Towards a Self-Organizing Replication Model for Non-Sequential Media Access

Anita Sobe
Institute of Information Technology
Klagenfurt University
Austria
anita@itec.uni-klu.ac.at

Wilfried Elmenreich
Institute of Networked and Embedded Systems
Klagenfurt University
Austria
wilfried.elmenreich@uni-klu.ac.at

Laszlo Böszörmenyi
Institute of Information Technology
Klagenfurt University
Austria
laszlo@itec.uni-klu.ac.at

ABSTRACT

Due to the vast amount of video available in the Internet new access patterns emerge. Users do not always want to watch all of the content sequentially - such as in a movie - but want to pick specific parts, which are interesting for them. Based on a model of small and semantically meaningful and active video units, we derive an artificial hormone replication system that provides the flexibility for non-sequential access of units. We evaluate our model by simulation and compare it to a reference system. We show that simple local decisions contribute to global properties such as delay and robustness. We further introduce a clean-up function, which leads to adaptive management of the number of replicas.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed Systems, Information Networks*

General Terms

Experimentation, Measurement, Performance

Keywords

video delivery, self-organization, replication, non-sequential video access

1. INTRODUCTION

A lot of research has been done on flexible, more or less self-organizing content delivery techniques (see more in Section 2). On the other hand, traditional video streaming imposes strict temporal constraints on delivery and thus abandons a great part of potential on flexibility. We are usually ready to take this restriction into account, because the common assumption is that users want to watch videos in long, temporal sequences. This assumption is, however, in many and in a growing number of cases simply wrong. The more video material is available on the Internet the more different access patterns emerge. Many users are only interested in tiny fractions of a video and not even necessarily in the original temporal order. Moreover, they might wish to dynamically "compose" portions of different videos into one presentation, which itself might consist of several parallel viewing threads. Take a video recording of a ski-jumping competition as an example. Some users might be interested in watching it sequentially. A trainer might be interested in studying the jumping-off technique of some athletes in parallel. Another user might be interested in the performance of a single jumper, etc. Another example could be a motorway surveillance system consisting of hundreds of cameras producing pictures all the day. Traffic operators might be interested in traffic jams, police in accidents; but definitely nobody in everything.

At the end - in the play-back window - moving pictures must be presented, of course, under the usual temporal constraints. During delivery, however, we could apply much more flexible schemes than usual. We call this non-sequential video access. In the following, we suggest a novel, non-sequential video delivery system, based on biologically inspired self-organizing techniques. We want to take use of the simplicity and robustness of self-organizing delivery, and apply them to provide valuable video presentations.

We postulate some preconditions, namely: (1) videos are available in a form that they are *decomposed* into small, semantically meaningful, atomic *units* (see [2] as an example for such decomposing technique), and (2) users are able to formulate requests, in form of sequential and parallel *compositions* with an appropriate tool (see [10] as an example). We assume furthermore that the delivery system consists of a number of proxies and clients, which are logical entities. If a proxy and a client share exactly one physical node then we get a peer-to-peer system. We strive rather for an implementation, where proxy and client entities are running on different devices. The proxies form a content delivery overlay network. They could also be implemented on network devices of a "Future Internet", e. g., as in [5].

The basic idea is the following. When a client request (in general a list or a set of unit identifiers) enters the system, the client agent starts to emit artificial hormones to attract the corresponding units [11]. We assume that each unit can be attracted by a specific hormone. The strength of the hormone expresses the power of attraction. The proxies of

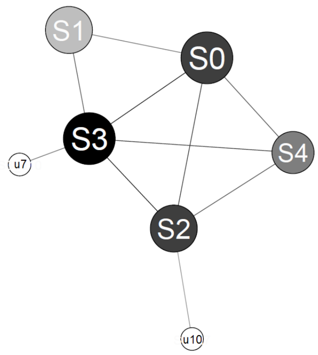

Figure 1: Network topology and location of units of interest

the delivery network propagate the hormones corresponding to a specific algorithm (see later). Hormones evaporate if not used for a while. Thus, we get a system of video units, moving autonomously, in a self-organizing manner to their targets, helped by proxies as "whistle stops". Units can also replicate themselves.

