WHY SEAMLESS? TOWARDS EXPLOITING WLAN-BASED INTERMITTENT CONNECTIVITY ON THE ROAD

JÖRG OTT DIRK KUTSCHER
{jo|dku}@tzi.uni-bremen.de

Technologiezentrum Informatik (TZI) Universität Bremen
Postfach 330440, 28334 Bremen, Germany

Abstract

This paper discusses new mobile usage scenarios for WLAN technologies and presents an architecture that is based on the notion of *intermittent connectivity* instead of *seamless connectivity*. The *Drive-thru Internet* approach is intended to support Internet applications of mobile users in environments where no permanent connectivity is available, a common case for nomadic users. We have chosen the extreme scenario of users in vehicles moving at high speed on the road and provide connectivity by means of WLAN access points. Our service architecture takes the transient character of local network access into account and provides for persistent transport connections and application layer mobility. From reviewing common Internet applications, we derive application-specific extensions to optimise various kinds of protocols and provide a concrete usage example. We also discuss the relation of Drive-thru Internet to technologies such as network layer mobility, authenticated network access, common WLAN hot spot setups, and WLAN roaming.

Keywords: IEEE 802.11, WLAN, hot spots, mobility, intermittent connectivity, roaming

1 Introduction

Seamless connectivity is a popular paradigm for mobile, wireless networking that describes an idealised service model for mobility protocols. During recent years, a broad range of wireless access technologies has been developed to provide the basis for Internet and VPN connectivity to mobile users. Most prominently, this includes on one hand wireless LAN (Wi-Fi) technologies for high performance coverage inside buildings or in limited outside areas; and on the other hand cellular telephony and data services (particularly GPRS and UMTS) offering vast geographic coverage at rather low performance.

Striving for ubiquitous connectivity for the mobile user, at least two (orthogonal) approaches are followed besides expanding the infrastructure for 3rd generation (and beyond) cellular networks: 1) Hybrid networks are designed to optimise network access for mobile users at any given time, providing for seamless roaming between e.g. GPRS/UMTS networks and WLANs thus allowing for continuous connectivity [Lei01], as also pursued under the 4G umbrella. 2) Ad-hoc networking between mobile devices is used to extend the reach of existing network infrastructures and thus increase the coverage areas [fle03] [BFW03].

Despite those efforts and the already very extensive 2/2.5G cellular coverage, ubiquity is yet to be achieved, seamlessness far from reality. Personal observations by the authors from travelling in various countries (particularly in Germany but also elsewhere in Europe and in the US) seem to suggest that intermittent connectivity is the rule rather than the exception: even main roads (such as highways and autobahns) and railway lines still suffer from gaps in connectivity, not to mention more remote areas in the countryside.[1] Broad cellular network

[1] For coverage maps, refer e.g. to http://www.gsmworld.com/roaming/gsminfo/index.shtml. Note that coverage experienced in practice is less than these rather optimistic maps indicate.

coverage is usually the only option for the wide area but is expensive for service providers to offer. And even if available, people often choose not to use cellular wide area connectivity because of its high cost (and relatively poor performance).[Shi03]

We have developed a networking architecture that accepts the intermittent nature of connectivity as a matter of fact and builds support for mobile users on top. In our *Drive-thru Internet* project[2] [OK04a], we investigate providing WLAN-based connectivity on the road, particularly on highways and on the autobahn with WLAN hot spots located at the roadside (e.g. at petrol stations, in rest areas, etc.). We have chosen WLAN technology because of the achievable performance, the low cost for the necessary equipment (and thus the acceptable overall investment), and because we just do not require a large range of connectivity. Furthermore, WLAN hot spots may be set up independently at minimal effort (which is already being done for fixed Internet access in several countries [Def03] [ST03]). Finally, our approach is able to leverage such existing WLAN infrastructures for which *Drive-thru Internet* services may be built as an add-on. The system architecture does not require particular support from ISPs nor even from the WLAN hot spots, with the exception of reasonable antenna placement and the use of suitable (outdoor) equipment, thereby allowing for incremental deployment.

In this paper, we provide an short introduction to the Drive-thru Internet concept and the practically experienced connectivity in section 2. We discuss intermittent connectivity and related work as background in section 3 and review typical Internet applications with respect to their use in a Drive-thru environment in section 4. Based upon these observations, we present our Drive-thru architecture and discuss technical solutions for enabling Drive-thru Internet access in section 5; an example for Internet application support is given in section 6. Section 7 concludes this paper with a summary and also highlights key issues to be resolved in the next stages to enable real-world deployment.

2 Drive-thru Internet

Figure 1 shows the basic elements of the Drive-thru system architecture (most of which we will address in section 5): Basic Internet access is provided by connectivity clouds (or islands), each established by one or more wireless LAN access points. Several (adjacent) clouds may be interconnected directly but connectivity between clouds will usually not be continuous. The connectivity clouds are independently managed, so that we expect different ISPs and access networks to be chosen and private address spaces and NATs to prevail.[3]

Vehicles travelling along the road will pass through these connectivity islands: they detect the presence of a wireless LAN, associate with the respective access point, perform some form of authentication and IP auto-configuration[4], and are then able to access hosts in the Internet. This connectivity period will last until the wireless LAN signal disappears – a duration that may be prolonged by multiple inter-connected access points using WLAN-based hand-over procedures or, in a more sophisticated approach, even by forming ad-hoc networks between vehicles as discussed e.g. in the FleetNet [fle03] and MultiNet [CBB04] projects.

We have performed a set of measurements with different configurations on highways and freeways that have helped to assess the characteristics of Drive-thru Internet access at relatively high speeds. The detailed results of our measurements so far are described in [OK04a] and [OK04b]. Besides validating the general feasibility, these measurements were targeted at analysing the network characteristics for the Drive-thru scenario and at evaluating the performance of transport protocols such as UDP and TCP under these conditions.

