# Bypassing the Cloud: Peer-assisted Event Dissemination for Augmented Reality Games

Björn Richerzhagen, Dominik Stingl, Ronny Hans, Christian Gross and Ralf Steinmetz
Multimedia Communications Lab, Technische Universität Darmstadt
Email: {richerzhagen, stingl, hans, chrgross, steinmetz}@kom.tu-darmstadt.de

***Abstract*—The rising number of mobile devices and their increasing computational capabilities enable new interactive context-sensitive applications. Popular examples are augmented reality games such as Google's Ingress, where users interact with each other in the real world while being part of the game at the same time. This local interaction pattern in the real world as well as in the game is not reflected in the underlying communication pattern. Every locally generated game event is first transferred to a backend server via a cellular connection, from where it is then further disseminated to all players within the given area of interest. This communiation pattern introduces significant delays and limits the interactivity of the game. In this work, we propose an event dissemination system that exploits the locality characteristics of mobile augmented reality games to (i) enable and configure local peer-to-peer dissemination of events when appropriate and (ii) reconfigure or replace the utilized peer-to-peer protocol to adapt to a wide range of requirements. Through extensive evaluation we show that the proposed system decreases the delivery delay by a factor of eight compared to the existing communication pattern, leading to significantly increased information accuracy.**

## I. INTRODUCTION

The number of mobile devices, such as smartphones and tablets, rapidly increases. They are equipped with increasingly powerful computational capabilities as well as plenty of sensors, enabling a new range of applications that take the *context* of a device and its user into account. Location-based services constitute one popular example, where the physical location of the user is used as input to find answers to questions like *Where is the best Italian restaurant nearby?*. Recently, more interactive smartphone applications involving context information of more than one user are becoming increasingly popular. One example is the augmented reality game Ingress[1] by Google. Here, users interact with each other in the real world while being part of a game at the same time. Contrary to the local interaction pattern in both the real world and the game, the underlying communication pattern does not reflect the sketched locality of interaction. In fact, every event that is generated by a player and has to be disseminated to nearby players is first transfered to the backend server of the game via a cellular connection. From there, it is then sent to all parties that are located within the given area of interest, again using the cellular network.

Latency measurements for cloud-based mobile games [9] have shown that the communication over a cellular network heavily influences latency and introduces a significant delay. This is especially problematic as most interactive games require action-to-reaction latencies of below 300 ms or less according to [2]. The lower bound for the action-to-reaction delay is then essentially limited by four times the latency between mobile users and the cloud-based game service. As the interaction pattern in an augmented reality game is bound to a given region, it seems natural and obvious to reduce the lower bound through direct ad hoc delivery of game events to nearby players. However, there are many challenges that arise from a pure ad doc dissemination of events. Existing publish/-subscribe solutions for mobile ad hoc networks (MANETs) are targeted towards general-purpose event dissemination [4], [6], [10]. Locality of interaction is not taken into account and the protocols in general suffer from intermittent connectivity and node mobility. Within the scenario of mobile augmented reality games one cannot always assume a connected mobile network, as the density of players and their movement varies significantly. Furthermore, the publishing rate of game events is by far higher than the assumed rate for most proposed MANET publish/subscribe systems. At the same time, a pure ad hoc solution cannot maintain a global state, which is required for nearly all types of interactive multiplayer games.

To overcome the aforementioned shortcomings that arise from the location-agnostic communication infrastructure and the inherent characteristics of MANETs, we propose a hybrid system, where the cloud-based game service and the cellular network is augmented with ad hoc peer-to-peer dissemination. The local interaction pattern of mobile augmented reality games is exploited for the local delivery of the events in a peer-to-peer fashion. At the same time, events can still be delivered to the cloud-based game service to ensure correct updates of the global state. To further cope with changing conditions for the mobile peer-to-peer dissemination, the proposed system adapts the utilized local communication protocol based on contextual information that is already included in the game events.

The following contributions are presented in this paper:

1) We propose a location-aware event dissemination system for mobile augmented reality games that exploits game-specific information to automatically enable and configure local peer-to-peer dissemination of events when appropriate.
2) To handle the inherent dynamics of the communication system and particularly of MANETs, the system adapts to a wide range of requirements by reconfiguring or replacing the currently active peer-to-peer dissemination protocol on the mobile devices.
3) We conduct an extensive simulation study of the proposed system comparing the results to (i)

[1] http://www.ingress.com

978-1-4799-6201-3/14/$31.00 

infrastructure-only communication used in today's deployments and (ii) a non-adaptive hybrid scheme. The proposed system decreases the delivery delay to below $150\,\text{ms}$ for a broad range of scenarios that are relevant for mobile augmented reality games.

The remainder of this paper is structured as follows. In Section II, the scenario is detailed and challenges for an event dissemination system are derived. The resulting hybrid system for mobile peer-assisted event dissemination for augmented reality games is presented in Section III. The system is extensively evaluated and compared against current dissemination schemes in Section IV. In Section V we discuss related work, before we summarize and conclude the paper in Section VI.

## II. Scenario

An overview of the scenario is given in Figure 1. A number of players, carrying mobile devices, participate in a location-based augmented reality multiplayer game. The devices connect to a cloud-based game service via the cellular infrastructure. Each device publishes events containing updates of its current game state at a given frequency or based on interactions with the environment. Those events include player actions and reactions, as well as contextual information that is required for the gameplay, e.g., location-specific data or other sensor readings. The cloud-based game service maintains the global state of the game and ensures consistency of events and reactions.Furthermore, it forwards events to other players, if those are affected.

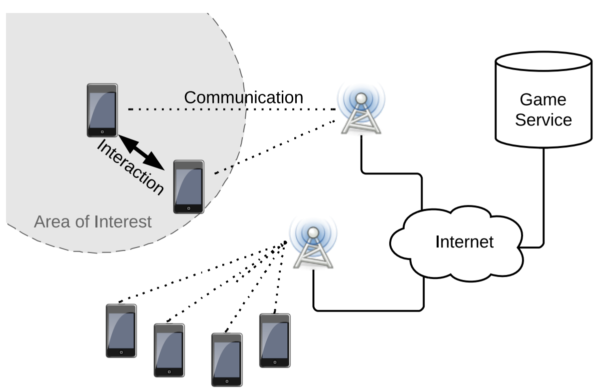



Figure 1: Mobile devices connected to the game service.

In a augmented reality multiplayer game, as in traditional virtual environments [16], players are affected by events issued by nearby players. Therefore, the augmented reality application defines an Area of Interest (AoI) around the player's physical location. All events issued by players within that AoI have to be delivered to ensure fluid and responsive gameplay. As soon as players move in the real world, their AoI has to be updated at the game service to ensure the delivery of events that are relevant at the new location. From Figure 1 it becomes apparent that this leads to global communication, although the interaction is mainly local. Furthermore, the number of other players within an AoI can vary significantly. Augmented reality games usually define points of interest, where the player can then trigger specific actions. Within Ingress, for example, historical landmarks and sights are considered points of interest in the game. Especially in urban scenarios the number of concurrently active players at such points of interest can be high compared to points of interests in rural locations.

