

Probleme beim Einsatz von DTNs

Gerhard Steffek

Betreuer: Nils Kammenhuber

Seminar Innovative Internet Technologien und Mobilkommunikation SS2011

Lehrstuhl Netzarchitekturen und Netzdienste

Fakultät für Informatik, Technische Universität München

Email: steffek@in.tum.de

KURZFASSUNG

Als Disruption and Delay Tolerant Networks (DTNs) werden Netzwerke bezeichnet, die darauf ausgelegt sind, trotz nicht durchgängig verfügbarer Verbindungen und Netzwerk-Teilnehmer eine gewisse Konnektivität zu gewährleisten. In dieser Arbeit wird der experimentelle Einsatz eines DTNs in Lappland und die dabei auftretenden Probleme vorgestellt. Außerdem werden der Aufbau und die Vor- und Nachteile eines DTNs zur mobilen Datenkommunikation in Autos und Zügen besprochen und verschiedene Übertragungsmethoden miteinander verglichen.

Schlüsselworte

Delay-Tolerant Network, DTN, HTTP-DTN, SNC, HTTP, MIME

1. Einleitung

In vielen Bereichen ist die Internetanbindung selbstverständlich. Zuhause haben viele Menschen Zugang zu einer Internetverbindung und auch in Cafés, Restaurants und Universitäten ist Wlan immer häufiger verfügbar. An Orten an denen kein Wlan vorhanden ist wird mittlerweile häufig auf Mobilfunk zurückgegriffen, welches großteils selbst in U-Bahnen verfügbar ist. Bewegt man sich jedoch außerhalb der Ballungszentren, beispielsweise bei einer Fahrt auf der Autobahn oder im Zug, lässt die Netzabdeckung stark nach und es treten vermehrt Verbindungsabbrüche auf. Aber auch hier gibt es Ansätze und Ideen die Internetversorgung, kostengünstiger als durch das Aufstellen weiterer Mobilfunkmasten, unterwegs zu gewährleisten.

Problematischer gestaltet sich die Netzabdeckung in manchen Ländern mit einer, im Vergleich zu Deutschland, sehr viel geringeren Bevölkerungsdichte. Ein Beispiel dafür sind die Rentierhüter Lapplands im Norden Schwedens die weder über eine Internetanbindung noch ein Stromnetz verfügen. Diese verändern mehrmals jährlich ihren Standort, teilweise um mehrere hundert Kilometer innerhalb weniger Wochen. Nötig ist dies aufgrund vieler Faktoren (Verhalten der Rentiere, Klima, Gesetze, Tourismus etc.) [2].

DTNs sind jedoch auch in anderen Gebieten wichtig. Ursprünglich wurde bei der Entwicklung an interplanetare Kommunikation und ihre Probleme gedacht. Hier treten hohe Fehlerraten (Bit Error Rates, BER) zusammen mit hohen Latenzen auf. Dies stellte andere Anforderungen an die Übertragungsprotokolle als beim Internet über Kabelverbindungen auf der Erde. Zum Beispiel dauert mit

Lichtgeschwindigkeit eine Übertragung zum Mond 1,7 Sekunden, zum Mars sogar 8 Minuten [7]. Da auch auf der Erde eine Verbindung immer und überall unabhängig von Kabelverbindungen erwünscht ist, finden mittlerweile viele Entwicklungen im Bereich DTNs für die Kommunikation auf der Erde statt. Hierzu gehört nicht nur die Verbindung zum Internet sondern auch die Datenübertragung innerhalb lokaler Netzwerke wie zum Beispiel in Sensornetzwerken.

In den folgenden Kapiteln werden unterschiedliche Einsatzzwecke für DTNs aufgezeigt. Dazu wird neben dem Einsatzzweck auch ein möglicher Aufbau eines DTNs für diesen erklärt, sowie auf die möglichen Schwachstellen und Schwierigkeiten eingegangen. Als erstes wird das genannte DTN in Lappland beschrieben, danach ein DTN für mobile Kommunikation. Im Anschluss daran wird der Einsatz und die Vorteile von HTTP in DTNs erklärt. Als letztes wird mit der Beschreibung eines Sensornetzwerkes für die Landwirtschaft noch ein weiterer Einsatzort für DTNs gezeigt.

2. Einsatz eines DTN in Lappland

2.1 Überblick

In Lappland wurde bereits ein DTN in einer wenig besiedelten Gegend zu Testzwecken eingesetzt. Beim Projekt „Sámi Network Connectivity“ wird versucht die Rentierhüter Lapplands mit einer Internetanbindung zu versorgen. Dies geschieht um ihnen die Möglichkeit zu geben sich weiterzubilden, aber auch um Kontakt mit der Familie oder das Lesen von Nachrichten zu ermöglichen. Durch eine Internetanbindung sind Kontakt zu Schulen und Universitäten aber auch neue Möglichkeiten der Arbeit denkbar (Heimarbeit, Internetgeschäfte usw.). Dadurch soll der Beruf des Rentierhüters an Attraktivität gewinnen und erhalten bleiben [2].

Die Anwendungen, die anfangs implementiert werden sollten sind daher [2]:

- Senden und Empfangen von Emails
- Übertragen von Dateien
- Zugriff auf Websites und Web Services

Aufgrund fehlender Infrastruktur, großer Entfernungen und des Nomadenverhaltens der Sámi sind nicht nur eine direkte Anbindung an das Internet (über Kabel oder kabellos) sondern auch eine Stromversorgung über das Stromnetz nicht möglich. Große Teile der zu versorgenden Gebiete befinden sich in Nationalparks, hier ist auch aus gesetzlichen Gründen der Aufbau einer festen Infrastruktur (Verlegen von Internet- oder

Stromkabeln, festes Anbringen von Antennen usw.) häufig nicht möglich. Auch die Kosten spielen eine große Rolle und sollten möglichst niedrig gehalten werden, da der Internetzugang eine Verbesserung der Lebensqualität und eventuell eine finanzielle Entlastung bedeuten sollte und nicht eine weitere finanzielle Belastung darstellen [2].