Until this point the model makes no distinction between the delivery of arbitrary data objects and video units. However, the notion of composition allows us taking dependencies among units into account. In the first step we regard only sequential compositions. Units on the front (near to their playback time) are attracted by especially strong smells, whereas those at the back are attracted less.

In the following we explain the hormone dynamics by a simple example:

We choose a small network of 5 nodes ($S_1 - S_5$) for sake of simplicity. The network topology including the position of the units of interest is shown in Figure 1. Initially, one instance of each unit exists. Clients at S_1 are interested in unit u_7 and unit u_{10} and request them sequentially. The hormones for both of the units are spread immediately, but the hormone level for u_7 will be higher than the hormone level of u_{10} and therefore unit u_7 is treated with higher importance than unit u_{10}. To attract the units, node S_1 disseminates a part of the local hormones to its neighbor nodes until the units reach their target. To further disseminate hormones to its neighbors, node S_1 has to produce hormones for the requested units periodically (e. g., every second). The unit will be transmitted to the node with the highest corresponding hormone level and follows therefore a frequented hormone trail to the requesting users. If the unit is currently in use it will be copied, otherwise moved. A clean-up function manages adaptively the number of replicas of these units.

We show in the following, how this model leads to an efficient and robust video delivery system. Even though units and proxies build only simple and local decisions they contribute to complex, globally emerging properties, such as low delay and robustness.

The paper is organized as follows: In the following section we provide background information and refer to related work on the topic. In section 3 we describe the proposed model in detail and discuss the calculation of the hormone level. In section 4 we evaluate the model by simulation. This section will be followed by the discussion of the results and future work.

2. BACKGROUND

As described in [4], most self-organizing systems are built based on behavior found in nature. Examples for that are found in [6]. The authors categorize nature inspired self-organizing mechanisms based on the type of information communicated, the information flow and the usage of information in a system. As an example the "Foraging" mechanism describes entities, called ants, searching for resources and if found, leaving a pheromone path for other ants. Pheromone disappears at some time, thus provide negative feedback.

The principle of foraging is found in several technical scenarios such as described in [9]. The design of a synthetic pheromone system was investigated for search in P2P file sharing. The authors describe a system's design using agents acting as ants. The peers hold pheromone and files. The agents searching for a specific resource can therefore act due to pheromone stored on each node.

We adopted the model of hormones for resources, but in our model requests and the resources themselves act as agents, triggered by hormones.

The idea of using an artificial hormone system for deployment of resources has also been applied in [1] by Brinkschulte et al. There, the authors simulate hormones with short messages in order to find proper deployments for real-time tasks among a set of processors. The principle is similar to our approach, although the problem statement is different, since our work also involves replication.

The design of self-organizing systems may not necessarily be bio-inspired. The author in [4] discusses self-organizing systems' design based on distributed system principles. The example for the design principle is a service replica placement system, where the cost of a request is measured in number of message transmissions per time unit to serve client requests. The goal is to minimize those costs without global coordination. Regarding the number of replicas the author defines a parameter ρ which describes the coverage radius of clients. If the number of requests for this service is below a given threshold, the service replica removes itself from the system.

Video units can also be regarded as a service, which has to be replicated, however, we assume that these units are not independent from each other. Therefore, a global cost function is out of scope for this purpose.

Another question related to replica placement is the number of replicas. The author of [8] proposes a replication scheme that adapts the number of video replicas according to their utilization rate. Temporary popular videos are replicated more often and the replicas are destroyed when the number of requests for the videos decreases. The number of replicas is handled centrally, our model considers local node decisions for solving this problem.

The authors of [7] introduce a new QoS parameter, Quality of Availability (QoA), and use it as a measure for replica placement. To increase QoA in general the authors describe heuristic based approaches, where resources are placed more likely on reliable nodes. Additionally, the authors propose a system that guarantees a given QoA. In this case all possibilities have to be tested to find the optimum resource placement. In this paper we do not consider the QoA parameter specifically, however, due to the hormones and replicas the model can handle node failures. QoS guarantees are out of scope of this paper.

