Topology Control Made Practical: Increasing the Performance of Source Routing[⋆]

Nicolas Burri, Pascal von Rickenbach, Roger Wattenhofer, Yves Weber

Computer Engineering and Networks Lab, ETH Zurich
CH-8092 Zurich, Switzerland
{nburri, pascalv, wattenhofer, webery}@tik.ee.ethz.ch

Abstract. Wireless ad hoc and sensor networks need to deal with unstable links. In practice the link quality between neighboring nodes fluctuates significantly over time. In this paper we evaluate the impact of topology control on routing performance. We propose a dynamic version of the XTC topology control algorithm. This simple and strictly local protocol removes unreliable and redundant links from the network. By means of physical experiments on an indoor mica2 testbed we study the beneficial effects of topology control on source routing, one of the most common routing schemes for ad hoc and sensor networks. In particular we compare the performance of source routing with and without topology control. Our results show that topology control reduces route failures, increases network throughput, and diminishes average packet delay.

1 Introduction

Sensor networks ask for highly optimized protocols. In order to meet the demands, we witness that more and more researchers acquit themselves of the orthodox layering hierarchy, pushing the envelope of their protocols with cross-layer design. Abandoning layering (which after all is one of the most well-accepted principles in networking) however comes at the cost of reusability. Moreover, it is not always clear that an integrated cross-layer design has advantages over a well-defined layered interface. In this paper we consider the effect of eliminating unreliable connections at the link layer (a technique usually known as topology control) on higher communication layers. Nowadays this task is often integrated into the network layer, leading to complex routing protocols. We limit our study to non-mobile wireless networks. Such networks are sometimes called mesh or rooftop networks. In spite of being static, link qualities vary over several different time scales, from seconds to hours, due to interference or mobile obstacles. This leads to frequent network topology changes. Hence, if it comes down to implementing a real system for wireless networks the network stack has

---

[⋆] The work presented in this paper was partially supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322

to deal with the problems arising from unreliable communication links. In literature, many studies have been published under the name of topology control trying to mitigate the effects of unstable network connections [1,2,3,4,5]. So far, topology control has only been considered from a theoretical point of view. Researchers have devised algorithms establishing a subgraph of the initial network by dropping specific connections such that the resulting topology features a variety of desired properties. However most topology control algorithms are based on assumptions that are questionable in practice. Besides too simplistic network models it is also often assumed that the nodes have detailed information about their neighbors. Furthermore, the communication graph is always supposed to be static; consequently, all proposed algorithms compute their resulting topology only once. In practice there is no such thing as a perfect topology since the underlying network graph changes over time. All these *one-shot* solutions would need major adaptations in order to deal with dynamic networks. Thus, despite the considerable body of research devoted to topology control and the theoretical and simulation-based evidence of its effectiveness, to date there is little *experimental* evidence that topology control can actually be used to circumvent the problem of unreliable links in wireless networks. Apart from the efforts made in the field of topology control there have been attempts to cope with unreliable communication links directly on the network layer. Many routing protocols for wireless networks were proposed trying to predict link stability based on signal strength and to choose the "best" route using these predictions [6,7,8,9,10]. These protocols have in common that their decisions are threshold based; that is, if the link quality is above a fixed threshold the link is incorporated in a route. This can result in failed routing attempts even if working paths exist. Sometimes unreliable links need to be chosen in order to prevent the network from being disconnected.

In this paper we provide an implementation of the XTC algorithm [11] which guarantees connectivity of the network while discarding unreliable communication links. We have extended the algorithm proposed in [11] such that it now also copes with dynamic networks comprising fluctuating links. If we refer to XTC throughout the rest of the paper we always refer to this extended version. The algorithm is implemented on the mica2 sensor node platform which facilitates rapid prototyping for wireless ad hoc networks. In order to examine the benefits of XTC we evaluate the performance of source routing as a generic sample application in wireless networks with and without XTC. The algorithm manages to identify stable connections and thus preserves source routing from taking unreliable links. To be more specific, XTC decreases the number of route failures and retransmission attempts and is therefore able to increase network throughput and to lower average packet delay. Moreover, the omission of fluctuating links (if not needed) by XTC only modestly increases average route lengths.

