EMERGENCY MANAGEMENT

A Mobile Cloud Computing System for Emergency Management

Karan Mitra, Saguna, and Christer Åhlund, Luleå University of Technology, Sweden

The mobility management system for mobile cloud computing, M²C², aims to select the best cloud and network for processing sensor data while responders are in an emergency area.

Natural and manmade emergencies can cause tremendous economic, environmental, and, most importantly, human loss. Recent emergencies, such as the Indian Ocean tsunami, Hurricane Katrina, and the 9/11 terrorist attacks in New York City, caused significant loss of human lives. It is estimated that more than 280,000 lives were lost due to the 2004 Indian Ocean Tsunami.[1] A recent report argued that the use of an information and communications technology (ICT) infrastructure could be beneficial for evaluating and responding to emergency situations.[2] However, ICT's full potential in emergency management is yet to be realized. For example, we need robust ICT systems that provide situational awareness of the emergency area and offer improved decision support to manage emergencies efficiently. Emergency management also requires

 2325-6095/14/$31.00

safe evacuation of people from the danger zone; thus, an emergency management system must also collect, process, analyze, and disseminate relevant, accurate, and timely information.[3–5]

Information about evacuees, responders, and the emergency area can be collected from various sources, including onsite or on-body sensors, wearable devices (such as Google Glass) worn by the responders, mobile nodes (such as smartphones), and cameras. This information can either be processed on responders' mobile nodes or offloaded to the remote facility for better decision making. However, three key issues need consideration. First, mobile nodes have limited processing, storage, and battery resources and might be unsuitable for processing the large amounts of data originating from sensors and cameras. Second, timeliness in emergency management is crucial, and data upload and download to and from remote sites might not be feasible. Therefore, the data originating from the emergency area should be processed near the source to minimize end-to-end latency. Finally, we expect the devices, sensors, and cameras to be with the responders, who might move between several access networks. These devices might experience handoffs, intermittent network connectivity, and limited bandwidth and coverage area.

Emergency management can benefit significantly from mobile cloud computing (MCC), which enables offloading of computation and storage from mobile devices to the nearest cloud, preferably at the first hop.[7] MCC can enable fast and secure access to relevant information required by the entities (such as responders) involved in an evacuation. Future wireless base stations or access points will likely have cloud functionality, bringing the cloud to the edge of the network, and significantly reducing end-to-end latency and increasing device battery lifetime.[7,8] But to successfully integrate MCC into emergency management, we must address mobile and cloud computing challenges such as mobility management, handoffs, intermittent network connectivity, network latency, limited network bandwidth, network congestion, limited coverage area, and battery lifetime.[6,9] MCC also needs to handle challenges associated with cloud resource management for efficient application provisioning while responders are on the move. For example, cloud quality-of-service (QoS) parameters (CPU, RAM, and disk I/O) can vary stochastically depending on user demands and workloads.[10]

Our *mobility management system for mobile cloud computing*, M^2C^2, uses multihoming to ensure seamless network handoffs (with low latency and packet loss) when the mobile node roams in an emergency area. It also incorporates cloud and network probing; and metrics for cloud and network selection based on application requirements and network and cloud load. To the best of our knowledge, M^2C^2 is the first MCC system to support cloud and network-aware mobility management while responders are roaming in an emergency area.

Emergency management can benefit significantly from mobile cloud computing (MCC).

Mobile Cloud Computing for Emergency Evacuation

Consider the scenario in Figure 1, where a group of responders are deployed in an emergency area to evacuate people. The evacuation area is divided into several zones, with each zone having an emergency response vehicle (ERV). Each ERV provides local cloud functionality accessible via Wi-Fi, 3G, and satellite networks. The responders are equipped with mobile nodes (for example, smartphones), wearable devices (for example, Google Glass), and sensors (for example, accelerometers; GPS; temperature, and humidity sensors; and so on), which connect to local clouds for low-latency data processing, storage and access. The local cloud in each zone might also connect to the command station via 3G or a satellite link for holistic situational awareness of the emergency area. As a failover mechanism, the local cloud can also connect to public clouds for redundant data processing and storage.

