Geocast into the Past: Towards a Privacy-Preserving Spatiotemporal Multicast for Cellular Networks

Sander Wozniak, Michael Rossberg, Franz Girlich, Guenter Schaefer
Ilmenau University of Technology
{sander.wozniak, michael.rossberg, franz.girlich, guenter.schaefer}[at]tu-ilmenau.de

***Abstract*—This article introduces the novel concept of Spatiotemporal Multicast (STM), which is the issue of sending a message to mobile devices that have been residing at a specific area during a certain time span in the past. A wide variety of applications can be envisioned for this concept, including crime investigation, disease control, and social applications. An important aspect of these applications is the need to protect the privacy of its users. In this article, we present an extensive overview of applications and objectives to be fulfilled by an STM service. Furthermore, we propose a first Cluster-based Spatiotemporal Multicast (CSTM) approach and provide a detailed discussion of its privacy features. Finally, we evaluate the performance of our scheme in a large-scale simulation setup.**

I. Introduction

In this work, we introduce the novel concept of Spatiotemporal Multicast (STM), i.e., the issue of sending a message to all cell phones that have been residing at a specific area during a certain time period in the past (see Fig. 1). This concept can be considered as an extension of Geographic Multicast (geocast) [1], introducing the temporal aspect of sending a geocast message "into a certain point in the past".

We envision a wide variety of applications for the sketched service. For example, it may be used to discover witnesses of a crime by sending a message to mobile phones that have been near the crime scene at the time the crime has presumably been committed. In addition, an STM service could help in controlling the outbreak of an infectious or contagious disease, as carriers of a pathogen are often unaware of their contamination. With incubation times delaying the appearance of first symptoms several hours, days, or weeks, it is hard to know who has been infected. While an early detection of carriers is vital to successfully treating a disease, it is usually impossible to quickly test large parts of the population for infection. Assuming the area of the outbreak or the path of one single carrier is roughly known (e.g., by evaluating the past movement of a victim), an STM service could advise other potential victims to seek medical attention. Finally, an STM service may be used for mobile social networking applications, e.g., in order to exchange information or to get in contact with people encountered in the past. These applications could range from exchanging pictures with fellow visitors after a festival to dating services enabling people to meet again after a random encounter where no contact information has been exchanged.

An important aspect of these applications is the privacy of the users receiving STM messages. For example, a witness may not want to reveal his presence to the crime scene to someone else, e.g., out of fear of retaliation by the perpetrator. A person potentially infected with a disease may want to be alerted without having revealed his identity to someone else than his doctor. Finally, for social applications, users may not want to be identified by the sender or other receivers, being free to ignore or react to the content of the message.

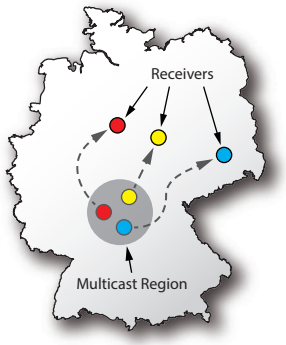

Fig. 1. Example for the concept of Spatiotemporal Multicast (STM).

A naïve approach for realizing a privacy-preserving STM is to let each mobile record its own movement. Then, a sender can deliver a message by broadcasting it to all users. A mobile only shows the received message to its user, when its recorded path intersects with the destination time and place. It is easy to see that, in this scenario, a sender has to make the information in the message available to everyone. However, a sender may desire to keep the information confidential between him and the users in the destination region. In this case, only users that have actually been present in the area at the time should receive the message. Therefore, a privacy-preserving STM approach should deliver messages only to users having resided in the spatiotemporal destination region.