The remainder of the paper is organized a follows: Section 2 compares our contributions with previous related work. In Section 3 an extended version of the XTC algorithm is described that also copes with a dynamic environment. The algorithm's behavior in practical networks is the subject of Section 4. Section 5 concludes the paper.

2 Related Work

The characterization of wireless links in different environments is analyzed in [12,13]. Both papers identify the existence of "gray links", links that are highly variant and unreliable. One of the goals of topology control is to shield the upper layers from the problems arising from these unstable links. However, modern topology control algorithms offer a multitude of other goals such as low node degree, planarity, or reduced power consumption (nodes adjust their transmission power level in order to save energy; a mechanism also known as *power control*). In this work, we concentrate on unreliable links. Power control is orthogonal to our solution and could be incorporated.

Most proposed topology control algorithms require hardware technology that is not available today. The algorithms presented in [4,3,5,14] require knowledge of exact node locations. Other work assumes that relative distance and directional information is available [2]. The protocol described in [15] needs the node distribution to be uniform-random. All mentioned protocols operate on idealized radio models such as unit disk graphs. The simple XTC algorithm [11] extended in Section 3 always maintains connectivity given a general weighted network graph. In [16] a generalized version of XTC called k-XTC is presented; this algorithms drops a communication link only if at least k alternative paths exist. In [17] the RTC algorithm is proposed that slightly changes XTC such that link qualities are determined randomly. In [18] the authors propose S-XTC, an extended version of [11] implemented on Bluetooth enabled sensor nodes.

To the best of our knowledge, there exists no practical evaluation of the influence of topology control to wireless networks – with one notable exception [19]. In [19] the authors provide an experimental study of the impact of variable transmission power levels on link quality. Their protocol uses power control and blacklisting to eliminate unreliable links but (in contrast to our work) does not give any connectivity guarantees of the resulting topology.

In the domain of routing protocols several previous papers improve the proposed protocols by predicting link qualities to enhance their performance. In [8] and [20] preemptive route maintenance algorithms based on signal strength are presented; they proactively initiate a new route discovery if a link on an active route becomes worse that a given threshold. In [21] a metric is presented to identify high-throughput routes when different links can run at different bitrates. However, their metric does not consider packet losses and is thus complementary to our work. There exists a number of wireless routing algorithms collecting per-link signal strength information and apply a threshold to avoid connections with high loss ratios [6,7,8,9]. In contrast to our work this approach may eliminate links that are necessary for connectivity (if the threshold is too high), or keep unnecessary bad-quality links (if the threshold is too low); both of these are likely to be issues in networks with many "gray" links.

The authors in [22] propose the ETX metric predicting the number of retransmissions required using per-link measurements of packet-loss ratios. The effectiveness of this approach is demonstrated by showing that the metric improves Dynamic Source Routing (DSR) in an experimental testbed using WLAN technology. A major drawback of their solution is that the flooding initiated during route discovery can generate a large amount of network traffic since in-

XTC Algorithm

```
Update order $\prec_u$ over $u$'s neighbors
Request current orders from neighbors
Select topology control neighbors:
    $N_u := \{\}$; $\widetilde{N}_u := \{\}$
    while ($\prec_u$ contains unprocessed neighbors) {
        $v :=$ least unprocessed neighbor in $\prec_u$
        if ($\exists\, w \in N_u \cup \widetilde{N}_u : w \prec_v u$)
            $\widetilde{N}_u := \widetilde{N}_u \cup \{v\}$
        else
            $N_u := N_u \cup \{v\}$
    }
```

termediate nodes are required to retransmit an already forwarded route request in case of a potentially better path. Consequently, network performance may degrade drastically since a single route search can trigger multiple (in theory, even exponentially many!) floodings—we have a broadcast storm.