Both IEEE 802.11b (up to 11 Mbit/s) and IEEE 802.11g (up to 54 MBit/s) WLAN technology were used. The IEEE 802.11b test series have shown promising results for both UDP and TCP measurements. We have seen that the production phase (the phase in a Drive-thru session

[2] http://www.drive-thru-internet.org/

[3] Working with these assumptions is a prerequisite for targeting any real-world deployment.

[4] Our measurements reported below have aimed at the gross data exchange rate and hence have assumed static IP address configuration and have not considered WLAN authentication.

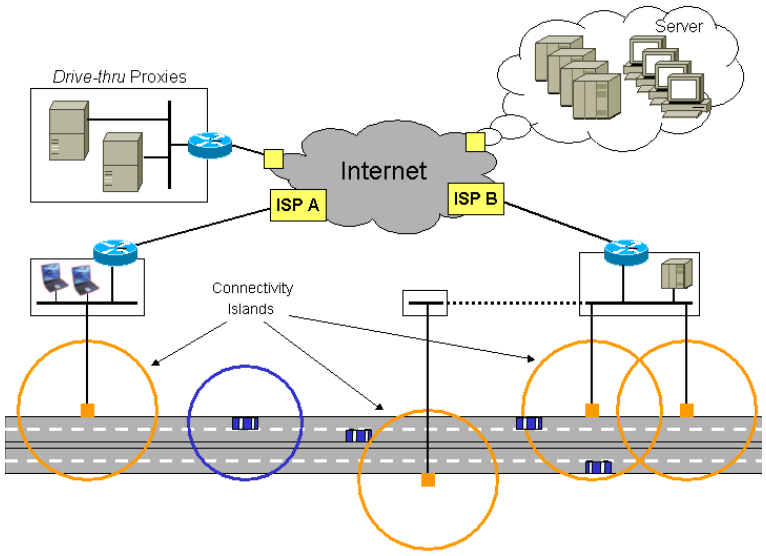

Figure 1: Architecture Overview

providing stable connectivity) allows for a throughput of up to 5 Mbit/s (mobile node sending) and up to 3.8 Mbit/s (mobile node receiving). The overall range of the connectivity area was about 500-600m, and, at 120 km/h, we achieved a UDP goodput (cumulative number of transmitted bytes) of almost 7 Mbytes when sending from mobile to fixed and a cumulative goodput of 2.4 MBytes when sending from fixed to mobile [OK04a]. The results have shown that although the overall range of the connectivity area is quite large, there are significant differences in transmission characteristics when passing through a connectivity cloud.

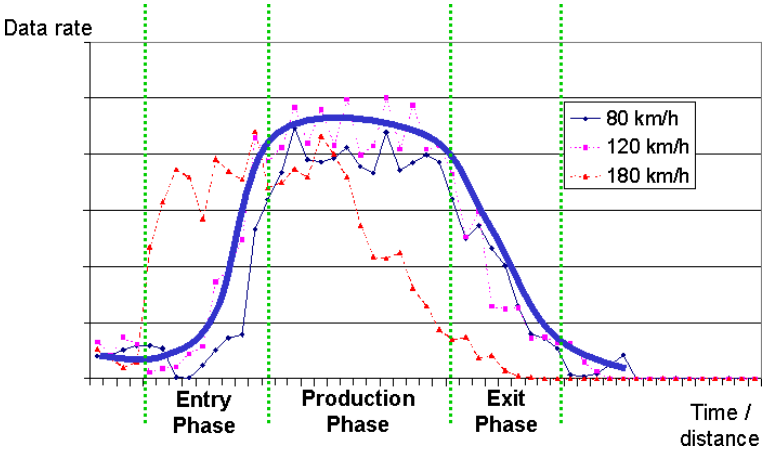

Figure 2: Different Phases of WLAN Access (UDP)

Our UDP measurement results suggest to subdivide a Drive-thru session into three distinct phases as depicted in figure 2. The production phase allows for a high sending rate that is close to the maximum throughput that we were able to obtain under laboratory conditions. In the entry and exit phases, the throughput is lower due to a higher number of lost packets, link-layer retransmissions, and the wireless hardware's switching to lower 802.11b sending rates. Never-

theless, a limited form of communication was possible that may be exploited for administrative functions.

For TCP, we have largely observed similar transmission characteristics, i.e., a significant difference in throughput for the entry, production and exit phases. At 120 km/h, we have achieved a temporary maximum throughput of almost 4.5 Mbit/s and a cumulative throughput in a single Drive-thru session of almost 5 MBytes when sending from fixed to mobile, i.e., both the maximum throughput and the cumulative throughput per session are higher than for UDP with the same configuration. In summary, TCP showed a good overall performance and was able to adapt to the varying transmission characteristics sufficiently well.

While our IEEE 802.11b measurements were primarily intended as a proof of concept, our IEEE 802.11g aimed at assessing the full communication capabilities of wireless LAN on the road. Therefore, for our IEEE 802.11g measurement series, we have significantly improved our test equipment and have deployed a high-gain antenna at the access point and a better antenna for the mobile system. The results have been very convincing. Similarly to our first measurement series, we have identified different phases of connectivity. However, the better radio hardware has resulted in a much larger extension of a single Drive-thru cloud. We have observed areas of useful connectivity with a diameter of more than 2.5 kilometres (i.e. 75 seconds at 120km/h). In addition, the relative increase in transmission rates and throughput was rather moderate compared to our first test series. We have observed IEEE 802.11 transmission rates of 1, 2, 5.5, 11, 22, 48, and 54 Mbit/s, depending on the distance to the access point and the signal quality. Consequently, we have observed a significant higher maximum throughput of some 15 Mbit/s and cumulative throughput of up to 110 MBytes at 80km/h on a highway, and a constant volume of 30+ MB at 120km/h on the autobahn. One important observation was that TCP has been able to adapt better to the varying maximum capacity of the WLAN link and has thus performed dramatically better than UDP in all of our IEEE 802.11g measurements.