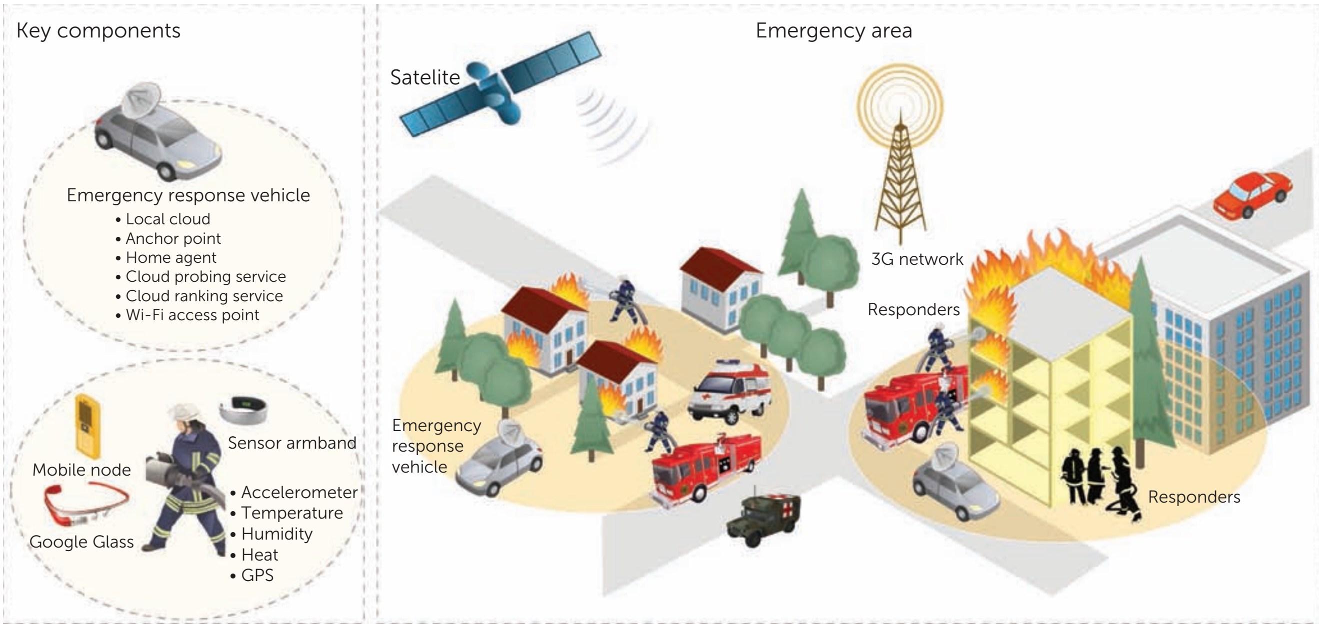

FIGURE 1. A scenario for emergency evacuation during urban fires using mobile cloud computing.

The data originating from all the responders in a particular zone—that is, from their sensors or applications running on the mobile node—is transmitted via Wi-Fi or 3G networks to local clouds for processing, storage, and analysis. If the Wi-Fi network doesn't provide sufficient QoS because of signal fading or network congestion, the responder's mobile node can handoff to a 3G or satellite network (or vice versa) for data transmission. However, a handoff can cause temporary network disconnection, leading to intermittent cloud connectivity for data processing and access. In addition, the nearest cloud might be susceptible to QoS degradation owing to large amounts of data being transferred and processed through it. Clouds must therefore be able to offload processing or storage to another local or public cloud. In regard to this, M^2C^2 provides fast and reliable data processing and access via the best clouds and the best available access networks.

A Mobility Management System for Emergency Evacuation

Figure 2 shows M^2C^2's high-level architecture, which includes Multihomed Mobile IP (M-MIP),[11] a mobility management protocol, to support efficient handoffs between several access networks. Using M-MIP, a mobile node connects to several access networks simultaneously and probes them before initiating the handoff process (between these access networks). In particular, mobile nodes perform network discovery, network configuration, and network registration for all available networks in advance, considerably reducing the number of steps during the handoff process, and resulting in low-latency handoffs with minimal packet losses.

M^2C^2 also incorporates several network and cloud entities to enable cloud and network probing and selection. These entities include local and public clouds, such as Amazon Elastic Compute Cloud (EC2), Microsoft Azure, and Google Cloud Platform, home agent, cloud probing service (CPS), cloud ranking service (CRS), mobile node, Wi-Fi and 3G networks, and an anchor point. In M^2C^2, an anchor point can perform several roles. It can act as the home agent and assist a mobile node by providing it with network probing and handoff management functionalities. The anchor point can also run the CPS to probe local and public clouds, as well as the CRS to select the best cloud for applications to offload computation and storage. The mobile node periodically tracks both clouds and networks via these M^2C^2 entities (anchor point, CPS, and CRS) so applications can determine the best cloud and network while responders roam in heterogeneous access networks (HANs), covering an emergency area.