It is important to note that the game service itself may consist of a distributed broker network for scalability reasons. In that case, players from a specific geographic region are most likely grouped at one broker. This is due to the fact that core objective of the distributed broker network is to minimize the communication overhead inside the broker network. In the following, we therefore refer to the game service as a single central entity, which is a reasonable assumption for a geographic region that could potentially benefit from direct ad hoc delivery.

## III. System Design

Core idea is the utilization of context required for augmented reality games to (i) reason about the utilization of local peer-to-peer communication and (ii) its adaptation within the communication system. An overview of the proposed system is given in Figure 2. Mobile nodes publish their game events, and in the default hybrid system, these events are then sent to the cloud-based game service via the cellular network. The game service filters the events and updates the global game state accordingly. It notifies all interested clients by sending them the events, again via the cellular network. If a mobile peer-to-peer protocol is enabled at a mobile node, it does not only send the event to the cloud but immediately spreads the information to nearby devices via the respective dissemination protocol. As long as the protocol is also enabled at the addressed nodes, the event is locally delivered at significantly lower delay when compared to the cellular network.

Besides the mandatory mobile peer-to-peer dissemination strategies, the adaptation engine constitutes another important building block of our hybrid system. The engine is located at the game service and remotely controls and configures the peer-to-peer dissemination protocols utilized at the mobile devices. All events that pass through the cloud service are monitored and local groups are detected during the filtering process. Together with game-specific data provided by the game service, the monitoring results serve as input for the adaptation engine, which in turn activates, deactivates, or reconfigures the local dissemination strategy of mobile nodes. If the monitoring results indicate significant condition changes, the local dissemination strategy can be switched to another one to achieve a better performance or reduced cost. In the following, we describe the semantics of the event dissemination system. In Section III-B we introduce the peer-to-peer dissemination protocols. Section III-C describes the adaptation engine in detail.

### A. *Subscription and Event Semantics*

Following the publish/subscribe communication paradigm, mobile nodes subscribe to their current AoI and are from there on notified upon state changes of other nodes in the given area. For simplicity, we assume a circular area, defined by the current location $l$ and a radius $r$, but other descriptions would be feasible as well. A subscription $s_j$ issued by node $j$ consists of the 3-tuple $s_j = (l_j, r, \mathrm{ID}_j)$, where $\mathrm{ID}_j$ is the unique node identifier of node $j$ that is later on relevant for

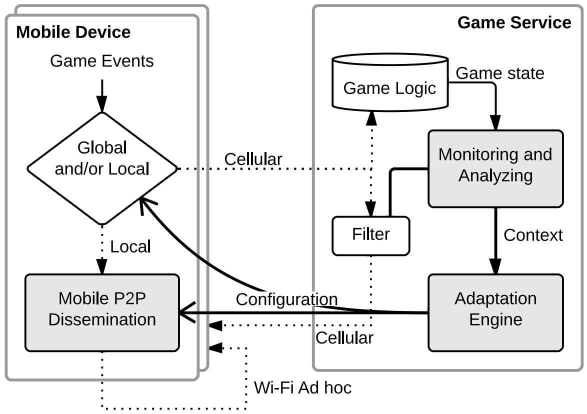



Figure 2: System design with mobile nodes and their peer-to-peer dissemination schemes on the left and the central game service including the adaptation engine on the right.

subscription updates. Each event $e_k$, issued by a client $k$ to update the game's state, contains at least the client's current position $l_k$ and its identifier $\mathrm{ID}_k$. On the cloud-based game service, matching is performed by comparing the event $e_k$ to all existing subscriptions $s \in S$. If $s$ matches the event $e_k$, meaning that the distance between $l_j$ and $l_k$ is smaller than $r$, the event is forwarded to the respective subscriber.

In traditional publish/subscribe systems, nodes would have to resubscribe each time they change their current position. However, as the events issued by a node already contain contextual information (in this case, the node's location), we automatically update subscriptions with the corresponding node identifier. In the above example, if node $j$ itself issues a new event with its updated position $l_j^*$, the subscription $s_j$ is updated to $(l_j^*, r, \mathrm{ID}_j)$. This way, the resulting overhead from contextual changes in the system is reduced dramatically compared to static subscriptions. This concept is also known as context-aware publish/subscribe. We refer the interested reader to [5] for more details.

### *B. Mobile Peer-to-Peer Dissemination Protocols*

Each mobile node in the system is equipped with a set of mobile peer-to-peer dissemination protocols that make use of the device's communication capabilities. While we only consider peer-to-peer protocols over Wi-Fi ad hoc in this work, the concept can be extended to other wireless standards such as the 802.15 protocol family for Wireless Personal Area Networks or Bluetooth, if they are available on the device. Goal of a dissemination protocol is to exploit the broadcasting capabilities of the wireless medium and to send events simultaneously to all nodes within physical proximity. Thereby, the latency of event delivery compared to a delivery via the cellular network and the central game service is significantly reduced.

A dissemination protocol is controlled and configured by the adaptation engine of the cloud-based game service. Therefore, nodes do not need to actively probe for neighbors by issuing periodic beacons. Instead, as long as the respective dissemination protocol is inactive, the communication interface can stay in its idle or off state and preserve battery life.

Once activated, a peer-to-peer dissemination protocol communicates with other nodes nearby, as long as the receiving nodes have activated the respective protocol. As the adaptation is centrally coordinated by the adaptation engine, it is ensured that all nodes within a local group have activated the same strategy. This concept is sketched in Figure 3, where groups of mobile nodes communicate via mobile peer-to-peer protocols. The light gray nodes do not have other nodes within their current area of interest, which is why they do not utilize a local communication protocol at all. Instead, they only communicate with the game service via the cellular connection as illustrated in Figure 2. If nodes move within the area of existing groups, the adaptation engine at the cloud-based game service enables the corresponding local communication protocol at all concerned nodes.

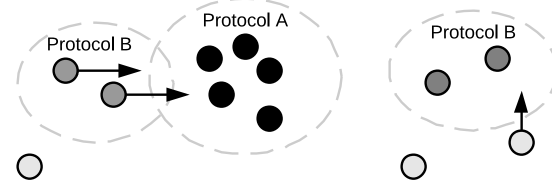



Figure 3: Local event dissemination in groups of nodes. Depending on the characteristics (here: density) different mobile peer-to-peer dissemination protocols are utilized.

