Content Routing Algorithms to Support Publish/Subscribe in Mobile Ad Hoc Networks

Stephan Schnitzer
Institute of Parallel and Distributed Systems
University of Stuttgart
schnitzer <at> ipvs.uni-stuttgart.de

Hugo Miranda
Departamento de Informática
University of Lisbon
hmiranda <at> di.fc.ul.pt

Boris Koldehofe
Institute of Parallel and Distributed Systems
University of Stuttgart
koldehofe <at> ipvs.uni-stuttgart.de

***Abstract*—Mobile Ad Hoc Networks (MANETs) allow communication between moving nodes without using infrastructure like access points, stationary routers or GPS. This offers new communication opportunities, especially in challenging environments. To communicate in MANETs we often need routing functionality, which usually provides unicast-based best effort packet delivery. Publish/Subscribe (Pub/Sub) is a well known and powerful paradigm that provides higher expressiveness than unicast routing. It decouples senders from receivers and allows information exchange between network nodes that offer certain data (called publishers) and nodes that declare their interest in data of some pattern (called subscribers). Especially in MANET applications, Pub/Sub provides useful functionality to support realistic scenarios and novel applications.**

This paper proposes a new algorithm called *TPSR*, tailored to efficiently support Pub/Sub in MANETs. It is based mainly on two principles: i) it uses the dissemination of subscriptions to create source routes; and ii) it uses the signal strength messages are received with, to optimize routes in terms of striking a good balance between long routes and fragile routes. Simulations based on ns-2 demonstrate its performance, in comparison with flooding and unicast-based solutions.

I. Introduction

Publish/Subscribe (Pub/Sub) with support for content-based routing has evolved as a key paradigm for building applications of the Internet's Next Generation to gather and exchange data in loosely coupled and cooperating applications. For instance, event processing tools [1], [2] used in data analysis for stock exchange, traffic monitoring, and logistics heavily rely on Pub/Sub to efficiently collect information such as sensor readings. Events disseminated with Pub/Sub may serve as actions to trigger real-time adaptation of (physical) application processes, e.g., trigger an alarm if the sensor value has exceeded a critical threshold [3] or support the detection and exchange of traffic congestion situations [4] (we refer to a detailed overview of many more applications to [3], [5]).

In particular, every piece of information disseminated by Pub/Sub may be of interest to a different set of application components (subscribers). For instance, some components react only to low temperature readings while others are only interested in high temperature readings. By utilizing the diversity of interest and providing expressive subscription languages as well as advanced methods for routing, filtering and matching, research on infrastructure-based Publish/Subscribe systems (e.g., [5]–[11]) has established methods to significantly increase the data capacity and rates that can be accommodated between dependent application components.

This property is particularly desirable in a resource sparse environment such as Mobile Ad hoc Networks (MANETs). In a MANET, nodes cannot rely on a dedicated broker and network infrastructure, but are in charge to build up their own communication infrastructure. MANETs can be seen as an integral part of Future Networks to i) backup critical communication infrastructure by extending the communication range of the infrastructure and ii) support the energy efficient integration of sensor readings and measurements. For instance, the reliability of emergency services for extreme catastrophes may in the future greatly benefit from Ad hoc networks even operating when a large portion of the infrastructure has failed [12]. With the advent of Public Sensing, Ad hoc communication between mobile users has shown to have great potential to reduce the energy required to fulfill sensing tasks [13] and thus to integrate sensor streams in applications dependent on them.

However, the lack of infrastructure in MANETs makes it also extremely challenging to realize Publish/Subscribe efficiently in such an environment. Whereas the design goals of Pub/Sub systems include speed, flexibility and decoupling (between subscribers and publishers) [14], in the setting of MANETs these goals must be extended to address the environment constraints, in particular the number of packets sent which adversely impact devices lifetime. One of the most challenging characteristics of MANETs is mobility, since nodes are expected to do arbitrary movements. Although we can usually make some assumptions like estimating a maximum node speed, we often cannot rely on any kind of stationary nodes. Hence a good approach must cope with broken routes, lost packets, lack of position information and low energy consumption of mobile nodes. Providing all this at the same time is not easy and is often a balancing act between opposing design goals.

In this paper, we present the Tree-based Power-aware Source Routing (TPSR) protocol. TPSR is an alternative that aims to be more generic than existing solutions.

II. System model & problem description

A *MANET* consists of a number of mobile, portable devices with a mobile power source (e.g., a battery). We assume that

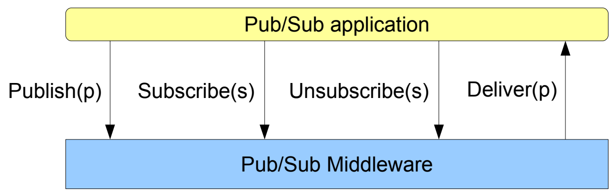


Fig. 1. Publish/Subscribe system model

nodes communicate by 2-way radio with other nodes that are within their transmission range. Moreover, they do not depend on infrastructure, so we cannot assume that stationary routers or GPS that provides location information, are available.

