Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 396136, 10 pages
http://dx.doi.org/10.1155/2015/396136

Research Article

Cyber Physical Sensors and Actuators for Privacy- and Cost-Aware Optimization of User-Generated Content Provisioning

Paolo Bellavista and Carlo Giannelli

Department of Computer Science and Engineering (DISI), University of Bologna, Viale del Risorgimento 2, 40136 Bologna, Italy

Correspondence should be addressed to Paolo Bellavista; paolo.bellavista@unibo.it

Received 25 June 2015; Revised 25 September 2015; Accepted 1 October 2015

Academic Editor: Vicente Traver

Nowadays there is growing interest in evolving the distributed sensors concept from the more traditional one of enabling technology to monitor the surrounding physical environment *towards Cyber Physical Systems (CPS) sensors and actuators*, that is, as a suitable tool to measure/influence the cyber activity of possibly worldwide communities of users (e.g., any geotagged operation leaving a cyber footprint and any cyber physical incentive to stimulate activity as in crowdsensing). To leverage this novel perspective, we propose a framework to integrate at best *multilayer CPS sensors and actuators* as the basis for autonomic management operations on both physical and cyber worlds. In this paper the specific application domain target is peer-to-peer content sharing based on social identities and relationships, but we claim that the proposed CPS framework is of general applicability. In particular, our original middleware solution adopts CPS actuators to move users' content temporarily from smart home environments to high-performance cloud resources to minimize the access time of a dynamically selected quota of contents. Then, based on social network sensors and connectivity/networking ones hosted at lightweight domestic Web servers, our CPS actuators can originally and dynamically move content back from the cloud to smart homes when appropriate, in order to both retain full ownership of user-generated content and reduce cloud hosting costs.

1. Introduction

Distributed sensors are traditionally identified as means for monitoring the physical environment where they are located, for example, to recognize inter-/intradaily vehicle traffic patterns or to promptly identify emergency events. For example, temperature, humidity, and CO_2 can be monitored to evaluate the level of comfort in theatres or large hotel halls or to detect potential fire hazards. Nowadays, however, there is growing interest in evolving this traditional sensor/actuator concept from merely physical to cyber physical. Cyber Physical Systems (CPS) sensors are defined as a mechanism to gather data from both cyber and physical environments, where a huge amount of data is generated by users, for example, in social network activities or when creating and sharing user-generated contents. It is worth noting that, provided that proper solutions for scalability are adopted, CPS sensors can allow monitoring the cyber activity of large communities of users, by collecting also data that users continuously generate in an unaware manner while interacting with CPS environments, that is, any geotagged activity capable of leaving a cyber footprint.

In particular, the paper focuses on two primary *sources of CPS data*:

(i) *Social information* that users willingly generate and publicly provide on social networking applications, eventually associated with differentiated privacy policies, for example, posts over Facebook and shared user-generated contents.

(ii) *Networking and application access information* that users generate in an unaware manner while accessing any resources via HTTP requests (not only Web resources but also, e.g., REST and UPnP invocations).

We claim that this rich and enlarged set of monitoring data could be beneficial to several application domains, by providing more in-depth knowledge of users' context/activities and monitoring it from multiple and novel points of view. For instance, consider the case of a user publishing an album of pictures on a social network, where other users may comment on the whole album and view its pictures separately. On the one hand, retrieving the amount of comments or likes of the album may reveal potential interest in them in the near future, for example, based on the intuitive assumption that albums with many comments are likely to keep their popularity longer than albums with few or no comments. On the other hand, monitoring how frequently the album pictures are downloaded provides a time-evolving trend for actual resource accesses.

To leverage this novel perspective, we originally propose a middleware framework to integrate and exploit at best CPS multilayer sensors and actuators as the basis for autonomic management on both cyber and physical worlds. The paper shows how the synergic exploitation of (i) CPS sensors gathering data at very different abstraction layers and (ii) CPS actuators opportunistically migrating content among distributed nodes may greatly improve the user's quality of experience. In particular, the application domain targeted here is peer-to-peer content sharing based on social identities and relationships. In this domain, the primary goal is to exploit CPS context data efficiently to improve performance and users' privacy through advanced content management operations, in a seamless way for final users and with minimum need of configuration management. As better detailed later, our original middleware transparently exploits CPS actuators to move users' content temporarily from smart homes with limited networking capabilities to high-performance cloud locations in order to minimize the access time of currently (or expected to be) most popular contents. Then, based on CPS sensors integrated on widespread social network applications (e.g., Facebook and Twitter) and on CPS networking sensors hosted by lightweight domestic Web servers, our CPS actuators can originally move back content from the cloud to smart homes at the best time. The goal is twofold: (i) retaining full ownership of user-generated content and (ii) reducing the cost of cloud storage. The strong technical originality of our proposal stems from the exploitation of transparent and application-level middleware for resource sharing in federated network localities, thus allowing immediate deployability in current smart homes with off-the-shelf consumer electronics.

The rest of the paper is organized as follows. Section 2 overviews the primary related work on cyber sensing in CPS. Section 3 presents our abstraction model, by pointing out the existing similarities between physical and CPS environments. Sections 4 and 5 demonstrate how CPS sensors and actuators can be valuably exploited in real-world scenarios to gather social/network-related data and to optimize the management of user-generated contents. Section 6 ends the paper with conclusive remarks and envisioned directions of future work.

2. Related Work

The relevance and technical suitability of jointly exploiting cyber and physical sensors to infer user context have been widely recognized in the recent literature. In short, the primary idea is that the physical world encompasses the cyber one and vice versa to fill the physical/cyber gap [1] in a user-centric way [2]. In addition, the widespread popularity of different social networking applications (facilitating mass-market users interaction and content sharing) has enabled several novel research fields that leverage on gathering and exploiting users' data. For instance, [3, 4] couple a body sensor network with a social network to leverage fast propagation of sensed raw data while simplifying data usage and integration; [5] exploits social networks to spread data gathered by a crowd of mobile devices.

In this context it is of paramount importance to valuably transform raw sensed data in meaningful information with adequately high-level semantics, that is, by transforming social networks from data repositories to knowledge sensors [6]. For example, [7] proposes mathematical modeling and human-based social sensors to gather information about interactions among decision makers on social networks. The work in [8] adopts social sensors to evaluate the similarity of users' opinions and to predict friendship/antagonistic relationships. Instead, [9] exploits Twitter as a cyber sensor to infer when earthquakes occur and where their epicenters are via the monitoring of users' tweets, by showing that it is possible to trigger alerts much quicker than with traditional alerting systems based on seismometers.

