterogeneous wireless sensor networks

- updating and reprogramming the sensor nodes
- aggregating and managing the sensor values [optional]

Management Architecture

The architecture to manage heterogeneous WSNs contains the following structural elements: one or more management stations, several wireless mesh nodes as management nodes, sensor node gateways plugged into a mesh node, and the different sensor nodes.

The **management station** is divided into two parts. It consists of a laptop or remote workstation with a web browser as a user interface and a management system for wireless mesh networks [5]. This management system includes a HTTP server with PHP and the management components. The WSN monitor is responsible for the monitoring requests of the WSN and its sensor nodes. It shows all available information of all mesh nodes and all sensor nodes. The WSN configurator is responsible for all requests concerning the configuration of the sensor nodes. It sends a configuration command to selected sensor nodes. The code update manager is divided into two different parts. One part propagates a new image into the mesh network using the CFEngine. The other part shows all available program versions for all sensor node platforms and starts the updating process for the selected sensor nodes. The sensor value manager queries sensor data, which is collected and stored in the mesh nodes.

The **WSN manager** is located on every mesh node and includes the functionality to manage the different sensor subnetworks. It consists of three databases, four program modules. The databases are: the program version database (stores all versions of all programs for all platforms, which can be installed on the sensor nodes), the WSN information database (stores all information about the sensor nodes and the WSN), and the sensor value database (stores all data measured by the sensors). The modules consist of the WSN monitor module, the WSN configurator module, the code update manager module, and the sensor value manager module. The WSN

monitor module answer to the monitor requests from the management station and stores the data coming from the sensor nodes into the databases. The WSN configurator module is responsible for all configuration tasks. It connects to the sensor nodes to query properties and send the configuration commands. The code update manager module is responsible for storing the newly received images (and related information) in the program version database, and for delivering the images to the sensor nodes. To reduce the amount of transmitted data to the sensor nodes, it compresses the image of the new program version or it calculates the differential patch between the old and the new version. The sensor value manager module is responsible for aggregating and storing the sensor values.

Every sensor node has a **sensor node manager** running to handle the management tasks. This consists of a sensor node monitor, a sensor node configurator, a sensor value sender, and a code updater. The sensor node monitor is responsible for handling the monitor requests and sends the requested values back to the mesh node. The sensor node configurator executes the configuration commands. The code updater is responsible for the code replacement on the sensor node. It recalculates the new image from the differential patch and the old image or it decompresses the image. To replace the modules the code updater uses the Contiki module loader to load the image into the memory. The sensor value sender is responsible for sending measured sensor values. It collects the newly measured values from a sensor and forwards them to the mesh node.

Management Protocols

As defined there are four management tasks. We consider the first three of them, because managing sensor values is an optional task and not an important issue of our management architecture. There are three management protocols to handle the considered management tasks: WSN monitoring protocol, WSN configuration protocol, and code update protocol.

Monitoring of the WSN can be performed in two ways. First the management station explores the mesh network and the subordinate SSNs. Alternatively the user can query a selected sensor directly. All information from every sensor node is independent from the sensor node platform stored in the WSN information database and distributed in the whole WMN. Thus, we have a general view over the whole heterogeneous WSN. For displaying sensor nodes, network topology, etc. no additional transmissions to the sensor nodes are required and the request of a mesh node is much faster than a request to a sensor node. In the alternative case, the user wants to request information from a sensor node directly.

Possible configuration scenarios for the **WSN configuration protocol** are: turn the sensors on/off, change sensing cycles, changing routing tables, configuring applications, etc. With the configuration protocol a configuration command is send to the sensor nodes. Because on the sensor node there are a universal interface which hides the sensor node type specific characteristics, the packets with the configuration commands are independent of the type. The heterogeneity of the WSN is hidden from the user.

The **code update protocol** consists of three main subtasks: uploading the new image and distributing it within the mesh network, providing information about all available programs for the different sensor node platforms, and finally performing the update. First the image of the new program has to be uploaded, distributed within the WMN and stored in the program version database. In a second part, the protocol provides the information about all available program version for all sensor node platforms to the user. The main part of the protocol is the updating process of the sensor nodes. The protocol delivers the images to the sensor nodes and handles the updating process on the sensor node.

Sensor Nodes

A WSN consists of a huge number of nodes, often randomly distributed in a large area. Generally, a SN consists of a micro-controller, some sensors, and a low-power radio for communication. Currently available SNs are mainly prototypes for research purposes. Beside the already owned ESB nodes [1], tmote SKY [2], BTnodes [3] and MICAz have been chosen. For the management backbone a Wireless Router Application Platform Board (WRAP) [5] has been selected.

Embedded Sensor Boards (ESB) [1] node provides several communication interfaces, some sensors for monitoring the environment and a MSP430F149 micro-controller. The platform has already been used in several research projects at University of Bern. A tmote SKY [2] provides an IEEE 802.15.4 radio interface, a USB Port for development and communication and a MSP430F1611 micro-controller. An 802.15.4 data frame can include 80 bytes payload of a TPC/IP packet. A BTnode [3] basically come with the same hardware features as the widely used Mica2 Mote. It provides several communication interfaces, including a Bluetooth radio and a ATMega128L micro-controller. A MICAz [4] node provides several communication interfaces and a ATMega128L micro-controller. Like the tmote sky node it has an integrated IEEE 802.15.4 radio transmitter. The WRAP board provides several communication interfaces. The WRAP boards can be used to build a fully meshed network using IEEE 802.11 radio transmitters. USB, RS232, LPC, I2C interfaces make it possible to connect every SN platform over a serial interface to the WRAP board. Figure 9 shows the memory capacities of the the different SN platforms.

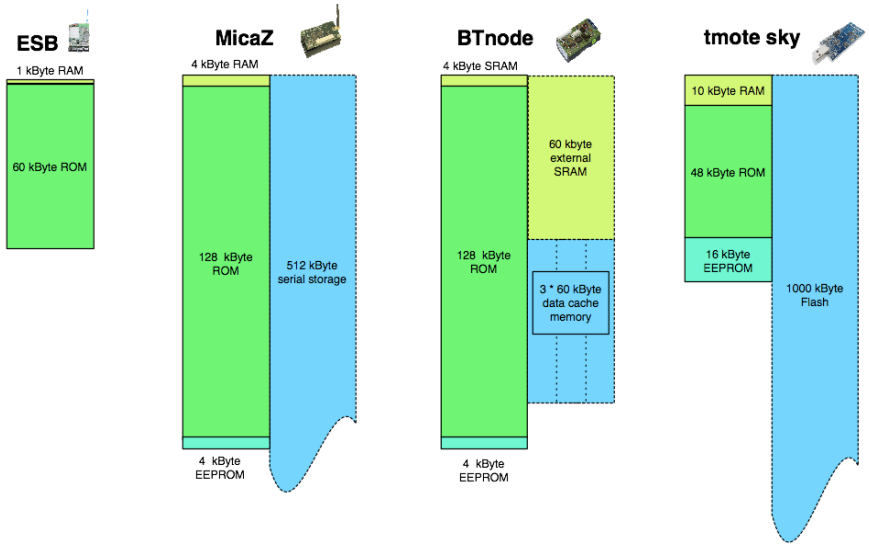

Figure 9: Memory capacities of the the different SN platforms

Contiki

We have chosen Contiki [6] [7] as an appropriate operating system to support dynamic and energy-efficient reconfiguration as well as code updates of different sensor node platforms. It is written in C and supports over 14 platforms and 5 CPUs. Contiki supports preemptive multi-threading, inter-process communication and dynamic run-time linking of standard Executable Linkable Format (ELF) files. ELF is a standard format for relocatable object code and executable files. Program modules can be updated and loaded at run-time. The lightweight and compact base

system is highly portable to other sensor platforms. Contiki also offers a minimal TCP/IP stack for communication. Contiki and the network simulator COOJA for Contiki are open source projects and run under BSD license. Contiki supports two different communication stacks: Rime [9] and (μIP) [8]. (μIP) has been developed for the communication with TCP/IP-based networks. Rime has been designed for wireless communication. Contiki applications can use (μIP) or Rime or both stacks simultaneously for communication. Rime can also run over (μIP) and vice versa as depicted in Figure. Protothreads, a novel thread-like construct on top of the event-driven kernel, reduces the complexity of event-driven programs by removing state machines. Code for applications and the Contiki core are strictly separated. To recompile applications for different platforms different make commands have been defined. Ideally, no changes are needed for the application to port it to another platform. In comparison to TinyOS [10] , Contiki appears to be more appropriate for our project. The kernel properties and the high portability of Contiki are important factors to support a management infrastructure for heterogeneous WSNs. Lately, the developers of Contiki arranged the first international Contiki workshop in Stockholm on the 26-27 March this year [6]. Discussions with the developers and other workshop participants confirmed the choice of Contiki as the most appropriate operating system for our purposes.

Communication Protocols

The Figure 10 shows possible protocol stacks for the sensor nodes. In this section the SLIP protocol to connect the sensor node and the mesh node and the protocol stack of the CC2420 radio on the tmote SKY nodes are described.

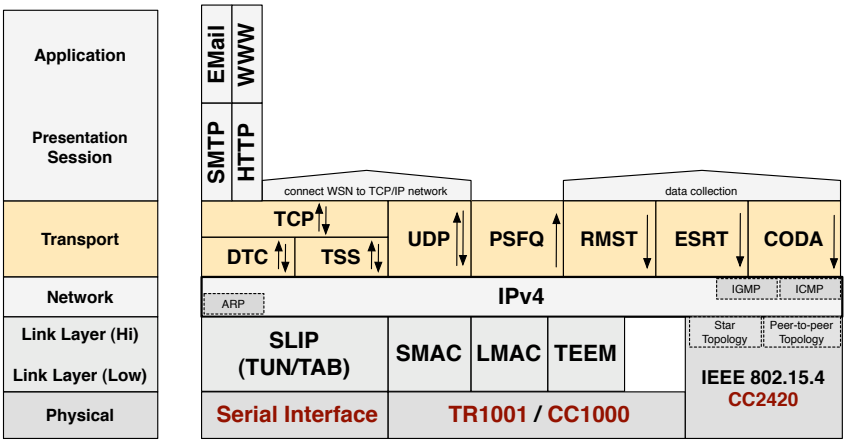

Figure 10: Possible protocol stacks for the sensor nodes

IEEE 802.15.4

The CC2420 [11] radio module used by tmote SKY and MICAz is designed to support IEEE 802.15.4 [12]. Because of the specification of the IEEE 802.15.4 protocol, the datalink frame have to be a maximum length of 127 bytes. In the star topology, the PAN (Personal Area Network) Coordinator broadcasts periodically a beacon with synchronization information. Every node can only communicate with the PAN Coordinator. In our WSN communication scenario we require a more flexible communication between the sensor nodes. Using the IEEE 802.15.4 P2P (Peer-To-

Peer) topology, every node can communicate with others. The synchronization in a P2P topology is not controlled by the PAN Coordinator.

SLIP

The Line Internet Protocol (SLIP) [13] connects the host IP layer of an mesh node with the client IP layer of a SN. It simply sends byte per byte over a serial interface. There is no addressing or error detection implemented in the SLIP protocol. The SLIP protocol defines two special characters: ESC and END. ESC is 333_8. END is 300_8. After the last byte of a packet the END character is sent. Therefore the data byte 3008 is substituted with a two bytes sequence of ESC and 334_8 $(300 + 334)_8$). The data byte 333_8 is substituted with a two bytes sequence of ESC and 335_8 $(300 + 335)_8$. A SLIP device has been implemented in the Contiki core. The mesh node uses the TUN/TAP driver, which is a part of the Linux kernel.

Reliable Data Transport

Reliable data transport is a very important issue in WSNs. We introduce some selected transport protocols: PSFQ, RMST, ESRT, and CODA. Further we introduce TSS/DTC as TCP support for WSNs.

Pump Slowly Fetch Frequently (PSFQ) [14] is a transport protocol for reliable data transport in WSNs. It distributes the data from the source at a relatively low rate, and allows the nodes to recover missing data packets from immediate neighbors aggressively. PSFQ cannot provide reliability for single packet messages, because it is a pure negative acknowledgment-based scheme. Further, the pump operation needs in-sequence forwarding for message delivery, which can waste bandwidth.

Reliable Data Transport in Sensor Networks (RMST) [15] has been designed to be used together with directed diffusion [16]. RMST is used for sensor data transfer, but not for control data transfer. It can provide a caching mechanism within the intermediate nodes, but requires additional negative acknowledgment messages. Those are sent by an intermediate node to its upstream neighbor, when it detects, for example using timeouts, holes in the data flow. As a reaction on negative acknowledgment, an upstream node can retransmit cached packets. The protocol designers assume a low number of bytes in flight and that the intermediate nodes can completely cache this amount of data. The authors found out that for packet loss rates below 10%, the combined caching and negative acknowledgment mechanism is more efficient than a reliable link layer approach based on ARQ due to the overhead caused by link level acknowledgments. On the other hand, end-to-end processing of negative acknowledgments is extremely inefficient for packet loss rates above 10%.