There are several research areas related to the concept of STM. The issue of Geographic Multicast (geocast) has been studied extensively since the past decade [1]. Nevertheless, it does not consider the temporal aspect of sending a message to nodes that have been residing in an area at a point in the past. In contrast, the concept of Just-in-Time Multicast (mobicast) does consider the spatiotemporal aspect [2]. However, its goal is to deliver a message to a geographic area residing in the future path of a node for just-in-time delivery. Therefore, it only considers the temporal aspect of sending a message to an area at a point in the future, not to the past. Furthermore, the

issue of privacy-aware spatiotemporal query processing has received some attention [3], [4]. However, these approaches focus on database solutions and do not consider the issue of multicast message delivery. Finally, there has been research on *missed connections* services, allowing strangers who have been residing in the same place at the same time in the past to get in contact [5], [6]. While such encounter-based services are similar to the concept of STM, they do neither consider the challenge of multicast message delivery nor aim to protect the privacy of receivers from the sender of a message.

Within this article we make the following contributions:

- the concept of spatiotemporal multicast is introduced,
- extensive privacy objectives are derived,
- we present a first method to realize a privacy-preserving spatiotemporal multicast service for cellular networks,
- and evaluate it by a comprehensive simulative evaluation.

The rest of the article is organized as follows: In section II we present objectives to be fulfilled by an STM service. Section III describes our CSTM scheme, while a discussion of its privacy features is given in section IV. Finally, we present the results of the performance evaluation of our approach in section V and conclude with an outlook in section VI.

Please note that we use the term *spatiotemporal region (st-region)* to refer to a specific geographic area during a certain time period. Furthermore, we use the term *spatiotemporal datagram (st-datagram)* when referring to a message that should be delivered to users having resided in a specific *st*-region. Regarding components of the cellular network, we rely on the terminology of Long-Term Evolution (LTE).

II. Design Objectives

A. Functional Objectives

- **Long-term support:** A sender should be able to send a message to users having resided in the area a long time ago. Given the above mentioned applications, this time can range from several hours or days to even weeks.
- **Delivery ratio:** Users present at the addressed *st*-region should receive an *st*-datagram with high probability.
- **Precision:** Senders should be able to address an *st*-region up to several minutes within specific radio cells.

B. Non-functional Objectives

- **Delivery speed:** The delivery delay (the time between sending and receiving) should be low.
- **Efficiency:** The service should be efficient in terms of computation, memory, and communication overhead. The latter includes that only users having actually been in the destined *st*-region should receive the *st*-datagram.
- **Scalability:** The objectives mentioned above should not be significantly degraded by an increasing number of users and senders (i.e. number of *st*-datagrams).
- **Robustness:** The service should deliver *st*-datagrams with high probability despite failures of the infrastructure.

C. Privacy and Security Objectives

For user privacy, the following aspects must be considered:

- **Anonymity:** Attackers must not infer the identity of users receiving an *st*-datagram. Otherwise, they could unwillingly be associated with events addressed in it.
- **Location privacy:** Attackers must not infer the past, present, or future locations of users up to a defined accuracy. This is important as knowing the location of users during the night time can reveal their home addresses, for example.
- **Co-location privacy:** Attackers must not be able to decide whether two users have been residing in the same radio cell at the same time.
- **Absence privacy:** Attackers must not determine the absence of a user from an *st*-region (e.g. by automatically testing whether he receives a message addressed to this region). This knowledge can be harmful if a user was not at a specific *st*-region although he was supposed to be.
- **Data confidentiality:** Only users having resided in the destined *st*-region should be able to read and detect the availability of a message for this region. This is required by some applications, as detecting a message can already compromise the confidentiality of the information (e.g. in case of a rare event like the outbreak of a disease).

III. Approach

In this article, we propose a Cluster-based Spatiotemporal Multicast (CSTM) approach which uses *Rendezvous Points (RPs)* to deliver *st*-datagrams. These RPs act like mailboxes where senders can deposit *st*-datagrams, allowing users to retrieve them later by polling the RPs for the *st*-regions they have been residing in. Thus, the message can also be retrieved if the *User Equipment (UE)* has not been available at the time the message has been sent. It also allows RPs to bundle multiple *st*-datagrams in one packet when responding to polls.