Publish/Subscribe is an asynchronous message delivery paradigm [15], [16], typically realized as middleware providing Publish() and Subscribe() methods to the application, as depicted in Fig. 1.

An application calls Subscribe(s) to declare its interest in publications satisfying some constraints set s (a *subscription*); from a Pub/Sub system perspective, the node where the application is running becomes a *subscriber*. Accordingly, a *publisher* is a node that offers some publication p to subscribers by calling Publish(p). The goal of the Pub/Sub system is to deliver p to subscribers with matching subscriptions by triggering the $Deliver(p)$ up-call.

The two most popular Pub/Sub paradigms are topic-based and content-based. The topic-based paradigm follows a channel model, with publications and subscriptions using a limited and well-known set of channel ids as the matching constraint. In contrast, content-based Pub/Sub uses arbitrary constraints over the publication attributes and values (e.g., value ranges) and hence is more expressive than topic-based. In this paper, we focus on the content-based paradigm.

The development of a generic content-based system for MANETs must address several challenges. The Pub/Sub system must keep track of every subscription, but nodes can run out of battery, become outside of network coverage or be shutdown. This unreliability invalidates any attempt to centralize subscription information. Node movement amplifies this problem given that, for each publication, a route must be found for each interested subscriber. While many solutions [17]–[19] try to reduce these challenges by restricting movement patterns or assume that subscriptions overlap, this paper proposes a more generic solution that takes advantage of additional information made available at network interfaces.

III. Related work

Early Publish/Subscribe systems (e.g., [15]) usually followed the topic-based subscription model and some of them have emerged to industrial grade solutions [14], [20]. Most existing Pub/Sub systems have been designed for wired networks (examples are [5]–[8] to name a few). Much of the related research focuses on their optimization, e.g., to satisfy latency requirements [9], [10] or to minimize bandwidth usage [11]. Enhanced versions like [21], [22] allow publishers and subscribers to migrate their location. They assume that central parts of the wired network (which routes most of the traffic) changes very rarely, while, e.g., a subscriber may check out, travel to a different location, check in there and gets delivered relevant publications that have been issued while he was offline.

Solutions like [17]–[19] aim to the case where the mobile network obeys certain restrictions, like having some stationary nodes or follow certain movement patterns. Such restrictions reduce the effort, e.g., stationary nodes can act as a statically assigned routing overlay. However, for the wide application scenarios that have been anticipated for MANETs, those restrictions do not always apply, which opts for a more generic approach. In [23], a gossiping protocol tailored for Pub/Sub is presented. However, it follows a proactive event routing approach and does not consider subscription updates. This penalizes both the network lifetime and applicability, especially in the more general scenarios where the role or interests of each participant may change with time. The use of an ODMRP [24] dissemination tree and Bloom filters to merge similar subscriptions is proposed in [25]. Merging contributes to improve efficiency but limits subscription expressiveness as, in practice, it emulates a content-based model with a topic-based one. The work described in [26] uses MAODV to enable content-based routing. Their main focus is on efficiently managing topology changes. They use a multicast-based routing tree and support limited subscriptions expressiveness.

To disseminate information in MANETs, flooding is a widely used technique, for example for route discovery [27], [28] and a similar approach can be used for disseminating subscriptions, especially when the identity and location of the publishers is unknown. In flooding, each node retransmits the same message once, which results in an excessive number of retransmissions consuming non negligible amounts of battery and bandwidth [29]. This effort can be reduced with a careful selection of the nodes that retransmit. In PAMPA [30], for example, nodes use the Received Signal Strength Indicator (RSSI) to determine the extent of their listening period, with nodes receiving messages with a lower RSSI (i.e., more distant from the source) to wait less. A node retransmits only if during its listening period, less than a predefined number of duplicates were received. A similar approach for node selection is proposed in [31] although as an alternative to source route or distance vector routing protocols for MANETs. Delay tolerant networks may avoid flooding by relying on the node movement for message propagation to its 1-hop neighbors (e.g., [32]). However, it should be noted that energy savings are gained at the cost of an increase in latency.

IV. TPSR

The Tree-based Power-aware Source Routing (TPSR) protocol consists of two parts: a subscription dissemination mechanism and a publication delivery mechanism. Subscriptions are flooded to the network. Like in source routing protocols (e.g., DSR [28]) the flooding is simultaneously used by nodes to learn routes to the subscriber. Both the route and the subscription information are stored by every node in their local subscription table.