About content popularity, instead, the literature recognizes a usual logarithmic distribution of the popularity of online videos (and more generally of multimedia contents available on the Web) [10]. Another work has recently stressed that frequently the popularity of user-generated contents presents a burst pattern, characterized by slow and continuous decrease of popularity together with sudden rises [11]. Based on this observation, [12] shows how it is possible to predict sudden popularity rises of YouTube videos by monitoring Twitter data. In particular, [12] detects topics from streams of twitters and compares them with topics of videos in order to rank their popularity.

As better detailed later, our middleware originally exploits CPS sensors and actuators to gather social/network-related information and to consequently optimize the management of user-generated contents at provisioning time in a completely user-transparent way. Note that while we aim at providing best-effort management based on most typical content access patterns, our solution can be easily adapted/extended to more articulated and sophisticated popularity patterns [11].

3. Modeling CPS Data: User-Aware and User-Unaware Generation

Physical sensors are generally exploited to gather real-time data about physical environments and then compared with (or applied to) a physical model representing the expected behavior of the monitored environment. The associated goals are (i) detecting unexpected events and (ii) predicting the probable evolution of physical phenomena. Then, (iii) raw data and the inferences based on them are used to properly select and operate some "management" actions, possibly in a proactive way. For instance, by monitoring the pressure level along oil pipes it is possible to discover oil leakages, for example, based on unusual pressure slopes, and properly actuate countermeasures, for example, remotely closing safety valves. Or, by gathering wind measurements of distributed weather stations, it is possible to determine the evolution and direction of adverse weather conditions such as thunderstorms and hurricanes and to automatically send alert messages very promptly.

However, when moving from physical to CPS environments it may be challenging to identify (i) which sensors provide most significant data about the real-time environment evolution, (ii) which models to apply on data for the identification of unexpected events and the prediction of future behaviors, and (iii) which actuators, based on CPS data, are the most proper ones to enforce operations on the controlled CPS environment. In fact, edges/borders of physical environments are usually well defined; for example, given an oil pipe or a hurricane, evolution models have been quite solidly developed in the past and are now already available, for example, provided by fluid dynamic physics or complex weather models, and possible actuator operations are quite well established, for example, standard valve settings or alert systems. On the contrary, it is often challenging to define the exact edges/borders of the CPS environments that users interact with, to identify data sources providing relevant (and possibly concise) real-time view of phenomena evolutions, and to take adequate and cost-effective countermeasures via CPS actuation management. Moreover, in many cases there is still the lack of proper models to mimic the evolution of phenomena in the CPS world.

Also to contribute to the addressing of the above limitations, we have the ambition to propose a novel solution that exploits CPS sensors at different abstraction layers. In particular, we envision the coupled exploitation of CPS data that users generate (in both aware and unaware ways). On the one hand, we call "user-aware generated CPS data" the information typically available at the application layer and corresponding to the wide set of user-generated contents posted on social networking applications. For instance, users can deliberately upload pictures, post comments, and like/share user-generated contents. In other words, in this paper, this category identifies the information explicitly offered by users while interacting with the CPS environment. On the other hand, we call "user-unaware generated CPS data" the information usually at the network layer and stemming from the interaction of users with Internet-connected resources, typically by generating "traffic traces" independently of the users' explicit willingness. For instance, users may browse albums of other friends/colleagues via HTTP requests by viewing either only few pictures (thus probably showing limited interest) or the whole album (thus demonstrating higher interest). Or they may watch posted videos either only for few seconds (browsing with limited interest) or till their ends (interested viewer).

Table 1: Synoptic overview of the adopted guidelines for integrated CPS monitoring.

	Modeling	Behavior
Awareness	Coarse-grained pattern discovery	Intentions
Unawareness	Refinement of discovered patterns	Actual activities performed

In other words, we propose to monitor CPS environments by adopting different perspectives and mechanisms and to integrate them to identify/compare both the intentions of users (eventually inferred by their "aware" activities) and their actual behaviors (mainly based on the traces of their actual activities). Table 1 gives a concise synoptic overview of the high-level guidelines behind our integrated management approach towards CPS systems monitoring.

In addition, the possibly huge volume of historical data available on social networks can be exploited to infer users' interest behavior patterns. For instance, the time-varying popularity of user-generated multimedia content can be evaluated by collecting posts and shares of similar content. We claim that, combining together historical data and resource access monitoring, we can generate a significantly improved CPS model, capable of relevantly better performance, as described in the following section. Based on such a model, we will show that CPS actuators can take more efficient resource management decisions, for example, by migrating user-generated content to the cloud and back.

4. User-Generated Content Management Based on CPS Sensors and Actuators

This section presents our middleware that simplifies the development and management of CPS components and, even most relevantly, favors easy deployment over existing environments and easy integration with existing applications. As detailed below, our middleware allows transparent migration of user-generated content based on dynamically retrieved CPS context, while seamlessly interacting with legacy and widespread applications for content sharing. In addition, it relieves developers from the burden of directly interacting with third-party services and can be easily extended, for example, to take advantage of additional social networking applications and cloud storage services.

4.1. RAMP Mechanisms for Social-Driven Home-to-Home Resource Sharing. To clearly describe our novel middleware in a self-contained way, in this section we overview very rapidly our Real Ad hoc Multihop Peer-to-peer (RAMP) framework on top of which the middleware solution originally presented in this paper has been designed, implemented, and experimentally evaluated [13–17].