Event-to-Sink Reliable Transport (ESRT) [17] aims to support reliable sensor data transport in wireless networks. It includes congestion control and mechanisms to achieve reliability. The reliability is controlled by adapting a rate at which the sink sends state reports back to the source. The frequency of the reports depends on the observed and desired reliability as well as the needs from congestion control. As in the case of PSFQ, a special protocol has been proposed.

Congestion Detection and Avoidance (CODA) [18] is based on congestion detection by monitoring channel utilization and buffer occupancy at the receiver. Detected congestion situations are signaled towards the source using back-pressure signals (open-loop). Nodes receiving signals indicating congestion throttle down their transmission rate. In addition, a closed-loop mechanism operates on a longer time-scale. Based on acknowledgments received from the sink, sources regulate themselves. Lost acknowledgments result

Another alternative would be the using of TCP/IP in wireless sensor networks. There are some problems using TCP over air links, such as high rates packet loss, low throughput, and poor energy-efficiency, but TCP is well understood, widely used as standard and provides interoperability with other networks (e.g. internet). TSS (TCP Support for WSNs) [20] supports mechanisms which allows the using of TCP in WSNs in an energy-efficient way. The most important mechanism is the Distributed TCP Caching (DTC) [19], which aims to avoid energy-costly end-to-end retransmissions by caching TCP data segments inside the network and retransmitting segments locally. It assumes limited memory resources available for caching and proposes to cache a single segment per node. It uses a backpressure mechanisms for congestion control, which means that a node stops forwarding until all previously lost packets are recovered. Further local regeneration and aggressive recovery of TCP ACKs are used. The proposed mechanisms reduce the number of TCP data segment transmissions needed to transfer a certain amount of data across a WSN with relatively high bit / packet error rates.

Bibliography

[1] "Scatterweb, Platform for self-configuring wireless sensor networks, available online: http://www.scatterweb.web," 2007.

[2] "Tmote SKY, platform for self-configuring wireless sensor networks, available online: http://www.moteiv.com," 2007.

[3] "BTnode, versatile and flexible platform for fast-prototyping of sensor and ad-hoc networks, available online: http://www.btnode.ethz.ch," 2007.

[4] "MICAz, a 2.4 GHz, IEEE/ZigBee 802.15.4, board used for low-power, wireless, sensor networks, available online: http://www.xbow.com," 2007.

[5] "WRAP, Wireless router application platform board, available online: http://www.pcengines.ch," 2007.

[6] "Contiki, available online: http://www.sics.se/contiki," 2007.

[7] A. Dunkels, B. Grönvall, and Th. Voigt, "A Dynamic Operating System for Memory-Constrained Networked Embedded Systems," in *Proceedings of the 1st IEEE Workshop on Embedded Networked Sensors*, (Tampa, FL, USA), November 2004.

[8] A. Dunkels, "Full TCP/IP for 8-bit Architectures," in *ACM MobiSys*, (San Francisco, USA), pp. 85–98, May 2003.

[9] A. Dunkels, "Rime - a lightweight layered communication stack for sensor networks," in *EWSN'07: European Conference on Wireless Sensor Networks, Poster/Demo session*, (Delft, The Netherlands), January 2007.

[10] "TinyOS, Open-source operating system designed for wireless embedded sensor networks, available online: http://www.tinyos.net," 2007.

[11] "Cc2420, Datasheet for the chipcon cc2420 2.4 ghz ieee 802.15.4 compliant rf transceiver, vailable online: http://www.tinyos.net," 2007.

[12] "802.15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans)," IEEE Computer Society, February 2006.

[13] J. L. Romkey, "Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP." RFC 1055, June 1988.

[14] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: A reliable transport protocol for wireless sensor networks" in *WSNA'02: Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications*, (Atlanta, GA, USA), September 2002.

[15] F. Stann, and J. Heidemann, "RMST: Reliable data transport in sensor networks," in *SNPA'03: Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, (Anchorage, AK, USA), May 2003.

[16] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," in *IEEE/ACM Transaction on Networking*, 11(1):216, February 2002.

[17] Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," in *MobiHoc'03: Proceedings of the 4th ACM international Symposium on Mobile Ad Hoc Networking and Computing*, (Annapolis, MD, USA), June 2003.

[18] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion detection and avoidance in sensor networks," in *SenSys'03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, (Los Angeles, CA, USA), November 2003.

[19] A. Dunkels, T. Voigt, J. Alonso, and H. Ritter, "Distributed TCP caching for wireless sensor networks," in *MedHocNet'04: Proceedings of the Third Annual Mediterranean Ad Hoc Networking Workshop*, June 2004.

[20] T. Braun, T. Voigt, and A. Dunkels, in "TCP support for sensor networks," *WONS'07: Proceedings of the 4th Annual Conference on Wireless on Demand Network Systems and Services*, (Obergurgl, Austria), January 2007.

Event Modeling and Membership Determination using Nonlinear Optimization

Markus Waelchli, University of Bern
waelchli@iam.unibe.ch

Introduction

Sensor networks consist of hundreds or thousands of small, battery-powered sensor nodes equipped with a wireless radio to communicate and a number of sensors, which are used to measure and monitor events in the physical environment. In this paper we discuss some state-of-the art and possible solutions to the detection, tracking, and classification of such events. Thereby, in the past, scientific focus was either spent on accurate localization by applying collaborative signal processing methods (CSP), or on efficient detection and tracking while keeping communication costs low. However, parts of both research directions need to be integrated to provide a solution which is satisfactory for real applications. With the Distributed Event Localization and Tracking (DELTA) framework we focus on doing exactly this. So far, DELTA provides methods to efficiently detect and track moving events. Hence, methods for accurate localization and classification of the events remain to be supplied what is addressed in this report.

Currently, a measurement-based leader election algorithm [1] determines a group leader that is responsible for group maintenance, data gathering and processing, and reporting information to a base station. The basic operations are depicted in Figure 11.

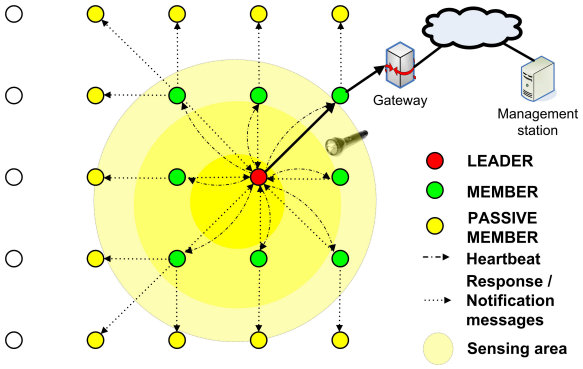

Figure 11: Event detection, tracking group organization, localization and reporting with DELTA.

Apart of their tasks in cluster organization, the heartbeat and response messages can further be used for an accurate computation of the event's current position and the signal strength with which the signal was emitted. This emitted signal strength is characteristic for the event, and is therefore used for the classification of the events. The computation of both the event location and the emitted signal strength are non-trivial problems addressed with nonlinear optimization methods. Once computed, the data is routed to the base station. After having collected a (training) set of data, the base station performs clustering on this data to learn classes of events, which are then used to model a fuzzy logic control engine (FLC). This FLC is downloaded on the sensor nodes and is used for online outlier detection and false alarms prevention. The DELTA framework is designed to run on tiny nodes and is being implemented on the ESB sensor platform [2].

Overview

The localization of events based on energy decay models has a long tradition in the collaborative signal processing (CSP) community. A number of contributions using sensor networks in that context can be found in the SensIT project [3]. One assumption in all these works is that the emitted signal propagates isotropically (e.g. sound and light from point sources). Thus, the received signal strength ρ_i at a sensor node i located at position ξ_i is depends on the event position $\mathbf{x}$ according to the following equation:

$$\rho_i = \frac{c}{\|\mathbf{x} - \xi_i\|^\alpha} + \omega \tag{1}$$

where c represents the amplitude of the emitted signal, α is the attenuation degree of the considered signal, ω is some additional white Gaussian noise, and $\|.\|$ is the Euclidean norm. The quadratic equation contains three unknown variables: $\mathbf{x}$ consisting of the x and y coordinates and c. Given N sensors, where N is greater than the problem dimension (i.e. greater than 3), the problem can be formualated as a non-linear least square objective function:

$$f(\mathbf{x}, c) = \sum_{i=1}^{k} \left(\rho_i - \frac{c}{\|\mathbf{x} - \xi_i\|^\alpha} \right)^2 \tag{2}$$

This function can be solved using nonlinear optimization methods. The resulting computations of the event location and its emitted signal strength(s) are then routed to the base station. Having collected a sufficiently large training set at the base station, a fuzzy k-means clustering algorithm is applied. The resulting clusters describe the different event types and are used to model a Fuzzy Logic Controller (FLC) including a Mamdani inference scheme.

Event localization with nonlinear optimization

Because of complexity issues in terms of memory and time, the Nelder-Mead's Simplex Downhill (SD) [[4],[5]] algorithm was chosen. In addition, the Conjugate Gradient descent method (CG) [5] was evaluated. CG is in general more efficient, but also more complex. Both algorithms search the solution space for minima. Once entered such a minimum there is no chance to exit it. Therefore, starting the search at a well located position is important. Searching the global minimum is far too complex as it requires an additional search procedure such as Monte Carlo. Both the Simplex Downhill and the Conjugate Gradient methods are described in detail in [1].

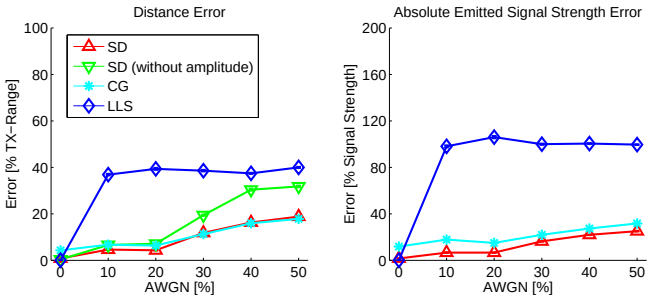

Figure 12: Localization accuracy of LLS, SD, and CG.

The SD and the CG methods have been evaluated in Matlab. Additionally, the localization problem was formulated as a linearized least square problem. Four

nodes are arranged in a square with a side length of 125. The event is randomly placed within this square. 200 localizations are performed with a confidence of 95%. For Simplex Downhill the starting simplex is constructed at the center of area of the sensing nodes and their measurements. Noisy measurements are introduced as additional white Gaussian noise (AWGN).

Fig. 12 shows the results. Obviously, the LLS method is not suitable. Almost independently from the noise level, the position error is always about 40% of the distance between two nodes. The signal amplitude error is even worse. On the other hand, the Simplex Downhill algorithm performs best and its search depth of approximately 120 iterations is also affordable. The problem of the LLS method is illustrated in Fig. 13.

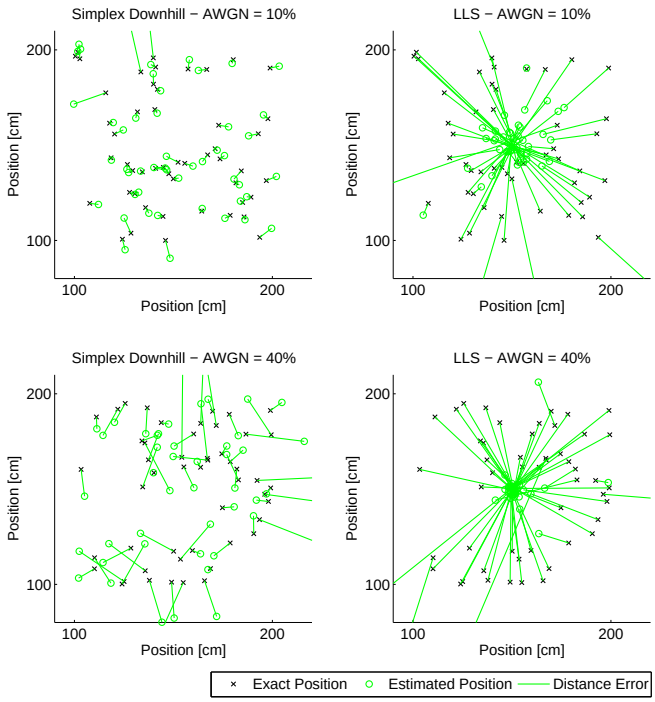

Figure 13: Location estimation of Simplex Downhill (left) and LLS (right).

