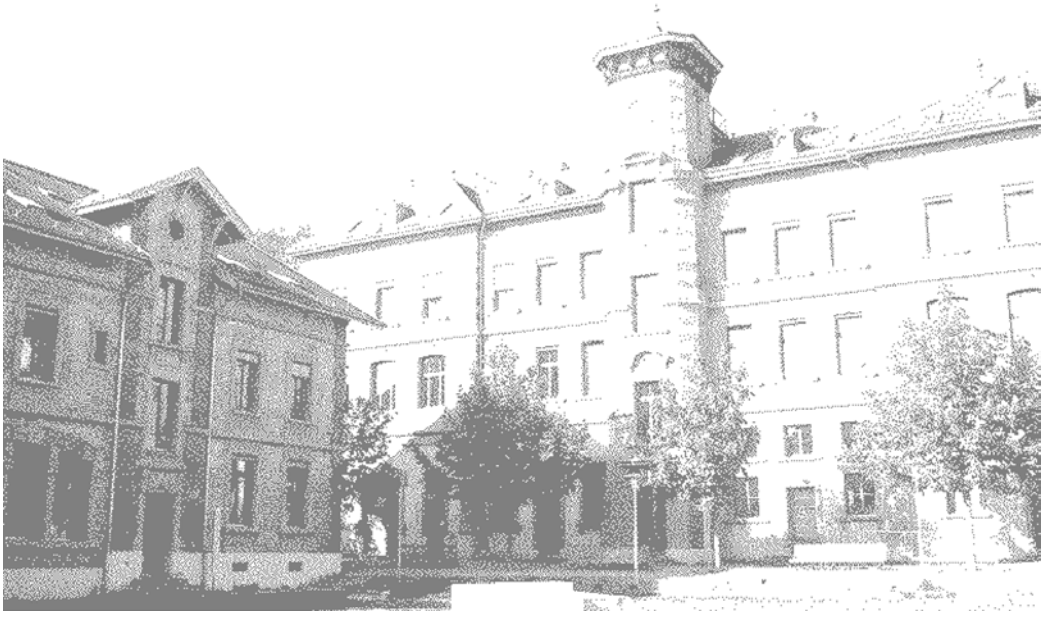
UNIVERSITÄT BERN

2013 Doctoral Workshop on Distributed Systems

H. Mercier, T. Braun, P. Felber, P. Kropf, P. Kuonen (eds.)

Technical Report IAM-13-002, July 15, 2013

Institut für Informatik und angewandte Mathematik, www.iam.unibe.ch



Doctoral Workshop on Distributed Systems

Hugues Mercier, Torsten Braun, Pascal Felber, Peter Kropf, Pierre Kuonen (eds.)

Technical Report IAM-13-002, July 15, 2013

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

Additional Key Words:
Wireless networks, opportunistic networks, content-centric networks, network emulation, cloud computing, software transactional memory, sensor network programming, fault tolerance, distributed storage, routing, overlay networks, cellular networks, localization

Institut für Informatik und angewandte Mathematik, Universität Bern

Abstract

The Doctoral Workshop on Distributed Systems was held at Les Plans-sur-Bex, Switzerland, from June 26-28, 2013. Ph.D. students from the Universities of Neuchâtel and Bern as well as the University of Applied Sciences of Fribourg presented their current research work and discussed recent research results. This technical report includes the extended abstracts of the talks given during the workshop.

LES PLANS-SUR-BEX
26 – 28 JUNE

2013 DOCTORAL WORKSHOP ON DISTRIBUTED SYSTEMS

UNIVERSITÄT BERN

UNIVERSITÉ DE NEUCHÂTEL

ECOLE D'INGÉNIEURS ET D'ARCHITECTES DE FRIBOURG

Workshop Program

Wednesday, June 26

Session 1 - Semantics

13:30 Speculative Message Processing with Transactional Memory in the Actor Model
Yaroslav Hayduk

13:55 Service-Centric Networks (SCN)
Dima Mansour

14:20 On Semantic Integration of Physical Business Entities into Enterprise IT Systems
Matthias Thoma

14:45 Distributed Programming using POPJava
Beat Wolf

Session 2 - Wireless Networks

15:40 Improving the TCP Performance of IEEE 802.11 Infrastructure Networks
Andrei Lapin

16:05 Distributed Object Oriented Programming for Wireless Sensor Networks
Yao Lu

16:30 TLG Opportunistic Routing Protocol and its Enhancement
Zhongliang Zhao

16:55 Content-Centric Communication During Opportunistic Network Contacts
Carlos Anastasiades

Thursday, June 27

Session 3 - Architecture Deployment and Optimization

9:00 Load-Balancing and High-Availability for a Machine Learning Architecture
Fabien Dubosson

9:25 SplayNet: Distributed User-Space Topology Emulation
Valerio Schiavoni

9:50 CDN and ICN Integration into the LTE EPS Architecture
André Gomes

10:15 Dynamic Optimization of Service Level Agreements
Alexandru-Florian Antonescu

Session 4 - Fault-Tolerance

11:10 Towards a Geo-Replicated File System
Raluca Halalai

11:35 Rollback Recovery Fault Tolerance with Clustering Methods in Message Passing Systems
Jianping Chen

12:00 Towards Eternal Storage using Data Entanglement
Verónica Estrada Galiñanes

Friday, June 28

Session 5 - Routing

9:00 StreamHub: A Massively Parallel Architecture for High-Performance Content-Based Publish/Subscribe
Raphaël Barazzutti

9:25 Secure Integrated Pub/Sub: Matching Scheme and Key Management Design
Emanuel Onica

9:50 High-Performance Elastic Content-Based Routing
Dr Marcelo Pasin

10:15 Performance Evaluation of Robustness for an Opportunistic Routing for Video Transmission in Dynamic Scenarios
Denis Rosário

Session 6 - Cellular Systems

11:10 Uplink Receiver for GSM Technology
Islam Alyafawi

11:35 Enhanced Timestamps in SDR for Time-based Localization Systems
Zan Li

12:00 Insight into a Self Organizing LTE-A Scheduler
Ioan Sorin Comsa

Workshop Proceedings

Uplink Receiver for Localization Services in GSM Technology

Islam Alyafawi
University of Bern
alyafawi@iam.unibe.ch

Abstract

GSM technologies wide availability motivates the research on the use of GSM as a common radio technology to support positioning or tracking and the related location-based services. GPS works flawlessly only outdoors and is not much of use in indoor environments, . Our target is to develop a stand-alone indoor localization system that relies solely on the signal perception of the Mobile Station and the Base Transceiver Station, without necessarily using the cellular operator's infrastructure. To tackle the problem we propose a new method to conduct time synchronization with the MS directly on the uplink using Normal Bursts. The method was validated against a benchmark solution on GSM downlinks and independently evaluated on uplink channels. Initial evaluation shows the promising up to 99% success rate in message decoding and reveal insights on the impact of signal power on the performance.

Keywords: gsm receiver; uplink capturing; localization.

1 Introduction

Most indoor location systems currently in use rely on a radio technology such as WiFi, Bluetooth or cellular networks [1] to support positioning or tracking applications and related location-based services. The wide availability of Global System for Mobile Communications (GSM) networks encourages research on the use of GSM as a common radio technology for localization systems. In addition, GSM signals seem to be more stable over time in comparison with WiFi or Bluetooth signals [2]. In general, a GSM localization solution can be implemented as an active or a passive system. Active systems can be implemented at the network side [6, 8], the terminal side [3, 4] or at both sides [5]. A passive system on the contrary requires no participation of the communicating parties but relies on overhearing radio (GSM) signals and their subsequent processing for the purpose of localization.

The latest development in the field of GSM signal capturing is OpenBTS [9]. OpenBTS is an ongoing project able to process uplink messages. However, it is an active tool since it acts as a base station to the served mobile devices. Until now there is no reliable passive tool that can offer comprehensive capturing and interpretation of uplink GSM signals. However, it meets the challenges in localization and users tracking. Our approach, presents the first efforts in the development of a GSM receiver that can be used for the purpose of passive localization. The initial evaluation shows reliable performance of signal capturing on real deployed GSM networks with a success rate up to 99%. The next research step seeks to follow and track a complete user communication flow with the network, e.g., location update flow.

2 GSM

One of the challenges of a GSM-based localization system is to capture the radio signal in order to allow its analysis for the purpose of positioning. As the signal information is divided into blocks referred to as bursts, which fit into a single time slot (TS) of the GSM time frame [10], signal capturing requires frequency and time synchronization between the transmitter and receiver. In order to support the MS in this task, the base station periodically transmits two burst types: Frequency correction burst and Synchronization burst. Additionally, Normal Bursts (NBs) are used to carry information (control and user traffic) exchanged between the MSs and their serving base station. A NB contains two information blocks separated by a 26-bit midamble, which carries one of eight predefined training sequences TSC(NB)s. In a passive system the challenge is to synchronize to the mobile station, given that the mobile station does not offer any information on frequency and time synchronization.

The processing steps to recover uplink messages relying on the NB's TSC(NB) are as follows: (i) Identifying the TSC(NB) used in the serving cell. (ii) Determining the time limits of the burst. (iii) Applying signal processing steps on the captured NBs. (iv) Parse the message content. (v) Store user identifications in database.

The implementation and evaluation work were performed on the USRP SDR platform by Ettus and in particular the N210 model. As there is no other such uplink capturing tool (passive uplink capturing) we decided to test it on a downlink channel with Airprobe as a benchmark. It is a passive tool, optimized for downlink capturing, i.e., recovery rate of above 95%. Table 1 shows that in best case performance the passive receiver recovers 99% of the messages captured by Airprobe, with a mean rate around 98.8%. The USRP hardware supports a timestamp and a recieved signal strength (RSS) measure for the captured NBs. We target a time-based solution for localization because of its promising accuracy [11]. But our approach can be adapted to RSSI-based solution as well.

Table 1: Success rate of passive receiver

Downlink message type	Passive receiver	Airprobe	Ratio (%)
Paging request type 1	61911	62581	98.8
Paging request type 2	19083	19262	99.0
Paging request type 3	17032	17206	99.0
RR System Info1	1477	1493	98.9
RR System Info3C	2955	2987	98.9
RR System Info4	2957	2986	99.0
RR immediate Assignment	6881	6973	98.6

3 Future Work

The imperfections in the observed performance are caused by imperfect synchronization. For further improvement of the receiver, we aim to address the following issues: improving the synchronization method and tackling frequency hopping. In GSM, often the used frequency may be changed, i.e., the signal hops, in order to limit the effect of interference. Since each user hops differently we started working on a wide-band capturing solution that allows us to later retrieve the individual narrow-band channels. Moreover, following a user communication flow is important for continuous user tracking.

Localization algorithms using RSS start by understanding GSM signal behaviour once panetrated indoors. A preliminary result that shows the outdoor to indoor signal propagation characteristics is shown in Figure 1. Future development should also include power analysis for uplink capturing since MSs are the target in localization services.

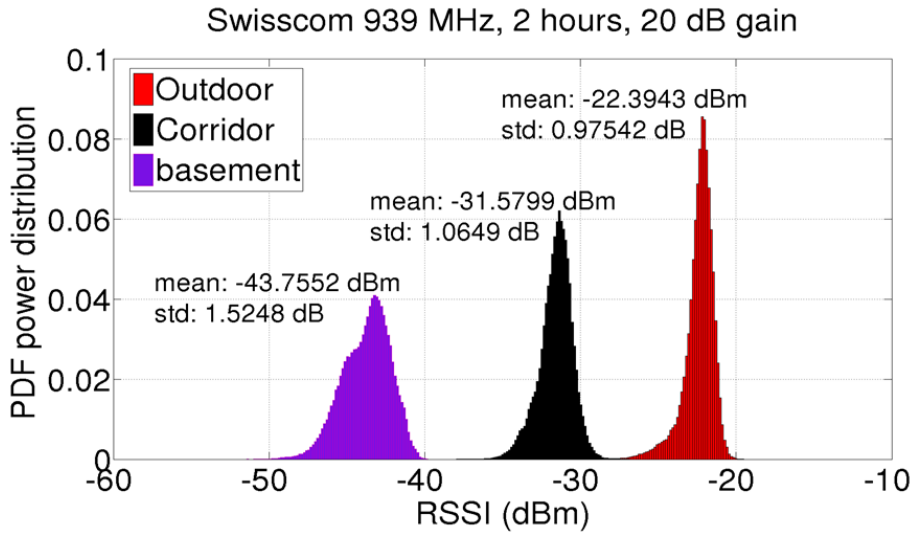


Figure 1: Outdoor to indoor signal propagation characteristics
Nisarg Kothari, Balajee Kannan, and M Bernardine Dias

References

[1] Jin-Shyan Lee , Yu-Wei Su and Chung-Chou ShenL, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," *Industrial Electronics Society*, pp. 46 - 51 , 2007.

[2] Veljo Otsason, Alex Varshavsky, Anthony LaMarca and Eyal de Lara, "Accurate GSM Indoor Localization," *The Proc. of UBICOMP 2005*, pp. 141-158, 2005.

[3] I. Lita, I. B. Cioc and D. A. Visan, "A New Approach of Automobile Localization System Using GPS and GSM/GPRS Transmission," in *in Proc. Int. Spring Seminar on Electronics Technology*, pp. 115-119, 2006.

[4] Nisarg Kothari, Balajee Kannan, and M Bernardine Dias, "Robust Indoor Localization on a Commercial Smart-Phone," in *Procedia Computer Science*, vol. 10, pp. 1114-1120, 2012.

[5] Dimitri Tassetto, Eriza Hafid Fazli and Markus Werner, "A novel hybrid algorithm for passive localization of victims in emergency situations," in *Advanced Satellite Mobile Systems*, pp. 320-327, 2008.

[6] Rose, R., Meier, C., Zorn, S., Goetz, A. and Weigel, R, "A GSM-Network for Mobile Phone Localization in Disaster Scenarios," in *Microwave Conference (GeMIC)*, pp. 1-4, 2011.

[7] S. Zorn, R. Rose, A. Goetz and R. Weigel, "A Novel Technique for Mobile Phone Localization for Search and Rescue Applications," in *Indoor Positioning and Indoor Navigation*, pp. 1-4, 2010.

[8] S. Zorn, R. Rose, A. Goetz and R. Weigel, "A Novel Technique for Mobile Phone Localization for Search and Rescue Applications," in *Indoor Positioning and Indoor Navigation*, pp. 1-4, 2010.

[9] "OpenBTS,", available at `http://wush.net/trac/rangepublic`

[10] Jörg Eberspächer, Hans-Joerg Vögel, Christian Bettstetter and Christian Hartmann, "GSM - Architecture, Protocols and Services," by *Wiley*, 2009.

[11] Amer Catovic and Zafer Sahinoglu, "The Cramer-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes," in *IEEE Communications Letters*, vol. 8, pp. 626 - 628, 2004.

Content-Centric Communication During Opportunistic Network Contacts

Carlos Anastasiades
University of Bern
anastasi@iam.unibe.ch

Abstract

Content-centric communication in opportunistic networks is promising because exchanged messages do not contain any source or destination addresses supporting caching in and content retrieval from any node. In this paper, we specify required implemented mechanisms such as a persistent cache extension that we implemented in CCNx and the open source content-centric prototype implementation that we evaluated on wireless mesh nodes. We also describe the developed CCN simulation framework for OMNeT++, a discrete event-based network simulator, to evaluate CCN in large mobile networks. The framework accurately reflects the content processing and storage structure of the CCNx prototype implementation. We evaluated different Interest request probing intervals and content lifetimes in mobile environments with intermittent connectivity. To reduce the retrieval time of desired content, we also implemented and evaluated different notification mechanisms that enable content sources to quickly notify one-hop neighbors about important events complementing Interest probing by requesting nodes. We implemented the persistent cache extension also in the simulation framework and evaluated it in networks with a varying number of requesters. Evaluations have shown that increasing the number of requesters providing received content to others improves content connectivity even in very sparse networks. This increases the overall retrieval time of all requesters in the network.

Keywords: content-centric; opportunistic communication; mobility.

1 Introduction

Opportunistic networking, a subset of delay-tolerant networking, defines communication in challenged networks where connectivity and contact durations between devices are unpredictable and intermittent. The main goal is to exploit contact opportunities between users to support best effort content and service interactions where fixed network infrastructure may not be available. In content-centric networking (CCN) [1], no device discovery is required because the communication is focussed on content names rather than on networking contacts. Nodes can express Interests to receive the corresponding Data from any node in response. The exchanged messages do not contain any source or destination addresses supporting caching in any node. Content discovery is performed using multicast/broadcast to quickly detect available content sources. Received content is cached locally but persistence is not guaranteed for a long time and, therefore, it can not be used in delay-tolerant networking. The challenges of opportunistic content-centric communication are as follows:

1. Discovery of available content names in continously and intermittently connected wireless networks.
2. Persistent caching of received partial files to enable delay-tolerant communication.
3. Content retrieval and forwarding in large mobile networks.

2 CCN in Opportunistic Networks

We have implemented two content discovery algorithms [2] to detect available content names in the presence of content sources and evaluated it in small scale emulations with VirtualMesh. The evaluations showed that additional transmission delays between subsequent multicast discovery requests can significantly reduce the number of received duplicates. We have also implemented a persistent cache extension to resume incomplete content downloads in case of long disruptions and evaluated it in a testbed with three wireless mesh nodes [3]. The evaluations showed that the developed extension significantly reduces the number of transmitted data packets and minimizes content download times with negligible processing overhead.

To evaluate opportunistic content-centric communication, the protocols need to be evaluated in large mobile networks where connectivity between nodes is intermittent. Deployments comprising many (mobile) wireless mesh nodes are not feasible and emulations using VirtualMesh do not scale with increasing number of nodes. Therefore, we implemented a CCN framework in OMNeT++ containing identical data structures and processing entities as in CCNx, the open source implementation of CCN. This framework is more accurate than existing CCN simulators such as ccnSim[1], which is a chunk-based CCN simulator implemented for the OMNeT++ simulator inspired by CCN concepts, or ndnSim[2], which is a NS3-based CCN framework that is more complete than ccnSim but uses different data structures in the CCN daemon, i.e., the CCN processing entity, and contains features not available in CCNx. When developing CCN applications for wireless mesh nodes, it is advantageous if the CCN daemon accurately follows the open source prototype implementation to use only available information and minimize the number of modifications when implementing it on wireless mesh nodes.

We performed different simulations in networks with playground sizes of 10'000m x 10'000m containing 100 nodes using Gauss-Markov mobility. In such networks where connectivity is intermittent, nodes can find content by periodically transmitting Interests probing for the content. With increasing Interest probing interval, i.e. time between unanswered Interests, the retrieval time until all nodes get the content increases since nodes may miss short contact opportunities. However, in networks with static or little mobility the probing interval should not be too short to avoid unnecessary Interest transmissions. Different values for the memory parameter α, which defines the mobility at time t based on the mobility at t-1, resulted in the same overall properties. We have also implemented adaptive algorithms to modify the beaconing interval but more realistic human mobility models are required for evaluation.

If there is important information that quickly needs to be distributed such as alarms or warnings, notification mechanisms enabling content sources to notify other nodes about certain events are advantageous. We implemented and evaluated two notification mechanisms based on Interest and Data beacons. The evaluations showed similar performance with respect to transmitted messages and notification time. In both cases, every node needed to meet the content source directly to receive the notification beacons. Therefore, notifications can only complement Interest probing but not replace them. However, a difference between Interest and Data beacons is that Interests are only forwarded to one-hop neighbors whereas Data beacons can be stored in the cache and events can be requested by nodes via multiple hops.

The content source can influence the time a content object is held in the cache by the

[1] http://perso.telecom-paristech.fr/ drossi/index.php?n=Software.ccnSim
[2] http://ndnsim.net/

content lifetime. Content with a longer lifetime value stays longer in the nodes' cache and subsequent probing requests can be satisfied by the nodes themselves without requiring the content source. This results automatically in load balancing. The content lifetime only affects cache storage but depending on cache size, number and length of data transmissions as well as cache replacement strategies, content may be replaced before the lifetime expires. Therefore, we also implemented a persistent cache extension and evaluated it in networks with a varying number of requesters. Every requester becomes a content source providing the content to others as soon as it has finished the content retrieval. This increases the connectivity to nodes providing content and even nodes that rarely or never see the original content source can quickly retrieve the content. Evaluations have shown that even in networks with only a few requesters every node is able to retrieve the content following this strategy.

3 Work in Progress and Future Work

Human mobility is self-similar: humans move in cycles meeting the same places and people on a daily basis. Long periods of immobility, e.g., at home, at work, in a train, are interrupted by short periods of high mobility when moving between locations. Therefore, we implemented different adaptive probing interval that change based on previous request success. Since Gauss-Markov mobility does not follow these properties and DTN mobility traces contain only binary contact information between two nodes which is not sufficient for wireless multicast communication, we will perform tests using the SLAW mobility model[4].