A recipient of some publication p is a node whose subscription matches p. The list of recipients and a route to access each of them is retrieved by the publisher from its local subscription table. Note that routes to several recipients may share the same next hop. For each publication, the publisher delivers a single copy of the publication and a list of recipients to each next hop, who then becomes responsible for forwarding the publication to these nodes. The following sections detail this simplified description of TPSR.

A. Subscription dissemination

A subscription message is broadcast by subscribers and flooded to the network. The message contains subscription filter, subscriber ID and serial number immutable fields. The minimum RSSI (Received Signal Strength) field is updated by any node with the minimum between the value originally in the field and the signal strength with which the message was received by the node. Finally, each node appends its own ID to the source route field, effectively creating a route between the subscriber and any node that receives the message. This is the route that will be used for delivering publications. This mechanism is usually named *source routing*. Source routing is effective if the route does not become broken between the moment the subscription is flooded and the moment at which the publication is delivered. However, in MANETs, a number of events, like node movement, interferences or node disconnection can result in route disruption. TPSR puts in place a number of mechanisms to prevent publications from not being delivered due to these events. One is subscription refreshing, periodically triggered by subscribers by repeating the dissemination of the subscription. The more frequently subscriptions are refreshed, the less likely will a route become broken before the delivery of a publication. However, subscription refreshing is a costly operation as it requires retransmissions from a large number of nodes. To address this problem, subscriptions are flooded using a variation of the PAMPA [30] broadcast algorithm that puts additional care in the mechanisms that lead to the definition of the source routes.

The broadcast nature of the wireless medium makes very likely that each node receives multiple copies of the same message, each retransmitted by a different node. Therefore, each node could learn different routes to the subscriber. However, these routes would perform differently, with some being longer or more likely to break due to node movement than others (see [18] for an example of a routing protocol addressing the problem). PAMPA reduces the number of retransmissions by inhibiting the retransmission of some message m by nodes that listen to enough retransmissions of m from other nodes. The listening period is locally determined by each node and is proportional to the RSSI. Therefore, a lower delay (expected on nodes more distant from the source) has a higher probability of retransmission. As a result, routes discovered by PAMPA tend to be composed by very distant nodes, which are more likely to become outside transmission range in a small time frame.

TPSR extends the original PAMPA algorithm by estimating the quality of each route. The goal is to have TPSR creating source routes that are assumed to provide a working route between subscribers and publishers for a reasonable amount of time (before nodes moved too much and the route is destroyed), while ensuring that the number of hops is small (thus reducing the number of retransmissions). To achieve this goal, TPSR prefers hops at an intermediate *target distance*, which are not too close to the maximum receive distance (fragile routes) and not too close to the previous node (longer routes). Determining a good target distance is challenging and depends of a number of factors like node speed, transmission range and signal attenuation. This issue is further addressed in Sec. V.

In PAMPA, the wait time is locally determined at each node by a function $del(RSSI) = k \times RSSI$, with k being a constant that maps RSSI units on time. For simplicity, in this paper we assume an estimated mapping of the RSSI on distance [33] on a network of homogeneous nodes and use function $del(d) = k' \times d$ which instead accepts the distance as its input parameter. For TPSR, we want to find a formula $del'(d)$, tailored to have nodes that are at an intermediate value of the transmission range to retransmit first, thus increasing their probability of participating in the route. This paper uses a function $del'(d)$ given by Eq. 1.

$$del'(d) = \begin{cases} del_{max} \times \left(+1 - \left(\frac{d}{d_t}\right)^2\right) & , d < d_t \\ del_{max} \times \left(-1 + \left(\frac{d}{d_t}\right)^2\right) \times \frac{1}{\left(\frac{d_{max}}{d_t}\right)^2 - 1} & , d \geq d_t \end{cases} \quad (1)$$

The function uses constants del_{max}, d_{max} and d_t representing respectively the maximum delay, the maximum transmission range and the target distance (i.e., the ideal location of the nodes that should retransmit). Function $del'(d)$ squares the ratio $\frac{d}{d_t}$ to reduce the probability of collisions from nodes in proximity and which would therefore, get a very similar delay value. Figure 2 depicts instances of $del'(d)$ function for different values of d_t when $d_{max} = 250\ m$ and $del_{max} = 0.2\ s$. In real deployments, RSSI readings are expected to be distorted by factors like distinct antenna gains and transmission power of the nodes and interference. However, it should be noted that their impact is limited as they, although changing the set of nodes selected by the algorithm for retransmission, do not prevent message propagation.