The confidentiality of a message between a sender and its intended receivers requires a key exchange. Since a key can only be exchanged proactively for future messages, we use the base stations, referred to as *evolved NodeBs (eNBs)*, to perform a key exchange. Therefore, the base stations generate and broadcast tokens τ containing symmetric keys K at regular time intervals t. Base stations generate these keys from an initial random seed s using a Cryptographic Pseudo-Random Number Generator (CPRNG). In order to generate these seeds, a trusted entity is required. This entity, referred to as Token Planning Server (TPS), is able to recover keys using its knowledge of seeds and time slots at which keys are generated by eNBs. Generating the keys on the eNBs has the advantage that online interaction with the TPS is only necessary when sending *st*-datagrams. Furthermore, adversaries are usually not able to compromise eNBs without being noticed.

The UEs use the keys contained in the tokens to determine which RPs to poll and which *st*-region identifiers to supply for message lookup. UEs obtain this *st*-region identifier $r_K = h(K)$ by applying a cryptographic hash function $h(x)$ to

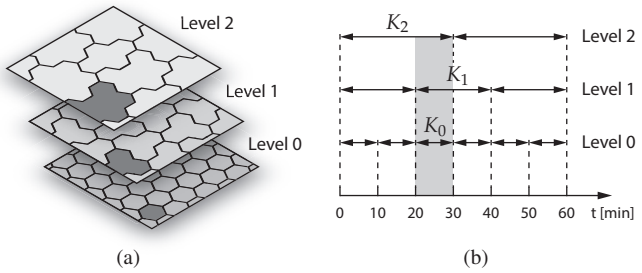

Fig. 2. Examples of token hierarchies in (a) space and (b) time. Each spatial cluster and time interval shown here uses a different key. The darker areas indicate which keys are part of a token $\tau_{c,t}$.

the key K. This identifier r_K allows RPs to perform a lookup for *st*-datagrams while preventing adversaries from obtaining K using a compromised RP. In order to determine which RPs to poll with r_K, UEs apply $h(x)$ to r_K. From this RP identifier $rp_K = h(r_K) = h(h(K))$, UEs obtain the name of the RP to be polled. UEs compute the name by appending the number $rp_K \bmod N$ to a known prefix, where N is the known number of RPs. Given this name, the IP address is resolved using the Domain Name System (DNS). Using $h(x)$ in this approach, the load is evenly distributed among the RPs while preventing attackers from inferring the *st*-regions being stored on a RP.

A potential issue of the CSTM approach is the polling overhead. When polling RPs, UEs check for messages for all keys contained in the tokens that they have received so far (using *st*-region and RP identifiers as described above). Hence, the polling overhead is directly related to the number of keys a UE has received. In order to reduce the number of poll messages between UEs and RPs, we introduce the concept of a *token hierarchy*. A token hierarchy is used to aggregate tokens over space and time (see Fig. 2), providing a trade-off between delivery accuracy and polling overhead (evaluated in section V). This is achieved by distributing tokens that consist of several different symmetric keys (one for each level l of all λ levels of the hierarchy). The token hierarchy $L = \{(l, t_l^s, t_l^v) \mid 0 \leq l < \lambda\}$ defines the time slot sizes t_l^s of level l and the continuous validity periods $t_l^v = [a_l, b_l)$ relative to the time a token has been received by a UE. Consequently, $a_0 = 0$ and $\forall l, 0 < l < \lambda : b_{l-1} = a_l$. When polling a RP, a UE uses the key based on the time that has passed since the reception of the token. Each level of the hierarchy defines the validity period t^v during which the key of this level is to be used. For example, assuming $L = \{(0, 10, [0, 30)), (1, 20, [30, 60)), (2, 30, [60, 90))\}$ (times in minutes), during the first 30 minutes after having received the token, a UE uses level 0 when polling for messages. Between 30 and 60 minutes after the reception, the key of level 1 is used and between 60 and 90 minutes, the UE uses the key of level 2. After that, the token is considered stale and deleted by the UE. Since with increasing levels, the granularity of the addressed *st*-region decreases according to Fig. 2, the number of keys to be polled by a UE should be decreased as well. However, in this case, the delivery accuracy is expected to degrade. Therefore, users not having resided in the region may receive messages not intended for them. While decreasing the delivery accuracy is not desired regarding the confidentiality between sender and receivers, it improves the privacy of users by obfuscating their movement trails according to the concept of k-anonymity.

