2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

Change as Chance: Transition-enabled Monitoring for Dynamic Networks and Environments

Nils Richerzhagen*, Patrick Lieser*, Björn Richerzhagen*, Boris Koldehofe*,
Ioannis Stavrakakis† and Ralf Steinmetz*
* Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany
{nils.richerzhagen|patrick.lieser|bjoern.richerzhagen|boris.koldehofe|ralf.steinmetz}@kom.tu-darmstadt.de
† National and Kapodistrian University of Athens, Greece – IMDEA Networks & UC3M, Spain
ioannis@di.uoa.gr

***Abstract*—Future mobile applications will increasingly rely on direct communication among devices or with the environment—e.g., sensors—to provide interactive experiences to their users. The resulting communication characteristics are highly dynamic due to mobility and social behavior of humans, requiring applications to continuously adapt to the prevailing conditions. To this end, they require accurate information about the current state of the network. This information is obtained by monitoring mechanisms. Current mechanisms, however, are limited in their applicability in such a dynamic scenario, given that they only perform well for a limited range of environmental conditions.**

In this work, we perform an in-depth analysis of these limitations. We propose a monitoring service that executes transitions between individual state-of-the-art monitoring mechanisms to adapt its operation to dynamic network conditions. We evaluate a prototype of the transition-enabled monitoring service to study the impact of transition execution on the performance of continuous monitoring. Our results indicate that the achieved recall can be more than doubled while the latency can be reduced in the order of magnitudes compared to state-of-the-art monitoring approaches.

I. INTRODUCTION

Wireless communication experiences massive growth, driven by new applications and services for the Internet of Things (IoT) and opportunistic mobile social networks. Here, personal devices such as smartphones dynamically interconnect with each other and with additional—resource-constrained—devices like sensors in smart environments.

To support such dynamic behavior among heterogeneous mobile devices, the research community focuses on adaptation and self-organization of mechanisms and applications, e.g., for content-delivery networks [1]. However, to enable adaptation and self-organization, gathering information about the current state of the network and applications is an essential prerequisite. Following the well-known MAPE-cycle (*monitor*, *analyze*, *plan*, and *execute*), this information is then used to reason about potential adaptations that are to be executed. To this end, countless monitoring mechanisms have been proposed in literature. However, current monitoring mechanisms, such as [2]–[4], are limited with respect to their applicability in highly dynamic scenarios. Thus, these mechanisms are not able to provide robust and accurate monitoring under dynamic network and environmental conditions.

In this work, we target this limitation by proposing a monitoring service that executes transitions between distinct monitoring mechanisms depending on the current application requirements and environmental conditions. Based on an in-depth analysis of current monitoring mechanisms the design of the transition-enabled monitoring service allows for the incorporation of state-of-the-art monitoring mechanisms. We assess the performance characteristics of the individual monitoring mechanisms and study the impact of executing transitions in-between these mechanisms within the Simonstrator platform [5], [6]. Within our evaluation, we focus on (i) the cost and performance of different strategies to spread the transition decisions and (ii) the impact of the execution of transitions on the performance of continuous monitoring. Our results reveal that the achieved monitoring quality can be improved significantly by relying on transitions when monitoring dynamic environments compared to current state-of-the-art mechanisms.

The remainder of this paper is structured as follows. In Section II, we detail the characteristics of the scenario considered in this work and discuss its dynamics. Afterwards, we classify and discuss relevant related work on monitoring solutions for mobile wireless networks in Section III. In Section IV, we present our *core contribution*: the transition-enabled monitoring service that is able to incorporate state-of-the-art monitoring mechanisms and execute transitions in-between these individual mechanisms. We further discuss how to spread transition decisions in the network in a resource-efficient fashion, given that monitoring is considered a background service. We evaluate a prototype of our proposed service and present the results of the in-depth simulation based evaluation in Section V. Section VI concludes this work and outlines potential for future work.

II. SCENARIO

We consider a scenario where mobile devices are not only connected to the Internet, but also dynamically interconnect with each other and—potentially—with additional sensors in a smart environment to enable novel services and applications to end users. Examples constitute future opportunistic mobile social networks and applications that rely on direct interaction with smart spaces as envisioned for the IoT. However, existing

ISBN 978-3-903176-02-7 © 2018 IFIP

mobile social applications such as augmented reality games—for example, Google's *Ingress* or *Pokémon Go*—can also benefit from direct and spontaneous interconnection among users to offer a more interactive gameplay. All of the above applications require information about their current environment in terms of available network capacity, nearby devices, and additional application-specific metrics to adapt themselves to the prevailing conditions.

The conditions themselves are determined by human behavior, most prominently by mobility, attraction to specific places, social ties, and interaction patterns. This determines the load on the cellular infrastructure and, consequently, the achievable service quality and availability of the cellular communication for the respective applications. It is assumed that with increasing load on the cellular infrastructure its operation becomes unreliable once passing a certain threshold [7]. When considering mobile applications and human behavior, obstacles such as roads and buildings need to be considered in the scenario. People can walk on pathways and streets and are able to communicate directly with other people in range of the utilized communication interface. To this end, today's off-the-shelf devices are equipped with interfaces for Wi-Fi, Bluetooth, and Bluetooth LE.