Additionally, to avoid duplicate transmissions and increase network throughput, we are currently also evaluating network coding in the context of mobile content centric networks. Content sources are encoding and combining data segments using lightweight Raptor codes[5] and Gaussian elimination. The implementation will be evaluated in the developed CCN simulation framework.

Furthermore, simulations with varying numbers of requesters showed that the more requesters request and provide content, the better the node connectivity to a suitable content source. Therefore, we developed and are currently implementing an agent-based content retrieval on Android smartphones. Requesters can delegate the content retrieval to other nodes, i.e., agents, enabling multi-hop communication in intermittently connected networks. Mobile agents store the retrieved content locally in their mobile repository, which may be occasionally synchronized with their home repository continuously connected to the Internet. Agents that retrieved the content can notify the original requesters via direct one-hop notifications or by contacting their home repository. To evaluate the agent-based approach in a mobile scenario, it will also be implemented in the CCN simulator framework.

Evaluations on wireless mesh nodes have shown that multicast data rates are very slow compared to unicast data rates. Therefore, we are also considering variable unicast and multicast addressing of packets. Discovery can be performed via multicast to quickly find an available content source. As soon as the content source is known, it can be addressed via unicast. The content source may reply to single unicast requests via unicast and switch to multicast in case of many requesters.

References

[1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, "Networking named content, " *Proceedings of the 5th international conference on Emerging networking experiments*, pp.1-12, 2009

[2] C. Anastasiades, A. Uruqi, T. Braun, "Content Discovery in Opportunistic Networks, " *Proceedings of 5th international workshop on Architectures, Services and Applications for the Next Generation Internet, pp. 1048-1056, 2012*

[3] C. Anastasiades, T. Schmid, J. Weber, T. Braun, "Opportunistic Content-centric Data Transmission During Short Network Contacts, " *submitted for journal publication*", 2012

[4] K. Lee, S. Hong, S. Joon, I. Rhee, S. Chong, "SLAW: self-similar least-action human walk, " *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 515-529, 2012

[5] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551-2567, 2006

Dynamic Optimization of Service Level Agreements

Alexandru-Florian Antonescu
University of Bern
antonescu@iam.unibe.ch

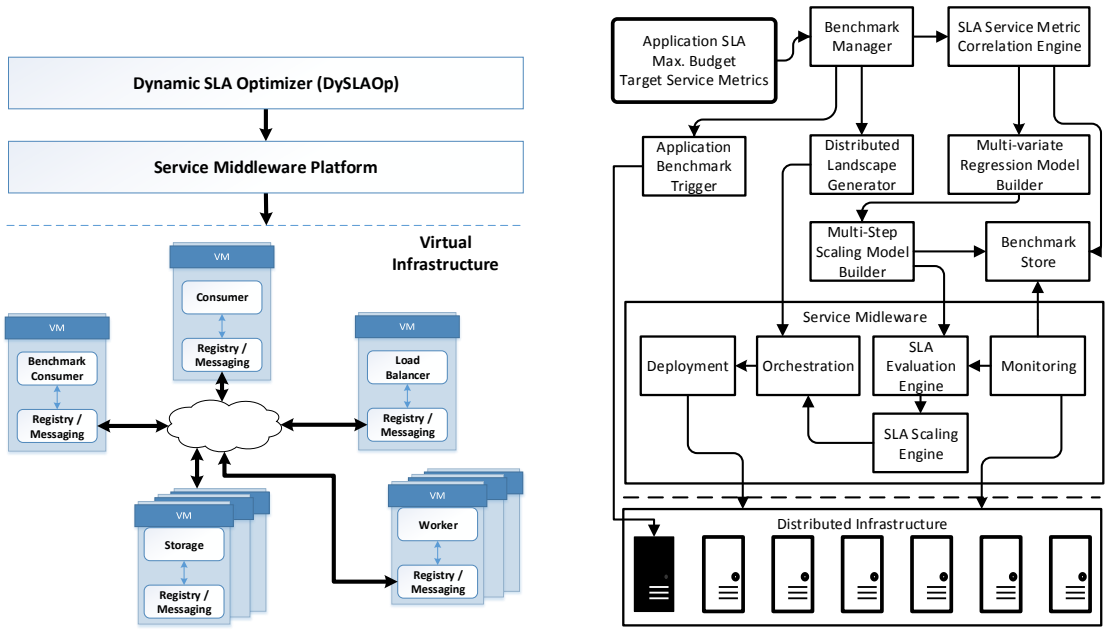
A Service Level Agreement ([1, 4]) (SLA) is a contract between a consumer and a provider of a service regarding its usage and quality. It defines guarantees or Quality of Service (QoS) terms under which the services are provided and the ways for checking those guarantees. We present how SLAs can be used as control input for a cloud management platform in order to guide both the allocation of distributed services to infrastructure resources and the dynamic scaling of services based on measured system state parameters.

We extend the Service Middleware Layer (SML) [2] with a new component that brings self-adjusting capabilities to our system. We investigate how SLAs can be dynamically optimized for enhancing the rules controlling the scaling-out of services belonging to distributed applications. We extend [3] a mechanism for discovering the dependencies between the service monitoring metrics contained in the SLAs by calculating the correlation coefficient between the time series corresponding to the services' metrics. This is done by generating multiple virtual machines (VM) infrastructure landscapes for hosting the services of targeted distributed application, followed by the instantiation of VMs, and finally benchmarking the application using an increasing number of concurrent application requests. Next, we apply linear regression on the obtained time series in order to learn the dependencies between the different service metrics contained in the SLAs.

We combine the discovered dependencies between the service metrics with the time series monitoring information for determining a Pareto frontier, which can be used for creating new SLA scaling rules. Finally, we present how data analytics mechanisms can be used for detecting periodicity patterns in the SLA monitoring data, which are then used for forecasting the future state of the application and for scheduling SLA actions on distributed applications.

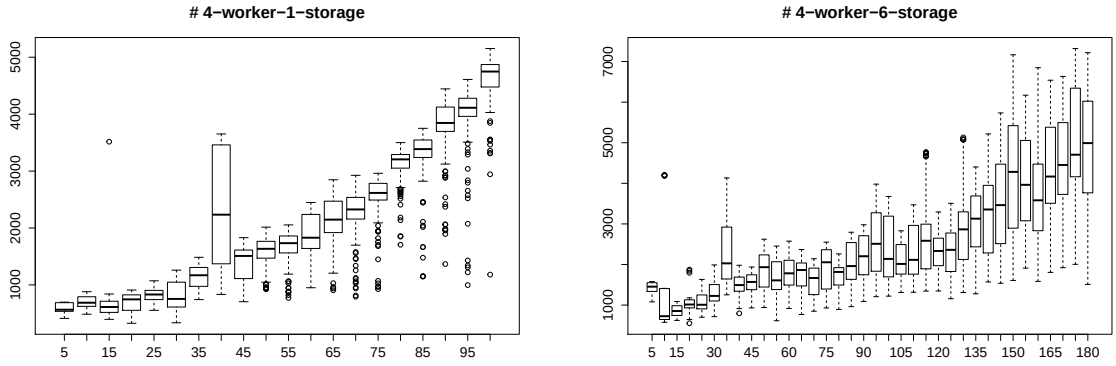
We consider a general Distributed Enterprise Information System (EIS) as the application under study, as this is simple to explain, has high availability and quality of service requirements, involves large data volumes, and is moreover a good representation of the multi-tier, distributed architecture seen in most modern enterprise systems. A typical EIS consists of the following tiers, each contributing to the SLA management problem: consumer, load balancer, business logic and storage layer. Fig. 2(a) provides an overview of the overall EIS topology.

We designed and implemented a new component called DySLAOp (Dynamic SLA Optimizer) that enables the SML to dynamically self-adjust to shifting application loads, while maintaining the contract with the user, i.e. the SLA. The architecture of the DySLAOp is presented in Figure 2(b). DyOSLA communicates with the SML for sending the distributed application provisioning requests and then for monitoring the status of the virtual machines deployment. Once the VMs have successfully started, the DySLAOp will wait for the EIS application to become operational (by checking the health status of the EIS Load Balancer, followed by issuing a single EIS request). Once the EIS operational status has been established, the EIS benchmark service will be started.



(a) (b)

Figure 2: a) EIS Topology and b) DyOSLA Architecture



(a) (b)

Figure 3: EIS Benchmark response for two application landscape configurations

The information flow through the DySLAOp component occurs in several stages. DySLAOp uses as input a SLA containing the service descriptions of a distributed application. Using the SLA numeric ranges associated with the minimum and maximum number of VMs that a service can use, DySLAOp will generate multiple application-landscape configurations. An application-landscape configuration contains the number of VMs that each application service will use. DySLAOp will sequentially instantiate each application-landscape, along with a special VM containing an application benchmark service capable of generating concurrent application requests. After all VMs become operational, the application benchmark is executed, while the SML monitors the SLA service metrics. Figure 3 depicts the dependencies between the increasing concurrent requests and the EIS response time SLA metric. Once the monitoring data is collected, DySLAOp begins the analysis phase, by performing the following steps:

1. From the collection of monitored SLA metrics, the set of predictor metrics for estimating the critical SLA parameters is determined. Next, linear models are calculated using the selected set of predictors.

2. Using a critical value (e.g. maximum) for the SLA target metric, the linear models are solved for determining the value of the system load (predictor metrics), which determines the critical system response (e.g. the maximum value of the target SLA metric).

3. An algorithm is used for calculating transitions from the initial application-landscape configuration to the final application-landscape configuration. The algorithm takes into consideration multiple criteria when selecting the next application-landscape configuration, such as minimizing infrastructure costs, minimizing application waiting times, minimizing SLA penalties, minimizing energy utilization, as well as supporting the application load which triggered the application-landscape transition.

4. The landscape configurations are added to a scaling 'path', including the number of service instance for each service type.

5. The scaling path is transformed into an estimation model for the critical SLA metric by considering the monotonically increasing load. The number of running services is also added to this model. Then, for each service type, an estimation model is calculated, which represents the scaling rule for the selected service.

Assuming that C is the critical value of the SLA target metric m^*, $m_i, i \in (1..p)$ are values of p SLA predictor metrics for m_c, v_j are number of VMs associated with service $S_j, j \in (1..s)$, $lm(m_1, m_2, .., m_p, v_1, v_2, .., v_s) = \alpha_0 + \sum_1^p \alpha_i m_i + \sum_1^s \beta_j s_j$ is the linear model estimating m^* and $T_k(m_1, m_2, .., m_p, v_1, v_2, .., v_s), k \in (1..s)$ is the application-landscape transition function for service k, then the determined scaling-out rules can be expressed as:

$if\ m^* > C\ then\ foreach\ k \in (1..s)\ do\ evaluate\ T_k(m_1, m_2, .., m_p, v_1, v_2, .., v_s)$

We validate our models by performing SLA-driven service scaling experiments using a distributed application representative for the enterprise information systems class of applications. We presented how time series analysis can be used for determining dependencies between the services composing the distributed application and how these dependencies can be used for enhancing the SLA rules controlling the application scaling.

References

[1] A-F Antonescu, P Robinson, and T Braun. Dynamic topology orchestration for distributed cloud-based applications. In *Network Cloud Computing and Applications, IEEE 2nd Symposium on*, 2012.

[2] A-F Antonescu, P Robinson, and T Braun. Dynamic sla management with forecasting using multi-objective optimizations. In *Integrated Network Management, IFIP/IEEE Symposium on*, May 2013.

[3] A-F Antonescu, P Robinson, and T Braun. Optimizing scalability and control of distributed applications using correlations and predictions in sla-driven cloud computing systems. In *International Conference on Network and Service Management (CNSM)*, 2013.

[4] A-F Antonescu, P Robinson, and M Thoma. Service level management convergence for future network enterprise platforms. In *Future Network & Mobile Summit (FutureNetw), 2012*, pages 1–9, 2012.

StreamHub: A Massively Parallel Architecture for High-Performance Content-Based Publish/Subscribe

Raphaël Barazzutti
Université de Neuchâtel
raphael.barazzutti@unine.ch

Abstract

By routing messages based on their content, publish/subscribe systems remove the need for applications to establish and maintain fixed communication channels between their components, thus yielding loosely coupled and highly flexible architectures. Such systems should provide high throughput, filtering thousands of subscriptions with low and predicable latency. Scalability remains an important factor in these architectures.

StreamHub decouples the functionalities associated with scalability from those that deal with filtering. Its design is oblivious to the semantics of the subscriptions and publications. It can support any type and number of filtering operations implemented by independent libraries. StreamHub exploits the natural scalability of the content filtering by carefully partitioning publications or subscriptions and processing event streams in parallel. This is achieved in a manner such that larger user populations or stricter throughput requirements can be supported by simply adding more machines to the system.

The scalability and performance of StreamHub are evaluated by using an implementation on a cluster with up to 384 cores (48 nodes). StreamHub behaves pretty well being able to register 150 K subscriptions per second and filter 8 K publications against 100 K stored subscriptions, resulting in nearly 700 K notifications sent per second. Comparison with well established broker PADRES shows an improvement of two orders of magnitude with the same hardware.

Keywords: Publish/subscribe; Content-based filtering; Scalability; Performance

1 Introduction

Content-based publish/subscribe (pub/sub) [1] is a strong contender for offering an efficient, yet *natural* communication paradigm to developers of large-scale applications. It supports decoupled interactions between the producers (*publishers*) and the consumers (*subscribers*) of information by the means of messages (*publications*). Decoupling occurs both in terms of space and time: publishers and subscribers do not need to know the existence or identity of one another, and no particular synchronization between them is necessary. They only communicate indirectly through a *pub/sub system*. It is the responsibility of this system to *route* publications from the publishers to interested subscribers. Routing is based on *subscriptions* registered by the subscribers to express their interest in specific content. The operation of *matching* the content of the publications against the subscriptions stored in the system is called *content filtering*.

We consider the following properties as key central ones:

(1) Scalability. The system should provide a speedup proportional to the computing resources additions. The ability to support increasing numbers of publishers/publications, subscribers/subscriptions, and notifications, as well as more computationally intensive filtering schemes, requires several levels of scalability. *Vertical scalability* is needed to benefit of additional resources available on nodes (more cores per CPU, more CPUs per machine).

Horizontal scalability is needed to take advantage of the addition of new nodes in the system.

Elasticity is needed to handle workloads which have computational requirements varying over time.

Most existing distributed pub/subs system [2, 3] are using a brokers overlay and we expect better results in terms of scalability.

(2) High throughput and low, predictable delays. The raw performance of such systems should meet the needs of demanding applications like high-frequency trading or network monitoring. The filtering operation being costly, the design should prevent multiple evaluations of a given publication against the same subscription several times. The latency, the time which spans between the submission of a message by a publisher and the actual delivery to the subscriber should remain of the same order as the communication time between that two clients and the pub/sub service.

(3) Filtering scheme agnosticism. The design and the architecture of a distributed pub/sub system should not rely on any assumption about the filtering scheme as well as on the representation of publications and subscribtions.

Most existing distributed pub/sub systems [2, 3, 4, 5, 6] are tight to fixed type of subscriptions and publications. More specifically, where publications are described as a collections of named values and subscriptions are built with a table containing attributes names of publication with a list of conjunctions which needs to be evaluated against each other.

2 Work Accomplished

In this work we make the following contributions:

- We built a set of operators running on a complex event processing framework, StreamMine. These operators span over the nodes of a cluster. The operators are separated in three discinct roles: 1) *subscribtion partitioning*, 2) *subscription storage and publication filtering* and 3) *publication delivery*.

- We built a piece of software, the *Data Converter & Connection Point* (DCCP) which allows several final users to access the engine and handles transparently conversion with popular formats (*Google Protocol Buffer*, *Apache Thrift*, *JSON*, ...).

- We compared our system, StreamHub, to PADRES a famous system using broker overlays to evaluate the actual performance of our system. We measured a higher performance per node with our system, but more important, we show that StreamHub scales linearly with the number of hosts. We notice that after a given number of nodes, PADRES stops to scale due to the use of a broker overlay which the involves costs due to the multiple evaluations of the same subscriptions and the routing of messages.

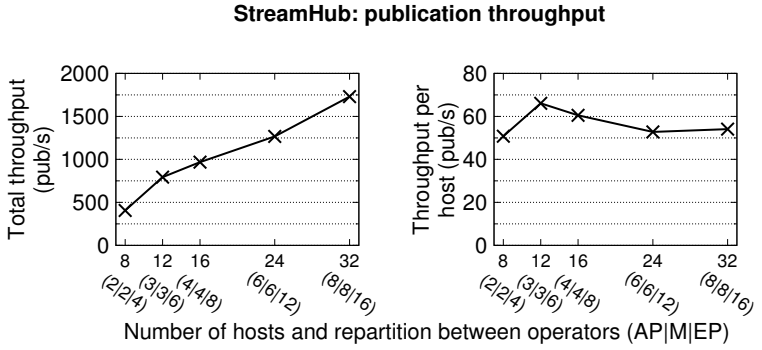


Figure 4: Throughput of StreamHub with 100,000 subscriptions, using the workload-optimal configurations for each number of available machines. The number of slices at each operator is indicated within parentheses.

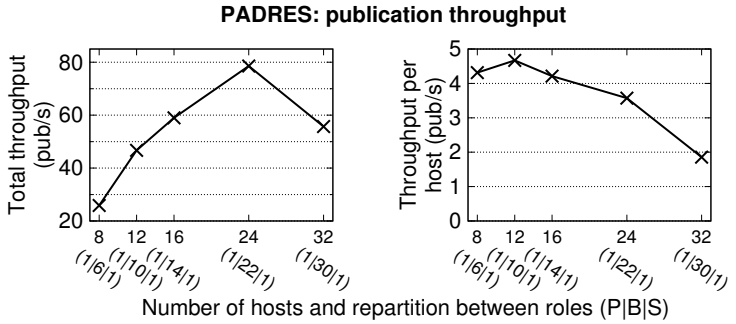


Figure 5: Throughput of PADRES with 100,000 subscriptions. The number of hosts is indicated within parentheses: a single publisher (P) and a single subscriber (S) were sufficient to fully load the brokers (B).

3 Work in Progress and Future Work

We are currently doing measurement on the elasticity in terms of impact on latency during the additions or removals of nodes while StreamHub is running. In future we plan to do more evaluations and to provide some extra improvements.

References

[1] P.T. Eugster and P. Felber and R. Guerraoui and A.-M. Kermarrec,"The Many Faces of Publish/Subscribe",*ACM Computing Surveys*, 2003

[2] A. Carzaniga and D. S. Rosenblum and A. L. Wolf, "Design and Evaluation of a Wide-Area Event Notification Service", *ACM Theoretical Computer Science*, 2001

[3] H.-A. Jacobsen and A. Cheung and G. Lia and B. Maniymaran and V. Muthusamy and R. S. Kazemzadeh, "The PADRES Publish/Subscribe System", *Handbook of Research on Adv. Dist. Event-Based Sys., Pub./Sub. and Message Filtering Tech.*, 2009

[4] R. Chand and P. Felber, "Scalable distribution of XML content with XNet", *IEEE Transactions on Parallel and Distributed Systems*, 19, 2008

[5] Y. Yoon and V. Muthusamy and H.-A. Jacobsen,"Foundations for Highly Available Content-based Publish/Subscribe Overlays", *International Conference on Distributed Computing Systems*, 2011

[6] A. Cheung and H.-A. Jacobsen,"Green Resource Allocation Algorithms for Publish/Subscribe Systems", *International Conference on Distributed Computing Systems*, 2011

Rollback Recovery Fault Tolerance with Clustering Methods in Message Passing Systems

Jianping Chen
Université de Neuchâtel and École d'ingénieurs et d'architectes de Fribourg
Jianping.Chen@edu.hefr.ch

Abstract

Message passing system is one of the most common paradigms of distributed system. A lot of applications, including HPC applications, are working on this paradigm. To avoid the message passing system failing becomes more and more important. A lot of strategies are proposed to reach the target. Hybrid or hierarchical fault tolerance strategy is one of the choices. In order to improve the performance and effective of hybrid or hierarchical fault tolerance strategy, we introduce clustering method. Clustering is basic method in scientific research, but seldom be used in the fault tolerance domain. Recently some researches show that fault tolerance also can apply clustering method to improve the performance. By introducing clustering method, the processes can be formed to different groups. Fault tolerance strategy will implement based on a group of processes but not only on single process compared to tradition fault tolerance strategies. It can reduce the memory overload thanks to the clustering method eliminating the number of records of message inside the cluster. We call this fault tolerance strategy as partition fault tolerance. Here we will discuss the idea and implementation of partition fault tolerance; compare the differences between the cases with clustering method and without clustering method. The scale problem is also an important issue in fault tolerance domain. Here we also consider the situation when the new process occurs and propose a suitable clustering method which is based on BIRCH clustering to scale the system of fault tolerance. With the help of clustering method, checkpointing and message logging techniques will cooperate in the strategy to form our partition fault tolerance.