RAMP supports home-to-home resource sharing in federated spontaneous networks [18], for example, in geographically distributed UPnP localities with possible clashing of IP

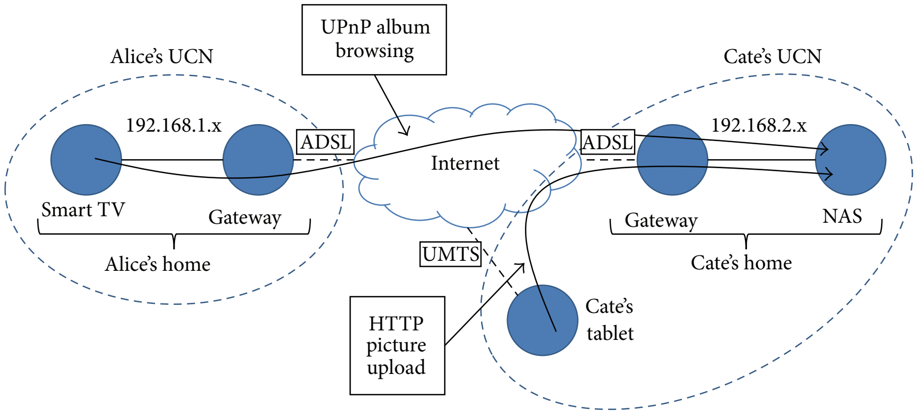


Figure 1: A simple example of federated UCNs.

addresses. RAMP federation is based on portable application-level mechanisms and the dynamic creation of loosely coupled User Centered Networks (UCNs). In particular, we start from UCNs that are personal overlays that tightly interconnect devices owned by a unique social identity. Devices are typically located in different physical networks and are virtually interconnected to easily support the full sharing of data belonging to the same user. Then, we enable UCN federations that represent the dynamic and loose interconnection of UCNs associated with different social identities and linked by social relationships.

Figure 1 shows how UCNs and their federation can be easily exploited to share content without third-party infrastructures. Cate associates her Internet-connected devices with her Facebook profile, thus automatically generating a UCN composed of her tablet, gateway, and Network Attached Storage (NAS). Now Cate's tablet can effortlessly upload pictures directly to her NAS Web server via HTTP, even if the latter resides in her private home LAN. Furthermore, since Alice and Cate are friends on Facebook, their UCNs are federated, thus supporting the browsing of DLNA AV Media Server content stored in Cate's NAS from Alice's smart TV as if they were in the same IP subnet, by exploiting legacy mass-market solutions based on standard UPnP enhanced by a transparent proxy-based gateway [14].

4.2. Coarse-Grained Access Model Based on Popularity and Access Prediction. Our novel middleware is based on three primary steps. First, we aim to identify resource access patterns of user-generated contents based on historical information provided by CPS sensors, such as the ones monitoring social network applications usage. In this way, we create a *coarse-grained model* of the typical access pattern to user-generated content (see Table 1). Then, we gather real-time CPS sensor data by periodically monitoring both social networks and network requests to understand the evolution of per-resource access patterns. We apply both social- and network-related data to the above coarse-grained model to *evaluate the actual popularity of resources* and to predict its future trends. Third, in the case of increasing popularity, our *CPS actuator opportunistically migrates content* from UCNs to cloud storage services to offload domestic networks. In the case of decreasing popularity, the same CPS actuator moves contents back from the cloud to the home locality both to save cloud resources and to (re)gain maximum control on user-generated contents for the sake of privacy.

To demonstrate the feasibility of our approach and to practically show, in a real case, how to model CPS data and actuation behavior in our middleware, based on the simple but realistic assumption that the greater the popularity of user-generated content is, the more frequently it is accessed, we exploit comments/shares/likes on social networking applications to make a first coarse-grained inference about content popularity and access prediction model. In particular, we have asked a significant number (i.e., 50) of statistically relevant and differentiated Facebook users the permission to fully monitor their activities via a specific CPS sensor, based on public Facebook API, that we have implemented and integrated in our RAMP-based content management solution. More specifically, we have applied a CPS sensor to gather the set of albums, photos, and links of each user and, for each retrieved element, its comments and shares with related timestamps. Let us rapidly note that currently we are not considering like actions since they are not associated with a timestamp, thus not allowing exploiting them to detect the time-varying behavior of content popularity. The outcome of this data acquisition process is a database of historical events, for example, posts of new albums or comments, associated with the timestamp when the event occurred.

Then, we have fed Weka [19] with those data and applied the simple Expectation Maximization (EM) clustering algorithm to detect typical usage patterns based on aggregated users' data, by separately considering albums, pictures, and links. Figure 2 shows the time distribution of the detected clusters and their size as a percentage of items in each cluster (new albums, pictures, and links are generated at time 0). The figure clearly demonstrates that most activities on new albums, pictures, and links are performed in the very first hours after their generation. For instance, 77% of comments on pictures are done in the first 25 hours, while 60/84% of picture sharing actions are done in the first 1/20 hours. Similar considerations apply to albums and links. In addition, while

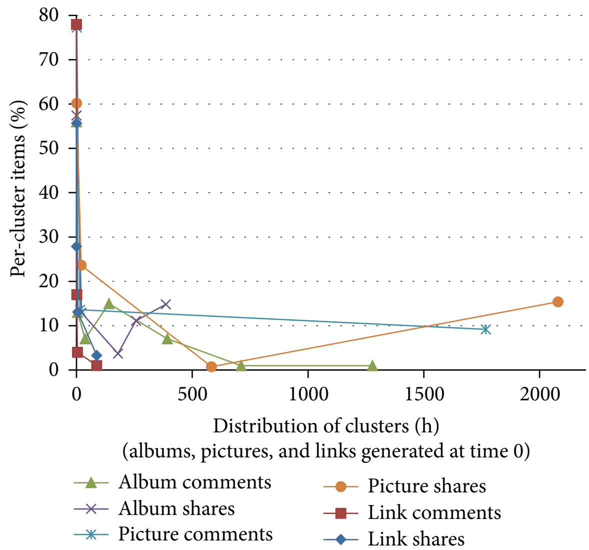


FIGURE 2: Time distribution for the detected clusters (obtained by applying EM to the CPS data retrieved from our 50 Facebook users' profiles).

any activity on links ends after a short time period (about 97% of activities in the first 6 days), albums and pictures exhibit a long-tail behavior (more than 10% of activities after a year). Moreover, we have observed that the network traffic due to access to user-generated content is primarily generated in the very first hours/days for many shared resources that fall into our primary interest, at least for the application domain considered in this paper.

The above considerations have been used to guide the design and efficient implementation of our middleware prototype described in the following subsections.

