Agile Communication Middleware for Next-generation Mobile Heterogeneous Networks

Giacomo Benincasa[1], Alessandro Morelli[2], Cesare Stefanelli[3], Niranjan Suri[4,5], Mauro Tortonesi[6,*]

1) Florida Institute for Human & Machine Cognition
15 SE Osceola Ave
Ocala, FL 34471, USA
E-mail: gbenincasa@ihmc.us
Phone: +1 352 387 3060

2) Department of Engineering
University of Ferrara
Via Saragat, 1
44122 Ferrara, Italy
E-mail: alessandro.morelli@unife.it
Phone: +39 0532 97 4933

3) Department of Engineering
University of Ferrara
Via Saragat, 1
44122 Ferrara, Italy
E-mail: cesare.stefanelli@unife.it
Phone: +39 0532 97 4992

4) Florida Institute for Human & Machine Cognition
40 South Alcaniz St.
Pensacola, FL 32502, USA
E-mail: nsuri@ihmc.us
Phone: +1 850 202 4444

5) US Army Research Lab
Adelphi, MD, USA

6) Department of Engineering
University of Ferrara
Via Saragat, 1
44122 Ferrara, Italy
E-mail: mauro.tortonesi@unife.it
Phone: +39 0532 97 4888
Fax: +39 0532 97 4888

*) Corresponding author

Agile Communication Middleware for Next-generation Mobile Heterogeneous Networks

Abstract

Mobile devices, including smartphones and tablets, have fundamentally changed Internet access and communication patterns, enabling today's ubiquitous consumption of Web and multimedia content as well as social- and location-aware services. However, the requirements of next-generation mobile and pervasive computing applications also expose the limits of traditional TCP/IP-based communication paradigms and call for new concepts and tools that can enable more adaptive and robust communications. Ultimately, innovative communication middleware will need to provide a new set of abstractions and programming paradigms to realize seamless mobility and disruption-tolerant communications. Here, we describe the requirements and programming models for next-generation middleware, stemming from our experience in designing and implementing the Agile Computing Middleware (ACM), starting from a realistic adoption scenario for next-generation applications.

Keywords

Programming paradigms (D.2.17.i); Application-aware adaptation (D.4.1.0.a); Middleware/business logic (J.8.I); Pervasive computing (J.9.d); Mobile Applications (J.9)

1. A Scenario

Innovative mobile and pervasive computing applications, characterized by an extensive use of location- and social-based features and an ever-increasing interaction with the surrounding environment, are rapidly evolving, contributing to the realization of smart cities and enabling the dynamic formation of virtual social communities. To understand which challenges software engineers will have to deal with from the communications perspective, let's follow Sybil, a fictitious character living in a metropolitan area and seamlessly using "smart" next-generation mobile and pervasive applications of the near future.

Sybil is walking around the city's fashion district while chatting on a smartphone with her friend Rosie, using a SmartChat application communicating over a 4G link. After a long walk, she finally arrives at her favorite shop. The network-sensing component installed in Sybil's smartphone detects the presence of the store's Wi-Fi network and immediately initiates the procedure to connect the device to it. Detecting a much higher bandwidth now available at no charge to the user, SmartChat automatically starts a negotiation process with the instance running on Rosie's device to "promote" the chat conversation to a videoconference. Rosie, who's enjoying a cappuccino and the free Wi-Fi at her favorite cafe, agrees to switch communication mode, thus allowing SmartChat to set up the videoconference between the two smartphones.

Meanwhile, SmartCRM, the shop’s customer relationship management service, transmits a beacon message that activates a SmartShop application on Sybil’s smartphone. SmartShop logs Sybil into SmartCRM, where she is recognized, her purchase history analyzed, and three dresses recommended to her from a new collection that could be of her liking, along with a special discount to reward her loyalty as a customer.

Sybil likes two of the recommended dresses, but, even with the discount, she can afford only one; so she wants to ask Rosie’s opinion. Being a gold customer, Sybil benefits from the shop’s customer-dedicated services, which means she can use SmartShop to find the set of flat-screen terminals in front of the dressing rooms and reserve one of them. Sybil then reaches her reserved terminal and moves her endpoint of SmartChat’s video conversation with Rosie to the larger screen. Through the webcam installed on top of the flat-screen terminal, Rosie can watch in real time a streaming video of Sybil trying on the dresses she likes and give her friend some purchasing advice.