Overall, we conclude that WLANs are well-suited to provide powerful connectivity to mobile users for relatively short time intervals. Active data communications should be limited to the production phase to avoid WLAN service degradation for all users concurrently passing through a Drive-thru cloud. The entry phase is deemed sufficient for auto-configuration and authentication purposes.[OK04a] This requires an appropriate architecture to support current and future applications.

3 Intermittent Connectivity

Intermittent (and variable) connectivity as observed in a Drive-thru Internet scenario is quite common to a (mobile) user today. For example, a user with a laptop computer or PDA may access the Internet from home and from the office; furthermore, when travelling, she may use WLAN hot spots or hotel networks to obtain connectivity in other places. That is, the user experiences alternating periods with and without being connected to the network. Those periods are of relatively long duration (usually at least several minutes) and allow – actually: require – the user to explicitly (manually) set up and configure the respective networking connections as well as to suspend/resume the operation of her Internet applications. While cumbersome, this overall procedure is generally accepted as the modus operandi for mobile users. This way of dealing with network access is workable as long as the duration of available connectivity is sufficiently long compared to the time required to (manually) establish connectivity. The interim disconnected periods are often less noticed because the user has her laptop turned off when she does not expect connectivity or would not be able to use it anyway (e.g. because of driving).

When network access is needed but no powerful network infrastructure is available, the disconnection periods may be reduced by including cellular networks as a backup – again, however, usually requiring manual setup and configuration. As mentioned in the introduction, one improvement is to enable seamless roaming between multiple (wireless) networks and providers (without manual user intervention) [ALMR01] [ZGGZ03] [PKH$^+$00]. But even if such mech-

anisms are ultimately deployed on a broader scale, the resulting services may turn out not to be (perceived as) ubiquitous: because full coverage is yet to be achieved and, even if it was, bandwidth may be deemed insufficient in areas with only cellular connectivity. For either case, transport and application protocols – and ultimately the users – have to deal with intermittent or at least highly variable connectivity.

The connectivity characteristics we obtain in the Drive-thru Internet environment are, in fact, very similar, just the time scales differ. Connectivity periods are much shorter – as short as only a few seconds –, disconnection times are arbitrary, lasting from a few seconds or minutes to hours or days. Obviously, human users are not really able to deal with such connectivity patterns manually so that suitable infrastructure and end system support is required to make effective use of this sporadic, short-lived connectivity.

In recent years, networking with intermittent connectivity has been a research aspect in several areas. For pervasive computing, *disconnected operation* of devices has been studied for mobile personal appliances (such as laptops and PDAs), with a strong focus on data synchronisation (file systems, calendars, off-line use of web browsers, and email among others) [KS92] [All02] [LGKT02].

Protocol and architecture aspects are focused on in (deep) space communications where communication paths between network elements may not be constantly available due to planetary constellations or because satellites may disappear behind the horizon, leading to interruptions for arbitrary time scales. Furthermore, pure signal propagation delays (from several seconds to hours and days) pose another obstacle to efficient communications. Such conditions render conventional protocol design unusable, so that special communication architectures and protocols have been developed that focus on asynchronous communications pushing responsibility to intermediary nodes [Fal03].[5]

Other approaches deviating from the traditional synchronous, end-to-end communication paradigm are used to enhance communication performance in other extreme networking environments [BKG+01]: to alleviate deficiencies resulting from propagation delay and highly asymmetric path characteristics in satellite communications; or to deal with connectivity interruptions and high link error rates in cellular networks [BB94] [BSAK95] [BS97] [Haa95] [MB98]. In all these cases, intermediary nodes (not necessarily routers) take up additional functionality to conceal certain properties of some of the involved links – sometimes breaking the end-to-end semantics of the applications' TCP connections – and ultimately enable or improve communications. Further performance enhancements may be implemented at the application layer, including HTTP prefetching and caching as one example (e.g. [PM96] and [Dav02]).

Our Drive-thru Internet approach also exhibits characteristics that preclude the unmodified application of the end-to-end paradigm: the connectivity periods are too short for many applications to complete their respective operation, so they would need to be able to sustain operation across disconnection periods of unpredictable length. Issues arise beginning with changing IP addresses, but even if those were kept stable by applying IP layer mobility mechanisms [(ed02], transport protocols are unlikely to persist through the outages – and, even if they did (e.g. by using mechanisms such as freeze TCP [GMP00]) application protocols on top are also usually not designed to deal with longer interruptions. Therefore, in the Drive-thru approach we employ a combination of mechanisms known from extreme networking environments discussed above to manage node mobility. As applications and their susceptibility to intermittent connectivity differ, we briefly investigate the different classes of applications before turning our attention to the Drive-thru architecture in section 5.

4 Drive-thru Applications

Looking at current uses of the Internet, roughly two ways of user interactions can be distinguished: continuous communications such as IP telephony, media streaming, and the like on one

[5] Refer also to http://www.dtnrg.org/, http://www.ipnsig.org/, and http://www.scps.org/.

hand and more transaction-based (request-response-style) information access, including e-mail, data synchronisation, and file sharing tools on the other. The former require largely persistent connectivity for the lifetime of the application instance (e.g. an IP phone call) and hence their usage is limited to the connectivity period – which may seriously limit their usability, at least in their present form. The latter may only need to complete individual transactions (e.g. sending a single e-mail message) during one connectivity period and may continue their operation in the next connectivity window and hence are much better suited for this kind of environment. Database/file access, web browsing, and messaging/chat are examples for applications somewhere in-between, much dependent on the actual user behaviour.

In the following, we organise Internet applications into five functional categories from a connectivity perspective – the first four classes comprising unmodified existing, the fifth one new Drive-thru-aware applications. We use the properties described below to characterise those application classes in more detail with respect to their communication patterns – based upon which we devise application-specific support in our Drive-thru architecture.

Direction. Primary direction of data transmission, i.e., fixed to mobile or mobile to fixed.

Initiative. Who initiates a transaction (mobile or fixed party).

Transfer mode. Push (initiator = data source) vs. pull (initiator = data sink) vs. bidirectional.