4.3. Dynamic Resource Migration. Here, we specifically describe our solution to opportunistically migrate content from UCNs to the cloud and vice versa based on CPS sensors and actuators. Before delving into finer details, let us give a rapid high-level overview of the main drivers pushing for storing resources on either UCNs or the cloud. In particular, we claim that user-generated content should

(i) *reside on user-owned UCNs* as long as possible to

  (a) *maintain full control and ownership*; in fact, when content is moved to third-party server-side components, at least in current commercial agreements, users tend to loose part of control on their data;

  (b) *reduce costs of server-side storage,* since cloud storage services, such as Dropbox, typically provide only a limited amount of storage freely, while they apply a fee in case of huge storage space;

(ii) *reside on cloud storage services* after their generation but only for the time interval required

  (a) *to reduce network traffic* (only when dynamically needed, e.g., because of high popularity peaks of selected user-generated contents) over possibly overloaded spontaneous domestic networks, which typically have limited bandwidth if compared with industry-level cloud storage;

  (b) *to reduce processing/energy consumption* on UCN nodes, by possibly compromising even their practical usability by final users due to CPU overloading.

Based on the above considerations and experimentally detected access patterns, our middleware adopts the straightforward but effective policy of transparently migrating user-generated contents from UCNs to the cloud as the default initial option, that is, as soon as the new contents are generated and shared, and a cloud storage resource is available. Thus, users who are willing to access resources typically get them from the cloud rather than directly from UCNs (which was the usual way in RAMP), in a completely transparent way. Then, CPS actuators can remove content from the cloud when its popularity decreases; again, the trigger for resource migration is determined by taking into consideration both the cluster-based model of Section 4.2 and online CPS sensors (monitoring of social network and network request activities).

4.4. Migration Mechanisms and Policies. To fully understand our migration solution, let us start with the description of how our CPS modules manage user-generated content (see Figure 3):

(1) User-generated content is uploaded on the NAS within the UCN.

(2) The CPS actuator residing on the gateway node running on the UCN identifies new content on the NAS and

  (2.1) uploads the new content on private cloud storage, for example, Dropbox, by getting a public URL allowing accessing it; if the cloud storage is full, it autonomously removes older contents based on a liveliness indicator (see Section 5.1);

  (2.2) automatically posts a link on a social network, for example, Facebook or Twitter, including the URL for the access to the content on the employed NAS (not directly to the cloud storage); this URL is RAMP-enabled and works independently of the possible IP address clashing in federated spontaneous networks [17].

(3) Interested users can access the URL posted on the social network.

(4) The gateway node intercepts associated requests and transparently redirects them to the URL on the cloud storage.

(5) Later, based on the observed numbers of actual requests and shares/comments on social networks,

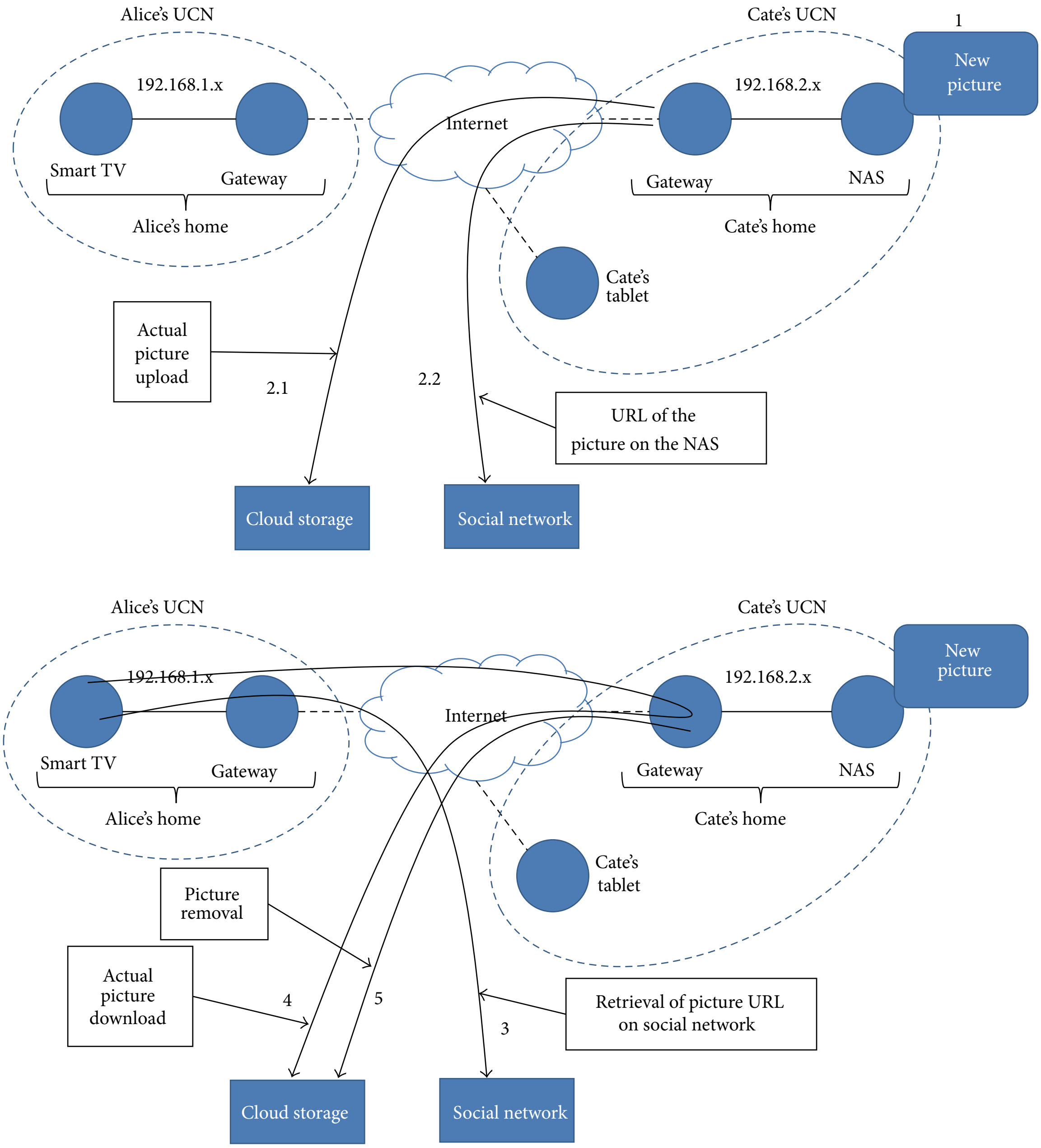


FIGURE 3: A simple case of user-generated content management where our middleware transparently migrates resources from UCN to cloud storage.

the CPS actuator may remove the content from the cloud storage. The successive requests are served by sending content from the UCN rather than by the cloud storage service. In any case, the URL posted on the social network keeps its validity, independently on the current and actual location of user-generated contents, since it seamlessly and transparently points now to the resource on the UCN, no more on the cloud storage service.