Within this work, we use two mobile peer-to-peer dissemination protocols: (i) a range-limited broadcast and (ii) a probabilistic broadcast scheme. Both protocols exploit the semantics of the transmitted events to limit the dissemination range. In the range-limited broadcast protocol, nodes initially broadcast the event to all neighbors. In addition to the event, the message contains a unique identifier that is used to detect duplicates. When a neighbor receives the broadcast and does not detect a duplicate, it matches the contained event against its own local subscriptions. The message is only forwarded if a match occurred. This mechanism ensures that broadcasts are not repeated outside of the AoI. The probabilistic scheme extends this concept by additionally limiting the probability $p$ that a node forwards the message after all checks were successful. Thereby, the probability of collisions is reduced for denser scenarios. The probability $p$ is configured based on the number of other nodes $n$ within the current AoI as $p = \frac{1}{\sqrt{n}}$. Thereby, the number of forwarded messages in denser scenarios is reduced. Compared to the range-limited broadcast, this lowers the probability of collisions on the MAC layer. However, a lower value for $p$ can result in information loss, as the message might not be forwarded at all. Within the proposed system, the value for $p$ can be reconfigured by the central adaptation engine, as detailed in the following section.

Events may arrive multiple times due to the local dissemination strategy and the parallel distribution via the game service. The system can be configured to ensure *at-most-once* delivery of events if required by the game design. In that case, a unique identifier is assigned to each event upon creation and the mobile node filters duplicates based on this identifier. Thus, as long as the communication protocol between mobile nodes and the game service is reliable, events are processed *exactly once* by the application.

While duplicates can be filtered quite easily, consistency within the game has to be ensured when events can take both, the local and the global dissemination path. The logical order of actions and the strategy on how to resolve inconsistencies depend on the game itself and are not part of the communication system [7]. The number of different states that have to be kept consistent is relatively small compared to the overall number of players, as events are only relevant to a limited number of nodes within the proximity of the event's creator. Therefore, well-known solutions such as vector clocks [11] can be utilized to retain the logical order of events on the application layer. For an extensive survey of consistency protocols for distributed games we refer the reader to [16].

### C. Adaptation Engine

As the cloud-based service performs the matching operation, it also calculates the fan-out of a given event. The fan-out of an event is defined as the number of subscriptions $|S|$ with distinct node identifiers that match the given event. Within our scenario, the fan-out therefore describes the number of other nodes that currently have the originator of the event within their AoI. If that value crosses a given threshold and does so for a predefined time $\delta_t$, one can assume that a *group* of nearby players has formed. Those players could directly exchange their events, using a mobile peer-to-peer dissemination protocol. Based on the size of the group and its persistence, i.e., the frequency at which members join and leave the group, different local dissemination methods might be feasible.

To this end, the adaptation engine analyzes the status of each group and decides whether to (i) enable or disable peer-to-peer dissemination, (ii) reconfigure a running dissemination protocol, or (iii) switch to another dissemination protocol. This design enables fine-granular adaptation: if there are only minor changes in the current conditions, a reconfiguration of the active protocol might be sufficient to ensure consistent quality of service. In the probabilistic scheme utilized in this work, the value for $p$ is configured based on the current density as observed at the game service. In larger groups, the local dissemination strategy is reconfigured to use a smaller value for $p$, leading to a lower probability of events being multiplied and, thus, reducing the overall number of messages on the wireless medium. As soon as the conditions change significantly, the whole protocol can be replaced with another local dissemination protocol that is better suited. By adding new dissemination protocols to the system, it can cope with a wide range of conditions. The concept of transitions between local dissemination protocols also enables the utilization of different wireless standards. By switching, for example, from Wi-Fi ad hoc to Bluetooth delivery for groups of nearby players, one can further reduce the energy consumption as well as decrease the probability of collisions on the physical medium. Which protocol to use depends on the current environmental conditions: the number and density of nodes as well as their distance to each other. The different characteristics of the chosen wireless standard in terms of range, bandwidth, and energy consumption motivates the use of more than one dissemination protocol. As the adaptations are orchestrated at the cloud-based service, it is ensured that they are executed in a coordinated fashion within a group.

Nodes are notified by the centralized adaptation engine and switch to the respective dissemination protocol as soon as they receive the corresponding control message. To reduce communication overhead, the control message can easily be piggybacked on top of any notification message sent to the respective node. The control message carries at least the identifier of the chosen local dissemination protocol. Additionally, it can be equipped with bootstrap information for the respective protocol, such as a list of initial contacts or an initial parameter configuration. Last but not least, the message may carry a configuration object for the global dissemination - this way, the frequency of event updates or the ratio of those updates that are transmitted to the cloud can be controlled. This becomes especially interesting if the adaptation engine relies on application-specific data, which are available on the cloud service. The decision process takes the game-specific information, such as in-game groups or different quality of service classes for event types, into account.

As the adaptation engine runs only on the game service and not on the local devices, the devices are not affected by its complexity. Furthermore, the approach does not require mobile nodes to actively probe for neighbors, as this information is deduced from the context information that is nonetheless being sent to the game service. If no local dissemination scheme is activated, the mobile device just behaves like in the initial scenario without any additional data being sent or computations being executed. The central coordination becomes especially important, if the physical layer protocol is to be switched as well. As nodes are no longer able to communicate when using different physical communication channels, the switching between those channels has to be coordinated to ensure seamless operation of the dissemination system. Furthermore, the centralized controller can enforce policies based on the game state: one example would be to only forward events to nearby players that are within the same team by configuring the respective dissemination strategy accordingly. This concept of centrally controlled local distribution also seems to make its way into the Android mobile platform with Google Nearby[2].

In this work, the adaptation engine utilizes a simple threshold-based strategy to configure the local dissemination protocols. The decision is based solely on the number of nodes within an AoI, an information that is readily available on the central game service due to the nature of the publish/subscribe system. The adaptation engine is configured with two different thresholds $\Delta_{\mathrm{bc}}$ and $\Delta_{\mathrm{p}}$, where $\Delta_{\mathrm{bc}}$ denotes the minimal number of nodes within an AoI that enables the local event dissemination using the range-limited broadcast protocol. Once the number of nodes reaches the second threshold $\Delta_{\mathrm{p}}$, the protocol is switched to the probabilistic scheme. Nodes that are moving at the verge of another node's AoI could lead to oscillations if they frequently join and leave the AoI. Finding matching subscribers for an event already includes calculating the distances between nodes. Therefore, the strategy can be further extended by taking the node distances into account without introducing additional computational cost. By neglecting nodes that are located in close proximity to the AoI bounds, oscillations are avoided.

[2]http://www.androidpolice.com/2014/06/06/exclusive-google-will-soon-introduce-nearby-to-let-other-people-places-and-things-know-when-youre-around/

## IV. Evaluation

The primary objective of the evaluation is a comparison between our proposed adaptive hybrid event dissemination system and (i) a pure cloud-based solution and (ii) the hybrid solution but without adaptivity. In the pure cloud-based solution, mobile devices connect to the cloud-based service and send all game updates via the cellular connection. In the non-adaptive hybrid solution, mobile devices additionally distribute their game events locally to reduce the latency for nearby players. The systems are compared in scenarios with varying node density and movement speed. Furthermore, the size of the area of interest as well as the rate of events and their size is varied to mimic different application scenarios.