TPSR delay function improves the quality of the routes learnt by each node. However, it does not prevent each node from receiving multiple copies of a subscription, each from a different neighbor and therefore, providing a different route. The subscription table stores a single route for each subscription as the benefits of keeping several routes in the presence of node movement are not clear [34]. Routes are updated during subscription renewal. Routes received concurrently from the same subscription renewal are evaluated by

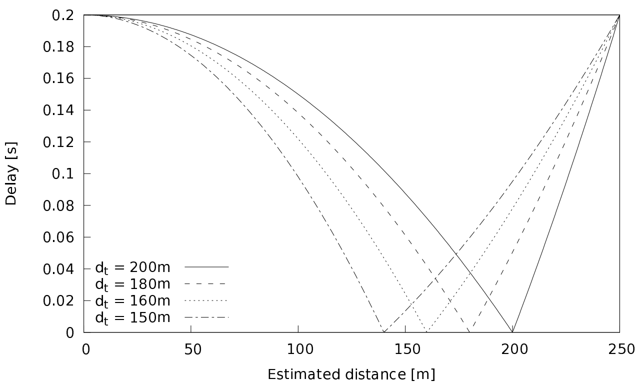


Fig. 2. TPSR forwarding delay based on estimated distance

```
procedure CREATERTJOBS(Recipients)
        ▷ Invoked by nodes preparing to broadcast publication message
    DELIVERPUBLICATIONTOME()          ▷ Check if node is subscriber
    routeJobs ← EmptyMap()
    delayIndex ← 0
    for all r ∈ Recipients do
        if r ≠ ThisNode() then
            nextHop ← (subscriptionTable[r] → nextHop())
            if nextHop ∉ routeJobs then
                routeJobs[nextHop] ← NewJob(delayIndex)
                delayIndex ← delayIndex + 1
            end if
            routeJobs[nextHop] → insert(r)
        end if
    end for
    return routeJobs
end procedure
```

Fig. 3. Algorithm to create route jobs

their length and minimum RSSI. Source routes carried on subscription dissemination are equally used to update routes to other subscribers present in the subscription table and that have forwarded the message. Routes are unconditionally updated as the newer route represents a more recent view of the network. An interesting side effect is that the probability of routes to different destinations to partially share the same path is increased. A benefit that will become visible once we discuss event routing.

B. Event routing

Publishers check which subscriptions match each particular publication by querying their local subscription table. For each recipient, the publisher uses the stored route to determine the *addressee* for the recipient, that is, the next hop that should be traversed by the publication so that it gets delivered to the recipient. A *route job* is a list of recipients sharing a single addressee. The algorithm depicted in Fig. 3 shows how route jobs are created. In addition to the publication itself, the publication message contains a minimum RSSI field analogous to the field with the same name in the subscription dissemination message and a list of route jobs, indexed by addressee.

The publication message is broadcast by the publisher to its 1-hop neighbors. The addressees receiving the message deliver

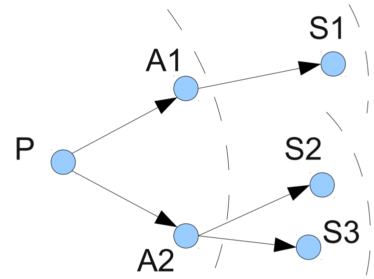


Fig. 4. Tree-based publication delivery by TPSR

the publication to the application layer if they are interested subscribers. In addition, they repeat the algorithm followed by the publisher for the recipients in its route job. That is, they $i)$ update the minimum RSSI field, if the signal strength of the message was lower than the one on the field of the incoming publication message; $ii)$ prepare a new array of route jobs, determined from their local subscription table; $iii)$ prepare a new message containing both the publication and the route jobs list; and $iv)$ broadcast this message to their 1-hop neighbors. To avoid collisions, resulting from concurrent retransmissions by addressees, each route job includes a unique delayIndex, which is an integer incremented by the sender and that is mapped by each addressee on a delay to be applied to its retransmission.

Figure 4 exemplifies the tree-based structures created during the delivery of publications and that justify the name of the protocol. In this figure, the publisher (P) broadcasts a publication message containing the publication and two route jobs, whose addressees are nodes A1 and A2, containing respectively subscribers $\{S1\}$ and $\{S2,S3\}$. Addressee A1 would then forward a publication message with a single route job for addressee S1. A2 in turn would create a new publication message with two route jobs, one for S2 and another for S3.

Publishers and addressees monitor the network to ensure the continuation of the propagation of the publications. Due to node movement, some of the addressees may no longer be in transmission range, thus creating broken links in the propagation tree. A node that sent a route job expects to receive the message the addressee forwards. Otherwise, it retries a fixed number of times. As a last resort, the publication is broadcast to the network using PAMPA.

C. Maintaining source routes

In spite of the node selection criteria followed during subscription dissemination, routes can still break due to node movement. TPSR implements two mechanisms to keep subscription tables with updated routes. As stated in the last section, when a route job cannot be delivered after several tries, it uses as its last option PAMPA to get the publication delivered. If a subscriber receives a publication from PAMPA, it will immediately reissue a subscription. This is justified by the assumption that at least one route from a publisher is permanently broken. Of course this assumption may not be true. For example a high collision rate may also be interpreted as a broken route. However, simulation results suggest that this assumption holds in the majority of the cases. We call this "reactive route repair" because the algorithm reacts by trying to repair broken routes.