We assume that our proposed transition-enabled monitoring service runs on all mobile devices within the considered scenario. In addition, we assume the existence of an application that generates monitoring requests, e.g., is interested in periodic updates of each node's neighborhood information. This application is used to measure the accuracy of monitored information and the performance of the monitoring service under varying conditions, as discussed in detail in Section V.

III. RELATED WORK

We classify existing monitoring mechanisms based on their communication characteristics as *centralized*, *decentralized*, or *hybrid*. Representative approaches for each category are discussed in the following, forming the foundation for our proposed transition-enabled monitoring service.

a) Centralized Monitoring: Centralized approaches do not use local on-demand wireless communications. Instead, they rely on a single powerful instance on the edge of the network to collect information from the mobile devices and calculate statistics based on that. This limits their applicability concerning today's and future networking scenarios such as IoT and opportunistic mobile social networks where the interconnectivity among users and sensors is a key characteristic.

Most centralized monitoring approaches are of commercial nature such as the solutions from SevOne[1] and Qosmotec[2]. Commercial products and approaches introduced by the research community such as ChuckWa [8] provide good performance unless the underlying communication mean is subject to dynamic quality fluctuations [7]. With the previously introduced limited applicability of those approaches for many of today's networking scenarios considered in IoT or opportunistic mobile social networks, e focus on the hybrid and decentralized monitoring approaches in this section.

b) Decentralized Monitoring: Decentralized approaches are organized in a flat or hierarchical structure, relying solely on on-demand wireless communication or disruption tolerant strategies. In hierarchical approaches the mobile devices obtain different roles, which are calculated relying on decentralized gateway-selection and clustering solutions [9], [10]. Decentralized monitoring approaches are usually applied when the infrastructure is overloaded or broken. However they can also be used for locality reasons, thus to keep information local.

The authors of [11] propose a hierarchical solution in which gateways are elected to monitor relevant information. Gateways, called MeshLeader, have to monitor their k-hop neighborhood to obtain a detailed local view. A sparse global view is obtained by gateways sharing their local views with other gateways, which helps in improving the monitoring result. However, using static nodes and, thereby, not considering mobility is a key limitation of [11]. Similarly to [11], the approach presented in [12] relies on a two-tiered topology. Still, gateways are non-mobile and pre-defined, which reduced the adaptability of the approach significantly. The approaches in [11], [12] can hardly guarantee a good monitoring quality under dynamic network conditions. The approach of Battat et al. [2] uses a three-tiered topology based on weights of the nodes to obtain monitoring data in a mobile network. Their approach strongly depends on the usage of a single routing protocol shared for data and management communication, which may not perform as well in different network conditions. In the system presented by Tuncay et al. [3], not all nodes are performing the measurements within an area of relevance. Instead, the load on individual nodes is reduced by selecting a subset of appropriate nodes that should do the monitoring. However, relying on an a-priori knowledge obtained by so called recruiting nodes, which decide which of the nodes should perform the monitoring, may render the approach inefficient in current on-demand networks, where mobility patterns and interconnections are not previously known.

c) Hybrid Monitoring: Hybrid monitoring approaches reduce the load on the edge infrastructure by offloading a fair share of the communication needed to distribute and collect the monitoring information to local on-demand wireless networks. The offloading is realized by a selected subset of nodes acting as gateways. Gateways, calculated by gateway selection and clustering solutions, act as relay points to connect to the edge infrastructure. Gateway collect and distribute information from non-gateway nodes using local on-demand wireless networks.

Al-Radaideh et al. [4] propose a structural health monitoring system for highway bridges. Equipping the bridges with a wireless sensor network and a master coordinator, which acts as gateway, the authors send locally collected information from the sensors to a cloud-based server. In an earlier work [13], we propose a hierarchical monitoring system that uses gateways for communication with a cloud-based server. Non-gateway nodes are overhearing advertisements from gateways, to con-

---

[1] https://www.sevone.com/solutions/4g-lte-wireless-network-monitoring

[2] https://www.qosmotec.com/products/mobile-network-tester/

nect to them using a multi-hop, contention-based forwarding scheme [14], [15]. While our approach shows benefits over a centralized solution, its applicability to dynamic and mobile scenarios is limited as a consequence of the static gateway selection. Therefore, in this work, we aim to benefit from the full range of available monitoring mechanisms by incorporating them into a transition-enabled monitoring service. Within this monitoring service the currently utilized mechanism can be exchanged dynamically at runtime.

IV. ADAPTMON: TRANSITION-ENABLED MONITORING

To allow the usage of different monitoring mechanisms, going beyond the excerpt discussed in the previous section, we propose the transition-enabled monitoring service AdaptMon. By exchanging the active monitoring mechanism at runtime, AdaptMon allows us to utilize the mechanism that performs best under the current conditions, e.g., network load and dynamics. Consequently, AdaptMon allows for centralized, decentralized, and hybrid monitoring mechanisms to be executed to provide accurate monitoring information to applications. In the following, we discuss the design of AdaptMon. First, the components of AdaptMon are explained in Section IV-A. This includes the requirements that need to be fulfilled to allow transitions between distinct monitoring mechanisms. Afterwards, the transition coordination components, including the spreading of transition decisions, are explained in Section IV-B.