3. MODEL

Our system consists of a number of connected proxy servers, each serving a number of clients. We use a connected Erdős-Rényi random graph for the network topology, i. e., each node has a fixed probability to connect to another node. Each server node has approximately the same number of connections. This network topology holds for small networks up to approximately 50-100 nodes. For larger networks we assume a more structured topology, for example a scale-free network.
Video units are considered to be of different content types mapped to a three dimensional array. Due to this mapping, similar units can be found by applying the Euclidean Distance. We assume that users have a preference for similar units, out of which sequential requests are "composed". After watching requested units the user's taste may change with a certain probability. E.g., a user first watches units of the category "news overview" and later the user is interested in units categorized as "ski jumping".

3.1 Hormone Dynamics

The hormone management is done on each of the proxies locally and we assume that the storage of the hormone values (represented as real numbers) requires negligible space on the nodes.
We define the hormone dynamics in the following:
If a client requests a video unit u_i which is not present at the server s_j, a certain amount η_0 of the respective hormone is spread on this server node. Until the video unit arrives, an additional amount η of this hormone is spread periodically. The hormone levels η_0 and η are optimized for sequential video unit consumption. Thus, hormone amounts also depend on the unit's position within a request. The nearer the unit to the playback time the higher its hormone amount.
The hormone amount (H) spread may add up to a possibly already present amount of this hormone at a server s_j:

$$\begin{aligned} H_{s_j,u_i} &= H_{s_j,u_i} + \eta_0 \\ H_{s_j,u_i}(t+1) &= H_{s_j,u_i} + \eta \end{aligned} \tag{1}$$

A server s_j periodically disseminates a given fraction α of a present hormone type to its neighbors. Another fraction ε of the hormone evaporates. Therefore, the decrease of a hormone can be calculated as:

$$decH_{s_j,u_i} = (\alpha + \varepsilon) H_{s_j,u_i} \tag{2}$$

The amount of $\alpha \cdot H_{s_j,u_i}$ is distributed evenly to the nodes which are connected to s_j. If the unit is present at the proxy s_j the amount defined in Equation 2 is subtracted, but not disseminated.

Thus, the hormone level H_{s_j,u_i} is influenced by:

1. An initial level of hormones on a new request.
2. A regular increase of hormones until the request is fulfilled.
3. Hormones that will be diffused to neighbors.
4. Hormones diffused from neighbor nodes on the same unit.
5. The level of evaporation.

hormone initial	0.40
hormones increased p. sec	2.48
hormones diffused p. sec	0.26
hormones evaporated p. sec	0.08
migration threshold	0.09
clean-up threshold in % of storage	31

Table 1: Parameter settings

3.2 Unit Transportation and Clean-up

The transportation of the video units is solely based on the hormone dynamics described before. Basically, a video unit will migrate to a neighbor if the difference of the respective hormone levels is higher than a given threshold. We assume that proxies forward content, even if not needed locally.
If a unit is in use (i. e., currently viewed or part of a viewed sequence), the unit is copied to the new proxy, otherwise it is moved.
A communication link is modeled to process the transmission of a unit in a non-preemptive manner. The order of transmissions depends on the respective hormone levels. Thus, a unit following a trail with a high level of hormone is preferred to a unit attracted only by a faint hormone level.
Furthermore, units can be (re-)moved by a clean-up mechanism. In order to use a proxy's disk efficiently, the proxy tries to remove units currently not following a hormone trail. The precondition for this deletion is that at least one replica exists on one of the proxies' neighbors. If this is not the case, the proxy tries to move the unit to a neighbor having the respective hormone. This leads in many cases to the advantageous effect that a unit is moved faster to its requesters than it would be without this movement. If the hormone for the unit to move does not exist on the neighbor nodes, the proxy tries to move the unit to a neighbor having a lower disk usage than itself.
The clean-up mechanism supports optimal disk usage and contributes to an advantageous replica distribution since a unit is moved as long as it finds a server where it is needed or a server with enough disk space.