Interaction style. How do the application entities interact? *One-shot* is a single (possibly idempotent) transaction with no relationship to others; *session-oriented* indicates that at least two (successive) transactions are logically linked and that some communication session is set up and torn down (explicitly). Each transaction may consist of one or more message exchanges (*operations*) that together make up a logical application function. We use the term *continuous data exchange* to denote those case in which logical transactions in a communication relationship cannot clearly be identified.

Duration. How much time does a single transaction (or a complete communication session) take? In particular, can such a transaction complete within a single connectivity cloud?

Recoverability. Is it possible to resume a failed (or interrupted) transaction, or must be it be repeated?

4.1 Asynchronous Applications

Many popular Internet applications rely on an asynchronous service model, namely e-mail transport that is based on the store-and-forward principle between any number of mail servers until the e-mail is received by the target mail server and deposited in the recipient's incoming mailbox. The user either interactively operates on her mailbox in *online mode* (e.g. via IMAP4) or simply downloads all available mails in *offline mode* (via IMAP4 or POP3). While the end user experience for e-mail transmission is asynchronous, the hop-by-hop transmission of each individual e-mail message using SMTP itself *does* rely on synchronous communication. The same applies to the user retrieving or accessing mails from her mail server.

For sending e-mail messages, the *direction* is from mobile to fixed, and it is also the mobile system that initiates a transaction. Message transfer can be characterised as a *push* application (from mobile to fixed). Typically, each messages leads to a one-shot transaction or, for transmitting multiple mails, to a sequence of transactions in a new SMTP session. Each transaction requires at least three operations, for sender authentication extra ones may be required. The *duration* obviously depends on the message size (and the data rate, of course). Interrupted message transmissions cannot be resumed but must be restarted all over.

For both offline and online e-mail access, the primary *direction* is from fixed to mobile, and the mobile system initiates transactions. The *transfer mode* is *pull mode*. The *transaction style* can be characterised as a *sequence of transactions*, and the *transaction duration* is again dependent on the size of individual messages and the data rate. With respect to *recoverability*, a failed

transaction cannot be resumed. For retrieving a single message within an existing session, a single transaction is required (for IMAP4, the retrieval may be split-up into several transactions, e.g., for multi-part MIME messages).

Apparently, e-mail transmission is well-suited for Drive-thru environments as no permanent connectivity is required beyond the message forwarding transaction to a next-hop SMTP server. For accessing a mailbox from a Drive-thru environment, offline mode is preferred because sessions can be initiated automatically and no user interaction is required for the entire retrieval process (which may be suspended and resumed on a per-message basis). Online mode requires a persistent connection between user agent and mail server and needs additional support from the Drive-thru infrastructure, as we will discuss in section 6.

4.2 World Wide Web

Web-based applications, i.e., web browsing and other HTTP-based applications, are generally transaction-based (due to HTTP's request-response semantics). For Web browsing, a web page may consist of several resources (HTML document, images, style sheets, etc.), each of which is transmitted in an independent transaction. In general, (GET) transactions are idempotent, however the concept of HTTP authorisation and cookies introduces the notion of sessions, i.e., a series of transactions that belong to a common context. An HTTP request is issued by a user agent and is sent to an origin server – over an end-to-end terminated TCP connection. HTTP also provides the notion of proxy servers – non-transparent intermediaries that forward requests and may provide additional services such as caching (which may as well be done locally in the client).

For most Web-based applications, the primary *direction* is from fixed to mobile, and it is the mobile system that initiates a transaction. However the initiator, i.e. the mobile client, cannot only *pull* from but also *push* data to a server on a per-transaction basis. HTTP may operate in one-shot as well as in session mode, each transaction consists of exactly one request-response pair. The *duration* of a transaction is obviously dependent on the data volume but may also be influenced by other factors, such as processing delay at the origin server. The duration to retrieve an entire web pages heavily depends on the number of objects contained in the page and their respective sizes. With respect to *recoverability*, a failed transaction can be resumed using the HTTP 1.1 `Range` header but this mechanism is not supported by all implementations; restarting a retrieval at the level of individual resources (objects) is always possible.

The transaction-based nature of HTTP communications is reasonably well suited to the Drive-thru environment as it provides fine-grained interruption plus recovery features. Still, the intermittent nature of connectivity does only provide a limited form of Web access, because interactive web browsing is only possible during (potentially rather short) connectivity phases. In addition, delays caused by the end-to-end communication between user agent and origin server may further limit the duration of available connectivity phases. Therefore, some present solutions for web access in fast moving vehicles where seamless connectivity cannot be sustained do not provide direct web access at all. Instead, on-board proxy servers provide a (selected) set of cached resources that can be accessed by user agents, and the cache is only updated when sufficient connectivity can be established (e.g., as currently implemented by the Clic TGV service [TMC04]). The Drive-thru architecture may provide further support to HTTP-based applications, drawing on experiences from the past (e.g. [PM94] [PM96] [Dav02]) and adopting them accordingly.

4.3 Audio/Video Communications

Interactive multimedia communication such as IP telephony and conferencing is based on the concept of a communication session involving two or more parties. Fundamental characteristics of these sessions include continuous media (mostly audio and video) transmission and (usually) a high degree of interactivity. This kind of audiovisual communication requires an environment

with reasonably constant data rates and low round-trip times and, most important, continuous connectivity. With its limited connectivity, a Drive-thru scenario environment is far from ideal interactive multimedia communication, as interactions are only possible during the short connectivity phases. With the existing 2G and 3G cellular networks readily providing voice (and video) communications, it is not particular attractive to develop a Drive-thru-based alternative – hence we will not consider this idea further.

However, at least two potentially interesting multimedia applications can be identified: the transmission of short audio (and video) talk spurts as *asynchronous push-to-talk* service where long-term sessions and interactive communication are not required. Furthermore, various flavours of media streaming are conceivable in a Drive-thru environment: short presentations, such as commercials or movie trailers, may be streamed within a single connectivity cloud, preferably from a local server. For longer-lasting real-time streaming sessions further support is required from a Drive-thru infrastructure (buffering large media volumes, tolerance for intermittent connectivity); they may eventually appear rather as a multimedia data download application.