A. *Design of the Transition-enabled Monitoring Service*

The transition-enabled monitoring service needs to represent monitoring approaches from each of the classes identified in Section III. Centralized, hybrid, and decentralized monitoring approaches comprise of different communication patterns. Accordingly, AdaptMon contains components that map to these communication patterns. In the following, we differentiate between client- and server-side components. Client-side components are shown in Figure 1. Three components are responsible for the communication of monitoring information: (i) cellular upload, (ii) local dissemination, and (iii) local collection. These components are all linked to the *request/response resolving* component, which accesses the local information storage containing a protocol-independent measurement mechanism. By being protocol-independent, the local information storage can measure information unaffected by any transitions, as discussed in [16], [17]. The resolving component is used to evaluate whether received requests or responses are of interest for the client. Therefore, incoming requests and responses are checked for validity and whether they should be answered or enriched with additional information by this client, e. g., by aggregating information from multiple sources.

a) Using Requests and Responses: Requests contain a unique identifier consisting of the requested metric, the type of the request, its scope, and (optionally) an aggregation function. This information is provided by mechanisms that register as information providers at the local information storage [16], [17]. By specifying a request type, applications can state whether they require the respective metric (i) only once (one-shot), (ii) periodic with a given interval, or (iii) event-based, i.e., based on a condition that triggers an update. The scope specifies if the request targets a sub-set of clients, e. g., those that contain a special application, or if the request is valid for all monitored clients. An aggregation function can be provided by the requesting application to allow for custom data aggregation within the monitoring service. Responses contain the requesting client, the unique identifier, the validity, and the aggregation function from the request in addition to the collected values from other clients.

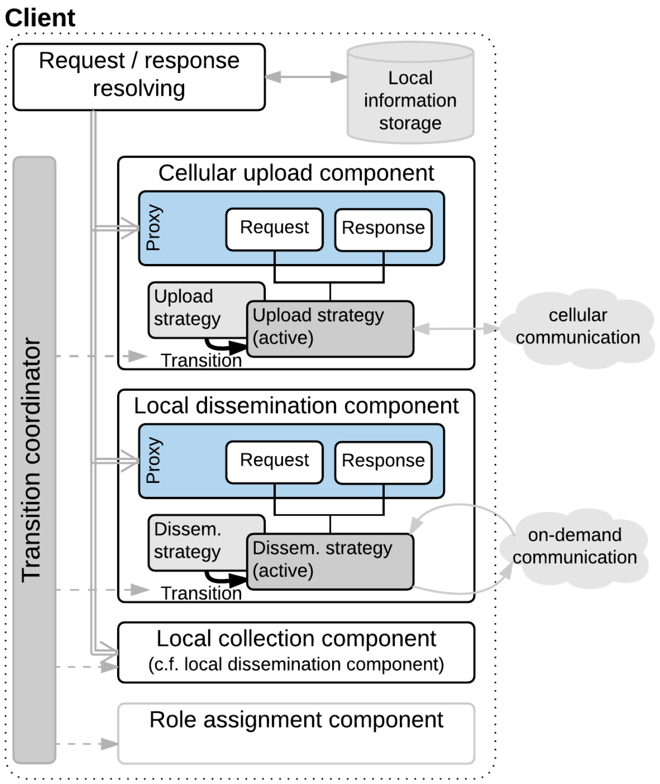


Figure 1: Client components of AdaptMon.

b) Encapsulating Communication Patterns: To allow for transitions between different communication patterns, the respective communication patterns are encapsulated as components within AdaptMon. The concept is explained for the client-side local dissemination component in the following. By applying it to all of the client-side components, one can compose arbitrary monitoring mechanisms within AdaptMon, representing the related monitoring approaches discussed in Section III. To allow for transitions between local dissemination strategies a common abstraction is needed. We rely on *transition-enabled proxies* for this abstraction within the monitoring service [18].

The proxies encapsulate the functionality of local dissemination strategies behind a common programming interface for requests and responses, as indicated in Figure 1. The proxy

Table I: Proxy configurations in AdaptMon.

Monitoring approach	Client proxies (request and response) Dissemination	Collection	Upload	Server proxy Dissemination
Centralized	none	none	direct	direct
Hybrid [13]	none	contention	gateway	gateway
Flood	flooding	flooding	none	none
Epidemic	epidemic	epidemic	none	none

used in the dissemination component provides a *disseminate* method used by clients to start the dissemination of new requests or responses encapsulated in a *LocalDisseminationMessage*. When such messages are received by a client the *onReceive* method is invoked on the respective mechanism hidden behind the proxy (shown as active strategy in Figure 1).