Having chosen her new dress, Sybil now uses SmartChat to move the videoconference endpoint back to her smartphone, and thanks Rosie for her valuable advice. While waiting for the sales assistant to fold her new dress and place it in a bag, Sybil can use SmartShop to confirm the dress purchase and proceed to the payment task through a digital transaction. Sybil then leaves the shop and walks to the nearest bus stop to travel back home. Once Sybil moves outside the premises, SmartChat detects the loss of free, high-bandwidth connectivity and automatically reduces its footprint by switching the videoconference with Rosie back to a chat conversation over 4G.

On the bus, Sybil decides to watch the latest episode of Person of Interest, so she opens SmartStream, an application that plays live streaming videos over 4G. Sadly, it is rush hour, and cellular communications are rather disturbed because of all the load and interferences caused by nearby users, which translates into occasional dropped video frames. Fortunately, other people next to Sybil are watching the same show, so SmartStream instances running on their devices transparently establish ad hoc connections, via Wi-Fi or Bluetooth, in an attempt to improve video quality by retrieving lost frames from nearby peers without placing any additional load on the cellular network.

While watching the show, Sybil also enjoys using SmartStream’s social networking functions, to open in SmartChat a new group chat with other fans of the series, such as her boyfriend, Sam, who’s watching the show at his house. This allows Sybil to comment on the most thrilling scenes of the show and interact in real time with other fans while on her way home.

This scenario perfectly illustrates the challenges that software engineers will face in developing next-generation mobile applications, which require specific support functions, such as for seamless session mobility and automatic discovery of nearby resources, at the communication middleware level. In fact, applications will need to deal with highly dynamic environments such as heterogeneous networks, characterized by the overlapping of several different wireless networks (4G and Wi-Fi, 4G macrocells and pico/femtocells, and so on). These scenarios will also present applications and systems with the opportunity to select the best network attachment point—the one providing the highest bandwidth or minimizing energy consumption for radio communications—or to leverage multiple communication links at the same time. [1]

Like SmartChat, which can automatically turn chat conversations into videoconferences when free Wi-Fi is available, next-generation applications will need to dynamically scale their quality of service as the quality of connectivity changes or the user switches from free to costly connections, to cope with abrupt changes in network conditions or to comply with user-defined policies. The highly varying environments also raise the opportunity to implement location- and context-aware communications that consider quality of communication as a primary context aspect. [2] Combined with the growing number of connected and remotely operable devices for the Internet of Things and with the adoption of innovative interfaces such as Google Glass, new communication paradigms might contribute to the creation of immersive virtual environments where humans will interact heavily with devices and service providers in their physical proximity and play the dual role of information consumers and producers.

Social awareness will also be a characteristic trait of next-generation applications. In fact, a recent and interesting trend is the emergence of mobile social networking middleware: platforms that monitor and manage contacts between people and support the creation of virtual social communities according to physical proximity criteria, common interests, or the relationships between the participants. Support for disruption-tolerant group communications and knowledge about the communication environment are essential to enable the development of place-centric mobile social middleware. [3]

In addition, the Sybil scenario illustrates the security and privacy requirements of next-generation mobile applications. Although a few of them are similar to those of current generation mobile applications in terms of authentication, authorization, and nonrepudiation, there are some notable differences. [4] In particular, confidentiality, anonymity, and location privacy will become increasingly important—for example, in our scenario, SmartStream needs to share video frames with nearby peers without leaking information about Sybil's identity, location, or preferences (including the show she's watching). Similarly, SmartChat needs to preserve the confidentiality of Sybil and Rosie's conversation even when running from the store's terminal, a device that belongs to neither of them. Application mobility raises additional security issues, especially when something like SmartChat isn't preinstalled in a shop's terminals, which means the store needs to allow for transparent activation. [5]

Sybil's story implicitly raises a last but still quite important issue. On Sybil's smartphone, SmartStream and SmartChat compete to access the cellular link as she watches and comments on her favorite show, but the real-time constraints of live streaming mean that the former application should have priority on channel access. Next-generation environments will host multiple applications competing for scarce resources, with various requirements and priorities, raising the very real need to mediate between applications' resource requests, considering their priorities and preventing excessive resource consumption.