Keywords: message passing; distributed system; clustering; fault tolerance.

1 Introduction

With the development of large distributed system, the scale of the system becomes larger and larger. Due to the number of computing nodes gets larger, the probability of multi-node failures increases. The data shows that the failure rate of large scale machine, like the ASCI-Q machine, is estimated to few hours: a 5 hours job with 4096 processes has less than 50% chance to terminate. To tolerance the multi-node failures will become a new topic in the distributed system fault tolerance. There are many reasons cause a failure. For example, process crashes down, machine crashes, the connection (the network) crashes, or hackers. But based on the behavior after the failure, we can classify the failures as following: 1) Fail-stop failure; 2) Byzantine failure. A fail-stop fault (fail-silent fault), correspondingly to fail-stop failure, is one where the faulty unit stops functioning and produces no bad output. More precisely, it either produces no output or produces output that clearly indicates that the component has failed. A Byzantine fault, correspondingly to Byzantine failure, is one where the faulty unit continues to run but produces incorrect results. Obviously, Byzantine fault is more difficult to handle. In this report, we do not talk about Byzantine tentatively; we focus

on the research of fail-stop fault and discuss about the techniques which are used to solve fail-stop faults. There are a lot of fault tolerance technique methods in distributed system and distributed system is also a large domain in computer science; here we keep focusing on the message-passing distributed system and the technique of the rollback recovery techniques, like checkpointing and message logging for the fail-stop failure.

2 Fault Tolerance With Clustering Method

Partition fault tolerance is a kind of hierarchical rollback recovery fault tolerance strategy, like partial message logging strategy. The core conceptions are:

1. divide the processes into different partitions;
2. apply different techniques on different partitions;

But the characteristic of partition fault tolerance is that it introduces clustering method in fault tolerance domain. By using clustering algorithm, reduce redundant information of fault tolerance techniques while improve the scalability of fault tolerance capability. As we know, hierarchical clustering algorithm is one of popular clustering algorithm in clustering and hierarchical clustering can be subdivided into two categories:

- Agglomerative Method
- Divisive Method

Based on the previous study, we have compared the differences between agglomerative method and divisive method. Here we design an agglomerative hierarchical clustering algorithm based on BIRCH as the clustering method for POP Model. With our clustering algorithm, we cluster the processes into different groups and apply different fault tolerance strategies based on the unit of processes group instead of single process. In this way, we can save a lot of memory or disk space and make a good scalability of the fault tolerance for system.

3 Future Work

1. Implement our clustering algorithm on POPC model and make the simulation;
2. Consider to cluster both messages and processes;
3. Make less strict constraint for no orphan process condition.

Insight Into A Self Organizing LTE-A Scheduler

Ioan Sorin Comsa
University of Bedfordshire and
University of Applied Sciences of Western Switzerland
Ioan.Comsa@beds.ac.uk

Abstract

Intelligent packet scheduling has the potential to make the radio resources usage more efficient in high-bit-rate demanding radio access technologies such as Long Term Evolution (LTE). The packet scheduling procedure works with various dispatching rules with different behaviors. In LTE networks, the scheduling decision is realized at each Transmission Time Interval (TTI). The length of TTI is considered to be 1ms duration. Then, the scheduling decision and resource allocation should be done within this interval. Therefore, at one TTI, different data packets from different queues are preferred to be scheduled based on the scheduling rule or discipline. In the literature, the scheduling disciplines are applied for the entire transmission sessions, thus the scheduler performance strongly depends on the exploited discipline. To the best of our knowledge Round Robin or proprietary schemes are usually applied in current LTE deployments. In this study, we discuss how a straightforward schedule can be provided within the TTI sub-frame using a mixture of dispatching disciplines per TTI instead of a single rule adopted across the whole transmission. Basically, in OFDMA cellular networks, there is a fundamental tradeoff between the cell throughput and the fairness levels between users which are sharing the same amount of resources at one TTI. The scheduling metrics which have been proposed so far are not able to achieve the desired level of fairness imposed by the system at each TTI. We thus propose a self-organizing scheduler that adapts the level of fairness depending on channel statistics and traffic load.

Keywords: LTE-A; TTI; CQI; OFDMA; Q-Learning.

1 Introduction

In modern cellular communications, the radio resource management plays a crucial role in which the whole radio spectrum is shared by the same users in the same time. The LTE packet scheduler is a particular sub-module which is responsible of delivering data packets to different users with an efficient allocation of radio resources and subject of different levels of service satisfaction constraints. Basically, the packet scheduler objectives can be divided in four main categories. The most important target from the operator point of view is to maximize as much as possible the system capacity or total cell throughput. When users experience the same radio channel conditions, the above problem does not represent a major concern. Therefore, the problem of sharing the radio resources by satisfying different service requirements becomes a very pretentious task. However, in reality the channel conditions of different users differ due to the temporal and frequency diversity. By maximizing the system throughput leads to the expense of user fairness, in which users with relatively better channel conditions are preferred to be scheduled to the detriment of other users located at the cell edge. Therefore, a special care of total cell throughput and user fairness tradeoff should be considered. The satisfaction of the Quality of Service (QoS) requirements is mandatory for the real-time (RT) and non-real-time (NRT) traffic types. In this case, delivering a minimum or maximum aggregated user throughput, a maximum average (normalized) packet delay and

a priority becomes another important task of the scheduling objective. The waste of the radio resources is a critical objective where in the absence of the full buffer model (FTP traffic type), packets are received from upper layers with different rates. In the scheduling process, the computation of transport block may require more data bits that exists in the queue at one time instance. Thus, the computed transport block carries a smaller amount of data leading to the waste of radio resources. It is the same case when the scheduler decides to serve some flows with empty data queues. To conclude, the network data supply should be considered together with the current state of the scheduling process. This technique is considered to be a cross-layer optimization due to the fact that information from multiple layers is considered in the MAC (Medium Access Layer) scheduling process. Due to the reason that there is no general performance metric that can highlight the overall scheduler performance, an intelligent scheduler should be proposed for implementation and validation in order to maximize the cell or sector total throughput by maintaining a minimum satisfaction level from the fairness, QoS and resource wastage points of view for all active users.

2 Work Accomplished

Four reinforcement learning algorithms (Q-Learning, QV,ÄìLearning, SARSA and ACLA) are used in order to find a proper scheduling rule at each TTI for the best matching conditions subject to different fairness constraints for each user. The intelligent scheduler considers a number of four elements inside of the state space: number of active users, the channel feedbacks statistics, standard deviation of the normalized user throughputs and the current value of the fairness parameter. The results indicate that our proposals are able to assure a faster convergence to the optimal solution in comparison with other existing methods. Finally, we prove that the system throughput gain is higher when the reinforcement learning algorithms are used.

3 Work in Progress and Future Work

In the future study other objectives such as average packet delay, minimum guaranteed bit rate and buffers statement will be included in the input state space. A special care should be paid in that case when traffic types are divided in different classes. Therefore, the intelligent scheduler should be able to stabilize the target performance for each of these traffic types.

Intelligent packet scheduling has the potential to make the radio resources usage more efficient in high-bit-rate demanding radio access technologies such as Long Term Evolution (LTE). The packet scheduling procedure works with various dispatching rules with different behaviors. In the literature, the scheduling disciplines are applied for the entire transmission sessions, thus the scheduler performance strongly depends on the exploited discipline. To the best of our knowledge Round Robin or proprietary schemes are usually applied in current LTE deployments. In this study, we discuss how a straightforward schedule can be provided within the transmission time interval (TTI) sub-frame using a mixture of dispatching disciplines per TTI instead of a single rule adopted across the whole transmission. Basically, in OFDMA cellular networks, there is a fundamental tradeoff between the cell throughput and the fairness levels between users which are sharing the same amount of resources at one TTI. The scheduling metrics which have been proposed so far are not able to achieve the desired level of fairness imposed by the system at each TTI. We thus propose a self-organizing scheduler that adapts the level of fairness depending on channel statistics and traffic load. Four reinforcement learning algorithms (Q-Learning, QV–Learning, SARSA and ACLA) are

used in order to find a proper scheduling rule at each TTI for the best matching conditions subject to different fairness constraints for each user. The results indicate that our proposals are able to assure a faster convergence to the optimal solution in comparison with other existing methods. Finally, we prove that the system throughput gain is higher when the reinforcement learning algorithms are used.

Load-Balancing and High-Availability for a Machine Learning Architecture

Fabien Dubosson
École d'ingénieurs et d'architectes de Fribourg
fabien.dubosson@hefr.ch

In the context of a project between the College of Engineering and Architecture of Fribourg, the University of Fribourg, and Softcom, a software company based near Fribourg, we are developing a platform where machine learning algorithms are used. A constraint of the platform is to answer almost in real-time and to scale up as the computational load increases. We have designed a specific architecture providing load-balancing and high-availability. The server is duplicated to provide more computational power, and because all the calls between entities are made through some RESTful interfaces, a project specific load-balancer can be inserted transparently in front of them. The load-balancer design permits to nest it at multiple levels to create sorts of computational clusters (which can be specialized for some algorithms, due to their hardware for instance). The load-balancer also acts as a high-availability tool by keeping a trace of all running jobs, by monitoring slave servers, and by redistributing jobs to other slaves in case of a server failure. A last important point is how to share the machine learning models among all server instances. This is not implemented for the moment but the idea will be presented and discussed during the workshop.

Towards Eternal Storage using Data Entanglement

Verónica Estrada Galiñanes
Université de Neuchâtel
veronica.estrada@unine.ch

Abstract

This work presents a new approach for data entanglement in a distributed storage system. The introduction of dependencies between stored content was initially proposed as a deterrent factor in censorship-resistant systems. The strategies found in the literature fail, however, to simultaneously provide a high level of robustness while being sufficiently efficient to be deployed in real-world systems.

High performance is a priority for the success of a practical implementation. The least amount of data required for disentangle the files, the better. In a similar vein, it is desirable to attain low bandwidth, low computation time and low storage requirements. However, the ultimate goal of entanglement is to achieve high levels of document dependency.

To address current limitations, we present a new entanglement method providing a sound compromise between strong robustness, pragmatism, and efficiency. The method to intertwine data blocks with both previously stored and forthcoming data defines a lattice structure that is able to repair a bounded amount of missing data blocks. In addition, the system requires cheap encoding mechanisms because all entanglement operations can be done with the Boolean XOR function. Furthermore, the implicit redundancy properties obtained by that technique could be used in combination with conventional encoding methods to protect data against various types of faults.

Keywords: data entanglement; fault tolerance; dependability.

1 Introduction

Availability and reliability are major concerns in large-scale distributed systems. They are traditionally addressed by the means of redundancy, which is applied in many different flavors. The Google and Hadoop file systems [1, 2], for instance, adopt a triple replication policy. The costs of such a policy become prohibitive due to the growing rates of "Big Data". Erasure coding is a popular approach to reduce storage overhead. Cloud storage solutions usually adopt *maximum distance separable* erasures codes, notably the *Reed-Solomon* error-correcting codes. Reed-Solomon is a mathematical technique that takes k data symbols of s bits to create n symbols based on the coefficients of a polynomial $p(x)$ over a finite field. Most storage systems that use erasure codes first split files into fixed size chunks before applying erasure codes and storing the resulting data blocks on geographically distributed servers [3, 4].

A question that remains poorly answered is how storage providers can guarantee data durability and recovery in untrusted settings. Durability requires that data is not permanently lost after failures, which typically involves affected files to be repaired. Although various sophisticated strategies exist [5], systems that use Reed-Solomon codes usually employs a naive method for repair: the whole file must be downloaded for reconstructing a single missing encoded block.

This work is an on-going research that primarily focus on effective ways to offer technical guarantees for data durability. Our source of inspiration are censorship resistant systems, in

which dependencies between stored content was proposed as a deterrent to censor attacks. For instance, in Tangler [6], an entanglement process establishes a one-to-many relationship between blocks and files. Moreover, the process of uploading a new document contributes to the replication of blocks from other documents as dependencies generate implicit redundancy. Our goal is to achieve high levels of data dependencies while designing an entanglement function with minimal requirements in terms of bandwidth, computation, and storage. In particular, durability mechanisms should be practical and efficient enough to be deployed in real-world systems.

At a high level, our approach works as follows. To upload a piece of data to the system, a client must first download some existing blocks (three by default, chosen deterministically) and combine them with the new data using a simple XOR operation. The combined blocks are then uploaded to different servers, whereas the original data is not stored at all. The newly uploaded blocks will be subsequently used in combination with future blocks, hence creating intricate dependencies that provide strong durability properties. The original piece of data can be reconstructed in several ways by combining different pairs of blocks stored in the system. These blocks can themselves be repaired by recursively following dependency chains. Note that this approach can be used in combination with standard erasure codes methods, with both data and redundancy blocks stored in entangled form in the system.

2 Work Accomplished

In this work we make the following contributions:

- We propose a novel method to entangle files with both previously-stored and forthcoming data, called *helical entanglement codes* (HEC). These codes are named after the helical lattice structure that supports our entanglement process, which is defined by $2*p+2$ independent chains of entanglements. HEC ensures all-or-nothing integrity, i.e., maximum achievable dependency, inside a cluster of entangled documents while requiring only cheap encoding mechanisms based on XOR computations.

- We sketch the architecture of an entangled data storage system that leverages HEC to provides a sound compromise between strong robustness, pragmatism, and efficiency.

- We analyze the properties of HEC and the trade-off between system robustness and lattice topology. The way the entanglement is done plays an important role in the durability of the file: dependency limitations are related to the factor p that describes the number of distinct helical strands that follow the same direction (left-handed or right-handed helix). In its default configuration, HEC's space overhead is similar to other solutions with 3 blocks generated per uploaded block.

3 Work in Progress and Future Work

In future we plan to do more evaluations and generalize the system to improve current bounds.

References

[1] Ghemawat, Sanjay and Gobioff, Howard and Leung, Shun-Tak, "The Google file system," *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 29-43, 2003.

[2] Shvachko, Konstantin and Kuang, Hairong and Radia, Sanjay and Chansler, Robert, "The Hadoop Distributed File System," *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pp. 1-10, 2010.

[3] Dabek, Frank and Kaashoek, M. Frans and Karger, David and Morris, Robert and Stoica, Ion, "Wide-area cooperative storage with CFS," *Proceedings of the eighteenth ACM symposium on Operating systems principles*, pp. 202-215, 2001.

[4] Wilcox-O'Hearn, Zooko and Warner, Brian, "Tahoe: the least-authority filesystem," *Proceedings of the 4th ACM international workshop on Storage security and survivability*, pp. 21-26, 2008.

[5] Dimakis, Alexandros G. and Ramchandran, Kannan and Wu, Yunnan and Suh, Changho, "A Survey on Network Codes for Distributed Storage," *Proceedings of the IEEE*, pp. 476-489, 2011.

[6] Waldman, Marc, "Tangler: A Censorship-Resistant Publishing System Based On Document Entanglements," *In Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 126-135, 2001.

CDN and ICN Integration into the LTE EPS Architecture

André Gomes
Universität Bern
gomes@iam.unibe.ch

Abstract

Recently, the demand for content in the Internet increased considerably. In fact, it keeps increasing at an even higher rate and, as the demand for content increases, also the amount of traffic in the networks is becoming higher. To tackle this issue, Content Distribution Networks have been used in the last years as a way to cache content closer to the user and hence eliminate some bottlenecks in the Internet core network and the overall amount of traffic in the peering between network operators. However, this did not solve the problem in one of the most important segments of the path between the content and the content source - the access network. In fact, as today the usage of mobile devices is getting each time higher, problems such as congestion and bandwidth inefficient usage that were tackled at the Internet core using Content Distribution Networks now need to be tackled in mobile networks. In this work, the objective is to propose solutions for these particular issues, mainly by introducing a new concept called Mobile Content Distribution Networks integrated with yet another concept named Information Centric Networking. Using both these concepts and their integration into the architecture of Long-Term Evolution Evolved Packet System, a solution is proposed to improve the Quality of Experience for the end user and to reduce the amount of traffic in mobile networks.

Keywords: CDN; ICN; LTE; Mobile Networks.

1 Introduction

Content Distribution Networks (CDNs) are well integrated into the Internet, accounting by some estimates for over 40% of the current web traffic with a perspective to grow to an even higher rate [1]. Efficient mechanisms to deliver all this content to the user are needed. For instance, the current CDNs are only used at the Internet core, mostly for inter-domain connections. At the same time, due to a number of limitations, cloud technologies are also not crossing the border of the big data centers and the principles they deliver cannot be used directly at the mobile networks. This issue is depicted in Fig. 1.

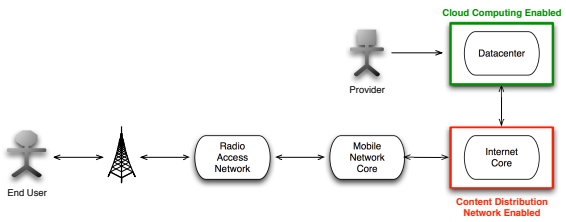


Figure 6: Content retrieval today

Considering all those factors, if we think about the explosive growth of mobile devices and their constant demand for content, we realize that a lot of optimization can be done inside the Mobile Network Operator (MNO) network to deal with all this traffic that demands spectrum and backhaul infrastructure (expensive network equipment, dark fiber optical links, etc.), both of which cost money to the MNO and do not generate revenue besides the data plans the subscribers may have.

One optimization approach is shown in Fig. 2. This involves the integration of CDNs into the Evolved Packet System (EPS) architecture used by the recently adopted Long-Term Evolution (LTE), extending the current CDNs, which are called Over the Top (OTP) CDNs because of their limitation in range, into the mobile networks and taking the cloud principles across the border of the big data centers.

This extension is often called Mobile CDNs [2], and they allow data to be cached and deployed closer to the user. In fact, all the involved partners can benefit from that: the end user gets a better quality of experience, the operator reduces traffic congestion on its network and gets a new business opportunity, and, finally, the content providers get more content availability with less resources and higher user satisfaction. Still, CDN strategies are not the most efficient way to obtain content while not used together with other enhancements. Considering video, for instance, the current paradigm of searching for something and then access a specific resource on a specific server is not efficient at all for the users and the network. Hence, the Information Centric Networking (ICN) paradigm emerges, providing a simpler way for the user to request content while providing caching and data relocation mechanisms and thus helping to reduce congestion in the network.

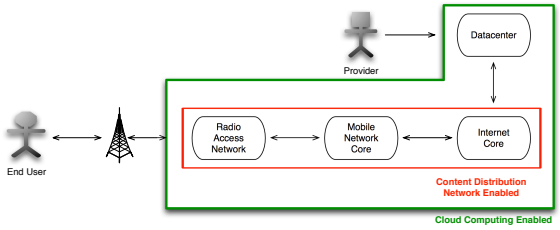


Figure 7: Content retrieval optimized scenario

2 ICN and CDNs in MCN

Currently, in the FP7 Mobile Cloud Project (MCN) [3], an approach that considers the virtualization of all the mobile network services is being considered. In fact, all the mobile network services are considered to be running in virtual machines provided by Infrastructure as a Service (IaaS) providers. Namely, Wireless as a Service (WaaS) and Evolved Packet Core as a Service (EPCaaS) are considered to be the services that provide the Radio Access Network (RAN) and the Core Network, respectively. Taking advantage of this extension of the cloud concept, which will run in smaller and more numerous data centers (micro data centers), an extension of the typical CDN concept is proposed.