2.2 Anwendungen

Neben den anfangs genannten Anwendungen war auch eine Funktion angedacht, um Rentiere über Sensoren aus der Entfernung überwachen und ihren Aufenthaltsort bestimmen zu können. In dem hier beschriebenen Experiment wurde diese Funktion aber nicht implementiert.

Stattdessen wurde versucht das System so zu gestalten, dass es mit gewöhnlichen Programmen für den Zugriff auf Websites und das Verfassen und Empfangen von Emails funktioniert, wie sie normalerweise in dauerhaft mit dem Internet verbundenen Systemen verwendet werden. Aufgrund sehr hoher Latenzen und der asynchronen Art des DTNs sind Verbindungen nur sehr schwer zu implementieren, die eine Authentifizierung oder Rückmeldung vom Benutzer benötigen. Daher wurden sie hier vernachlässigt [2].

Die Bereitstellung eines Email Services wurde dabei einerseits durch seine asynchrone Art der Verbindung als theoretisch recht einfach betrachtet, andererseits aber auch als sehr wichtig eingestuft da Emails eine grundlegende Art der Kommunikation und Datenübertragung sind und eventuell sogar die „killer application“ für das ganze Projekt darstellen. Zwar sind die Protokolle (SMTP, POP3 und IMAP) eigentlich für eine synchrone Internetverbindung ausgelegt, aber durch eine Ausstattung der Hotspots mit Gateways, die einen Email Server vorgeben, wird es möglich mit normalen Email Programmen Emails zu versenden und zu empfangen. Durch einen bereitgestellten Webmail Service konnte man Emails über den Browser betrachten und verfassen. Die vom Hotspot angebotenen Email Server hatten alle den Internetgateway als Ziel. Dadurch wurden alle Emails zu diesem weitergeleitet, woraufhin er sie versendete. Ankommende Emails wurden vom Internetgateway an sämtliche Hotspots weitergeleitet, hier konnten die Benutzer sie dann abrufen [1].

Auch der Zugriff auf Websites muss von einem Projekt, dass das Internet für Privatpersonen bereitstellen möchte, natürlich ermöglicht werden. Dies ist eine sehr viel interaktivere Tätigkeit und aufgrund der hohen Latenzen um einiges aufwendiger zu implementieren. Während des Tests wurden fest eingerichtete Websites vom Internetgateway immer wieder automatisch aktualisiert und auf sämtliche Hotspots übertragen [1].

2.3 Theoretischer Aufbau

Das Netzwerk sollte aus mehreren festen Hotspots, die eine WLAN Verbindung zur Verfügung stellen und den Zugriff auf Emails und Websites ermöglichen, bestehen. Da diese Hotspots mehrere Kilometer auseinander liegen sollten (Abb. 1), wurde eine direkte drahtlose Kommunikation aus (energie-)technischen und monetären Gründen ausgeschlossen. Stattdessen sollten mobile Datenspeicher (Relays) zum Einsatz kommen, die in der Nähe eines Hotspots die Daten mit diesen austauschen. Da Speicherplatz und Bandbreite beschränkt sind kommt hier ein Protokoll namens „PROPHET“ zum Einsatz [1].

2.3.1 PROPHET

PROPHET steht für “Probabilistic Routing Protocol using History of Encounters and Transitivity”. PROPHET wurde anderen Routing-Protokollen, beispielsweise dem einfacheren Epidemic Routing, vorgezogen. Beim Epidemic Routing werden bei einem Datenabgleich alle Synchronisationsknoten auf den gleichen Stand gebracht indem jeder Knoten alle Daten erhält. Da dies zwar effektiv jedoch auch ressourcenintensiv ist, wurde stattdessen PROPHET eingesetzt, um die Verteilung der Daten auf die Knoten zu bestimmen. Hierbei wird es von der Wahrscheinlichkeit der Knoten A und B den Zielknoten C eines Pakets direkt oder indirekt zu erreichen abhängig gemacht, ob A oder B die Daten erhält. Dadurch soll nicht nur eine schnelle Übertragung der Daten ermöglicht sondern auch die von und zu den Knoten übertragene Datenmenge reduziert werden [1].

2.3.2 Datenübertragung über Middleware

Momentan basiert Internet großteils auf dem Transmission Control Protocol (TCP) zur Datenübertragung. TCP hat viele Eigenschaften, wie die Erkennung und Behebung von Fehlern bei der Datenübertragung, weswegen es für viele Bereich, ob mit oder ohne Kabel, geeignet ist. Andererseits benötigt TCP eine end-to-end Verbindung von der Quelle zum Ziel, weswegen es für ein DTN ungeeignet ist. Hier kann für gewöhnlich entweder eine end-to-end Verbindung nicht garantiert werden (mobile Anwendungen) oder ist erst gar nicht möglich (wie in diesem Fall). In einem solchen DTN muss deswegen die Datenübertragung anders gelöst werden. Da bisherige Programme wie Browser und Email Programm auf einer TCP Verbindung basieren und weiterverwendet werden sollten, wurde in den Gateways eine Middleware eingesetzt. Die Middleware ist dabei der Endpunkt für die Anwendung an den diese, über TCP, ihre Dateien schickt. Danach fasst sie die Pakete der Programme und andere für die Zustellung notwendigen Metadaten zu einem einzigen Bundle zusammen, welches über das DTN übertragen werden kann. Die Daten werden also auf den mobilen Datenspeicher übertragen und dort gespeichert, bis sie an einen anderen mobilen Datenspeicher oder Hotspot weitergegeben werden können. Diese Pakete können dabei über eine Ablaufzeit verfügen nach der sie verworfen werden, die aber für gewöhnlich nicht im Bereich von Sekunden liegt (wie die Round Trip Time bei TCP, nach Ablauf derer das Paket als verloren angesehen und erneut versendet wird) sondern mehrere Stunden, Tage oder gar Monate lang sein kann [2].