2. Limits of the TCP/IP Protocol Stack

The scenarios we just described can't be easily supported by traditional TCP/IP-based communication paradigms. Although some readers might already be familiar with the problems associated with TCP, we summarize them here for completeness. TCP assumes continuous end-to-end connectivity between two peers (breaking in case of temporary disconnections), suffers from severe performance issues in mobile environments, and conceals network conditions to applications, thus hindering the design of next-generation applications. On the other hand, best-effort, connectionless UDP-based communications semantics are often insufficient for disruption-tolerant group communications. This forces applications to develop ad hoc mechanisms to send messages to a specific set of recipients in the network or to ensure that (at least the critical) messages are correctly delivered.

In spite of these problems, developers and software engineers still leverage TCP connections as the basic building block for their network applications, as they provide a simple and familiar interface. However, by concealing network conditions, the TCP programming model hinders the design of next-generation mobile applications such as SmartChat and SmartStream. Many solutions claim to address TCP mobility issues at the network layer [6] or to optimize TCP performance on wireless networks. [7] The Stream Control Transmission Protocol (SCTP) [8] addresses some of TCP's issues by supporting multihoming and implementing a multiple-stream model designed to mitigate the latency issues of the TCP FIFO queuing model, a well-known problem even in wired environments. [9] However, SCTP still doesn't deliver sufficient levels of disruption tolerance to be adopted as the basic building block for next-generation mobile applications.

Opportunistic networking, a recently emerged paradigm, advocates a completely different conceptual approach to avoid the set of problems that traditional communication solutions pose on the design of next-generation mobile applications. [10] Opportunistic networking aims to support communications in highly dynamic scenarios with temporary contact opportunities between nodes by leveraging store-and-forward techniques and by taking advantage of node mobility to support the information dissemination process. [11]

The aforementioned proposals are interesting and significant advances, but none of them can be considered as a comprehensive solution on which to build next-generation applications. The need for better communication programming models is clearly perceived in state-of-the-art middleware for context-aware applications and ambient intelligence, both of which typically leverage traditional TCP and/or UDP communications and use Mobile IP to deal with mobility issues. Research projects such as HYDRA (www.hydramiddleware.eu) and SM4LL (http://sm4all-project.eu) could benefit from next-generation programming models that enable the tuning of the communication function to current network conditions. We believe this motivates the design of specific communication solutions at the middleware level to bridge the gap between the requirements of next-generation mobile applications and the communication functions provided by the TCP/IP stack.

3. Requirements for Next-generation Communications Middleware

The networking scenario we presented earlier illustrates a major challenge for software engineers. Without middleware providing specific functions, such as seamless mobility, QoS adaptation, and exploitation of resources in physical proximity, engineers would have to spend a lot of effort to develop the communication functionalities required by applications such as SmartChat, SmartShop, and SmartStream. The characteristics of next-generation mobile applications demand a radical rethinking of the programming models for communication.

To facilitate the development of applications such as SmartChat, which automatically adapts its QoS to current network conditions, we need network-aware programming models. Next-generation applications should be specifically designed to consider the dynamicity and heterogeneity characteristics of mobile environments, with frequent and severe fluctuations in link conditions and node mobility not being the exception but the rule. Middleware should provide accurate channel state information to applications, collected from the continuous monitoring of resource availability and utilization, through the analysis of data on communication links (signal strength, lifetime, throughput, and so forth), and neighboring nodes (number, contact lifetime, and so on). [12]

With a network-aware programming model, this middleware could support applications in implementing a continuous adaptation process of their communication function to reach a trade-off between application requirements and current network conditions. For instance, the middleware could notify SmartChat about severe degradations in network conditions to trigger a prompt reaction, such as demoting the videoconference to a much less resource-hungry chat conversation.