Within the current prototype of AdaptMon, we integrated a multitude of ad-hoc dissemination protocols, such as contention-based schemes [13] or hierarchical cluster-based dissemination schemes [4], [11]. Furthermore, we incorporated the disruption tolerant dissemination strategies proposed in [19]. The client-side components for cellular upload and local collection are encapsulated following the same design principle. The cellular upload proxy, for example, contains strategies for direct and gateway-assisted uploading as seen in [9], [10]. Correspondingly, the cellular dissemination proxy on the server contains strategies for sending monitoring information to mobile clients.

c) Executing Transitions: Relying on both, server- and client-side proxies, AdaptMon is able to execute transitions between centralized, hybrid, and decentralized monitoring approaches. Table I shows the configurations of AdaptMon to embody different monitoring approaches used in this work. Note that this is only an excerpt of the possible monitoring approaches that AdaptMon supports in its current design stage and implementation. In the centralized configuration, the system does not rely on any local communication. For that reason, only the local upload proxy on the client and the cellular dissemination proxy are configured to send the incoming monitoring information directly to the targeted clients. To represent the behavior of our previously published hybrid monitoring solution [13], AdaptMon is configured as follows (cf. Table I): Clients and the cloud-based server only communicate with each other when the respective client is selected as a gateway. The assignment of gateways is achieved using a centralized role assignment service [9], [10]. Local communication between clients is based on a contention-based collection of responses. As [13] focuses on the collection of results, the delivery of requests is not described. Thus, to allow for comparison with the other configurations discussed in the following, we assume requests are delivered relying on the centralized communication between clients and server.

Transitions between the different monitoring approaches within AdaptMon are achieved by exchanging the strategies used in the proxies. To this end, each strategy has to implement basic life-cycle methods as defined in [18]. The life-cycle methods ensure atomicity of a transition execution, thus guaranteeing a correct execution of transitions on the respective components. A transition for client c between strategy A and B is defined as $\tau^{c}_{A \rightarrow B}$. If a transition targets only a subset of nodes, it is referred to as *regional*, whereas if it targets all nodes, it is referred to as *global transition*. In cases where a transition only leads to a reconfiguration of the already running mechanism, it is referred to as a *self-transition*. We assume that transitions within AdaptMon originate from trusted edge devices or gateways. However, as it cannot be guaranteed that these devices can reach all affected nodes, the decision to execute a transition needs to be spread in the network. Respective mechanisms for transition decision spreading are explained in Section IV-B. When a transition from strategy A to B is executed, it is important that relevant and supported state of strategy A is transferred to the following strategy B. As an example, the neighborhood of a client is state that may be of interest for the subsequent strategy. The state transfer during transitions is explained in the following paragraph.

d) State Transfer: To support seamless execution of transitions, AdaptMon supports two types of state transfer. First, state can be transferred within a proxy from one mechanism to the subsequent one during a transition, as discussed in [20]. In addition, within AdaptMon, we propose a state transfer mechanism that supports migrating state from one proxy to another. Consequently, state transfer is no longer limited to mechanisms within one single proxy, which is an essential step in supporting transitions between completely different monitoring structures (e.g., centralized to decentralized).

In both cases, state transfer does not include any additional communication as otherwise atomicity of transitions cannot be guaranteed. To allow for state transfer between different proxies, a predetermined order of transitions must be satisfied. We take the transition from decentralized monitoring using epidemic local communication to a centralized monitoring solely using cellular communication as an example. This transition leads to a set of transitions on multiple proxies within different components as visible in Table I. In the decentralized monitoring clients buffer the requests and responses received over time. With the transition from decentralized to centralized monitoring, those buffered requests and responses would be lost, which in return would cause a significant performance drop. By allowing state transfer between proxies, AdaptMon is able to overcome this limitation. A persistent storage component is used to register the respective state transfers. In the following, we formalize this process on the example of an abstract transition consisting of two transitions, one within proxy X and another one within proxy Y. The following attributes must be specified for the transition $\tau^{c}_{A_X \rightarrow B_X}$ on proxy X and a transition $\tau^{c}_{A_Y \rightarrow B_Y}$ for proxy Y:

```
transferState(proxy X<→ B_X>, proxy Y<→ B_Y>, ...
... Set<stateVariable>)
```

Using the *transferState* method ensures that the state between proxies X and Y is only transferred when: (i) a transition on proxy X to strategy B_X is executed and (ii) a transition on proxy Y to strategy B_Y is executed at the same time. If this is the case, the given set of *state variables* is transferred between the two proxies X and Y.

B. Transition Coordination and Decision Spreading

To coordinate the execution of transitions on clients, each client is equipped with a transition coordination component (cf. Figure 1). This component can include additional means to distribute transition decisions within the network if the transition is a regional or global transition. However, as monitoring is a background service, the overhead introduced by spreading transition decisions in the network should be kept minimal. We evaluate two spreading strategies integrated into the coordinator: a server-assisted strategy that relies on cellular communication and a piggybacking strategy that does not induce any additional communication to spread a decision. The piggybacking strategy appends the transition decisions to each message that is sent by a client via the local on-demand network. Even if its performance may not be the best concerning latency and reliability of the decision spreading, it introduces only minimal overhead in the network, which is important for ADAPTMON. In the following, we evaluate the impact of transition execution on the monitoring service, including an analysis of the impact of the spreading strategy.

V. EVALUATION

The evaluation of the transition-enabled monitoring service ADAPTMON addresses three main aspects. First, monitoring mechanisms representing the different categories identfied in the related work are compared under dynamic environmental conditions, identifying the potential for transitions (Section V-B). Second, we evaluate the impact of transition decision spreading, given that the monitoring service should operate in a resource-efficient fashion (Section V-C). Third, we assess the impact of transitions between monitoring mechanisms on the performance of continuous monitoring with ADAPTMON (Section V-D). This includes a comparison of ADAPTMON against stand-alone state-of-the-art monitoring mechanisms. Before discussing our results, we detail the evaluation setup in the following section.