2.4 Praktischer Aufbau

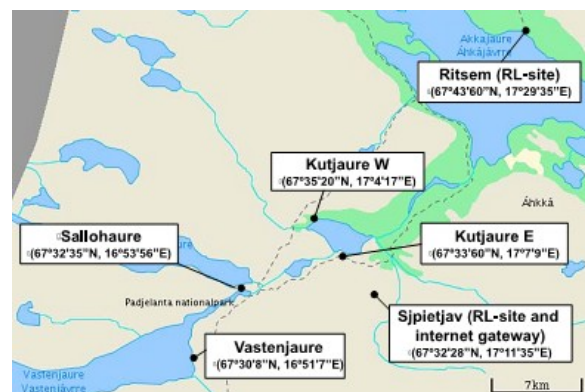


Abb. 1: Karte des Testgebiets [1]

Im Einsatz bestand das Netzwerk aus 4 fixen Hotspots und 7 mobilen Geräten um die Kommunikation zwischen diesen zu ermöglichen. Die Internetanbindung des Internetgateways (Sjpietjav) konnte dabei jedoch nicht direkt erfolgen, sondern musste auf einer kabellosen Verbindung (mit Sichtkontakt) zu einem Dorf (Ritsem) in ca. 20 km Entfernung basieren. Die Entfernungen zwischen den einzelnen Hotspots betragen ca. 10 km und der Internetgateway hatte eine Entfernung von 5 km zum nächsten Hotspot [1].

Freiwillige sorgten dabei für die Verbindung der Hotspots durch die mobilen Datenspeicher [1].

Als Hardware kamen Laptops und Tablet PCs zum Einsatz auf denen sowohl Windows als auch Linux verwendet wurden. Die Akkulaufzeit war dabei ein wichtiges Thema für die Geräte [1].

2.5 Schlussfolgerungen aus dem Projekt

2.5.1 Email Implementation

Wie erwartet wurde die Email-Funktion stark genutzt. Da PRoPHET zu debug-Zwecken über eine graphische Oberfläche verfügte über die sich auch Nachrichten direkt zustellen ließen, ergab sich eine Art „interne Email“. Diese erwies sich als schneller da dadurch der Umweg über das Internetgateway, über das sämtliche normalen Emails geleitet wurden, eingespart werden konnte, falls eine schnellere Verbindung direkt über die Hotspots möglich war. Um die Zustellung von internen Emails zu beschleunigen und Datenverkehr einzusparen, sollte in Zukunft automatisch überprüft werden ob der Empfänger sich innerhalb des Netzwerks befindet und gegebenenfalls die Email direkt intern zustellen. Während des Tests hatte es keinen negativen Einfluss, dass sämtliche Emails an alle Hotspots zugestellt wurden. Bei einem größeren und stärker benutzten Netzwerk könnte aber Bandbreite und Speicherplatz eingespart werden, wenn die Nachrichten nur an die Hotspots zugestellt würden, bei denen sich die Benutzer vorher angemeldet haben oder in deren Nähe sie sich wahrscheinlich momentan aufhalten [1].

2.5.2 Zugriff auf Websites

Das implementierte Caching-System hat funktioniert, aber der Funktionsumfang war sehr begrenzt, weswegen hier noch viel Raum für Verbesserungen bleibt.

So fehlte beispielsweise eine Funktion, die den Benutzern die Möglichkeit gibt Websites, die sich nicht im Speicher befinden, vom DTN laden zu lassen. Hier wären auch Techniken von Vorteil, die bestimmen welche weiterführenden Links mitgeladen werden sollten (intelligent prefetching). Außerdem sollten häufig besuchte Websites automatisch aktualisiert werden oder zumindest die Möglichkeit gegeben sein, sich für Updates von Websites einzutragen und dann automatisch neue Versionen der Seite zu bekommen [1]. Dadurch könnte zum Beispiel eine Nachrichtenseite oder ein Forum aktuell gehalten und die Beiträge automatisch mitgeladen werden.

Diese häufigen Updates würden jedoch auch eine Verbesserung der Übertragungsart der Websites benötigen. Während des Tests wurde beim Update einer Website immer die gesamte Seite durch das DTN geschickt und auf den Hotspots gespeichert anstatt nur ein inkrementelles Update zur bereits vorhandenen Version zu speichern. Dadurch würden sich Übertragungsdauer und Speicherbedarf bei der Aktualisierung von Websites verringern [1].

3. DTN für mobile Kommunikation

3.1 Anwendungsbereiche

Für gewöhnlich wird über die Verwendung von DTNs im Zusammenhang mit dem Erschließen neuer Kommunikationsmöglichkeiten diskutiert. Dabei kann ein DTN auch in bereits existierende Kommunikationslösungen integriert werden, um diese zu verbessern. Eine dieser bereits existierenden Lösungen stellt der Mobilfunk dar. Die Netzabdeckung wird zwar stetig verbessert, aber in abgelegeneren Gebieten oder bei hohen Geschwindigkeiten (Auto, Zug) treten immer noch häufige Verbindungsabbrüche auf, mit denen bisherige Netze (und Programme) große Probleme haben [3]. Auch die Kosten mobiler Datenübertragung (über UMTS oder GSM) sind, verglichen mit lokalen (drahtlosen) Netzwerken, noch um ein Vielfaches höher und die Übertragungsgeschwindigkeiten langsamer [4].

3.2 Theoretischer Aufbau

3.2.1 Aufbau mit direkter Verbindung

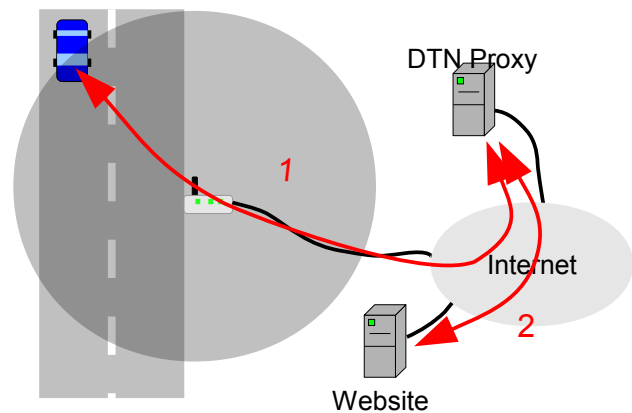


Abb. 2: Aufbau eines DTNs zur mobilen Kommunikation

Beim Projekt „Drive-thru-Internet“ soll dabei die Netzabdeckung nicht auf Ebene der Mobilfunkkommunikation (UMTS, GSM) verbessert werden, sondern durch Wlan. Dazu werden Wlan Hotspots, die aus mehreren Access Points bestehen können, am Rande des Weges dazu verwendet in kurzer Zeit große Mengen Daten zu übertragen (Abb. 2, Verbindung 1). Ein DTN Proxy entpackt dann die übertragenen Anfragen und Daten (Verbindung 2) und gibt die Antworten wieder an das Fahrzeug weiter (Verbindung 1). Solche Hotspots könnten dabei extra für Reisende von Raststätten und Tankstellen bereitgestellt werden, oder sie existieren schon, beispielsweise bei Cafés, Restaurants und Hotels [3].