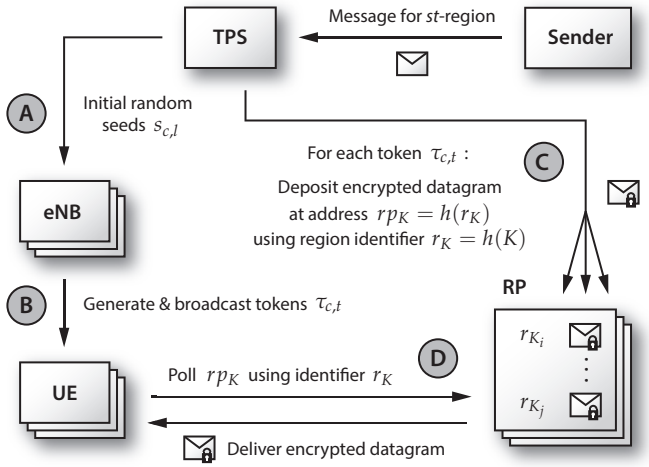

Fig. 3. Overview of the phases of the CSTM approach: (A) token planning, (B) token distribution, (C) message deposition, and (D) message delivery. Messages exchanged between entities are protected by TLS.

We now provide a detailed overview of the four phases of the CSTM approach in the following sections (see Fig. 3).

A. Initialization / Token Planning

In the first phase, the TPS uses its knowledge of the layout of the cellular network to generate initial random seeds $s_{c,l}$ for each cell $c \in C$ and level l of the token hierarchy. Here, C is the set of all radio cells. eNBs use these seeds for key generation (e.g., given 3 levels, an eNB will receive 3 initial seeds). By distributing the same $s_{c,l}$ to different cells, spatial aggregation is achieved with eNBs generating the same key at this level (see Fig. 2a). Finally, the TPS sends the initial seeds to the eNBs using a cryptographic protocol providing confidentiality, data integrity, and entity authentication, for example, using the Transport Layer Security (TLS) protocol.

B. Token Generation and Distribution

As mentioned above, establishing message confidentiality requires a proactive key exchange. Hence, tokens $\tau_{c,t}$ are distributed to all UEs in the cell $c \in C$ during each time slot t. In cell c at the beginning of time slot t, an eNB creates the token $\tau_{c,t} = \{(t_l^v, K_{c,t,l}) \mid 0 \leq l < \lambda\}$ by generating the symmetric keys $K_{c,t,l}$ using the Cryptographic Pseudo-Random Number Generator of level l (CPRNG$_l$) initialized with random seed $s_{c,l}$. How often a new key is generated for a level l depends on the time slot t_l^s used at this level. E.g., in Fig. 2b, an eNB generates a new key every 10, 20, and 30 minutes at level 0, 1, and 2 respectively. Then, an eNB sends the token $\tau_{c,t} = \{(t_0^v, K_0), (t_1^v, K_1), (t_2^v, K_2)\}$ to all UEs residing in its cell c at some time during the time interval $t = [20, 30)$.

C. Message Deposition

When a legitimate sender wants to send a message to a specific *st*-region, he sends the message to the TPS, which encrypts and distributes it to the RPs. Here, the connections between sender and TPS, as well as between RPs and TPS must be secured by a cryptographic protocol like TLS. The request of the sender contains a description of the intended *st*-region, for example a rectangular area and a time interval. If the sender can be authenticated and is authorized to send an *st*-datagram, the TPS looks up the initial random seeds for the cells spatially overlapping with the destination region. Then, the TPS uses these seeds and the known time spans t_l^s at which keys are generated by eNBs to calculate the necessary tokens.