The transmission of multimedia data can be bidirectional and each side can initiate transactions, for multimedia streaming the *transmission direction* is fixed to mobile and it is usually the mobile system that initiates a transaction. *Push-to-talk transmission* is certainly a *push application*, while multimedia streaming is generally a *pull application*. *Push-to-talk* may operate in one-shot mode as well as in session mode. The duration is typically limited by the size of the talk-spurt. For multimedia streaming, a transaction can be the request to initiate a streaming session. While the transaction duration itself is rather short, the streaming session *can* be long-lived, e.g., for real-time streaming the duration correlates to the real-time duration of the content. In both cases, failed or interrupted transactions may be resumed (or simply repeated) through an appropriate session control protocol.

4.4 Distributed Object Synchronisation

In mobile office scenarios, access to remote file system, e.g., access to enterprise file servers, is often used in conjunction with VPN tunnels to allow remote staff to access company resources in the same way as when working locally. Distributed file systems are one representative example of distributed object synchronisation, others include distributed data bases and calendaring systems. They are typically based on synchronous communication and impose strong requirements on latency and reliability. In essence, traditional distributed files systems such as NFS do require seamless connectivity and are therefore not particularly well suited to Drive-thru environments.

Other file systems such as the Coda file system[6] are explicitly targeted at supporting disconnected operation and WebDAV technologies may be used to implement Web-based distributed file systems that also support disconnected operation. In general, replicated file systems with support for partial synchronisation appear to be more suitable for Drive-thru scenarios than classical ones such as NFS or SMB. Still, the relatively short connectivity phases may impose problems with respect to incomplete synchronisation, essentially depending on the data volume to be exchanged for synchronisation – which may not be an issue for other distributed applications.

Distributed file system protocols typically employ *bidirectional* transmission. It depends on the specific protocol, who actually initiates transactions, e.g., for WebDAV-based solutions, this will be the client, i.e., the mobile system. With respect to file system synchronisation, this application can be characterised as *bidirectional* application as both *read* and *write* operations need to be supported. Other aspects such as recoverability strongly depend on the specific protocol in use.

As disconnected operation is already supported by such applications, the major issue to successful utilisation of distributed systems in a Drive-thru environment is synchronising the distributed application protocol with the connectivity phases when passing through a cloud. Further support depends, again, on the actual protocol in use.

[6] http://www.coda.cs.cmu.edu/, [KS92]

4.5 New Applications

In addition to these traditional types of Internet applications, a number further applications are specifically interesting in the automotive environment. E.g., telematic services for cars is a class of applications where a service provider typically distributes navigation and traffic information to cars. This information is processed by on-board navigation computers to optimise the routing. Some existing systems rely on GSM technologies for communicating with corresponding service centres, e.g., the AutoPC technology[7] developed by Citroën, Microsoft and Clarion. Services may be localised, and requests may provide car-specific location information. For telematic applications, communication is typically initiated by the in-car computer, and the primary data transmission is from fixed to mobile (*pull*) but also *push* usage scenarios are possible. Usually, navigation transactions will be short and rare (e.g. to request the position of a particular point of interest), while for distributing traffic information, the service centre may periodically send status updates and event notifications. The *transaction style* may be *one-shot* but there may also be sessions that provide a *sequence of transactions*. The *duration* is expected to be short because of the low data volumes, but may also be dependent on processing time at servers, e.g., for navigation requests. The *recoverability* of transactions depends on the specific implementation but information requests are usually idempotent and so are traffic updates.

Another future application class is collecting telemetric information such as car status data, observed road and traffic conditions etc. and periodically transmitting the data from the car to some service centre. For example, in order to enhance the accuracy of traffic information broadcasts, a service centre could receive individual, anonymised status information from cars about their current position, current speed etc. The primary *transmission direction* is mobile to fixed and transaction will be initiated by the mobile system. Transmission of telemetric information is a *push* application, and the information will be transmitted in a *sequence of transactions*. The *transmission duration* in mainly dependent on the data size. Obviously, both applications may be combined and also inter-vehicle communications – as studied e.g. in the FleetNet project[8] – may be exploited (e.g. as early warning in case of emergencies).

These applications are particularly suited to the Drive-thru environment because they do not require seamless connectivity. Instead, it is possible to suspend the transaction of requesting traffic information or the transmission of a batch of measured road and traffic data until a Drive-thru cloud has been reached. In addition, the amount of data is typically not very large, i.e., the transactions can probably complete in a single Drive-thru session.

4.6 Implications

As discussed above, only few applications (and application protocols) are suitable "as is" for use in a Drive-thru environment or otherwise intermittently connected network. Some applications – such as media streaming and interactive real-time conversations – rely on permanent connectivity and hence are only of limited use in our scenario. Fortunately, particularly the latter one is well-addressed by other technologies (e.g. analogue or digital broadcasting and cellular phones). For the remaining applications – that are probably the ones mostly used in today's Internet – we have devised a system architecture that supports their use in the presence of intermittent connectivity as much as possible.