### A. Evaluation Setup

The evaluation is conducted by means of simulations using the peer-to-peer overlay simulator PeerfactSim.KOM [15] on top of the NS-3 802.11g underlay model [12]. Experiments simulate 30 minutes of operation, with the first 10 minutes being neglected in the evaluation results to allow the system to reach a steady state. All experiments are repeated five times with different seeds for the random generator. Bar charts show the average over those runs as well as the corresponding 95% confidence interval unless otherwise stated. For most metrics, the corresponding box plots are shown as well, providing a better understanding of the distribution of the underlying data. The size of the box denotes the upper and lower quartile of the values, and the median is shown as a solid line within the box. Whiskers show the lowest (highest) data point within $1.5\times$ the interquartile range of the lower (higher) quartile. Outliers are shown as individual crosses.

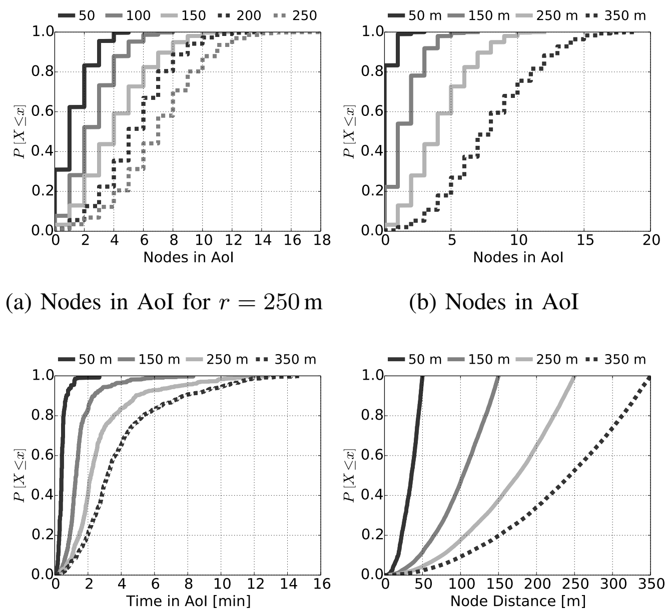
Table I summarizes the settings of the simulation setup. We vary the node density by keeping the size of the simulated area constant while changing the number of mobile nodes. Communication with the cloud-based service uses the cellular network, which is assumed to be reliable in our simulations. Latency to the cloud server is derived from a measurement study by Lampe et al. [9]. The cloud server is equipped with sufficient bandwidth to handle all incoming requests.

Table I: Simulation setup and scenario parameters

| | |
|---|---|
| Simulated Area | $2800\,\mathrm{m} \times 2800\,\mathrm{m}$ |
| Cellular Network | Reliable, $200\,\mathrm{ms}$ latency $\pm 100\,\mathrm{ms}$ [9] |
| Ad Hoc Network | NS-3 802.11g model [12] |
| Movement Model | Random Waypoint, no pause |
| Mobile Nodes | $50, 100, \underline{150}, 200, 250$ |
| Movement Speed | $1, \underline{2}, 3, 4\,\mathrm{m/s}$ with $\pm 0.5\,\mathrm{m/s}$ |
| Event Rate | $\underline{4}, 6, 8$ events/s |
| Event Payload | $32, \underline{128}, 512$ Byte |
| Area of Interest | Circular, $r = 50, 150, \underline{250}, 350\,\mathrm{m}$ |

Each mobile node issues updates of its current state at the given event rate and with a given payload size. The event rate is derived from the latency requirements for games as described in the introduction, with $4$ and $8$ events/s leading to a theoretical latency of $250\,\mathrm{ms}$ and $125\,\mathrm{ms}$, respectively. The payload of an event represents additional data that are sent with each event and are only relevant for the game but not for the communication mechanism (i.e., data that is not modeled as an attribute in the publish/subscribe system). We consider a default payload of $128$ Byte with each event, leading to a data rate of $\approx 4\,\mathrm{kbit/s}$ for the default event rate.

Nodes move according to the random waypoint model without pause times. For our scenario, random movement constitutes a worst-case situation, as there is little chance of longer-lived groups. In reality people participating in an augmented reality mobile multiplayer game would move in small to mid-sized groups in a nomadic fashion and be attracted by specific locations, as analyzed in [13]. In such a case, the benefits of our proposed event dissemination system are expected to be even higher.




(a) Nodes in AoI for $r = 250\,\mathrm{m}$ (b) Nodes in AoI

(c) Duration of stay (d) Distance distribution

Figure 4: Characteristics of the simulated scenario under varying number of nodes (4a) and AoI radii (4b to 4d)

Figure 4 shows the resulting characteristics of the scenarios for varying AoI radii and varying node density. The number of nodes within an AoI as well as their duration of stay within the AoI determine the workload of the event dissemination system, as events have to be disseminated to all nodes within the AoI. Both conditions vary significantly in real-world scenarios, as described in Section II. The variation of both, AoI radius and number of nodes, lead to the desired workload characteristics. For a fixed AoI with $r = 250\,\mathrm{m}$, the maximum number of nodes inside the AoI increases significantly with the total number of nodes (Fig. 4a). This is due to the fact that nodes tend to move through the center of the simulated area when following the random waypoint model. This leads to increased node densities when compared to edge regions [8]. We argue that this effect mimics the behavior of players in a mobile augmented reality game albeit the simple movement model, as players tend to gather around attraction points as described in Section II.

For a fixed number of nodes and varying radius of the AoI, the same effect can be observed (Fig. 4b). Figure 4c shows the corresponding duration of stay for nodes within other nodes'

AoIs. The average duration of stay ranges from $\approx$ 30 s for $r = 50$ m to $\approx$ 3 min for $r = 350$ m at an average movement speed of 2 m/s. Last but not least, the distribution of node distances within the AoI is presented in Figure 4d, showing that for an AoI radius of $r = 350$ m, more than 50 % of the nodes are located outside of the broadcast-radius[3] of a single node. To deliver events to those nodes, the mobile peer-to-peer dissemination protocol has to forward events over multiple hops.

### B. Metrics

In order to quantify and compare the performance and cost of the different systems in the presented scenarios, the following metrics are measured.

*1) Delivery Ratio:* The ratio of events that are successfully delivered to interested subscribers. For all systems that rely on the central cloud service, this ratio should be equal to one, as the cellular communication channel is assumed to be reliable.

*2) Dissemination Delay:* The time it takes to successfully deliver an event to a receiver. The dissemination delay is determined by the latency of the communication medium, as we do not take local processing time into account.

*3) Staleness:* The staleness is defined as the time that elapsed since the last update of a given node was received, including the delay introduced by the transmission. It thereby combines the dissemination delay and the event rate of the given scenario into a single metric. The state at a node is at most as old as the staleness observed for that node.

*4) Detection Delay:* The detection delay denotes the time until a node $j$ that just entered the AoI of node $k$ is detected by $k$. It is, thus, determined by the accuracy of $j$'s subscription as well as the dissemination delay for node $k$'s events.

*5) Local Drop Ratio:* The ratio of events that were sent using a local dissemination strategy but did not arrive at the intended receivers due to collisions or disconnected MANET topologies.