A. Evaluation Setup, Scenario Model and Metrics

We evaluate a prototype of ADAPTMON within the Simonstrator framework [5], relying on a social movement model that further utilizes OpenStreetMap map data for realistic node mobility, as proposed in [6]. The source code of the platform as well as our research prototype of ADAPTMON is available online[3] for interested readers. For local on-demand communication between nodes, we rely on the Wi-Fi 802.11g model included in the ns-3 network simulator [21].

As motivated, in our scenario the cellular coverage and the quality of the connection varies with the density of nodes in an area [7]. We model the threshold after which the cellular connection becomes unreliable as a fraction of nodes in the scenario. Given that the communication characteristics are heavily influenced by human mobility, we compare the performance for our social mobility model against a Gaussian mobility model and the random waypoint mobility model. Thereby, we can isolate the impact of social ties between users and the resulting group formations, as later discussed in Section V-B. We simulate mobility for the city center of Darmstadt (Germany), spanning an area of 1500×1500 meters. Studying different node densities allows us to compare the monitoring approaches in sparse and densely populated scenarios. To assess the performance of the monitoring service, we issue monitoring requests on at least two clients every 150 seconds. Each request targets all clients in the network and has a validity of ten minutes. The simulation setup is summarized in Table II, with default values being underlined. Two hours of operation are simulated, with measurements starting after a warm-up period of 10 minutes.

Table II: Scenario and simulation setup.

Max. Wi-Fi Range	$88\,m$
Cellular Network	$150\,ms, \pm 100\,ms$
Max. Cellular Conn.	min. $25\,\% - 100\,\%$; $\underline{50\,\%}$
Mov. Speed $[\frac{m}{s}]$	$1.5 - 2.5$
Movement Model	$\underline{\text{social [6]}}$, gauss [22], RWP [23]
Density $[\frac{\text{clients}}{km^2}]$	$22 - 222$; $\underline{88.8}$
Request	interval: 2.5 min; validity: 10 min
Mon. Approach	centralized [8], hybrid [13], flood, epidemic
Transition Spreading	$\underline{\text{server-assisted}}$, piggybacking

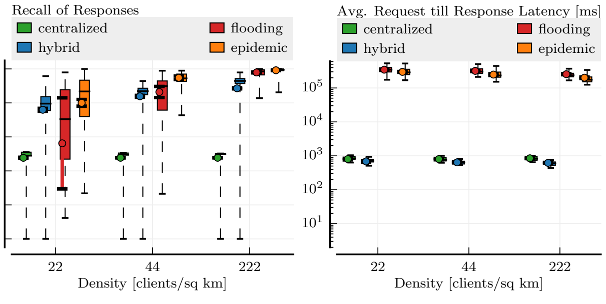
To assess the performance of our proposed monitoring service, we consider the following metrics: (i) the achieved recall of requests and responses, (ii) the average latency from request invocation till response reception, and (iii) the latency from transition execution at the source client till execution on other targeted clients. The distributions of the results are shown as box plots. The median is represented by a solid line inside the box, while the lower and upper quartile are represented by the boxes. Whiskers show the largest and smallest data point within 1.5 of the interquartile range. As box plots show the results of a single simulation run, a marked dot with error bars is plotted to the left side of the boxes indicating the confidence intervals over ten repetitions with different random seeds.

B. Comparison of Monitoring Approaches

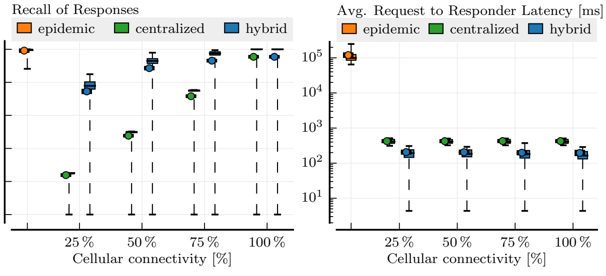
For the comparison of the monitoring approaches the following parameters are varied: (i) the density of clients, (ii) the underlying movement model, and (iii) the cellular connectivity in the scenario. The density variation provides insights into the scalability of the approaches. With $50\,\%$ of nodes being able to utilize the cellular connection, the following can be observed when looking at the recall of responses (cf. Figure 2(a)) and the latency from request till response (cf. Figure 2(b)). Disruption tolerant approaches do need a minimum density in the network to perform, which is consistent with findings in [15]. Once this density is reached or exceeded, decentralized approaches delivered a better recall compared to the centralized and the hybrid approach, where access to cellular communication is restricted. The latency, shown in Figure 2(b), unveils that centralized and hybrid monitoring approaches deliver superior latency (more than

---

[3] www.simonstrator.com



(a) Recall of responses (b) Latency request till response



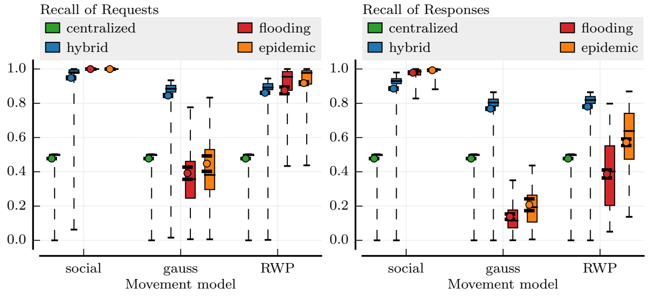
(c) Recall of responses (d) Latency of requests

Figure 2: Impact of the client density (2(a), 2(b)) and the cellular connectivity (2(c), 2(d)) on the performance.

two magnitudes better) compared to the disruption tolerant approaches or epidemic information dissemination. However, the latency of successfully delivered information does not further differ with changing density.