Network Probing and Selection Mechanism

To select the best available network interface i, where $i \in I$, the mobile node performs passive network path probing to compute the relative network load (RNL_i) metric for each i.[11] In particular, a mo-

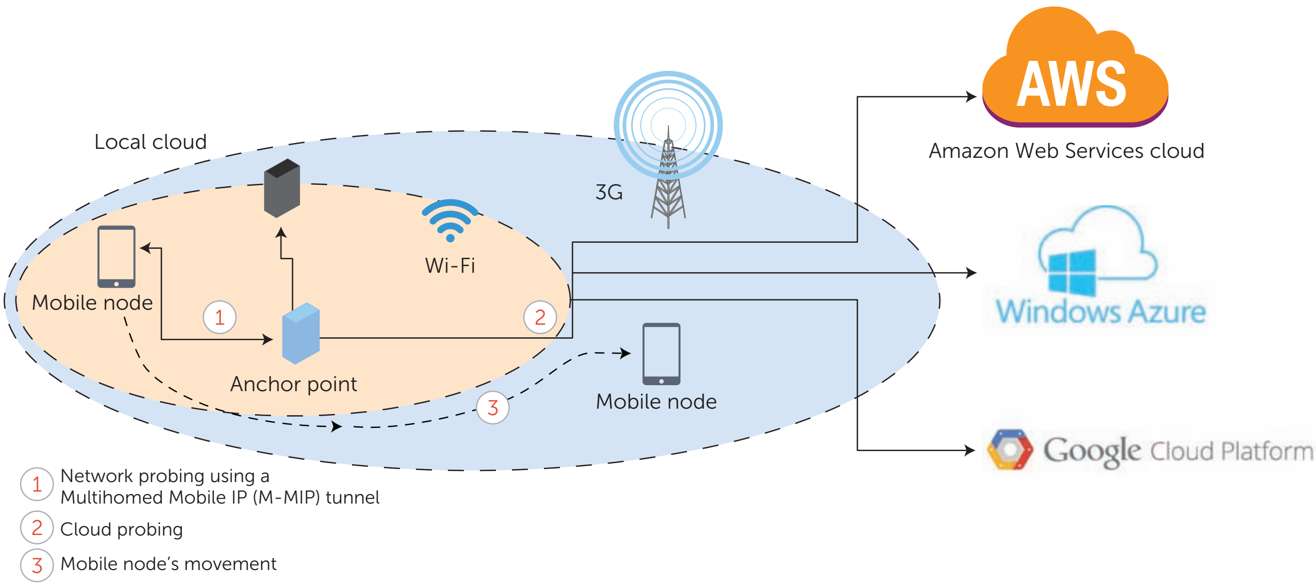

FIGURE 2. The mobility management system for mobile cloud computing (M^2C^2) supports cloud and network probing and cloud and network selection when a responder roams in an emergency area and is connected via heterogeneous access networks such as Wi-Fi and 3G. While roaming, the responder's application can also access both local and public clouds. M^2C^2 selects the best cloud and the best network for low-latency data computation and access.

bile node probes all the available networks I simultaneously by periodically sending binding update messages to the home agent and by receiving the corresponding binding acknowledgment messages from the home agent (see Figure 2). The amount of time between the mobile node sending a binding update message to the home agent and receiving the corresponding binding acknowledgment message from the home agent is the *roundtrip time* (RTT). M^2C^2 uses the RTT values to compute the RNL metric for determining the load on access networks.

We compute the RNL metric as follows:

$$\mathrm{RNL} = Z_n + cJ_n \tag{1}$$

$$Z_n = \frac{1}{h}\mathrm{RTT}_n + \frac{h-1}{h}Z_{n-1} \tag{2}$$

$$\mathrm{RTT}_n = R_n - S_n \tag{3}$$

$$D_n = \mathrm{RTT}_n - \mathrm{RTT}_{n-1} \tag{4}$$

$$J_n = \frac{1}{h}|D_n| + \frac{h-1}{h}J_{n-1}, \tag{5}$$

where S_n is the sending time for the binding update packet from the mobile node to the home agent where $n \in N$; R_n is the time the binding acknowledgment packet is received at the mobile node from the home agent; h is the history window for calculating the weighted average, where $h = 5$ is considered to be an optimal value[11]; and c represents the weight of the RTT jitter value compared to the RTT value. For instance, the value $c = 5$ means that the RTT jitter value contributes five times more than the RTT value. Finally, we initialize the variables Z, D, and J as $Z_0 = \mathrm{RTT}_0$, $D_0 = 0$, and $J_0 = D_1$. The network i with a lowest RNL value [$\min(\mathrm{RNL}_i)$] is the target network for handoff.