Note that while gateways receive every request for user-generated content, whenever the content is on the cloud they can efficiently reply with very small redirect messages instead of sending the whole multimedia content at any request. This relevantly contributes to the scalability of the overall middleware.

5. Middleware Design/Implementation and Experimental Performance Evaluation

As already mentioned, with no loss of generality and to report in-depth technical details and performance results about our original approach, this section presents how we have designed and implemented our middleware prototype for the specific domain of peer-to-peer social-aware content

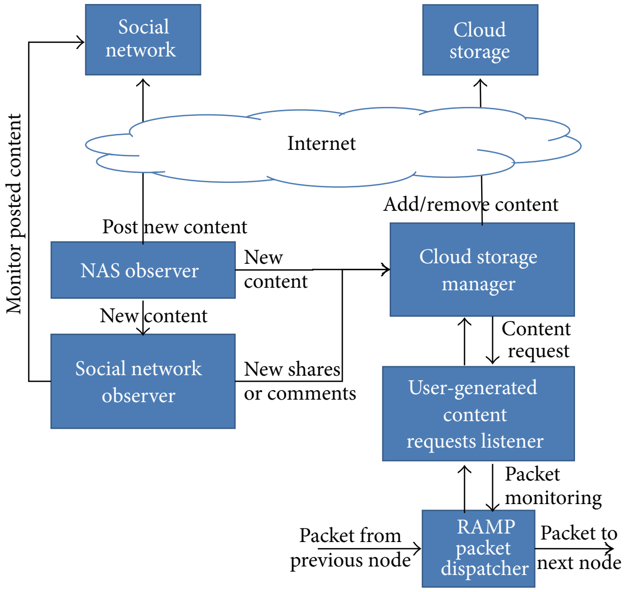


Figure 4: The architecture of our CPS middleware.

sharing. In particular, we describe how we have adopted CPS sensors and actuators to improve both performance results and users' privacy (content ownership) through advanced content management operations.

5.1. Architecture and Implementation Insights. Figure 4 shows the overall architecture of our solution, which runs at RAMP-enabled gateways (except for social network and cloud storage) in a completely transparent way for final users:

(i) *RAMP Packet Dispatcher* is in charge of application-level dispatching, also by supporting the registration of (possibly application-specific) listeners to allow external components to monitor traversing packets.

(ii) *User-Generated Content Requests Listener* is the CPS sensor that is registered as listener at the packet dispatcher and that monitors traversing packets. Whenever it identifies requests for user-generated contents under observation, for example, a picture stored in a NAS, it interacts with the Cloud Storage Manager both to notify the request and to ask whether the requested resource resides on the cloud (and where). In the positive case, it modifies the traversing packet to transparently give to the client the information about the location of the resource on the cloud; otherwise, it forwards the packet without any modification.

(iii) *NAS Observer* is the CPS sensor that monitors NAS content and, whenever new content of interest is detected, both posts its URL on the social network and notifies Cloud Storage Manager and Social Network Observer.

(iv) *Social Network Observer* is the CPS sensor periodically interacting with the social networking applications (in the current prototype with Facebook, but the component is easily extensible to include additional integration mechanisms; e.g., an extension for Twitter is now under final refinement) to gather comments/shares for posted content. The collected data continuously feed Cloud Storage Manager (default period of 30 minutes, which can be set at runtime as a JMX-manageable configuration parameter).

(v) *Cloud Storage Manager* is the CPS actuator responsible for adding new content to the cloud storage and for removing it when there is no more space available or the content has become obsolete (additional details below). We currently exploit the Dropbox API [20] to migrate and store user-generated contents over the popular Dropbox repository. Anyway, the proposed middleware is modular and can be easily extended to integrate with other cloud storage services provided that they offer adequate programmatic API.

It is worth noting that Cloud Storage Manager also keeps metadata associated with each content request in order to appropriately manage content migration towards the cloud and back. For each user-generated content item, in fact, it maintains

(i) name of the user-generated content;

(ii) local URL on the NAS;

(iii) remote URL on the cloud storage;

(iv) shares and comments (with timestamp) on the integrated social networks;

(v) number of requests (with their associated timestamps) on the gateway.

In addition, for each item, Cloud Storage Manager computes the corresponding *liveliness* indicator, namely, the number of minutes in the future a content should reside on the cloud to optimize user-generated content management according to the targeted objectives described previously. For new items, liveliness is initially set with the goal of serving at least 75% of the requests by redirecting them to the cloud storage service. Based on the model described earlier, we have set the initial liveliness to 24 hours. Then, for each share/comment on social network and for each request to gateway, Cloud Storage Manager increases the liveliness value of a given delta (i.e., 1 minute as the default setting). In this way, more popular content will reside on the cloud longer, thus avoiding overloading the UCN network with huge traffic for few popular items and consequently improving its scalability. While we intend to further investigate which is the optimal value of delta to use when shares/comments/requests trigger the increase of liveliness, our preliminary experiments show that 1 minute is adequate for many content sharing applications, with a good tradeoff between responsiveness and peak burst avoidance.

Then, Cloud Storage Manager periodically decreases the liveliness value of stored items; moreover, it removes content from the cloud whenever the associated liveliness value reaches 0. Cloud Storage Manager also removes content when a new content should be uploaded but there is no free room