4. EVALUATION

We evaluated our model by simulation. First, the settings for the simulation are discussed.
In Section 3.1 we showed the transport of units is based on spread, disseminated and evaporated hormones. Additionally, we introduce a number of thresholds for the resource management, e.g., a threshold of hormone needed to migrate a unit from one node to another, a threshold for handling the number of clean-up operations, a.s.o. In order to get reasonable values for all of these parameters, we used a genetic algorithm such as in [3]. We optimized for global throughput of units.
In the following Table 1 the relevant parameters are shown. We can see that the initial value of the hormone amount is low in comparison to the regular increase of the same amount. Therefore, the level of hormone considerably increases over time until the unit arrives at the requesting node. The diffusion mechanism distributes the hormone to the neighbors, and from the neighbors to their neighbors and so on. Due to the periodical increase of hormones, there are always more hormone diffusing away from the requesting node. However, if the unit is present at the current node

the diffusion will not take place, but the amount still subtracted. Thus, if a unit arrives at a node the evaporation is faster. In the opposite case the evaporation is very low to allow the hormone to spread throughout the network.
The low migration threshold makes the system very active, units are very likely to move. The clean-up limit is rather high (31 % of storage space), so for some scenarios with with low disk usage a clean-up might not take place at all.
According to these settings we evaluate our system in two cases: (1) without node failure (2) with node failure.
In both of the scenarios we compare the hormone based system with a reference replication model. This reference replication model routes units through the system via the shortest (minimum number of hops) path and replicates them at the target. In the following we refer to this model as routing model. Additionally, we compare our advanced clean-up system to distributed LRU clean-up (if the least recently used unit exists in the neighborhood it can be deleted). We show the difference in performance for networks of 5, 10, 20 and 50 nodes. Each node has a random number (between 1 and 20) of clients to serve. In the start of the simulation units are generated until the storage of 10 % for each node is reached, i.e., the number of units is proportional to the number of proxies.
Initially, there is only one instance for each unit. The size of a unit is between 5 KB and 5 MB, with a standard deviation of 500 KB. A request is defined as a sequence of 1-4 units, and triggers the initial hormone level of each of the units.
Since these parameters are generated randomly, we average the results for every experiment using 10 different runs. The simulation runtime was set to 500 seconds.
We compare the delay, which is measured from the time the playback of the most recently unit played to the time the current units starts to play. The delay of the first unit (start-up delay) of the request is measured from the request time to the startup time. We define the hit rate as the rate of units, which are immediately available at the target proxy when playback starts (i.e., delay is 0 seconds).

4.1 Scenario 1

In this scenario we investigate if a hormone based system performs equally or better to the reference system regarding delay and hit rate without node failure.

4.1.1 Network of 5 nodes

In the following we show the performance of the models on a small network of 5 nodes. In Figure 2 it is shown that the hormone based system with the clean-up procedure (in the following referred to as h.cl) outperforms the routing model with the same clean-up and with LRU replacement (r.cl and r.lru). The threshold that triggers the clean-up function is the same for LRU and hormone clean-up, therefore LRU is often executed. This results in high delay values because a high number of deletions take place. We applied the hormone clean-up function to the routing model as well. Although the routing model performs in this case better and more stable than LRU, the delay is still higher than in the hormone model. If the node number increases the hormone model (h.cl) always outperforms the routing (r.cl) regarding delay, except for the startup phase, where a high number of movements make the delay rise for h.cl.
In Figure 3 it is seen that the hormone model has a better

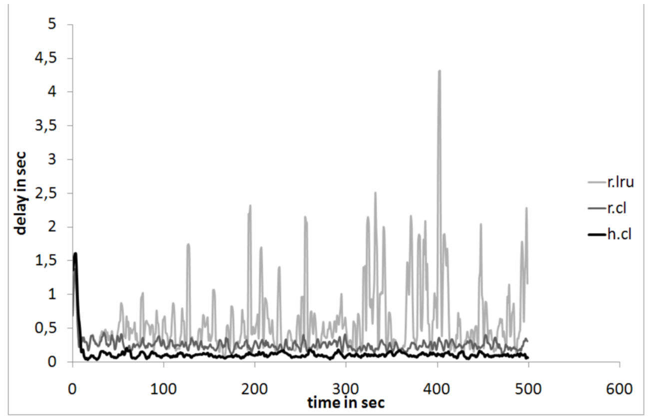

Figure 2: Delay comparison of hormone and routing system for 5 nodes with different clean-up functions