Cloud Probing and Selection Mechanism

M^2C^2 mechanisms for cloud probing and cloud ranking support QoS-aware cloud selection. The anchor point or any other dedicated network element on a particular access network can run the CPS (see Figure 2), which tracks the QoS statistics (CPU, memory, and I/O) of both public and local clouds by probing them regularly using RESTful APIs. The anchor point or any other dedicated entity in the network can also run the CRS, which retrieves the QoS stats from CPS (using a RESTFul API) and computes the rank for each cloud $k \in K$ (where K represents all available clouds) based on criteria such as application type, CPU utilization, memory utilization, and disk I/O. Using a RESTful API, the mobile node can then retrieve the best cloud k based on the computed rank ($\mathfrak{R}_k$) for a particular application. The mobile node will then use the kth cloud for task processing and storage.

The URL of the RESTful API (for retrieving the best cloud from CRS) is http://<CRS IP addr.></cloudrankservice>. The tag <CRS IP addr.> represents the IP address of the CRS, and the tag </cloudrankservice> is the Web resource where the CRS is running. The mobile node makes a GET call to this URL and retrieves the best cloud as an HTTP response. Also, a GET call to http://<CRS IP addr.></cloudrankservice><appl. type> retrieves the best cloud k based on the supplied application type where the tag <appl. type> represents the different types of applications—for example, a critical medical response application.

CRS selects the best cloud k by computing the rank of each cloud $\mathfrak{R}_k \; \forall \; K$ using the simple additive weighting (SAW) scheme (Equation 6), a multicriteria decision-making method (MCDM).[12] We rank the clouds using the following formula:

$$\mathfrak{R}_k = w_l(\mathrm{QoS}_j) + (1 - w_l)(\mathrm{Cost}_j), \tag{6}$$

where $j \in J$ represents the jth QoS and cost parameter for cloud $k \in K$; $\mathfrak{R}_k$ represents the rank of the kth cloud; w_l represents the weights associated with each parameter QoS_j and Cost_j for each application $a \in A$; and $\sum_{l=0}^{N} w_l = 1$. The parameter Cost_j can be a monetary or probing cost related to a cloud service. The parameter QoS_j represents QoS parameters, such as CPU utilization, network throughput, and end-to-end latency. Some QoS parameters, such as throughput need to be maximized, whereas others, such as cost, need to be minimized. Therefore, we normalize the parameters using the following generic equations:

$$\mathrm{QoS}_j = \begin{cases} 1, & \text{if } \mathrm{QoS}_j \geq \max\left(\mathrm{QoS}_j\right) \\ \dfrac{\mathrm{QoS}_j - \mathrm{QoS}_{j_{\min}}}{\mathrm{QoS}_{j_{\max}} - \mathrm{QoS}_{j_{\min}}}, & \begin{array}{l} \text{if } \min\left(\mathrm{QoS}_j\right) < \mathrm{QoS}_j \\ \quad < \max\left(\mathrm{QoS}_j\right) \end{array} \\ 0, & \text{if } \mathrm{QoS}_j \leq \min\left(\mathrm{QoS}_j\right) \end{cases} \tag{7}$$

$$\mathrm{Cost}_j = \begin{cases} 1, & \text{if } 0 < \mathrm{Cost}_j \leq \min\left(\mathrm{Cost}_j\right) \\ \dfrac{\mathrm{Cost}_{j_{\max}} - \mathrm{Cost}_j}{\mathrm{Cost}_{j_{\max}} - \mathrm{Cost}_{j_{\min}}}, & \begin{array}{l} \text{if } \min\left(\mathrm{Cost}_j\right) < \mathrm{Cost}_j \\ \quad < \max\left(\mathrm{Cost}_j\right) \end{array} \\ 0, & \text{if } \mathrm{Cost}_j \geq \max\left(\mathrm{Cost}_j\right) \end{cases} \tag{8}$$

We select the cloud with the highest $\mathfrak{R}_k$ (computed using Equation 6) as the best cloud.