In Tests [4] konnten dabei auch bei Geschwindigkeiten von 120 km/h (~33 m/s) noch Datenraten von 15 Mbit/s erreicht werden, wodurch bei einer Strecke von 2 km (60 s Verbindungsdauer) 110 Mbyte Daten übertragen werden konnten. Die hohe Reichweite von 2 km wird durch entsprechende Antennen und ohne besondere Technik erreicht, wenn keine Gebäude die Verbindung beeinträchtigen, was Zuhause für gewöhnlich der Fall ist, beim Einsatz entlang einer Autobahn jedoch nicht. Bei mehreren Teilnehmern verteilt sich aber die zur Verfügung stehende

Bandbreite auf diese [4]. Da die Tests bereits vor mehreren Jahren durchgeführt wurden (basierend auf IEEE 802.11g Hardware), sollten die Ergebnisse mit moderner Hardware nach IEEE 802.11n sowohl im Bereich der Reichweite als auch der Übertragungsgeschwindigkeit übertroffen werden können.

Um die benötigten Daten dabei in dem kurzen Zeitraum der Verbindung schnell und effizient übertragen zu können, werden die Daten nicht als TCP Stream übertragen sondern von einem lokalen DTN-Proxy in Bundles umgewandelt und als solche zu einem entfernten DTN Proxy (Internetgateway) übertragen. Dieser stellt für das DTN die Verbindung zum Internet her [2].

3.2.2 Aufbau mit lokalen DTN Proxys

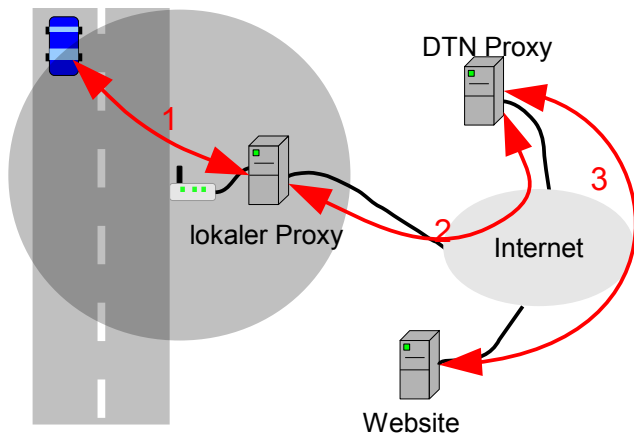


Abb. 3: Aufbau eines DTN mit lokalen Proxys

Um die Datenmenge, die während der Verbindung übertragen werden kann, weiter zu erhöhen wurde auch ein Ansatz mit lokalen DTN Proxys entwickelt. Diese befinden sich bei den Hotspots und funktionieren als Buffer. Innerhalb des Übertragungsfensters können dadurch Daten mit voller Geschwindigkeit zwischen Fahrzeug und Hotspot übertragen werden (Abb. 3, Verbindung 1) auch wenn die Internetanbindung des Hotspots langsamer ist (Verbindung 2). Der lokale Proxy überträgt die Daten dann an das Internetgateway auch nachdem die Verbindung zum Fahrzeug beendet wurde [3].

Besonders bei großen Datenmengen kann dieses System den Durchsatz erhöhen, erfordert jedoch eine Bevorzugung interaktiver Verbindungen und ihrer Daten, damit diese nicht in Warteschleifen auf ihren Upload warten müssen. Ein weiteres Problem ist, dass auf eine Authentifizierung beim lokalen Proxy verzichtet werden sollte, um Wartezeiten und Datenverkehr zu vermeiden. Der Proxy wird dadurch leicht angreifbar für DoS (Denial of Service) Attacken, bei denen einer oder mehrere Angreifer große Datenmengen auf den Hotspot laden, was sowohl bei der lokalen Datenspeicherung als auch bei der Datenübertragung über Verbindung 2 zu Überlastungen, beziehungsweise großen Verzögerungen, führen kann. Gleichzeitig kann auch schon die übertragene Datenmenge der normalen Nutzer dazu führen dass sich große Datenmengen im Buffer ansammeln und zu einer Verzögerung der Übertragung führen. Dies führt zu einer verspäteten Verarbeitung durch das Internetgateway und zu zusätzlicher Latenz bei der Übertragung und erschwert es den richtigen Ziel-Hotspot für die Bundles zu

wählen. Die Auswahl sollte abhängig sein von der noch zu erwartenden Verbindungsdauer zum momentanen Hotspot, der Auslastung der Internetverbindung und dem Füllstand des Buffers. Außerdem sollte es eine Rolle spielen wann eine Verbindung zum nächsten Hotspot hergestellt werden kann und wie dieser ausgelastet ist [3].

Dieses Routing-Verhalten könnte verbessert werden. Dazu müsste der mobile Proxy im Fahrzeug den Metadaten des Bundles auch Informationen über die geplante Route und voraussichtlichen Kontaktzeiten mit weiteren Hotspots beifügen [3]. Diese Daten zu erfassen beziehungsweise vorherzusagen, zeitnah an den Internetgateway zu übertragen und auszuwerten, ist sehr aufwendig und könnte die zur Verfügung stehende Bandbreite weiter verringern.

3.3 Testaufbau

Die Datenübertragung muss nicht über das Bundle-Protokoll stattfinden, auch andere Methoden sind möglich. Um diese miteinander zu vergleichen wurde ein Testaufbau verwendet der dem Aufbau ohne lokale Proxys entspricht (siehe 3.2.1).