This extension aims at placing the content closer to the user. However, this poses several challenges. One of them, and the main focus of this proposal, is to integrate the Mobile CDNs with the EPS architecture of the LTE. This integration must not violate the 3GPP standards, and therefore the proposal is to co-locate the new additions with the existing components of the EPS, not breaking the current flow of data. In Fig. 3 that integration is depicted:

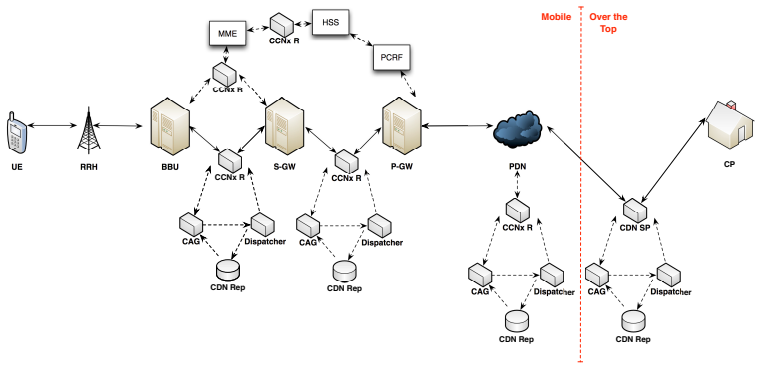


Figure 8: ICN/CDN Integration with EPS

As one may observe, there are a number of components involved. Some are directly related to CDNs, others enable the use of ICN. The process for the user to get the content starts at the User Equipment (UE). The UE then communicate with the Remote Radio Head (RRH), which is the radio interface of the evolved Node B (eNodeB). The communication will then get to the processing component of the eNodeB, the BaseBand Unit (BBU)m which will already have ICN (also named CCN, Content Centric Networks) and CDN components co-located with it (still very close to the user, around 10 km from the RRH usually). The first component is a proxy, which will have a simple role: to convert the common HyperText Transfer Protocol (HTTP) requests into ICN/CCN Interest messages, which usually include at least the content provider/source and a content identifier. This enables transparency from the user side, which may not be aware of ICN and the associated Interest messages. After the request is converted to an Interest message, it is forward to a Content Centric Networking X (CCNx) [4] router which implements an ICN layer. Then, using ICN, three different checks are made in the following order: check the content store if the requested content is already cached, check if the user already requested the content and the delivery is pending, check the forwarding table to redirect the request to another router. In the first case, content is delivered back to the user but the request must follow to the Serving Gateway (S-GW) and the PDN Gateway (P-GW) for the 3GPP defined extra processing. In the second case, the router must wait for the content and then it will forward it to the user. Finally, in the third case, the request is only forwarded to a S-GW with the next CCNx router derived from the forwarding table. At this gateway, the process is similar. However, two things change: there is no proxy as any request has been converted to an Interest message and CDN repositories can also be co-located. CDN repositories also play a key role for the proposed system, as if the CCNx router enters the third check, it will compare the network distance and load between it and the next CCNx router in the forwarding table together with its distance to the closest CDN repository, which has the content cache and which should be used to fulfill the request. To verify if a CDN repository has the content and obtain it, its Content Access Gateway (CAG) is accessed by the CCNx router and a dispatcher will do the storage check. If the content exists, it is delivered to the CCNx router, which will forward it to the user. Otherwise, routing data containing the closest cache Internet Protocol (IP) address is provided or the content can even be fetched in real-time. This process repeats, until the current OTP CDNs are reached and the process is the same as today.

3 Challenges

To achieve a feasible system based on the proposal stated in the previous section, a number of challenges must be overcome. The first one is the naming scheme to use for ICN, as the level of detail for the content can be important for the message processing itself (eg. if made at the BBU, content details should be known to forward the correct information for charging to the P-GW) and also because it should be easy for the user to adopt it while being unique for the different contents. The second one, and perhaps one of the most demanding challenges, is how to deal with user mobility. When the user moves from one location to the other, a number of things may happen/be required: the cached private (or public if considering group mobility) content should move with the user (Follow-Me Cloud), content fetching requests should be kept alive and fulfilled at the new location and ICN/CDN infrastructure may need to be instantiated. Even with good solutions to deal with these requirements, predicting users mobility may be the difference between a good and a bad quality of experience for the moving user. In fact, if there is a way to predict the user mobility, actions can be taken in advance (proactively) and the user does not have to wait or feel any initial difference (reactively). Finally, the third one is related with the caching strategies. A number of things need to be considered, such as hierarchic caching levels, policies for caching and caching distribution, where/when to relocate content and the need for supporting services closer to the caches (eg. transcoding). Here the emphasis goes for the caching locations (can depend on load, energy, pricing, etc.), the decision process to know when to cache (and how to cache) and the factors to take into account when prioritizing content to be cached.

Yet, the biggest challenge of which all the previous challenges depend on is how to integrate CDN and ICN into EPS really without affecting its functionality and avoiding all the details that can be seen as a violation of 3GPP standards that both carriers and hardware vendors use today.

4 Conclusions and Future Work

We may conclude that the major issues currently affecting content download are the usage of bandwidth and the processing bottlenecks. This is mainly valid at the mobile networks, where efficient caching mechanisms and content oriented protocols are not used. In order to find a solution for that specific issue, a proposal based on current CDN mechanisms and with the addition of ICN was presented. Together with the use of the cloud principles according to the related European project Mobile Cloud Networking, this proposal can be implemented and integrated with the LTE EPS. This, however, presents challenges due to needed compliance with the very important standards that are part of a joint effort of all the involved parties. Additionally, all the other challenges demand plenty of research to achieve valid algorithmic solutions with both performance and feasibility that can be used in the real world.

To achieve the proposal and overcome the challenges, a set of next steps has been defined. First, a testbed will be implemented to provide a first proof of concept and to be used to test the subsequent deployments. This will provide all the feedback to realize what are the technical issues and later to test the solutions for them. However, the most important research questions, which concern intelligent algorithms, will need answers from large scale scenarios that cannot be provided by the testbed. Therefore, and to test mainly the scalability and performance of the proposed solutions, a simulation environment will also be considered.

References

[1] STL Partners - Telco 2.0 Research, “Customer Experience: Is it Time for the Mobile CDN?,” *URL: http://goo.gl/poKDQ*, 2013.

[2] Yousaf, F.Z.; Liebsch, M.; Maeder, A.; Schmid, S., "Mobile CDN enhancements for QoE-improved content delivery in mobile operator networks," *Network, IEEE* , vol.27, no.2, pp.14,21, March-April 2013

[3] EU FP7 Mobile Cloud Networking, “MCN Project Website,” *URL: http://www.mobile-cloud-networking.eu*, 2013.

[4] Palo Alto Research Center, “Project CCNx,” *URL: http://www.ccnx.org*, 2013.

Towards a Trustworthy Geo-replicated Data Store

Raluca Halalai
Université de Neuchâtel
raluca.halalai@unine.ch

Abstract

Cloud storage systems have gained momentum over the past years. However, the Service Level Agreements of commercial cloud storage providers do not guarantee security, nor 100% robustness. Therefore, many users are reluctant to entrust their data to the cloud.

Our goal is to make cloud storage systems trustworthy. We are currently building a distributed data store that guarantees robustness, consistency and high performance. Robustness is achieved by replicating data within and across multiple datacenters, belonging to different cloud storage providers. However, replication implies a trade-off between the desired properties. Since ensuring high availability is crucial for a data store, the challenge is to find a good trade-off between consistency and performance.

Keywords: cloud computing; geo-replication; trustworthy storage.

1 Introduction

Cloud computing offers access to inexpensive and scalable resources, ideal for storing increasing amounts of data. Cloud resources are typically provided by third party entities whom become in charge of the stored data. Therefore, cautious users and enterprises are reluctant to entrust their data to the cloud.

The goal of this project is to build a *trustworthy* cloud storage system that supports data anonymity and confidentiality, ensures long-term cryptographic safety, provides secure and verifiable data deletions, and is able to geographically control data placement and access. In particular, my work is focused on developing a distributed data store that guarantees robustness, consistency, and high performance. Robustness is typically achieved by replicating the service across multiple sites. However, a replicated system implies a trade-off between availability, performance, and consistency [4]. Since availability is crucial for a trustworthy data store, the challenge is to find a good trade-off between consistency and performance.

Section 2 gives an overview of related work, which constitutes the foundations of my research. Section 3 presents the current state of my work and discusses the most important challenges. In Section 4, we conclude.

2 Related Work

FlexiFS Previous work from Valerio et al. [1] resulted in the prototype of a distributed file storage service (DFSS). FlexiFS was designed to be modular, in order to serve as a testbed for evaluating different DFSS designs (in particular, different consistency levels). Figure 9 presents the high-level architecture of FlexiFS. In short, it consists of two types of nodes: *client nodes*, which provide a filesystem interface to the users and *storage nodes*, organized in a one-hop distributed hash table. Client nodes access the storage service by sending a Web service request to a proxy, which hides the topology and operation logic of the data store from the client.

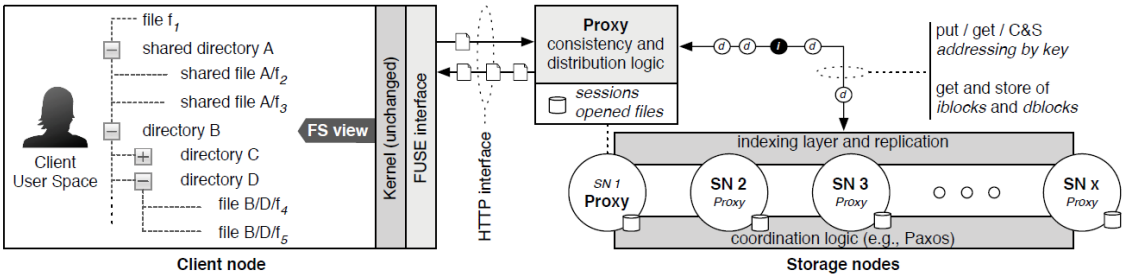


Figure 9: High-level architecture of FlexiFS

Ring Paxos Paxos is a fault-tolerant consensus algorithm, widely used to implement state machine replication – the fundamental approach for achieving fault-tolerance in a distributed system. Marandi et al. [3] proposed a cluster-friendly variant of the Paxos protocol. U-Ring Paxos disposes all participating nodes in a ring and, thus, leverages pipeline communication to maximize the throughput of the protocol. The authors show that U-Ring Paxos achieves high throughput, almost constant with the number of participating nodes. Latency grows with the number of nodes; however, it is comparable to other state-of-the-art protocols involving ring topology. To enhance the scalability of U-Ring Paxos, Marandi et al. [2] proposed Multi-Ring Paxos, a protocol that orchestrates multiple instances of U-Ring Paxos.

3 Designing a Geo-Replicated Data Store

Our plan is to build a data store that guarantees robustness, consistency, and high performance. To this end, we intend to use FlexiFS as the base platform for our data store. To boost robustness, we intend to replicate data across sites located in different regions of the world and belonging to different providers (geo-replication). We intend to use Ring Paxos to obtain strong consistency among the replicas located at the same site, while having weaker consistency across sites (e.g., eventual consistency or causal consistency). We are currently surveying literature to find weaker consistency models that have meaningful semantics in the context of file block storage. We are also working on a mechanism for controlling the geo-replication in a distributed hash table. Finally, we are dealing with implementation issues related to integrating the Java Ring Paxos implementation with our Lua framework.

4 Conclusion

The lack of trust in cloud storage providers has been a major hurdle for the adoption of cloud storage services. We are currently designing a geo-replicated data store that guarantees robustness, consistency, and performance. Such a data store would represent a key component in the development of trustworthy cloud storage systems.

References

[1] J. Valerio, P. Sutra, E. Riviere, and P. Felber, "Evaluating the Price of Consistency in Distributed File Storage Services," in *DAIS 2013*.

[2] P. J. Marandi, M. Primi, and F. Pedone, "Multi-Ring Paxos," in *DSN 2012*.

[3] P. J. Marandi, M. Primi, N. Schiper, and F. Pedone, "Ring Paxos: A high-throughput atomic broadcast protocol," in *DSN 2010*.

[4] E. A. Brewer, "Towards Robust Distributed Systems," in *PODC 2000*.

Speculative Message Processing with Transactional Memory in the Actor Model

Yaroslav Hayduk
Université de Neuchâtel
yaroslav.hayduk@unine.ch

Abstract

Parallelism can be supported by the means of message passing and/or shared memory depending on the type of the underlying architecture: distributed or multi-core systems. As an example, the *Actor Model* has been successfully used for scalable computing in distributed systems. Actors are objects with a local state, which can only be modified by the exchange of messages. One of the fundamental principles of the Actor Model is to guarantee sequential message processing, which avoids typical concurrency hazards, but limits the achievable message throughput. We propose to add support for speculative concurrent execution in actors using *transactional memory* (TM). Our approach is designed to operate with message passing and shared memory, and can thus take advantage of parallelism available on distributed and multi-core systems. The processing of each message is wrapped in a transaction executed atomically and in isolation, but concurrently with other messages. This allows us (1) to scale while keeping the dependability guarantees ensured by sequential message processing, and (2) to further increase robustness of the Actor Model against threats due to the rollback ability that comes for free with transactional processing of messages. We test the effectiveness as well as the validity of our design within the Scala programming language and the Akka framework. Our tests reflect that the overhead of using transactions is hidden by the improved message processing throughput, thus leading to an overall performance gain.
Keywords: concurrency; actors; transactional memory; speculative processing.

1 Introduction

Recent advances in the integration technologies have transformed Moore's law such that the exponential increase in transistors now converts to an exponential increase in number of cores in a chip. This pace of integration suggests that not only concurrent programming should become mainstream (i.e., "*free lunch is over*") but also scalability of concurrent applications should become a first class concern. Programming environments for current multi-core architectures largely rely on a shared-memory model. This is the case, for instance, of the "transactional memory" (TM) paradigm, which has received considerable attention lately. TM supports speculative execution of concurrent threads and, using a checkpoint/retry strategy, can seamlessly handle conflicting data accesses. Further, by relying on transactions, TM provides suitable tools for the development of dependable applications.

This approach is, however, only effective up to a certain number of cores, typically 8–16 cores on 1–4 CPU sockets. Beyond this threshold, contention on shared resources (memory, bus, caches) quickly limits scalability, with performance sometimes degenerating to that of sequential execution as accesses to shared data end up being serialized. In such settings, we need a more appropriate programming model with lower synchronization overheads, such as message-passing, which can scale better than shared memory on large numbers of processors [1].

For the currently existing multi-core architectures the widely used programming model is based on shared memory. The shared memory model is efficient for a small number of cores, because it allows concurrently executing parties to optimistically access shared data. However, an increasing number of cores also increases the concurrent accesses to shared data and there comes a point where the accesses to shared data needs to be serialized, leading to a performance close to sequential execution. This observation suggests that a concurrent programming solution that can also deal with scalability issues should adopt message passing paradigm.
The Actor Model, initially proposed by Hewitt [3], is a successful message-passing approach that has been integrated into popular distributed computing frameworks [4]. The Actor Model introduces desirable properties such as encapsulation, fair scheduling, location transparency, and data consistency to the programmer. It also perfectly unifies concurrent and object-oriented programming. While the data consistency property of the Actor Model is important for preserving application safety, it is arguably too conservative in concurrent settings as it enforces sequential processing of messages, which limits throughput and hence scalability.

We address this limitation by proposing a mechanism to boost the performance of the Actor Model while being faithful to its semantics [4]. The key idea is to apply speculation, as provided by TM, to handle messages concurrently as if they were processed sequentially. In cases where these semantics might be violated, we rely on the rollback capabilities of TM to undo the operations potentially leading to inconsistencies.

We see a high potential for improvement in scenarios where actors maintain state that is read or manipulated by other actors via message passing. With sequential processing, access to the state will be suboptimal when operations do not conflict (e.g., modifications to disjoint parts of the state, multiple read operations). TM can guarantee safe concurrent access in most of these cases and can handle conflicting situations by aborting and restarting transactions.

Speculation can also significantly improve performance when the processing of a message causes further communication. Any coordination between actors requires a distributed transaction, which we call *coordinated transaction*. We combine coordinated transactions and TM to concurrently process messages instead of blocking the actors while waiting for other transactions to commit.

2 Work Accomplished

We have implemented our approach in the Scala programming language and the Akka framework [5]. We showed that concurrent message processing and non-blocking coordinated processing can considerably reduce the execution time for both read-dominated and write-dominated workloads.

3 Work in Progress and Future Work

We are currently evaluating our approach further using a distributed linked list benchmark already used with other concurrent message processing solutions [2]. Also, we are planing to implement a message passing version of the k-means datamining algorithm and observe its behaviour when executed in a modified Akka environment, which uses our optimizations.

References

[1] B. J. Smith, "Shared memory, vectors, message passing, and scalability," *Parallel Computing in Science and Engineering*, vol. 25, no. 3, 1988, pp. 27-34.

[2] S. M. Imam, V. Sarkar, "Integrating task parallelism with actors," in *Proc. 2012 ACM international conference on Object-oriented programming systems languages and applications. (OOPSLA)*, 2012, pp. 753-772.

[3] C. Hewitt, P. Bishop, R. Steiger, "A universal modular actor formalism for Artificial Intelligence," in *Proc. of the 3rd international joint conference on Artificial Intelligence (IJCAI)*, 1973, pp. 235-245.

[4] R. K. Karmani, A. Shali, G. Agha, "Actor frameworks for the JVM platform: A comparative analysis," in *Proc. of the 7th International Conference on Principles and Practice of Programming in Java (PPPJ)*, 2009, pp. 11-20.

[5] P. Haller, "On the integration of the Actor Model in mainstream technologies: The Scala perspective," in *Proc. of the 2nd edition on Programming systems, languages and applications based on actors, agents, and decentralized control abstractions (AGERE!)*, 2012, pp. 1-6.

Improving the TCP Performance of IEEE 802.11 Infrastructure Networks

Andrei Lapin
Université de Neuchâtel
andrei.lapin@unine.ch

1 Introduction

Wireless networks have become ubiquitous and are extensively used at public places, such as airports and restaurants. In order to facilitate communication, wireless networks use the *Open Systems Interconnection (OSI)* model tailored to the wireless scenario. The OSI model uses the *TCP/IP* protocol as the de facto standard for internet communication. The *TCP/IP* protocol, however, may exhibit poor performance in wireless networks due to bandwidth degradation, high latency and packet losses. In my work I am targeting to boost the performance of wireless networks by identifying specific issues with the *TCP/IP* protocol, which limit its performance.

2 Background

In the OSI model, when the packed is transmitted, it eventually goes past the *transport layer*. The TCP protocol is among the most commonly used protocols, which the transport layer uses to ensure reliable packet delivery. In an event when the packet cannot be successfully delivered to its destination, the retrying behaviour may be triggered after either (1) the expiry of the TCP-specific *delivery timeout* or (2) when the *fast-retransmit* mechanism is triggered.

Every time a packet is transmitted, it is associated with a sequence number N. Once the packet arrives at its destination, the receiving party responds with an acknowledgement packet. The acknowledgement packet is assigned with a sequence number $N+1$, which follows the sequence of the received packet's sequence number. For packets that arrive out of order, the receiver replies with the acknowledgement packet containing a sequence number, which is based on the last acknowledgement packet received in order.

After the transmitter receives 3 or more duplicate acknowledgements, both, the *fast-retransmit* and the *congestion control*, mechanisms are triggered. The *fast-retransmit* mechanism resends the missing packet without waiting for the timeout to expire. The *congestion control* mechanism resets the *congestion window (CWND)* [1] and re-initiates the *slow-start* [2] procedures.

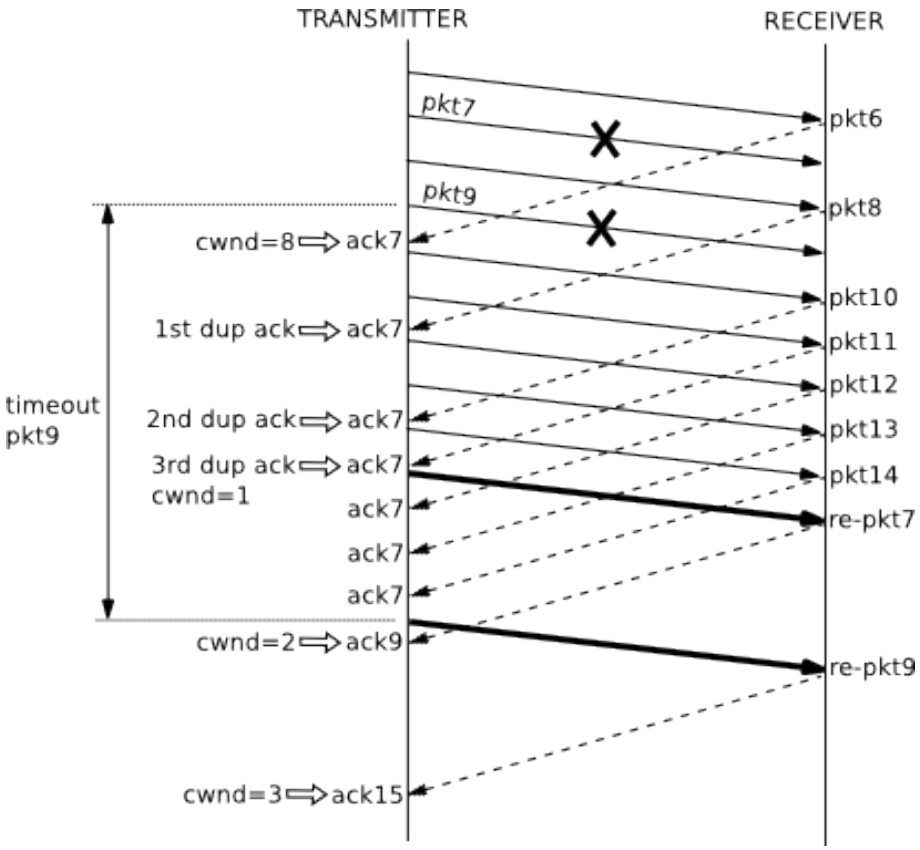


Figure 10: The fast-retransmit mechanism

Figure 10 illustrates the fast-retransmit mechanism for the case, in which the delivery of two packets fails (packets pkt7, pkt9). For the pkt7 packet, the *fast-retransmit* mechanism will be triggered after receiving the third duplicated acknowledgement. The pkt9 packet will be retransmitted because of the expired timeout.