5 Drive-Thru Architecture

In order to enable useful communication in these environments of "extreme" intermittent connectivity and mobility, we have taken an approach that does *not* attempt to provide seamless connectivity at the network layer but takes interruptions, mobility, IP stack reconfiguration, etc. into account and introduces a connectivity management service *above* the transport layer. The

[7] http://www.citroen.de/de/home_p197.html
[8] http://www.fleetnet.org/

fundamental idea of the Drive-thru architecture is to enhance the concept of connection splitting for the purpose of concealing the above characteristics of a Drive-thru environment. Two dedicated entities are introduced: *Drive-thru clients* on the mobile node and *Drive-thru proxies* in the fixed Internet. They operate peer-wise and relay transport and application layer sessions on behalf of mobile application instances (usually clients) and the corresponding peers (usually servers) in the fixed network. Drive-thru clients and proxies maintain connection state and can thus provide *persistent connections* that survive the loss of connectivity for very long periods, changes of IP addresses, and the like. The Drive-thru clients are also responsible for detecting WLAN access, initiating auto-configuration and authentication, and providing hints about the Drive-thru connectivity phases; the Drive-thru proxies authenticate clients and provide additional resources in the fixed network. Such a setting is depicted in figure 3. Furthermore, a Drive-thru PEP (performance enhancing proxy) may be placed in a connectivity island (e.g. in or close to the router in figure 3) to separate the characteristics of the WLAN from the access link and backbone network and enable quick adaptation of TCP connections to the highly variable Drive-thru conditions.[OK04b]

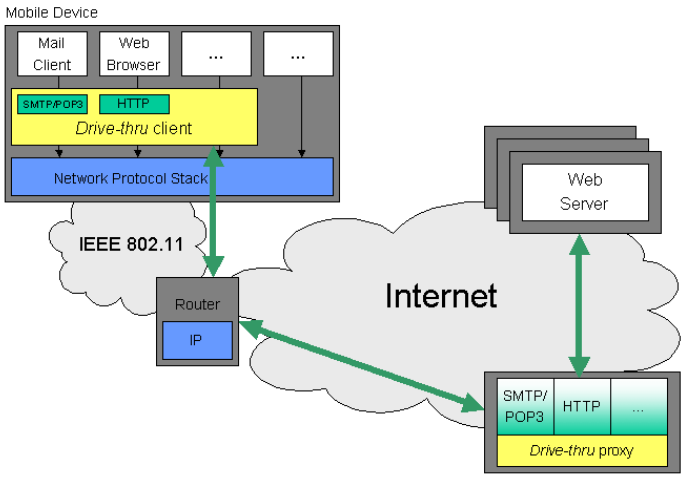

Figure 3: Drive-thru architecture overview

The *Persistent Connection Management Protocol (PCMP)* [OK04b] runs between Drive-thru client and proxy and allows bidirectional one-shot and session-oriented communications to be suspended/interrupted and later resumed without loss of information, independent of transaction boundaries. Thus, PCMP helps to eliminate the impact of transaction or session durations on the usability of applications for arbitrary (push, pull, bidirectional) interaction styles. As in the Drive-thru architecture, mobility management is actively performed in the mobile node, the Drive-thru client is responsible for re-establishing underlying transport connections after a connectivity interruption. Therefore, supporting communications initiated by the mobile party is relatively straightforward given minimal support by the Drive-thru client. In contrast, communication activities originating from arbitrary entities in the fixed network have no knowledge about the mobile node's reachability and are likely to fail in most cases unless specific support is added by means of application layer gateways/proxies. For the time being, we concentrate our efforts on the former (mobile-node-initiated applications) and limit support for the latter to cases in which the application architecture itself supports the concept of intermediaries.

The Drive-thru entities above contribute to application-specific support as follows:

- The *Drive-thru client* may act as an application layer gateway for different application layer protocols such as HTTP, SMTP, POP3, etc. It implements a persistent connection management protocol that is employed for the communication with the Drive-thru proxy in the fixed network. On top, a plug-in mechanism is provided to add application-protocol-specific support as needed.

  For push-style applications, such a Drive-thru client extension locally accepts data to be pushed and initiates the actual transactions as soon as connectivity becomes available. For pull-style applications, the Drive-thru clients may aggregate requests while not connected. The client may initiate transactions as batches when connected again and receive incoming data which is either forwarded immediately to application or cached for later use upon application request. For bidirectional applications, protocol-specific schemes may combine the above with other synchronisation mechanisms.

- The *Drive-thru proxy* is the counter-part of the Drive-thru client in the fixed network and conceals the mobile node's temporary unavailability from the corresponding (fixed) application peers (e.g. web or mail servers). The Drive-thru proxy follows an application-extensible proxy approach similar to the Drive-thru client.

  The Drive-thru proxies receives (batches of) requests from the Drive-thru client and executes these one by one, regardless of whether or not the latter remains connected. It accumulates data and requests/responses from peers in the fixed network and forwards these information bundles to the Drive-thru client at the next opportunity. It may provide heuristics to anticipate future actions of the mobile user (e.g. while accessing web pages), pro-actively carry out those actions as far as possible (e.g. pre-fetching web pages), and then push the contents to the mobile node.

  For push-style applications, the Drive-thru proxy may temporarily store the data to avoid that the peer in the fixed network becomes the bottleneck in the conversation. For pull-style applications, the Drive-thru proxy may buffer incoming data from the fixed peer while the mobile node is disconnected and forward this data (as batches) whenever the mobile node becomes reachable again. For bidirectional applications, a combination of the above mechanisms may be used.

The mobile application peers (e.g. web browser and mail client) are the user's unmodified standard applications. They may need to be configurable to communicate with the Drive-thru client (e.g. as a web proxy). During times without connectivity, user requests will be queued (and the applications kept on hold if possible), results will be delivered during the next connectivity cloud. The fixed application peers do not even need a re-configuration; they remain entirely unchanged. They communicate with the Drive-thru proxy just as they do with any other peer. It is the Drive-thru entities' responsibility to preserve end-to-end semantics of application layer protocols (e.g. successful submission of an e-mail) as much as possible, possibly requiring additional mechanisms.

6 Drive-thru Application Example

This section outlines the operation of the Drive-thru architecture for a particular application example: a mobile user travelling in a car sending and retrieving email messages. First, we focus on the process of establishing connectivity when entering a Drive-thru cloud; this process can be divided into four steps. Subsequently, we address the application-specific protocol operation.

1. IEEE 802.11 association
   The first step is always the association of the WLAN devices, e.g., the WLAN adapter on the mobile node and the access point. As soon as the mobile node enters an access point's range, the 802.11 association process begins. For the Drive-thru scenario, we assume

a typical WLAN hot spot configuration, i.e., WEP (and WPA) is disabled and ESSID broadcasting is enabled. The IEEE 802.11 association is the actual *network attachment*. In order to utilise the connectivity phases efficiently, it is important to perform the detection of network access quickly. More recently, standardisation work is on its way on automatically detecting network attachments [Abo04] which is likely to help automating the handling of intermittent connectivity.

