Drohnengestützte Kommunikation mittels Delay Tolerant Networks für Krisensituationen

Martin Stille

Hochschule für Angewandte Wissenschaften Hamburg
Berliner Tor 5, 20099 Hamburg, Deutschland
Martin.Stille@haw-hamburg.de

Abstract. Diese Arbeit untersucht die Integration von Delay-Tolerant-Networks (DTNs) mit Drohnentechnologie im Rahmen des Rescue-Mate Projekts, das darauf abzielt, das Lagebild der Einsatzkräfte bei Sturmfluten in Hamburg zu verbessern. Dabei analysiert die Arbeit die grundlegende Architektur von DTNs, insbesondere die Funktionsweise der Bundle-Schicht und des Store and Forward Mechanismus, sowie die Evaluierung von Routing-Strategien wie Epidemic Routing, Spray-and-Wait und Custody-Transfer. Im Anschluss werden zwei Ansätze zur Einbindung von Drohnen in das DTN evaluiert. Drohnen dienen als Data-Mule und als Relay-Knoten. Anschließend werden Herausforderungen hervorgehoben, darunter die Auswirkung der extremen Wetterbedingungen auf den Drohnenbetrieb, die Latenz und der Ressourcenverbrauch von Sensorknoten. Die Arbeit schließt mit Forschungsvorschlägen, darunter die Implementierung und Erprobung eines RIOT-Knotes sowie Feldtests zur Untersuchung der Drohnen-Sensor-Kommunikation und des Energieverbrauchs.

Keywords: Delay-Tolerant-Network · IoT · Drohne · LoRa

1 Einleitung

Hamburg, die zweitgrößte Stadt Deutschlands und bedeutender Seehafen an der Elbe, ist als Hafenstadt besonders anfällig für Sturmfluten und hat eine lange Geschichte mit diesen. Trotz der Erhöhung der Deiche und der Installation von Flutschutztoren bleibt Hamburg aufgrund des Klimawandels anfällig für Sturmfluten. Die Stadt steht deshalb vor der kontinuierlichen Aufgabe, ihre Schutzmaßnahmen an die sich verändernden Bedingungen anzupassen. Die Motivation hinter dem Rescue-Mate-Projekt zielt auf die Verbesserung des situationalen Lagebildes der Einsatzkräfte bei einer Sturmflut in Hamburg. Dabei ergibt sich die Relevanz aus der zunehmenden Komplexität von Krisensituationen und der Notwendigkeit, Einsatzkräfte und Entscheidungsträger mit relevanten Informationen zu versorgen, die als Basis für Entscheidungen genutzt werden sollen [1]. Deshalb sollen die Herausforderungen untersucht werden, die bei der Erstellung eines Lagebildes entstehen, die die Integration von diversen Datenquellen, darunter Umwelt- und Verkehrssensordaten, Drohnenvideos sowie Social-Media-Informationen und Rettungsfunk, voraussetzen [1].