*6) Traffic:* We distinguish between traffic that goes over the cellular communication link and traffic that is observed on the local ad hoc networking interface of mobile devices. The traffic is compared to the event rate and the corresponding event payload to determine the overhead of the communication system.

### C. Impact of the Area of Interest

Within this part of the evaluation, the basic cloud-based communication scheme as utilized in today's systems is compared to a static hybrid solution and an adaptive hybrid solution for different sizes of the area of interest. In the static hybrid solution, each event is sent to the cloud and also disseminated locally via ranged flooding. In the adaptive hybrid solution, a simple version of the adaptation engine is used to turn off local dissemination in cases where the AoI of a node is empty. Furthermore, the adaptation engine switches from ranged flooding to probabilistic broadcasting for groups with more than five mobile nodes.

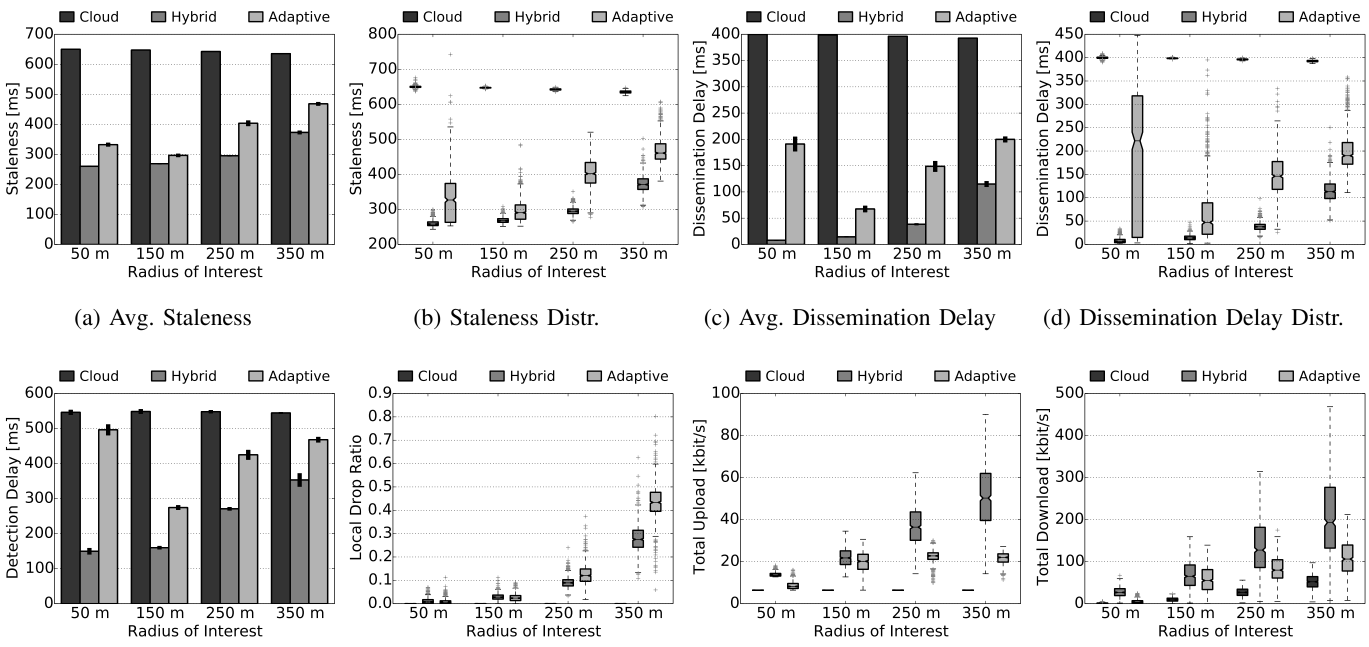
Goal of this part of the evaluation is to understand the upper and lower bounds of cost and performance of the proposed hybrid system. The pure cloud-based system thereby is considered the lower bound in terms of performance, while a static hybrid solution is the upper bound for cost in terms of traffic. All systems achieve a coverage of one, as they deliver events reliably via the cellular network in parallel to any local distribution.

Figures 5a and 5b show the staleness of information for the different systems. The average staleness for the pure cloud-based approach is 650 ms, as expected with an average transmission delay of 400 ms (Fig. 5c) and 250 ms delay between updates. Both, the hybrid and the adaptive approach reduce the average staleness significantly by lowering the transmission delay due to local dissemination (Fig. 5c). The probability of collisions and, thus, packet drops increases for larger radii. Therefore, an increasing fraction of the events is obtained via the cellular infrastructure, leading to an increase in the observed staleness with increasing size of the AoI.

The detection delay for nodes entering an AoI remains nearly constant at around 550 ms for the cloud-based scheme (Fig. 5e). With the hybrid scheme, most nodes are detected after 150 ms for a small AoI, whereas the detection delay approaches the upper bound set by the cloud-based scheme for larger AoIs. The adaptive strategy exhibits a slightly worse detection delay, as the local dissemination needs to be activated when a node enters a group. For $r = 50$, local dissemination strategies are disabled most of the time as no other node is within the current AoI, leading to high initial delays as the first event has to travel via the cloud-based game service.

With increasing AoI, the fan-out of events increases significantly. This leads to message loss on the ad hoc communication medium, as more and more collisions occur. Figure 5f shows the ratio of events that were lost on the ad hoc communication channel. In the case of the hybrid scheme, this is solely due to collisions or disconnected network parts. For the adaptive scheme, this ratio is slightly increased by the impact of the protocol switches. If a node is currently utilizing a different local dissemination protocol than the sending node, the event cannot be received. Figure 6 shows the local distribution characteristics for the adaptive scheme. The CDF in Figure 6a shows the probabilities that a given local dissemination strategy is used depending on the different AoI sizes. For $r = 50$ m only 32 % of the generated events are transmitted via a local dissemination strategy. For larger AoIs and, thus, more nodes within a group, nearly all nodes utilize a local dissemination scheme. For larger group sizes as caused by $r = 250$ m and $r = 350$ m, the adaptive strategy switches to the probabilistic dissemination strategy. For $r = 250$ m, nearly half of the nodes utilize the range-based broadcasting protocol, while the other half uses the probabilistic scheme. This results in a higher ratio of events that are lost due to different protocols at sender and receiver, as shown in Figure 6b. The x-axis denotes the ratio of events that are dropped due to mismatching local protocols out of the total number of dropped events as shown in Figure 5f. More sophisticated strategies at the adaptation engine could lead to a reduction of this ratio, as briefly sketched in Section VI, thereby decreasing the overall loss rate of the wireless dissemination strategies compared to the non-adaptive scheme.

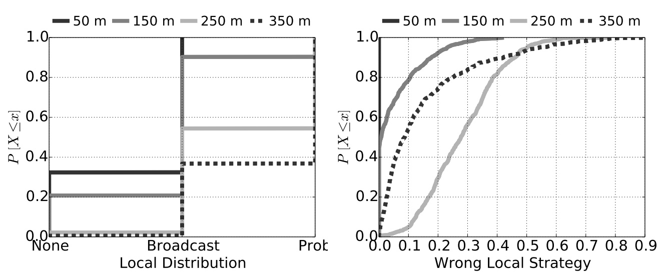
[3]The maximum distance of a broadcast transmission resulting from the default 802.11g model of NS-3 is $\approx$ 230 m.