As the mobile node starts roaming in the HANs, it constantly computes the RNL_i metric (using Equations 1 through 5) for all networks I. Using the RNL_i values, the mobile node makes handoffs with low latency and packet loss that don't adversely affect the applications running on the mobile node. At the same time, the mobile node retrieves the best cloud k from the anchor point and offloads the computation/storage based on the selected cloud k.

Results

To validate our proposed system, we developed an activity recognition application service that uses various sensors (accelerometers, temperature sensors, GPS, and so on) to determine responders' activities in an emergency evacuation. The use of activity recognition applications in areas such as cognitive assistance, emergency healthcare, and emergency management will likely increase significantly in the near future. In these areas, an activity recognition application might require a large amount of sensor data collection, fast activity recognition, and timely delivery of results to the user (responders and command center). However, performing all these steps on a responder's mobile node could reduce battery lifetime and/or increase latency.[8,13,14] Therefore, we envision the system performing activity recognition on clouds instead of mobile nodes to maximize battery lifetime while providing low-latency computation and access. The major challenges posed by activity recognition applications running on clouds are

- efficient data collection from sensors;
- timely sensor data delivery to cloud-based activity recognition applications;
- timely activity recognition using activity recognition algorithms; and
- timely results delivery to the user.

We focus on the latter three tasks. For timely sensor data and results delivery, the mobile node must select the best network i that provides high throughput and low packet loss and delay. For timely activity recognition, the mobile node must select the best cloud k that determines the responder's activity in a timely manner.

Prototype Implementation

We consider a scenario in which a responder using a mobile node roams seamlessly in an emergency area (see Figure 1). While roaming, the activity recognition application running on the responder's mobile node collects sensory data and sends it to the best cloud, k, for processing. Upon recognizing the responder's activity, the cloud k delivers the results to the mobile node/command center for emergency situation-awareness. To validate $\mathrm{M^2C^2}$ in such a scenario, we developed an $\mathrm{M^2C^2}$ prototype that uses several components (see Figure 2)—Wi-Fi and 3G

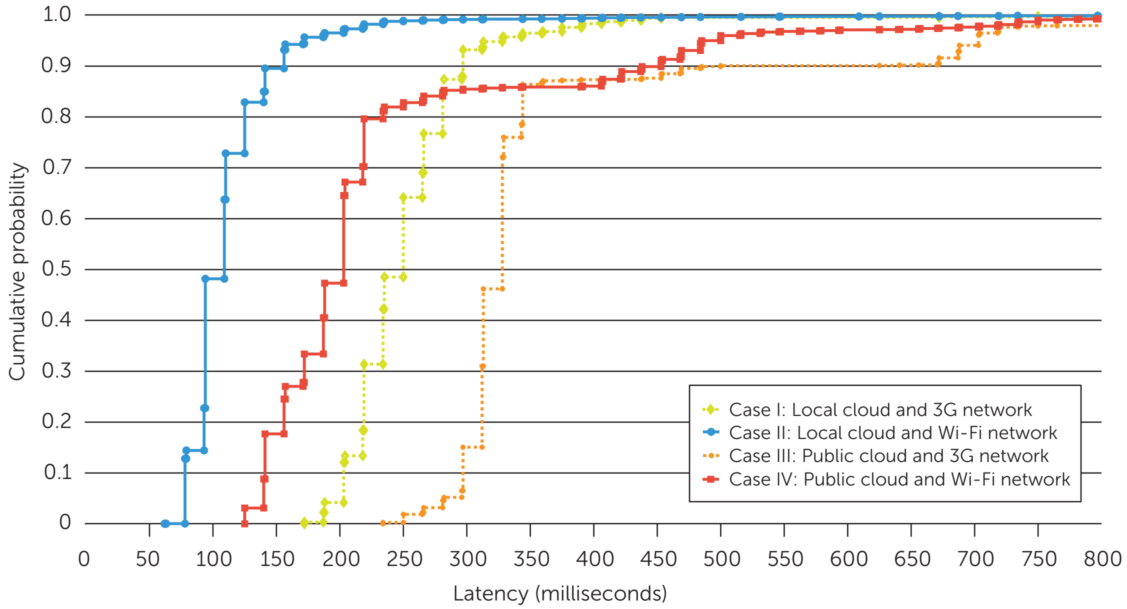

FIGURE 3. Cumulative distribution function of overall latency of performing activity recognition on local and public clouds using 3G and Wi-Fi networks.

networks, M-MIP, home agent, CPS and CRS running on the anchor point, the activity recognition application running on a mobile node, and the activity recognition service running on several cloud instances (both local and public). We developed the M-MIP prototype using C++ to handle multihoming in HANs.