2. IP auto-configuration
   After link layer connectivity has been established, the mobile node's IP configuration needs to be updated. Again, we assume a typical hot spot configuration, i.e., IP configuration will be performed automatically using DHCP. Personal observations in different hot spot installations have shown that the complete DHCP configuration step may take some 4 - 5 seconds.

3. Authentication
   After the initial IP configuration step, there is typically a further authentication, authorisation (and potentially: accounting) step that may be required by the local hot spot. The respective procedures are highly operator-specific. For example, many hot spots rely on a web redirection approach, where the first HTTP request in such a session is redirected to an operator's authentication page that allows the user to enter credentials in order to obtain network access (as is recommended practice by the Wi-Fi Alliance [ABS03]. The details of dealing with different authentication/authorisation methods are beyond the scope of of this paper and will be discussed elsewhere. But we can state the requirement, that the Drive-thru infrastructure on the mobile node must provide support to automatically authenticate users to the hot spot operator. The Drive-thru infrastructure must be able to detect the appropriate authentication method for the current hot spot – a process that should not require manual user interaction. In a sample setting, the hot spot infrastructure may employ periodic service announcements that describe the authentication method and other hot spot parameters. Apparently, the duration of this step depends on the deployed technical solution (and many other factors). Packet traces carried out by the authors in different commercial hot spot installations have shown that the manual authentication typically takes about 10 seconds.[9]

4. Application interaction
   The overall duration for the configuration and authentication steps may be some 15 seconds if we assume automated execution of the aforementioned steps. After this period the actual application transactions can commence respectively can be continued as soon as the production phase is entered. Our tests and measurements with different configurations have shown that Drive-thru connectivity phases are long enough to allow the first 15 seconds to be used for the initial preparation. However, we expect that dedicated Drive-thru hot spot architectures will help to reduce the overall duration significantly.

There are different ways how the Drive-thru architecture can support applications in the presence of intermittent connectivity. Fundamentally, the architecture provides a persistent connectivity service that can conceal interruptions and address changes. In addition, it is possible to employ application-specific support functions in the Drive-thru infrastructure, e.g., application layer gateways in the mobile Drive-thru infrastructure and in the Drive-thru proxy.

When we consider the example of accessing and sending e-mail messages, both forms of Drive-thru support are useful. For accessing e-mail (considering *offline* access with POP3 in this example), the *Drive-thru client* can provide a POP3 application layer gateway (ALG) that represents a POP3 server to the local application and forwards requests via the Drive-thru proxy to the actual POP3 server. The ALG in the Drive-thru client is aware of the current connectivity

[9] Not counting the time it takes the user to retrieve an often unmemorable account identifier and long authentication code and enter this information into a web form.

status and can thus suspend the processing of requests until connectivity has been established. In an advanced configuration, the Drive-thru client may perform active pre-fetching of e-mail messages on behalf of the user agent. In this scenario, the Drive-thru client may operate as an independent POP3 user agent and use its knowledge about the connectivity status to schedule the downloading of messages during the connectivity phases. The retrieved messages may be stored in a local messages store, and the user agent may contact the Drive-thru client and fetch the messages independent of their actual transmission. In addition, a corresponding POP3 instance in the Drive-thru proxy may also perform prefetching in order to further optimise the utilisation of the short connectivity periods.

Similar support functions can be defined for other applications as well. For example, to send e-mail messages the Drive-thru proxy could act as an SMTP relay and store messages locally until a connectivity phase begins.

7 Conclusions

This paper has discussed *seamless connectivity for mobile applications* from an unusual perspective: instead of proposing new mechanisms for achieving seamless connectivity we have made the point that seamless connectivity is hard to achieve but, fortunately, not required in many cases. The notion of *intermittent connectivity* applies to many usage scenarios of mobile and nomadic computing of which the Drive-thru environment is just a special variant with short-lived but powerful connectivity periods.

We have identified different classes of Internet applications and have analysed their communication characteristics and their requirements with respect to network connectivity. We conclude that many applications are workable in scenarios with intermittent connectivity even if they do not provide native support for such environments. Nevertheless, a suitable (network) infrastructure is needed to use connectivity clouds effectively. Our Drive-thru architecture is based on the idea of connection splitting and introduces a Drive-thru client on the mobile system and a corresponding Drive-thru proxy in the network communicating via a specifically designed transport protocol that supports persistent communications even in the presence of lower layer interruptions. As we have shown for different application classes, application-layer proxying/gatewaying functions may be added on top, executing on different components of the Drive-thru infrastructure, in order to provide an optimised service for unchanged end user applications.

Obviously the introduction of intermediary systems (the local Drive-thru client and the Drive-thru proxy) conflicts with end-to-end security considerations: Network-layer security mechanisms such as IPsec and transport-layer mechanisms such as TLS are not compatible with the connection splitting approach: While it is possible to employ these mechanisms for the PCMP connection between Drive-thru client and Drive-thru proxy, they cannot be applied end-to-end. Although this limitation may not be a problem for the majority of usage scenarios that are predominant in the Internet today, the issue should not be ignored. For Drive-thru scenarios, application layer security may prove to be a viable alternative for some applications. For example, e-mail messages and SIP messages can be secured end-to-end by employing S/MIME.

The Drive-thru architecture is not tied to specific operators, ISPs, or deployment models (it is not even tied to WLANs). Instead, it is targeted at independent deployment by individual operators, i.e., similar to conventional WLAN hot spots today. Therefore, similar administrative issues need to be addressed: auto-configuration, AAA and billing, roaming support, and quality of service. The Drive-thru approach requires additional mechanisms in order to efficiently use the short periods of connectivity; i.e., host configuration and user authorisation (and the Drive-thru connectivity management) must be performed quickly upon entering a Drive-thru cloud. However, this requires us to go beyond the commonly recommended hot-spot architectures with HTTP-based authentication pages and manual user intervention, a core area of our current research.