(a) Avg. Staleness (b) Staleness Distr. (c) Avg. Dissemination Delay (d) Dissemination Delay Distr.

(e) Detection Delay (f) Local Drop Ratio (g) Upload Traffic per Node (h) Download Traffic per Node

Figure 5: Observed staleness and dissemination delay for the proposed hybrid and adaptive solutions compared to the current cloud-based scheme. Both are decreasing significantly for the hybrid schemes. The traffic per node increases compared to the cloud-based solution, with the adaptive scheme reducing this effect.




(a) Usage of local protocols (b) Event drops

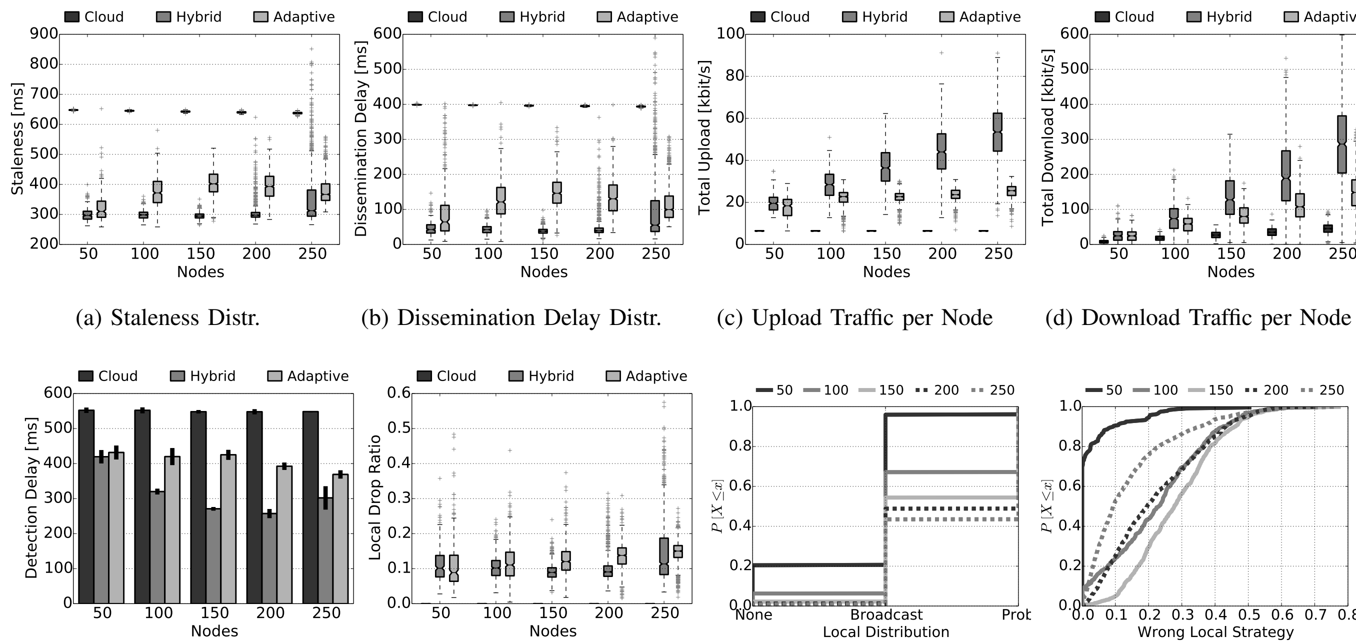
Figure 6: The adaptive system switches between local dissemination protocols. This leads to event drops due to mismatching protocols especially with $r = 250\,\text{m}$.

Comparing the observed cellular upload rate of $\approx 6.4$ kbit/s (shown for the cloud-based system in Figure 5g) to the raw event payload rate of $4\frac{1}{s} \times 128\,\text{bit} \times 8 \approx 4$ kbit/s, the system introduces an overhead of $2.4$ kbit/s. Events carry the current location of a player as well as a unique identifier as attributes in addition to the application payload. These attributes are not included into the calculation of the raw event payload rate, as their representation depends on the implementation of the publish/subscribe mechanism and may therefore vary. However, as this information is used by the game service, it further reduces the actual overhead introduced by the system.

The increase in performance for the hybrid systems obviously comes at the cost of increased traffic due to local data transmission. The static hybrid system utilizes the local dissemination strategy all the time and is therefore considered the upper bound for traffic as shown in Figures 5g and 5h. As the peer-to-peer dissemination protocol exploits the event semantics to limit message forwarding, the local traffic increases with the size of the AoI. The adaptive strategy, in turn, reduces the local traffic by switching the dissemination protocol off in cases when the AoI of a node is empty. Furthermore, by switching to the probabilistic protocol, the traffic for larger AoIs is reduced as well compared to the static hybrid system.

### *D. Impact of the Density of Nodes*

In this section, the performance of the system under varying node densities is evaluated. In the cloud-based communication system, an increased density only affects the download traffic via the cellular network, while the upload remains constant. This is due to the fact that more nodes are located within a node's AoI, as detailed in Section IV-A. Therefore, the performance and cost metrics of the cloud-based system shown in Figure 7 closely resemble those shown in the previous section. However, changing the node density has a significant impact on the mobile peer-to-peer dissemination protocols. The node density directly determines the number of messages that are sent and received by the dissemination protocol (Figures 7c and 7d). The probability for collisions on the wireless medium increases with the number of broadcasts, leading to increasing staleness and dissemination delay for the hybrid system (Figures 7a and 7b). The adaptive system reduces this effect as it switches to the probabilistic dissemination protocol if




(a) Staleness Distr. (b) Dissemination Delay Distr. (c) Upload Traffic per Node (d) Download Traffic per Node

(e) Detection Delay (f) Local Drop Ratio (g) Usage of local protocols (h) Event drops

Figure 7: For varying densities on a fixed world size (2800 m $\times$ 2800 m), the probability of collisions for the local event dissemination increases. This, in turn, has an impact on the observed staleness and dissemination delay. The detection delay decreases slightly, as the network becomes more connected. Traffic increases significantly, especially for the hybrid scheme.

the density increases. This results in decreased local traffic compared to the static hybrid system. Compared to the cloud-based system, both hybrid systems perform better in terms of staleness and dissemination delay, even in denser networks.

The detection delay (Figure 7e) decreases slightly for both hybrid systems. This is due to the fact that the probability of a connected network using the same dissemination strategy increases, as shown in Figure 7g. The number of message drops due to mismatching local dissemination protocols decreases (Figure 7h), while at the same time the probability of collisions on the wireless medium increases. Therefore, the overall drop ratio (Figure 7f) remains stable for the adaptive scheme. For the hybrid scheme, the collisions lead to a significantly higher skew in the local drop distribution for denser scenarios.