3 XTC in Dynamic Networks

In [11] the XTC topology control algorithm was introduced exhibiting several desirable properties. However, the algorithm assumes the underlying network to be static. In this section we adapt the original algorithm to deal with dynamic networks comprising unreliable links. The algorithm still consists of three main steps: Neighbor ordering, neighbor order exchange, and link selection. Each node repeats these three steps periodically in order to cope with changing topologies. Although XTC is executed at all nodes, the following description assumes the point of view of a network node u.

In the first step a node u updates its total order $\prec_u$ over all neighbors in the network. From an abstract point of view, this order is supposed to reflect the quality of the links to the neighboring nodes. The neighbors of u are thereby arranged in $\prec_u$ with respect to decreasing link qualities. From an implementation point of view, node u first has to update the link qualities according to a particular metric. In our experiments the applied metric is based on the packet loss ratio of past transmissions. The XTC algorithm in [11] assumes that both endpoints of a link agree on the quality of their connection. In reality this does not have to be the case. Both endpoints can only judge the link quality from their own perspective and thus may come to different results. Consequently, the concerned nodes have to negotiate and settle on the same link quality value.

In the second step the neighbor order information is exchanged among all neighbors. To limit communication overhead, node u only requests $\prec_v$ if it has outdated order information from neighbor v.

During the third step, node u locally selects the neighboring nodes for the next iteration of the algorithm. For this purpose node u traverses $\prec_u$ with decreasing link quality: “Good” neighbors are considered first, “worse” ones later.

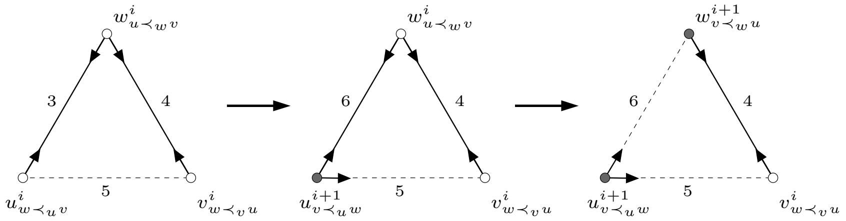


Fig. 1. On the left, the initial state of a sample network that consists of three nodes u, v, and w is depicted including links picked by XTC (solid). The quality of link (u, w) deteriorates from 5 to 2. The topology graph after an iteration of XTC at node u is shown in the middle. On the right, it can be seen that the topology becomes disconnected after w's update cycle.

Node u only builds a direct communication link to a neighboring node v if u has no "better" neighbor w that can be reached more easily from v than u itself. For a more detailed description of the third step we refer the interested reader to [11].

Besides the above mentioned issues related to the XTC algorithm itself, difficulties arise from the fact that neighboring nodes are not attuned to one another. In particular, we have to ensure that the properties of XTC as shown in [11] are still valid in the presence of partially updated order information. In case of changing link qualities it is necessary that a node detecting a change, triggers other neighbors to start another iteration of the algorithm to avoid temporary network partition. The problem is illustrated by means of a simple network consisting of three nodes u, v and w in Figure 1. The link quality values are depicted next to all possible connections. Higher values thereby indicate better link quality. We assume that all nodes have already executed i iterations of the algorithm at the beginning of our examination. Furthermore, let $u^i_{w \prec_u v}$ denote node u after the i-th iteration of XTC with the neighbor order $w \prec_u v$. On the left hand side of Figure 1 the network is in a consistent state where all nodes have finished iteration i. The arrows pointing out of a node indicate the links selected by XTC and solid lines show the resulting topology graph. In the middle, a snapshot of the network is shown after node u finishes its $(i+1)$-th update cycle. Note that the link (u, w) got worse. Consequently, node u selected (u, v) in iteration $i+1$ to be in the topology control graph. However, the connection between u and v is not yet established since v has not executed iteration $i+1$. On the right, Figure 1 depicts the situation after node w successfully completed the $(i+1)$-th iteration of XTC resulting in an order $v \prec_w u$. Using this order w drops the connection link (u, w) which results in a temporary partition of the network. However, when v has executed XTC as part of the completion of iteration $i+1$ the topology control graph will be connected again.