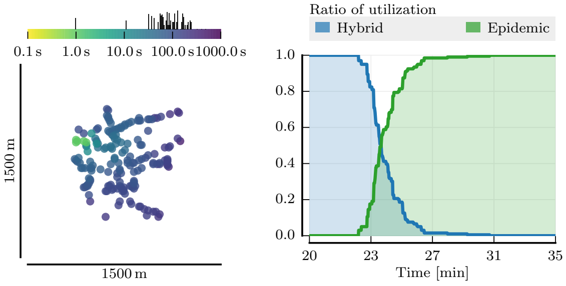
Varying the cellular connectivity affects only the centralized and hybrid monitoring approaches, given that the decentralized approaches do not use the cellular connection at all. Therefore, we only report results for the epidemic approach for comparison. The recall of responses shown in Figure 2(c) illustrates the strong dependency of the centralized monitoring approach on the cellular connectivity. As only nodes with a connection are able to route their monitoring information to the requester the resulting recall shows a linear dependency. The hybrid monitoring approach, however, shows the benefits of combining on-demand communication with cellular offloading of monitoring information. Still, the approach is limited in functionality if only a few clients (e. g., 25 % or less) are able to use a cellular connection. The latency from request invocation till response reception does not depend on the cellular connectivity as shown in Figure 2(d). This is due to the fact that although less information is collected (i. e., lower recall of responses), the way the information is gathered—and, consequently, the latency—remains unchanged.

Mobility has a strong impact on all approaches that rely on on-demand networking. Figure 3(a) shows the recall of the requests when modeling human mobility with the social movement model [6] in comparison to the generic Gaussian mobility model (gauss) [22] and the random waypoint model (RWP) [23]. Correspondingly, in Figure 3(b) the recall of the responses is reported. The Gaussian movement model reduces the interconnection times of clients, which becomes apparent in both figures as both recall values decrease. For the decentralized monitoring approaches, an increased recall can be observed using RWP as clients meet more frequently and for longer periods. The recall of the hybrid approach is not affected significantly, as gateways are selected by the respective selection strategy to obtain good coverage based on the current location of clients. As expected, the centralized approach is not affected at all, given that only cellular communication is used in this case. Instead, the recall simply reflects the fraction of clients that can connect to the server (in this case, 50 %). Using the social mobility model, both recall measures increase significantly for approaches relying on local interaction between clients. Here, modeling human behavior by considering attraction to specific places and interaction with other humans is crucial to correctly assess the performance of the monitoring service.



(a) Recall of requests (b) Recall of responses

Figure 3: Impact of the mobility model.

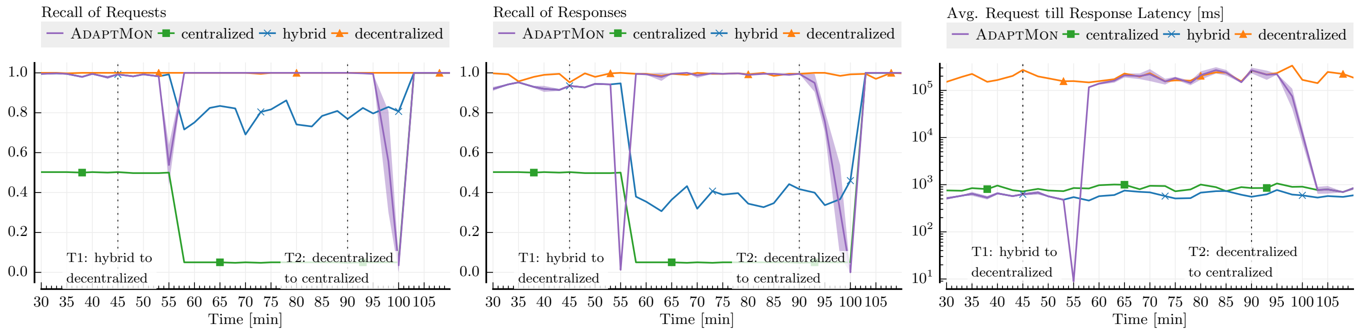


(a) piggybacking (b) gateway-assisted

Figure 4: Impact of the transition decision spreading strategy.