In diesem Zusammenhang stellen die zunehmende Verbreitung von Internet-of-Things (IoT) Anwendungen hohe Anforderungen an Energieeffizienz und autarken Betrieb. Dabei müssen eingebettete IoT-Systeme lange Zeiträume ohne Batteriewechsel oder gar durch Energie-Harvesting, also komplett ohne externe Energiequelle, funktionsfähig bleiben. Hierbei spielen sowohl stromsparende Hardware als auch passende Betriebssysteme eine große Rolle.
Dabei bietet sich RIOT, ein speziell für ressourcenarme IoT-Geräte entwickeltes Open Source Betriebssysteme an, dass sich durch seine geringe Speicheranforderung und Energieeffizienz auszeichnet [2]. In Kombination mit energieeffizienten Funktechnologien, wie LoRa [3], was sich zur Kommunikation über große Distanzen bei geringem Energieverbrauch eignet, lassen sich nachhaltige IoT-Lösungen entwickeln.
Die Forschungsgruppe INET der HAW beteiligt sich an dem Teilprojekt "Resiliente Sensornetze für die Flutschutztorüberwachung und die Notfallkommunikation". Ihr Beitrag umfasst ein Konzept zur Überwachung von Hochwasser-Angriffspunkten. Dazu gehören das Design einer energieminimierenden Sensorbox als Grundlage der Datenerfassung, die Entwicklung und Analyse von Verfahren zur koordinierten langreichweitigen Funkkommunikation sowie die Implementierung geeigneter Sicherheitsmaßnahmen für die resiliente Funkkommunikation, um die Integrität der übertragenen Daten zu gewährleisten. Darüber hinaus entwickelt die Forschungsgruppe ein resilient aufgebautes, sicheres und fehlertolerantes Sensornetzwerk, das als Bestandteil der Rückfallkommunikation dient [4].
Im Rahmen dieser Arbeit wird untersucht, inwieweit ein Delay-Tolerant-Network (DTN) im Zusammenspiel mit Drohnen als resiliente Rückfallkommunikation für ein fehlertolerantes Sensornetz genutzt werden kann und welche Problematiken hierbei entstehen. Dabei sollen an den Drohnen befestigte Knoten als Rückfallkommunikation genutzt werden, die beim Ausfall der Kommunikation zwischen dem Sensornetz und dem Krisenstab gestartet werden und als mobiler Knoten dienen, sodass die Kommunikation wiederhergestellt werden kann.
Deshalb wird zuerst auf den Aufbau eines DTNs eingegangen und die zentralen Funktionsweisen werden erklärt. Zusätzlich werden verschiedene Routing-Strategien auf ihre Zuverlässigkeit und ihren Ressourcenverbrauch bewertet. Im nächsten Schritt werden zwei Ansätze dafür vorgestellt, wie die Drohnen im Rescue Mate Projekt integriert werden können. Daraufhin werden die Problematiken wie Wind und Ressourcenverbrauch diskutiert und im Anschluss ein Fazit gezogen.

2 Delay-Tolerant-Network

Etablierte Netzwerktechnologien, wie TCP/IP-basierte Kommunikation, basieren weitgehend unter der Annahme einer dauerhaft stabilen Verbindung zwischen den Kommunikationsknoten. In vielen realen Szenarien, wie Katastrophengebieten, abgelegenen Regionen oder im Weltraum, können diese Voraussetzungen jedoch nicht erfüllt werden. Hier stoßen die traditionellen Netzwerke an ihre

Grenzen, da sie auf durchgängige Ende-zu-Ende-Konnektivität angewiesen sind. Um diese Herausforderungen zu erfüllen, wurde das Konzept der Delay-Tolerant Networks entwickelt. In diesem Abschnitt wird einmal der grundsätzliche Aufbau eines DTNs mit der Bundle-Schicht erklärt und dem Store-and-Forward-Mechanismus. Im nächsten Schritt werden unterschiedliche Routing-Strategien betrachtet und nach Zuverlässigkeit und Ressourcenverbrauch bewertet.

2.1 Bundle-Schicht

Die zentrale Architekturkomponente des DTNs bildet die Bundle-Schicht und ermöglicht zuverlässige Kommunikation auch bei Verbindungsabbrüchen. Als Overlay-Netzwerk operiert die Bundle-Schicht zwischen der Anwendungs- und der Transportschicht, wie in Abbildung 1 gezeigt wird, und abstrahiert die zugrunde liegende Netzwerkheterogenität. Dabei werden die zu verschickenden Daten in *Bundles* strukturiert, die weitere Metadaten für Routing und Zustellung beinhalten. Die darunter liegenden Protokolle können für jeden Knoten unterschiedlich sein und werden für die jeweiligen Anforderungen des Knotens ausgewählt [5]. Zusätzlich unterstützt die Architektur die Fragmentierung der Bundles, um zu große Bundles zu behandeln. Dabei kann proaktiv fragmentiert werden, sodass bei geplanter Kommunikation die Fragmente so angepasst werden, dass die passende Menge bei der nächsten Verbindung übertragen werden kann. Bei unerwarteten Übertragungsstörungen wird reaktiv fragmentiert. Hierbei wird das teils übertragene Bundle in zwei logische Fragmente aufgeteilt. Ein Fragment enthält den schon übertragenden Teil der Daten, während das andere Fragment die verbleibenden Daten enthält. Die verbleibenden Daten werden anschließend als eigenständiges Bundle verpackt und übertragen. Beim Zielknoten werden die Daten schließlich wieder zusammengefügt [6].