In order to obviate the above mentioned problem, a node detecting a change in link quality of a particular connection informs its neighbors. If a node receiving such a trigger message is adjacent to the affected link or contains both endpoints of the connection in its neighborhood it instantly executes XTC in order to minimize the duration of potential network partition.

4 Experiments

For empirical study the XTC algorithm described in Section 3 was implemented on the mica2 sensor node platform. On an office floor we set up networks of different sizes and ran various experiments evaluating the practical impact of topology control on source routing in wireless ad hoc and sensor networks.

4.1 Link Quality Metric

Defining a reasonable link quality metric is a fundamental requirement for a physical implementation of the XTC algorithm. To specify such a metric it is necessary to contemplate link characteristics of mica2 networks. Therefore, several experiments were made on various networks evaluating the behavior of links in different settings. It is a surprising result that even in segregated environments links do not have constant error probabilities over time. To exemplify this, a setup consisting of two nodes exchanging 1000 messages in 30 minutes is used of which 537 arrived. As can be seen in Figure 2 the link reliability worsened in the course of the experiment. After transmitting approximately 620 packets (of which 360 arrived at the receiver) packet loss increased drastically and the link started to fail for up to 50 consecutive packets. Such breakdowns could be observed in nearly all of our experiments.

Packet loss is a property which can be used to characterize the quality of a wireless link[1]. Figure 3 shows a link quality indicator for this experiment based on packet loss. This metric attempts to predict future packet loss based on past transmission failures. To achieve a reasonably stable link quality indicator we apply a moving average function on measured consecutive packet loss. Figure 3 shows that a moving average using 0.1 as weight for the current value leads to a fluctuating link quality indicator (dotted). Using a weight of 0.01 results in a more stable curve (solid). With this metric the lower link reliability during the last third of the experiment leads to a distinct decrease of the link quality indicator. Ultimately, the goal of the XTC algorithm is to avoid unreliable links. Ordering neighbors according to a metric based on packet loss allows to achieve this. Therefore, we decided to use this metric for all further experiments.

4.2 Source Routing and XTC

To evaluate the fitness of topologies created by XTC for real life purposes we implemented a basic source routing protocol. This implementation was designed to incorporate knowledge gained from the topology control algorithm. That is, for route discovery only edges which are part of the current XTC graph are used. Discovered routes are cached and reused until a route error occurs, independent of whether all used links are still part of the XTC graph. This proceeding is reasonable since the XTC graph changes over time. It always contains the currently most reliable links and thus it may occur that good links in the graph

---

[1] We have also evaluated the Received Signal Strength Indicator (RSSI) as a potential link metric. However it turned out that RSSI is no adequate link quality indicator for our purposes. Details can be found in [23].

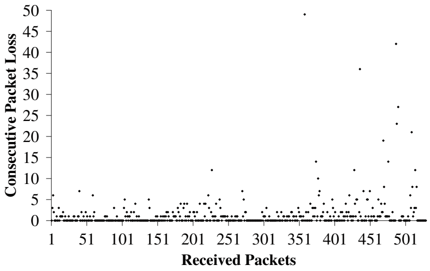


Fig. 2. Consecutive packet loss while sending 1000 messages over a link.