Only little affected by the noise level, the majority of the LLS estimations is close to the center of the sensing area. The distance errors (lines between the exact event positions and the estimations) are accordingly high. The linearization lacks drastic accuracy in presence of signal noise and if the event is not located close to the center of area of all observing sensors and is therefore not suitable for our purposes. In contrast, the Simplex Downhill method does not have this limitation (shown in the subfigures on the left side of Fig. 13). With an overdetermined system the problems of the LLS become smaller. However, SD still performs better and additional data is often unavailable.

Classification of events

In this chapter the clustering of the output derived from the Simplex Downhill method is described. Based on a fuzzy k-means clustering algorithm, k fuzzy sets modeling the signal amplitudes of k different events are derived. The output of the clustering of a training set generated on ESB sensor nodes with 5 different light

sources is depicted in Fig. 14.

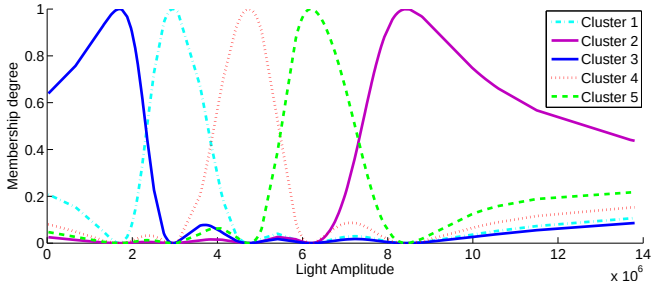

Figure 14: Fuzzy clusters learned from a training set generated from real world data.

These fuzzy clusters already model the different shapes of the emitted signal strengths and can therefore be directly used for defuzzification. For the defuzzification we intend to use the Mamdani [6] inference scheme. Thereby, the membership of each computed signal amplitude to the learned fuzzy sets (fuzzy clusters) is determined. From the resulting fuzzy set, which is a union of the individual membership degrees, the center of area is computed, and based on this value the event is classified.

The fuzzy sets generated by the clustering algorithm are generic and downloaded on the sensors. Whenever the fuzzy k-means is reinitiated, most probably because the performance of the classifier requires an update, the resulting fuzzy sets are distributed in the network and replace the previous versions.

Outlook

In future work we will evaluate the Simplex Downhill method in more detail and also its performance in realistic scenarios, i.e. real world experiments. Furthermore, the FLC system will be implemented and its applicability tested on the sensor nodes. With the resulting system we hope to be able to reliably classify events and filter outliers.

Bibliography

[1] M. Wälchli, P. Skoczylas, M. Meer, and T. Braun, "Distributed event localization and tracking with wireless sensors," in *5th International Conference on Wired/Wireless Internet Communications (WWIC '07)*, (Coimbra, Portugal), May 2007.

[2] "Scatterweb, available online: http://www.scatterweb.net," 2007.

[3] "SensIT, available online: http://www.ece.wisc.edu/ sensit/," 2007.

[4] J. A. Nelder, and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol 7, pp. 308–313, 1965.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C: The Art of Scientific Computing," Cambridge University Press, 1992.

[6] E.H. Mamdani, "Advances in the linguistic synthesis of fuzzy controller," *International Journal of Man-Machine Studies*, vol. 8, no. 6, pp. 669–678, 1976

Network Security

Stronger Authentication in E-Commerce - How to protect even naïve Users against Phishing, Pharming, and MITM attacks

Carolin Latze, University of Fribourg
carolin.latze@unifr.ch

Introduction

Common attacks on e-commerce application nowadays are *Phishing*, *Pharming*, and *Man-in-the-Middle* attacks.

In a *Phishing* attack, the attacker tries to steal user credentials using social engineering methods. *Pharming* attacks are a little bit more sophisticated. They include DNS poisoning or manipulating the user's host file to redirect a connection to a malicious server. That server will then ask for the credentials the user uses to authenticate to the original server. The most complex attacks are *Man-in-the-Middle* (*MITM*) attacks. The attacker has to install itself between client and server and snoop all the traffic. Such an attack is very easy on unsecured networks, but also possible on secured connections. In the latter case, the attacker has to establish a secure connection between itself and the client and between itself and the server. An experienced user would recognize this for instance because of wrong certificates, but a naïve user will just give away its credentials without recognizing anything.

There are several common solutions to overcome these problems. In general, they can be classified into "Bigger Warning Messages", "Better Passwords", "SSL Extensions" and "Trusted Devices".

"Bigger Warning Messages" include solutions that propose to use so called trusted areas in webbrowsers to display certificates (as logos) [1], solutions that want to monitor websites to detect spoofing sites [2], or solutions that extend SSL certificates with visual artefacts [3]. It is obvious, that such solutions do not prevent the attacks mentioned above especially when we speak about naïve users.

The authors of [4] argue that most of the security problems arise because of very simple user passwords. That is why they propose to generate a new password for every account. Such a solution does not avoid any attacks, but reduces them to one account.

In 2007, the authors of [5] proposed to extend the SSL protocol in a way that SSL authentication and user authentication are coupled. The authors think that the decoupling of those two session makes *MITM* attacks much more easier. In our opinion, this solution does not hinder the user to accept wrong certificates and therefore allows *MITM* attacks.

The last bundle of solutions makes use of trusted devices. One approach that makes use of such devices has been presented in [6]. The authors of [6] propose to use a so called "secure wallet" to store credentials and to decide whether a website is authentic or not. This approach may be extended using trusted computing to protect against malware phishing. This approach includes the need for the client to know the server it wants to connect to. To realize this in a comfortable way, the client is allowed to make the first connect to any server and stores then a fingerprint of this server for future connects. That means, that this scheme is highly vulnerable to attacks on the first connect. Another approach using trusted devices proposed to use such devices as second authenticator [7]. Such a device may be a cellphone, a PDA or a smart watch. We think, that always using such a device is not very convenient.

Stronger Authentication Using TPMs

We provide a solution, that makes also use of trusted devices, since such approaches are the only one that are able to influence the user strongly. For the remainder of this section, we will first describe the trusted device we use, called *Trusted Platform Module* (*TPM*), followed by an detailed description of our authentication scheme.

The Trusted Platform Module (TPM)

The *TPM* as specified by the Trusted Computing Group (TCG) [8], is a module, which provides cryptographic functions and is able to store hashes securely in so called Platform Configuration Registers (PCRs). The *TPM* should be a low cost device, otherwise nobody will buy it. This leads to the need of only a minimal subset of cryptographic methods provided by the module and minimal hardware requirements.

The module holds confidential information like cryptographic keys in a protected environment. Furthermore, the *TPM* provides methods to "seal" sensible data to a special state of the platform. This means that these data may only be released if the platform is in a specified trustable state. A very important feature of the *TPM* is that this module can be uniquely identified. It is equipped with a so called Endorsement Key Pair, which is unique. The private part of this key pair is never released from the *TPM* which ensures the identification.

TPM based Authentication

The main problem causing the attacks mentioned above is that user credentials are intangible. This can be solved using trusted devices, but this is usually not very comfortable as most of them are extern devices, which have to be used in addition to a computer. Therefore, we propose to use the *Trusted Platform Module*, which is included in many modern computing devices. Furthermore, there are two fallback algorithms in case, the user is not able to use a computer equipped with a *TPM*.

Before the authentication takes place, an offline SSL handshake has to be done. We propose to do this offline, since online is risky by technology (especially for unexperienced users). Such an offline handshake starts with a CD-ROM sent by the e-commerce provider, containing the following:

1. the e-commerce' provider's public key with a piece of software sealing the key in the *TPM* of the clients machine,
2. a piece of software for client key generation and printing the client's public key's fingerprint[1], sealing the client's private key in te *TPM*, and sending the client's public key to the e-commerce provider,
3. the e-commerce software using user credentials *and* a mutually authenticated challenge-response protocol where verification of the merchant's authentication information and computation of the client's authentication information is done by the *TPM*.

Now, we can start with the authentication itself.

Authentication using the Registered Home Machine In general the authentication protocol is based on a challenge response protocol. The client starts the session with a connect message including a challenge. The challenge has to be generated by the client's *TPM* and verified by the server. In case, the verification was successful, the server sends back the response and its own challenge, which

[1] The printed fingerprint will be sent to the merchant by registered mail for key verification.

will be verified by the client's *TPM*. If the server receives the client's response, calculated by its *TPM*, the normal user authentication can start.

It is clear, that all the messages sent during client authentication have to be encrypted and include time stamps to avoid replay and *MITM* attacks.

Authentication using a Remote Machine and a Registered Mobile Device Even though most users use most of the time their "home machine", they will infrequently access their on-line account from remote, unknown and therefore unregistered machines. If the user is in possession of a mobile device, and the e-commerce application can communicate with the mobile device for instance via MMS or WLAN, then the user can also register his/her mobile device with the bank. Furthermore, the mobile device must have a private and a public key, where the latter is known by the e-commerce server. Then the mobile device can act on behalf of the home machine's *TPM*. The only difference is that the server will communicate directly with the mobile device, which was not possible with the *TPM*. The first connect message is again sent by the client, but the challenge response protocol will take place only between server and mobile device. If this protocol succeeds, the user authentication between server and user will go on.

As the server communicates directly with the mobile device using encrypted messages, this solution does not enable a *MITM*. But as the connection between the user's computer and the server is not really protected, this one is vulnerable to *MITM*. To reduce the risk, we require the user first to confirm the authentication process on the mobile phone and later all transactions. This way, there might be a man-in-the-middle, but he cannot cause harm as the user had to confirm it. In case, there is no user confirmation, an attacker could use formerly phished user credentials on an unknown computer and the mobile phone would acknowledge silently everything. Furthermore, the attacker would be able to act as *MITM* and cause harm on the user's account.

Authentication with Unregistered Devices As not all users will possess a mobile device, which can store keys and execute additional programs, bypassing solutions must be available. We focus here on a solution which still requires a mobile phone, which is able to receive SMS (but nothing else), for ensuring strong authentication. The strong authentication relies on the authentication in mobile networks (e.g. GSM). The last solution is "clumsier" than the first two, but by being a little more complex ensures that phishing attacks still will not be possible.

The authentication steps are as follows (it is assumed that the mobile phone number of the user was registered securely with the e-commerce provider): First of all, the users tries to connect to e-commerce server, which then sends an one time password (OTP) to the user's mobile phone. The OTP has to be send to the server using the unknown machine in addition to the user's "normal" credentials. As this mechanism does not really provide prevention against proxy *MITM* attacks, the server has to send another OTP plus the transaction details to the user's mobile phone, if a transaction will be done. The user then has to acknowledge this transaction with the OTP. Using such a mechanism, the user will be notified about every transaction on her account, which renders proxy *MITM* useless.

Conclusion and Outlook

The proposed authentication scheme supports mutual authentication and some kind of two channel authentication using *Trusted Platform Modules* or mobile devices. Using the proposed mechanism, attacks like *Phishing*, *Pharming*, and *MITM* attacks on the pure user credentials are still possible, but useless.

Now, the whole authentication algorithm has to be implemented and evaluated in a real world environment.

Remark

The work described in this report has been published in a more detailed version at CSNA 2007 in Beijing/China [13].

Bibliography

[1] A. Herzberg, and A. Gbara, "Protecting (even) naïve web users, or: preventing spoofing and establishing credentials of websites," 2006.

[2] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *11th Annual Network and Distributed System Security Symposium (NDSS '04)*, (San Diego, USA), February 2004.

[3] S. Gajek, J. Schwenk, and C. Wegener, "SSL-VA-Authentifizierung als Schutz von Phishing und Pharming," in *Sicherheit*, pp. 6–17, 2006.

[4] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger Password Authentication Using Browser Extensions," in *Proceedings of the 14th Usenix Security Symposium*, 2005.

[5] R. Oppliger, R. Hauser, D. Basin, A. Rodenhaeuser, and B. Kaiser, "A Proof of Concept Implementation of SSL/TLS Session Aware User Authentication (TLS-SA)," in *Kommunikation in Verteilten Systemen*, pp. 225–236, Springer Verlag, 2007.

[6] S. Gajek, A.-R. Sadeghi, C. Stüble, and M. Winandy, "Towards Multicolored Computing - Compartmented Security to Prevent Phishing Attacks," in *Workshop on Information and System Security (WISSEC'06)*, (Antwerpen, Belgium), 2006.

[7] B. P., C. Kuo, and A. Perrig, "Phoolproof Phishing Prevention," *Financial Cryptography*, 2006.

[8] "Trusted Computing Platform Alliance. Main Specification Version 1.1b,", The Trusted Computing Group, 2003.

[9] "TrustedGRUB, available online: http://sourceforge.net/projects/trustedgrub," University of Bochum, 2007.

[10] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design and Implementation of a TCG-based Integrity Measurement Architecture," in *Proceedings of the 13th USENIX Security Symposium*, 2004.