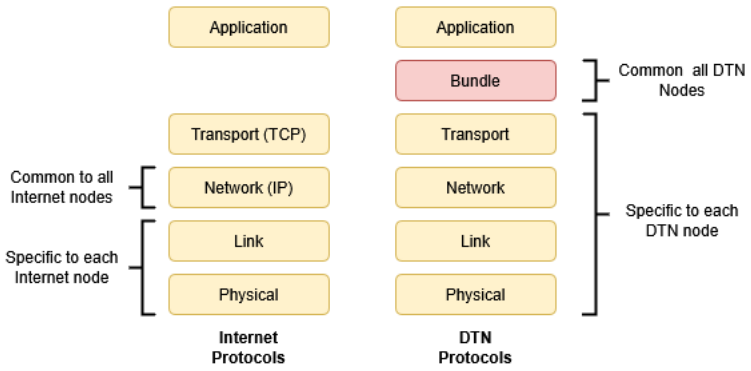

Fig. 1. Networkstack DTN nach [5]

2.2 Store-and-Forward

Ein weiteres zentrales Paradigma, welches ein DTN von einem klassischen TCP/IP-Netzwerk unterscheidet, ist das Store-and-Forward-Weiterleiten, welches für die Datenübertragung genutzt wird. Dies ermöglicht die Kommunikation in Umgebungen mit hohen Latenzen, häufigen Unterbrechungen oder instabilen Verbindungen. Dieses Konzept wird als Teilstreckenverfahren bezeichnet und ist für Szenarien konzipiert, in denen keine kontinuierliche Ende-zu-Ende-Verbindung besteht. Dabei gliedert sich das Konzept in zwei Phasen.

Store: Die Daten, die ein Sensor oder Knoten generiert, werden lokal in einem Speicher persistent gespeichert, bis eine Weiterleitung möglich ist. Somit kann auf eine dauerhaft stabile Verbindung verzichtet werden, auf die z. B. TCP angewiesen ist.

Forward: Sobald ein Knoten die Möglichkeit besitzt, eine Verbindung zu einem anderen Knoten aufzubauen, kann er die gespeicherten Daten nach dem entsprechenden Routing-Protokoll übertragen, sodass die Daten entweder an ihr gültiges Ziel gelangen oder an andere Zwischenknoten weitergeleitet werden, die dem Ziel möglicherweise näher sind.

In manchen DTN-Topologien kann es vorkommen, dass Knoten sich niemals begegnen und somit ihre Daten nicht weiterleiten können. In diesen Fällen wird eine dritte Phase, der Carry, zwischen Store und Forward eingefügt.

Carry: Einige Knoten können sich physisch durch das Netz bewegen und transportieren dabei die Datenpakete von anderen Knoten. Diese speziellen Transportknoten werden als *Data-Mules* bezeichnet und behalten die Datenpakete, solange, bis eine Weiterleitungsmöglichkeit besteht. Im Kontext des RescueMate-Projekts übernehmen Drohnen diese Funktionalität und stellen bei Verbindungsverlust eine Rückfallkommunikation her.

2.3 Routing-Strategien

Routing-Mechanismen in DTNs stellen eine zentrale Herausforderung in der Entwicklung von robusten Kommunikationssystemen dar. Im Gegensatz zu traditionellen Netzwerken, die auf stabilen Ende-zu-Ende-Verbindungen basieren, müssen DTNs mit häufigen Verbindungsabbrüchen und der Abwesenheit von durchgängigen Pfaden umgehen können. Dabei werden in diesem Abschnitt die verschiedenen Routing-Strategien Epidemic Routing, Spray-and-Wait und Custody-Transfer vorgestellt und in Bezug auf Zuverlässigkeit und Ressourcenverbrauch bewertet.