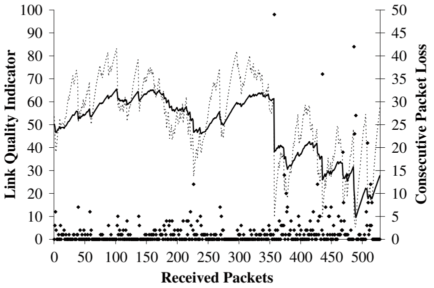


Fig. 3. Link quality indicator with weight 0.1 (dotted) and weight 0.01 (solid).

are replaced by even better ones. Existing routes using the replaced edges may still work perfectly well and should therefore not be discarded until message transmissions start to fail.

4.3 Small Testbed

The first experiment with our implementation of XTC was run on a small network of seven nodes placed in a zig-zag like topology shown in Figure 4. The transmission power of the nodes was adjusted such that the topology featured stable short links (solid) and unreliable longer links (dashed). It was set up in an empty corridor where it was possible to minimize external interference. Over a period of eight hours node 1 sent multi-hop ping messages to all other nodes in the network. To evaluate the impact of topology control on this experiment we measured the successful number of transmissions, route lengths, and necessary route searches with and without XTC, respectively. Figure 5 shows the results of this experiment. Since the network was designed to be reasonably stable, the number of successful transmissions is above 90% independently of whether topology control was enabled or not. However, XTC manages to reduce transmission failures to nearly all receivers. Especially the route stability to distant nodes was improved as can be seen in Figure 5(a). Since source routing always communicates over the first connection replying to a route request the risk of having unstable, long links on the used path increases with the number of hops; thus also the packet loss probability increases with increasing route length. XTC does not hinder communication over fast edges but prevents source routing from

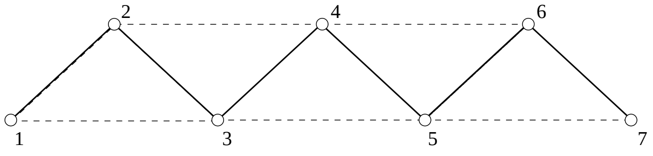
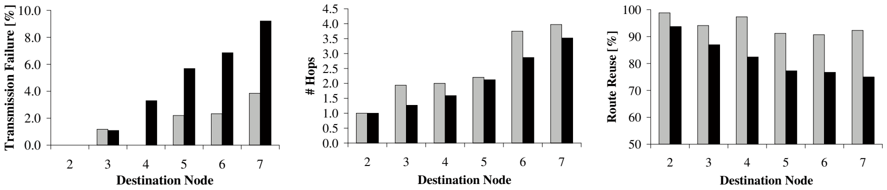


Fig. 4. Test network consisting of seven nodes. Node 1 is the initiator of all communication in the network.



(a) Transmission failures (b) Average route length (c) Route reuse

Fig. 5. Evaluation of the small testbed with (gray) and without (black) XTC.

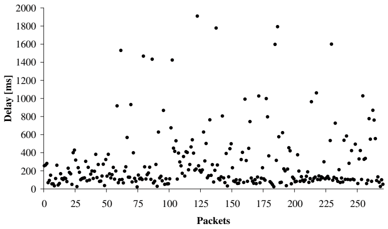


Fig. 6. Delay for successful message transmission across the small testbed *without* XTC.

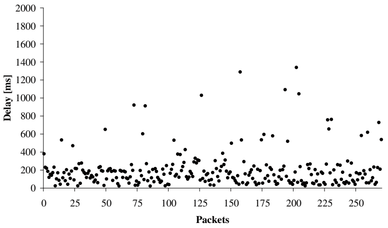


Fig. 7. Delay for successful message transmission across the small testbed *with* XTC.

choosing unreliable links. Consequently, source routing on XTC uses the path with minimal delay consisting of stable links. This is often achieved by replacing one unreliable link with multiple high-quality links. Hence, the measured route lengths on the XTC graph are generally longer than the ones chosen by pure source routing (see Figure 5(b)). However, for most receivers the chosen paths have similar lengths for pure source routing and the topology control assisted version. The maximum average route length difference is 0.9 hops for the path to node 6. For all other nodes the increase is less than 0.5 hops.

