Corridor Routing in Mobile Ad-hoc Networks

Christian Vogt
Institute of Telematics
University of Karlsruhe
chvogt@tm.uka.de

Michael Gerharz · Christian de Waal
Institute of Computer Science IV
University of Bonn
{gerharz,dewaal}@cs.uni-bonn.de

Abstract

A new reactive strategy for multi-path routing in mobile ad-hoc networks is presented. Given a pair of communicating stations, we use the complete set of minimal-length paths – a so-called corridor – between the two. The potentially high number of paths within a corridor is efficiently installed by virtue of a new destination-discovery mechanism that is exclusively based on broadcast transmissions. Data streams are split and distributed over an entire corridor to exploit available network resources.

1 Introduction

Ad-hoc networks are wireless multi-hop networks that are independent of any kind of infrastructure. They are thus predestinated for spontaneous employment wherever circumstances prohibit using conventional communications means – be it at exhibitions, at conferences, or during military engagements. In ad-hoc networks, all stations cooperatively serve as routers. The theory of ad-hoc networks has been shaped over the recent years, and a large body of effort has since been dedicated to help ad-hoc networks become a realistic option.

One could certainly anticipate email, Web browsing, file transfer, and other conventional Internet applications to be used in ad-hoc environments. Yet, many people desire more complex technology like video conferencing or telephony. Such real-time applications bring about rigid requirements in terms of maximum-delay, maximum-jitter, and minimum-bandwidth bounds the provision of which we call *quality of service* (QOS). QOS is difficult to realize in ad-hoc networks due to ceaseless topology changes and the wireless medium's sparse bandwidth.

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) as part of the IPonAir project (http://www.iponair.de/).

One differentiates proactive and reactive routing protocols for ad-hoc networks. *Proactive routing protocols* constantly examine a network's topology for broken old and emerging new links. *Reactive routing protocols* limit network-topology exploration to when and where user data is available for transportation. Proactive routing protocols have the advantage that, when a new path is needed for communication, they usually can provide one without delay. Reactive routing protocols lose time as they search for a path in an on-demand manner. On the other hand, proactive routing protocols produce control traffic even in the absence of user-data traffic, whereas the overhead generated by reactive routing protocols scales with the number of active communication sessions. This way, reactive routing protocols help to conserve power when no communication is ongoing. We focus on reactive routing protocols in this paper.

A reactive routing protocol's procedure of searching for a new path is called *destination discovery.* A destination discovery typically comes in two phases. During the first phase, a *request message* (REQUEST) is emitted by the searching station and flooded across the network. The REQUEST identifies a path from the searching station to the requested destination when it reaches the destination. The destination commences the second phase by sending back to the searching station a *reply message* (REPLY) that contains the discovered path. Destination discoveries are responsible for reduced QOS. They generate extra-ordinary bursts of control traffic, which in turn heighten protocol latency and occupy bandwidth that could otherwise be spent on user-data traffic. Destination discoveries are thus an important issue with reactive routing protocols, and a primary protocol-design goal should be to repress as many as possible.

In many classic reactive routing protocols, a communication session between two stations exclusively depends on a single path. When one link on this path breaks, the path is typically rendered unusable in its entirety and needs to be replaced by a new one.

These *single-path routing protocols* are little robust to network-topology fluctuations and accordingly inclined to recurrent destination discoveries. Furthermore, single-path routing protocols are subject to local congestion when the full workload of multiple data streams accumulates at a bottleneck router.

A more efficient routing approach is to maintain a set of multiple paths between the same end stations. Then, as the network topology changes, and an active path becomes unusable, backup paths are readily at hand. Datagram relay can thus continue without disruption. These *multi-path routing protocols* also allow datagrams to be transported along different paths such that traffic bottlenecks can be avoided. Moreover, traffic bottlenecks are likely to be *prevented* when data is dispersed across a wide network region.

We observe three impediments to routing performance in existing reactive multi-path routing protocols. First, many multi-path routing protocols unicast a separate REPLY along each new path during destination discovery. This mechanism is inherited from single-path routing protocols and sparks excessive control traffic when conveyed to multi-path routing protocols. In order to curb such control traffic, a widely applied approach is to limit the number of paths that can be acquired during one destination discovery. Second, many multi-path routing protocols accept longer-than-optimal paths. Those paths are responsible for unnecessarily many datagram transmissions and squandered bandwidth. Furthermore, a path's risk-to-failure increases substantially with its length. A long path has hence a much shorter expected lifetime than a short one and implies an earlier destination rediscovery. Third, many multi-path approaches require paths between a pair of communicating stations to be router- or link-disjoint. Links on router-disjoint paths are independent such that a single station's movement cannot impact a communication session twice (cf. section 4.3.4). However, ad-hoc networks are usually geographically dense such that router-disjoint paths are difficult to find. Link disjointness comes as a relaxation of router disjointness, yet fails to provide the desired link independence.

With this paper, we contribute a new reactive multi-path routing strategy which we call *corridor routing.* We start out with a discussion on existing multi-path routing protocols in section 2. Corridor routing deviates from those approaches in key design principles. We explain the design characteristics and protocol procedures of corridor routing in section 3. In section 4, we analyze the performance of corridor routing with respect to two well-known single-path routing protocols. Concluding remarks are made in section 5.

2 Related Work

In the prospect of higher robustness to network-topology changes and augmented QOS provision, several contributions towards multi-path routing in ad-hoc networks have been made in recent research. Gerla et al. propose *Split Multipath Routing* (SMR), a reactive multi-path routing protocol based on the well-known Dynamic Source Routing (DSR) single-path routing protocol [6]. SMR takes a short-delay path as a base and computes a set of paths maximally router-disjoint from that base path.

Router disjointness brings about two benefits. One is higher robustness to network-topology changes. If the paths belonging to the same communication session are mutually router-disjoint, a single station's movement cannot disrupt that session more than once (cf. section 4.3.4). This limits the impact a network-topology change may have and reduces the number of related datagrams losses.