Epidemic Routing Epidemic Routing basiert auf dem Prinzip des Floodings, ähnlich wie bei einer Krankheit. Jeder Knoten schickt seine Datenbundles an alle

erreichbaren Nachbarn, sodass jeder Knoten in Reichweite über alle Datenbundles der anderen Knoten verfügt. So verbreiten sich durch wiederholtes Treffen der Knoten nach und nach alle Nachrichten im Netzwerk. Sobald der Zielknoten die Nachricht erhält, ist die Zustellung abgeschlossen [7]. Einer der Hauptvorteile von Epidemic Routing ist, dass es keinen festen Pfad im Netzwerk benötigt, um beim Empfänger anzukommen, und deshalb auch keine feste Netzwerkstruktur benötigt wird. Dadurch ist es sehr zuverlässig, solange die Buffer groß genug sind. Epidemic Routing ist jedoch sehr ressourcenintensiv, da jeder Knoten potenziell alle Nachrichten von allen Knoten speichern muss und es sehr viele Duplikate im Netzwerk gibt. Im gleichen Zug gibt es eine hohe Netzwerknutzung, da viele Nachrichten zwischen den Knoten hin- und her geschickt werden müssen [7]. Zugleich werden Nachrichten auch nach der Zustellung weiter durch das Netz propagiert. Somit wird die Batterie bei entsprechenden Knoten sehr stark beansprucht, was zum Ausfall dieser führen kann.

Spray-and-Wait Spray-and-Wait [8] reduziert die Nachteile von Epidemic Routing, indem es eine begrenzte Verbreitung der Nachricht mit einem Warteprozess kombiniert, um Ressourcen zu sparen. Dabei teilt sich das Protokoll in zwei Phasen. Bei der Spray-Phase besitzt der Quellknoten eine bestimmte Anzahl L an Kopien, die er an seine Nachbarn schicken kann.
In der Wait-Phase warten alle Knoten, die eine Kopie besitzen, darauf, den Zielknoten direkt zu treffen. Ein weiteres Weiterleiten der Kopien findet nicht mehr statt. Da es vorkommen kann, dass der Zielknoten nie innerhalb der ersten zwei Hops erreicht werden kann, gibt es den verbesserten Binary-Spray-and-Wait-Ansatz. Hierbei werden nicht alle Nachrichten direkt an Nachbarknoten des Quellknotens weitergegeben, sondern die Anzahl der Kopien halbiert sich mit jedem Hop [8]. Startet ein Knoten mit zehn Kopien und gibt fünf weiter, dann zwei, dann eins usw.
Die Vorteile sind eine reduzierte Netzwerkbelastung und ein verringerter Energie- und Speicherverbrauch der einzelnen Knoten, jedoch kann es vorkommen, dass Nachrichten, die viele Hops zu ihrem Ziel brauchen, verloren gehen, was die Zuverlässigkeit beeinträchtigt.

Custody-Transfer Um die Zuverlässigkeit der Datenübertragung zu gewährleisten, unterstützt das Bundle-Protokoll den Custody-Transfer-Mechanismus, da in einigen Use Cases für manche Sender eine Sendewiederholung der Applikationsdaten nicht möglich ist. Dies kann aufgrund von physischer Bewegung im Netzwerk oder Energiemanagement der Fall sein [9]. Dabei baut der Custody-Transfer auf dem Store-and-Forward-Prinzip auf. Die Verantwortung für die Zustellung eines Datenpakets wird zwischen den Netzwerkknoten übertragen.
Hierbei kann ein Sender einen Knoten anfragen, ob dieser die Custody für ein Bundle übernimmt. Falls ja, wird dieser zum *Custodian*. Dieser speichert das Bundle so lange, bis es erfolgreich an den nächsten Knoten oder ans Ziel übertragen wurde. Nach der Übertragung wartet der Custodian auf eine Bestätigung des Empfängers. Erhält er diese, gibt er die Custody ab und löscht das Bundle