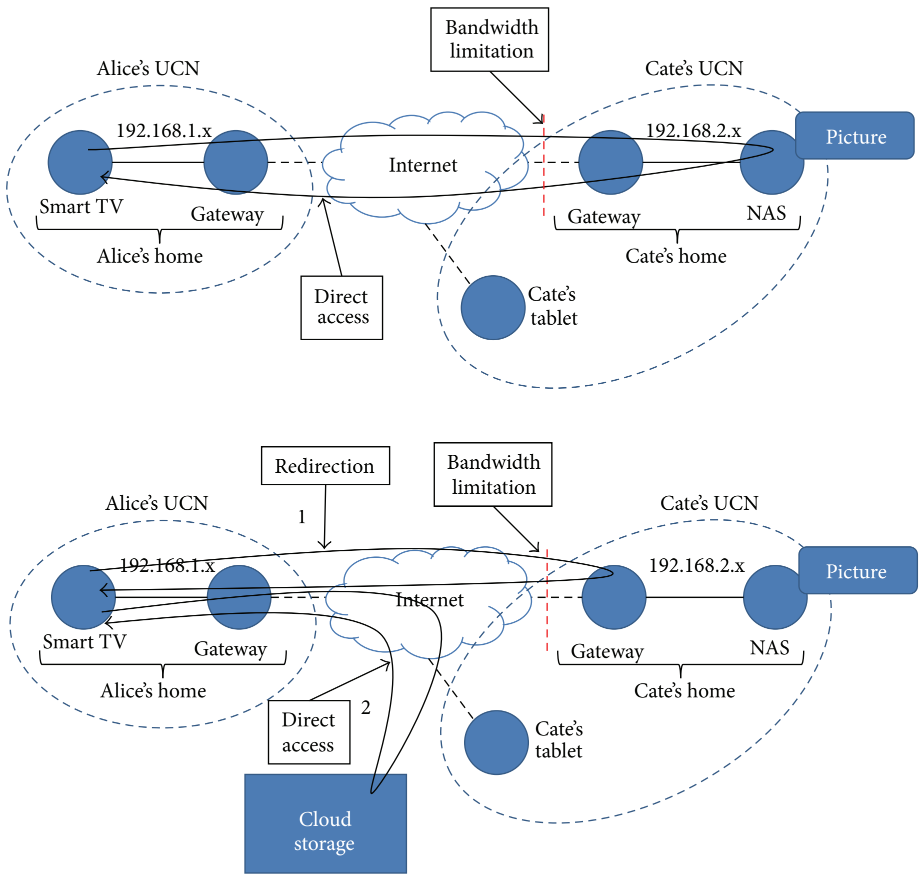


Figure 5: Simple testbed scenarios for performance evaluation without (up) and with (down) dynamic content migration.

available for it; in this case, it adopts the simple policy of removing content with the least liveliness value till there is not enough room for the new content (other policies are under evaluation and out of the scope of this paper, e.g., considering together liveliness and content size within a single metric).

5.2. In-the-Field Feasibility Evaluation and Performance Results. To demonstrate the feasibility of our approach, we have deployed our middleware prototype in a realistic in-the-field execution environment, also to experimentally evaluate whether the adoption of CPS sensors and actuators can actually improve the quality of experience perceived by final users. In particular, we focused our efforts on quantitatively verifying if and how much our CPS sensors and actuators can minimize the download latency perceived by final users. Here, we report on the performance results achieved with our CPS sensors and actuators residing on top of Win 7 Pro desktop PCs with Intel Pentium i5 2410M, 2.30 GHz, 3 MB cache L3, and 8 GB RAM. We have considered four different scenarios, characterized by

(1) content directly provided by UCN, with wide bandwidth (Figure 5, up);

(2) content directly provided by UCN, with limited bandwidth (Figure 5, up);

(3) UCN redirecting clients to the cloud storage, with wide bandwidth UCN (Figure 5, down);

(4) UCN redirecting clients to the cloud storage, with limited bandwidth UCN (Figure 5, down).

For each scenario, we have evaluated the end-to-end time interval starting with the client content request and ending with the completed reception of the required resource (possibly including the time for request redirection in the case of cloud storage).

Figure 6 presents the in-the-field performance results when varying

(a) the number of concurrent clients from 1 to 200;

(b) the size of the requested content from 1 KB to 1024 KB and 7168 KB.

Figure 6 left-up (wide bandwidth, content provided by UCN) clearly shows that the considered UCN can correctly serve many concurrent clients when content size is small, that is, 1 KB (serious scalability limitations for medium-large size contents). Instead, this time considerably increases if content size is larger (1024 KB and 7168 KB). In particular, requests are not correctly handled and lead to application failures with 200 concurrent clients and a payload equal to 7168 KB. In case of limited bandwidth and content directly provided by UCN (Figure 6, right-up), the performance results get worse, in