Increased bandwidth can be a second benefit of router disjointness. It accrues in wired or multi-channel wireless networks when datagrams are simultaneously forwarded on several router-disjoint paths. In single-channel wireless networks, however, this is generally not the case. First, there are two important bandwidth bottlenecks at the paths' common end points. Second, certain network constellations may prohibit independent datagram relay even at intermediate routers. This is due to what Perlman et al. call *coupling* [8]: Two paths are coupled when a station on one path competes with a station on the other path for the same resources. In wired or multi-channel wireless networks, this is only the case when the two paths intersect. In single-channel wireless networks, path coupling in addition occurs when routers on different paths are within radio range. IEEE-802.11 networks, for instance, are single-channel, and interferences accordingly curtail the asset of router disjointness. The entanglement of paths is significant in dense ad-hoc networks. It limits the effect of traffic dissipation and bandwidth exploitation, as few paths between common end points can actually be regarded decoupled. We henceforth ignore the potential for increased bandwidth of router-disjoint paths.

By using a short-delay path as a base path, SMR effectively reduces destination-discovery latency. However, the latency reduction comes at the high price

of potentially having longer-than-optimal paths, which have a two-fold disadvantage: One is that the additional links lead to a higher number of datagram transmissions, consuming valuable bandwidth and increasing the data's delivery delay. Another disadvantage is that a path's risk-to-failure grows with its length. A long path is less stable than a short one and entails an earlier destination rediscovery.

The *Ad-hoc On-demand Multi-path Distance Vector* (AOMDV) routing protocol extends the well-known Ad-hoc On-demand Distance Vector (AODV) routing protocol by multi-path capabilities [7]. AOMDV inherits the sequence-number mechanism from AODV. The original mechanism is slightly adjusted to fit the multi-path concept. Like SMR, AOMDV uses paths of potentially suboptimal length. Unlike SMR, AOMDV replaces router disjointness by link disjointness. Link-disjoint paths are easier to find than router-disjoint ones. However, requiring paths to be link-disjoint has no effect on robustness to network-topology changes. Instead, two link-disjoint paths may well cross at one or more routers. This calls into question whether link disjointness is a worthwhile restriction.

When launching a new communication session, reactive single-path routing protocols unicast a REPLY along the one path to be used for that session. SMR and AOMDV take up this principle and unicast one REPLY along *each* of usually multiple paths to be used for the new session. This may lead to excessive REPLY generations if the number of available paths is high. To avoid such REPLY storms, SMR and AOMDV limit the number of paths that can be established throughout one destination discovery. This strategy obviously reduces a multi-path routing protocol's efficiency.

3 Corridor Routing

Driven by the improvement opportunities identified in section 2, we propose a new reactive multi-path routing strategy, which we call *corridor routing*. The notion of a corridor illustrates the collection of paths that form a communication session. A *corridor* between a pair of communicating stations is the set of links that belong to a minimum-length path connecting those stations.

In this section, we summarize the design characteristics of corridor routing and explain the procedures for corridor establishment, datagram relay, and corridor teardown.

3.1 Design Characteristics

Corridor routing is a reactive approach: A corridor is established only if a station actually wishes to send a datagram to another station to which no path is known.

Rather than a set of paths, a corridor ought to be regarded as a set of links. A *path* in the corridor is a directed sequence of adjacent links from the corridor connecting the two end stations. Corridor routing exclusively uses paths of minimum length. This way, no bandwidth is squandered by unnecessary datagram transmissions. Moreover, since a short path is more stable against network-topology changes than a long one, corridor routing requires less destination discoveries than would otherwise be required. Corridor routing uses the set of *all* minimum-length paths available between two communicating stations. In particular, corridor routing does not restrict paths to be router- or link-disjoint. When a link on an active path fails, the functioning links on that path continue to be used as long as they can be weaved into a different minimum-length path. Corridor routing thus operates more economic than disjointness-oriented approaches, which down a complete path upon a link break.

Corridor routing maintains network-topology information in a distributed manner: Given a destination, *D*, each router in the corridor ending at *D* keeps a list of next-hop neighbors to which datagrams addressed to *D* can be forwarded. For any particular datagram, the router selects a next-hop neighbor from the list based on a certain traffic-distribution algorithm. Since all paths are of minimum length, routing loops do not occur.

3.2 Corridor Establishment

When a station, *S*, wishes to send a datagram to another station, *D*, to which no path is known, *S* initiates a destination discovery for *D*. The potentially large number of paths within a corridor is efficiently set up by a new destination-discovery concept. While reactive single-path routing protocols in general as well as the multi-path routing protocols introduced in section 2 generate a separate unicast REPLY for each new path during destination discovery, a corridor comes into being much more economically by virtue of *broadcast* REPLIES. The REPLIES adhere to the shortest paths between *S* and *D*.

3.2.1 REQUEST Phase

When generating a REQUEST, *S* includes in the message its own and *D*'s addresses. The REQUEST has a *hop-count field* which is zeroed by *S* and incremented by one after each hop the message takes. This way, the hop count equals, at any time, the REQUEST's distance from *S*.

A REQUEST is flooded network-wide. A router, *X*, may thus receive REQUESTS pertaining to the same destination discovery from different neighbors. *X* determines its distance from *S* as the minimum hop count of all received REQUESTS. The neighbors from which *X* receives a REQUEST with minimum hop count form *X*'s set of upstream neighbors with regard to *S*. *X* may memorize its upstream neighbors if bidirectional corridors are desired. Otherwise, if unidirectional communication is sufficient, *X* does not need to keep its upstream neighbors.

Initially, *X*'s distance from *S* is considered infinity (∞). Whenever *X* receives a REQUEST generated by *S* from another station, *Y*, *X* verifies its current distance estimation. If the incoming REQUEST's hop count is smaller than *X*'s estimated distance from *S*, *X* sets its distance from *S* to the hop count included in the REQUEST. Furthermore, if bidirectional corridors are desired, *X* deletes all entries from its list of upstream neighbors with regard to *S* and includes *Y* into the now-empty list. *X* propagates the REQUEST, if *X* is a station different from *D*. If the incoming REQUEST's hop count equals *X*'s estimated distance from *S*, *X* adds *Y* to its list of upstream neighbors with regard to *S* if bidirectional corridors are desired. If the incoming REQUEST's hop count is greater than *X*'s estimated distance from *S*, *X* silently discards the message.