aus seinem Speicher. Der nächste Knoten ist somit für die Retransmission verantwortlich [9]. Dieser Mechanismus reduziert die Ressourcen des ursprünglichen Senders und erhöht die Reliability der Nachrichtenübertragung, jedoch steigt die Komplexität bei Fragmentierung, da nur von Teilen des Bundles die Custody übernommen wird und nicht vom originalen Bundle und somit das Zusammenfügen erschwert [10]. Zusätzlich kann es bei einem Ausfall des Custodian dazu führen, dass die Nachricht komplett verloren geht, was sich negativ auf die Reliability auswirkt und somit ein *Single Point of Failure* geschaffen wird.

3 Integrationsansätze eines DTNs mit Drohnenknoten für das Rescue-Mate Projekt

Da sich Drohnenknoten unterschiedlich in einem DTN einsetzen lassen, um eine Verbindung zum Kontrollstab wiederherzustellen, werden in diesem Abschnitt zwei unterschiedliche Ansätze vorgestellt. Einmal den Einsatz der Drohne als Data-Mule und einmal als Relay zwischen Sensornetz und Krisenstab.

3.1 Data-Mule Ansatz

Die Drohnen lassen sich als Data-Mule in dem DTN einsetzen. Dabei fliegt die Drohne über alle Sensorknoten und sammelt die Daten von diesen ein und speichert sie zwischen. Sind alle Knoten angeflogen worden, kehrt die Dronhe zur Groundstation zurück, an die sie die Daten übergeben kann, sodass der Krisenstab auf diese zugreifen kann. Dieser Ansatz wurde auch in [11] erfolgreich getestet und wird in Abbildung 2 noch einmal veranschaulicht. Dabei kann mit wenig Ressourcen (wenig Drohnen) eine große Fläche abgedeckt werden und eine Verbindung zum Kontrollstab hergestellt werden. Der größte Nachteil bei diesem Ansatz ist die Latenz, die durch die Drohne verursacht wird, da die Nachrichten erst bei der Groundstation ankommen, wenn die Drohne in der Nähe dieser ist. Dafür ist aber diese Latenz planbar, da durch die Route festgelegt ist, wie lange die Drohne unterwegs ist. Wenn nur einige Knoten im Netz nicht erreichbar sind, kann die Route so angepasst werden, dass nur diese Knoten angeflogen werden und somit die Zeit, die die Drohne unterwegs ist, reduziert werden kann. Des Weiteren muss in Betracht gezogen werden, dass die Drohne ein *Single Point of Failure* darstellt, da beim Verlust der Drohne die Daten nie zum Zielknoten gelangen würden.

3.2 Relay Ansatz

Als weiteren Ansatz lässt sich die Drohne (oder mehrere) als Relay-Knoten benutzen, sodass von allen Knoten wieder ein Pfad zur Groundstation besteht und die Daten an den Krisenstab gelangen können. Dieser Ansatz ist in Abbildung 3 noch einmal aufgezeigt. Der klare Vorteile dieses Ansatzes ist die Reduktion der Latenz, da Knoten wieder direkt mit der Groundstation kommunizieren können. Diese Reduktion hat jedoch bei physisch weit verteilten Knoten den Preis,

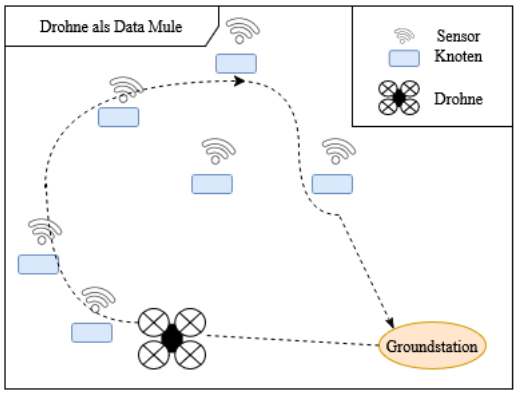

Fig. 2. Drohne als Data-Mule

dass mehrere Drohnen benötigt werden, um die Verbindung aufrechtzuerhalten. Des Weiteren muss hier ebenfalls die Flugzeit der Drohne berücksichtigt werden, da Hin- und Rückflug zur gewünschten Position nicht zum eigentlichen Zweck genutzt werden können. Somit müssen die Drohnen häufiger ausgewechselt werden, was deren Effektivität und Nutzzeit reduziert.