Since a token $\tau_{c,t}$ has different keys $K_{c,t,l}$, the TPS distributes the *st*-datagram based on the level that is currently valid according to the validity periods t^v of the levels. It only uses keys that are currently valid and encrypts the message with each key K. For example, given token hierarchy $L = \{(0, 10, [0, 30)), (1, 20, [30, 60)), (2, 30, [60, 90))\}$ (times in minutes) and a token that has been announced 40 minutes ago, the TPS will encrypt and deposit the *st*-datagram using the key of level 1. According to the procedure described above, the TPS obtains the RP identifier $rp_K = h(h(K))$ and the *st*-region identifier $r_K = h(K)$ for each key K. Again, the hash function is necessary to enable message lookup without revealing the key The TPS obtains the addresses of the RPs by resolving $rp_K \bmod N$ via DNS, where N is the known number of RPs. Then, for each key K, the TPS encrypts and sends one *st*-datagram as well as r_K to the respective RP. Finally, the RPs store the datagrams for lookup using r_K.

D. Message Delivery

UEs store the tokens received from the eNBs during their visit in the respective cells. This trail of tokens is then used to check for new messages at regular time intervals. When checking for messages, for each token $\tau_{c,t}$, a UE chooses the key based on the level that is currently valid according to the validity periods t^v. It also obtains the *st*-region identifier $r_K = h(K)$ from the key and the address of the RP $rp_K = h(r_K)$. Then, the UE sends a polling message containing region identifier r_K to all RPs rp_K (one polling message for each keys K). RPs receiving a polling message perform a lookup for the given region identifier r_K and deliver all *st*-datagrams that are available. Here, the connections between UEs and RPs are also protected by a cryptographic protocol like TLS. Finally, UEs decrypt the received *st*-datagrams with the respective symmetric key K.

IV. Privacy Discussion

As stated in section II, we assume that potential attackers have one or more of the following goals: to infer the identity of users receiving an *st*-datagram, their locations, co-location, absence, or a plaintext of an *st*-datagram for a region he has not been residing in. Attackers may observe the communication between entities, employ UEs, and send *st*-datagrams in order to achieve these goals. More sophisticated attackers might even compromise RPs and eNBs. However, attackers may not compromise the TPS and obtain access to the initial random seeds. We consider this a legitimate assumption since it is easier to control access to one or only few TPSs, whereas a large number of RPs may be needed for scalability reasons. Furthermore, we assume that attackers are not able to compromise the Packet Data Network Gateway (PGW), Home Subscriber Server (HSS), or Mobility Management Entity (MME) of the cellular operators. Otherwise, an adversary would already be able to track the movement of all subscribed users and obtain their identity. Given the above mentioned abilities of adversaries, we consider the following four attacks of increasing power.

A. Observation Attack

In the first potential attack, adversaries observe the communication between eNB and UEs, UEs and RPs, TPS and RPs, as well as sender and TPS. Due to the employed TLS protocol, attackers may only violate the objectives of location, co-location, and absence privacy of the users within the cell he is currently residing in. This, however, is not due to the STM service, but the possibility to directly observe users in his surroundings. Furthermore, attackers may only infer the identify of users from their location, as inferring the identity from IP address, International Mobile Subscriber Identity (IMSI), or Temporary Mobile Subscriber Identity (TMSI) requires additional knowledge from the cellular operator. Finally, due to his presence in the cell at the specific time, an adversary becomes a legitimate receiver. Hence, obtaining the symmetric key K does not violate the objective of confidentiality.

B. Probing Attack

In the second potential attack, adversaries observe the communication between TPS and RPs, as well as between UEs and RPs. By sending an *st*-datagram to a specific region, he tries to infer the RP that is responsible for the given *st*-region. Then, he tries to identify users having resided at this region by observing which UEs poll the respective RP. Here, an attacker must be able to recognize his message among other messages being exchanged between TPS and RPs. This, however, is only possible if he is the only one sending an *st*-datagram as the employed TLS protocol does not allow him to decrypt these messages. Nevertheless, even if an adversary can obtain the corresponding RP identifier rp_K, he is not able to obtain the region identifier r_K due to the preimage resistance of $h(x)$. Accordingly, he cannot obtain K without having resided in the intended *st*-region. Given these constraints, we now provide a detailed discussion of each privacy and security objective:

1) Anonymity: In this attack, adversaries are not able to infer the identity of users from their IP address as this requires access to additional knowledge from the cellular operator.