The proactive approach permanently monitors the quality of the routes. Publication messages carry the minimal signal strength of the route. This value is updated by every addressee and delivered to the recipient together with the publication. A minimal RSSI value below some threshold will trigger a subscription renewal at the recipient. Establishing a threshold is not trivial. An early renewal increases network traffic. On the other hand, a late renewal gives no benefit and will force addressees to broadcast publications. A reasonable threshold probably also depends on the expected node speed (because speed is expected to destroy routes faster), the publishing rate (which influences the sampling rate at which route quality can be checked) and the movement model. Addressing these issues is left as future work.

V. Evaluation

Evaluation focuses on two aspects: $i)$ the success of TPSR in delivering publications to the interested subscribers and; $ii)$ the resources consumed by the network participants. The former is evaluated by the delivery rate metric, defined as the proportion of publications that were successfully delivered to interested subscribers. That is, let s_p be the number of subscribers known to be interested in some publication p and r_p be the number of interested subscribers receiving publication p. The delivery ratio dr is therefore defined by $dr = \frac{1}{|P|} \times \sum_{\forall p \in P} \frac{r_p}{s_p}$, where P is the set of publications issued during a simulation.

It has been shown (e.g. in [35]) that the number of transmissions significantly affects the lifetime of the devices. In addition, transmissions occupy air time and consume bandwidth and therefore, impact network performance. The second metric used is the total number of transmissions, performed by all nodes in the network.

The performance of TPSR was evaluated using the ns-2 network simulator in a number of different scenarios, whose parameters are summarized in Tbl. I. Values in [] represent the default value used when the other variable parameters are changed. TPSR was configured so that nodes decide not to retransmit if they listen to 3 copies of the message. Elsewhere [30], it was shown that this value provides a good tradeoff between delivery ratio and power consumption.

Results presented in this section are the average of 10 simulations in identical conditions using different randomization seeds and instances of the random waypoint movement pattern. Error bars in plots show the minimum and the maximum values out of the set of 10 simulations.

TPSR is compared with P-PUBS, a control approach which consists in naively flooding all publications, thus avoiding the overhead induced by subscription dissemination. P-PUBS uses PAMPA as the flooding algorithm with nodes configured to not retransmit after listening to two retransmissions. The interested reader is referred to [36] for an extended evaluation of TPSR.

Attribute	**Value [Default]**
Node density	50 to 200 nodes/km^2 [140]
Max. node speed	0 to 10 m/s [3]
Publishing rate	0.5 publications/s/publisher
RWP pause time	200 s
Number of publishers	10
Number of subscribers	10
Number of nodes	200
Target distance of TPSR	0 to 250 m [180]
Max. receive distance	250 m
Network links	11 MBit/s @914 MHz
Transmission power	25 dBm
Antenna gain	1
ns-2 propagation model	TwoRayGround

TABLE I
Simulation Parameters

By default, nodes renew their subscriptions every 200 s. The subscription interval can be lower due to the need to rebuild source routes or by a node decision to change its subscription. Nodes re-evaluate their subscription every 60 s. To approach the simulations to realistic scenarios, for example, to applications using geo-referencing, the subscription is probabilistically biased by the node location.

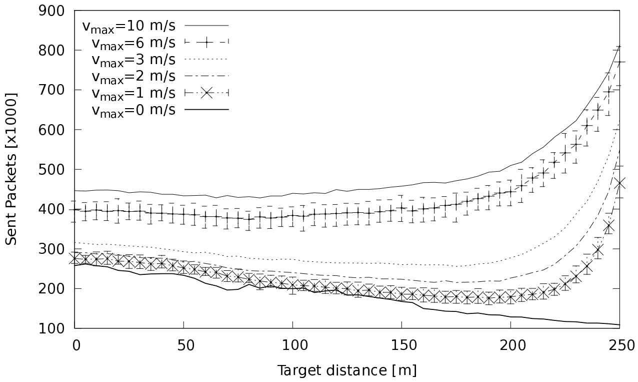
A. Finding a good target distance

While nodes can easily obtain the RSSI for a message, deriving a distance from this value is hard. For real deployments, [37] proposes $RSSI = P_t \times \left(\frac{c_1}{d}\right)^n \times c_2$ as an estimation with device specific factors being the transmission power P_t and constants c_1, c_2 and n. In particular, using our evaluation setup $n = 2$ and $c_1 \times \sqrt{P_t \times c_2} = 0.01386645232$. Therefore, our evaluation of TPSR estimates the distance by the simplified formula $d = \frac{0.01386645232}{\sqrt{RSSI}}$.