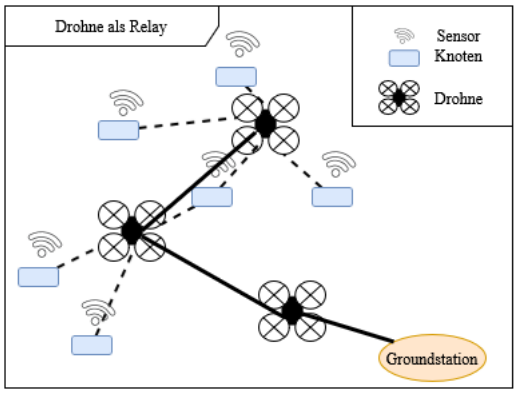

Fig. 3. Drohnen als Relay

3.3 LoRa

Um in den oben gezeigten Abschnitten 3.1 und 3.2 eine Verbindung zwischen Drohne und Sensorknoten herzustellen, wird eine Funkverbindung benötigt, die energiearme Kommunikation über weite Entfernungen ermöglicht. LoRa bietet

sich hierfür an, da es in städtischen Gebieten eine Reichweite von bis zu 5 km und in ländlichen Gebieten bis zu 15 km aufweist [3]. Des Weiteren besitzt LoRa eine hervorragende Durchdringungsfähigkeit in Gebäude, sodass auch Keller und Untergeschosse erreicht werden können. LoRa bietet dabei Datenübertragungsraten von 0,3 kbit/s bis zu 50 kbit/s an, was für IoT-Anwendungen ausreichend ist [12].
Ein weiterer Vorteil ist der geringe Energieverbrauch von LoRa. Diese liegt beim Senden bei ca. 80 mA und beim Empfangen bei ca. 30 mA. Im Ruhezustand werden nur noch 90 µA verbraucht, was sich in einer langen Batterielebensdauer widerspiegelt [13].

3.4 Problemkreis

Wind Eine große Problematik beim Einsatz von Drohnen als Netzwerk-Relay bei schweren Sturmfluten ist die Wetterlage. Dabei spielt vor allem die Windgeschwindigkeit eine große Rolle. Bei der letzten Sturmflut 2013 in Hamburg mit dem Tief Xaver wurden in Hamburg-Fuhlsbüttel Böen bis 99 km/h und eine durchschnittliche Windgeschwindigkeit von 59 km/H erreicht [14]. Auch in dem Szenario des Rescue-Mate-Projektes werden Windgeschwindigkeiten bis 120 km/H angenommen [1]. Diese Windgeschwindigkeiten überschreiten die angegebene maximale Windgeschwindigkeit von Enterprise-Multi-Rotor-Drohnen. Als Beispiel wurde hierfür die Matrice 30 vom Hersteller DJI betrachtet, die einer Windbelastung von maximal 43 km/H standhält und für den Einsatz von Search and Rescue und für die Aufklärung von Einsatzkräften entwickelt wurde [15].
Auch kommerzielle Starrflügler-Drohnen, die durch ihre Aerodynamik effizienter sind, da die Motoren lediglich Vortrieb erzeugen und die Flügel den Auftrieb, werden den angenommenen Windgeschwindigkeiten nicht gerecht. Als Beispiel wurde hierfür die Trinity Pro vom Hersteller Quantum Systems betrachtet. Diese Drohne wurde für den professionellen Kartierungsgebrauch entwickelt. Der Hersteller gibt als maximale Windbelastung 64,8 km/h an [16], welche von der Windgeschwindigkeit im Rescue-Mate-Szenario weit überschritten wird. Zusätzlich haben starke Winde die Fähigkeit, die Geschwindigkeit und Flugroute zu beeinträchtigen. Dabei kann Gegenwind, der stärker als die Fluggeschwindigkeit der Drohne ist, dazu führen, dass die Drohne zum Stehen kommt oder in den Rückwärtsflug gerät [17]. Dies kann insbesondere beim Rückflug zum Problem werden, da die Drohne möglicherweise nicht mehr zum ursprünglichen Landeplatz zurückkehren kann und zwischenlanden muss.
Seitenwinde können dazu führen, dass die Drohne von ihrer ursprünglichen Flugroute abweicht und dabei über bewohntes Gebiet, Menschenmengen oder in Gebiete geweht wird, in denen die Drohne nicht mehr geborgen werden kann[17]. Im schlimmsten Fall kann die Drohne abstürzen und dabei Gebäude beschädigen oder Menschen verletzen. Des Weiteren wirken sich die Windgeschwindigkeiten auch auf die Flugzeit der Drohne aus, da die Motoren mehr Schub erzeugen müssen, um gegen den Wind zu arbeiten. Dies kann dazu führen, dass sich die Flugzeit erheblich verkürzt und geplante Flugrouten nicht mehr ausgeführt wer-