2) Location privacy: Attackers are also not able to obtain the locations of receivers as they may only learn about the RPs that are responsible for certain *st*-regions. However, as RPs are responsible for a large number of different *st*-regions in an unpredictable manner according to $rp_K = h(r_K)$, they cannot infer the *st*-regions addressed by UEs polling a RP.

3) Co-location privacy: According to the objective of location privacy, attackers can also not learn whether two UEs polling the same RP have been residing in the same *st*-region.

4) Absence privacy: Adversaries can only learn about the absence of users from an *st*-region if the corresponding RP is not being polled by a UE. This, however, is not the case as RPs are responsible for a large number of different *st*-regions.

5) Confidentiality: Attackers are not able to obtain K unless they have resided in the addressed *st*-region.

C. Compromising RPs

More sophisticated adversaries may also compromise one or more RPs. This enables attackers to gain access to the region identifiers r_K mapped to this RP. However, they are still not able to obtain the original *st*-regions of these identifiers due to the preimage resistance of $h(x)$. In order to obtain the mapping $r_K \rightarrow$ *st*-region, adversaries have to rely on probing messages as described above. Therefore, they have to guess an *st*-region which is mapped to the compromised RP. Please note that guessing an *st*-region requires the adversaries to actually send probing messages with different regions via the TPS. This is due to the fact that they cannot guess keys K for regions resulting in an identifier $r_K = h(K)$ located on the compromised RP. Furthermore, in their probing messages, attackers have to address a single cell c during one time slot t^s to obtain a unique mapping of r_K to *st*-region. This results in a potentially large number of probing messages. Since authenticated senders cannot perform a Sybil attack, a TPS can detect and filter probing messages (such a mechanism is also needed to prevent spamming and Denial-of-Service (DoS) attacks). Moreover, due to the token hierarchy, messages are delivered to different RPs using different *st*-region identifiers r_K based on the current level l. Since l depends on the time that has passed since the destined *st*-region, attackers only have a limited time for probing *st*-regions before a higher level and thus different RPs are responsible. Given these constraints, we now discuss the given objectives individually:

1) Anonymity: According to the previous attack, adversaries are not able to infer the identity of users from their IP address as this requires knowledge from the cellular operator.

2) Location privacy: Attackers may be able to partially violate the location privacy of users. This, however, requires that they are able to obtain the mapping of r_K to *st*-region. Furthermore, while adversaries may infer a specific *st*-region where several anonymous users have been residing, they are not able to track their movements over several time slots and cells unless they control all RPs. This is due to the fact that different *st*-regions are mapped to RPs using $h(x)$.

3) Co-location privacy: With this attack, the co-location privacy of users may be violated by detecting whether two users poll using the same r_K. Nevertheless, in order to find out if two known users have been neighbors at some time, attackers have to be able to specifically compromise the RP responsible for the assumed *st*-region of the meeting. Furthermore, they have to know the current IP addresses of both UEs to detect whether these UEs poll using the same r_K.

4) Absence privacy: Attackers may violate the absence privacy of users with this attack by detecting whether a user does not poll a certain r_K. However, in order to find out if a known user has been absent from a specific *st*-region, an attacker has to know the current IP address of the user and the r_K of the relevant region. Moreover, due to the token hierarchy, attackers have to be able to specifically compromise the RPs responsible at the current level.

5) Confidentiality: Adversaries are not able to obtain K from $r_K = h(K)$ due to the preimage resistance of $h(x)$.

D. Compromising eNBs

Assuming that attackers are able to compromise eNBs, they may violate all given objectives for users in this region. Although it may be rather unlikely for attackers to compromise eNBs without being noticed, the approach is still able to provide graceful degradation. Hence, the violation is limited to the cell, affecting only *st*-regions after the compromise.