[11] "BitLocker Drive Encryption: Executive Overview, available online: http://technet.microsoft.com/en-us/windowsvista/aa906018.aspx," 2007.

[12] "Standardizing Network Access Control: TNC and Microsoft NAP to Interoperate," Trusted Computing Group and Microsoft, May, 2007.

[13] C. Latze and U. Ultes-Nitsche, "Stronger Authentication in E-Commerce: How to protect even naïve Users against Phishing, Pharming, and MITM attacks," in *Proceedings of the IASTED International Conference on Communication Systems, Networks and Applications*, pp. 111–116, (Beijing, China), October 2007.

Visualizing Forensic Data by means of Semantic Layering

Michael Hayoz, University of Fribourg
michael.hayoz@unifr.ch

Introduction

The horizon of security threats widens by the day, as technology along with its emerging possibilities evolves at a faster pace than ever. The ever growing need for security in Information and Communication Technologies (ICT) is ubiquitous and demands a high degree of awareness from both users and professionals.

The field of Computer Forensics has been making inroads over the past ten years, detaching itself from the broader area of IT security to become a self-contained scientific discipline of its own. Many forensic tools have been developed since the early days of Computer Forensics, mostly within the Open Source community, with the goal to reduce the complexity of the process of forensic analysis and to provide specialists with a set of auxiliary tools. Nonetheless, the professional application of Computer Forensics still finds itself in the fledgling stages and offers a huge potential for research and improvement. Experience shows that for different reasons, many so-called IT Forensics specialists lack the appropriate know-how when it comes to security incidents. In most cases common mistakes are made during the first steps of Incidence Response, which eventually result into alteration of a suspect system's state. This inevitably leads to the loss of unrecoverable evidence.

This paper describes an attempt to optimize and simplify the process of forensic analysis for security professionals and forensic specialists by means of visualization and *Semantic Layering*.

Overview

The first part of this paper will briefly introduce the field of Computer Forensics and give an overview of the process of forensic analysis with focus on the Post-Mortem analysis, which sets up the context for our approach. The second part will emphasize the challenges of Computer Forensics and the issues they induce. The third part will motivate our approach and describe the notion of Semantic Layering and briefly allude the concepts behind our idea. The concluding fourth part will outline ongoing and future work of this project.

Part I : Computer Forensics

Computer Forensics, IT-Forensics or Digital Forensics is considered as a branch of Forensic Science. It is concerned with the identification, the safekeeping and the analysis of suspect data, a process which eventually results in the processing and presentation of the evidence in a court of law. One among several sources defines the notion of Computer Forensics as follows: *"Computer Forensics is simply the application of computer investigation and analysis techniques in the interests of determining potential legal evidence."* [1] Computer Forensics aim at gathering intelligence about the method(s) and weak point(s), which might have lead to a system intrusion. The investigator wants to detect the originated damage after such an intrusion and trace or even identify the attacker(s).

The process of forensic analysis

The process of forensic analysis describes a sequence of steps to follow in any forensic investigation. We divide this process into five phases:

1. **Covering the site of crime** : concentrating on seized hardware is not enough, the surroundings of a crime scene can reveal the missing part in the puzzle (print-outs, mail, notes, newspaper articles etc.).
2. **Immediate safeguarding of volatile data on any running system** : gathering as much critical information as quick as possible without leaving one's own footprints, i.e. without changing the suspect system's state.
3. **Forensic duplication** : bitwise duplication / imaging of all connected storage drives.
4. **Post-Mortem analysis** : systematic analysis of forensic duplicates.
5. **Consolidation of the results of the investigation** : bringing all evidence into a causal and chronological correlation. Drawing appropriate conclusions to allow plausible and reproducible chains of activity.

The Post-mortem analysis

As aforementioned, the Post-Mortem analysis sets up the context for our approach. This phase of the forensic analysis process consists of a systematic examination of the data gathered during phase three of the forensic analysis. The Post-Mortem analysis is always conducted offline. The investigator works on copies, never on the original data and performs the examination within a safe environment. This requires the analysis environment to be protected from unauthorized access and locally saved results and notes to be encrypted. The investigator might be subject to an attack himself, a situation which might put him in a delicate position. Using forensic duplicates allows the investigator to work pressure-free without having to worry about destroying evidence.

Data of interest during the Post-Mortem analysis are File-Slacks, MAC times, Swap- and hibernation files, log-file entries, system protocols, hidden- and deleted files, to name a few. It is of the utmost importance that the forensic tools being used for investigation be reliable and their pristine state be undoubted. The same rule holds for system files used when collecting volatile data from a running system. Investigators always use their own trusted and statically compiled system files in order to avoid the use of libraries on the compromised system.

Part II : Challenges and issues

It is not very hard to collect data from storage devices. Current forensic toolkits provide the necessary means which partly automate this process. The majority of these tools offer a large set of filters and sorting algorithms which narrow down the data to be analyzed to the small percentage of relevant data which eventually might or might not lead to some valuable evidence.

The biggest challenge lies in the ability to correlate collected evidence with other events. This is where the human factor comes into play. Forensic investigation tries to draw coherent and plausible conclusions about former system states. Furthermore, a forensic analysis is considered to be a time-critical process. External circumstances define the available time for the investigation. It is very likely that those external factors will have an impact on how fast supportive evidence must be found in a case. Investigative authorities often lack the required qualified

human resources, which puts even more pressure on the team responsible for the investigation.

Part III : Visualizing forensic data by means of Semantic Layering

Motivation

Our approach builds upon the following assumptions:

- as seen before, forensic analysis is a time-critical process
- most security incidents involve more than just one storage device, this quickly increases the amount of data to analyze
- the bigger the data volume, the harder it gets to know where to start with the analysis
- the meaning of data is context-sensitive; the closer data gets to its physical nature, the more it loses of its initial meaning
- common forensic analysis procedures examine suspect storage devices in sequential fashion which is inefficient and time-consuming
- the output of current forensic tools mostly comes in the form of exhaustive tabular lists; looking for evidence is like seeking the needle in the haystack
- studies show that the human brain is more efficient when working with visual aids

We now use these assumptions to frame the basic idea of our approach. The results of current forensic tools are only partially structured and will serve as an input to our system and represent what we call *primitive entities*. Considering the fact that data loses its initial meaning while being processed top-down towards an electrical signal, we will use a bottom-up approach to give these primitive entities some new meaning through aggregation. To put it short, primitive objects can be aggregated to form new, more complex objects and will hence obtain additional semantics. This process of graphically combining and recombining entities allows us to create new views or data structures our suspect data can be searched against (e.g. by means of logical expressions, some IP addresses with an E-Mail address etc.). We call this projection of views on sets of suspect data the process of *Semantic Layering*.

Our system does not aim at delivering forensic evidence, this would be impossible. Its purpose is to deliver hints as to where it would make sense to start looking for evidence among the data to be examined. Our approach will simplify the process of forensic analysis during the Post-Mortem phase and will hopefully bypass the less efficient and time-consuming process of sequentially examining one storage device after another through the use of *Cross-Drive examination* (as mentioned in the following section).

Ongoing and future work

Our approach is in its early stages. The following outlines ongoing and future work in this project:

Getting in touch with the forensic community / authorities :

- Find out about current infrastructure, methods of investigation
- Get to know about common problems, difculties (requirements analysis)

- Define feasible improvements and gather reactions, inputs to our approach

Forensic data material :

- setting up a forensic test environment
- getting forensic sample data to study possible input for our approach

Elaborate applicable techniques :

- Efficient technique of visualization
- Approach to realize semantic layering (correlation / aggregation)
- Have a look at common data mining techniques

An additional milestone would be to introduce the ability of what we call *Cross-Drive examination*. This technique would apply our semantic views to a set of storage devices in parallel and significantly increase the performance of our system and address the time issue mentioned in the second part.

Bibliography

[1] "Computer Forensics and Computer Expert Witness Services, available online: http://computerlegalexperts.com," 2007

[2] S. L. Garfinkel, "Forensic feature extraction and cross-drive analysis," in *Digital Investigation 3S*, pp. 71–81, 2006

[3] D. Farmer, and W. Venema,"Forensic Discovery," Addison Wesley, 2005

[4] A. Geschonneck, "Computer-Forensik. Systemeinbrueche erkennen, ermitteln, aufklaeren," Dpunkt Verlag, 2. aktualis. A., January 2006

[5] B. Carrier, "File System Forensic Analysis," (Amsterdam, NL), Addison-Wesley Longman, April 2005

[6] "Wikipedia, available online: http://en.wikipedia.org," 2007

[7] "Computer Forensics World, available online: http://www.computerforensicsworld.com," 2007

[8] "Open Source Digital Forensics, available online: http://opensourceforensics.org," 2007

[9] "HELIX, available online: http://www.e-fense.com/helix," 2007

[10] "F.I.R.E., available online: http://fire.dmzs.com," 2007

[11] "The Sleuth Kit, available online: http://www.sleuthkit.org/proj.php," 2007

[12] "Forensics Wiki, available online: http://www.forensicswiki.org," 2007

Immune based Intrusion Detection System

Christoph Ehret, Univeristy of Fribourg
christoph.ehret@unifr.ch

Introduction

Intrusion detection systems are nowadays very important for every IT company which is concerned with security and sensitive systems. Even if a lot of research was already done on this topic, the perfect IDS was still not found and it stays a hot and challenging area in computer security research. Recently a new approach started to make its way to intrusion detection, namely the immune system. It has a lot of interesting features we would like to find in an IDS. A new artificial intelligence paradigm was created from the immune system, namely the artificial immune system; this paradigm is rather new comparing to neural networks or fuzzy logic, but it is very promising for different areas in computer science.

We will make a little overview of the immune system followed by a brief introduction to the artificial immune system. In the last section we will describe the common design of the intrusion detection systems and finish with an outlook.

The immune system

Overview

The human immune system (HIS) is quite complex and elaborated. The defense of the HIS is organised in different layers, mainly the exterior defenses, which are biochemical and physical barriers like for example skin or bronchi, the physiological barrier, where pH and temperature provide inappropriate living conditions for pathogens, the innate system and finally the adaptive system. Every layer has different defense mechanisms and stops different types of pathogens. The innate and adaptive systems are again divided into several different cells. Every leucocyte has very specific functions, like for example the Neutrophil [2] which migrates to sites of inflammation or infection and ingests microorganisms or particles, destroys them and dies, or the Eosinophil [3] which is responsible to combat parasites and is the main effector in allergic responses and in asthma. The B- and T-cells are the actors of the adaptive system; they are responsible to detect yet unknown pathogens, produce the specific antibodies and destroy them. Every B- and T-cells have different detectors, called epitopes, that interact with different kind of pathogens. In order to improve the diversification, new B- and T-cells die and are created with randomly generated receptors every day, what modifies continuously the set of possible detected pathogens. There is a great interaction between all the different cells of the HIS; some immune cells secrete special substances that will attract some other type of immune cells, or some are responsible to produce an inflammation what will allow more immune cells to reach this particular region.

For more information on the different leucocytes and their role within the HIS consulte [6].

Tolerization and activation cycle

The adaptive system plays with the B- and T-cells a crucial role to detect, fix and kill yet unknown antigens. In order to avoid the lymphocytes [4] to recognize the

[2] The Neutrophil constitutes the majority of blood leucocytes and is part of the phagocyte cells
[3] The Eosinophil constitutes 1-5% of blood leucocytes
[4] The lymphocyte is a type of white blood cell among which we find B- and T-cells

own cells as foreign cells or *intruders*, they need to be *trained* to recognize *self* from *non-self*. The lymphocytes are called negative detectors because they are trained to bind to *non-self*, what means that they bind to foreign cells; this form of learning is called *tolerization*.

T-cells for example are tolerized in the thymus[5] where most of the self proteins are found. In this way T-cells only interact with self-cells, cells that they should not recognize as intruders. T-cells that are activated during the tolerization phase, what means they have recognized a self-cell as an intruder, are called immature and die through programmed cell death. The T-cells that survive during the training or recognition phase will be tolerant to all the self-proteins.

Once lymphocytes are mature there exist different mechanisms to minimize the risk that they attack the cells of the own body. As soon as a lymphocyte binds to a protein, the activation cycle starts; if the cycle finishes, the protein is recognized as an intruder and the lymphocyte is activated what will start an immune response. The first step of the cycle is that a certain number of the lymphocyte's receptors must bind to a pathogen; the second step is that the binding must occur in a certain period of time. If one of these steps or conditions fails, the lymphocyte will not be activated and the protein will not be recognized as an intruder. Once a T-cell has completed the recognition cycle it needs a co-stimulation signal from other immune cells before it is activated, what means that other immune cells also recognized the checked cell as a pathogen. The lymphocyte, actually a B-cell, that has the most effective antibodies against a given pathogen will become a memory cell; when this pathogen, or one structurally similar, is later again encountered, the immune response will be much quicker and effective. It is interesting to notice that there is a kind of competition between the lymphocytes to become a memory cell, what improves the effectiveness of the immune response.