From the evaluation of different node densities and different AoI sizes we conclude that the hybrid scheme achieves a low dissemination delay and low information staleness compared to the cloud-based approach. Furthermore, by enabling adaptive control and configuration of the utilized mobile peer-to-peer dissemination protocols, the overall system overhead can be reduced significantly compared to the non-adaptive system. In scenarios with high node density, the transition to another peer-to-peer dissemination protocol enables the system to better utilize the available resources of the wireless medium. The adaptation engine disables the local dissemination protocol if there is no other node within reach, thereby reducing the traffic and preserving battery life. However, with the simple threshold-based adaptation engine, events being dropped due to protocol mismatches can make up for up to 40% of the local message drop. This counteracts the benefits obtained by switching to a dissemination protocol that better utilizes the wireless medium in denser scenarios.

## E. Impact of the Adaptation Engine Thresholds

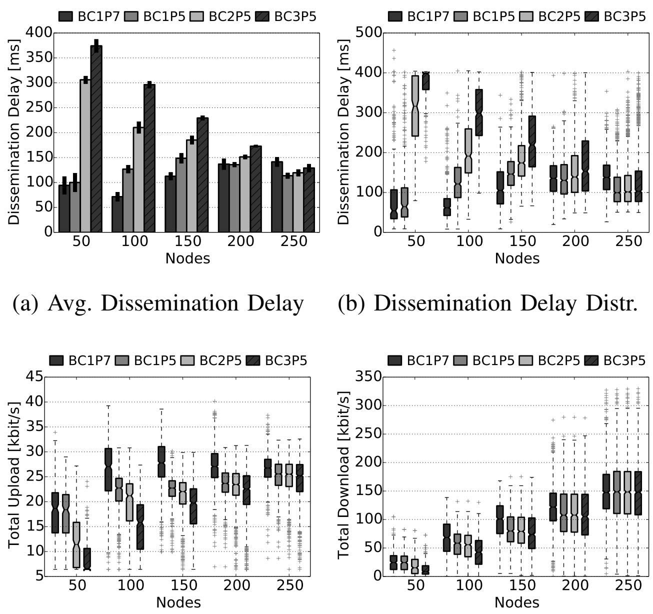
As detailed in Section III, the adaptation engine relies on two thresholds, $\Delta_{\mathrm{bc}}$ and $\Delta_{\mathrm{p}}$. In the following, the impact of those thresholds on the system performance and cost is evaluated for varying node densities. The thresholds are varied according to Table II to assess the impact of (i) switching from one strategy to the other and (ii) activating local dissemination for larger groups only.

Table II: Threshold variations

| Configuration | $\Delta_{\mathrm{bc}}$ | $\Delta_{\mathrm{p}}$ |
|---|---|---|
| BC1P7 | 1 | 7 |
| BC1P5 | 1 | 5 |
| BC2P5 | 2 | 5 |
| BC3P5 | 3 | 5 |

When activating the local dissemination protocols only for larger groups ($\Delta_{\mathrm{bc}} = \{2, 3\}$), the average dissemination delay increases for scenarios with lower node density (Figures 8a and 8b). Compared to a direct activation of local event delivery (BC1P5), especially the upload traffic (Figure 8c) is reduced. However, the thee to four times increase in the dissemination delay for lower node densities justifies the default value $\Delta_{\mathrm{bc}} = 1$. This ensures that ad hoc delivery of events is utilized as soon as two nodes are located within proximity.

When to switch between the dissemination protocols is determined by $\Delta_{\mathrm{p}}$. For scenarios with lower density, the range-based broadcast is more robust, leading to slightly lower dissemination delays for $\Delta_p = 7$ with lower node densities. However, the system utilizes the range-based broadcast protocol for denser networks, leading to an overall increase in traffic consumption.




(a) Avg. Dissemination Delay (b) Dissemination Delay Distr.

(c) Upload Traffic per Node (d) Download Traffic per Node

Figure 8: Impact of the Adaptation Engine Thresholds on the system performance and the resulting overhead.

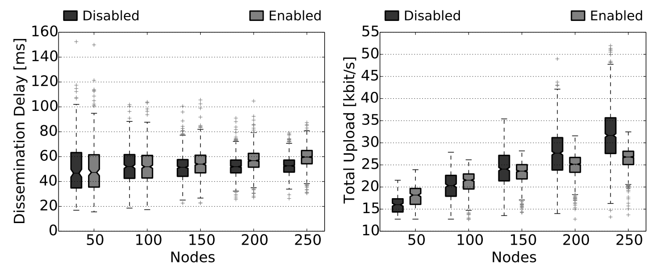
### F. *Impact of Parameter Reconfigurations*

The adaptation engine support the reconfiguration of local dissemination strategies. Figure 9 shows the comparison between the hybrid probabilistic scheme without reconfigurations and the scheme with reconfigurations enabled. As detailed in Section III, the probabilistic scheme can be reconfigured by altering the value for the parameter $p$. Based on the current node density as observed by the adaptation engine, $p$ is decreased for denser scenarios to lower the probability of collisions. As a result, the upload traffic saturates at around 25 kbit/s compared to up to 50 kbit/s in case of disabled reconfigurations. The impact on the observed dissemination delay (Figure 9a) is negligible, as it stays well below 100 ms for both configurations.

While the current dissemination protocols are rather simple, parameter reconfigurations are especially important for more complex protocols. Here, one can provide state information such as initial routing tables based on the global knowledge available at the adaptation engine.

## V. RELATED WORK

Characteristics of networked virtual environments (NVEs) can be transfered to mobile augmented reality games. However, the limitations of the cellular infrastructure as well as the inherently unreliable communication in MANETs limit the usability of existing distributed communication systems for NVEs in the context of mobile applications. For a general view on distributed multiplayer gaming and the implications on the communication architecture, albeit with a focus on fixed networks, we refer to [16]. The authors further state that providing location-aware, low-latency communication for interactive mobile games is still an unresolved issue.




(a) Dissemination Delay Distr. (b) Upload Traffic per Host

Figure 9: Reconfigurations of the local distribution protocol.

The remainder of the related work focuses on (i) mobile peer-to-peer publish/subscribe systems and (ii) adaptive middleware concepts for event dissemination. In a MANET, dynamically changing conditions such as the movement speed and density of nodes have significant impact on the performance of a pub/sub system. A simple approach to deal with node mobility is to avoid the establishment of a topology. Topology-less approaches are either purely flooding-based or utilize gossiping as a way to reduce the amount of broadcasts, as presented by Paridel et al. [10]. They are inherently robust to node movement and provide reasonable delivery delays, but do not scale well with the number of nodes or an increasing workload. Self-adaptive broadcasting or gossiping schemes aim at overcoming these limitations by adapting the forwarding algorithm to the currently observed node density [4], [6]. On the one hand, incorporating a self-adaptive gossiping scheme into our proposed architecture can potentially reduce the overhead caused by the centralized reconfiguration. On the other hand, the wait and count mechanism utilized in those protocols increases the message latency to gain information about the network state that are already available at the central adaptation engine (e.g., node densities). Comparing the performance of centrally reconfigured mechanisms against self-adaptive mechanisms remains an interesting future work, especially as the event rates for our scenario are significantly higher than those considered in [4], [6].