By the time the REQUEST phase concludes, all stations in the network should have an accurate estimation of their distance from *S*.

3.2.2 REPLY Phase

Like any intermediate router, *D* determines its distance from *S* as the minimum hop count of multiple received REQUESTS. Hence, when *D* receives the first REQUEST of which the targeted destination it is, *D* defers its REPLY for a while during which additional REQUESTS are expected to arrive. When *D* eventually generates the REPLY, *D* includes in the message its own and *S*'s address. In the REPLY's *corridor-length field*, *D* advertises its estimated distance from *S*. Finally, the REPLY has a *hop-count field* which is zeroed by *D* and incremented by one after each transmission. The hop count thus equals the REPLY's distance from *D* at any time.

A REPLY is relayed along all minimum-length paths between *S* and *D*. When an intermediate router, *X*, receives the REPLY from a neighbor, *Y*, *X* can determine from the message's fields both its distance from *D* as well as its supposed distance from *S*. The former is directly given by the REPLY's hop count, whereas the latter can be calculated by subtracting the hop count from the posted corridor length. *X* compares its distance from *S* estimated during the discovery's REQUEST phase with the supposed distance. If the estimated distance is greater than what the distance is supposed to be, *X* must silently discard the REPLY, because *X* does not lie on a shortest *S*-*D* path and hence not within the *S*-*D* corridor. If the estimated distance equals the supposed distance, *X* lies within the *S*-*D* corridor. In this case, *X* adds *Y* to its list of downstream neighbors with regard to *D*. *X* propagates the REQUEST if *X* is a station different from *S*. If *X* equals *S*, *S* should directly start to send out datagrams targeted to *D*, albeit further REPLIES are expected to come in via different neighbors. Due to contingencies during the destination discovery's REQUEST phase, *D* might not have received a REQUEST over an actually shortest *S*-*D* path. *X*'s estimated distance from *S* may then be smaller than the supposed distance. If so, *X* proceeds as if the two values were equal.

By the time the REPLY phase concludes, each router on a minimum-length *S*-*D* path should have a list of downstream neighbors with regard to *D* and, if bidirectional corridors are desired, a list of upstream neighbors with regard to *S*. Timer mechanisms ought to ensure that stations which do not lie on a shortest *S*-*D* path remove the information stored during the REQUEST phase after appropriate time.

3.3 Datagram Relay

When a router wishes to relay a datagram, the router selects one entry from its list of next-hop neighbors with regard to the datagram's destination and forwards the datagram to the selected neighbor.

Several neighbor-selection algorithms are conceivable. We use a simple round-robin scheme. This method provides efficient and easy-to-implement traffic distribution. More sophisticated algorithms may take into account a neighbor's current workload. While round robin attempts to spread datagrams over a corridor in a well-balanced fashion, a workload-based approach could cause datagrams circumvent traffic bottlenecks and equilibrate network utilization.

3.4 Corridor Teardown

When a router, *X*, attempts to forward a datagram to a next-hop neighbor, *Y*, it may turn out that *Y* has moved out of *X*'s radio range such that the transmission fails. In this case, *X* removes *Y* from *all* its lists of next-hop neighbors in which *Y* shows up. If available, *X* may further choose an alternative next-hop neighbor to which to try and forward the datagram that could not be transmitted to *Y*.

When *X* desires to forward a datagram, but *X* does not know an appropriate next-hop neighbor, *X* broadcasts an error report (ERROR). *X* includes in the ERROR the address of itself and the datagram's destination, *D*. The ERROR is targeted at all neighbors of *X* which maintain a list of next-hop neighbors with regard to *D* that includes *X*. Let *Y* be one of those neighbors. When *Y* receives the ERROR, *Y* removes *X* from its list of next-hop neighbors with regard to *D*. If *X* is the only record in that list, *Y* itself generates an ERROR. Moreover, should *Y* take an interest in a communication session with *D*, *Y* may choose to initiate a new destination discovery for *D*. ERRORS are broadcasted as they usually have multiple recipients.

4 Performance Evaluation

We have implemented the concept of corridor routing and evaluated its performance with respect to DSR and AODV using NS-2 simulations [2]. We henceforth refer to our implementation as the *Corridor Routing Protocol* (CRP). The overall goal of our studies has been to identify CRP's, DSR's, and AODV's capabilities to provide QOS in the face of network-topology fluctuations. In this paper, we focus on the employment of real-time applications with special emphasis on voice over IP (VOIP). Real-time applications in general are characterized by high sensitivity to data delay. VOIP in particular is a real-time application that produces a steady data stream. It requires moderate but constant bandwidth.

We have examined CRP, DSR, and AODV under a wide range of conditions in order to arrive at the presented performance results. Section 4.1 describes our simulation environment. Section 4.2 summarizes the performance metrics in terms of which we have evaluated the protocols. Section 4.3 analyzes the measurements obtained from the simulations.

4.1 Simulation Environment

The NS-2 simulator models the physical characteristics of wireless networks and provides support for simulating the medium-access-control (MAC) protocols required in such networks. Moreover, NS-2 allows for different station-movement and traffic patterns. In this section, we describe the scenario parameters that apply to our simulations.

4.1.1 Physical and Data-Link Model

At data-link level, we use the IEEE-802.11 Distributed Control Function (DCF) [9]. Mobile stations are equipped with single-channel radios with communication ranges of approximately 50 meters. Stations within communication range share a nominal bandwidth of 2 Mbps. DCF applies physical carrier sense to reduce the probability of transmission collisions. With physical carrier sense, a station willing to send a datagram listens for other stations using the medium at that time. If the medium is idle, the station may transmit. If the medium is busy, the transmission is deferred, and a randomized back-off reduces the likelihood that two or more stations simultaneously attempt to use the medium once it is idle again.