den können. Zusätzlich muss neben Wind auch Regen in Betracht gezogen werden, da dieser sonst die Elektronik beschädigen kann.

Latenz Die Latenz von Informationen kann im Krisenfall entscheidende Auswirkungen auf die Entscheidung haben, sodass diese minimiert werden sollte. Hier müssen noch weitere Anforderungen erhoben werden, um festzustellen, welche Latenzen noch akzeptabel sind, um das DTN und den Einsatz der Drohne auf diese abzustimmen.

Ressourcenverbrauch Eine weitere Problematik ist der Ressourcenverbrauch der einzelnen Knoten im DTN. Da die meisten Sensorknoten über Akkus betrieben werden, ist der Stromverbrauch entscheidend für den Dauerbetrieb im Feld. Deshalb sollten die Routing-Protokolle im DTN so gewählt werden, dass der Stromverbrauch möglichst gering gehalten wird. Da der meiste Strom beim Senden/Empfangen verbraucht wird, sollte deshalb das Routing-Protokoll so gewählt werden, dass dieses minimiert wird. Dies ist insbesondere wichtig, da durch Knoten, bei denen die Akkus verbraucht sind, tote Zonen entstehen, die das Routing erschweren. Des Weiteren muss beachtet werden, dass die Sensorknoten nicht unbegrenzt Speicher zur Verfügung haben und deshalb mit redundanten Daten der anderen Knoten sparsam umgegangen werden sollte.

Single Point of Failure Ein weiterer Problempunkt ist, dass die Drohnen einen Single Point of Failure schaffen können, sodass die Daten aus dem Sensornetz nicht mehr eingespeist oder dauerhaft verloren werden können. Dies kann besonders durch Wind beeinflusst werden, da dieser sich auf die Flugzeit und die Flugroute auswirken kann. Dies kann durch Redundanz der Drohnen vermieden werden, jedoch erhöhen sich hierbei die Betriebskosten.

4 Fazit und Ausblick

Die Integration von Delay-Tolerant-Networks mit Drohnentechnologie im Rahmen des Rescue-Mate-Projektes zeigt verschiedene Ansätze, wie die Krisenkommunikation bei Sturmfluten in Hamburg verbessert werden kann. Die vorgestellten Konzepte des Data-Mules- und Relay-Ansatzes bieten eine flexible Lösung für die Datenübertragung. Es wurden auch Herausforderungen identifiziert, insbesondere die Auswirkungen der Wetterbedingungen auf den Drohneneinsatz, Latenzprobleme und der Ressourcenverbrauch der Sensorknoten.
Es wurden ebenfalls verschiedene Routing-Strategien, wie Epidemic Routing, Spray-and-Wait und Custody-Transfer, in Bezug auf Zuverlässigkeit und Ressourcenverbrauch analysiert.
Für zukünftige Forschungsarbeiten soll sich im ersten Schritt darauf konzentriert werden, die Mule- bzw. Relay-Technologie zu implementieren und auszuprobieren. Dafür soll ein RIOT-Knoten entworfen werden, der mithilfe von LoRa ein DTN implementiert. Im nächsten Schritt soll dieser Knoten an einer Drohne