Exposing knowledge on the currently available network channels is also essential to enable applications to exploit multiple network access points, such as SmartChat and SmartStream do with Wi-Fi and 4G as Sybil travels back home on the bus. Next-generation applications will need to smartly take advantage of multiple communication links in accordance with their communication semantics, user preferences, and current network state, enabling mobile data offloading when possible. [1]

To deal with abrupt changes in channel conditions, next-generation applications such as SmartChat will need to implement smart service-tailoring strategies that also consider finer-grained QoS adjustments. To this end, the middleware should provide multiple delivery semantics to match the requirements of the different message types that applications need to exchange. Ensuring sequential and reliable message delivery can be very expensive in terms of both bandwidth and latency, and not every application needs (or might be willing) to pay that cost. In addition, this middleware should give applications advanced control of message queues, providing high-level mechanisms to enable the deletion of a message pending in the transmission queue and the modification of its content, priority, or delivery semantic.

In our example, when Sybil is watching the live video stream on 4G, SmartStream needs to access a much larger share of the channel than SmartChat. To enable smart resource management, the middleware should implement a priority-based channel assignment to competing applications running on the node. Preemption and reallocation of channel resources are both major requirements of next-generation mobile applications and a fundamental departure from the TCP/IP model, which is based on flow control procedures that tend to assign an equal share of bandwidth to each connection. Decision making in application prioritization should consider both user preferences and application priorities and characteristics.

The SmartChat capability of moving Sybil's videoconference from her smartphone to the terminal in front of the fitting room while ensuring continuous servicing also highlights another requirement for next-generation communication middleware: support for seamless mobility. More specifically, the middleware should enable connections' mobility, that is, the dynamic and transparent rebinding of connections to another device or network interface without disrupting the related service session.

Sybil's example enables us to point out some interesting security and privacy-related requirements for next-generation applications. Protecting confidentiality, anonymity, and location privacy in communications will require explicit security mechanisms, especially for occasional communications with unknown entities. SmartStream's discovery process also requires access to Sybil's privacy settings, as she needs to be able to specify the content she's willing to share and the users she wishes to cooperate with during streaming. Also, users' different sensibilities about privacy might raise the need to provide management tools that ensure the desired levels of confidentiality, thus requiring middleware to integrate with policy management systems that enable user-defined policies. Next-generation environments will also raise issues that demand a better education of users about security and privacy issues.

Finally, the SmartStream recovery of lost video frames by sharing data over Wi-Fi calls for disruption-tolerant peer-to-peer communication functions that enable local resource sharing, improve resource availability, and decrease energy consumption. More specifically, the middleware should support local resource discovery, the dynamic creation of groups of interest, and robust information dissemination. These features would allow Sybil’s SmartStream instance to exchange video frames with other peers that remain only temporarily within communications range, such as those people getting off the bus or waiting at the bus stop.

4. ACM: An Example of Next-generation Communication Middleware

Our experience in communication middleware stems from the design and realization of the Agile Computing Middleware (ACM; www.ihmc.us/Research/distributed.php), which represents our response to the challenges of next-generation network environments. [13] ACM has the goal of providing a valuable and comprehensive collection of components to support the development of efficient distributed applications. Each ACM component, as depicted in Figure 1, is designed to address a set of commonly related problems that often recur in mobile wireless scenarios.

The Mockets component, for example, offers a rich set of functionalities that, as many experiments in the field have confirmed, are very effective in addressing several challenges that applications face in next-generation mobile environments. [12] Mockets provides a comprehensive API that implements message- and connection-oriented communications on top of UDP.

Within each Mockets connection, applications can open many independent data flows and configure each one with different QoS and priority levels that can be dynamically changed at any time. Each flow is then mapped into an independent queue in the middleware; this implies that QoS and prioritization settings impact all open connections on the device, but Mockets has the ultimate control over them. The middleware can therefore leverage user settings and information about current channel conditions to further adjust the QoS and priority levels set by the applications. Finally, Mockets guarantees complete orthogonality in data delivery semantics: each flow can be configured for any combination of sequenced/unsequenced and reliable/unreliable message delivery.

The Mockets support for session mobility would allow SmartChat developers to easily implement the functions that allow Sybil to move her communication endpoint from her smartphone to the flat-screen terminal in the shop, and then back again, without ever interrupting the videoconference session with Rosie. In fact, not only does Mockets enable seamless handovers in networks that don’t support the IEEE 802.21 Media Independent Handover standard, but it also allows applications to suspend a service session, save the local endpoint’s state, migrate it to another network interface or even to another node, and finally resume the suspended connection from there.