Physical carrier sense assumes that all stations can hear each other. For various reasons, this is not always the case [5]. DCF hence offers an optional virtual-carrier-sense protocol for unicast transmissions. With virtual carrier sense, each unicast transmission is preceded by a brief message exchange between the sender and the receiver to reserve the medium in both stations' vicinities for the duration of the data transmission. We use virtual carrier sense in our simulations.

Correct unicast-datagram reception is approved by an acknowledgement to the sender. The sender continues to repeat transmitting the datagram for up to a certain number of times until it receives an acknowledgement. We use a maximum of seven retransmission attempts in our simulations. Lack of reception of an expected acknowledgement does not necessarily imply that the datagrams has not been correctly delivered. It may likewise indicate an error to the acknowledgement transmission.

4.1.2 Network Topology and Movement Model

We have simulated a network of 50 mobile stations moving about on a flat rectangular field. The roaming area is 300 meters long and 60 meters wide. The stations form a single network partition at all times. Their

movement behavior adheres to the *Random Waypoint model* [4]: At the beginning of a simulation, each station chooses a position on the movement field where it starts its journey. There, the station pauses for a while. When the pause time elapses, the station chooses a new location and a movement speed with which to approach that location. The station moves on a straight line. Upon arrival, the station pauses again, and the procedure repeats itself.

The movement-area positions are determined by randomly and uniformly selecting *x* and *y* coordinates from the available dimensions. The speed and pause time are also randomly and uniformly chosen. We use average speeds of 2, 3, 4, 5, 6, and 7 m/s with pause-time averages of 150, 125, 100, 75, 50, and 25 seconds, respectively. These movement parameters are primarily intended to reflect people's behavior when walking on foot at exhibitions or conferences. In particular, the chosen speeds embrace a range one would consider moderate to hasty strolling speeds. The pause times are supposed to accommodate the behavior of exhibition visitors stopping by at one booth or other, or conference attendees getting involved in a short conversation with colleagues.

4.1.3 Traffic Model

We have conducted simulations with offered workloads of 1, 2, 3, 4, 5, and 6 parallel communication sessions. With regard to the character of VoIP applications, each session is realized by a bidirectional constant-bit-rate connection of 60 seconds length. Both the originator and the callee produce a net data rate of 12.2 kbps. The net data rate may, for example, be generated by an AMR speech codec without voice-activity detector (VAD) [1]. Data streams are segmented into datagrams of 193 bytes issued at a rate of 10 pps. Each datagram includes 40 bytes of RTP, UDP, and IP headers.

We found that randomly distributing communication sessions over the available simulation time resulted in very irregular network traffic with unintentional peaks and lows. In order to provide a basis for a more transparent analysis, we decided to homogenize the workload. In particular, we keep the number of simultaneous communication sessions constant by launching a new session whenever an old one is terminated.

4.2 Performance Metrics

We have evaluated CRP's performance with respect to DSR and AODV in terms of the following five metrics.

- *Datagram-delivery ratio:* The number of application-generated datagrams which the routing protocol successfully and timely delivers to the addressee divided by the total number of application-generated datagrams.
- *Datagram-delivery delay:* The time period during which a datagram is being relayed through the network.
- *Buffer-overflow ratio:* The number of application-generated datagrams lost at a router without sufficient buffering capacity divided by the total number of application-generated datagrams.
- *Routing-failure ratio:* The number of application-generated datagrams lost because of routing failures divided by the total number of application-generated datagrams.
- *Destination-discovery frequency:* The number of destination discoveries pursued throughout the course of one 60-seconds communication session.

The forthcoming discussion is based on the arithmetic means of the three routing protocols' datagram-delivery ratios, buffer-overflow ratios, routing-failure ratios, and destination-discovery frequencies, as well as the 90^{th} percentiles of their datagram-delivery delays. The arithmetic means are depicted along with their 95%-confidence intervals.

4.3 Simulation Results

In this section, we analyze the results obtained from the simulations described in section 4.1 and deduce the performance of CRP, DSR, and AODV using the five metrics defined in section 4.2.

4.3.1 Datagram-Delivery Ratio

A routing protocol's overall performance can be described in terms of its datagram-delivery ratio. The datagram-delivery ratio is the fraction of those datagrams generated by the sending applications that

can be turned to account by the receiving application. What determines a datagram's appropriateness is highly application-specific. Oftentimes, a datagram being usable is equated with a datagram being delivered to the addressee. This definition is adequate to ordinary applications like email, Web browsing, or file transfer, for which buffer overflows or routing failures are the only causes for datagram loss. However, delay-sensitive real-time applications in addition do not accept datagrams older than a certain age. Since we concentrate on VOIP in this paper, we redefine the datagram-delivery ratio to be the fraction of all application-generated datagrams which the routing protocol successfully *and timely* delivers to the addressee.

ITU experiments show that significant degradations in conversation quality are perceived if the time lag between speech recording and playback exceeds 250 ms [3]. The time lag is caused by speech-data compression, datagram assembly, the datagram's propagation through the network, and speech-data decompression. With a datagram-sending rate of 10 pps (cf. section 4.1.3), assembling a datagram takes 100 ms. We reserve an additional 50 ms for speech-data compression and decompression at the communication end sides. In order not to exceed a total time lag of 250 ms, datagram-delivery delay should not go beyond 250 ms - 100 ms - 50 ms = 100 ms. At routing level, we hence consider stale and discard all datagrams older than 100 ms.

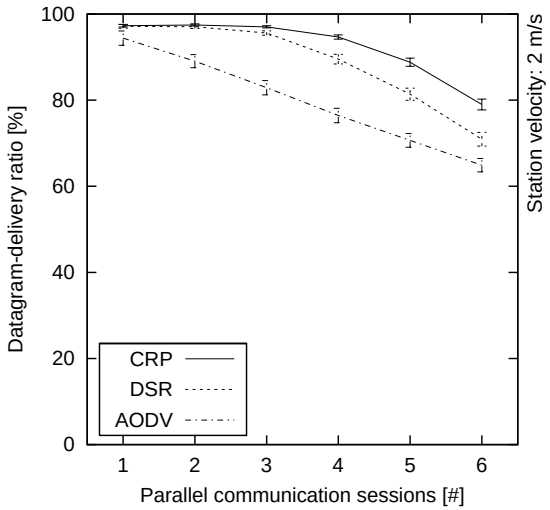

Figure 1. Datagram-delivery ratios as a function of the number of parallel communication sessions.