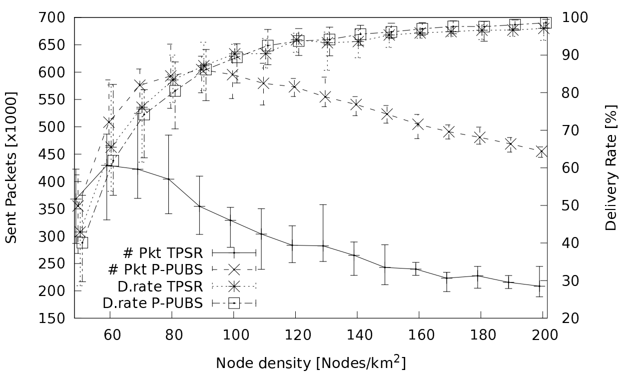
As discussed before, the target distance d_t impacts the number of retransmissions and route stability. However, several other parameters influence on route stability and therefore on an adequate target distance. Examples are:

- Speed, since this influences how long a route is usable
- Subscription rate, influences route rebuild frequency
- Publication rate, which influences how many times a route can be used

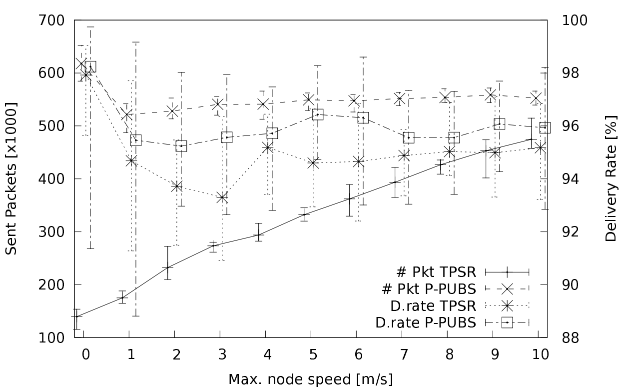
As speed is supposed to have the biggest influence, we evaluated the impact of this parameter on the target distance. Figure 5a depicts the number of packets sent for different target distances and maximum speeds, using the default evaluation parameters. As expected, plots show that the number of retransmissions increase with the speed and target distance. The performance at all speeds appears to degrade more sharply for target distances above 180 m. Therefore, this will be the target distance used in the remaining simulations.



(a) Influence of node speed and d_t



(b) Influence of node density



(c) Influence of max. node speed

Fig. 5. TPSR simulation results

B. Speed and Node density

Figures 5b and 5c compare the number of packets sent and the delivery ratio between TPSR and P-PUBS for distinct node densities and speeds. Plots show that TPSR is able to provide a delivery rate comparable to flooding although transmitting significantly less messages. The poor performance of the flooding approach in low node densities suggests that these scenarios are affected by network partitions. The number of transmissions of P-PUBS confirm the results presented in [30], specifically, that PAMPA self-adapts the number of nodes required to retransmit to the node density. Because TPSR uses PAMPA to flood subscriptions, a similar pattern is found when node density is varied. However, TPSR always requires a lower number of retransmissions, what suggests that the cost of flooding the subscriptions is rapidly compensated by the publications delivery mechanism. The growing number of retransmissions required by TPSR with an increasing average node speed supports this conclusion. As nodes move faster, routes tend to break more often and TPSR initiates an higher number of flooding operations to repair them. The curve patterns suggest that at some speed above 10 m/s it would be preferable to rely on P-PUBS. Estimating this point is left for future work.

C. Performance of unicast-based protocols

Although specifically tailored for a Pub/Sub environment, TPSR implicitly defines a new routing protocol for MANETs. Hence it is important to understand if some other routing protocol could not be used for the publications delivery. Figure 6 compares the performance of AODV and DSR, two well known representatives of unicast-based routing protocols, with P-PUBS(1) and P-PUBS(2) which configure PAMPA to avoid retransmissions from nodes listening respectively to 1 and 2 retransmissions. The most favorable conditions for unicast protocols were chosen for this test. In particular, a single subscriber exists for each publication, meaning that a single unicast message must be delivered per publication. The scenario considers 10 publishers, 4 subscribers and a publication ratio of 0.5 publications per second per publisher on a 11 MBit/s Ad hoc network. To avoid any bias concerning subscription dissemination, it was assumed that nodes have in advance all knowledge about the filters of the subscribers. In brief, for each publication he produces, the publisher will be required to send it to one subscriber, known in advance.

Results show that in the general case, flooding a message to all nodes is more efficient, both in the number of retransmissions and in the delivery ratio, than sending it only to a single recipient using unicast routing. This result supports our decision to develop TPSR and compare its performance with P-PUBS. The poor performance of both AODV and DSR is attributed to the observed high packet loss rates. In the majority of the cases, when a packet could not be delivered using an old route, both AODV and DSR initiate a route discovery. However, route discovery packets are flooded to the network. A small number of concurrent route discoveries will result in a non-negligible number of collisions and in contention, a phenomenon usually coined as broadcast storm [29] and which is known to further degrade throughput.