Another interesting feature of SmartChat is its ability to continuously analyze current network conditions to detect new and improved links. This lets SmartChat automatically switch to Wi-Fi when Sybil enters the shop, sense the increased bandwidth available, and promote the chat session to a videoconference. The Mockets component fosters a feedback-loop-based programming model, in which applications hand the middleware the data specifying the type of delivery they need (reliable, sequenced, and so on), and, in turn, Mockets constantly provides them with information on the current channel conditions. This enables software developers to design adaptive applications that can choose the delivery semantics that best fit the type of traffic produced, while tailoring the transferred amount of data to the network's current state (for instance, by increasing/decreasing the generation rate, compression level, or information resolution).

Let's go back to Sybil on the bus, watching the live stream of Person of Interest via SmartStream. Other people nearby are watching the same show, which enables SmartStream instances to initiate, using ad hoc Wi-Fi, a collaborative sharing of those frames that each device had successfully downloaded over the 4G connection, thereby improving the quality of everyone's streaming experience. To implement such a complex function, SmartStream's developers could leverage and combine multiple features of ACM: the dynamic creation of group of interests (the group of people watching the same show), the discovery and sharing of resources between nearby nodes (the video frames), and the definition of the desired privacy policy (to preserve confidentiality).

ACM has a dedicated component, the Group Manager, that supports the efficient discovery of peers (and the resources and services they share) through an adaptable combination of proactive signaling and reactive search. [13] The discovery process is performed in the context of a group that peers need to join before they can participating in sharing their resources or querying for other peers that can answer their needs. We designed the Group Manager to operate effectively in a mobile environment, optimizing the discovery process to minimize bandwidth consumption, reduce latency, and favor the discovery of resourceful nodes. The Group Manager's discovery and management algorithms operate in a fully decentralized manner to withstand the high churn rate that characterizes next-generation mobile networks.

To support the collaborative sharing of video frames using ad hoc wireless connections, similarly to what SmartStream does when Sybil is watching Person of Interest on the bus, we designed DisService, a component that implements robust and adaptive information dissemination between peers. [13] Nodes running DisService can leverage a publish-subscribe model designed to operate in highly dynamic environments that allows applications to exchange messages via independent virtual communication channels, called "subscriptions". By joining the same subscription—in this case, the video stream of the Person of Interest episode—many instances of SmartStream running on different devices could take advantage of the communication model provided by DisService to set up a network of cooperating peers that are willing to share the video frames they receive.

The communication concepts implemented in DisService are based on the opportunistic networking paradigm and the principle of opportunistic listening. In accordance with them, DisService pushes to the next level the use of broadcast and multicast messages in next-generation environments. Through techniques like store-and-forward communications, the use of aggressive message-caching policies, and the possibility of prioritizing traffic on a per-message basis, multiple nodes can receive and store the data, thereby significantly increasing their availability and enabling disruption-tolerant information dissemination.

During this process, nodes keep track of each other's contact history and resource availability, thus allowing each node to build a knowledge about its surrounding environment (namely, "world state"). According to their specific (user-defined) policies, DisService instances can share information about node types (mobile, access point, sensor, vehicle, and so on) and characteristics (remaining battery life, computational power, storage, and so on). Social-, location-, and context-aware applications relying on DisService can access the knowledge contained in the world state to implement smart resource management policies.

Going back to our scenario, when Sybil enters the shop and SmartShop logs her into SmartCRM, the latter recognizes her and grants her access to a set of benefits reserved for gold customers. This enables her to discover a group of available terminals, reserve one of them, and transfer her SmartChat session onto it. DisService supports applications like SmartShop by implementing the concept of private virtual channels, which permits the sharing of confidential data among a restricted number of peers that have access rights to that subscription.