Figure 1 shows CRP's, DSR's, and AODV's mean datagram-delivery ratios as a function of the number of parallel communication sessions. Stations move at an average velocity of 2 m/s. We observe that CRP is more stable than DSR and AODV to an increase in offered workload: CRP's datagram-delivery ratio remains constant with one, two, and three parallel communication sessions and decreases to an only negligible degree with four. The decrease accelerates as the number of parallel sessions grows further. DSR reacts similar to CRP in that its datagram-delivery ratio is almost stable when traffic is low, but shrinks faster as more communication sessions join. AODV's datagram-delivery ratio appears to be rather linearly dependent on the number of parallel communication session, shrinking by about 6 percent for each additional one. The different tendencies are caused by the protocols' individual datagram-delivery delays (cf. section 4.3.2) and buffer-overflow ratios (cf. section 4.3.3).

We explain the performance lead of CRP as follows: Corridor routing distributes data streams over multiple paths. This allows for higher bandwidth exploitation and reduces the probability of traffic bottlenecks. Should a link failure or traffic bottleneck occur, its impact is generally limited as few datagrams take identical paths.

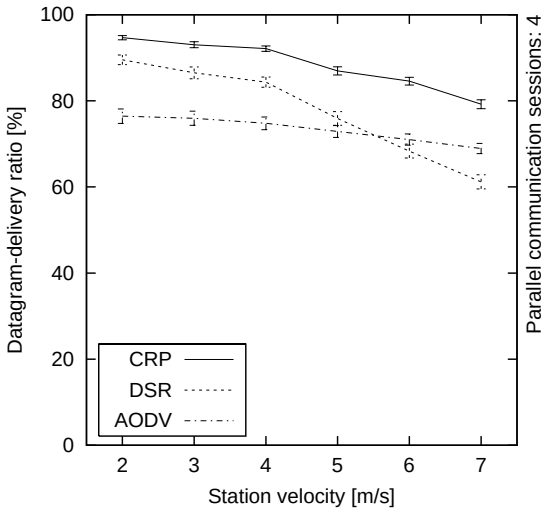

Figure 2. Datagram-delivery ratios as a function of the average station velocity.

Figure 2 shows the three protocols' mean datagram-delivery ratios as a function of the average station velocity. The number of parallel communication sessions is now fixed at four. Again, CRP proves to be most

robust to unfavorable circumstances. Its datagram-delivery ratio is well above DSR's and AODV's. We observe that DSR's performance deteriorates as station velocities exceed 4 m/s. This is because DSR relies on cached routing information, which oftentimes is invalid in high-mobility scenarios. With CRP, routing paths are constantly in use such that link failures can be detected early.

4.3.2 Datagram-Delivery Delay

The importance of timely data delivery in the context of delay-sensitive real-time applications motivates taking a look at CRP's, DSR's, and AODV's datagram-delivery delays. A datagram's delivery delay is the time period between the originating station emits the datagram's first bit and the addressee receives the datagram's last bit. We analyze the routing protocols' datagram-delivery delays with respect to the offered workload and station mobility. Figures 3 and 4 show the respective measurements in terms of their 90th percentiles. An average strolling speed of 2 m/s is used in the former case, a constant offered workload of four parallel communication sessions in the latter.

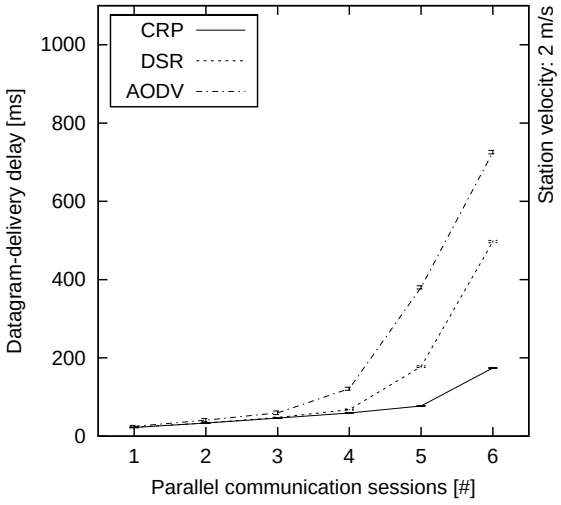

Figure 3. Datagram-delivery delays (90th percentiles) as a function of the number of parallel communication sessions.

We observe from figures 3 and 4 that DSR's and AODV's datagram-delivery delays are multiples of CRP's regardless of offered workload or station velocity. Evidently, DSR and AODV need, on average, substantially more time to deliver a datagram than CRP does. The celerity gap widenes with increasing workload. Furthermore, DSR and AODV react rather sensitive to an increase in station velocities, whereas CRP works fine at any of the examined mobility levels.

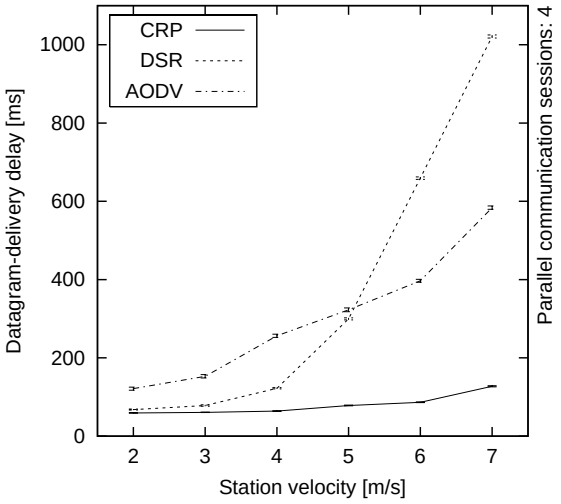

Figure 4. Datagram-delivery delays (90th percentiles) as a function of the average station velocity.