Why a rich source of inspiration?

As we have seen, the immune system is a complex but very powerful system (within an organism) that detects and kills pathogens. *"It is hard to find another biological system that embodies such a powerful and diverse set of features"[5].* It is therefore normal to take the immune system as a source of inspiration to create computational models. There are a lot of features we would like to use, like for example pattern recognition, anomaly detection, adaptiveness (learning and memory), fault tolerance, distributivity and autonomy, or diversity, to name only a few; a more complete list can be found in [5] and [1].

The artificial immune system

We can find quite different definitions of an artificial immune system (AIS) in the literature, like in [5] which enumerates definitions from different authors. One possible definition could be "*Artificial immune systems (AIS) are adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving*" (de Castro and Timmis, 2002).

The artificial immune system paradigm is rather young comparing to other artificial intelligence paradigms like Neural Networks, Fuzzy Logic or the genetic algorithms. AIS began in 1986 with Farmer, Packard and Perelson's paper on immune networks[8], but there was only in the mid-90's that it kept the attention of scientists with, among others, the works of Forrest et al.[4, 2], Dasgupta, Timmis and de Castro.

What do we need if we want to implement an AIS framework? If we abstract the immune system in a simplistic way we have a population of different types of immune

[5]The thymus is an organ located just behind the sternum and is the reason of the "T" in *T*-cells

cells and interactions between them through receptors. For our AIS we therefore need to have a population, defined as a set, a way to describe each element of the set, its length, and a way to measure an interaction. To describe the population we will use the concept of *shape space* (S); it is used in immunology to quantitatively describe the interactions between immune cells and antigens. An element of S is described by a set of N_p parameters (length, width, charge, ...). To cover the whole shape-space, we actually need to generate $N = k^L$ different elements, where k is the size of the alphabet, L the length of one element of the set, and N is called the *potential repertoire*. As we have seen, one antibody can detect pathogens with similar structure, i.e it is not bound to only one specific pathogen[6]. For that we will introduce the notion of *coverage*; $C = \sum_{i=0}^{L-\epsilon} \frac{L!}{i!*(L-i)!}$ gives us the number of antigens covered by one antibody, where L is the string length of the antibody and ϵ the *cross-reactivity threshold*[7]. The minimum elements necessary to cover the shape-space S is therefore given by $N_m = \text{ceil}(\frac{N}{C})$, where N is the potential repertoire and C the coverage. The interaction, i.e the affinity between an antibody and an antigen, both of length L, is evaluated with a distance measure between their attribute strings S^L x $S^L \rightarrow \Re^+$. To measure the distance, the Euclidean, Manhattan or Hamming distance functions are often used. Finally, the training phase is often done using the negative selection[4] improved sometimes with some genetic algorithms.

The application domain of AIS is becoming quite large. It is used for example in computer security, data analysis, search and optimization methods, agent-based systems, or autonomous navigation and control systems. Much more information and details on AIS can be found in [5].

Intrusion detection systems

The goal of an Intrusion detection system (IDS) is to detect if an intrusion [3] has occurred in computer systems. We can compare it to a house alarm system that rings as soon as someone breaks into the house.

There are two different approaches on where to install the IDS, namely a host based IDS (HIDS) or a network based IDS (NIDS). Each approach has its advantages and disadvantages. The NIDS can cover a whole subnet of systems, is OS independent and stealthy, identifies network layer errors and is not really prone to DoS attacks, but it cannot analyze encrypted traffic, has performance problems with high bandwidth, does not know if an attack was successful or not, has no information of the destination hosts OS, nor can it detect fragmented attacks. On the other hand, the HIDS has the knowledge of the host (OS and running services), can analyze encrypted and fragmented packets, and can associate a user with an event, but it requires resources of the host (even if this should not be shown as a negative point as we will discuss it in the next section); if a DoS attack was successful on a host the IDS may also be down, or if an intrusion was successful the HIDS is not reliable anymore. To minimize the disadvantages of both we can build a hybrid detection system that is a mixture of both approaches.

The IDS comes also with two different detection strategies, one misuse based and the other anomaly based. The misuse detection alternative is based on rules, what means that it is a static solution and it cannot detect yet unknown attacks, but it has a low false positive/negative rate. The anomaly based strategy tries to detect anomalies on a system that could be due to an intrusion, what means that it is a dynamic solution and is able to detect yet unknown attacks, but it often has

[6] Imagine the number of antibodies we would need if each could detect only one given pathogen

[7] The cross-reactivity threshold characterizes the fact that each antibody interacts with all antigens whose complement lies within a small surrounding region

a rather high false positive/negative rate. Both solutions actually still need human intervention to analyze the intrusions found by the IDS.

If we compare the IDS paradigm with the immune system, we can make quite some comparisons, like the adaptive system with the anomaly based detection strategy or the innate system with the misuse detection alternative. In fact, we can even further improve the IDS paradigm thanks to ideas from the immunology, for example to lower the false positive rate of the anomaly detection.

Conclusion and Outlook

The immune system is complex but very powerful; it can detect a lot of different types of pathogens, even unknown one, and thanks to a strong interaction between all the different actors of the immune system the pathogens can be destroyed. As the immune system has some very interesting features, a new artificial intelligence paradigm called the artificial immune system was created from it. Computer security, especially the antivirus and IDS fields, is of course an interesting candidate to apply AIS. The immune system itself is actually a very interesting approach to intrusion detection[1].

In a future work, we will try to implement a hybrid system that will be mostly a HIDS, like the immune system which is active in the body and not outside of it, and integrate both misuse and anomaly detection, like the immune system with the innate and adaptive system. To minimize the false positive rate of the anomaly detection, we will introduce a co-stimulation mechanism and some activation cycle similar to the T-cell. When a human being is ill he will stay in the bed and all his resources will be used to recover; this is another idea we will try to apply to our IDS model. Diversity, distribution and interaction with other systems should also find a place in an IDS model based on the immune system.

Bibliography

[1] U. Aickelin, J. Greensmith, and J. Twycross, "Immune System Approaches to Intrusion Detection - A Review," in *Proceedings ICARIS-2004, 3rd International Conference on Artificial Immune Systems*, LNCS 3239, pp. 316–329, Springer Verlag

[2] S. Forrest, S. A. Hofmeyr, and A. Somayaji, "Computer Immunology," in *Communications of the ACM*, March 1996

[3] J. M^{c}Hugh, "Intrusion and intrusion detection," in *International Journal of Information Security*, vol. 1, pp. 14–35, Springer Verlag, July 2001

[4] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, "Self-Nonself Discrimination in a Computer," in *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, 1994

[5] L. N. de Castro, and J. Timmis, "Artificial Immune Systems: A New Computational Intelligence Approach," Springer Verlag, 2002

[6] I. Roitt, J. Brostoff, and D. Male, "Immunology," sixth edition, Mosby, 2001

[7] M. Burgess, "Computer Immunology," in *LISA '98: Proceedings of the 12th USENIX conference on System administration*, pp. 283–298, USENIX Association, 1998

[8] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," in *Physica D*, vol. 2, pp. 187–204, 1986

Distributed Systems

Federation of Experimental Networks for Teaching and Research

Torsten Braun, Univeristy of Bern
braun@iam.unibe.ch

Introduction

Experimental networks are required for both teaching and research. Students and researchers need a playground for performing experiments, because mistakes or unexpected effects during an experiment that is performed for education or research purposes should not harm the productive networks and traffic. While students may perform configuration errors causing unexpected effects, in case of research activities, new non-standard protocols might be investigated and evaluated. Research with new protocols might also result in unexpected network behaviour.

Experimentation Networks Today

Local Test-Beds

In the past, network research has mainly used local or rather statically configured distributed test-beds. Local test-beds are limited to experimentation facilities of a single organization and are under full control of this organization. They are rather limited in terms of number of nodes and geographic size. In most cases, local test-beds are set up in a local experimentation laboratory. In that case, these experimentation networks are not representative in terms of wide area network characteristics such as delay and traffic load. The lack of wide-area connections does not allow making meaningful experiments for protocols that should work in a global network like the Internet, since many protocols heavily depend on characteristics such as (long) round trip times. Local test-beds require significant efforts to maintain and control them. It is rather easy to protect them from unauthorized access via the Internet, but they are not open to the general public then.

Distributed Test-Beds

In contrast to local test-beds, geographically distributed test-beds allow more realistic experiments when they are interconnected via the Internet. An advantage is that system and network resources can be shared among institutions. In this case, the management and financial effort to run a large experimentation network can be distributed to the involved organizations. The effort per organization can be decreased compared to a situation when a single organization is running and operating a large test-bed of the same size. However, in this case unauthorized access to the shared resources via the Internet must be prohibited and controlled access is required. Many research activities have urgent need for distributed wide-area experimentation networks. This has been achieved by mainly two approaches:

1. Local test-beds have been interconnected by dedicated links or tunnels. In particular, several European research projects followed this approach. One example is the EuQoS project [2]. The resulting distributed experimentation network is rather isolated from others and only a limited set of partners are interconnected by such an approach. Only this limited set of partners can then typically use the experimentation facilities. When using dedicated links, traffic is isolated from other traffic, which may have advantages (no disturbance of

experiments by other traffic) and drawbacks (no cross traffic as it is typical in the Internet).

2. An example of a popular distributed experimentation network with shared resources is PlanetLab [1]. Research organizations (in particular universities) are connected to this world-wide network. PlanetLab is based on interconnecting slices, where a slice is a set of interconnected virtual machines. Several virtual machines are running on a single computer system. The computer systems are provided by different organizations participating in PlanetLab and interconnected mainly by IP-IP or MPLS (Multi-Protocol Label Switching) tunnels. PlanetLab can be considered as an overlay network with rather central control. The involved organizations provide the computer equipment, which is configured and maintained by a central management operation center. This reduces the flexibility and the possibility to bring in new network resources easily. Access to PlanetLab is limited to organizations providing equipment and does not support short-term experiments by organizations that do not need permanent PlanetLab involvement. Moreover, overlay networks such as PlanetLab are not ideal for network level protocol research, e.g., IP level and routing protocols, since the protocols under test run as user space processes and not on genuine router systems. This clearly distorts important aspects of real protocol behavior. By June 2007, 800 nodes at 400 sites have been connected to PlanetLab.

Federated Test-Bed Activities

A federation of test-beds is an interconnection of two or more independent and autonomous test-beds for the creation of a richer environment for testing and experimentation, and for the increased multilateral benefits of the users of the individual test-beds. In a federation, geographically distributed testbeds are owned by different organizations. Federations should be rather open and dynamic and evolve over time based on user requirements. Two activities, one in the USA (GENI) and one in Europe (FIRE), are currently going into this direction.

Global Environment for Network Innovations (GENI)

GENI is a US initiative funded by the National Science Foundation and goes somewhat into the direction of test-bed federations, but addresses a rather centralized and less open approach [3]. BBN Technologies, which has already implemented the Integrated Message Processors for the ARPAnet in 1968, has recently been selected to manage the planning and design of an US advanced network facility spanning. GENI extends the concept of slices to lower layers. A slice is defined as a set of slivers, while a sliver is a piece of a component resulting from virtualization (e.g., virtual router) or partitioning. Slivers access the network via virtual interfaces, which might be a socket interface or a virtual wire interface, e.g., realized by a dedicated wavelength. Programmable core and edge routers shall support the flexible installation of novel protocols. A management framework for resource allocation, monitoring, reboot etc. has been defined by the GENI Management and Control architecture, which is hierarchically structured and closely oriented to the centralized PlanetLab management. Therefore, it reflects a rather centralized management approach and the openness and establishment of federations seem to be rather limited.

Future Internet Research and Experimentation (FIRE)

The FIRE programme has been initiated by the European Commission in the Seventh Research Framework Programme (FP7) [4]. Within the research challenge on "Pervasive and trusted network and service infrastructures" of the ICT theme of FP7, the European Commission has launched a call for proposals on the research objective for "New paradigms and experimental facilities". This research objective has two dimensions. The first one addresses advanced networking approaches to architectures and protocols, designed to cope with increased scale, complexity, mobility and requirements for security, resilience and transparency of the future Internet. These approaches shall be validated in large scale testing environments based on a combination of physical and virtual infrastructures. The second dimension focuses on interconnected test-beds addressing novel distributed / reconfigurable protocol architectures, novel distributed service architectures, infrastructures and software platforms as well as advanced embedded or overlay security, trust and identity management architectures. Concrete research projects funded by the European Commission are expected to start in 2008.