Finally, we designed ACM to enable the policy-driven tuning of the communication function behavior. More specifically, ACM allows users and applications to define policies that tailor the generated traffic according to their needs and current network conditions, thereby optimizing resource usage. To this end, ACM integrates with the KaoS policy manager. [14]

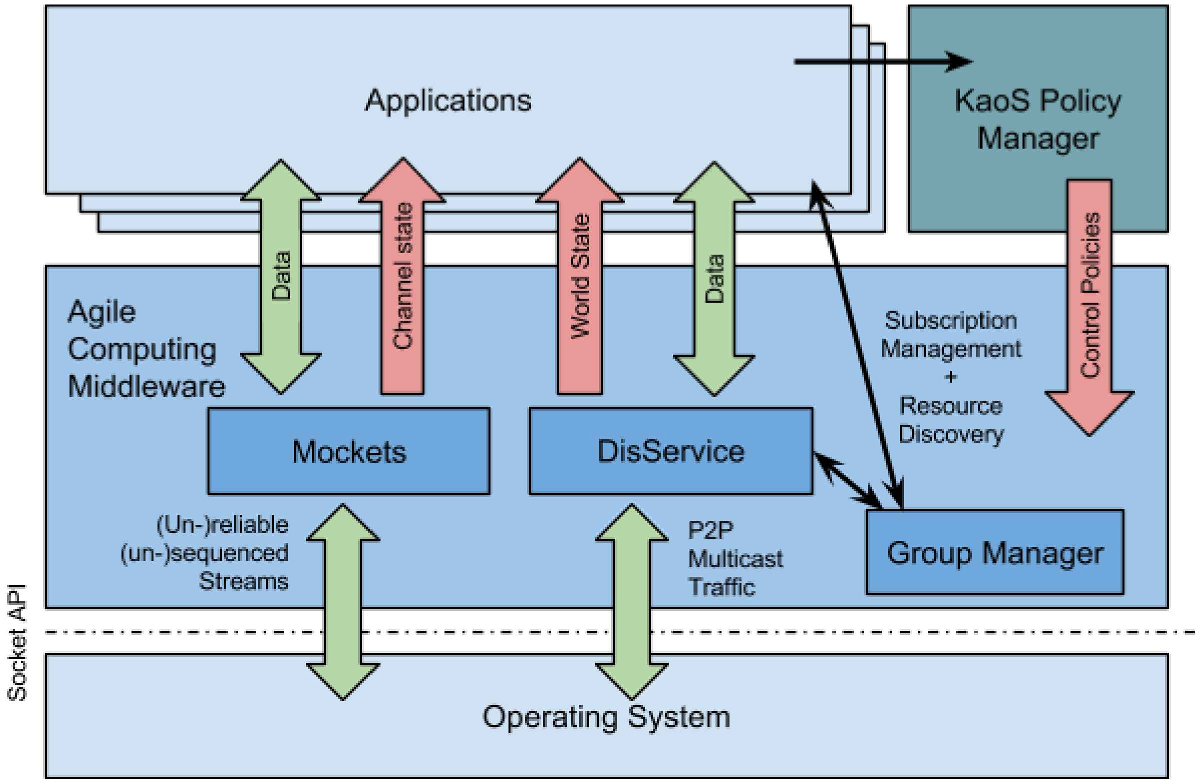

Figure 1. Architecture of the Agile Computing Middleware, highlighting the Mockets, Group Manager, and DisService components and their interactions with next-generation applications.

5. Conclusions

The challenges issued by next-generation mobile communication scenarios clearly emphasize the inadequacy of traditional communication solutions in supporting efficient mobile and pervasive applications. Research to date has mostly focused on higher-layer middleware or on the optimization of TCP-based solutions, but communication middleware could provide the right set of concepts and tools to support next-generation mobile applications.

Despite the features and the level of optimization that such a middleware can provide, software engineers will require a deeper knowledge of the complexities and problems that characterize communications in mobile heterogeneous scenarios. The next-generation software engineer will face the challenging task of understanding the real communication requirements of the mobile applications he or she is developing, and of consistently choosing and combining the best suited middleware tools to make a wise and efficient use of the available network resources.

References

[1] K. Lee et al., "Mobile Data Offloading: How Much Can WiFi Deliver?", *IEEE/ACM Transactions on Networking*, Vol. 21, No. 2, 2013, pp. 536-550.

[2] P. Bellavista et al., "A Unifying Perspective on Context-Aware Evaluation and Management of Heterogeneous Wireless Connectivity," *IEEE Communications Surveys & Tutorials*, Vol. 13, No. 3, 2011, pp. 337-357.

[3] P. Bellavista et al., “Mobile social networking middleware: A survey”, *Pervasive and Mobile Computing*, Vol. 9, No. 1, 2013, pp. 437-453.