3 Research goal

The primary goal of my research is to increase the throughput in wireless networks. Throughput may be reduced because of the congestion control mechanism working on the transmitter side. In the *fast-retransmit* mechanism, the transmitter reduces the *CWND* when a packet is dropped. This happens because the transmitter cannot determine whether the packet was dropped due to congestion or due to poor link quality. When such a transmitter receives a *TCP negative acknowledgement (NACK)* from the proceeding router, indicating that the packet was dropped due to the link layer, the *CWND* reduction should be cancelled and previous *CWND* size should be restored. Specifically, the *CWND* reduction should not be performed while the *fast-retransmit* mechanism is triggered in the case when that packet was dropped by the link layer. In the baseline case, when the network is congested or the packet is lost due to poor link quality, the size of the *CWND* is always reduced. We can, however, avoid reducing the size of the *CWND*, for the packets dropped by the link layer. By eliminating the unneeded resizing of the *CWND*, I expect to increase the overall network throughput.

4 IEEE 802.11 layer transmission

In IEEE 802.11, packets are transmitted in two time slot phases. First, during the transmitting time slot, the transmitter sends a single packet. Next, during a waiting time slot, the transmitter waits for the acknowledgement of the previously sent packet. If no acknowledgement is received, the transmitter tries to resend the unacknowledged packet, within the *maximum retransmission limit* (figure 11).

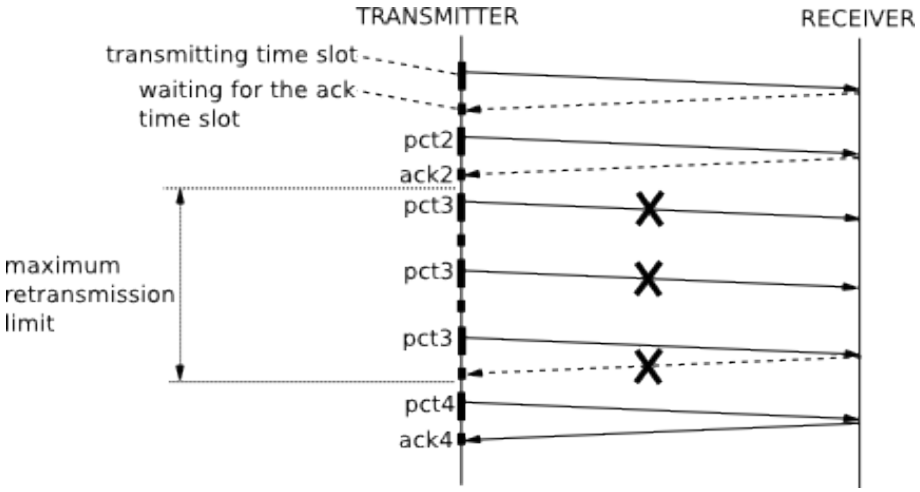


Figure 11: IEEE 802.11 transmission

5 Problem

The *slow-start* procedure assumes that unacknowledged packets are lost due to network congestion. While this is a reasonable assumption for many networks, packets may be lost for other reasons, such as poor *link layer* transmission quality. In this case, the *CWND* is reduced, resulting in a lowered throughput.

6 My proposal

The *link layer* is notified immediately about the packet delivered status, since the probability of lost *802.11* acknowledgement packets is low. I can use such information communicated to the *link layer* for creating a *TCP NACK* and informing the transmitter about the delivery failure. Such packets may be sent again without using the *fast-retransmit* and the *congestion control* mechanisms.

The agent on the *access point (AP)* side is involved in the creation of *TCP NACK* messages. This agent also interacts with the *transport* and *link* layers of the *OSI model* (figure 12).

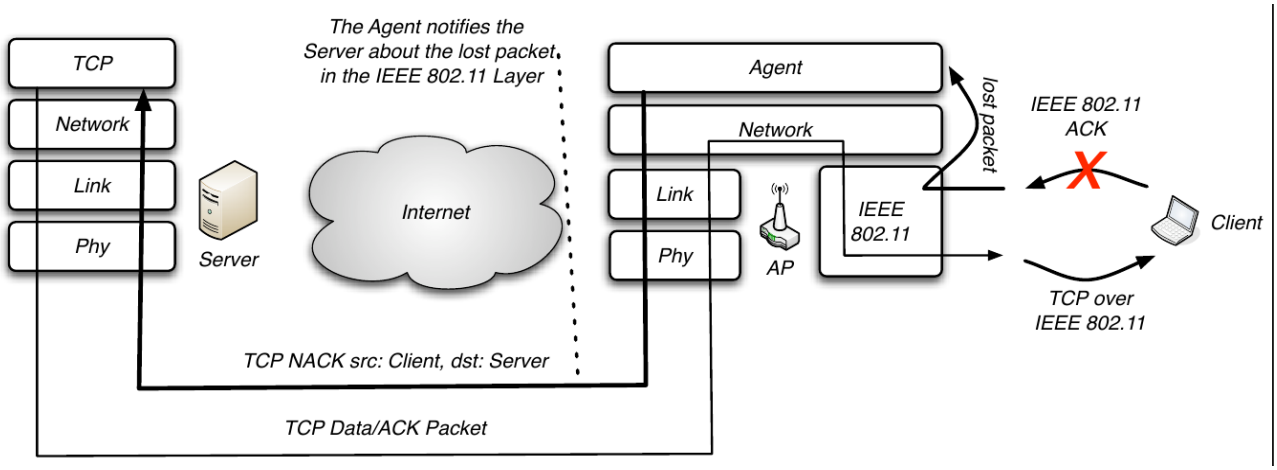


Figure 12: My proposal

7 Expected improvements

By identifying a scenario, in which the *CWND* windows does not need to be resized, I expect to improve network throughput. More specifically, there are two possible cases: (1) the *TCP NACK* is received either before (figure 13) or (2) after (figure 14) the *CWND* reduction happens. In both cases, the *CWND* size is saved after a packet is dropped.

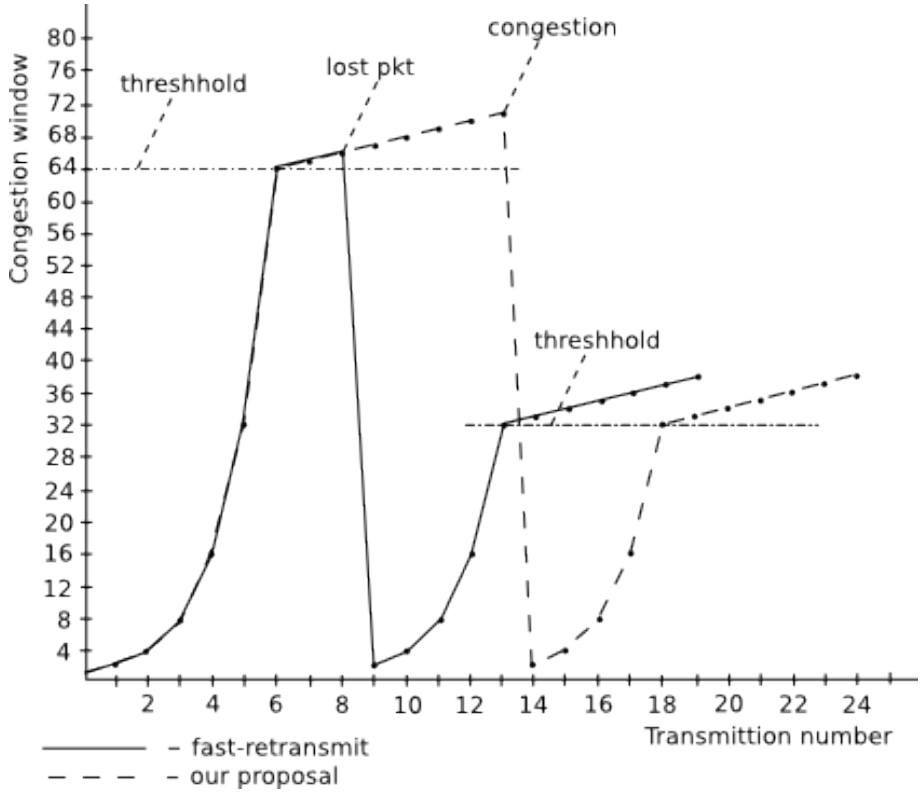


Figure 13: Expected CWND reduction after packet loss

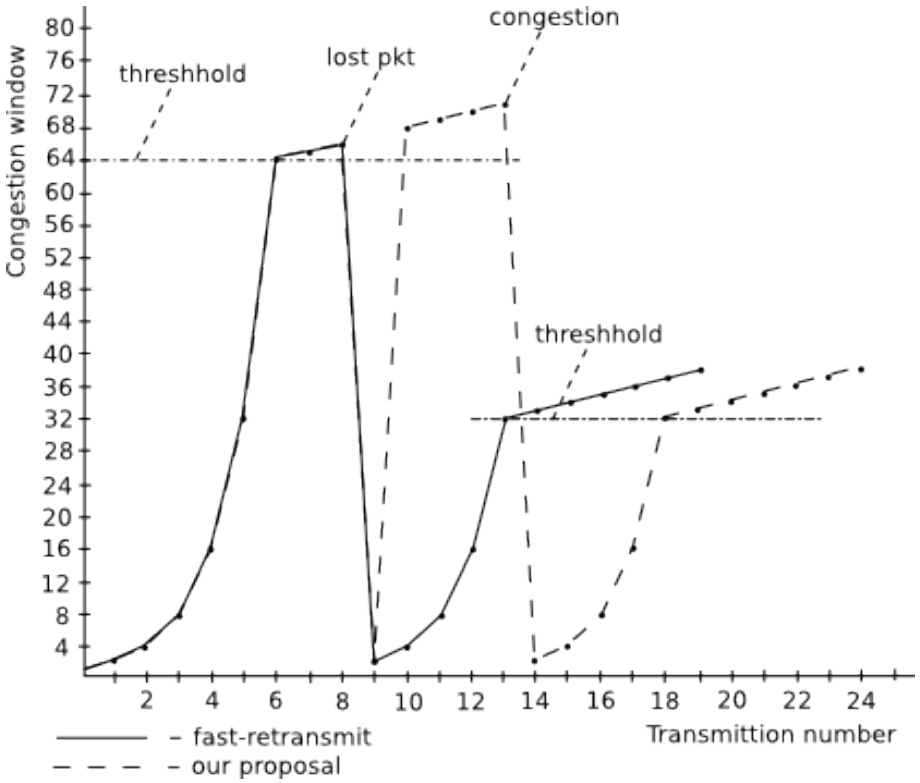


Figure 14: Expected CWND reduction before packet loss

References

[1] M. Handley, J. Padhye and S. Floyd, "TCP Congestion Window Validation," *RFC Editor, RFC 2861*, 2000.

[2] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *RFC Editor, RFC 2001*, 1997.

Enhanced Timestamp in SDR for Time-based Localization Systems

Zan Li
University of Bern
li@iam.unibe.ch

Abstract

This paper gives a general overview of the challenges that arise in using narrow-band signals, such as GSM, for localization based on the time properties of the signal. Specifically, synchronization and retrieving of time information are addressed.We pursue two contributions, namely, analysis of achievable synchronization precision and processing of narrowband signals that can enable timestamps down to nanoseconds.

Keywords: TDOA; Time recovery; synchronization.

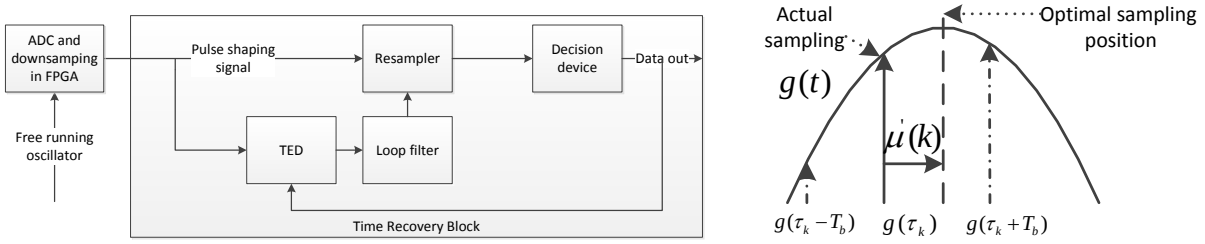
1 Introduction

Indoor localization has attracted more and more attention as GPS does not work well indoor. RSS (Received Signal Strength) and time-based localization are two common methods for indoor localization. Compared to RSS, time-based localization has a high potential to provide accurate localization services [2]. However, for time-based localization systems, accurate synchronization and time ranging are two well-known challenges. For time-ranging, Time of Flight (TOF) from the transmitter to receivers need to be measured in TOA (Time of Arrival) methods and Time Difference of Arrival (TDOA) among different ANs are needed in TDOA methods. One typical limitation in both cases is the difficulty to obtain a timestamp with high resolution and accuracy. In our work, Software Defined Radio (SDR) devices such as USRP [3] are used to capture the GSM signal and localize the mobile phone user. However, the resolution of the typical timestamp based on the clock of the FPGA inside the USRP is limited by the symbol rate of the signal [4]. We propose an enhanced timestamp based on signal reconstruction to obtain a nanosecond-resolution timestamp, which is able to distinguish the time difference within one symbol interval. Based on the proposed method, by comparing the enhanced timestamps of the same packet at different receivers, we are able to evaluate the synchronization accuracy between two devices, which are synchronized by a GPS signal down to nanoseconds. We then implement the proposed timestamp for accuracy time-ranging to obtain the Time Difference Of Arrival (TDOA). Based on some indoor measurements, we demonstrate that the TDOA based on the enhanced timestamp is sensitive to position changes of the transmitter but does not linearly change according to the distance difference because of the indoor multi-path propagation.

2 Accurate Timestamp via Time Recovery

Sampling at the right moment is critical for the good overall performance of the digital receiver and shift between the actual and optimal sampling position should be compensated for (as Figure 15(b)). To achieve that, the *time recovery* method [1] was developed to synchronize the sampler with the pulses of the received analog waveform. Figure 15(a) shows the architecture of a forward time recovery loop. The sample stream at the output

of the ADC is fed into a Timing Error Detection (TED) module to extract the timing error information between the current and optimal sample positions. The timing error is passed to a loop filter, which decides on the correction in the re-sampler. It can then adjust the sampling position to be closer to the optimum 15(b).



(a) Time Recovery (b) Optimal Sampling

Figure 15: Enhanced Timestamp

The output of the loop filter is denoted by $\mu(k)$ with k as sample number, to choose the correct sample and adjust the sample position in the re-sampler. It is given by

$$\mu(k) = \frac{\Delta T(k)}{T_s}, \tag{1}$$

where $\Delta T(k)$ is the offset between real and optimal sampling position, and T_s is the sampling interval.

The main responsibility of the time recovery method is to maximize the accuracy in signal reconstruction. At the same time, the timing error μ calculated in the time recovery loop can also be used to improve the resolution of the timestamp.

At the physical level a timestamp for the kth sample from the beginning of the sample stream can be obtained as follows:

$$T'(k) = T'(1) + T_s * (k-1), \tag{2}$$

where $T'(1)$ is the timestamp for the first sample in the stream. With this method, the resolution of the timestamp is limited by the sample interval. With the time error obtained by the time recovery, we can increase the resolution as follows:

$$T(k) = T'(k) + \Delta T(k) = T'(k) + \mu(k) \cdot T_s \tag{3}$$

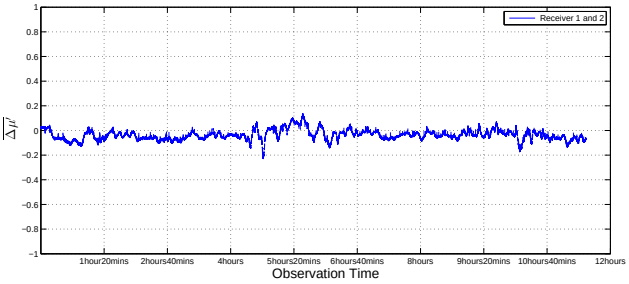
where $T(k)$ is the enhanced timestamp. Now, assuming that the kth sample is the first sample of the received packet, $T(k)$ is the enhanced timestamp for this packet.

3 Measurement of Synchronization and Time-ranging

Based on the proposed enhanced timestamp, we have taken some measurements to 1) evaluate the GPS synchronization and 2) evaluate the time ranging in an indoor environment.

3.1 GPS Evaluation

(a) Setup for Synchronization Measurement



(b) GPS Synchronization Evaluation Results

Figure 16: GPS Synchronization Evaluation

Figure 16(a) shows the setup used to calculate the synchronization offset between two GPS synchronized receivers listening to the source transmitter.

Assuming the hardware delays included in RF frontends and ADCs are the same between two receivers, two factors will cause some time offset of the same packet between two receivers, multipath propagation and synchronization offset. The two USRP receivers are co-located to ensure as much as possible the same propagation path of the signal. To minimize the influence from the surrounding environment (i.e., signal reflections), the distance between the transmitting USRP and the receivers is set to only $0.5m$.

Knowing the value of $\Delta\mu$, we can calculate the relative clock offset between the two receivers as follows:

$$C(t) = \Delta\mu(t) * T_s \tag{4}$$

where $C(t)$ is the relative clock offset, T_s is the sample interval ($2\mu s$ in our setup). Over the period of 12 hours, our findings show that the maximum registered clock offset between the two receivers is $423ns$ and most of clock offsets are within $200ns$ which is still very large for time-based localization, e.g., a corresponding positioning error of 60 meters is to be expected.

3.2 Time-ranging in Indoor Environment

Since the GPS synchronization is still not accurate enough for a time-based localization system, the measurement for time-ranging should eliminate the influence of the imperfect synchronization. Otherwise the synchronization offset will completely overcome the measured TDOA value. Therefore, we design a setup for this measurement as shown in Figure 17(a). In this measurement, the Reference Node (RN) should periodically transmit the packet to the Anchor Nodes (ANs) to compensate the synchronization offset. The RN and ANs are at fixed positions. The position of the object should be moved in each measurement.

In each measurement, once the position of the object is decided, the distances from RN to ANs (L_{R1} and L_{R2}) and from the object to ANs (L_{O1} and L_{O2}) can be measured. Then, the expected distance difference we can calculated as:

$$\overline{\Delta d} = (L_{R1} - L_{R2}) - (L_{O1} - L_{O2}). \tag{5}$$

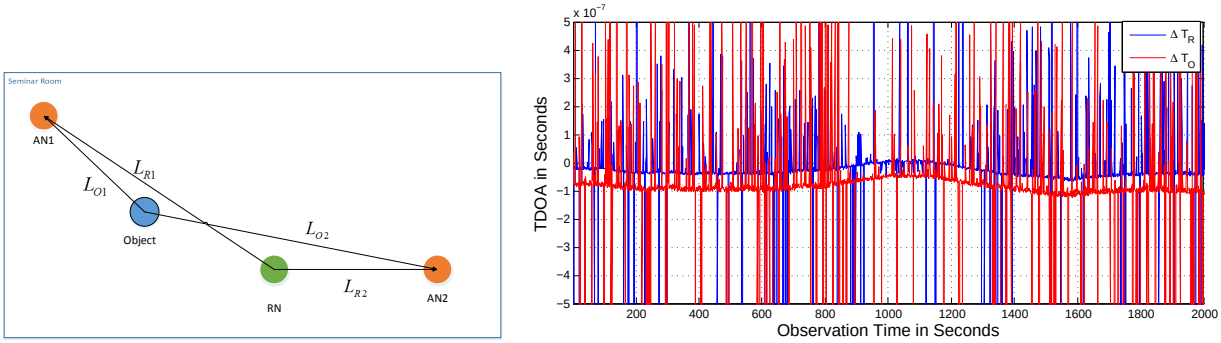
According to the timestamps for the packets from object and RN at two ANs, two TDOAs for object and RN can be calculated and denoted as ΔT_R and ΔT_O. Therefore, the measured distance difference based on the enhanced timestamp can be calculated as:

$$\Delta d = c * \Delta T_{RO} = c * (\Delta T_R - \Delta T_O), \tag{6}$$

where c is the speed of light and ΔT_{RO} is the Differential TDOA (DTDOA). Then, we can compare Δd with $\overline{\Delta d}$ to evaluate the time-ranging based on the enhanced timestamp.