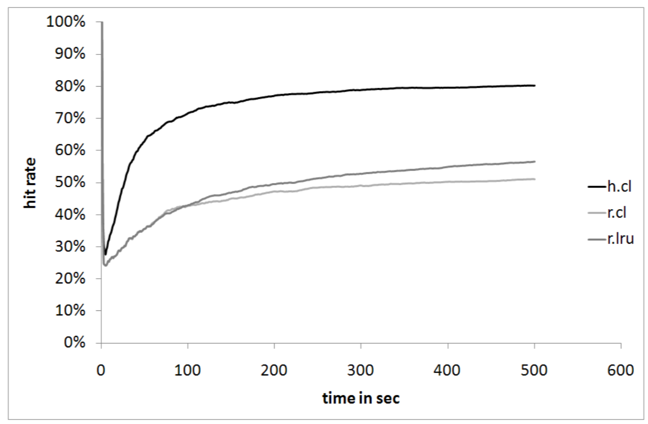

Figure 3: Hit rate comparison of hormone and routing system for 5 nodes

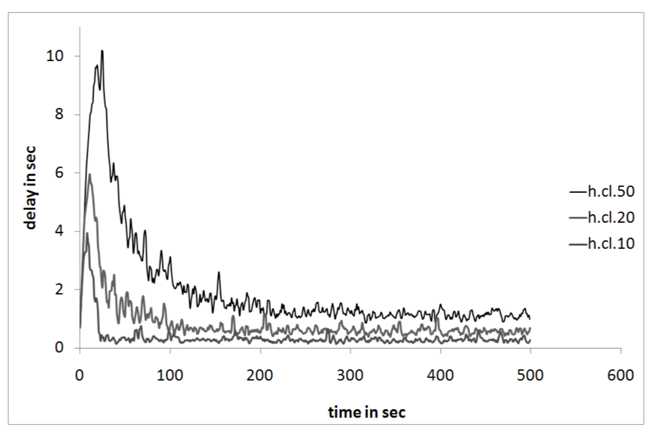

Figure 4: Delay comparison of hormone system for an 10, 20, and 50 nodes

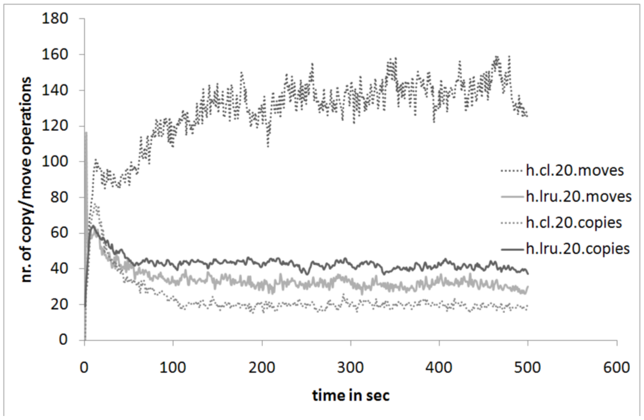

Figure 5: Copy and move comparison if LRU or the proposed clean-up function is applied

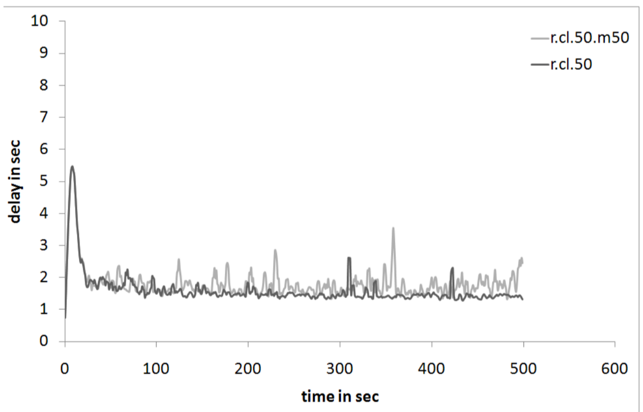

Figure 6: Delay comparison for a 50 nodes network with up to 50 % node failure, routing model

placement for units than both versions of the routing algorithms, which results in the high hit rate. As shown for the routing models r.cl and r.lru the clean-up function does not have a considerable impact on the hit rate.