[4] D. Ma, G. Tsudik, "Security and privacy in emerging wireless networks", *IEEE Wireless Communications, V*ol. 17, No. 5, 2010, pp. 12-21.

[5] P. Yu et al., “Application mobility in pervasive computing: A survey”, *Pervasive and Mobile Computing*, Vol. 9, No. 1, 2013, pp. 2-17.

[6] A. Gladisch et al., “Survey on mobility and multihoming in future internet”, *Wireless Personal Communications*, 2012.

[7] K.-C. Leung, V. Li, “Transmission control protocol (TCP) in wireless networks: issues, approaches, and challenges”, *IEEE Communications Surveys & Tutorials*, Vol. 8, No. 4, 2006, pp.64-79.

[8] Ł. Budzisz et al., “A Taxonomy and Survey of SCTP Research”, *ACM Computing Surveys*, Vol. 44, No. 4, 2012, Article 18.

[9] A. Erbad, C. Krasic, “Sender-side Buffers and the Case for Multimedia Adaptation”, *ACM Queue*, Vol. 10, No. 10, 2012.

[10] L. Pelusi et al., “Opportunistic networking: data forwarding in disconnected mobile ad hoc networks”, *IEEE Communications Magazine*, Vol. 44, No. 11, 2006, pp. 134-141.

[11] A Martìn-Campillo et al., “Evaluating opportunistic networks in disaster scenarios”, *Journal of Network and Computer Applications*, Vol. 36, No. 2, 2013, pp. 870-880.

[12] N. Suri et al., “Communications Middleware for Tactical Environments: Observations, Experiences, and Lessons Learned”, *IEEE Communications Magazine*, Vol. 47, No. 10, 2009, pp. 56-63.

[13] N. Suri et al., “Peer-to-Peer Communications for Tactical Environments: Observations, Requirements, and Experiences”, *IEEE Communications Magazine*, Vol. 48, No. 10, 2010, pp. 60-69.

[14] A. Uszok et al., “KAoS policy management for semantic web services”, *IEEE Intelligent Systems*, Vol. 19, No. 4, 2004, pp. 32-41.

Biographies

GIACOMO BENINCASA (gbenincasa@ihmc.us) is a research associate at the Florida Institute of Human & Machine Cognition, working on information dissemination in tactical networks and on biologically-inspired security infrastructures. He holds a B.Sc. and an M.Sc.

in computer science engineering from the University of Modena and Reggio Emilia, Italy.

ALESSANDRO MORELLI (alessandro.morelli@unife.it) received his B.Sc. degree in Information Engineering and his M.Sc. degree in computer science engineering from the University of Ferrara, Italy. He is currently a Ph.D. student in the Engineering Department of the University of Ferrara. His research interests focus on distributed and mobile computing, opportunistic networking, and smart cities.

CESARE STEFANELLI [M] (cesare.stefanelli@unife.it) received his Laurea degree in electronic engineering and his Ph.D. in computer science engineering from the University of Bologna, Italy. He is currently a full professor of computer science engineering in the Engineering Department of the University of Ferrara. His research interests include distributed and mobile computing, adaptive and distributed multimedia systems, network and systems management, and network security.

NIRANJAN SURI [M] (nsuri@ihmc.us) is a research scientist at the Florida Institute for Human & Machine Cognition (IHMC) and also a visiting scientist at the U.S. Army Research Laboratory, Adelphi, Maryland. He received his Ph.D. in computer science from Lancaster University, England, and his M.Sc. and B.Sc. in computer science from the University of West Florida, Pensacola. His current research activity is focused on the notion of agile computing, which supports the opportunistic discovery and exploitation of resources in highly dynamic networked environments. His other research interests include coordination algorithms, distributed systems, networking, communication protocols, virtual machines, and software agents.

MAURO TORTONESI [M] (mauro.tortonesi@unife.it) received his Laurea degree in electronic engineering and his Ph.D. in computer science engineering from the University of Ferrara, Italy. Currently, he is an assistant professor in the Engineering Department of the University of Ferrara. His research interests include distributed and mobile computing, opportunistic networking, IT service management, business-driven IT management, and large-scale e-maintenance.