The "Drive-thru" Architecture: WLAN-based Internet Access on the Road

Jörg Ott
Dirk Kutscher

Technologiezentrum Informatik (TZI), Universität Bremen,
Postfach 330440, 28334 Bremen, Germany
Email: {jo|dku}@tzi.uni-bremen.de

***Abstract*— We present an architecture and discuss performance enhancement strategies for WLAN-based Internet access for moving vehicles. Measurements, on-going research activities, and the development of first prototypes have shown that WLAN-based Internet access for moving vehicles is feasible and can be an interesting and cost-effective alternative to GSM/3G-based approaches. We present an approach that is based on the concept of "nearlynets", explicitly considering the intermittent nature of connectivity that may be limited to short periods of time. Based on measurements and simulations, we have developed an architecture enabling useful Internet connectivity when driving through isolated hot spots. This architecture supports mobility in such networks relying on application-layer mobility mechanisms. We also discuss transport and application protocol performance enhancements as well as various operational issues such as detection of network access and address assignment.**

I. Introduction

Wireless access technologies are key to providing Internet and VPN connectivity to mobile users. Today, we can observe two approaches towards mobile network access: 1) Cellular (phone) networks strive for enabling ubiquitous – permanent – connectivity by building up an extensive infrastructure of base stations. Those *permanets* [1] are usually run by mobile phone operators; they are expensive to build and maintain, but they support continuous network access (at lower bit rates) and do not require users nor applications to adapt significantly to the mobile environment. 2) Wireless LANs, in contrast, provide only *hot spots* of network access thus yielding irregular – intermittent – connectivity. Those *nearlynets* [1] are often provided as part of a service arrangement (in hotels, airports, cafés etc.) and run by individuals, companies or by mobile network operators (e.g. ISPs). They are rather simple and inexpensive to operate, however with only local geographic coverage – even if many WLAN access points cooperate to extend a network's reach, e.g. on a university campus [2] or in a city center [3]. As a consequence, users have to adapt, e.g. by not moving out of a hot spot area. In addition, applications can be adapted to support offline operation.

One important class of mobile users are those traveling by car: they currently have to rely on cellular access technology, which is expensive, bandwidth constrained, and still not ubiquitous. Rather than attempting to provide truly ubiquitous network coverage using GSM or UMTS possibly in cooperation with other networks, our approach is to accept the intermittent nature of connectivity on the road. Therefore, we focus on providing powerful network access where it is available by means of WLAN hot spots along the road as a cost-effective alternative. In our *Drive-thru Internet* project, we have proven that WLAN technology is suitable to build connectivity islands for wide area networks with intermittent connectivity supporting mobile users in cars [4]. In such a network, the hot spots are located in irregular distances along the road, most likely provided by single or groups of access points at gas stations, restaurants, or in rest areas, with no (or limited cellular) connectivity in-between.

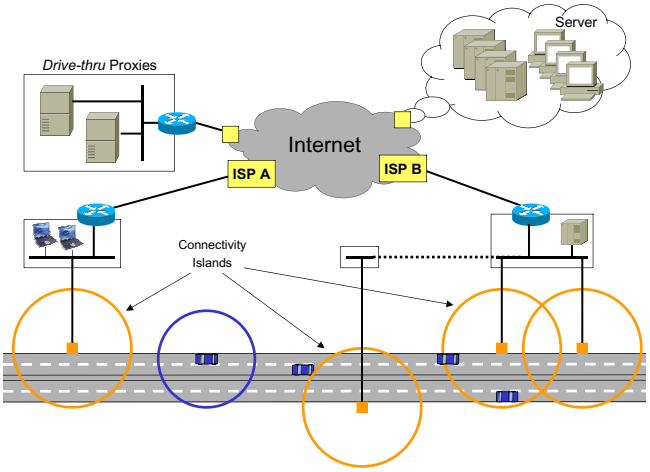

Fig. 1. Architecture Overview

The Drive-thru Internet project focuses on providing useful Internet services in environments with intermittent connectivity. While the approach we have taken in the Drive-thru Internet project is quite similar to the FleetNet project [5] in some architectural aspects, we explicitly consider instantaneous and incremental deployability as well as current practices for setting up hot spots [6] [7]. In particular, we do not assume mobile IP, and we also do not assume cars capable of packet forwarding. Figure 1 outlines our system architecture: basic Internet access is provided by connectivity clouds, each established by one or more 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. Similarly to FleetNet, we use an intermediary (a *Drive-thru proxy*) situated in a corporate or other home network or run by a third party to act as rendezvous point for all (Internet) communications. Mobility management takes place only at the application layer: the mobile node is responsible for re-establishing connectivity in each cloud with the Drive-Thru proxy. The latter acts as a fixed point for efficient communication setup: it fetches and buffers content from the Internet on behalf of the mobile node and forwards this data whenever the mobile node is reachable.