For local clouds, we used Apple Macbook Pro computers with 16 Gbytes RAM and Intel i7 2.8-GHz processors. For public clouds, we used Amazon EC2 micro instances running in the "eu-west-1c" region. To probe the Amazon EC2 instances, we used Apache jclouds, an open source multi cloud toolkit for Java (https://jclouds.apache.org), APIs to programmatically orchestrate cloud operations such as starting, stopping, and terminating the virtual machines. We can also use this toolkit to gather QoS stats for the Amazon cloud instances. To gather QoS stats from the local cloud, we used the SIGAR Java API (https://support.hyperic.com/display/SIGAR/Home), which provides statistics such as CPU utilization and memory usage. The CPS and CRS ran as RESTful Web services on the anchor point in an emergency area. The CRS sent the GET calls to the CPS to retrieve the QoS stats for all clouds. It then computed cloud ranks $\mathfrak{R}_k$ for all clouds K (using Equations 6 through 8), to determine the single best cloud, k, for performing the responder's activity recognition.

We implemented the situation and context aware activity recognition (SACAAR) algorithm[15] as a RESTful Web service on local and public clouds to determine the responder's activities. The mobile node subscribed to the CRS and retrieved the URL of the kth cloud. It then sent the sensor data to this cloud for performing activity recognition.

Experimental Analysis

We performed extensive experimentation using several scenarios to validate M^2C^2. First, we studied the overall latency of performing activity recognition on both local and public clouds. Our aim with this experiment was to benchmark the best and worst case scenarios while the responder is constantly connected to the Internet via Wi-Fi or 3G networks using M-MIP. Time-critical applications such as augmented reality, virtual reality, cognitive assistance, and activity recognition are extremely sensitive to latency.[8,13]

Kiryong Ha and his colleagues have suggested that for these applications classes, the overall latency shouldn't be more than a few tens of milliseconds.[8] In this article, we consider the case of activity recognition, where the overall latency of performing activity recognition on clouds is the summation of:

- the time taken by the mobile node to send data to the activity recognition service running on clouds;
- the time taken by the activity recognition algorithm to determine the responder's activity; and

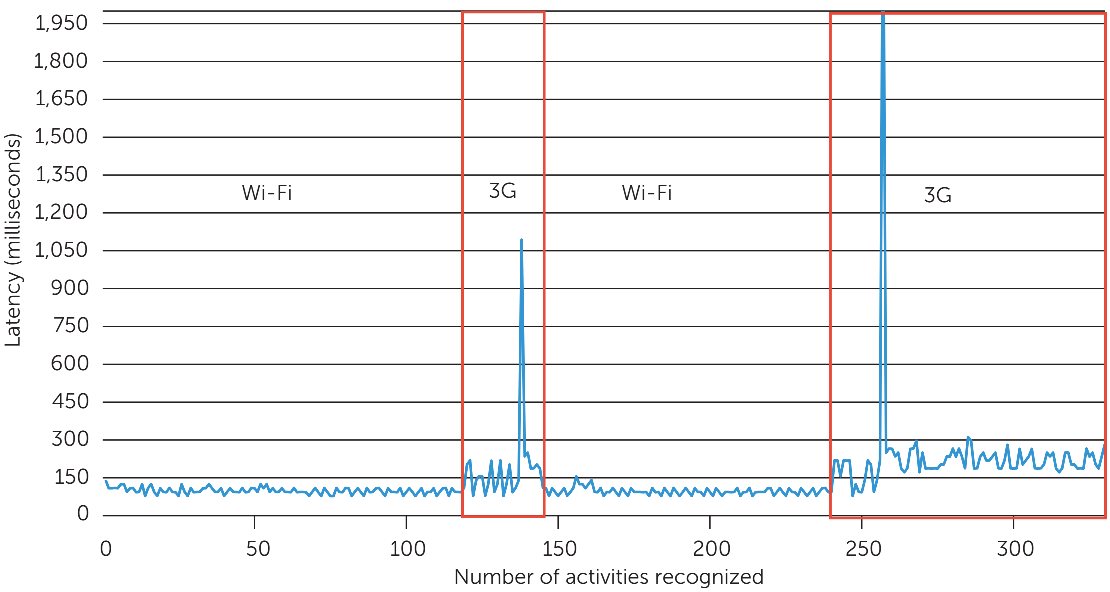

FIGURE 4. Overall latency of performing activity recognition on local cloud while the mobile node handoff's between the Wi-Fi and 3G networks.