Test-Bed Activities at University of Bern

At the University of Bern, several activities on distributed or federated test-beds are going on. Within the EuQoS project, a distributed test-bed based on GRE (Generic Routing Encapsulation) tunnels have been set up manually among the project partners. Various protocols and architectures resulting from the EuQoS research activities have been tested and evaluated using the resulting distributed test-bed.

Within the different e-learning projects at University of Bern such as VITELS (Virtual Internet and Telecommunications Laboratory of Switzerland) [5] and OSlab (Operating Systems Laboratory) [7] several components required for federated test-beds have been developed. These are in particular a management system for resource reservation of several entities of a single type of resource. The Laboratory Portal Server includes a firewall, authentication, authorization, and accounting (AAA) functions, and facilities for management of equipment, e.g., reset [6]. The various e-learning modules form a kind of distributed test-bed. Access control is achieved by the authentication and authorization infrastructure coordinated by SWITCH [8].

A new project in the area of federated test-beds will start in 2008 at University of Bern. The project Wisebed (Wireless Sensor Network Testbeds) has been accepted within the FIRE research programme by the European Commission. University of Bern will implement together with eight other European partners - mainly universities - a network of heterogeneous sensor network testbeds to support experimental research in wireless sensor networks.

Bibliography

[1] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: an overlay testbed for broad-coverage services," in *SIGCOMM Computer Communications Review*, vol. 33, no. 3, pp. 3–12, July 2003.

[2] X. Masip-Bruin, M. Yannuzzi, R. Serral-Gracia, J. Domingo-Pascual, J. Enriquez-Gabeiras, M. A. Callejo, M. Diaz, F. Racaru, G. Stea, E. Mingozzi, A. Beben , W. Burakowski, E. Monteiro, and L. Cordeiro, "The EuQoS system: a solution for QoS routing in heterogeneous networks," in *IEEE Communications Magazine*, vol. 45, no. 2, pp. 96–103, February 2007.

[3] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet Impasse Through Virtualization," in *GENI Design Document 05-01*, April 2005.

[4] "New Paradigms and Experimental Facilities, available online: http://cordis.europa.eu/fp7/ict/fire/," December 2007.

[5] T. Braun, and M.-A. Steinemann, "The Virtual Internet and Telecommunications Laboratory of Switzerland," in *ACM SIGCOMM 2003 Workshop on Networking Education*, (Karlsruhe, Germany), pp. 2–3, August 2003.

[6] S. Zimmerli, M.-A. Steinemann, and T. Braun, "Resource Management Portal for Laboratories Using Real Devices on the Internet," in *ACM SIGCOMM Computer Communication Review*, vol. 33, July 2003.

[7] M. Wulff, and T. Braun, "OSLab: An Interactive Operating System Laboratory", in *ERCIM News*, no. 71, pp. 46–47, ERCIM EEIG, October 2007.

[8] M.-A. Steinemann, C. Graf, T. Braun, and M. Sutter, "Realization of a Vision: Authentication and Authorization Infrastructure for the Swiss Higher Education Community", in *Educause 2003*, November 2003.

Self-Organisation in Distributed Systems

Markus Wulff, Univeristy of Bern
mwulff@iam.unibe.ch

Introduction

With the increasing complexity of networks, the management overhead is growing to an extend where manual interaction becomes inefficient. Large distributed systems like peer-to-peer (P2P) networks or wireless sensor networks (WSN) with several hundreds or thousands of nodes cannot efficiently be managed by humans. These systems do usually not have a fixed network topology and the number of nodes might change over time. A global view on the system is in most cases not available. Furthermore, changing environmental conditions may require a re-configuration of the network nodes or changes in the network topology.

A solution to this problem is the use of special algorithms to make the networks self-organising (This includes management, configuration, optimisation and similar tasks.). This article gives a short introduction to self-organisation in distributed systems. First, some basic terms and principles are explained. In the main part the examples of peer-to-peer networks and wireless sensor networks show that self-organisation algorithms must obey different basic parameters in different systems.

Fundamental terms

Distributed systems are collections of interconnected individual nodes that are spatially distributed. Special software and algorithms enable these nodes to appear as a single system that is able to fulfill a given task. The control entity might be central or distributed as well (see Fig. 15).

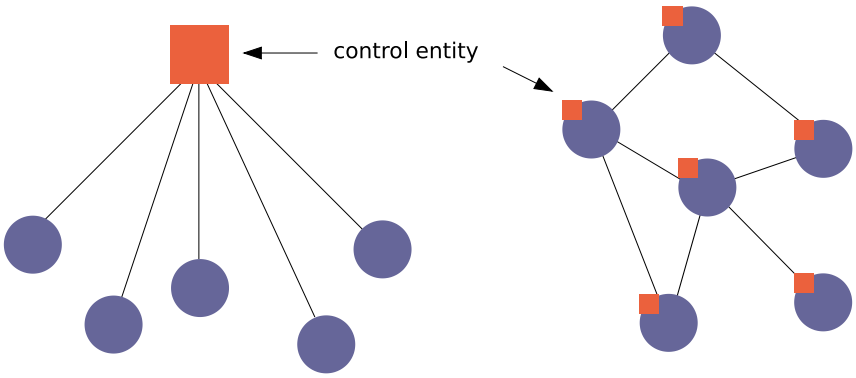

Figure 15: Two basic types of distributed systems.

In the following, only completely distributed systems are considered, which:

- have no central control/coordination
- have no central data base
- have no global view of the system
- consist of autonomous nodes
- nodes and interconnections are unreliable

Self-organisation requires common interests/goals and a shared context for the autonomous individuals/entities. It is assumed that no single unit can achieve the same as the system can. Only by *cooperating* and *communicating* the individuals may act as a community that is able to solve tasks that are too complex for a single entity. While interacting with the environment, positive and negative feedback is obtained. Only then a global behaviour can emerge from local interactions between individuals.

This collective behaviour requires the division and distribution of tasks (work division) and is based on coordination, solution of conflicts, and negotiations. As a basis, cooperation requires communication. this can be direct communication, i. e. one-to-one or one-to-many communication, or indirect communication, i. e. communication by observing other individuals and their influence on the environment (in biology: stigmergy). Accordingly, we distinguish direct and indirect cooperation.

As a result from indirect or direct communication the individual must adapt its behaviour. Figure 16 shows the difference between simple adaption and the application of learning algorithms.

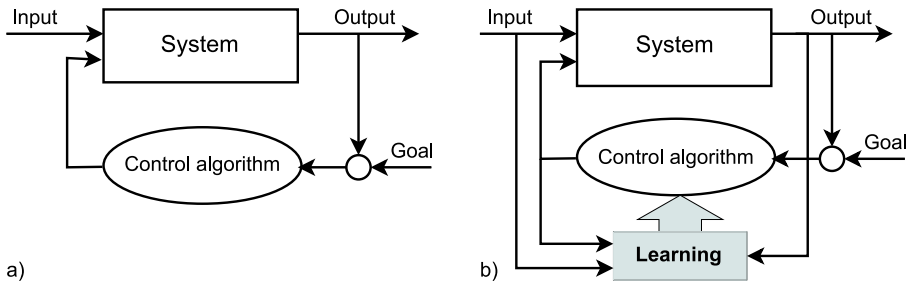

Figure 16: The basic principles of adaption a) and learning b).

Adaption can be achieved by adjusting the parameters for the internal algorithm of the individual depending on the output values. If a learning system is applied, input and output parameters can be used to optimise the system's control algorithms. One learning approach is reinforcement learning. Here, the algorithm learns how to respond to observations of the environment. Every action of the individual has some impact in the environment, and the environment provides feedback that guides the learning algorithm.

Self-organisation in P2P networks

Peer-to-Peer (P2P) systems are decentralised systems built by a set of peers forming a loosely coupled network through the knowledge of some other peers of the system kept locally on the nodes. Such systems, mostly known from file sharing applications, propose to solve some problems of classical client/server systems. In a P2P system, every peer (normally composed of a user and a certain machine represented by its IP address) can request and provide services.

Generally speaking, the functionality of every participant (node) is equivalent in such a completely decentralised P2P system. Since the presence of a centralised coordinator is eliminated, the problem of lookup can be encountered, which is stated by Balakrishnan et al.: "given a data item X stored at some dynamic set of nodes in the system, find it". This is a critically ordinary problem in P2P systems [1].

Several algorithms have been developed to solve the problem of locating content and services in P2P systems by building a topology which supports the search.

Based on this used algorithms, the P2P systems are categorised into unstructured and structured systems. Unstructured P2P systems, define no constraint of connection between nodes. In general, their lookup algorithms are simple (and mostly based on an expensive broadcast of the search queries) and the network's dynamism involves a low cost of maintenance. However, the lookup algorithms of these systems are non deterministic, i. e., lookup for a resource may fail even if the resource exists somewhere in the network. Structured P2P systems, like Distributed Hash Table (DHT) systems, define constraints of connection between nodes. Such systems can be regarded as a hash table distributed on the network's nodes, each of which is responsible for a chunk of the table.

Self-organisation in P2P systems is not limited to build and maintain the network topology. But it is on major aspect in those systems.

Self-organisation in WSN

Wireless sensor networks (WSM) are comparable with P2P systems in that point that they are also built from a high number of individual nodes, the sensor nodes. However, the development of self-organisation algorithms for these devices must obey the special requirements of WSN. Sensor nodes usually have a number of restrictions compared to computers or work stations. The nodes are usually very small and run on (very limited) battery power. This and the fact that sensor nodes in most cases have a very limited memory, computing power, and a small communication range demand fitted algorithms for WSN.

In Fig. 17 the typical setup of a "traditional" WSN is shown. The network can be accessed through a gateway node and configured by a management station.

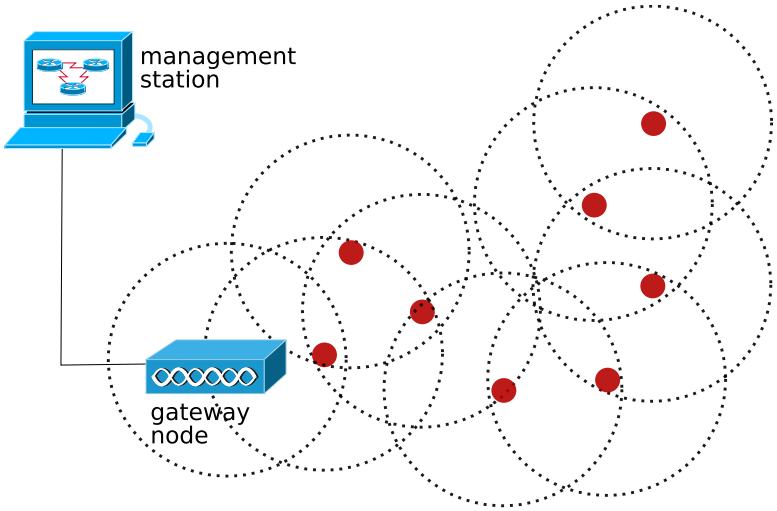

Figure 17: Example of a wireless sensor network with a central management station. The dashed circles denote the sensor node's radio range.

This approach becomes inefficient if a high number of sensor nodes is used or if the configuration must frequently be changed. The goal of self-organising WSN is to achieve high throughput and low latency communication, redundancy for recovery, or improved resource and energy management. The traffic flow inside the WSN can be improved by backbone formation, transmission management or enhancing routing efficiency. Efficient sensing can be achieved by classification and recognition and improved detection. Furthermore, self-organisation algorithms are

used to create distributed and ad-hoc services [2].

Issues in a WSN environment are the coordination the communication (media access problem), the energy consumption and topology changes. The latter might be caused by node mobility, failures or weather conditions [4].

This results in certain requirements for self-organisation algorithms. As in P2P systems the algorithms can also use only the local knowledge of the node. The limited resources of the sensor nodes require low overhead as well as low bandwidth and memory usage. Further issues are reliability and scalability.

Collaboration between the nodes is necessary for instance for medium access control, time synchronisation and routing.

An example for a self-organising algorithm is a distributed role assignment which enables role bases node update and configuration [3].

A huge number of sensor nodes usually produces a huge amount of sensed data. To communicate all data to a central node would be expensive. The solution for this problem is a data fusion inside the wireless sensor network. By combining information from multiple nodes and transmitting only useful information to a base station, the network load can significantly be decreased.

Summary

Self-organisation in computer systems follows the same rules as in other complex systems (organisms, societies, ecologies ...). Self-organisation is necessary in large distributed systems to enable maintenance, management, scalability, and reliability without or with limited human interaction.

Even though many research has been done on the field of self-organisation is still an important topic which becomes even more important with the growing system complexity.

Bibliography

[1] Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., and Stoica, I., "Looking up data in p2p systems," *Communications of the ACM*, p. 43–48, 2003.

[2] P.K. Biswas, and S. Phoha, "Self-organizing Networks for Integrated Target Surveillance," *IEEE Transactions on Computers*, vol. 55, no. 8, August 2006.