Figure 17(b) indicates one example for the measurement results. We can find that the gap between the ΔT_R and ΔT_O is the value of ΔT_{RO} for 2000 seconds. The first row in Table 2 is the expected distance difference $\overline{\Delta d}$ and the second row is the mean value of Δd.

Based on the measurement results, we can find that 1) the measured distance difference based on the enhanced timestamp is sensitive to position changes. 2) If the position of the object is fixed, the measured distance difference is stable. 3) The measured distance difference does not exactly linearly follow the distance change, because it is influenced by multi-path propagation.



(a) Setup for Time-ranging Measurement

(b) Time-ranging Evaluation Results

Figure 17: Time-ranging Evaluation

Table 2: Time-ranging Based on Enhanced Timestamp

Expected distance difference ($\overline{\Delta d}$)	$0m$	$10.3m$	$12m$	$11.8m$
Measured distance difference (Δd)	$0m$	$14m$	$24m$	$45m$

4 Future Work

Since we find that the multipath propagation seriously influences the accuracy of the time-ranging, we are now focusing on channel estimation to model the indoor channel. Training sequences, which are widely used in wireless communication system to estimate the channel for GSM and LTE, are one of our options to estimate the channel.

In the future, we propose to first design an algorithm to compensate the biased time-ranging, which shifts from the real value caused by multipath propagation, in different channel conditions. This step can be first designed in a simulator such as MATLAB. Then we can apply the real channel model to compensate the biased time-ranging. For the case of synchronization, we propose to implement the DTDOA method, which applies a reference node to compensate the GPS synchronization offset.

References

[1] M. Heinrich, M. Moeneclaey, and Fechtel S.A.: Digital Communications Receivers: Synchronization, Channel Estimation and Signal Processing. John Wiley and Sons 1998

[2] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Volume 37, pages 1067-1080, Nov. 2007

[3] USRP N200/N210 networked series data sheet.

[4] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste. Enabling MAC Protocol Implementations on Software-Defined Radios. Proceedings of the 6th USENIX symposium on Networked systems design and implementation, pages 91–105, 2009

Distributed Object Oriented Programming for Wireless Sensor Networks

Yao Lu
École d'ingénieurs et d'architectes de Fribourg
Yao.Lu@edu.hefr.ch

1 Introduction

Wireless sensor networks (WSNs) constitute a new form of distributed computing where sensors are deployed for a specific application. However, it is very challenging to program and deploy WSNs applications. Programmers with knowledge of low-level systems are thus usually required, and the view of programmer has been shifted away from the application logic.

To simplify the burden of programmers, the macroprogramming paradigm, which targets the global behavior of the whole WSNs, could be used to provide high-level abstraction [1]. Nevertheless, it also has some constraints and limitations.

In order to automate nodal behavior, the special middleware should include components for coordinating sensors to dispatch tasks, and for aggregating data into high-level results. In addition, WSNs applications are closely coupled with conventional distributed applications, and it is possible to consider WSNs as an extension of a large distributed system. Whereas the great mass of previous works are focus on developing exclusive paradigm for WSNs application. If the application requires to manipulate heterogeneous computing resource, and single popular and existed programming paradigm could express all parts of the application, then the programming work could be considerably alleviated without using multiple paradigms.

2 Extending POP model into WSNs

We propose to extend the POP (Parallel Object Programming) model for WSNs [2]. The application can be abstracted into parallel objects, which are suitable to be distributed, and executed over heterogeneous distributed hardware, and able to interact. It means sensor functions could be abstracted into special resources, the attributes and methods of parallel objects could perform as WSN accessing interfaces. Furthermore, the separation of resource requirements and method invocation provides the possibility to isolate the nodal sensing function from the application task.

A large-scale WSN application is normally comprised of multiple subtasks. Because of common characteristics, parts of them could be abstracted as same type of objects. Due to this reason, programmers expect intensive manipulating multiple same type of objects. The concept of parallel object group in POP model coincidentally satisfies these demands. Whatever the parallel objects in a group located on different machines, they could be simultaneously operated by the group interface.

Given the restricted capability of sensors, the OO paradigm is too costly to be directly applied. This problem has already been addressed by POP developers in the previous preliminary work presented in [3] and which has shown very promising results to tackle this

issue. Gateway nodes have been used to bridge other nodes located in different network environments and are responsible for replying to callers after interacting with the local WSNs. The middleware of Gateway nodes performs as a network management system, because it is the final point to access these sensor resources. And the underlying runtime provides the necessary services to enable the execution of applications. In addition, the modification on syntax and semantics should be compliant with the POP paradigm as far as possible.

3 Preliminary experiment

In order to validate our proposal, we cooperate with Green-Mod project which aims to develop a WoT (Web of Things) gateway to bridge sensor network with IP network. This is the first example to demonstrate the possibility of extending the POP model to sensor network. The POP-C++ compiler has been modified to generate additional code to access sensor network [4]. The function of this matching mechanism between object attributes and sensor variables has been already substituted by the WoT gateway. When the methods are invoked to operate these attributes, the matching operations from additional code are executed targeting on specific sensors. The operations are packed and encoded into HTTP messages, which are subsequently transmitted to the server by socket. Gateway is responsible for transforming from HTTP request into sensor network request, and submitting request to specific sensors. Eventually sensors react in response according to the content of request.

4 Conclusion and Future work

Most WSNs applications are usually designed under the functional motivation of ‚Äúsense and send‚Äù. The new model could enable most of the common WSN applications based on OO paradigm, including simple data-collection applications that sense and actuate tasks with complex logic. At the next stage, the way to clearly and accurately expose the model will be considered. Meanwhile, the POP runtime needs to be enhanced to support heterogeneous application.

References

[1] L. Mottola, G. P. Picco, Programming wireless sensor networks: Fundamental concepts and state of the art, ACM Comput. Surv, vol.43, no.19, pp.1-46, 2011

[2] T.A. Nguyen, An object-oriented model for adaptive high-performance computing on the com-putational GRID, PhD thesis, EPFL, no. 3079, 2004

[3] A. B. Oliveira, L. F. Wanner, P. Kuonen, Integrating wireless sensor networks and the grid through POP-C++, In IESS, pages 411-420, 2007

[4] POP-C++,`http://gridgroup.hefr.ch/popc/doku.php`

Service-Centric Networking (SCN)

Dima Mansour
University of Bern
mansour@iam.unibe.ch

Keywords: Service-Centric Networking (SCN); Content Centric Networking (CCN); Services; Future Internet Architecture.

1 Introduction

The current Internet Architecture is built on the host-to-host communication model where each host has a name (identity) and an address (location). This architecture was well-suited for the early use case scenarios of the Internet but it introduced many limitations in current use cases[1]. Those limitations are related to security, mobility, and point-to-multipoint traffic (multicast). More importantly, we are witnessing a huge increase in the delivered content over the the Internet (iTunes, Youtube, etc) and nobody actually cares where that content is. The main concern in the current use of the Internet is to get the content, not where it is.

The use of the Internet has changed making researchers argue that named content is a better abstraction than named hosts for solving todays' communication problems. To tackle this issue, we introduce the concept of service-centric networking (SCN) [2] [3] where we identify services as the building blocks in the SCN communication paradigm. All messages in SCN are actually service requests and service replies in a location independent manner.

2 Content-Centric Networking (CCN)

Content-centric networking (CCN) [4] is a new and promising networking paradigm. CCN aims at moving from a host-to-host communication style to a new paradigm that focuses on content rather than hosts. CCN is about what content users want, not where content is. The goal of CCN is to achieve a network architecture that better suits the common use of networks today with respect to content distribution and mobility.

CCN is built upon a simple communication model with two different packet types. The first type is the Interest packet, which is sent by the content consumer to request the content of interest. The other type is the Data packet, which is sent by the content provider and has the data that satisfies the Interest packet.

Names in CCN are hierarchical, human readable, and appropriate for longest prefix match routing. CCN naming supports versioning by capturing the temporal evolution of the content.

However, CCN does not take into consideration the concept of services in its architecture and deals only with static content. We believe that services, rather than content, should be the center of focus in future network architectures. This is because content is just a subset of services and what applies to services can easily apply to content, but not the other way around.

The idea of CCN is a revolution in the networking world. But we argue that there is a limitation behind this idea, which is manifested when the requested content is a service but

not content. CCN supports static content, but it does not really support dynamic services. In this work, we want to generalize the CCN concept to cover dynamic services.

3 Service-Centric Networking (SCN)

SCN is a new paradigm for future Internet architecture. The main communication in SCN consists of service requests and service replies as correspondence to Interest packets and Data packets in CCN. This is mainly because the concept of services is a better abstraction than content for the future Internet architecture. Content is a special case of services when the service is "get" and can never cover for all communication scenarios when it comes to dynamic services (when the results depend on parameters).

One of the challenges of SCN is service naming and we discuss the issues and solutions in that regard in the following subsections.

3.1 Service Naming Issues

Naming in CCN is actually simple and straightforward if content is static and does not change. When it changes, versioning can cover for it. But if we think about services as functions or methods, we will have to deal with the concept of overloading. In other words, we have to deal with the following two issues:

- Services might have similar names but different signatures (Overloading). Service names are not enough to identify services because two services might have the same name but different numbers of parameters and/or different parameter types. For example:
  EvaluateExam (name, time)
  EvaluateExam (name, time, degreeOfLastExam).

- Service parameters and returned results might be of simple types (int, String . . .) or complex types (Objects). For example:
  boolean login(name:string, password: String)
  void register (obj:Person)
  Date getBirth(id:int).

3.2 Our Approach in Service Naming

To solve the name overloading issue, we assume that we should send the data types of parameters as a part of the service name to allow the router to differentiate between services. So, the Interest Packet will have the form *Interest(ServiceName;TypeOfParam1;TypeOfParam2;...)*,e.g., *Interest (cds.unibe.ch/getSeminar;Int;String;Date)*

The other important issue in SCN naming is how to send the parameter values to the service provider. We propose two ways to allow clients to send parameters:

1. The server asks the consumer for parameters in additional Interest packets. That means, for each parameter there is an Interest packet sent from the server and a Data packet sent from the consumer as an answer. When the service gets all the parameters, the server sends the data packet that contains the requested service data (return value).

2. The service requester sends the parameters within the Interest message represented as JSON, XML, or any other standard form. So in this case, the form of the Interest

message will be
Interest(ServiceName;TypeOfParam1;TypeOfParam2;....,parameter values)

4 Future Work

The idea of our work is to extend the idea of CCN to support services. We started by coming up with a naming convention for the services and we are going to implement it on top of the open source CCNx prototype implementation [5].
We will further discuss about service routing mechanisms to achieve optimal or near optimal routing for service requests. Then, we will evaluate these mechanisms on large-scale network scenarios. On the other hand, we will have to come up with a way to systematically decide where and when to deploy which services in the network and how these services will be adapted and terminated.

References

[1] The Future Internet Architecture Group, "Fundamental Limitations of current Internet and the path to Future Internet" June 2011, available at `http://www.future-internet.eu/uploads/media/FIArch_Current_Internet_Limitations_March_2011__FINAL_.pdf`

[2] T.Braun, A.Mauthe, and V.Siris,"Service-Centric Networking Extensions" *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp. 583-590, 2013.

[3] T. Braun,V. Hilt, M. Hofmann, I. Rimac, M. Steiner, and M.Varvello, "Service-Centric Networking ," *Communications Workshops (ICC), 2011 IEEE International Conference on*, pp. 1-6, 2011.

[4] V.Jacobson, D.K .Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L.Braynard, "Networking named content ," *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, pp. 1-12, 2009.

[5] "CCNx project" available at `http://www.ccnx.org/`

Secure Integrated Pub/Sub: Matching Scheme and Key Management Design

Emanuel Onica
Université de Neuchâtel
emanuel.onica@unine.ch

Abstract

Publish/subscribe (pub/sub) is an efficient paradigm of information dissemination often used by services running in today's cloud environments. One critical issue in such scenarios is the lack of trust in the cloud located brokers responsible of matching the flow of *publications* with *subscriptions* submitted by the clients of the service. We describe the desired characteristics of a secure integrated solution featuring a matching scheme and supporting key management. In the first part of our work we discuss design traits for a secure matching scheme. We consider the possibility of adapting techniques applied in database security. In particular we refer to a kNN query on a database, i.e., searching for the nearest k records that are closest to a query. We argue that any such secure query could be adapted to a pub/sub architecture following a set of common steps. As an already established use case we refer to the Asymmetric Scalar Product-preserving Encryption (ASPE) scheme [1]. ASPE features a data perturbation technique based on matrix multiplication also present in other possible adaptable designs. In the second part of our work, we discuss how to obtain a practical key management solution. For this we focus on a simple hierarchical design that leverages the benefits provided by Zookeeper, a service for coordinating distributed applications [2]. The main advantages of this approach are the synchronization support and ease of implementing a key exchange protocol. Other benefits are the native fault tolerance and the modularity permitting extensions for supporting more complicated key management designs.

Keywords: content based routing; key management; security.

1 Introduction

Publish/subscribe is an attractive communication paradigm used in large-scale application scenarios. Publish/subscribe offers a decoupled communication model between entities that act as either *publishers* of, or *subscribers* to, data. Subscribers inform the system about what data they wish to receive by registering a *subscription* to an overlay of *brokers*. Publishers send new events to the system in the form of *publications*. The brokers have the role to route each publication towards subscribers with matching interests, as expressed in their respective subscriptions. The set of the destination subscribers is the result of the *matching* operation for an incoming publication against the subscriptions stored by the brokers.

The publish/subscribe model in the context of our work is *content-based filtering*. In this, publications contain a *header* that includes a set *attributes* describing the publication contents. Subscriptions are composed of *predicates* specifying *constraints* over these attributes.

An often met example of content-based publish/subscribe use is that of a financial network supporting automated trading. Stock quotes form a flow of publications provided by a financial institution, each publication describing the quotation for a given stock. A publication header can include the symbol of the stock, the price, the volume of exchange, and

other information. Subscriptions express the interest of companies or private investors who set constraints on these attributes, wishing to invest or speculate on stocks.

In such scenarios, the actual publish/subscribe service can be outsourced to brokers situated in a public cloud. However, this leads to security concerns. Recent research [3] has exemplified possible exploits at the hypervisor level or based on virtual machines collocation, which can be used to obtain private information from a virtual machine running on a public cloud. Therefore outsourced cloud infrastructures are typically not considered trusted in relation to data confidentiality. In the described pub/sub architecture we can express this as the lack in trust in brokers. These manifest an honest-but-curious behavior, performing correctly their role in the system but leaving room for attempts to obtain knowledge about the data on which they operate. A solution for the problem is to encrypt the subscriptions and publications in such a way that the broker is still able to perform matching between them, but without accessing the publication headers or the subscription constraints values. Our work is focused on the design of such an encryption scheme and the related secure exchange of the information that has to be shared between the communicating peers.

2 Towards an encrypted matching scheme

One of the published works presenting a secure matching scheme [1] adapts to pub/sub the Asymmetric Scalar Product-preserving Encryption (ASPE), a technique initially used to secure kNN queries on databases [4]. Our research focused on introducing a prefiltering technique that alleviates the performance drawbacks of ASPE, and also on studying the potential to adapt other schemes from the secure database field to pub/sub systems. In the following we describe some of the advances obtained in the latter, which are also based on and extend the work in [1].

A kNN query on a database searches for the nearest k records $p_1, ..., p_k$ that are closest to a query q. The database records and the query can be modeled as points in a d-dimensional space, where each dimension corresponds to a field in the database schema. The relation of being closer is modeled by comparing the Euclidean distance between a first point and a second point (e.g., $dist(q, p_1)$) with the Euclidean distance between the first point and another third point (e.g.,$dist(q, p_2)$). In particular for a kNN query the distance is evaluated between a point of interest - the query, and each other point - the records, and selecting the k-closer results to the query.

The subscriptions and the publications in a pub/sub system can be modeled similarly as points in a d-dimensional space. Unlike the kNN query case, the interest is not to evaluate how close is a point to another one, but the exact relation $(>, <, =)$ between a dimension of a point (the subscription) and the same dimension in another one (the publication). A building block in the previously described database kNN query is a result of distance comparison between three points. This principle does not take into account any relation between particular dimensions. To extend the previous example, points $p_1 = (1, 2)$ and $p_2 = (3, 0)$ are equally close to point $q = (2, 1)$ and are not distinguishable in a kNN query result towards q.

Let us take as a particular example the case where all the corresponding dimensions in each point are equal, except one dimension d_i. More, let us consider that we know the relation between the values corresponding to d_i in p_1 and p_2, $(<, >, =)$. In this particular setting, the result obtained in the kNN query when comparing $dist(q, p_1)$ and $dist(q, p_2)$ is sufficient to obtain, in a false positive exclusive manner, further information about the relation with the value corresponding to d_i in q. It is trivial to observe the exclusive factor,

since the only dimension that influences the comparison between $dist(q, p_1)$ and $dist(q, p_2)$ is d_i. In other words the three points are on an axis determined by the other dimensions where the only variating value is the one corresponding to d_i.

To extend our particular case, we have formally proved that it is enough only for points p_1 and p_2 (and not necessarily for q) to have all the corresponding dimensions equal except d_i. In brief, this is based on the fact that the relation of the comparison between $dist(q, p_1)$ and $dist(q, p_2)$ is preserved when q is projected on the axis determined by p_1 and p_2.

The idea of applying this property in a matching scenario is to derive in a first step two reference points p_1 and p_2 as above for each dimension of interest in a subscription, such as the value of the dimension is at the middle of the interval [p_1,p_2]. Then, the distance comparison with a publication q can be evaluated, which determines the matching result. Our position is that such a technique can be applied for any secure scheme that permits the distance comparison between three points in the manner previously described. We acknowledge the work in [1] which already applies this to obtain encrypted matching in pub/sub scenarios. Their discussion is though mostly targeted on the single dimension case and is centered around the ASPE scheme. We extended their work through a mathematical formalism that generalizes a model potentially applicable for other schemes and takes also in account extensive issues like the secure coverage between multidimensional subscriptions.

3 Key management design

Using a secure scheme such as ASPE or others, requires the exchange of key related information between the trusted publishers and subscribers in order to encrypt the registered subscriptions and the publication submitted in the pub/sub system. Moreover, for security reasons, this key requires updates at certain times (e.g., a change of trust in the set of participating peers). Since publish/subscribe is a decoupled paradigm in which we do not know the exact sender and the destination it is apparently difficult to provide a key management solution that will fit the system architecture.

Our ongoing work approach overcomes this issue by considering entire trust groups as members of a global group serviced by a key management coordinator. Although the *exact* sender and receiver of a publication cannot be determined a-priori to the matching, a trust group formed of publishers that will disseminate information targeted by trust groups formed of subscribers can be known (e.g., the group formed of individual workstations in a company that subscribes for stock quote information to a stock market branch). Our current solution follows this setting, by establishing a hierarchical key dissemination protocol on three levels: the top tier formed by a key management group coordinator, a middle tier formed by trust groups coordinators and the bottom tier formed by the individual subscribers, publishers and brokers. In the development of this protocol we make use of Zookeeper [2], a coordination service for distributed applications. This allows distributed processes to coordinate with each other through a shared hierarchical namespace. An important feature that we use in the key dissemination is the ability of a Zookeeper client to monitor a node in this namespace in order to obtain information about a change, which in our case reflects the events generated by a key update.

One of the most challenging parts of a key management implementation for pub/sub is the fact that a key update invalidates the encrypted subscriptions stored by a broker. This can cause a major disruption in the service since there is a need to resubmit all the subscriptions encrypted with the new key. We overcome this problem through the means of a secure transformation that allows the update of the stored subscriptions directly by the

broker itself. ASPE [1] seems to permit such a transformation and part of our ongoing and future work is focused in finding and analyzing other schemes that fit a similar characteristic.

References

[1] S. Choi, G.Ghinita, and E. Bertino, “A privacy-enhancing content-based publish/subscribe system using scalar product preserving transformations", *Lecture Notes in Computer Science*, vol. 6261, pp. 368–384, 2010.

[2] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, “ZooKeeper: wait-free coordination for internet-scale systems", in *Proc. of the 2010 USENIX conference on USENIX annual technical conference*, 2010, p. 11.