VI. Conclusions and Future Work

Although challenging, Pub/Sub in MANETs can provide a valuable contribution to support different kinds of applications. This is especially true for scenarios where network infrastructure is not present or not working. Existing MANET solutions are often quite restrictive in respect to node movement or

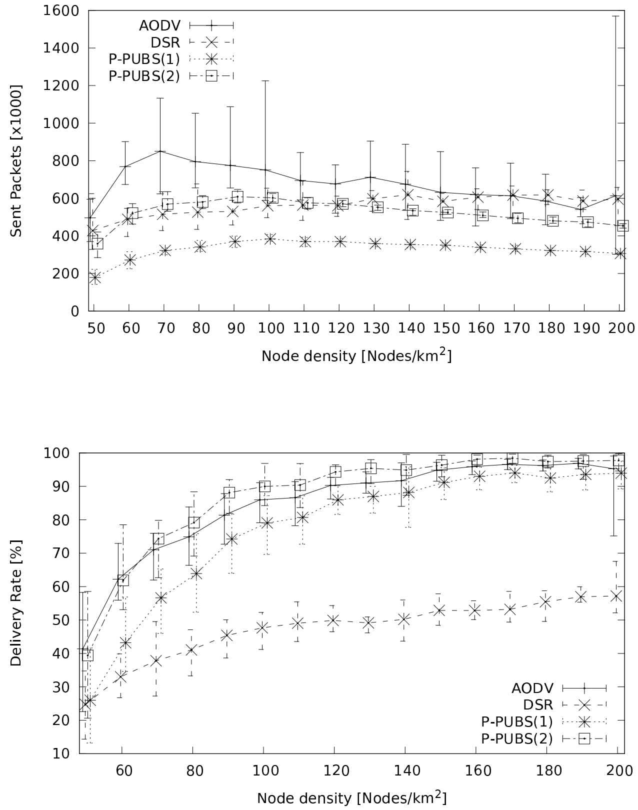


Fig. 6. AODV/DSR simulation results

subscription patterns. Likewise, at least for some scenarios, AODV and DSR cannot offer decent performance.

This paper presented TPSR, a novel content-based Pub/Sub protocol specifically tailored for MANETs. The protocol floods subscriptions and uses the flooding to establish source routes to the subscriber. In addition, a number of mechanisms was put in place to efficiently repair broken routes. Simulation results have shown that in many cases, TPSR outperforms both publication flooding and their delivery using unicast routing protocols.

As part of future work, we plan to implement a *local route repair* mechanism to further avoid flooding operations. Another improvement would be to make the subscription forwarding delay to self-adapt to the detected network conditions.

Acknowledgments

Stephan Schnitzer and Boris Koldehofe were partially supported by the German Academic Exchange Service (DAAD) in the context of the UP2P project. Hugo Miranda was partially supported by Fundação para a Ciência e Tecnologia (FCT) through project PTDC/EIAEIA/103751/2008 - PATI.

References

[1] A. Buchmann and B. Koldehofe, Eds., *IT-Information Technology*. Oldenbourg Verlag, October 2009, vol. 51.

[2] G. G. Koch, B. Koldehofe, and K. Rothermel, "Cordies: Expressive event correlation in distributed systems," in *Proceedings of the 4th International Conference on Distributed Event-Based Systems (DEBS'10)*. ACM, July 2010, Conference Paper.

[3] A. Hinze, J. Bacon, A. P. Buchmann, S. Chakravarthy, M. Chandi, A. Gal, D. Gawlick, and R. Tibbetts, "Panel," in *Principles and Applications of Distributed Event-Based Systems*, 2010, pp. 432–454.

[4] B. Koldehofe, B. Ottenwälder, K. Rothermel, and U. Ramachandran, "Moving Range Queries in Distributed Complex Event Processing," in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS).* Berlin: ACM, July 2012, Conference Paper, pp. 201–212.

[5] H.-A. Jacobsen, A. K. Y. Cheung, G. Li, B. Maniymaran, V. Muthusamy, and R. S. Kazemzadeh, "The padres publish/subscribe system," in *Principles and Applications of Distributed Event-Based Systems*, 2010, pp. 164–205.

[6] P. R. Pietzuch and J. Bacon, "Hermes: A distributed event-based middleware architecture," in *ICDCSW '02: Procs. of the 22nd Int'l Conf. on Distributed Computing Systems*, 2002, pp. 611–618.

[7] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Design and evaluation of a wide-area event notification service," *ACM Transactions on Computer Systems*, vol. 19, no. 3, pp. 332–383, 2001.

[8] G. Mühl, "Large-scale content-based publish-subscribe systems," Ph.D. dissertation, TU Darmstadt, November 2002.