C. Spreading Transition Decisions

An important factor for the performance of AdaptMon is the strategy used to spread transition decisions among clients. A decision taken by a logically centralized entity, such as a server or a selected gateway, needs to be spread in the network. For the evaluation of different spreading strategies we select a single random client as source for a transition execution at minute 22. Due to space constraints, the results for a server-assisted spreading are omitted here. When using server-assisted spreading, the source of a decision uploads



(a) Average recall of requests

(b) Average recall of responses

(c) Average latency request till response

Figure 5: Comparison of the transition-enabled monitoring service AdaptMon and current state-of-the-art monitoring approaches under dynamic environmental conditions. Transition execution is indicated with vertical lines at marked times.

the transition decision to the server, which distributes that decision to all affected clients. Thus, the latency introduced by that approach corresponds to the round trip time of the cellular network. However, as this spreading strategy is rather intrusive regarding the overhead and its additional usage of the cellular network, AdaptMon can also use a piggybacking-based spreading strategy. This strategy does not introduce additional messages. Instead, as the name suggests, transition decisions are attached to every message that is sent by nodes. Figure 4(a) shows the latency of transition spreading and executions for the piggybacking strategy. Here, AdaptMon executes a transition from hybrid to decentralized monitoring (cf. Figure 4(b)). While being non-intrusive, the piggybacking strategy leads to a significant increase in the time till the transition is performed by all nodes in the network. As visible in Figure 4(b), it takes up to five minutes until most clients receive the transition decision. However, this is only reflecting the case where the monitoring workload is the only workload in the network. Considering the scenario of smart environments, more communication takes place between clients which results in a better spreading of the transition decision when relying on the piggybacking strategy.

D. Impact of the Transition Execution

To assess the execution of transitions within AdaptMon, we compare its performance against static configurations of monitoring approaches discussed in Section III. To model a dynamic scenario, we alter the availability of the cellular connection over time. A monitoring approach should be able to provide reliable and accurate monitoring even under such environmental fluctuations. The cellular coverage changes as follows. In the first 45 minutes up to 50 % of the clients are able to connect to the cloud-based server if required. Afterwards, for the next 45 minutes only 5 % of the clients are able to establish a connection, mimicking an overload situation or an outage. After 90 minutes, the cellular communication infrastructure is able to cater for all nodes whenever needed, corresponding to normal operation.

Figure 5 shows the comparison of the centralized [8], hybrid [13], and decentralized monitoring approach with AdaptMon. AdaptMon executes transitions between the individual approaches at the points in time where the connectivity changes significantly. Hence, AdaptMon initially uses the hybrid monitoring approach before executing a transition to the fully decentralized monitoring approach when the cellular connection becomes highly unreliable. Later, with the cellular communication infrastructure recovering AdaptMon executes a transition from decentralized to fully centralized monitoring. It is important to mention that results are shifted from the points in time where the transitions are actually executed. This is due to the fact that requests and responses are evaluated at the end of each validity period of 10 minutes (cf. Table II). Thus, the characteristics of a request invoked around the time of transition T1 (after 45 minutes) for example are visible around minute 55.

First, we examine the recall of requests (Figure 5(a)) and responses (Figure 5(b)). Both, the centralized and the hybrid monitoring approaches depend on the cellular network connectivity (cf. until T2 in Figure5(a)). Still, the hybrid approach is able to outperform the centralized solutions significantly as it further benefits from on-demand communication to disseminate requests in the network. The decentralized approach is able to provide a reasonably good constant recall as it is not affected by any changes in the cellular infrastructure. However, looking at the latency from request till response reception in Figure 5(c) the trade-off when completely relying on the disruption tolerant decentralized approach becomes clear. The latency of the decentralized approach is, as expected, significantly larger than the latency of the centralized and hybrid approaches, but still it is able to provide for high recall for both requests and responses. Thus, AdaptMon is executing transition T1 from the hybrid to the decentralized approach (cf. Figure 5). In doing so, AdaptMon is able to provide high recall during the time span with significantly disrupted cellular communication. Drops in the recall achieved by AdaptMon after each transition (cf. Figure 5(a), 5(b)) are due to missing state transfer (cf. Section IV-A) in this configuration. This causes the loss of all request and response values at the times of the transitions as the used proxy strategies change, motivating our state transfer mechanism proposed in Section IV. The effects can be reduced significantly with state transfer (not shown here due to space constraints).

The proposed transition-enabled monitoring service is able to provide high recall and low latency in face of changing scenario conditions by relying on transitions between monitoring approaches. Consequently, ADAPTMON can configure itself based on environmental conditions or application requirements to provide the required monitoring characteristics. Considering the time interval up to transition T1 (cf. Figure 5), ADAPTMON is able to provide a recall above 90 % while delivering results with lowest latency. Instead of dropping to recall values below 40 %, as the centralized and hybrid monitoring approaches do for responses, ADAPTMON keeps recall values above 98 % by executing transition T1 to the decentralized approach until transition T2. However, this comes at the cost of increased latency (cf. Figure 5(c)). Nevertheless, for the time before transition T1 and the time after transition T2 the achieved latency is reduced by the order of two magnitudes compared to the decentralized approach.

VI. CONCLUSIONS

Monitoring in mobile networks faces increasing dynamics caused by client mobility and human social behavior, combined with a trend towards direct connectivity between devices. However, current monitoring approaches address rather specific environmental conditions, limiting their applicability in such dynamic settings.

We therefore propose the execution of transitions between different monitoring approaches to adapt to the respective situation. To this end, we introduce the transition-enabled monitoring service ADAPTMON that enables seamless transitions between state-of-the-art centralized, decentralized, and hybrid monitoring approaches. We study the execution of transitions and their impact on the performance of our monitoring system in an in-depth evaluation. Specifically, we (i) identify the application ranges of individual monitoring approaches, (ii) study the cost and performance of distributing the transition decision in the network, and (iii) show how transitions enable continuous monitoring even under significant changes in the environment. Our results show that the latency for information collection is reduced significantly when utilizing the transition-enabled monitoring service in dynamic network conditions. Furthermore, the system achieves high recall even during highly dynamic conditions.