montiert werden, um diesen als mobilen Knoten zu benutzen. Des Weiteren sollen Feldversuche durchgeführt werden, um die optimale Reichweite zwischen Sensor und Drohne zu bestimmen.
Insgesamt lässt sich sagen, dass die Kombination aus DTNs und Drohnen einen vielversprechenden Ansatz für resiliente Kommunikationssysteme liefert. Weitere Tests und Forschungen sind jedoch erforderlich, um die identifizierten Herausforderungen zu bewältigen.

References

1. Rescue-Mate, "Projektszenario," 2025. [Online]. Available: https://www.rescue-mate.de/szenario/
2. E. Baccelli, C. Gündogan, O. Hahm, P. Kietzmann, M. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch, "RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4428–4440, December 2018. [Online]. Available: http://doi.org/10.1109/JIOT.2018.2815038
3. Lora-Alliance, "LoRa Alliance Massive and Critical IoT Requirements Infographic," 2022. [Online]. Available: https://resources.lora-alliance.org/technology-comparisons/lora-alliance-massive-and-critical-iot-requirements-infographic
4. INET, "Rescue mate," 2025, (Date last accessed 15-July-2014). [Online]. Available: https://www.inet.haw-hamburg.de/projects/rescue-mate
5. F. Warthman *et al.*, "Delay-and disruption-tolerant networks (DTNs)," *A Tutorial. V.. 0, Interplanetary Internet Special Interest Group*, pp. 5–9, 2012.
6. M. Ho and K. Fall, "Poster: Delay tolerant networking for sensor networks," in *Proc. of IEEE Conference on Sensor and Ad Hoc Communications and Networks*, 2004.
7. C. Srividya and N. Rakesh, "Enhancement and performance analysis of epidemic routing protocol for delay tolerant networks," in *2017 International Conference on Inventive Systems and Control (ICISC)*, 2017, pp. 1–5.
8. T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 252–259. [Online]. Available: https://doi.org/10.1145/1080139.1080143
9. C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, "Delay- and disruption-tolerant networking (dtn): An alternative solution for future satellite networking applications," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1980–1997, 2011.
10. K. Fall and S. Farrell, "Dtn: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 828–836, 2008.
11. M. Zhang and X. Li, "Drone-enabled internet-of-things relay for environmental monitoring in remote areas without public networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7648–7662, 2020.
12. M. H. M. Ghazali, K. Teoh, and W. Rahiman, "A systematic review of real-time deployments of uav-based lora communication network," *IEEE Access*, vol. 9, pp. 124 817–124 830, 2021.
13. M. Nurgaliyev, A. Saymbetov, Y. Yashchyshyn, N. Kuttybay, and D. Tukymbekov, "Prediction of energy consumption for LoRa based wireless sensors network," *Wireless Networks*, vol. 26, no. 5, p. 3507–3520, Jul. 2020. [Online]. Available: https://doi.org/10.1007/s11276-020-02276-5

14. B. u. G. Landesbetriebe Straßen, "Die Sturmflut nach dem Tief Xaver vom 5. bis 7. Dezember 2013," 2014. [Online]. Available: https://lsbg.hamburg.de/resource/blob/784538/54f1a9c9124460439c06e501b4d89442/bericht-nr-16-die-sturmflut-nach-dem-tief-xaver-vom-5-bis-7-dezember-2013-data.pdf
15. DJI, "Matrice 30 Series," 2025. [Online]. Available: https://enterprise.dji.com/de/matrice-30
16. Quantum-Systems, "Next-generation evtol fixed-wing mapping drone," 2025. [Online]. Available: https://quantum-systems.com/trinity-pro/
17. E. Ranquist, M. Steiner, and B. Argrow, "Exploring the range of weather impacts on UAS operations," in *18th Conference on Aviation, Range and Aerospace Meteorology, Seattle, WA*. American Meteorological Society (AMS), 2017.