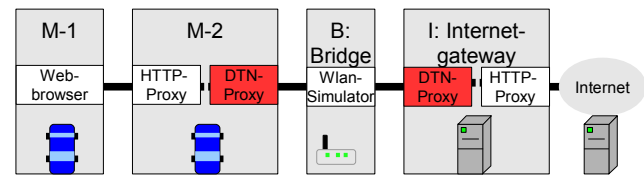


Abb. 4: Testaufbau (ohne lokale Proxys)

Der Aufbau (Abbildung 4) besteht aus vier Computern (Hosts) die mit Linux 2.4 betrieben werden, untereinander mit 100 Mbit/s schnellen full-duplex Ethernet-Verbindungen verbunden. Im praktischen Einsatz wäre M-1 und M-2 Teil des Systems im Fahrzeug, B entspräche dem Wlan Router und der Internetgateway wäre ein DTN Server irgendwo im Internet. Für den Test wurde M-1 und M-2 aufgeteilt, da dies ein Überwachen der ausgetauschten Daten erlaubt. Auf M-1 wird Firefox als der Webbrowser verwendet dessen Ladezeiten später verglichen werden. M-2 enthält den HTTP-Proxy an den M-1 die Seitenaufrufe sendet. Dieser sendet die Anfragen an den DTN-Proxy Host-intern weiter, der sie dann entsprechend verpackt und an Host B weitergibt. Auf Host B läuft ein Wlan-Simulator, der packet loss und Latenz eines Wlan mit unterschiedlicher Auslastung und Verbindungsqualität simulieren kann. Für dieses Netzwerk stellt Host I den Internetgateway dar. Auf diesem entpackt der DTN-Proxy die Anfragen und reicht sie an den HTTP-Proxy weiter. Dieser ruft die Website auf und gibt sie an den DTN-Proxy zurück. Die Internetanbindung von Host I ist dabei 155 Mbit/s schnell [2].

3.4 Zugriffsmethoden

Für den Vergleich des Bundle-Protokolls mit anderen Methoden zur Datenübertragung kommen vier verschiedene Programme mit unterschiedlicher Übertragungsart als DTN Proxy zum Einsatz. Auch der direkte Zugriff wurde mitgetestet.

- **Direkter Zugriff:** Hier werden keine Proxyserver verwendet und der Zugriff erfolgt direkt.
- **HTTP-Proxy:** Um die Unterschiede bei der Kommunikation über das Bundle-Protokoll genauer messen zu können, werden hier HTTP-Proxys (an der Stelle der DTN-Proxy) auf Host M-2 und Host I zwischengeschaltet. Das soll es ermöglichen zu unterscheiden welche Verzögerung durch die Bundle-Kommunikation und welche durch die zwischengeschalteten Proxys entsteht.
- **PCMP:** Bei dieser Zugriffsmethode wird das Persistent Connection Management Protocol (PCMP) des Drive-thru Projekts zwischen Host M-2 und Host I verwendet.
- **DTN:** Hierbei werden die Seitenaufrufe von einem DTN-HTTP-Proxy (dtnhttp) angenommen. Dieser gibt sie an den DTN-Proxy weiter. Die Datenübertragung zwischen den DTN-Proxys von M-2 und I findet dann nach dem Bundle-Protokoll statt.
- **DTN mit Prefetching:** Anstatt alle Anfragen des Webbrowsers einzeln abzuarbeiten, versucht I beim Prefetching mithilfe von wget auch alle in die angeforderte Website eingebetteten Objekte, beispielsweise Bilder, zu erkennen, herunterzuladen und in ein einziges Bundle-Paket zu packen. Dieses Paket wird dann an M-2 geschickt, der es entpackt und an den HTTP-Proxy weiterreicht. Dieser kann dann (im Optimalfall) alle Anfragen von M-1 nach weiteren Ressourcen direkt beantworten ohne weitere Anfragen bei I.

(Abb. 4) [3]

3.5 Geschwindigkeit

Um die einzelnen Methoden miteinander zu vergleichen wurden unterschiedliche Websites mit jeder Methode mehrmals aufgerufen. Manche Websites waren lokal auf dem Universitätsgelände, auf dem der Test stattfand, angesiedelt und hatten dementsprechend eine relativ geringe Paketumlaufzeit (Round Trip Time, RTT) von 1-10 ms. Andere waren weiter entfernt wodurch sich RTTs von bis zu 1 s ergaben. Die meisten Websites hatten jedoch RTTs von 10-100 ms. Gemessen wurde die Zeit von der ersten Anfrage durch Firefox bis zum letzten übertragenen Datenpaket zwischen M-1 und M-2 mithilfe von Ethereal [3].

Ohne künstliche Latenz durch B zeigte sowohl der HTTP-Proxy als auch der PCMP-Proxy eine Verzögerung von wenigen zehntel ms verglichen mit dem direkten Zugriff. Dies ist zwar messbar, dürfte aber nur von den wenigsten Anwendern wahrgenommen werden [3].

Bei Betrachtung der Messwerte mit dem DTN-Proxy zeigen sich jedoch durchweg Zugriffszeiten, die um ein vielfaches höher liegen als bei direktem Zugriff oder Zugriff über den HTTP- oder PCMP-Proxy. Diese brauchten beim Laden der 45 Objekte von

Ebay ~4,5 s, mit DTN-Proxy musste man 64 s auf die vollständige Seite warten. Eine der lokalen Seiten (TZI) mit nur 11 Objekten wurde ohne DTN-Proxy in weniger als 0,5 s geladen, mit DTN-Proxy dauerte es 17 s. Auch von den anderen Websites wurde mit DTN-Proxy keine in weniger als 10 s geladen. 6 der 12 getesteten Websites konnten bei direktem Zugriff (oder Zugriff über HTTP-/PCMP-Proxy) in weniger als 2 s geladen werden, während es bei Zugriff über den DTN-Proxy zwischen 10 s und 59 s benötigte um sie vollständig aufzubauen [3, Table II].

Dies ist darauf zurückzuführen, dass ohne Prefetching die Objekte der Website nacheinander vom Browser angefordert werden müssen. Dabei muss jedes mal gewartet werden bis das angeforderte Objekt übertragen wurde. Dann kann Host I es in ein Bundle verpacken und an M-2 zustellen, dessen DTN-Proxy es wieder entpackt und an den HTTP-Proxy zustellt. Dadurch dauert es je nach Website durchschnittlich 1–3 s um ein Objekt zu übertragen und entsprechend lange um die gesamte Website aufzubauen (die getesteten Websites hatten zwischen 6 und 65 Objekte) [3].