4.1.2 Network of 10,20,50 nodes

We further show the delay development for the hormone model (h.cl) if having 10, 20 and 50 nodes. The delay increases with the number of nodes, since also the number of users and the number of units increase. In Figure 4, it is shown that in all of the runs the delay stabilizes at around 150 simulation seconds. The routing algorithm performs similarly to the 5 node case, i. e., the delay is always a bit higher than in the hormone case. E.g., the delay for 10 nodes of the hormone model h.cl is the same as the delay for 5 nodes of the routing model. In all of the runs the hormone system performs better than the routing system. The hit rate does not drop considerably on increasing the node number. For h.cl having 50 nodes the hit rate stabilizes at around 60 %, in comparison to h.cl having 20 nodes the hitrate is around 65 %. Although the LRU clean-up has a bad impact on delay, our proposed clean-up function does not come without cost either. This is shown in the following by the example of a 20 nodes network. In Figure 5 we show

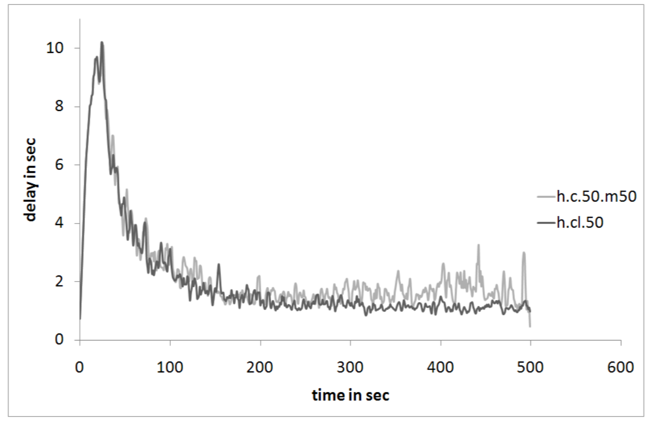

Figure 7: Delay comparison for a 50 nodes network with up to 50 % node failure, hormone model

that in the LRU case the system leads to about the same amount of copy and move operations. However, with every LRU clean-up a number of copies is deleted, so that most of the copy operations are done for nothing. In the proposed clean-up scheme most of the operations are movements and a very low number of copy operations are done. If we sum up the number of copies and moves the proposed clean-up is more expensive, but if we compare the overall delay, the movements do not have a bad influence.

4.2 Scenario 2

In the scenario before we saw that the hormone based model is a valuable solution for video delivery networks. In the following we compare the behavior of the hormone model and the routing model regarding the presence of node failures. Node failures are introduced regularly, depending on the node size and the simulation time. We simulated for 10, 20 and 50 nodes node failures of 10 %, 20 % and 50 % (m10, m20, m50). The first two failure modes showed marginal delay differences at around 10 milliseconds for all models. We compare in Figure 6 the original delay of the routing model with the 50 % failure case of the same model. The delay curve shows some peaks due to regular node failures, which lead to the rerouting of units. In Figure 7 it is shown that for the hormone model the delay starts to rise in the end of the simulation, because hormones have to be resetted. For the hormone and routing model it is seen that 50 % of node failure can still be handled but the performance starts to drop. A higher failure might lead to requests that cannot be fulfilled anymore, because of complete loss of units.

5. CONCLUSION AND FUTURE WORK

We have investigated a concept for a network of distributed servers hosting relatively small video units. In order to support a flexible system for non-sequential video viewing we introduce a self-organizing hormone based algorithm that derives decisions for unit replication and transport in a distributed manner. The model defines a simple set of interactions based only on local information is, nevertheless, able to guide the global placement of units in the system. Several parameters can be configured to guide the transport of units. We have applied a genetic algorithm to find applicable parameter sets optimized for non-sequential video access. If the hormone system is adapted to a more general object based delivery, it could also perform similarly.

A simple clean-up function is responsible for adapting the replica number, according to the current usage patterns of units.

We show that our proposed model performs equal and even better to a reference model routing units straightforwardly regarding delay and hit rates. The dependency of units can be exploited to guard their correct order. Thus, a self-organizing delivery system for flexible multimedia consumption is an applicable alternative to traditional systems. The proposed clean-up function introduced considerably more traffic in comparison to the LRU clean-up function because of move operations, however, this had no negative effect on the delay. If a real network is heterogeneous or only consists of mobile nodes, this might have a negative impact, what has to be further investigated.