According to the ITU experiments cited in section 4.3.1, speech-transmission times should not exceed 250 ms [3]. Hence, datagram-delivery delays should not go beyond 100 ms. According to this guideline, AODV allows for up to only three parallel communication sessions. DSR performs better and manages four simultaneous sessions. With CRP, quality degradations are perceived only beyond five parallel communication sessions. CRP also works fine at any of the examined mobility levels, whereas DSR and AODV react rather sensitive to an increase in station velocities. In particular, DSR suffers from cached routing information invalidated by frequent network-topology changes, the futile application of which is responsible for prolonged datagram-delivery delays.

Corridor routing tackles the issue of high datagram-propagation delay from two directions. First, the distribution of user-data traffic lowers the probability of traffic bottlenecks and curtails the associated delay. Second, protocol-control traffic is reduced when a broken path can be substituted by an available backup path without pursuing a new destination discovery.

4.3.3 Buffer-Overflow Ratio

Buffer overflows are the result of local traffic peaks. They occur when a router cannot handle the workload it is confronted with. Such workload may originate from locally accumulating user-data traffic or from protocol-control messages caused by destination discoveries. The buffer-overflow ratio thus provides a means to determine a routing protocol's inclination towards, or its capability to avoid, local congestion. Since all datagrams affected by buffer overflows are lost, the buffer-overflow ratio is an important determinant of a routing protocol's datagram-delivery ratio. Figure 5 shows the mean buffer-overflow ratios of CRP, DSR, and AODV as a function of the number of parallel communication sessions. Stations move at 2 m/s on average.

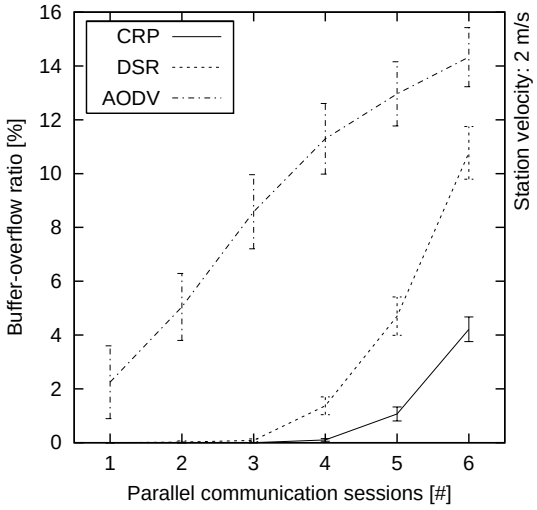

Figure 5. Buffer-overflow ratios as a function of the number of parallel communication sessions.

DSR and AODV are single-path protocols. The buffering capacity available to a communication session is thus limited to what the routers on one path can provide. CRP performs better than DSR and AODV because it distributes each user-data stream into multiple paths and exploits the accumulated buffering capacity of an accordingly higher number of routers. Figure 5 provides insight into how efficient the workload distribution of corridor routing is. With low to moderate traffic, splitting data streams exploits the bandwidth and buffering capacity in network regions where routers would otherwise be idle. Buffer overflows are thus highly exceptional with CRP at these traffic levels. Workload distribution looses impact as traffic grows and the medium becomes occupied to capacity in all parts of the network.

Figure 5 evidences a strong sensitivity to high workload of all three routing protocols. The reason is that part of the routers is overwhelmed with the accumulated data volume of multiple intersecting routing paths when the number of ongoing communication sessions is high. The narrow, oblong shape of the station's movement area further encourages the formation of traffic bottlenecks. When a station transmits, its 50-meters radio range may cover an entire slice of the area such that all cross traffic is blocked. CRP and DSR exclusively use routing paths of minimum length. They thus keep the number of required transmissions as low as possible and mitigate the issue of contention. AODV does not have this property.

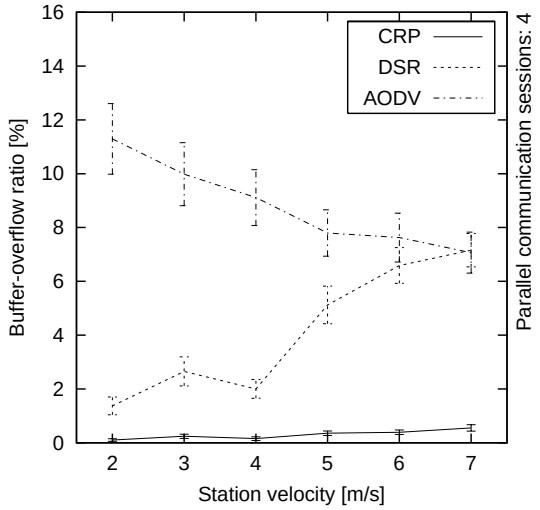

Figure 6. Buffer-overflow ratios as a function of the average station velocity.

Figure 6 plots the protocols' mean buffer-overflow ratios as a function of the average station velocity. The workload is fixed at four parallel communication sessions. It is conspicuous that AODV's ratio shrinks as the mobility increases. The reason is that AODV immediately drops a datagram when the datagram cannot be forwarded because of a link failure. If stations move fast, and the network topology changes swiftly, frequent link failures cause a large quantity of datagrams to be abandoned. Obviously, buffering capacity is spared whenever a datagram is thrown away. In contrast, DSR seeks to salvage each datagram that cannot be routed along the primary path by using a cached alternative. Though it may eventually turn out that the

cached path does no longer exist, the datagram waiting to be salvaged potentially occupies valuable buffer space.

4.3.4 Routing-Failure Ratio

The routing-failure ratio is another determinant of a routing protocol's datagram-delivery ratio besides the buffer-overflow ratio. A routing failure is the event in which a datagram is lost because of a broken link. Link breaks, in turn, are the result of network-topology changes. Obviously, when a single station moves, all links adjacent to that station are subject to breakage. We call the set of links that share a common end station *dependent.* Figure 7 shows an example scenario with two dependent links, *A*-*E* and *B*-*E*. Both of them fail as station *E* moves. Link dependencies may be more complex, involving an arbitrary number of links.