[3] P.J. Marrón, A. Lachenmann, D. Minder, M. Gauger, O. Saukh, and K. Rothermel, "Management and Configuration Issues for Sensor Networks," *Int. Journal of Network Management*, vol. 15(4), 2005.

[4] T.C. Collier, and C. Taylor, "Self-Organization in Sensor Networks," Preprint, December 2003.

Automata Theory

Weakly Continuation Closed Homomorphisms on Automata

Thierry Nicola, Univeristy of Fribourg
thierry.nicola@unifr.ch

Introduction

A major limitation of system and program verification is the state-space explosion problem. There exists several approaches to reduce the state space. Several methods try to keep the state space small during the verification run, other methods reduce the original state space prior to the verification. One of the later are abstraction homomorphisms. *Weakly Continuation Closed* homomorphisms [4] are abstraction homomorphisms preserving exactly those properties of the original behaviour which are satisfied inherently fair. However, the practical use of wcc homomorphisms is limited by the lack of an algorithm, checking whether or not a homomorphism is wcc, which performs reasonably well.

There exists several methods to tackle the state space explosion problem for system verification [1]. The optimal solution is to use combinations of the known state space reducing methods. Abstraction homomorphisms try to reduce the original system behaviour state space to a smaller abstract state space prior to the actual verification run. As the abstract system model is used for the verification, the abstract model should satisfy exactly those properties the original system model satisfies. Weakly Continuation Closed homomorphisms preserve exactly those properties which are satisfied inherently fair [2]. This means that if the verification succeeds or fails on the abstract model, the conclusion to the original system model is immediate and requires no further computations.

A problem of abstraction is that the complete state space has to be computed before abstraction can be applied. This leads to a dilemma, it is not possible to apply the abstraction onto the original state space due to it's vast state-space size. This problem is resolved for wcc homomorphisms, as these abstraction homomorphisms work as well on abstract-compatible trace reductions [2].

Motivation

The purpose of weakly continuation closed(*wcc*) homomorphisms is, as for every abstraction method, to reduce the size of the system state space. The motivation for weakly continuation closed homomorphisms lies exactly in those points which differentiate them from normal abstraction homomorphisms. First of all wcc homomorphisms preserve the inherently fair linear time [3] properties of a system. This means it is possible to do the complete verification(with Inherently Fair Linear Time satisfaction relation) on the abstract system behaviour, as we know the abstract system behaviour satisfies exactly those properties the original behaviour does.

A general problem to abstraction homomorphisms is that the complete state space must be computed before it can be reduced by abstraction. This means that abstraction is only useful if the original state space actually fits in the computer's memory, but does not help if the original state space is too big. Weakly continuation closed homomorphisms are known, to work on a particular partial-order reduction called trace reduction [2]. Partial-order reduction methods are used to reduced the system state space. These methods remove equivalent computations from the state space.

With wcc homomorphisms it is possible to use trace reductions instead of the complete original state space. This allows to verify systems which normally are impossible to be verified as it's state space is too large to fit in the computer's memory. Starting from the trace reduction, and applying a weakly continuation closed homomorphism before the verification promises therefore to be a strong tool for model checking and system verification. Together with inherently fair linear time verification, this verification process might improve the model checking especially if fairness is involved.

The motivation for this paper is the lack of an efficient algorithm for checking whether or not a homomorphism is weakly continuation closed or not. Even though the property of wcc is a decidable, there exists no efficient algorithm yet.

Abstraction Homomorphisms

A *homomorphism* is a function on sequences/strings that substitutes a substring of the original string by a sequence of symbols. Let h be a homomorphism on alphabet Σ, and $w = a_1 a_2 \dots a_n$ a sequence of symbols from Σ, then we have that $h(w) = h(a_1)h(a_2)\dots h(a_n)$. An homomorphism h can be applied on a language L, by applying h to each string $w \in L$, $h(L) = \{h(w)|w \in L\}$.

A special class of homomorphisms are *abstraction homomorphism*. Abstraction Homomorphisms are defined as follows:

Definition 1 *$h : \Sigma^\infty \to \Sigma'^\infty$ is an abstraction homomorphism if and only if the following conditions hold:*

- $h(\Sigma) \subseteq \Sigma' \cup \{\epsilon\}$,
- $\forall v, w \in \Sigma^*, x \in \Sigma^\omega : h(vw) = h(v)h(w)$ *and* $h(vx) = h(v)h(x)$,
- $h(\Sigma^\omega) \subseteq \Sigma'^\omega$

Abstraction homomorphisms are partial mappings since they are not defined for ω-words that would be taken to finitely long words, and that they do not increase the length of the words, cf. $|h(w)| \leq |w|$.

Weakly Continuation Closed Homomorphisms

Weakly Continuation Closed Homomorphisms are abstraction homomorphisms. As in general, abstraction homomorphisms do not preserve the properties of the original system, meaning it is impossible to conclude that when the abstracted behaviour satisfies a property the original system would satisfy the same property too. This fact makes abstraction homomorphisms in general difficult to use for model checking. Weakly Continuation Closed homomorphism have been shown to preserve exactly those properties which are satisfied inherently fair [7]. So weakly continuation closed homomorphism work best with inherently fair linear time verification(IFLTV) [3] and provide a promising tool for model checking purposes when fairness is enabled [6].

The definition of weakly continuation closed homomorphisms is as follows

Definition 2 *h is weakly continuation closed on a language L if and only if, for all $w \in \Sigma^*$, there exists $v \in cont(h(w), h(L))$ such that $cont(v, cont(h(w), h(L)))$ $= cont(v, h(cont(w, L)))$.*

Actually there exists no efficient algorithm able to verify whether or not a homomorphism is weakly continuation closed. There exists some *sufficient* conditions that allow to cover some special cases; e.g. if the automaton representing the behaviour has no more than one strongly connected bottom component, in this case

any h defined on the behaviour will be weakly continuation closed. It is known that the property of weak continuation is decidable. This paper tries to close the gap between the theoretical promises of weakly continuation closed homomorphisms and it's application to the practice of system verification.

In general, we know that $h(cont(v, L)) \subseteq cont(h(w), h(L))$ holds, which implies that $cont(v,h(cont(w, L))) \subseteq cont(v, cont(h(w), h(L)))$ for all $v \in cont(\ h(\ w), h(L))$ holds too.

Homomorphisms on Automata

Let $\mathscr{B} = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be an automaton representing a system behaviour and let $h : \Sigma \to \Sigma'$ be an abstraction homomorphism. Applying the homomorphism h onto the automaton $\mathscr{B}$, results in a new automaton. We will refer to this resulting automaton by $\mathscr{H} = (Q_H, \Sigma', \delta_H, q_H, F_H)$.

The homomorphism h applied on the automaton $\mathscr{B}$, translates the automaton's alphabet Σ to another alphabet Σ'. The homomorphism changes the symbol $a \in \Sigma$ of the transition (q, a, p) of automaton $\mathscr{B}$ to a new symbol $\alpha \in \Sigma'$. The translation therefore becomes (q, α, p) in the automaton $\mathscr{H}$. The automaton $\mathscr{H}$ is identical to the automaton $\mathscr{B}$, except for the translation symbols and the alphabet used. In other words the state set, the initial state and the set of accepting states are equal for both automata, $Q_H = Q_B$, $q_H = q_B$ and $F_H = F_B$. The transition relation δ_H contains the transition (q, α, p) if and only if the transition (q, a, p) is in the original automaton's transition relation δ_B and h translates the symbol $a \in \Sigma$ to the symbol $\alpha \in \Sigma'$, meaning $h(a) = \alpha$.

In general, the automaton $\mathscr{H}$ might be non-deterministic, when for a state q there exists two transitions in the original automaton $\mathscr{B}$, (q, a, p_1) and (q, b, p_2), where $a, b \in \Sigma$ such that $h(a) = h(b)$ the resulting automaton $\mathscr{H}$ will be non-deterministic, as for the state q there exists two outgoing transitions with the same symbol.

The resulting automaton $\mathscr{H}$ is therefore not an automaton representing a system behaviour, as we want these automata to be deterministic. To get a valid automaton representing a system behaviour $\mathscr{H}$ needs to be determinized in a second step. Remember that all states of $\mathscr{H}$ are accepting, by applying the powerset construction the resulting deterministic automaton accepts the same language as the automaton $\mathscr{H}$. We will refer to the resulting automaton by $\mathscr{D} = (Q_D, \Sigma', \Delta, q_D, F_D)$.

Let us briefly explain why the accepted language is not alternated by applying the powerset construction.[8] In [5], it was shown that the powerset construction if only creating macrostates, where all states within this macrostates have the same mode(accepting or non-accepting), the resulting automaton accepts the same language as the original automaton. As the automaton $\mathscr{H}$ consists only of accepting states, this means that the powerset construction can be fully applied without alternating the language.

As we will work on these automata, it is important to specify certain observations about the relation between these automata. A first observation is that $h(L(\mathscr{B})) = L(\mathscr{H}) = L(\mathscr{D})$. This guarantuees that the final resulting automaton $\mathscr{D}$, corresponds again to a system behaviour, as it is deterministic and it has only accepting states, and is in fact the abstracted automaton to $\mathscr{B}$.

[8] In general, the resulting automaton $\mathscr{D}$ obtained by applying the powerset construction onto a non-deterministic automaton $\mathscr{N}$, accepts a super-set of the language of $\mathscr{N}$, in general $L(\mathscr{N}) \subseteq L(\mathscr{D})$.

Outlook

The weakly continuation closed homomorphisms are an interesting concept for model checking purposes. Especially together with inherently fair linear time verification, it might improve the model checking complexity under certain circumstances. Therefore we are currently working on developing an algorithm able to decide whether or not a homomorphism is weakly continuation closed or not on a given automaton.

A major problem of checking for weakly continuation closed homomorphism is that an infinte amount of prefixes need to be verified. We try to limit the number of verification steps which are necessary to perform the verification. Therefore we focus on the automata representing the system behaviour and especially on its graph structure. We believe that it is possible to reduce the number of verifications necessary, first of all to the number of states within the automaton. But we also believe that it is still possible to reduce this number. Actual work focus on the Strongly Connected Bottom Components of the automaton's graph. By following this idea it might be possible to reduce the number of necessary steps to the number of strongly connected bottom components of an automaton.

Bibliography

[1] E. M. Clarke, O. Grumberg, and D. E. Long, "Model Checking and Abstraction," in *ACM Transactions on Programming Languages and Systems*, vol. 16, no. 5, pp. 1512–1542, September 1994.

[2] U. Ultes-Nitsche, and S. St. James, "Improved verification of linear-time properties within fairness: weakly contiuation-closed behaviour abstractions computed from trace reductions," *Software testing, Verification and Reliability*, vol. 13, pp. 241-255, 2003.

[3] Th. Nicola, F. Niessner, and U. Ultes-Nitsche, "Model-checking Inherently Fair Linear-time Properties," in *Proceedings of the 3rd International Workshop on Modelling, Simulation, Verification, and Validation of Enterprise Information Systems (MSVVEIS 2005)*, pp.3–8, 2005.

[4] U. Ultes-Nitsche, and P. Wolper, "Relative Liveness and Behavior Abstraction (Extended Abstract)," in *Proceedings of the 16th ACM Symposium on Principles of Distributed Computing (PODC'97)*, (Santa Barbara, USA), pp. 45–52, 1997.

[5] U. Ultes-Nitsche, "A power-set construction for reducing Büchi automata to non-determinism degree two," *Information Processing Letters (IPL)*, vol. 101, no. 3, pp. 107–111, 2007.

[6] N. Francez, "Fairness," Springer Verlag, 1986.

[7] U. Ultes-Nitsche, and P. Ochsenschläger, "Approximately satisfied properties of systems and simple language homomorphisms," *Information Processing Letters*, vol. 60, pp. 201–206, 1996.

A Power-set Construction for Reducing Büchi Automata to Non-determinism Degree Two

Ulrich Ultes-Nitsche, University of Fribourg
uun@unifr.ch

Introduction

It is well known that deterministic and non-deterministic Büchi automata [1] accept different classes of ω-languages [10]. An example is $(a + b)^* \cdot b^\omega$, the ω-language of all ω-strings of a's and b's containing a finite number of a's, which can only be accepted by a non-deterministic Büchi automaton , but not by a deterministic one. In particular the usual subset construction used to determinize automata [4] will fail: In general, the determinized version of a Büchi automaton will accept a superset of the given non-deterministic automaton.

In this paper, a power-set construction to a given Büchi automaton is presented, which reduces the degree of non-determinism to at most two, meaning that to each state and input symbol, there exist at most two distinct successor states. The constructed Büchi will accept the same language as the given automaton.