This small concession made to improve route stability pays off if the percentage of reusable routes is considered. With XTC route reuse is above 94% (cf. Figure 5(c)). Pure source routing reaches a reuse rate of 82%. This increased route stability implies that less of the expensive route discoveries were needed which in turn leads to an improved throughput. To evaluate this expected throughput increase we performed an additional experiment. Node 1 sent 300 messages as fast as possible—using end-to-end acknowledgments—to node 7 at the other end of the network. We allowed three retransmissions per packet followed by an unlimited number of route searches until node 7 could be reached again. We measured the total transmission time for each data packet including time spent on retransmissions and route discoveries.

Figures 6 and 7 show the measured delays for all packets with and without XTC, respecively. In both scenarios the majority of all packets could be sent using the cached route. These packets had a delay of 50 to 250 milliseconds depending on the number of necessary retransmissions. For some packets the cached route failed and thus, a new route discovery was initiated. Due to the various timeouts of the routing protocol their delay increased up to two seconds.

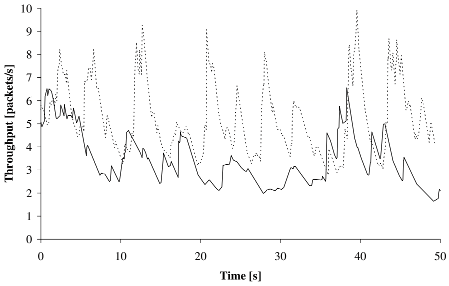


Fig. 8. Message throughput across the small testbed with (dotted) and without (solid) XTC.

Consequently, the average delay dropped from 290 ms to 200 ms if XTC was active. The measured throughput as depicted in Figure 8 was also generally higher if XTC was used for route discovery. On average it increased from 3.45 to 4.98 packets per second which corresponds to an improvement of more than 44%.

4.4 Office Floor Network

The promising results gained from the small testbed encouraged us to evaluate XTC in a larger network. We therefore distributed 33 nodes on an office floor (see Figure 9) and performed the same experiments as in the small testbed considered in Section 4.3. Since these experiments were executed during day time with numerous people working on the floor various real world effects such as moving obstacles and temporary interference with other wireless devices occurred. Such perturbations seem to have existed for example in the region of nodes 9, 10, and 28. We assume that the sources of this observed interference lie in the nearby elevators and an adjacent student lab.

Figure 10 shows that XTC manages to decrease transmission failures. On average the relative improvement is 11%. For the few nodes, such as 22, where XTC results in a decreased routing performance the additional penalty is below

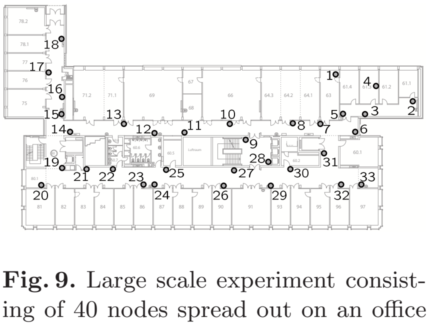


Fig. 9. Large scale experiment consisting of 40 nodes spread out on an office floor. Node 1 is the initiator of all data communication.

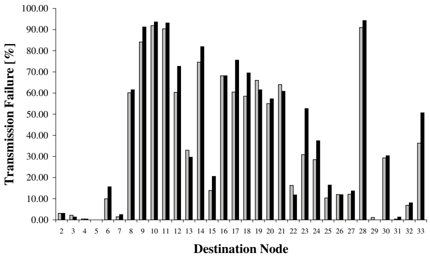


Fig. 10. Transmission failures in percent with (gray) and without (black) XTC.