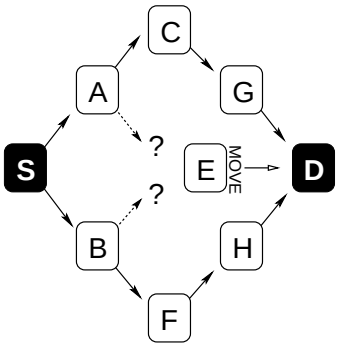

Figure 7. Corridor between stations *S* and *D*. Router *E* moves out of *A*'s and *B*'s radio range and causes two related link failures.

In single-path routing protocols, a station's movement may break several dependent links on paths belonging to different communication sessions. In multi-path routing protocols, a station's movement may in addition break dependent links on paths belonging to the same session if no countermeasures are taken. Some multi-path routing protocols exclude link dependencies on paths belonging to the same communication session by requiring those paths to be router-disjoint (cf. section 2). With corridor routing, paths do not need to be disjoint even if they belong to the same session. On one hand, this approach allows to flexibly respond to link failures as described in section 3.1. On the other hand, link dependencies may lead to an increased number of routing failures and lost datagrams.

Figure 8 shows the mean routing-failure ratio of CRP, DSR, and AODV as a function of the average station velocity. In order to accentuate the impact that

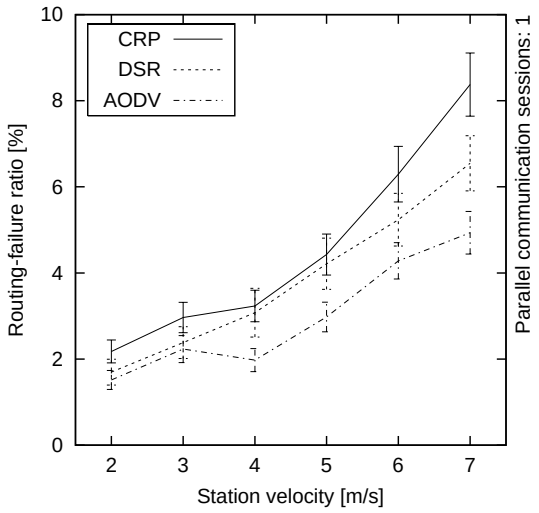

Figure 8. Routing-failure ratios with a single communication session as a function of the average station velocity.

link dependencies have on paths belonging to the same communication session, we limit network traffic to a single session here. As expected, the percentage of routing failures is higher in CRP than it is in DSR and AODV. The discrepancy increases with station mobility.

4.3.5 Destination-Discovery Frequency

A destination discovery indicates a state in which a reactive routing protocol defers datagram transportation for the purpose of acquiring the necessary routing information. The deferral entails substantial delay to those datagrams waiting for the destination discovery to conclude. Beyond this, the destination discovery becomes perceptible as a burst of high-priority protocol-control messages, which can crowd out normal-priority application-generated datagrams. Overall, destination discoveries are responsible for increased datagram-delivery delays and more numerous buffer overflows.

Figure 9 plots CRP's, DSR's, and AODV's mean number of destination discoveries per 60-seconds communication session as a function of the number of parallel sessions. We use an average station velocity of 2 m/s. Unsurprisingly, the measurements indicate that there is no correlation between the number of ongoing communication sessions and the destination-discovery frequency of any single session. We rather observe constant numbers of destination discoveries

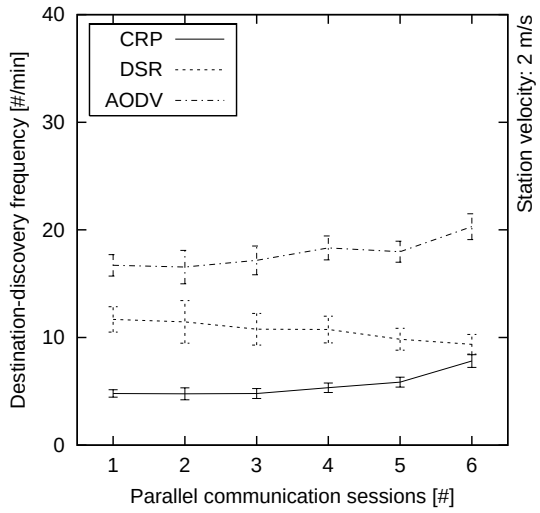

Figure 9. Destination-discovery frequencies as a function of the number of parallel communication sessions.

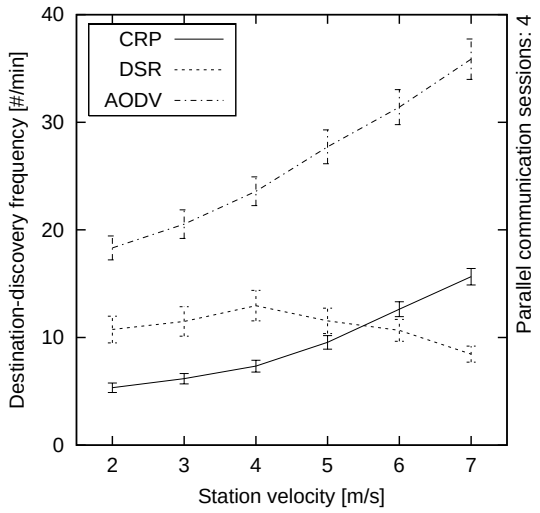

Figure 10. Destination-discovery frequencies as a function of the average station velocity.

for all routing protocols with CRP requiring less than DSR and AODV at all traffic levels.