Beim DTN mit Prefetching ergaben sich bei manchen Zugriffen signifikant bessere Zeiten (26 s statt 72 s, 3 s statt 17 s) bei anderen jedoch keine oder geringere Unterschiede. Dies lag am unterschiedlichen Erfolg von wget die eingebetteten Objekte zu erkennen und herunterzuladen. Bei manchen Websites erkannte wget alle oder fast alle eingebetteten Objekte wodurch die Übertragungsdauer mehr als halbiert wurde. Bei anderen erkannte wget nur sehr wenige oder gar keine Objekte weswegen sich die Übertragungsdauer kaum änderte. Um die Übertragungszeiten des DTN mit Prefetching zu verbessern, sollte wget die Objekte genauso anfordern wie ein Webbrowser und diese über mehrere TCP Verbindungen laden anstatt über eine einzelne wie im Test [3].

Eine genauere Betrachtung der Kommunikation zwischen M-2 und I zeigt, dass ca. 80 % der Zeit für das Senden und Empfangen der Bundles verbraucht wird. Durch eine Optimierung der Implementierung des DTN-Proxys könnte diese Zeit verringert werden. Auch ein Start der Bundle-Übertragung bevor die Daten für das Bundle vollständig vorhanden und verpackt sind ist denkbar [3].

Tabelle 1. Auszug der getesteten Websites mit Zugriffszeit [3]

Seite	Direkt	HTTP-Proxy	DTN	DTN mit prefetch	Anzahl Objekte
TZI ¹	0,3 s	0,5 s	17,0 s	3,1 s	11
KDDI ²	12,8 s	10,0 s	72,6 s	26,2 s	37
Ebay ³	4,5 s	4,5 s	64,2 s	65,5 s	45
Apache ⁴	0,9 s	1,6 s	59,4 s	25,0 s	21

¹ <http://www.dmn.tzi.org/>

² <http://www.kddi.com/english/>

³ <http://www.ebay.com/>

⁴ <http://cocoon.apache.org/>

3.6 Overhead

Die Qualität des Prefetchings hat nicht nur einen Einfluss auf die Latenz bei der Übertragung der Websites. Von ihr ist auch die zu übertragende Datenmenge abhängig, da jedes Bundle auch einen gewissen Overhead hat. Die Übertragung der Bundles findet über TCP statt, aber jedes Bundle beinhaltet auch den HTTP Overhead und Informationen über das DTN. Zusammen ergeben diese einen Overhead von 100 bytes pro übertragenem Bundle, oder 20 %, da eine HTTP Anfrage im Normalfall 400-600 bytes groß ist. Findet Prefetching statt müssen zudem auch Name und Dateigröße der einzelnen Objekte übertragen werden [3].

Die übertragene Datenmenge von M-2 zu I dürfte sich dabei durch prefetching verringern lassen da hier im Idealfall nur noch eine einzelne Anfrage nötig ist. Dadurch sollte sich gegenüber den vielen einzelnen Anfragen mehr Volumen einsparen lassen als der Overhead des DTNs hinzufügt. Von I zu M-2 wird das kaum möglich sein, allerdings sollte es hier möglich sein durch Kompression den Overhead des DTNs auszugleichen [3].

3.7 Folgerung

Aus dem Test geht hervor, dass es möglich ist HTTP über das Bundle-Protokoll zu übertragen, aber die Performanz für den realen Einsatz zu schlecht ist. Hierfür müssten das Prefetching und die Effizienz der DTN-Proxys verbessert werden. Aber auch mit annehmbaren Übertragungszeiten würde das DTN dem Anwender langsamer vorkommen als bei direktem Zugriff. Beim DTN wird auf das Übertragen sämtlicher Objekte an den Internetgateway und des gesamten Bundles an M-2 (Abb. 4) gewartet, erst dann wird die Seite aufgebaut. Beim direkten Zugriff hingegen werden Objekte nach und nach dargestellt. Dies erscheint dem Benutzer für gewöhnlich schneller, auch wenn die vollständige Darstellung der Website die gleiche Zeit benötigt. Außerdem ist zu beachten, dass sich HTTP, solange es zustandslos verwendet wird, relativ einfach über ein DTN übertragen lässt. Interaktive Dienste (wie ein Chat) wären mit der momentanen DTN Implementation jedoch nicht möglich und sichere Verbindungen stellen auch ein großes Problem dar. Deswegen sollte auf das DTN nur zurückgegriffen werden, wenn eine konstante Verbindung nicht verfügbar ist. Abhängig von Signalstärke und Verbindungsschwankungen könnte die Auswahl des Netzwerks automatisch, ohne Eingriff des Benutzers, durch den DTN-Proxy erfolgen. Ein weitergeben der Verbindungsart vom Betriebssystem an die Anwendungen könnte es ermöglichen, dass diese ihr Verhalten in einem DTN anpassen. Dadurch sollten die Unterschiede zwischen den Übertragungsarten für den Benutzer verringert werden können, dieser aber auch auf Einschränkungen hingewiesen werden können wenn nötig [3]. So könnte beispielsweise ein Instant Messenger seinen Benutzer nicht abmelden, sondern ihm und seinem Kommunikationspartner anzeigen, dass er momentan keine dauerhafte Verbindung hat. Die Daten werden übertragen sobald wieder eine Netzwerkverbindung vorhanden ist. So könnte ein DTN eine nützliche Erweiterung unserer bisherigen Kommunikationsnetze darstellen.

4. Weitere Entwicklungen im Bereich DTN

4.1 HTTP in DTNs

4.1.1 MIME und HTTP

Ein wichtiger Faktor der Kommunikation heutzutage ist das Übertragen von Dateien, seien es nun Bilder, Dokumente oder Programme. Dies wird fast immer durch MIME (Multipurpose Internet Mail Extension) realisiert, das längst nicht mehr nur in Emails Anwendung findet, sondern auch zusammen mit HTTP verwendet wird. Durch MIME wird bei der Nachrichtenübertragung die Art der Nachricht beschrieben. Hierzu gehört ihr Typ (Content-Type, beispielsweise „text/plain“ oder „image/gif“) als auch die verwendete Kodierungsart, wobei eine Email mit Texten und Bildern aus mehreren, unterschiedlichen Content-Types besteht. Die mit MIME beschriebenen Daten werden dabei als Binär-Stream über das Übertragungsprotokoll HTTP übertragen. Dies wird selbst fast ausschließlich mit TCP als Transportprotokoll verwendet [5].