Our proposed model showed to be robust for up to 50 % node failure for different network sizes. Units and hormones are just pushed to another route, which introduced a slight delay increase in the end of the simulation.

For future work, we plan to investigate if a more powerful decision model (e.g. by processing hormone levels with an Artificial Neural Network) allows for a significant improvement of the system performance.

The problem will be extended to support decision for further unit composition types. As an example, a user wants to browse through more content; the presentation might show several units in parallel, those units have to be shown almost immediately, but maybe not at full quality. If the user decides to navigate to a more detailed view (i.e., sequential), the quality has to be higher. These patterns can be exploited by the transport. Units often shown in parallel may then be replicated more often than the sequential ones. We further plan to support larger, but clustered, (overlay) networks as described by the authos in [12]. A real implementation is under progress and will be tested under real Internet conditions on the PlanetLab.

Acknowledgments

The results presented in this paper are part of the research efforts for the SOMA (Self-organizing Multimedia Architecture, `http://soma.lakeside-labs.com/`) and DEMESOS (Design Methods for Self-Organizing Systems, `http://www.demesos.tk/`) project, a Klagenfurt University and Lakeside Labs GmbH cooperation.

6. REFERENCES

[1] U. Brinkschulte, M. Pacher, and A. von Renteln. Towards an Artificial Hormone System for Self-organizing Real-Time Task Allocation. In *The 5th IFIP Workshop on Software Technologies for Future Embedded & Ubiquitous Systems*, pages 339–347, 2007.

[2] M. del Fabro and L. Böszörmenyi. Video Scene Detection Based On Recurring Motion Patterns. In *the Second International Conference on Advances in Multimedia, MMedia 2010, IARIA/IEEE, Athens, Greece, June 2010*, 2010.

[3] W. Elmenreich and G. Klingler. Genetic evolution of a neural network for the autonomous control of a four-wheeled robot. In A. Gelbukh and Á. F. Kuri Morales, editors, *Sixth Mexican International Conference on Artificial Intelligence*, pages 396–406. IEEE Computer Society, 2007.

[4] K. Herrmann. Self-Organizing Replica Placement - A Case Study on Emergence. In *SASO '07: Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems*, pages 13–22, Washington, DC, USA, 2007. IEEE Computer Society.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12, New York, NY, USA, 2009. ACM.

[6] M. Mamei, R. Menezes, R. Tolksdorf, and F. Zambonelli. Case Studies for Self-organization in Computer Science. *Journal of Systems Architecture*, 52(8-9):443 – 460, 2006. Nature-Inspired Applications and Systems.

[7] G. On, J. Schmitt, and R. Steinmetz. Quality of availability: Replica placement for widely distributed systems. In *Quality of Service IWQoS 2003*, volume 2707 of *Lecture Notes in Computer Science*, pages 325–342. Springer Berlin / Heidelberg, 2003.

[8] L. Rong. Multimedia Resource Replication Strategy for a Pervasive Peer-to-Peer Environment. *Journal of Computers*, 3(4):9–15, 2008.

[9] K. Schelfthout and T. Holvoet. A Pheromone-Based Coordination Mechanism Applied in Peer-to-Peer. In *Agents and Peer-to-Peer Computing*, volume 2872 of *Lecture Notes in Computer Science*, pages 109–132. Springer Berlin / Heidelberg, 2005.

[10] A. Sobe, L. Böszörmenyi, and M. Taschwer. Video Notation (ViNo): A Formalism for Describing and Evaluating Non-sequential Multimedia Access. *IARIA, International Journal on Advances in Software*, 3(1 & 2), August 2010.

[11] W. Trumler, T. Thiemann, and T. Ungerer. An artificial hormone system for self-organization of networked nodes. In *Biologically Inspired Cooperative Computing*, volume 216 of *IFIP International Federation for Information Processing*, pages 85–94. Springer Boston, 2006.

[12] S. Wieser and L. Böszörmenyi. Flocks: Interest-based Construction of Overlay Networks. In *the Second International Conference on Advances in Multimedia, MMedia 2010, IARIA/IEEE, Athens, Greece, June 2010*, 2010.