V. Performance Evaluation

We evaluated the efficiency of our scheme in terms of the polling overhead and the delivery accuracy. Therefore, we implemented CSTM using OMNeT++ [7] and INET. We built a simplified architecture of a cellular network consisting of UEs, eNBs and a PGW connected to a router and the RPs.

We modeled the mobility of users using SUMO [8] and the TAPAS Cologne scenario (between 6 and 8 am) [9]. On one hand, we chose this vehicular scenario, because it provides realistic mobility for a large-scale setting with over 100 000 vehicle trips. On the other hand, we assume that UEs traveling at a speed of about 50 km/h present a high-load situation as UEs receive a lot of tokens. We extended SUMO to incorporate the locations of eNBs and used a Voronoi diagram of eNB coordinates to model their coverage area. In order to obtain a trace of cell switches, we calculated the intersection between the positions of vehicles and the Voronoi cells. We used this simple approach, as we are only interested in cell mobility patterns, not the exact points where UEs relocate to other cells.

The eNB coordinates were obtained from a website collecting such locations for Germany (http://www.senderliste.de/). We converted the given street addresses to geo-coordinates using Nominatim (http://nominatim.openstreetmap.org) and adjusted them to the (x, y)-coordinates of the Cologne scenario.

In our simulation, the 604 eNBs announced tokens every 3 minutes or delivered them to UEs when they entered their coverage area, whereas UEs polled RPs every 10 minutes. We used a token hierarchy with $\lambda = 4$ levels $L = \{(0, 3, [0, 15)), (1, 6, [15, 30)), (2, 9, [30, 60), (3, 12, [60, 120))\}$ (times in minutes). For spatial aggregation, we used simple random graph partitioning, grouping radio cells into random k-clusters at level l, where $k = l + 1$.

For each of the 32 repetitions, we let a sender send 100 *st*-datagrams to a randomly chosen *st*-region (uniform distribution). Its rectangular shape was randomly chosen anywhere in the city. The time span of the addressed *st*-region was $[a, b]$, where a was randomly chosen from interval $[0, 5]$ and

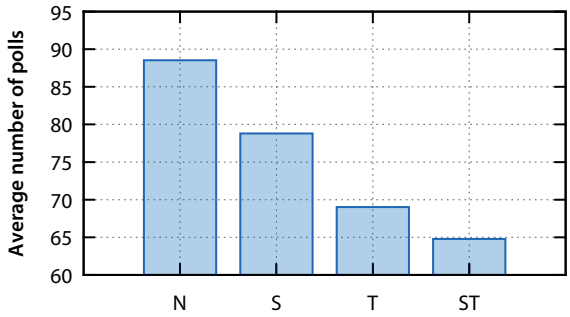

Fig. 4. Average number of poll messages sent by an UE.

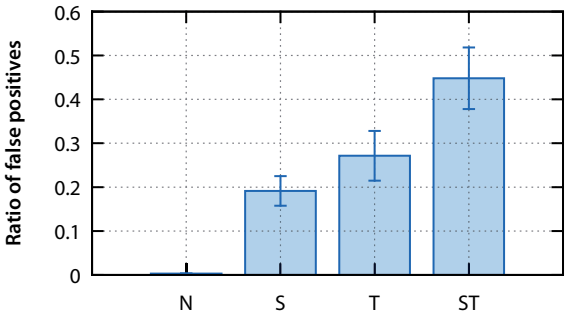

Fig. 5. Ratio of false positives among all messages received by UEs.

b from interval $[10, 15]$. Hence, the duration of the *st*-region was between 5 and 15 minutes. The sender waited a random time between 10 and 60 minutes after the end of the addressed *st*-region before sending the respective *st*-datagram.

We evaluated the following scenarios: (N) no token hierarchy, (S) only spatial aggregation, (T) only temporal aggregation, and (ST) both spatial and temporal aggregation.