We are currently investigating the *best* location or entity to start the transition execution process. This allows for better spreading of transition decisions while also ensuring that transitions are planned with enough knowledge about the current networking conditions. Before starting transitions in the network, obtaining insights of the near future is essential. Thus, instead of executing transitions on random clients, the planning of transitions based on predictions of the network conditions in near future is highly relevant.

ACKNOWLEDGMENT

This work has been funded by the German Research Foundation (DFG) as part of projects B1, C2, and C3 within the Collaborative Research Centre (CRC) 1053 – MAKI and the LOEWE initiative (Hesse, Germany) within the NICER project

REFERENCES

[1] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, “Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems,” *IEEE Comm. Mag.*, vol. 52, no. 2, pp. 131–139, 2014.

[2] N. Battat and H. Kheddouci, “HMAN: Hierarchical Monitoring for Ad Hoc Network,” in *IFIP EUC*, 2011.

[3] G. S. Tuncay, G. Benincasa, and A. Helmy, “Autonomous and Distributed Recruitment and Data Collection Framework for Opportunistic Sensing,” *ACM SIGMOBILE*, vol. 16, no. 4, 2013.

[4] A. Al-Radaideh, A. R. Al-Ali, S. Bheiry, and S. Alawnah, “A Wireless Sensor Network Monitoring System for Highway Bridges,” in *ICEIT*, 2015.

[5] B. Richerzhagen, D. Stingl, J. Rückert, and R. Steinmetz, “Simonstrator: Simulation and Prototyping Platform for Distributed Mobile Applications,” in *ACM SIMUTOOLS*, 2015.

[6] N. Richerzhagen, B. Richerzhagen, D. Stingl, and R. Steinmetz, “The Human Factor: A Simulation Environment for Networked Mobile Social Applications,” in *IEEE NetSys*, 2017.

[7] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, “A First Look at Cellular Network Performance During Crowded Events,” in *ACM SIGMETRICS*, 2013.

[8] J. Boulon, A. Konwinski, R. Qi, A. Rabkin, E. Yang, and M. Yang, “Chukwa, a Large-scale Monitoring System,” in *Proceedings of CCA*, vol. 8, 2008, pp. 1–5.

[9] F. Rebecchi, M. D. de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, “Data Offloading Techniques in Cellular Networks: A Survey,” *IEEE Comm. Surv. Tut.*, vol. 17, no. 2, pp. 580–603, 2015.

[10] Y. Li, M. Qian, D. Jin, P. Hui, Z. Wang, and S. Chen, “Multiple Mobile Data Offloading Through Disruption Tolerant Networks,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1579–1596, 2014.

[11] S. Nanda and D. Kotz, “Mesh-Mon: A Multi-radio Mesh Monitoring and Management System,” *Comp. Communications*, vol. 31, no. 8, 2008.

[12] K. N. Ramachandran, E. M. Belding-Royer, and K. C. AImeroth, “DAMON: A Distributed Architecture for Monitoring Multi-hop Mobile Networks,” in *IEEE SECON*, 2004.

[13] N. Richerzhagen, D. Stingl, B. Richerzhagen, A. Mauthe, and R. Steinmetz, “Adaptive Monitoring for Mobile Networks in Challenging Environments,” in *IEEE ICCCN*, 2015.

[14] Y. Kiri, M. Sugano, and M. Murata, “Self-Organized Data-Gathering Scheme for Multi-Sink Sensor Networks Inspired by Swarm Intelligence,” in *IEEE SASO*, 2007.

[15] M. Zorzi and R. R. Rao, “Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.

[16] N. Richerzhagen, T. Li, D. Stingl, B. Richerzhagen, R. Steinmetz, and S. Santini, “A Step Towards a Protocol-independent Measurement Framework for Dynamic Networks,” in *IEEE LCN*, 2015.

[17] A. Le, J. Varmarken, S. Langhoff, A. Shuba, M. Gjoka, and A. Markopoulou, “AntMonitor: A System for Monitoring from Mobile Devices,” in *ACM SIGCOMM*, 2015.

[18] A. Frömmgen, B. Richerzhagen, J. Rückert, D. Hausheer, R. Steinmetz, and A. Buchmann, “Towards the Description and Execution of Transitions in Networked Systems,” in *IFIP AIMS*, 2015.

[19] D. Reina, S. Toral, P. Johnson, and F. Barrero, “A Survey on Probabilistic Broadcast Schemes for Wireless Ad Hoc Networks,” *Ad Hoc Networks*, vol. 25, pp. 263–292, 2015.

[20] B. Richerzhagen, N. Richerzhagen, J. Zobel, S. Schönherr, B. Koldehofe, and R. Steinmetz, “Seamless Transitions between Filter Schemes for Location-based Mobile Applications,” in *IEEE LCN*, 2016, pp. 348–356.

[21] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, “ns-3 Project Goals,” in *ACM WNS2*, 2006.

[22] B. Liang and Z. J. Haas, “Predictive Distance-based Mobility Management for PCS Networks,” in *IEEE INFOCOM*, 1999.

[23] T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for ad hoc Network Research,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.