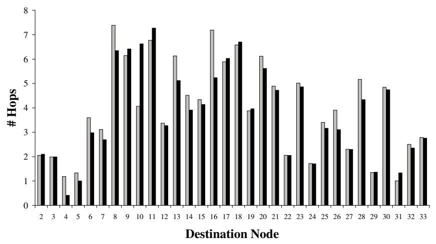


Fig. 11. Average route length with (gray) and without (black) XTC.

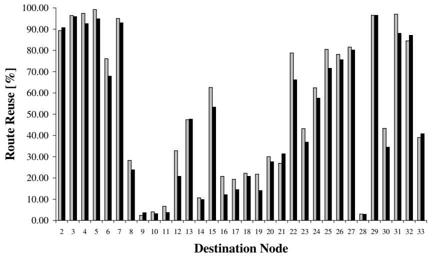


Fig. 12. Percentage of route reuse for communication with (gray) and without (black) XTC.

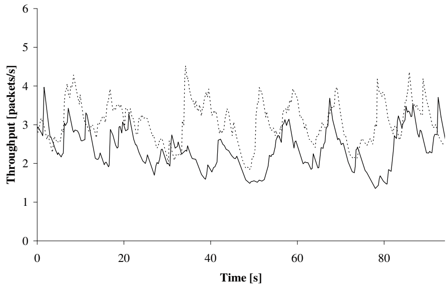


Fig. 13. Message throughput between node 1 and 23 in the office floor testbed with (dotted) and without (solid) XTC.

5%. Similarly, route reuse increased if XTC was active (cf. Figure 12). Figure 11 shows that the route length did not increase drastically with XTC. That is, the average hop count increases from 3.77 to 3.96. Analogous to the small testbed experiment we also evaluated throughput gains induced by XTC. Figure 13 exemplifies the benefit of XTC by showing the performance measurements from node 1 to 23. Without XTC the average throughput was 2.29 packets per second and increases to 3.02 with XTC. This is a relative improvement of roughly 32%. Summarized, the positive impact of XTC on source routing as seen in the small testbed also applies to this experiment. Especially, in the gray regions XTC proved its usefulness and lead to improved throughput.

5 Conclusions and Future Work

In this paper we have presented a practical implementation of an extended version of the XTC topology control algorithm which dynamically adapts to network changes. Using a packet loss based metric unreliable links are identified and excluded while connectivity is maintained. The beneficial effect of XTC on source routing—a common application in wireless networks—was first shown in a secluded environment using a small number of nodes. We then verified the obtained results in a real world scenario. Also in this environment XTC performed well reducing packet loss and delay and thus leading to improved network throughput.

So far, the usefulness of XTC is only evaluated in non-mobile wireless networks. We believe XTC will also show its advantages in networks with mobile nodes. In such environments link qualities change more frequently and thus it is important to select reliable and long-living connections. Adapting XTC's link quality metric to incorporate link stability may be required. Another interesting field of application are networks with high density. Due to the large amount of potential paths from one node to any other there is a high probability of choosing a route containing an unreliable link. Consequently, the ability of XTC to exclude such links is important. Finally, the impact of XTC on other applications than source routing is worth studying. For example, alarm systems with a low tolerance toward communication failures or other data gathering applications may also benefit from the XTC protocol.