- the time taken by the activity recognition service to return the results to the activity recognition application running on the mobile node.

Figure 3 shows the cumulative distribution function of the overall latency of performing activity recognition on local and public clouds while using Wi-Fi and 3G networks, respectively. For this experiment, we considered four cases:

- local cloud using a 3G network,
- local cloud using a Wi-Fi network,
- public cloud using a 3G network, and
- public cloud using a Wi-Fi network.

For each case, the mobile node's activity recognition application made an average of 3,353 (N = 13,412) API calls to activity recognition services running on clouds.

From this experiment, we concluded that, on average, the overall latency of performing activity recognition on the local cloud is significantly lower than on the public cloud for both Wi-Fi and 3G networks, where the Wi-Fi network performs significantly better than the 3G network. Wi-Fi offers significantly lower RTTs than a 3G network. For instance, the average latency (252.27 milliseconds) of performing activity recognition on a local cloud using a 3G network is more than twice as high as the average latency on a local cloud (113.90 ms) using a Wi-Fi network. Second, the latency of performing activity recognition on a public cloud using a 3G network is on average almost 130 ms higher than when using a public cloud with Wi-Fi. This is because the distance between a mobile node and the cloud contributes to the increase in overall wide area network latency. From this experiment, we conclude that local clouds perform approximately 60 percent better than public clouds for activity recognition on both 3G and Wi-Fi networks. We also assert that the computation should be performed as much as possible at local clouds placed in an emergency area using Wi-Fi networks, assuming sufficient compute and storage resources are available at local clouds.

After benchmarking both local and public clouds using HANs, we performed experiments to determine whether M^2C^2 can select the best cloud, k, and the best network, i, under stochastic network and cloud conditions such as network congestion, handoffs, and variation of CPU utilization (related to clouds). In our second set of experiments, the home agent, CPS, and CRS ran on the anchor point (see Figure 2). To validate CPS and CRS, we deployed our activity recognition Web service on two micro instances on a public cloud (Amazon EC2). We assigned these instances two separate public IP addresses—54.77.183.180 (cloud 1) and 54.77.218.113 (cloud 2)—and used CPS to probe them for CPU utilization. To verify the correctness of CPS and CRS, we generated a random synthetic workload using the stress utility on one of the public cloud instances and imposed a load of 100 percent on the CPU cores. We didn't put additional stress on the other cloud instance. Based on the randomly generated workloads and repeating the experiments more than a hundred times, we concluded that the CRS correctly selected the best cloud *k*. Whenever a responder's mobile node requested *k*, the CRS gave the IP address for *k* to the mobile node, which then used *k* to perform activity recognition. The activity recognition Web service running on cloud *k* then inferred the user activity and sent it back as the response to the mobile node.

As a third and final experiment, we studied whether M^2C^2 can support seamless handoffs (with low latency and packet loss) while the responder is on the move in an emergency area and his or her mobile node is connected via 3G and Wi-Fi networks. In this experiment, we started the activity recognition application on the mobile node after initializing it. The activity recognition application probed the CRS for the best cloud instance, *k*, using the APIs. The mobile node also probed the anchor point (running home agent) for the best network, *i*, by computing the RNL metric for all networks *I*. In our experiments, we observed that the RNL values for the Wi-Fi network were less stable than those for the 3G network. Initially, the mobile node was connected to the Wi-Fi network because it had lower RNL values than the 3G network. After approximately 13 seconds ($t = 13$ sec) into the experiment, the mobile node made a successful handoff without packet loss to the 3G network and stayed with that network for nearly 4 seconds, after which it made another successful handoff to the Wi-Fi network. Then, at time $t = 23$ sec, the mobile node performed another successful handoff from the Wi-Fi to the 3G network. Figure 4 shows the results related to handoffs and their effect on the overall latency of performing activity recognition. As the figure shows, the latency of performing activity recognition on a 3G network is higher than that on a Wi-Fi network. Most importantly, we observed that during the handoff process, the mobile node didn't experience any packet loss and the activity recognition process wasn't disrupted.