The Drive-thru environment we have presented in this paper is just one particular case in which users and applications need to deal with intermittent connectivity. We have deliberately chosen a scenario with extreme connectivity characteristics which therefore requires strong infrastructure support. The Drive-thru concepts are applicable to other means of transportation (buses, trains, etc.) and also to the rather simple nomadic computing scenarios of everyday life where user may similarly benefit from persistent communications in spite of connectivity interruptions and location changes.

References

[Abo04] Bernard Aboba. Detection of Network Attachment (DNA) in IPv4. Internet Draft draft-ietf-dhc-dna-ipv4-06.txt, Work in Progress, March 2004.

[ABS03] B. Anton, B. Bullock, and J. Short. Best Current Practices for Wireless Internet Service Provider (WISP) Roaming, Version 1.0. Wi-Fi Alliance, February 2003.

[All02] Open Mobile Alliance. Building an Industry-Wide Mobile Data Synchronization Protocol. Available from http://www.openmobilealliance.org/syncml/technology.html, 2002.

[ALMR01] Juha Ala-Laurila, Jouni Mikkonen, and Jyri Rinnemaa. Wireless Access Network Architecture for Mobile Operators. IEEE Communications Magazine, Vol 39, No 11, November 2001.

[BB94] Ajay Bakre and B.R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. Technical report, Department of Computer Science, Rutgers University, October 1994.

[BFW03] Marc Bechler, Walter J. Franz, and Lars Wolf. Mobile Internet Access in FleetNet. In *13. Fachtagung Kommunikation in verteilten Systemen, Leipzig, Germany*, April 2003.

[BKG+01] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance enhancing proxies intended to mitigate link-related degradations. RFC 3135, June 2001.

[BS97] Kevin Brown and Suresh Singh. M-TCP: TCP for Mobile Cellular Networks. *ACM Computer Communication Review*, 27(5), 1997.

[BSAK95] Hari Balakrishan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proceedings of the 1st ACM International Conference on Mobile Computing and Networking (Mobicom)*, November 1995.

[CBB04] Ranveer Chandra, Paramvir Bahl, and Pradeep Bahl. MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. Proceedings of IEEE Infocom 2004, March 2004.

[Dav02] Brian Douglas Davison. The Design and Evaluation of Web Prefetching and Caching Techniques. PhD Dissertation, Rutgers University, October 2002.

[Def03] DefaultCity. DefaultCity rolls out wireless broadband in Stockholm and Sweden. URL: http://www.defaultcity.net/pressrelease/pressrelease20030519_eng.html, May 2003.

[(ed02] Charles E. Perkins (ed). IP Mobility Support for IPv4. RFC 3344, 2002.

[Fal03] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. Proceedings of ACM SIGCOMM 2003, Computer Communications Review, Vol 33, No 4, August 2003.

[fle03] Homepage of FleetNet. http://www.fleetnet.de/, 2003.

[GMP00] Tom Goff, James Moronski, and D.S. Phatak. Freeze-tcp: A true end-to-end tcp enhancement mechanism for mobile environment. In *Proceedings of the IEEE Infocom Conference*, 200.

[Haa95] Zygmunt J. Haas. Mobile-TCP: An Asymmetric Transport Protocol Design for Mobile Systems. 3rd International Workshop on Mobile Multimedia Communications, September 1995.

[KS92] James J. Kistler and M. Satyanarayanan. Disconnected Operation in the Coda File System. ACM Transactions on Computer Systems, Vol 10, No 1, February 1992.

[Lei01] Gosta Leijonhufvud. Multi access networks and Always Best Connected, ABC. November 2001. MMC Workshop.

[LGKT02] R. Lee, K. Goshima, Y. Kambayashi, and H. Takakura. Caching Schema for Mobile Web information Retrieval. Proceedings of the 2nd International Workshop on Web Dynamics, Part of WWW 2002, May 2002.

[MB98] David A. Maltz and Pravin Bhagwat. MSOCKS: An Architecture for Transport Layer Mobility. Proceedings of IEEE Infocom 1998, 1998.

[OK04a] Jörg Ott and Dirk Kutscher. Drive-thru Internet: IEEE 802.11b for „Automobile“ Users. In *Proceedings of the IEEE Infocom 2004 Conference, Hong Kong*, March 2004.

[OK04b] Jörg Ott and Dirk Kutscher. The “Drive-thru” Architecture: WLAN-based Internet Access on the Road. In *Proceedings of the IEEE Semiannual Vehicular Technology Conference May 2004, Milan*, May 2004.

[PKH+00] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J.P. Makela, R. Pichna, and J. Vallstron. Handoff in hybrid mobile data networks. IEEE Personal Communications, Vol 7, No 2, April 2000.

[PM94] Venkata N. Padmanabhan and Jeffrey C. Mogul. Improving HTTP latency. Proceedings of the Second International World Wide Web Conference: Mosaic and the Web, October 1994.

[PM96] Venkata N. Padmanabhan and Jeffrey C. Mogul. Using Predictive Prefetching to Improve World-Wide Web Latency. Proceedings of the ACM SIGCOMM ’96 Conference, 1996.

[Shi03] Clay Shirky. Permanet, Nearlynet, and Wireless Data. http://shirky.com/writings/permanet.html, March 2003.

[ST03] Terry Schmidt and Anthony Townsend. Why Wi-Fi Wants To Be Free. Communications of the ACM, Vol 46, No 5, 2003.

[TMC04] TMCnet.com. Clic TGV Brings Wifi Onboard France’s High Speed Trains. available online at http://www.tmcnet.com/usubmit/2004/Jan/1022655.htm, January 2004.

[ZGGZ03] Qian Zhang, Chuanxiong Guo, Zihua Guo, and Wenwu Zhu. Efficient mobility management for vertical handoff between WWAN and WLAN. IEEE Communications Magazine, Vol 41, No 11, November 2003.