4.1.2 HTTP ohne TCP

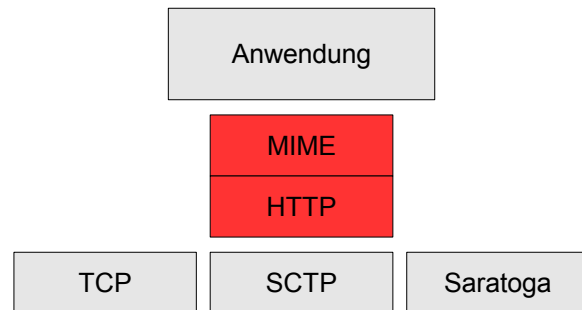


Abb. 5: HTTP und MIME zwischen Anwendung und Transportprotokoll

Für die Verwendung in DTNs ist TCP jedoch häufig ungeeignet, da hier andere Anforderungen an Transport und Routing gestellt werden als in normalen Netzwerken. Bei diesen kann mit einer dauerhaften, durchgehenden Verbindung der Kommunikationspartner ausgegangen werden. Bei einem DTN ist jedoch häufig keine durchgehende Verbindung möglich, oder die Kommunikationspartner sind nur zu unterschiedlichen Zeiten verfügbar. Auch unterstützen kleine Systeme (beispielsweise bei Sensornetzwerken) manchmal kein TCP, könnten aber trotzdem HTTP verarbeiten. Andere Netzwerke, beispielsweise lokale Netzwerke mit geringen Bit Error Rates und Latenzen, sind eigentlich gut für TCP geeignet, hier gibt es jedoch mittlerweile modernere Protokolle, die einige Vorteile gegenüber TCP aufweisen können beispielsweise SCTP (Stream Control Transmission Protocol). Um trotzdem die Übertragung von Daten in diesen Fällen zu ermöglichen, eignet sich HTTP als zustandsloses, von einem Transportprotokoll unabhängiges Übertragungsprotokoll. Die Anforderungen von HTTP an das Transportprotokoll sind dabei relativ gering und es könnten problemlos Protokolle wie Saratoga oder SCTP zur Übertragung von HTTP verwendet werden [5].

4.1.3 Eignung von HTTP für DTNs

Um Daten in einem DTN optimal zu übertragen, kann es auch nötig sein unterschiedliche Protokolle für die Subnetze zu verwenden. Auch das ist mit HTTP kein Problem, da es erlaubt

weitere „Content-*:“ Header einzuführen. So können mit „Content-Source:“ und „Content-Destination:“ die Quelle und das Ziel der Daten unabhängig vom Subnetz angegeben werden, während mit „Host:“ das Ziel innerhalb des Subnetzes gemeint ist [5].

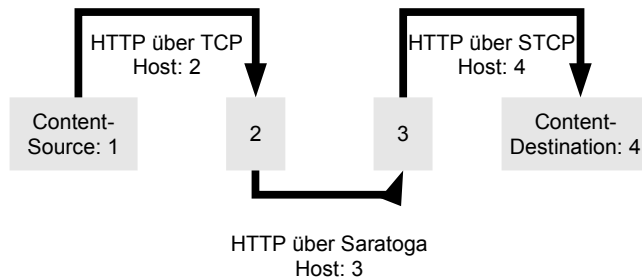


Abb. 6: HTTP in einem DTN über mehrere Transportprotokolle

Wichtig ist dabei, dass ein HTTP-Server Übertragungen mit unbekanntem Content-* Typ ablehnt. Dadurch können in einem Netzwerk auch normale HTTP-Daten neben den HTTP-DTN-Daten im Netzwerk existieren, da hier die DTN-Daten von den normalen HTTP-Servern abgelehnt und nur von den HTTP-DTN-Servern angenommen und verarbeitet werden [5].

Da HTTP von sich aus kein Ablaufdatum für die Daten enthält lassen sich damit auch große Distanzen (sowohl zeitlich als auch räumlich) überwinden, vorausgesetzt das Transportprotokoll unterstützt diese. Mit HTTP/1.1 werden noch mehr für DTNs nützliche Funktionen unterstützt. So ist es durch Pipelining möglich mehrere Objekte direkt nacheinander zu verschicken ohne nach jedem auf eine Antwort warten zu müssen [5].

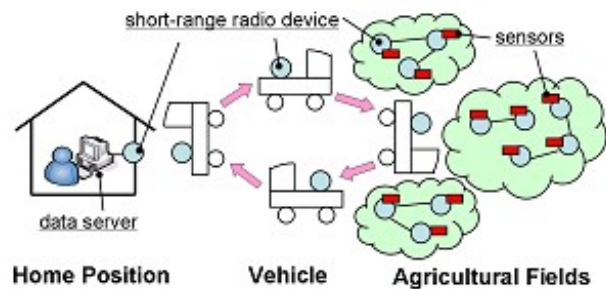
4.1.4 HTTP und das Bundle Protokoll

Bei der Verwendung von HTTP (mit MIME) anstelle des Bundle Protokolls in einem DTN bleiben viele Probleme bestehen. So ist es bei der Implementierung eines Ablaufdatums wichtig, die Zeit über den gesamten Übertragungsweg synchron zu halten. Auch die Probleme beim Routing in einem DTN bleiben gleich. Trotzdem hat HTTP einen großen Vorteil gegenüber dem Bundle-Protokoll. Durch die als Text spezifizierten Header ist es bei HTTP leichter möglich diese zu verändern und anzupassen. Beim Bundle-Protokoll ist diese Information im Binärformat gespeichert, dies erschwert die Anpassung des Protokolls [5].