Our results clearly validate that M^2C^2 can efficiently support time-critical applications in emergency management scenarios while responders are on the move. M^2C^2 successfully tracked the QoS parameters of all clouds *K* and all wireless networks *I* to determine the best cloud *k* and network *i* for activity recognition.

We're in the process of integrating speech and location recognition and face detection application to M^2C^2, which we believe might be beneficial for evacuating people in an emergency area. The proposed system can also easily be extended to include other applications that can be valuable in emergency evacuation scenarios, such as optical character recognition and augmented reality.

References

1. "Indonesia Quake Toll Jumps Again," BBC News Report, 25 Jan. 2005, http://news.bbc.co.uk/2/hi/asia-pacific/4204385.stm.
2. R. R. Rao et al., *Improving Disaster Management: The Role of IT in Mitigation, Preparedness, Response, and Recovery*, Nat'l Academies Press, 2007.
3. P. Currion, C. de Silva, and B. Van de Walle, "Open Source Software for Disaster Management," *Comm. ACM*, vol. 50, no. 3, 2007, pp. 61–65.
4. Á. Monares, "Mobile Computing in Urban Emergency Situations: Improving the Support to Firefighters in the Field," *J. Expert Systems with Applications*, vol. 38, no. 2, 2011, pp. 1255–1267.
5. M. Tsai and N. Yau, "Improving Information Access for Emergency Response in Disasters," *Natural Hazards*, vol. 66, no. 2, 2013, pp. 343–354.
6. Z. Sanaei et al., "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges." *IEEE Comm. Surveys Tutorials*, vol. 16, no. 1, 2014, pp. 369–392.
7. M. Satyanarayanan et al., "The Role of Cloudlets in Hostile Environments," *IEEE Pervasive Computing*, vol. 12, no. 4, 2013, pp. 40–49.
8. K. Ha et al., "Towards Wearable Cognitive Assistance," *Proc. 12th Ann. Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 14), 2014, pp. 68–81.
9. A. Gani et al., "A Review on Interworking and Mobility Techniques for Seamless Connectivity in Mobile Cloud Computing," *J. Network and Computer Applications*, vol. 43, Aug. 2014, pp. 84–102.
10. M. Armbrust et al., "A View of Cloud Computing," *Comm. ACM*, vol. 53, no. 4, 2010, pp. 50–58.
11. K. Andersson, D. Granlund, and C. Åhlund, "M4: Multimedia Mobility Manager: A Seamless Mobility Management Architecture Supporting Multimedia Applications," *Proc. 6th Int'l Conf. Mobile and Ubiquitous Multimedia*, 2007, pp. 6–13.
12. S.H. Zanakis et al., "Multi-Attribute Decision Making: A Simulation Comparison of Select Methods," *European J. Operational Research*, vol.

107, no. 3, 1998, pp. 507–529.
13. M. Satyanarayanan et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, 2009, pp. 14–23.
14. N. Fernando, S.W. Loke, and W. Rahayu, "Mobile Cloud Computing: A Survey," *Future Generation Computer Systems*, vol. 29, no. 1, 2013, pp. 84–106.
15. S. Saguna, A. Zaslavsky, and D. Chakraborty, "Complex Activity Recognition Using Context-Driven Activity Theory and Activity Signatures," *ACM Trans. Computer–Human Interactions*, vol. 20, no. 6, 2013, article 32.

KARAN MITRA *is a postdoctoral researcher at Luleå University of Technology, Sweden. His current research interests include cloud computing, quality of experience measurement, modeling, and prediction, context-aware computing, and Internet of Things. Mitra has a dual-badge PhD in information technology and media technology from Monash University, Australia, and Luleå University of Technology. He's a member of IEEE and ACM. Contact him at karan.mitra@ltu.se.*

SAGUNA *is a postdoctoral researcher at Luleå University of Technology, Sweden. Her research interests include activity recognition, context-awareness, mobile and pervasive computing systems, smart grids, Internet of things, and cloud computing. Saguna has a PhD from Monash University, Australia and Luleå University of Technology as a double degree in information technology and media technology She's a member of IEEE and ACM. Contact her at saguna.saguna@ltu.se.*

CHRISTER ÅHLUND *is a chaired professor of pervasive and mobile computing at Luleå University of Technology, Sweden, and a scientific director of excellence in research and innovation. His research interests include Internet mobility, wireless access networks, Internet of Things, and cloud computing. Åhlund has a PhD in media technology from Luleå University of Technology. Contact him at christer.ahlund@ltu.se.*