References

1. Hu, L.: Topology Control for Multihop Packet Radio Networks. IEEE Trans. on Communications **41**(10) (1993)
2. Wattenhofer, R., Li, L., Bahl, P., Wang, Y.M.: Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In: Proc. of the *20th* Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2001)
3. Li, X.Y., Calinescu, G., Wan, P.J.: Distributed Construction of Planar Spanner and Routing for Ad Hoc Wireless Networks. In: Proc. of the *21st* Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2002)
4. Li, N., Hou, C.J., Sha, L.: Design and Analysis of an MST-Based Topology Control Algorithm. In: Proc. of the *22nd* Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2003)
5. Li, X.Y., Song, W.Z., Wang, W.: A Unified Energy Efficient Topology for Unicast and Broadcast. In: Proc. of the *11th* Annual International Conference on Mobile Computing and Networking (MOBICOM). (2005)
6. Chin, K.W., Judge, J., Williams, A., Kermode, R.: Implementation experience with manet routing protocols. ACM SIGCOMM Computer Communications Review **32**(5) (2002) 49–59
7. Dube, R., Rais, C.D., Wang, K.Y., Tripathi, S.K.: Signal Stability-Based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks. IEEE Personal Communications (1997) 36–45
8. Goff, T., Abu-Ghazaleh, N.B., Phatak, D.S., Kahvecioglu, R.: Preemptive Routing in Ad Hoc Networks. In: Proc. of the *7th* Annual International Conference on Mobile Computing and Networking (MOBICOM). (2001)
9. Hu, Y., Johnson, D.: Design and Demonstration of Live Audio and Video over Multihop Wireless Ad Hoc Networks. In: Proc. of the Military Communication Conference (MILCOM). (2002)
10. Qin, L., Kunz, T.: Increasing Packet Delivery Ratio in DSR by Link Prediction. In: Proc. of the *36th* Annual Hawaii International Conference on System Sciences (HICSS). (2003)
11. Wattenhofer, R., Zollinger, A.: XTC: A Practical Topology Control Algorithm for Ad-Hoc Networks. In: Proc. of the *4th* International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN). (2004)

12. Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., Wicker, S.: Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor. Technical Report 02-0013, UCLA CS (2002)
13. Zhao, J., Govindan, R.: Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. In: Proc. of the 1^{st} ACM Confernce on Embedded Networked Sensor Systems (SenSys). (2003)
14. Moaveni-Nejad, K., Li, X.Y.: Low-Interference Topology Control for Wireless Ad Hoc Networks. In: Proc. of the 2^{nd} Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, California, USA (2005)
15. Blough, D., Leoncini, M., Resta, G., Santi, P.: The K-Neigh Protocol for Symmetric Topology in Ad Hoc Networks. In: Proc. of the 4^{th} ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC). (2003)
16. Ghosh, S., Lillis, K., Pandit, S., Pemmaraju, S.V.: Robust Topology Control Protocols. In: Proc. of the 8^{th} International Conference on Principles of Distributed Systems (OPODIS). (2004)
17. Lillis, K., Pemmaraju, S.V.: Topology Control with Limited Geometric Information. In: Proc. of the 9^{th} International Conference on Principles of Distributed Systems (OPODIS). (2005)
18. Dyer, M.: S-XTC: A Signal-Strength Based Topology Control Algorithm. TIK Report 235, ETH Zurich, Switzerland (2005)
19. Son, D., Krishnamachari, B., Heidemann, J.: Experimental study of the effects of Transmission Power Control and Blacklisting in Wireless Sensor Networks. In: Proc. of the 1^{st} IEEE Conference on Sensor and Ad Hoc Communication and Networks (SECON). (2004)
20. Qin, L., Kunz, T.: Pro-active route maintenance in dsr. ACM SIGMOBILE Mobile Computing Communications Review **6**(3) (2002) 79–89
21. Awerbuch, B., Holmer, D., Rubens, H.: High Throughput Route Selection in Multi-Rate Ad Hoc Wireless Networks. In: Proc. of the 1^{st} Annual Conference on Wireless On-Demand Network Systems (WONS). (2003)
22. Couto, D.S.J.D., Aguayo, D., Bicket, J., Morris, R.: A High-Throughput Path Metric for Multi-Hop Wireless Routing. In: Proc. of the 9^{th} Annual International Conference on Mobile Computing and Networking (MOBICOM). (2003)
23. Burri, N., von Rickenbach, P., Wattenhofer, R., Weber, Y.: A Practical Evaluation of the XTC Algorithm. TIK Report 263, ETH Zurich, Switzerland (2006)