Karpínski showed already in [5] that all regular ω-languages can be accepted by Büchi automata with non-determinism degree two.[9] However, Karpínski's construction takes the detour of constructing a Muller automaton [7] to a given regular ω-language, and from that Muller automaton, Karpínski constructs the Büchi-automaton of non-determinism degree two.

The construction presented in this paper is a much simpler power-set construction directly on a given Büchi automaton of arbitrary non-determinism degree. The construction works analogously to the determinization of finite-string-accepting automata, but is performed "modulo accepting and non-accepting states": instead of considering sets of states in general, sets of purely accepting and sets of purely non-accepting states will be considered.

Preliminaries

Let Σ be an alphabet (set of finite cardinality). The set of all infinitely long strings (aka ω-strings) over Σ is represented by Σ^ω. Sets of ω-strings, i.e. subsets of Σ^ω, are called ω-languages over Σ.

A finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ consists of a finite set Q of states, an input alphabet Σ, a transition relation $\delta : Q \times \Sigma \to 2^Q$, an initial state $q_0 \in Q$, and a set F of accepting states [4].

Let $x = x_0 x_1 x_2 \ldots \in \Sigma^\omega$ be an ω-string in Σ^ω. A run of A on x is a sequence $q_0 q_1 q_2 \ldots$ of states such that q_0 is A's initial state and $q_{i+1} \in \delta(q_i, x_i)$ for all $i \geq 0$. For a run r of A on x, $\omega(r)$ denotes the set of all states that repeat infinitely often in r. A run r is *successful* if and only if $\omega(r) \cap F \neq \emptyset$ (r contains at least one accepting state infinitely often). Automaton A *Büchi-accepts* x if and only if there exists an accepting run of A on x [1, 10]. The ω-language represented by A is $L_A = \{x \in \Sigma^\omega \mid A \text{ Büchi-accepts } x\}$. ω-languages that are Büchi-acceptable by some finite automaton are called *regular ω-languages* [10]. The automaton is then called a *Büchi automaton*.

[9] Non-determinism degree two is therefore the minimal degree of non-determinism which Büchi automata must have if the automaton class is supposed to accept the class of *all* regular ω-languages; automata of non-determinism degree one — i.e. deterministic Büchi automata — are too limited as discussed briefly above using the example $(a + b)^* \cdot b^\omega$.

If, for all states q in Q and all input symbols a in Σ, the successor state of q with respect to a is uniquely determined (i.e. $\delta(q,a)$ is a singleton set or the empty set for all states q and symbols a), then A is called deterministic. Otherwise it is called non-deterministic. For ω-languages, the language classes accepted by deterministic and non-deterministic automata differ: There exist ω-languages that can only be represented by non-deterministic finite automata; an example is given in the introduction of this paper.

The *non-determinism degree* $\nu(q)$ of state q is the maximal number of successor states it can reach by reading an input symbol: $\nu(q) = max_{a\in\Sigma}\{|\delta(q,a)|\}$. The degree of non-determinism $\nu(A)$ of automaton A is the maximal degree of non-determinism of one of its states: $\nu(A) = max_{q\in Q}\{\nu(q)\}$.

In the next section, the central lemma will be proved using König's Lemma [6] in the version of Hoogeboom and Rozenberg [3]:

Let $R \subseteq E \times E$ be a relation over an arbitrary set E. For all $n \geq 0$, let E_n be a finite nonempty subset of E such that $\bigcup_{n\geq 0} E_n$ is infinite and to each $e \in E_{n+1}$ there exists an $f \in E_n$ such that $(f,e) \in R$. Then there exists an infinite sequence $(e_n)_{n\geq 0}$ such that $e_n \in E_n$ and $(e_n, e_{n+1}) \in R$ for all $n \geq 0$.

Reducing the Degree of Non-determinism to Two

Let $A = (Q, \Sigma, \delta : Q \times \Sigma \to 2^Q, q_0, F)$ be a (non-deterministic) Büchi automaton. Let its *reduced version with non-determinism degree two* be

$$R = (2^F \cup 2^{Q-F}, \Sigma, \bar{\delta} : 2^Q \times \Sigma \to 2^{2^Q}, \{q_0\}, 2^F)$$

such that

$$\bar{\delta}(q,a) = \{\delta^*(q,a) \cap F, \delta^*(q,a) \cap (Q - F)\}$$

with $\delta^* : 2^Q \times \Sigma \to 2^Q$ being the usual extension of δ to sets of states (q is a subset of Q):

$$\delta^*(q,a) = \{r \in Q \mid \exists p \in q : r \in \delta(p,a)\}.$$

Note that in this subset construction only sets of states which are entirely accepting or which are entirely non-accepting are considered (i.e. the set of states of R is $2^F \cup 2^{Q-F}$). The initial state is $\{q_0\}$ (singleton sets trivially contain only either accepting or non-accepting states). The accepting states of R are sets of accepting states of A (i.e. the set of accepting states of R is 2^F). $\bar{\delta}$ relates a set of states and a symbol to all reachable accepting and all reachable non-accepting states respectively ($\delta^*(q,a) \cap F$ and $\delta^*(q,a) \cap (Q - F)$). As usual, R will be reduced by removing all states which are not reachable or which are not co-reachable.[10]

This construction is "nearly a determinization step", but by distinguishing accepting from non-accepting states, the definition of $\bar{\delta}$ restricts the degree of non-determinism $\nu(R)$ of R only to two: for each state q and each symbol a, $\bar{\delta}(q,a)$ contains at most two elements.[11]

Subsequently, let A be a Büchi automaton and let R be constructed from A as described above.

Lemma 1 $L_A \subseteq L_R$.

Proof Let $x = x_0x_1x_2\ldots \in L_A$. Let then $q_0q_1q_2\ldots$ be an accepting run of A on x and let r_0 be $\{q_0\}$. Let, for $i \geq 0$, r_{i+1} be the one of the two sets $\delta^*(r_i, x_i) \cap F$ and $\delta^*(r_i, x_i) \cap (Q - F)$ which contains q_{i+1}. By construction, $r_0r_1r_2\ldots$ is a run of R

[10] One removes all states not reachable from the initial state and one removes all states from which no accepting cycle can be reached.

[11] It contains *at most* two elements because the empty set will be removed from $\bar{\delta}(q,a)$ whenever $\bar{\delta}(q,a)$ contains it.

on x. Because infinitely many different q_i are accepting states of A, the infinitely many "matching" states r_i (the ones which contain an accepting q_i) are accepting states of R, and hence $r_0 r_1 r_2 \ldots$ is an accepting run of R on x.

Lemma 1 is immediate as the power-set construction always yields an automaton that Büchi-accepts a superset of the original automaton. The next lemma, Lemma 2, is the interesting one as it states that in the power-set construction used to reduce the degree of non-determinism to two, this superset is always the trivial one, leading to the result of Corollary 3 that A and R Büchi-accept the same ω-language.

Lemma 2 $L_R \subseteq L_A$.

Proof Let $x = x_0 x_1 x_2 \ldots \in L_R$. Let $r_0 r_1 r_2 \ldots$ be an accepting run of R on x. We show that there exists an accepting run $q_0 q_1 q_2 \ldots$ of A on x. Let $s_i = \{p_{(i)} \mid p \in r_i\}$, i.e. all states in r_i are simply labeled with "(i)". Then the s_i are finite sets for all $i \geq 0$, because the r_i are finite, and $\bigcup_{i \geq 0} s_i$ has infinite cardinality.[12] For two elements $p_{(i)}$ and $p'_{(j)}$ in $\bigcup_{i \geq 0} s_i$ let relation *succ* satisfy $(p_{(i)}, p'_{(j)}) \in succ$ if and only if $j = i + 1$ and $p' \in \delta(p, x_i)$. By definition of $\bar{\delta}$ of R, there exists a $p_{(i)} \in s_i$ to each $p'_{(i+1)} \in s_{i+1}$ such that $(p_{(i)}, p'_{(i+1)}) \in succ$. Application of König's Lemma establishes that there exists a sequence $q_{0_{(0)}} q_{1_{(1)}} q_{2_{(2)}} \ldots$ with $q_{0_{(0)}} \in s_0$, $q_{1_{(1)}} \in s_1$, $q_{2_{(2)}} \in s_2$, etc., such that, for all $i \geq 0$, $(q_{i_{(i)}}, q_{i+1_{(i+1)}}) \in succ$. Hence, by definition of *succ* and removing of the labels "(i)", $q_0 q_1 q_2 \ldots$ is a run of A on x such that $q_i \in r_i$. Because $r_0 r_1 r_2 \ldots$ is an accepting run of R on x, infinitely many different r_i are subsets of F, and thus infinitely many of the q_i are accepting. Hence $q_0 q_1 q_2 \ldots$ is an accepting run of A on x.

Note that the above construction fails in the case where the r_i are not either subsets of F or disjoint to F as it were, for instance, the case if one determinized A completely. The reason for that observation is that then the run $q_0 q_1 q_2 \ldots$ constructed in the proof is not guaranteed to be accepting.

The main result of this paper is an immediate consequence of the two lemmas:

Corollary 3 $L_R = L_A$.

Conclusions

The construction and related proofs in this paper showed that the degree of non-determinism of a Büchi automaton can be reduced to two by a simple subset construction without limiting the acceptance capabilities of the automaton. The presented result is a side result of work in direction of constructing a reasonably efficient algorithm for a satisfaction relation of linear-time temporal properties [2] with an inherent fairness condition [8, 9, 11]. The construction of the reduced Büchi automaton given in this paper will work in parallel with the computation of the synchronous product of a behavior and a property automaton — the synchronous product can not be computed in parallel with Karpínski's construction [5] of a Büchi automaton of non-determinism degree two. The ultimate goal of future work, using the result of this paper, will be the development of an algorithm for the inherently fair satisfaction of linear-time properties [8, 9, 11] which performs better than the currently known naïve algorithm using Boolean operations on Büchi and finite-string-accepting automata in the most straightforward fashion.

[12] The union of the s_i is infinite because of the introduced labeling: same elements in r_i and r_j, $i \neq j$, become different in s_i and s_j by labeling them with "(i)" and "(j)" respectively. Making the union of the s_i infinite is the only purpose of the introduced labeling.

Acknowledgements

I would like to thank Thierry Nicola and Frank Nießner for many fruitful, interesting discussions through which they have contributed to this paper. The work presented in this paper was supported by the Swiss National Science Foundation under grant # 200021-103985/1 and by the Hasler Foundation under grant # 1922.

Bibliography

[1] J. R. Büchi, "On a decision method in restricted second order arithmetic," in *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science 1960*, pp. 1–11, Stanford University Press, 1962.

[2] E. A. Emerson, "Temporal and modal logic," in [12], pp. 995–1072.

[3] H. Hoogeboom, and G. Rozenberg, "Infinitary languages: Basic theory and applications to concurrent systems," in *Current Trends in Concurrency*, LNCS, vol. 224, pp. 266–342, Springer Verlag, 1986.

[4] J. E. Hopcroft, R. Motwani, and J. D. Ullman, "Introduction to Automata Theory, Languages and Computation," Addison Wesley Longman, 2001.

[5] M. Karpínski, "Almost deterministic ω-automata with existential output condition," in *Proceedings of the American Mathematical Society*, vol. 53, no. 2, pp. 449–452, December, 1975.

[6] D. König, "Über eine Schlußweise aus dem Endlichen ins Unendliche (Punktmengen. Kartenfärben. Verwandtschaftsbeziehungen. Schachspiel)," in *Acta Litterarum ac Scientiarum Regiae Universitatis Hungaricae Francisco-Josephinae, Sectio Scientiarum Mathematicarum*, vol. 3, pp. 121–130, 1927.

[7] D. E. Muller, "Infinite sequences and infinite machines," in *AIEE Proceedings of the 4th Annual Syposium on Switching Theory and Logical Design*, pages 3–16, 1963.

[8] U. Nitsche, and P. Ochsenschläger, "Approximately satisfied properties of systems and simple language homomorphisms," in *Information Processing Letters*, vol. 60, pp. 201–206, 1996.

[9] U. Nitsche, and P. Wolper, "Relative liveness and behavior abstraction (extended abstract)," in *Proceedings of the 16th ACM Symposium on Principles of Distributed Computing (PODC'97)*, pp. 45–52, (Santa Barbara, CA, USA), 1997.

[10] W. Thomas, "Automata on infinite objects," in [12], pages 133–191.

[11] U. Ultes-Nitsche, and S. St James, "Improved verification of linear-time properties within fairness — weakly continuation-closed behaviour abstractions computed from trace reductions," in *Software Testing, Verification and Reliability (STVR)*, vol. 13, no. 4, pp. 241–255, 2003.

[12] J. van Leeuwen (editor), "Formal Models and Semantics," *Handbook of Theoretical Computer Science*, vol. B, Elsevier, 1990.