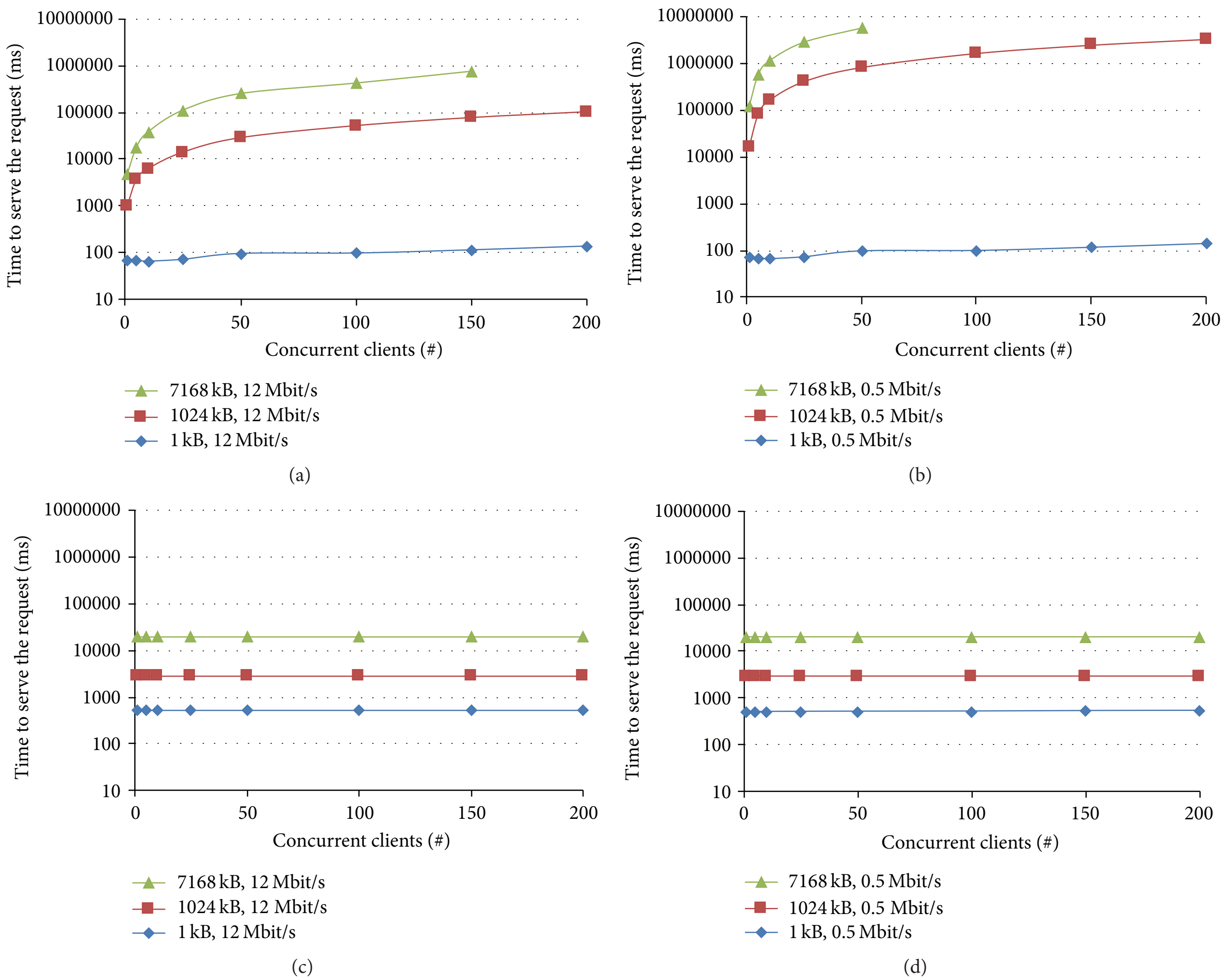


Figure 6: Performance with wide bandwidth and without/with content on cloud storage (left, up/down) and limited bandwidth and without/with content on cloud storage (right, up/down).

particular with increased end-to-end service time for 1024 KB payload, and request failures with 100 concurrent clients and 7168 KB payload.

Figure 6 down (content provided by cloud storage) demonstrates that, thanks to the content management operations transparently performed by our CPS actuators (seamless content migration to the cloud and automated request redirection), UCN can seamlessly serve many concurrent requests: requests are correctly and timely served despite the amount of concurrent clients and the requested content size, with service time that has shown to depend mainly on the size of the requested shared resource. However, in the case of very limited content size (1 KB), the measured performance results are slightly worse than the case without content on the cloud storage (compare Figure 6 up and down, 1 KB), and this gives additional information about the overhead imposed by our application-level middleware. In fact, in addition to the time for content downloading (about 100 ms), we experience the overhead of redirecting requests and interacting with the cloud storage service (about 400 ms), which could be considered negligible in the case of more usual payload size of 1024 KB and 7168 KB.

Overall, the reported performance results demonstrate that our dynamic and transparent UCN-to-cloud content migration based on CPS sensors and actuators relevantly improves final users' experience; this is assessed notwithstanding the portable and application-layer approach, which has in its turn relevant advantages for legacy integration and rapid deployability. It is also worth noting that, by supporting cloud resource access, UCN gateways incur in much lower computing and traffic loads, thus also improving the quality of UCN users' access to "regular" Web, thanks to the limited impact of interferences due to shared content requests.

6. Conclusive Remarks and Future Work

The paper presents the design, implementation, and evaluation of our novel middleware prototype to exploit CPS sensors and actuators at different abstraction layers in order to enable the effective migration of user-generated contents from smart environments to the cloud and back. Key original aspects of our proposal are (i) exploiting CPS sensors that gather high-level social networking and low-level packet networking monitoring data and (ii) exploiting CPS actuators that properly and effectively interact with the CPS environment to maximize the final user's quality of experience in a personalized and dynamic way. The reported performance results demonstrate the feasibility of our approach, by showing that, probabilistically speaking, (i) the access to user-generated content generally occurs briefly after content generation and (ii) temporarily migrating user-generated contents on cloud storage can relevantly improve UCN scalability and user-perceived performance indicators.

The encouraging results achieved so far are stimulating our further research activities along two main directions. On the one hand, we are working on more articulated and sophisticated content access patterns to correspondingly evolve our model. On the other hand, we are developing and testing less straightforward migration policies to maximize content ownership and privacy while imposing acceptable overhead on federated spontaneous networks, for example, federated domestic networks [17], always by considering personalized preference profiles and behavior patterns.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the EU Artemis Arrowhead project (http://www.arrowhead.eu/). The authors would like to take this opportunity to thank also the colleagues and M.S. students, in particular G. Capuzzo, who contributed to some results presented in this paper.

References

[1] A. Sheth, "Computing for human experience: semantics-empowered sensors, services, and social computing on the ubiquitous web," *IEEE Internet Computing*, vol. 14, no. 1, pp. 88–91, 2010.

[2] M. Broy and A. Schmidt, "Challenges in engineering cyber-physical systems," *Computer*, vol. 47, no. 2, Article ID 6756843, pp. 70–72, 2014.

[3] M. Domingo, "A context-aware service architecture for the integration of body sensor networks and social networks through the IP multimedia subsystem," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 102–108, 2011.

[4] M. A. Rahman, A. El Saddik, and W. Gueaieb, "Building dynamic social network from sensory data feed," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 5, pp. 1327–1341, 2010.

[5] X. Hu, T. H. S. Chu, H. C. B. Chan, and V. C. M. Leung, "Vita: a crowdsensing-oriented mobile cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 148–165, 2013.

[6] I. Kompatsiaris, D. Gatica-Perez, X. Xie, and J. Luo, "Special section on social media as sensors," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1229–1230, 2013.

[7] V. Krishnamurthy and H. V. Poor, "A tutorial on interactive sensing in social networks," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 1, pp. 3–21, 2014.

[8] B. Guler, B. Varan, K. Tutuncuoglu et al., "Using social sensors for influence propagation in networks with positive and negative relationships," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 2, pp. 360–373, 2015.

[9] T. Sakaki, M. Okazaki, and Y. Matsuo, "Tweet analysis for real-time event detection and earthquake reporting system development," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 919–931, 2013.

[10] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, 2010.

[11] J. Ratkiewicz, F. Menczer, S. Fortunato, A. Flammini, and A. Vespignani, "Traffic in social media II: modeling bursty popularity," in *Proceedings of the 2nd IEEE International Conference on Social Computing*, pp. 393–400, IEEE, Minneapolis, Minn, USA, August 2010.