Figure 10 shows the number of destination discoveries per communication session as a function of the average station velocity. We use a constant offered workload of four parallel communication sessions. High mobility encourages frequent network-topology changes and accelerates the breakage of active links. In general, this substantially increases a routing protocol's destination-discovery frequency. According to figure 10, however, the issue does not become obvious in DSR: While CRP's and AODV's destination-discovery counts grow with station velocity, DSR's lessens slightly. The reason is that high mobility actuates DSR's path-caching mechanism inasmuch as stations browse a large area and get to know the network's topology in many different places. Yet, the expected lifetime of routing paths is short in networks with fast topology fluctuations. Since many cache entries are unused for a long time, they are stale with high likelihood when they are eventually retrieved from the cache. This means that many attempts to replace an unusable path by a cached alternative fail. As a matter of fact, with increasing mobility, DSR's datagram-delivery ratio declines fastest amongst the observed protocols, because valuable resources are spent in vain when multiple successively chosen backup paths are defunct (cf. figure 2). A strong increase in DSR's datagram-delivery delays underlines this observation (cf. figure 4).

Corridor routing uses broadcast REPLIES in order to obtain a multi-path communication session. A broadcast REPLY transmission allows to efficiently set up multiple links on different paths in parallel. The drawback of broadcast REPLIES is that they cannot be acknowledged at data-link layer. The insecurity of broadcast REPLIES exceeds the insecurity of broadcast REQUESTS, because network-wide REQUEST flooding generates more redundant messages – and is hence more robust to collisions – than corridor-confined REPLY propagation. Indeed, when testing an early version of our CRP implementation, we found that frequent collisions during destination discovery's REPLY phase led to an unexpectedly high number of destination-discovery *attempts*, which contrasted with a much lower number of destination-discovery *initiatives*. We thus implemented an implicit-acknowledgement mechanism in order to curb the impact of REPLY collisions: When a station, X, broadcasts a REPLY, X listens whether one of its neighbors propagates the message. If no propagation seems to appear, X transmits the REPLY anew. We used this mechanism in all simulations presented in this paper.

A drawback of implicit acknowledgements is that they trigger unnecessary message retransmissions in two cases. First, a router may erroneously expect a message's propagation although none of its neighbors is authorized to forward that message. Second, the router fails to hear a propagation, which indeed takes

place, due to radio interferences.

In principle, implicit acknowledgements can sustain a destination discovery's REQUEST phase as well. Since REQUESTS are much more numerous than REPLIES, however, their potential for needless retransmissions is much higher, too. This may result in a notable impact on protocol performance. Moreover, the REQUEST phase is anyway more stable than the REPLY phase due to the higher redundancy. We hence limit implicit acknowledgements to the REPLY phase.

5 Conclusions

Multi-path routing is an approach towards higher robustness to mobility and greater exploitation of available resources in ad-hoc networks. In this paper, we present a new reactive multi-path strategy for such environments, which we call corridor routing. A corridor between a pair of communicating stations is the set of links that belong to a minimum-length path connecting those stations. Corridor routing exclusively and exhaustively uses the existing minimum-length paths. Paths are not required to be router- or link-disjoint. User data that two communicating stations exchange is distributed into all minimum-length paths between those stations. This helps to utilize available resources in different network regions and to balance network load.

We compare the QOS provision of our corridor-routing implementation – the Corridor Routing Protocol (CRP) – to that of the well-known DSR and AODV single-path routing protocols. Our study indicates that corridor routing is preferable to single-path routing in two aspects. One advantage of corridor routing is robustness to mobility: CRP uses multiple paths per communication session such that a link failure not necessarily results in a disconnection and a new destination discovery. The low number of destination discoveries is tantamount to reduced protocol-control traffic and permits short datagram-delivery delays.

The second advantage of corridor routing is a more efficient exploitation of network resources: By splitting traffic into multiple paths, CRP spatially distributes the offered workload and thus utilizes the bandwidth and buffering capacity available in different network regions. This allows CRP to transport datagrams both faster and with less buffer overflows than DSR and AODV do. We observe that CRP's lead over DSR and AODV increases with offered workload and station mobility.

CRP's prominence comes in spite of a higher number of routing failures which is brought about by link dependencies between non-disjoint routing paths. Even regionally confined station movements may lead to related routing failures on multiple paths that use the same link or cross at a common router. However, we find that CRP outperforms DSR and AODV with respect to the ratio of datagrams being successfully and timely delivered. Apparently, CRP's increased robustness to mobility and efficient exploitation of network resources outweighs a higher number of routing failures.

References

[1] The European Telecommunications Standards Institute. *Digital Cellular Telecommunications System (Phase 2+) (GSM); Adaptive Multi-Rate (AMR) Speech Transcoding (GSM 06.90 version 7.2.1 Release 1998)*, April 2000. ETSI Standard EN 301 704 version 7.2.1.

[2] K. Fall and K. Varadhan, editors. *The NS Manual*. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, April 2002.

[3] The International Telecommunication Union. *Recommendation G.114 – One-Way Transmission Time*, May 2000. ITU-T Standard G.114.

[4] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad-hoc Wireless Networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, February 1996.

[5] P. Karn. MACA – A New Channel Access Method for Packet Radio. In *Proceedings of the 9th ARRL/CRRL Computer Networking Conference*, pages 134–140. The American Radio Relay League, September 1990.

[6] S.-J. Lee and M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad-hoc Networks. In *Proceedings of the IEEE International Conference on Communications, Helsinki, Finland*, pages 3201–3205. The Institute of Electrical and Electronics Engineers, IEEE Press, June 2001.

[7] M. K. Marina and S. R. Das. On-demand Multipath Distance Vector Routing in Ad-hoc Networks. In *Proceedings of the 9th IEEE International Conference on Network Protocols, Mission Inn, Riverside, California, USA*, pages 14–23. The Institute of Electrical and Electronics Engineers, IEEE Press, November 2001.

[8] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi. On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad-hoc Networks. In *Proceedings of the 1st ACM Annual Workshop on Mobile Ad-hoc Networking and Computing, Boston, Massachusetts, USA*, pages 3–10. The Association for Computing Machinery, IEEE Press, August 2000.

[9] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1997. IEEE Standard 802.11-1997, The Institute of Electrical and Electronics Engineers.