First, we measured the average number of poll messages sent by each UE as shown in Fig. 4. Since the 99 % confidence intervals were narrow, we omitted them here. From this figure, we can see that without a hierarchy (N), a UE usually sends about 89 poll messages. In contrast, when employing a hierarchy, we can see the expected decrease of poll messages. For spatial aggregation (S), the number is reduced to about 79 messages. We also see that temporal aggregation (T) is more effective than (S) with about 69 messages. On one hand, this behavior can be explained by the fact that UEs tend to stay within the same cell for a longer time. On the other hand, for (S), cells are only clustered randomly without considering the road network. This may lead to cells being grouped into one cluster, although no UEs may travel between these cells. Finally, with both spatial and temporal aggregation (ST), we can see a further decrease to about 65 messages.

While the token hierarchy is able to provide the expected decrease of poll messages, it should also result in *st*-datagrams being received by UEs not having resided in the intended region. Therefore, we measured both the total number of messages being received and the total number of false positives for all UEs. We then divided the number of false positives by the total number of messages to obtain the ratio of false positives. Fig. 5 shows the false positives ratio with 99 % confidence intervals. According to our expectations, for (N), there are no false positives. For (S), the false positives increase to about 19 % while (T) results in a ratio of about 27 %. Furthermore, for (ST), we obtain the highest ratio of about 45 %. This shows the trade-off between the polling overhead and the delivery accuracy and corresponds to our observation that random spatial aggregation is less efficient than temporal aggregation. Hence, the ratio of false positives is slightly higher for (T) than for (S).

Our evaluation shows the potential benefit of using a token hierarchy in order to reduce the polling overhead. Therefore, we consider finding the optimal trade-off between the polling overhead and the false positives ratio part of our future work.

VI. Conclusion

In this article, we introduced the novel concept of STM. We provided an extensive overview of the envisioned applications and objectives to be fulfilled. Moreover, we proposed a first approach to realize such a service and provided a detailed discussion of its privacy features. Finally, we evaluated our approach in a large-scale simulation setup. We showed the potential benefit of employing a token hierarchy which provides a trade-off between delivery accuracy and polling overhead.

Several issues remain to be improved in further studies. First of all, we plan to evaluate the performance of our approach regarding the functional and non-functional objectives in more detail. Furthermore, instead of random graph partitioning, optimization-based strategies should be considered for spatial aggregation. Finally, we plan to investigate the potential of distributed approaches for realizing STM services.

Acknowledgment

This work is supported by the German Research Foundation (DFG Graduiertenkolleg 1487, Selbstorganisierende Mobilkommunikationssysteme für Katastrophenszenarien).

References

[1] C. Maihöfer, “A Survey of Geocast Routing Protocols,” *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, pp. 32–42, 2004.

[2] Q. Huang, C. Lu, and G.-C. Roman, “Spatiotemporal Multicast in Sensor Networks,” in *ACM SenSys*, 2003, pp. 205–217.

[3] W.-S. Ku, Y. Chen, and R. Zimmermann, “Privacy Protected Spatial Query Processing for Advanced Location Based Services,” *Wireless Personal Communications*, vol. 51, pp. 53–65, 2009.

[4] K. Mouratidis and M. L. Yiu, “Anonymous Query Processing in Road Networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 1, pp. 2–15, 2010.

[5] J. Manweiler, R. Scudellari, and L. P. Cox, “SMILE: Encounter-based Trust for Mobile Social Services,” in *ACM CCS*, 2009, pp. 246–255.

[6] A. Mohaisen, E. Y. Vasserman, M. Schuchard, D. Foo Kune, and Y. Kim, “Secure Encounter-based Social Networks: Requirements, Challenges, and Designs,” in *ACM CCS*, 2010, pp. 717–719.

[7] A. Varga, “The OMNeT++ Discrete Event Simulation System,” in *European Simulation Multiconference (ESM)*, 2001, pp. 319–324.

[8] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “SUMO – Simulation of Urban MObility: An Overview,” in *SIMUL, International Conference on Advances in System Simulation*, 2011, pp. 63–68.

[9] S. Uppoor and M. Fiore, “Large-scale Urban Vehicular Mobility for Networking Research,” in *IEEE VNC*, 2011, pp. 62–69.