[12] S. D. Roy, T. Mei, W. Zeng, and S. Li, "Towards cross-domain learning for social video popularity prediction," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1255–1267, 2013.

[13] P. Bellavista, C. Giannelli, L. Iannario, L.-W. Giox, and C. Venezia, "Peer-to-peer content sharing based on social identities and relationships," *IEEE Internet Computing*, vol. 18, no. 3, pp. 55–63, 2014.

[14] P. Bellavista, P. Gallo, C. Giannelli, G. Toniolo, and A. Zoccola, "Discovering and accessing peer-to-peer services in UPnP-based federated Domotic Islands," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 3, pp. 810–818, 2012.

[15] P. Bellavista, A. Corradi, and C. Giannelli, "Middleware for differentiated quality in spontaneous networks," *IEEE Pervasive Computing*, vol. 11, no. 3, pp. 64–75, 2012.

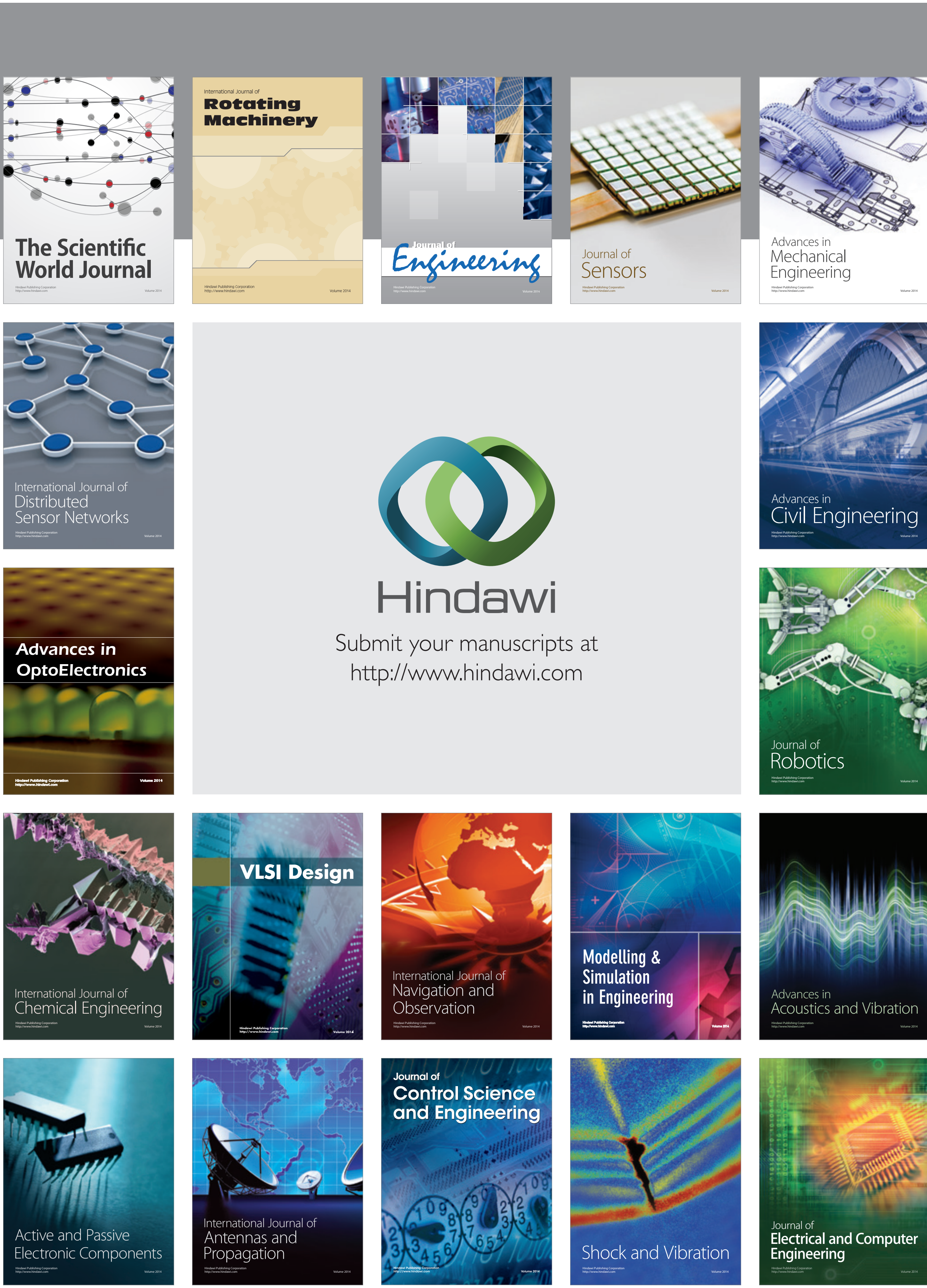
[16] P. Bellavista, A. Corradi, and C. Giannelli, "The real ad-hoc multi-hop Peer-to-peer (RAMP) middleware: an easy-to-use support for spontaneous networking," in *Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC '10)*, pp. 463–470, Riccione, Italy, June 2010.

[17] P. Bellavista, A. Corradi, and C. Giannelli, "Middleware-layer quality-aware collaborative re-casting of live multimedia in multi-hop spontaneous networks," *Journal of Network and Systems Management*, vol. 23, no. 3, pp. 620–649, 2015.

[18] L. S. Ferreira, M. D. De Amorim, L. Iannone, L. Berlemann, and L. M. Correia, "Opportunistic management of spontaneous and heterogeneous wireless mesh networks," *IEEE Wireless Communications*, vol. 17, no. 2, pp. 41–46, 2010.

[19] "Weka 3: Data Mining Software in Java," 2015, http://www.cs.waikato.ac.nz/ml/weka/.

[20] 2015, http://dropbox.github.io/dropbox-sdk-java/api-docs/v1.8.x/.


The Scientific World Journal
International Journal of Rotating Machinery
Journal of Engineering
Journal of Sensors
Advances in Mechanical Engineering
International Journal of Distributed Sensor Networks
Hindawi
Submit your manuscripts at
http://www.hindawi.com
Advances in Civil Engineering
Advances in OptoElectronics
Journal of Robotics
International Journal of Chemical Engineering
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
Advances in Acoustics and Vibration
Active and Passive Electronic Components
International Journal of Antennas and Propagation
Journal of Control Science and Engineering
Shock and Vibration
Journal of Electrical and Computer Engineering