4.1.5 Folgerung

HTTP als vom Transportprotokoll unabhängiges Übertragungsprotokoll sollte nicht nur zusammen mit TCP verwendet werden. Für viele Anwendungszwecke gibt es geeignetere Transportprotokolle, hierdurch wäre HTTP mit MIME für die Datenübertragung in einem DTN geeignet. Die Anwendungen sollten dabei HTTP (und MIME), unabhängig vom Transportprotokoll, verarbeiten können. Dies lässt Flexibilität bei der Wahl des Transportprotokolls und Netzwerkaufbaus zu.

4.2 DTNs in der Landwirtschaft



DTNs können auch in der Landwirtschaft benutzt werden. Um den Ertrag zu steigern und die Arbeit besser planen zu können, sind detaillierte Informationen über Temperatur sowie Luft- und Bodenfeuchtigkeit notwendig. Lässt man diese Informationen von Sensoren sammeln, muss die Übertragung der Daten zur Auswertung geklärt werden. Mobilfunkkommunikation ist zwar technisch möglich, jedoch sind hierbei die monatlichen Kosten nicht unerheblich. So können mehrere Sensoren für ein Feld nötig sein. Bei einer Anbindung dieser Sensoren über Mobilfunk müsste für jeden von ihnen eine Gebühr bezahlt werden, was in der Praxis häufig zu teuer ist [6]. Auch ist eine lückenlose Netzabdeckung in landwirtschaftlichen Gebieten nicht gewährleistet.

Um diese Kosten einzusparen kann ein DTN zum Sammeln dieser Daten eingesetzt werden. Hierbei kann die Übertragung zwar nicht so häufig stattfinden wie beim Mobilfunk, dafür fallen jedoch, neben den Anschaffungskosten, keine weiteren Kosten für die Datenübertragung an. Die Sensoren sind dabei mit Nahfunk ausgestattet, um ihre Daten kabellos an mobile Datenspeicher (wie beim Lappland-Experiment) zu übertragen. Diese Datenspeicher werden dabei entweder von den Arbeitern bewegt, oder an den Landmaschinen angebracht (Abb. 7). Die Landmaschinen sammeln dann automatisch Informationen während der Arbeit. Nach Beendigung dieser können die gesammelten Informationen dann übertragen und ausgewertet werden [6].

Eine solche Lösung kann vergleichsweise kostengünstig eingeführt und betrieben werden, erfordert aber mehr Aufwand als bei Übertragung über Mobilfunk da die Sensoren immer wieder abgefragt werden müssen. Abhängig vom Zeitintervall dieser Abfragen ist die Aktualität der Daten [6].

5. Zusammenfassung

Die betrachteten Experimente und Tests zeigen, dass DTNs ein Bereich sind in dem zwar geforscht wird und vieles theoretisch klar scheint, beispielsweise der Vorteil des Bundle-Protokolls beim DTN für die mobile Kommunikation (siehe Kapitel 3), in dem es aber auch einigen Platz für Optimierung gibt. Diese können sowohl praktischer (Verbesserung der Geschwindigkeit der DTN-Proxys und der Prefetching Effizienz) als auch theoretischer (Routing interner Emails) Art sein.

Auch zeigt sich, dass DTNs in vielen unterschiedlichen Bereichen hilfreich sein können. In Gebieten mit schwacher Infrastruktur zur Bereitstellung eines Internetzugangs und in Gebieten mit starker Infrastruktur zur Beseitigung der letzten Versorgungslücken und zur Entlastung der Netzwerke mit weniger Kapazität. Probleme

wie die Identifizierung des Zielknotens und das Routing haben die meisten DTNs gemeinsam. Auch die Auswahl des richtigen Protokolls zur Bündelung der Daten ist in einem DTN sehr wichtig. Funktionierte das Bundle-Protokoll in Lapland gut so zeigte sich trotzdem (Kapitel 3), dass es noch viel Optimierung bedarf, um es in zeit- und bandbreitenkritischeren Umgebungen einsetzen zu können.

Kaum Beachtung findet bei diesen Projekten für gewöhnlich das Transportprotokoll. Bei näherer Betrachtung sollte jedoch klar werden, dass hier, ein entsprechendes Protokoll auf Anwendungsebene vorausgesetzt, ein Wechsel auf ein anderes Protokoll sinnvoll und möglich sein könnte. Durch Wahl eines geeigneten Transportprotokolls, in einigen Bereichen gibt es gute Alternativen zu TCP, kann die Effizienz und Stabilität des DTNs erhöht werden.

Nicht zu vernachlässigen ist jedoch, dass immer mehr Einsatzgebiete für DTNs mittlerweile von Mobilfunk abgedeckt werden können. Aufgrund fortschreitender Entwicklung im Mobilfunkbereich, sowohl auf Technologie- als auch auf Kostenebene, könnte direkter Mobilfunk in Zukunft die dem DTN bevorzugte Alternative sein. Grund dafür sind niedrigere Latenzen und dauerhaft verfügbare Verbindungen mit stetig steigenden Übertragungsraten, während der Stromverbrauch und die Kosten immer niedriger werden. Dadurch kommt er für immer mehr Gebiete in Frage.

6. Quellen

- [1] Lindgren A. und Doria A.; „Experiences from Deploying a Real-Life DTN System“, IEEE, 2007
- [2] Doria A., Uden M. und Pandey D.P.; „Providing connectivity to the Saami nomadic community“, ThinkCycle, 2002
- [3] Ott J. und Kutscher D.; „Applying DTN to Mobile Internet Access: An Experiment with HTTP“, Technical Report TR-TZI-050701, 2005
- [4] Ott J. und Kutscher D.; „The „Drive-thru“ Architecture: WLAN-based Internet Access on the Road“, IEEE Semiannual Vehicular Technology Conference Mai 2004, 2004
- [5] Wood L., Holliday P., Daniel Floreani und Psaras I.; „Moving data in DTNs with HTTP and MIME“, IEEE, 2009
- [6] Ochiai H., Ishizuka H., Kawakami Y., Esaki H.; „A Field Experience on DTN-Based Sensor Data Gathering in Agricultural Scenarios“, IEEE Sensors 2010 Conference, 2010
- [7] http://www.nasa.gov/mission_pages/station/research/experiments/DTN.html, aufgerufen am 24.05.2011