[3] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off my cloud: exploring information leakage in third-party compute clouds", in *Proc. of the 16th ACM conference on computer and communications security (CCS)*, 2009, pp. 199–212.

[4] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, “Secure kNN computation on encrypted databases", in *Proc. of the 35th SIGMOD international conference on management of data*, 2009, pp. 139–152.

Performance Evaluation of Robustness for an Opportunistic Routing for Video Transmission in Dynamic Scenarios

Denis Rosário
Universität Bern and Federal University of Pará
rosario@iam.unibe.ch

Abstract

Mobile multimedia nodes expand the Internet of Things (IoT) portfolio with a huge number of multimedia services. Those services run on dynamic topologies due to node mobility or failures, as well as wireless channel impairments. For such mobile multimedia IoT applications, a robust routing service must adapt to topology changes with the aim of recovering/maintaining the video quality and reducing the impact of the user's experience. In this paper, we assess the robustness and reliability of our recently developed OR protocol, called cross-layer Link quality and Geographical-aware OR protocol (LinGO), for efficient video transmission under failures in mobile multimedia IoT scenarios. Simulation results show that LinGO is able to adapt to network and topology changes without increasing the signalling overhead in case of topology changes. Hence, LinGO achieves multimedia dissemination with QoE support and robustness.

Keywords: Multimedia transmission; QoE support; Robustness.

1 Introduction

Beaconless Opportunistic Routing (OR) allows increasing system robustness to support routing decisions in a completely distributed manner based on protocol-specific characteristics. Moreover, the addition of a cross-layer scheme could enhance the benefits of beaconless OR and enable multimedia dissemination with Quality of Experience (QoE) support. However, existing beaconless OR approaches do not support a reliable and efficient cross-layer scheme to provide effective multimedia transmission under topological changes, increasing packet loss rate, and thus reducing the video quality level from the user's experience.

In specific terms, given a pair of source and destination, the goal of beaconless OR is to find a subset of forwarders to create a connected path. In particular, the forwarding selection mechanism must find a subset of optimal forwarders to provide packet delivery guarantees. The optimal forwarder must provide greater progress towards the destination with reliable links and enough energy to forward video packets with low packet loss rate.

Additionally, a compressed video is composed of I-, P-, and B-frames with different priorities, and from the human visual system's standpoint, the loss of high priority frames causes severe video distortions. The loss of an I-frame causes error propagation through the other frames within a Group of Picture (GoP), since the decoder uses the I-frame as a reference point for reconstruction of all the other frames within the same GoP. Thus, the video quality only recovers when the decoder receives an unimpaired I-frame. For the loss of a P-frame, the impairments extend to the remaining frames within a GoP, and also the loss of P-frames at the beginning of a GoP causes higher video distortion than loss at the end of a GoP. The loss of a B-frame only affects the video quality of that particular frame.

By tackling the above issues, we contemplate a feasible cross-layer solution by integrating the application and network layers. A cross-layer solution also helps to achieve robust and reliable multimedia dissemination with high video quality level from a user's experience.

We assessed the efficiency, reliability, and robustness of our OR protocol, called LinGO (cross-layer Link quality and Geographical-aware OR for video transmission in mobile multimedia IoT environments) under dynamic scenarios [1]. The assessment takes into account node failures, as well as changes in the wireless links.

2 A Cross-layer Link quality and Geographical-aware beaconless OR (LinGO)

LinGO relies on a QoE/video-aware redundancy mechanism to add redundant packets based on frame importance, enabling robust and efficient multimedia transmission with reduced overhead. In addition, LinGO has two operational modes and uses multiple metrics to establish a reliable virtual backbone to provide video distribution with QoE support.

The QoE/video-aware redundancy scheme [2] achieves robust video transmission over a bandwidth-limited unreliable networking environment, as experienced in mobile multimedia IoT environments, since it uses Reed-Solomon coding to create redundant packets. It adds redundant packets only for priority frames based on user's experience, i.e., not in a black-box manner as it happens in many non-QoE approaches. Thus, it protects the priority frames in congestion/link error periods, and supports QoE-aware multimedia transmissions together with reduced packet overhead compared to existing redundancy mechanisms.

Whenever a source wants to send a video sequence, it broadcasts the data packet, and the possible forwarders compute a delay timer based multiple metrics, i.e., energy, geographical information, and link quality, before forwarding the received packet. The node with the best conditions to forward the packet generates the shortest delay, and thus rebroadcasts the packet first. The neighbour nodes recognize the relaying, and cancel their scheduled transmission for the same packet. In this way, LinGO builds a reliable backbone between the source and destination node via multiple forwarder nodes. As soon as the nodes established the backbone, they must forward subsequent video packets explicitly addressed to the selected forwarder node and without include additional delay.

We have performed massive omnet++ simulations to analyse the impact of temporary or permanent node failures on the video quality level. After analysing the results, we identify that the existing beaconless OR protocols perform poorly compared to LinGO, since LinGO establishes a reliable virtual backbone by using multiple metrics. These are features desirable for many mobile multimedia IoT applications, such as safety & security, environmental monitoring, and natural disaster recovery.

3 Future Work

For future work, we are planning to reduce the impact on the video quality level of node mobility by detecting when the links are broken. To achieve this, the nodes must send acknowledgments for a set of n received packets, and if m acknowledgments are lost, we can conclude that the link is not valid any more. Thus, nodes must re-establish the virtual backbone. In addition, in case of buffer overflow, the nodes have to drop packets according to frame importance and/or packet deadline.

References

[1] Z. Zhao, D. Rosário, T. Braun, E. Cerqueira, H. Xu, L. Huang, Topology and link quality-aware geographical opportunistic routing in wireless ad-hoc networks, in the 9th IEEE International Wireless Communications & Mobile Computing Conference, 2013.

[2] Z. Zhao, T. Braun, D. Rosário, E. Cerqueira, R. Immich, M. Curado, QoE-aware FEC mechanism for intrusion detection in multi-tier wireless multimedia sensor networks, in the 1st International Workshop on Wireless Multimedia Sensor Networks (WiMob'12 WS-WMSN), Barcelona, Spain, 2012, pp. 697–704.

SplayNet: Distributed User-Space Topology Emulation

Valerio Schiavoni
Université de Neuchâtel
valerio.schiavoni@unine.ch

Abstract

Network emulation allows researchers to test distributed applications on diverse topologies with fine control over key properties such as delays, bandwidth, congestion, or packet loss. Current approaches to network emulation require using dedicated machines and low-level operating system support. They are generally limited to one user deploying a single topology on a given set of nodes, and they require complex management. These constraints restrict the scope and impair the uptake of network emulation by designers of distributed applications. We propose a set of novel techniques for network emulation that operate only in user-space without specific operating system support. Multiple users can simultaneously deploy several topologies on shared physical nodes with minimal setup complexity. A modular network model allows emulating complex topologies, including congestion at inner routers and links, without any centralized orchestration nor dedicated machine. We implement our user-space network emulation mechanisms in SplayNet, as an extension of an open-source distributed testbed. Our evaluation with a representative set of applications and topologies shows that SplayNet provides accuracy comparable to that of low-level systems based on dedicated machines, while offering better scalability and ease of use.

Keywords: Topology emulation, large-scale networks, testbeds.

1 Introduction

A key aspect of distributed systems evaluation is the capacity to deterministically reproduce experiments and compare distributed applications in the same deployment context, and in particular when operating under the same network conditions. Distributed testbeds such as PlanetLab (`www.planet-lab.org`) allow testing applications in real-world conditions, by aggregating a large number of geographically distant machines. While extremely useful for large-scale systems evaluation, such testbeds cannot be reconfigured to expose a variety of network infrastructures or topologies. Furthermore, the high load and the unpredictable running conditions of shared testbeds are a hindrance for the reproducibility of evaluation results, or for the fair comparison of different applications.

Network emulation supports *controllable* and *reproducible* distributed systems evaluation. It allows running a distributed application on dedicated machines as if it were running on an arbitrary network topology, and observe the behavior of the application in various network conditions. The emulation of communication links is based on an input *topology*, i.e., a graph representation of nodes, routers, and the properties of their connections. A cluster with a high-performance local network can typically support the execution of applications and the emulation of topologies.

The focus of this work is on providing support for easy evaluation of networked applications (e.g., indexing [1], streaming [2], coding [3], data processing over non-standard topologies [4], etc.) under diverse yet reproducible networking conditions. Furthermore, we seek to provide support for concurrent deployments of emulated topologies and distributed applications, where the physical nodes of a cluster can be used for running multiple experiments with different topologies, without interference and loss of accuracy for any of

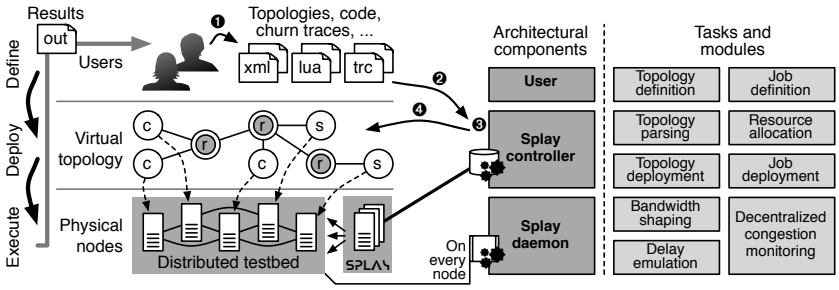


Figure 18: The SplayNet architecture.

the experiments. Note that our work focuses on the evaluation of networked applications on top of standard TCP and UDP connections, when presented with various end-to-end characteristics: bandwidth, delay, packet loss, and congestion.

2 SplayNet Architecture

In this section we briefly describe the various components supporting our SplayNet prototype. Figure 18 presents an overview of the architecture.

Users write an abstract description that maps vertices and edges of an undirected cyclic graph to the physical connections of a network (Figure 18-❶). Users can specify the interconnections between nodes and routers, as well as the physical properties of the links (delays, bandwidth, and packet loss rate). The second step is the deployment (Figure 18-❷). The user submits to SplayNet the topology description, the code to execute, and any additional files required to drive the experiments. SplayNet's topology parser extracts the graph topology. SplayNet allocates testbed resources for executing the user code on the emulated topology (Figure 18-❸). In the context of SplayNet, this problem corresponds to selecting a minimal set of `splayd`s for executing the job. The allocation procedure ensures that the deployment of a topology does not impair on the accuracy of other deployed topologies, by avoiding saturating the bandwidth of physical links beyond a safety margin. Finding a *minimal* set that satisfies all constraints on a shared infrastructure is a NP-hard problem [5]. In SplayNet, we adopt a simple greedy approach to guide the selection of `splayd`s. Finally, the system dispatches the code along with the topology information required to initialize the network emulation layer to be executed to the selected `splayd`s (Figure 18-❹).

3 Evaluation

This section presents a preliminary evaluation of SplayNet, in particular with respect to latency and bandwidth emulation. We set up a SplayNet cluster on top of a 1 Gb/s switched network with 60 machines, each with 8-Core Xeon CPUs and 8 GB of RAM.

Latency. To evaluate the accuracy of link latency emulation, we deploy a simple client-server application using remote procedure calls (RPCs) at the edges of a point-to-point topology. Figure 19.top presents the cumulative distribution function (CDF) of observed delays.

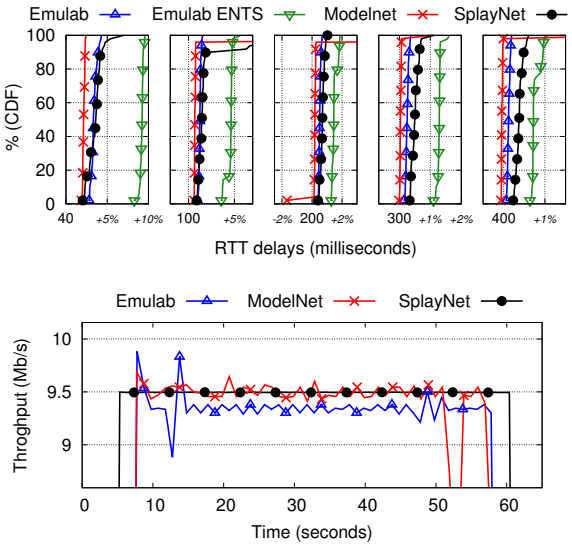


Figure 19: Latency and bandwidth accuracy.

The expected RTT is shown on the y-axis for each of the link latency values, with variations expressed as percentages. Performance over three testbeds (Modelnet [6], Emulab [7] and SplayNet) is very similar (the emulated latencies never deviate more than 10% from the expected values. **Bandwidth.** We deploy the same topology to evaluate the accuracy of the bandwidth emulation. The client node continuously streams data over a 10 Mb/s link over a TCP connection. Figure 19.bottom shows how the three systems let the application-level data stream saturate the available link bandwidth up to the theoretical limits.

4 Conclusion

Network emulation allows researchers to evaluate distributed applications by deploying them in a variety of network conditions. Previous solutions often relied on dedicated machines to shape the network traffic across the nodes involved in an experiment, and did not allow the concurrent deployment of different network topologies on the same nodes of a testbed. This extended abstracted introduced the architecture and some of the mechanisms in SplayNet, an integrated user-space network emulation framework. The preliminary evaluation shows that SplayNet offers equivalent performance to state-of-the-art systems, both in terms of latency emulation and bandwidth shaping accuracy. The most direct perspective to this work is the integration of wireless and mobile emulation capabilities, as well as the support for non-fully connected and dynamic topologies. SplayNet could then allow experimenters to emulate protocols for sensor and ad-hoc networks, for which building a physical testbed is a complex and costly operation.

References

[1] Ion Stoica and Robert Morris and David Liben-Nowell and David R. Karger and M. Frans Kaashoek and Frank Dabek and Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications," *IEEE Transactions Networking*, vol. 11, no. 1, pp. 17-32, 2003.

[2] Biersack, Ernst W and Rodriguez, Pablo and Felber, Pascal, "Performance analysis of peer-to-peer networks for file distribution," *Quality of Service in the Emerging Networking Panorama*, pp. 1-10, 2004.

[3] Gkantsidis, C. and Rodriguez, P.R., "Network coding for large scale content distribution", *IEEE INFOCOM*, pp. 2235-2245, 2005.

[4] Costa, Paolo and Donnelly, Austin and Rowstron, Antony and O'Shea, Greg,"Camdoop: exploiting in-network aggregation for big data applications", *USENIX NSDI*, 2012.

[5] Ricci, R. and Alfeld, C. and Lepreau, J.,"A solver for the network testbed mapping problem",*SIGCOMM Comput. Commun. Rev.*,vol.33, n.2, pp. 65-81,2003.

[6] A. Vahdat and K. Yocum and K. Walsh and P. Mahadevan and D. Kostic and J. Chase and D. Becker, "Scalability and accuracy in a large-scale network emulator", *USENIX OSDI*, pp. 271-284, 2002.

[7] Brian White and Jay Lepreau and Leigh Stoller and Robert Ricci and Shashi Guruprasad and Mac Newbold and Mike Hibler and Chad Barb and Abhijeet Joglekar,"An Integrated Experimental Environment for Distributed Systems and Networks", *USENIX OSDI*, pp. 255-270, 2002.

On Semantic Integration of Physical Business Entities into Enterprise IT Systems

Matthias Thoma
University of Bern
thoma@iam.unibe.ch

Abstract

Cyber-physical or Internet of Things based systems are anticipated to gain widespread use in industrial applications. Standardization efforts, like 6LoWPAN and CoAP have made the integration of wireless sensor nodes possible using Internet technology and web-like access to data (RESTful service access). We present a platform for integration of real-world objects into enterprise systems and enabling interoperability using semantic web technologies. We introduce an abstraction of real-world objects, called Semantic Physical Business Entities (SPBE), using Linked Data principles. This allows a business object centric view on real-world objects, instead of a pure device centric view. A prototype implementation was used to perform a quantitative analysis.

Keywords: Smart Objects, Internet of Things, Semantic Enterprise

1 Introduction

In the enterprise community real-world aware business models and software innovations gained a lot of momentum recently. One of the main obstacles in enterprise integration is still the gap between the specialized knowledge needed to program the sensor nodes and add them to an enterprise backend system, and the (business) model driven way enterprise software is written and customized. In IoT research, one of the main observations is, that people and thus many business processes are interested in the things, the properties of the physical objects, and they are not at all interested in sensor readings. The main research goal is to find novel ways enabling programming of WSN applications, which integrate into how enterprise software is written and cusomized today. Furthermore, interoperability and automatic code generation at a semantic level is researched enabling rapid application development and interoperability beyond simple schemas. Below the semantic interoperability layer there need to be build some application layer protocols. Currently, in the IoT community CoAP [4] is discussed, while in the enterprise world the OData [5] protocol is emerging.

The lower layers of the IoT-stack have been under intense research in the past, have reached a sufficient maturization level which lead to standardization (e. g. 6LoWPAN), while the upper layers just started to gain widespread attention. The three main blocks of interest to us and driving our research are:

Application Layer Protocols One design goal is to be as protocol agnostic as possible. Therefore not only standardized protocols like CoAP or HTTP should be supported, but also custom protocols, as a "one-fits-it-all" solution is not likely to exist. Additionally, further application layer protocols are to be investigated. One candidate, as an alternative to CoAP and HTTP would be OData [5].

Interoperability On the interoperability level the goal is to go beyond traditional binary based protocols, where interoperability is achieved through documentation and to some

degree through descriptive elements, like they exist in XML documents. We suggest to use semantic service descriptions (like SAPs Linked USDL [3]), which use semantic web technologies to describe the input, output and effects of a service call.

Programmability Sensor Network applications are still cumbersome to write and maintain. Very specialist knowledge is necessary and custom programming languages are used, for instance nesC. Here the goal is to explore what a "good" programmability means for such applications and how to do business process decomposition. Business process decomposition means breaking a business process into smaller pieces and run parts of it on motes. This could include the use of Service Oriented Architecture, complex event processing via rule engines or the use of standard languages like Java or C#.

For the understanding of the integration platform it is necessary to understand the concept of Semantic Physical Business Entities, which we define as follows: *A Semantic Physical Business Entity is a conceptual representation of a real-world object processed by one or more enterprise IT systems. Information about it is discoverable. It is described through well defined vocabularies, that make internal and external relationships explicit.*

Important to note is the decoupling between the SPBEs and the devices observing it. An enterprise system's user is usually not interested in the value of sensor no. 0815, but in the temperature of some given good, which could be monitored by several sensors. This abstraction, moving away from the pure technical view concentrating on sensing devices, towards the "things" they monitor is one of the main ideas in the the IoT community.

2 Integration Platform

In this section we will briefly present our integration platform. The main ideas of our envisioned framework are shown in Figure 20. We will briefly discuss each layer. The main research so far has been done on the integration layer and the semantic layer.

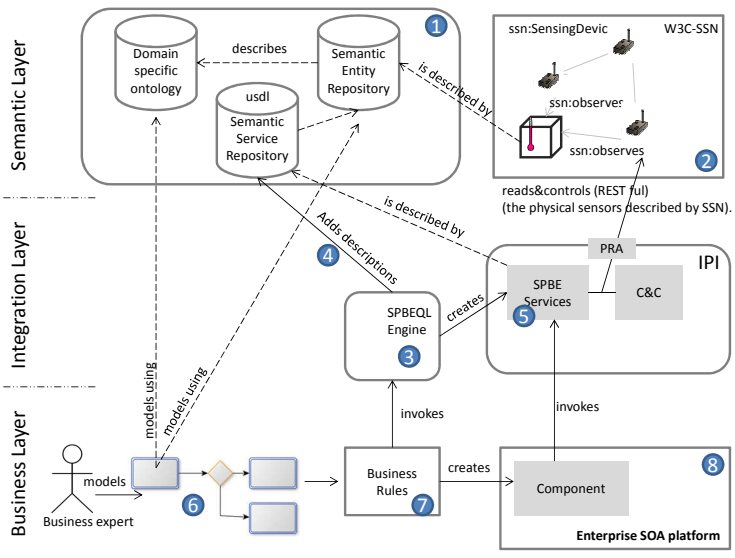


Figure 20: Integration Platform Framework

As shown in the Semantic Layer (1) our system compromises various semantic repositories. As far as possible existing vocabularies, like the SSN ontology, are used (2). We aim towards supporting interoperability by using linked service descriptions. We extended Linked USDL, a semantic service description language developed by SAP, towards supporting multiple technical endpoints (e. g. WSDL and COAP) and added support of different communication patterns (Request and Response, Publish/Subscribe and Time triggered). We added an event ontology to support the publish/subscribe pattern. The technical interfaces can be transferred into each other via a graph rewriting algorithm. This happens mainly on the integration platform instance (IPI), that connects the edge mote to the enterprise network.