[9] A. Tariq, B. Koldehofe, G. Koch, and K. Rothermel, "Providing probabilistic latency bounds for dynamic publish/subscribe systems," in *Proceedings of the 16th ITG/GI Conference on Kommunikation in Verteilten Systemen (KiVS).* Springer, 2009.

[10] M. A. Tariq, G. G. Koch, B. Koldehofe, I. Khan, and K. Rothermel, "Dynamic publish/subscribe to meet subscriber-defined delay and bandwidth constraints," in *The Sixteenth International Conference on Parallel Computing (Euro-Par)*, 2010.

[11] M. A. Tariq, B. Koldehofe, G. G. Koch, and K. Rothermel, "Distributed spectral cluster management: A method for building dynamic publish/subscribe systems," in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS)*, 2012.

[12] S. Alves, B. Koldehofe, H. Miranda, and F. Taiani, "Design of a Backup Network for Catastrophe Scenarios," in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, ser. IWCMC '09. New York, NY, USA: ACM, 2009, pp. 613–617.

[13] P. Baier, F. Dürr, and K. Rothermel, "PSense: Reducing Energy Consumption in Public Sensing Systems," in *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012).* IEEE, Mar 2012, pp. 136–143.

[14] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, 2003.

[15] K. P. Birman and T. A. Joseph, "Exploiting virtual synchrony in distributed systems," *SIGOPS Oper. Syst. Rev.*, vol. 21, no. 5, pp. 123–138, 1987.

[16] D. S. Rosenblum and A. L. Wolf, "A design framework for internet-scale event observation and notification," *SIGSOFT Softw. Eng. Notes*, vol. 22, no. 6, pp. 344–360, 1997.

[17] J. A. Briones, B. Koldehofe, and K. Rothermel, "SPINE : Adaptive Publish/Subscribe for Wireless Mesh Networks," *Studia Informatika Universalis*, vol. 7, no. 3, pp. 320–353, October 2009.

[18] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, "Signal stability based adaptive routing (SSA) for ad-hoc mobile networks," University of Maryland, Tech. Rep. CS-TR-3646 and UMIACS-TR-96-34, 1996.

[19] R. Meier and V. Cahill, "Steam: Event-based middleware for wireless ad hoc network," in *ICDCSW '02: Procs. of the 22nd Int'l Conf. on Distributed Computing Systems*, 2002, pp. 639–644.

[20] B. Garbinato, H. Miranda, and L. Rodrigues, *Middleware for Network Eccentric and Mobile Applications*. Springer, 2009.

[21] J. Bacon, K. Moody, J. Bates, R. Hayton, C. Ma, A. McNeil, O. Seidel, and M. Spiteri, "Generic support for distributed applications," *Computer*, vol. 33, no. 3, pp. 68–76, 2000.

[22] M. Caporuscio, P. Inverardi, and P. Pelliccione, "Formal analysis of clients mobility in the siena publish/subscribe middleware," Department of Computer Science, University of L'Aquila, Tech. Rep., 2002.

[23] S. Yoo, J. H. Son, and M. H. Kim, "A scalable publish/subscribe system for large mobile ad hoc networks," *J. Syst. Softw.*, vol. 82, no. 7, pp. 1152–1162, 2009.

[24] S. J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," vol. 7, no. 6. Hingham, MA, USA: Kluwer Academic Publishers, Dec. 2002, pp. 441–453.

[25] E. Yoneki and J. Bacon, "Dynamic group communication in mobile peer-to-peer environments," in *SAC '05: Procs. of the 2005 ACM symposium on Applied computing*, pp. 986–992.

[26] L. Mottola, G. Cugola, and G. P. Picco, "A self-repairing tree topology enabling content-based routing in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 8, pp. 946–960, 2008.

[27] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Procs. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.

[28] D. B. Johnson and D. A. Maltz, *Mobile Computing*. Kluwer Academic Publishers, 1996, ch. Dynamic Source Routing in Ad Hoc Wireless Networks, pp. 153–181.

[29] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, 2002.

[30] H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen, "A power-aware broadcasting algorithm," in *Procs. 17th IEEE Int'l Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*, 2006.

[31] M. Meisel, V. Pappas, and L. Zhang, "Listen first, broadcast later: Topology-agnostic forwarding under high dynamics," UCLA Computer Science Department, Tech. Rep. 100021, 2010.

[32] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 748–760, Jun. 2008.

[33] T. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2001.

[34] C. Perkins, E. Royer, S. Das, and M. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 16–28, Feb. 2001.

[35] L. M. Feeney, "An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–249, 2001.

[36] S. Schnitzer, "Geographical distribution schemes to support Publish/Subscribe in Mobile Ad Hoc Networks," Diploma thesis, University of Stuttgart, Germany, 2010.

[37] E. K. Wesel and E. Khayata, *Wireless Multimedia Communications: Networking Video, Voice and Data*. Addison-Wesley, 1997.