To increase scalability, Yoo et al. [17] propose a hierarchical system where nearby nodes are grouped into clusters. Nodes within a cluster maintain a tree structure formed by a selection of the cluster's nodes and communicate with other clusters via a gateway protocol, similar to BGP routing in the Internet. The performance is highly dependent on the density of nodes and its distribution. To this end, Friedman et al. developed a density-based pub/sub system [3], where the nodes with the densest neighborhood act as brokers. Routing is based on a gradient walk executed on a virtual topography that results from each node's current number of two-hop neighbors. While the approach scales well in dense networks, the delivery delay of a publication increases significantly with the area of the network. Furthermore, the virtual topography changes frequently with increasing movement speed, leading to high overhead due to broker handoffs.

From the discussion it becomes apparent that each system is tailored towards a specific optimization goal, and is suitable for a specific set of conditions. To adapt a system if conditions change, we propose to control and configure the utilized mobile peer-to-peer protocol from the central game service. Our evaluation results indicate that a simple dissemination scheme can already significantly improve the overall performance. As content in the given scenario is only relevant at nearby nodes and not for the whole MANET, it remains questionable whether more sophisticated protocols are required at all. However, in the proposed system, one could easily add more sophisticated protocols and switch to those, once the conditions justify that.

Sivaharan et al. present the configurable pub/sub middleware Green [14], which enables the exchange of components to adapt the system to different network environments or application requirements. However, configuration of the utilized protocols is static and there is no adaptation or reconfiguration during runtime. The conditions in terms of node density, movement, and workload of mobile augmented reality games are highly dynamic, rendering a static configuration inappropriate. Our evaluation shows the benefits of fine-grained group-based configuration of local dissemination strategies, especially if the node density becomes very low or increases suddenly.

## VI. Conclusion

In this paper, a novel hybrid event dissemination system for mobile multiplayer augmented reality games is introduced. The system exploits the locality in the interaction pattern of players and their physical proximity to augment the cloud-based game with local peer-to-peer dissemination of events. Thereby, the dissemination delay of events can be reduced significantly when compared to the pure cloud-based system. At the same time, the system exploits the contextual information that is contained in events passing through the central game service to control and configure the peer-to-peer dissemination protocols on the mobile nodes.

The presented system is currently being integrated into an augmented reality gaming prototype that features direct player interaction. We plan to compare our simulative results presented in this paper to a real deployment to gain important insights into real-world savings that are hard to assess in simulations. This includes, for example, potential energy savings as well as the performance of other communication technologies such as Bluetooth and Wi-Fi Direct. Furthermore, the impact of application-specific knowledge on the performance of the adaptation engine is to be assessed. The prototype features in-game groups of players, which can be a viable indicator for the longevity of a local group.

Decisions made by the adaptation engine are currently solely rule-based. Here, a self-learning approach is a promising direction for future research, as events form a feedback-loop between mobile devices and the game service. This loop can be utilized to monitor the performance of the mobile peer-to-peer dissemination scheme itself and to adapt the respective configuration accordingly. As shown during the evaluation, merging of groups utilizing different dissemination protocols can be further improved. To this end, the adaptation engine could employ clustering algorithms to detect groups [1] that are not limited to the AoI of a node. While this reduces additional overhead at the central node, it can reduce the number of dropped events due to locally mismatching dissemination protocols. Studying the performance vs. cost trade-off of more complex strategies at the adaptation engine is part of our future work.

## Acknowledgement

This work has been funded by the German Research Foundation (DFG) in the Collaborative Research Center (SFB) 1053 "MAKI – Multi-Mechanisms-Adaptation for the Future Internet".

## References

[1] A. A. Abbasi and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks," *Computer communications*, vol. 30, no. 14, pp. 2826–2841, 2007.

[2] M. Dick, O. Wellnitz, and L. Wolf, "Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games," in *Proc. SIGCOMM NetGames*. ACM, 2005, pp. 1–7.

[3] R. Friedman and A. Kaplun Shulman, "A Density-driven Publish Subscribe Service for Mobile Ad-hoc Networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 522–540, 2013.

[4] A. Holzer, F. Vessaz, S. Pierre, and B. Garbinato, "PLAN-B: Proximity-Based Lightweight Adaptive Network Broadcasting," in *Proc. Network Computing and Applications (NCA)*. IEEE, 2011, pp. 265–270.

[5] K. Jayaram, P. Eugster, and C. Jayalath, "Parametric content-based publish/subscribe," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 2, p. 4, 2013.

[6] A. Khelil, P. J. Marrón, C. Becker, and K. Rothermel, "Hypergossiping: A Generalized Broadcast Strategy for Mobile Ad Hoc Networks," *Ad Hoc Networks*, vol. 5, no. 5, pp. 531–546, 2007.

[7] J. Kienzle, C. Verbrugge, B. Kemme, A. Denault, and M. Hawker, "Mammoth: a Massively Multiplayer Game Research Framework," in *Proc. Foundations of Digital Games (FDG)*. ACM, 2009, pp. 308–315.

[8] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET Simulation Studies: the Incredibles," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 50–61, 2005.

[9] U. Lampe, R. Hans, and R. Steinmetz, "Will Mobile Cloud Gaming Work? Findings on Latency, Energy, and Cost," in *Proc. Mobile Services (MS)*. IEEE, 2013, pp. 103–104.

[10] K. Paridel, Y. Vanrompay, and Y. Berbers, "Fadip: Lightweight Publish/Subscribe for Mobile ad hoc Networks," in *Proc. OTM*, 2010.

[11] M. Raynal and M. Singhal, "Logical time: Capturing causality in distributed systems," *Computer*, vol. 29, no. 2, pp. 49–56, 1996.

[12] G. Riley and T. Henderson, "The NS-3 Network Simulator," in *Modeling and Tools for Network Simulation*. Springer, 2010.

[13] S. Shen and A. Iosup, "Modeling Avatar Mobility of Networked Virtual Environments," in *Proc. Massively Multiuser Virtual Environments (MMVE)*. ACM, 2014, pp. 1–6.

[14] T. Sivaharan and G. Blair, "GREEN: A Configurable and Reconfigurable Publish-Subscribe Middleware for Pervasive Computing," in *Proc. On the Move to Meaningful Internet Systems (OTM)*, 2005.

[15] D. Stingl, C. Gross, J. Ruckert *et al.*, "PeerfactSim. KOM: A Simulation Framework for Peer-to-Peer Systems," in *Proc. High Performance Computing and Simulation (HPCS)*. IEEE, 2011.

[16] A. Yahyavi and B. Kemme, "Peer-to-Peer Architectures for Massively Multiplayer Online Games: A Survey," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, p. 9, 2013.

[17] S. Yoo, J. Son, and M. Kim, "A Scalable Pub/Sub System for Large Mobile ad hoc Networks," *Journal of Systems and Software*, 2009.