To support the SPBE abstraction we created a query language called SPBEQL, with an SQL like syntax. The advantage is, that SPBEs can dynamically be created and accessed during the runtime of the system. An a-priori knowledge of the SPBEs is not required. Furthermore any kind of sensing source can be used, even if they are just data pushed to the internet. All domain knowledge is encoded in the semantic repositories and can change at any time. As an example of how SPBEQL is used the following instructions show how to create a service endpoint for getting the temperature of a given temperature zone:

```
CREATE SERVICE PUSH FOR
  SELECT AVG MIN MAX temperature FOR SPBE e WHERE
    e:hasLocation:city = "Zurich" AND e:is_an = "temperature zone"'
    WITH CollectionInterval = 30s AND CollectionTime = 60s
```

The system includes a placement heuristic, which automaticaly places the services on the motes. It supports not only a simple sense and send (S&S) approach, but is able to do aggregation. We showed that aggregation on the motes is able to save energy compared to a sense and send approach. The service creation is parameterizable. It supports a PUSH mode, where data is regularly send to the top-level SPBE endpoint and a pull mode where data is only gathered upon request.

Compared to other SOA approaches, we are not utilizing a middleware for gathering data, but automatically generates code and deploys it to the motes, thus reducing the gap between the high-level application designer and the low-level hardware programming (3). Furthermore service descriptions for these service are generated and stored in the service repository making them available to a SOA based enterprise platform (4).

The business expert formulates the process in a modelling language (6). Based on that, the system creates business rules (BR) (7). These BRs trigger the generation of SPBEQL, which in turn generates code to be deployed on the motes (5) and on the IPIs (8).

3 Preliminary Results

We conducted a qualitative and quantitative analysis of the framework using a prototype implementation. We conducted several experiments and measured the following parameters for the platform using the generated code and service descriptions for a temperate service automaticaly adapted using graph rewriting. In this experiment we measured the system setup time, the size of different endpoint descriptions (in bytes) both compressed and uncompressed, for an equivalent temperature service. Furthermore, the energy consumption for different endpoint technologies (CoAP, Reliable UDP, UDP and MAC protocol only) with different payload encodings (payload encoded in ASN.1 and as RDF), the memory footprint of the different generated service endpoints and the service response times for service endpoints placed on the motes themselves or the IPI.

In another series of experiments we evaluated the propoerties of the different types of services (**D**ata **G**athering **S**erivce, **D**ata **A**ggregation **S**ervice, and **SPBE** Endpoint) SPBEQL is able to create. In this second series of experiments we measured the system setup time, the memory footprint, and the energy consumption of the system in push mode and pull mode.

An evaluation of the *size of different endpoint descriptions* for a temperature service, showed that a semantic service description consumes less bytes than an equivalent description in WSDL. The results get closer to each other when considering larger services, as the compression of WSDL then plays a more important role than for smaller services.

A measurement of the *energy consumption for different endpoint technologies* shows that the main driver is the transport layer protocol and below. The influence of the application layer and the payload encoding is minimal. 6LoWPAN contributes most to the energy consumption, as can be seen in the difference between UDP/CoAP and the MAC level communication. The additional energy needed for the CoAP layer is less than 1%.

We measured the *Service Access Time* (SAT) that means the time from issuing a service request by the service consumer until the response has been decoded. As benchmark we used a CoAP-based REST style protocol on the mote as well as just sending the temperature and the time stamp. We then measured SAT times for (i) direct access on the same machine via IPv6, (ii) remote access from a different machine over the local network over IPv6, (iii) Remote Access via generated SOAP and (iv) HTTP REST interfaces. Direct access is the fastest method, directly followed by access over the local network. REST access over http was slightly faster than SOAP due to reduced stack complexity.

We measured (in simulation) the energy profile of the system in pull mode, in push mode, and for a S&S counterpart with all motes sending data to the IPI. The aggregation approach proofed to be more energy efficient, as the computation time for the aggregation needed less energy than the additional transmission and receive cycles needed in the S&S approach. Pull is more energy efficient, for small request rates, for the price of higher response times. The energy cost of aggregation on mote for 8 pulls/10min was 24134mAs compared to 12927mAs for sensing. RDF on motes takes more time and energy and therefor should be avoided.

We did some first experiments on combining sensor reading from actual sensing devices attached to IPIs with external RDF data as sensor readings. Fort his, we setup a second IPI with a WSN attached to it. Instead of adding it the platform directly we publish the sensor readings as RDF data to a local server and to a server on the internet. We added them as external RDF sensor sources and measured the service response times. In push and pull mode with an endpoint on the IPI, the influence on the system is negliable, both for local and remote data sources as long as the collection time is large enough to compensate the latency. As soon as the WSN responds faster than the external source the picture changes.

4 Future Work

The actual realization of the business layer is an open research question. We plan to investigate how sensor data from external sources (sensor web, linked open data) and internally sensed data from sensing devices can be combined to realize a truly sensing enterprise.

Interestingly, there is not much information available on the pros and cons of how WSN applications are written today. Therefore, it would be interesting to see if Java for programming sensor nodes is a suitable choice, or if more proprietary languages like nesC are still the better candidates. Additionaly, it is still an open question how domain experts should program sensor networks. Some research in the area of BPMN and BPEL has already been conducted, for example Glambitza et al. [6]. As an alternative to the BPEL or direct compilation of BPMN approaches we want to investigate rule engines: Can rule engines be used for WSN programming, like BPEL today? Furthermore, it is an interesting question if ontologies can be used to mine rules, that are then automatically transferred into code.

References

[1] Internet of Things Architecture (IOT-A) Retail Demonstrator, presented at IoT-Week 2013, June 17th-20th, Helsinki.

[2] Thoma, Meyer, Sperner, Meissner, Braun, “On IoT-services: Survey, Classification and Enterprise Integration”, iThings 2012, December 10, 2012.

[3] SAP et. al, “Linked USDL”, http://www.linked-usdl.org

[4] CoRE Working Group, “Constrained Application Protocol (CoAP)”, IETF Draft, 2013

[5] Microsoft Corporation, “Open Data Protocol (OData) Specification”, 2013. Online available: http://www.odata.org/documentation/odata-v3-documentation/

[6] Glombitza, Pfisterer, Fischer. “Integrating wireless sensor networks into web service-based business processes.” Workshop on Middleware Tools, Services and Run-Time Support For Sensor Networks. ACM, 2009.

Distributed programming using POP-Java

Beat Wolf
University of applied science Fribourg, University of Würzburg
beat.wolf@hefr.ch

Abstract

The POP-Model is an innovative way to distribute objects over the network, while at the same time simplifying parallel programming. A first implementation of this model, called POP-C++, has been realized as an extension of the C++ programming language. An implementation as an extension of the Java programming language, called POP-Java, started as a bachelor project. This first prototype of POP-Java basically reimplemented the functionality of POP-C++. We adapted this prototype to better integrate with the Java language and fix some remaining problems. The POP-Java implementation was benchmarked against RMI, both implementing the same algorithm. POP-Java yielded comparable results in terms of performance. Now that the foundation of POP-Java has been put in place, expanding it to allow the usage of clouds and run on the windows operating system is a priority.
Keywords: Java; Distributed; Parallel, OO programming;

1 Introduction

Despite progresses made in programming distributed environments this activity remains complex. Indeed often complex code is needed to distribute an algorithm over the network and complex thread synchronization needs to be done to scale over multiple cores and over multiples computers. These are the problems the POP-Model tries to solve thanks to the introduction of the notion of parallel class. Several new keywords have been defined allowing the programmer to easily distribute objects and define how concurrent calls to methods are handled. POP-C++ [1] was the first language to implement these notions and these keywords, allowing for a completely transparent distribution of the objects. The only thing that the programmer needs to do is to annotate the classes he wants to distribute with the appropriate keywords defining how the methods of distributed objects are called. The programmer can declare methods as synchronous or asynchronous, and restrict their concurrent access using 3 keywords: concurrent, sequential and mutex. The details on how those keywords work can be found in [1] and [2].

2 Work Accomplished

A prototype of POP-Java implementing most of the basic features was developed by Valentin Clément [2]. This prototype uses a similar approach than POP-C++ by adding new keywords to the java language. Thus, a new programming language was created, called POP-Java. Because of the added keywords, the POP-Java syntax was no longer compatible with the standard Java language. To correct this, we defined a new version of the POP-Java language using Java annotations instead of keywords, resulting in a syntax that is completely compatible with the standard Java language. As in the original POP-Java prototype, there is still the need for a parser that translates POP-Java source code into java, but only to include some boiler plate code to make programming easier and not to change the syntax.

3 Benchmark

To validate the POP-Java in terms of functionality and performance, the DNA sequence aligner presented last year [3], was ported to use POP-Java. This allowed for a direct comparison between the RMI and POP-Java performance of the same algorithm. The algorithm also consist of many short calls on remote objects, where as the POP-Model was designed to be used for fewer but longer calls. This usecase was an opportunity to optimize POP-Java under those circumstances, increasing performance overall. The test was done on a local grid installation, using 1-6 machines. Those machines where equipped with quad-core CPUs and connected trough a 1GB/s network. The performance test was done using a dataset of 4 billion sequences of an average length of 120 bases, aligned against the human chromosome 7. The first machine only uses 3 cores of the 4 available to align the data, this is to keep one core available for the read and write operations, making sure that they don't bottleneck the algorithm. For this reason, the optimal speedup is adapted. Figure 21 shows the speedup over 6 machines.

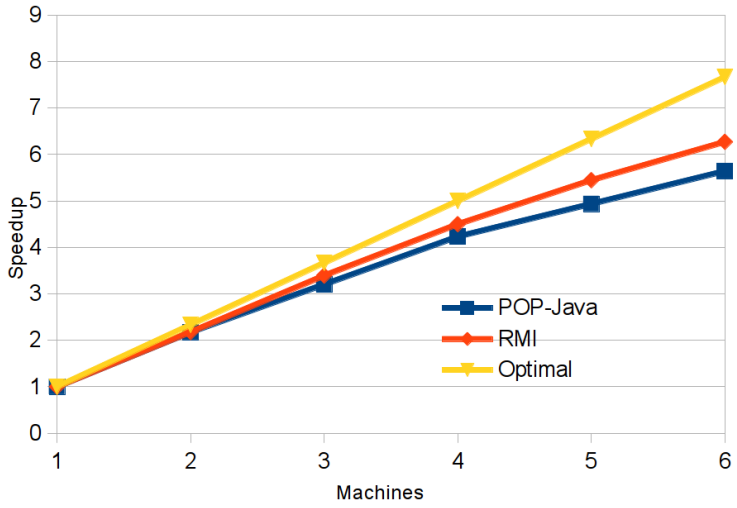


Figure 21: Speedup over 6 machines, with the first machine only using 3 cores

We can see that the speedup progression of both implementations is similar. For the absolute difference in execution speed, RMI is 13% faster on one machine and 30% on 6 machines than the POP-Java version. The exact cause of this difference has yet to be found.

4 Work in Progress and Future Work

While POP-Java works well under linux, much is still left to do. Now that distributing objects over multiple computers in a grid like environment works well, the next step is to bring POP-Java to the cloud. Currently an exploratory project is evaluating this possibility. One of the goals is to make the distribution on a cloud just as easy as the distribution on a grid, possibly by adding a new annotation similar to *@POPConfig(Type.URL)*. Additionally to be able to distribute the application over more network configurations, windows will also be targeted. With java beeing a multiplatform language, windows support is the natural next step after linux and osx. The main obstacle to port POP-Java to windows is the way how objects are instanciated on remote machines. Currently POP-Java opens a secure SSH connection to the remote machine and executes the command needed to start the required object. In a windows environment, setting up SSH is too complicated for most users, specially with the

premise of POP-Java to make distributed programming easier. How this will be solved is still an open question that we hope to address soon. Further improvements are also planned to make the usage of POP-Java easier. The currently used intermediate step between .pjava and pure java files complicates the compilation of application using POP-Java. Making this step either easier or remove it completely is a goal of further development

References

[1] T. A. Nguyen, P. Kuonen, "Programming the Grid with POP-C++," in *Future Generation Computer Systems (FGCS)*, N.H. Elsevier, Volume 23, Issue 1, 1 January 2007, pages 23-30.

[2] Valentin Clément, Pierre Kuonen, "POP-Java, Bachelor thesis," at *University of applied sciences, Fribourg*, 2010

[3] Beat Wolf, Pierre Kuonen, David Atlan, "General purpose distributed DNA aligner,"at the *Doctoral Workshop on Distributed Systems*, 2012 available at `http://beat.wolf.home.hefr.ch/documents/aligner-vuedesalpes.pdf`

TLG Opportunistic Routing Protocol and Its Enhancement in Mobile Ad-hoc Networks

Zhongliang Zhao
University of Bern
zhao@iam.unibe.ch

Abstract

Opportunistic routing (OR) takes advantage of the broadcast nature and spatial diversity of wireless transmission to improve the performance of wireless ad-hoc networks. Instead of using a predefined path to send packets, OR postpones the choice of the next-hop to the receiver side, and lets the multiple receivers of a packet to coordinate and decide which one will be the real forwarder. Existing OR protocols choose the next-hop forwarder based on a pre-ordered candidate list, which is calculated using single network metric. In this paper, we propose TLG - a Topology and Link quality-aware Geographical opportunistic routing protocol. TLG uses multiple network metrics such as network topology, link quality, and geographical location to implement the coordination mechanism of OR. We compare TLG with well-known existing solutions under both static and mobile environments. Simulation results show that TLG outperforms others in terms of packet delivery ratio (PDR) and goodput.

The evaluation of TLG and the comparison with other routing protocols will be presented. Some drawbacks of the existing TLG design will be discussed. Possible enhancement will be proposed for future improvement.

Keywords: MANETs; Opportunistic routing; Geographical routing; Topology awareness; Performance evaluation.

1 Introduction

Wireless ad-hoc networks promise a wide scope of applications in both civilian and military areas, which require scalar and multimedia information in applications such as surveillance, environmental monitoring, emergency recovery, etc. For example, in case of a disaster, such as earthquake or hurricane, the recovery process demands an efficient and rapid deployment of a communication system due to the fact that the standard telecommunication infrastructure might be damaged. In this scenario, wireless ad-hoc networks enable to build a temporary communication network.

Routing in multi-hop wireless network is a challenging issue. The main difficulty lies in that wireless links are unstable and unreliable. Traditional wireless routing protocols treat the wireless link as a wired one, and focus on finding a fixed path between a source-destination pair. However, the selected path may be broken if the environment or topology changes. In this context, Opportunistic Routing (OR) [1] was proposed to cope with the unpredictability of wireless links. In OR, instead of sending unicast packets to a specific node, the sender just broadcasts the packet. The neighbors that successfully receive this broadcast transmission have to coordinate with each other to select one node to forward the packet.

2 TLG Protocol

In this work, we describe the design of our proposed OR protocol - Topology and Link quality-aware Geographical opportunistic routing algorithm, called TLG. This protocol takes into account different network metrics to make a joint routing decision. TLG uses the idea of Dynamic Forwarding Delay (DFD) by considering link quality, progress, and remaining energy to compute the dynamic delay function.

2.1 Dynamic Forwarding Delay (DFD)

When the source node has data to transmit, it includes the geographical information of itself and of the final destination into the packet and broadcasts it. The neighbor nodes that receive the packet, first check whether they are closer to the final destination than the last hop. If not, they drop the packet. Otherwise, they are considered as possible relay nodes, and apply a Dynamic Forwarding Delay (DFD) function. The node with the smallest DFD will be the real forwarder, and it will repeat the broadcast process to find another relaying node. Other nodes will overhear this re-broadcast and cancel their transmission.

We propose a **DFD** [4] function based on multiple metrics, i.e., *progress*, *remaining energy*, and *link quality*, to increase reliability and energy-efficiency. The proposed DFD is calculated according to (7), in which the sum of (α, β, γ) equals to 1. Details of the protocol can be found in [2] and [3].

$$\text{DFD} = (\alpha \times \text{Remaining Energy} + \beta \times \text{Link Quality} + \ \gamma \times \text{Progress}) \times DFD_{Max} \tag{7}$$

2.2 Link Validity Estimation

Our algorithm includes the estimation of the validity time of a link between two connected mobile nodes, and this information will be used in the routing decision. After a node has been selected as the relay node for a sender, the sender will finish the transmission of subsequent packets using unicast to that node. Therefore, the duration of this unicast transmission needs to be determined beforehand. A Link Validity Estimation (LIVE) protocol will run at every node to estimate the validity time (T_{LV}) of each link with its 1-hop neighbors. This value will be used to decide how long the unicast transmission will last. When this link validity timer expires, the sender will start another broadcast process to find a better forwarding node.

2.3 Simulation Results

TLG is evaluated through OMNeT++ simulations. To show that TLG outperforms existing approaches that consider single metrics, we compare TLG with the well-known GPSR and BLR protocols. Figure 22 shows the PDR and goodput of three protocols when the source generates UDP packets with a rate of 2 packets per second. TLG performs much better than GPSR, which can deliver only 20% of the packets. This is because GPSR greedily chooses the neighbor that is closest to the destination as next hop. However, the farthest neighbor has the highest probability to suffer from a bad connection with the packet sender, which leads to packet loss. The not fully covered network might be another reason for GPSR's bad performance. BLR performs better than GPSR, because it does neither have to discover and maintain routes nor to maintain a neighbor table that may be outdated and inconsistent. In

TLG, any nodes are closer to the destination could be the candidates, which increases the coordination overhead and thus reduces performance. However, BLR is still worse than the best case of TLG (#13), since it uses only progress to compute the DFD.

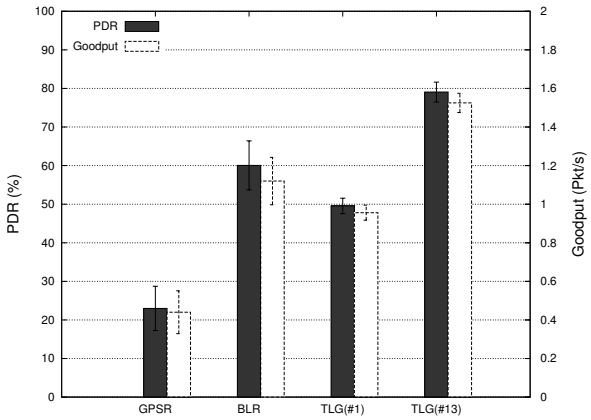


Figure 22: PDR and goodput of three protocols under static network

The same observation could be found when nodes are moving. Figure 23 shows the relationship between PDR/Goodput and moving speed of nodes. As we can see, TLG still performs better than others in mobile environments.

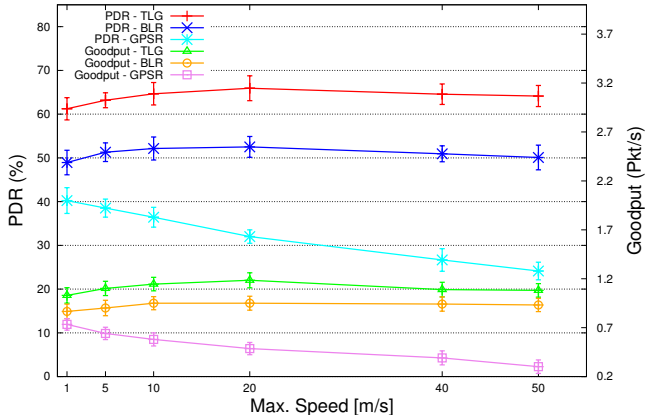


Figure 23: PDR and goodput of three protocols under mobile network

3 Future Work

Several enhancements of TLG could be made. For example, nodes should consider their relative movements when selecting a relaying node, such that a source node does not choose a node that is only valid for short period as its relaying node. Additionally, the fixed weighting values assigned to different metrics could be adjusted, to enable the dynamic adaption of the weights of different parameters according to their real-time values.

References

[1] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 133–144, August 2005.

[2] Z. Zhao, R. Denis, T. Braun, E. Cerqueira, H. Xu and L. Huang, "Topology and Link Quality aware Geographical Opportunistic Routing in Wireless Ad-hoc Networks," *The 9th International Wireless Communications and Mobile Computing Conference (IWCMC 2013)*, Cagliari, Sardinia - Italy, July 1-5, 2013.

[3] Z. Zhao, R. Denis, T. Braun and E. Cerqueira, "On the Performance of Topology and Link-quality Aware Geographical Opportunistic Routing in Mobile Ad-hoc Networks," *to be submitted.*

[4] T. Braun, M. Heissenbuettel, and T. Roth, "Performance of the beacon-less routing protocol in realistic scenarios," *Ad Hoc Networks*, vol. 8, pp. 96–107, 2010.