Apparently, this architecture deviates from the well-established end-to-end paradigm [8] that the Internet and most Internet protocols are based on. In this paper, we will discuss the communication characteristics further, motivate the necessity for our approach and describe the Drive-thru Internet architecture.

The rest of this paper is structured as follows: Section II presents specific measurement results and section III classifies our approach with respect to existing work. Section IV describes the Drive-thru architecture that we have designed based on these observations, and section V concludes this paper and describes future research and engineering activities for the Drive-thru Internet architecture.

II. Findings from Measurements and Simulations

We have performed two series of measurements on German highways and autobahns: a series of IEEE 802.11b tests and a series of IEEE 802.11g tests. The objective of theses tests was to validate the feasibility of Drive-thru Internet access at higher speeds, to analyze the link layer characteristics of IEEE 802.11 in these mobile scenarios, and to evaluate the performance of UDP and TCP under the observed conditions. Details of the first series of IEEE 802.11b measurements have been published in [4].

For all measurement series we have essentially employed a setup with a mobile system (the car with a computer providing an IEEE 802.11 interface) and a fixed system consisting of a fixed station that has been connected to an IEEE 802.11 access point via a Fast Ethernet switch. We have employed different measurement equipment for the two test series:

- For the IEEE 802.11b tests, we have used a Cisco 340 series IEEE 802.11b access point that has been connected to the fixed network infrastructure. For the mobile system, we have used a laptop computer with a Orinoco 802.11b "Gold" PCMCIA card equipped with an external omni-directional 5 dBi antenna (a Lucent "Range Extender") that has been mounted on the car roof.
- For the IEEE 802.11g tests, we have used a D-Link DWL-2000-AP IEEE 802.11g access point that has been equipped with an external omni-directional 8 dBi antenna (a Hawking Technologies H-AO8SI Hi-Gain outdoor antenna). For the mobile system, we have used a laptop computer with a Buffalo G54 IEEE 802.11g PCMCIA card equipped with an external omni-directional 5 dBi antenna (a Freebird IC6500 antenna) that has been mounted on the car roof.

We have measured both UDP and TCP performance in different scenarios. For all UDP measurements, we have used two tools, one for configurable packet transmission (including statistics reporting) and a receiving tool providing detailed logs for the incoming packets. We have transmitted packets of different sizes and at different intervals. In each measurement, we have used one active sender station transmitting packets to the other, using UDP/IPv4 unicast; our tests covered both directions, fixed (Ethernet-based) laptop to mobile and vice versa.

For all TCP measurements, we have used a client and a server, with the client residing on the mobile host, i.e., in the vehicle, and the server running on a fixed Ethernet-based host. Upon entering a connectivity cloud, the client connected to the server and initiated a data exchange according to a test specification. Both sides periodically reported the transmitted and received bytes per time frame until the connection was interrupted due to a loss of the connection from the mobile host to the access point.

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 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 have 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 [4]. 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.

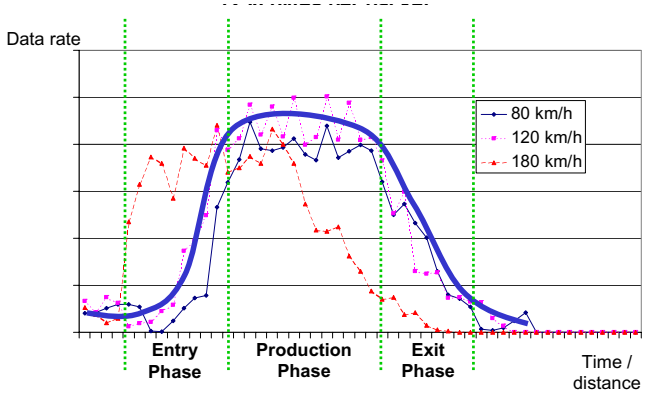

Fig. 2. Different Phases of WLAN Access (UDP)

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 have been able to obtain under laboratory conditions. In the entry and exit

phases, the throughput is decreasing due to a higher number of lost packets, link-layer retransmissions, and the wireless hardware's switching to lower 802.11b sending rates. Nevertheless, a limited form of communication has been possible. Our UDP measurements (that have been targeted at analyzing the maximum throughput for UDP sessions) have indicated that – even with significantly reduced sending rates – packet loss is likely to occur and the transmission delay increases due to link-layer retransmissions and queuing.

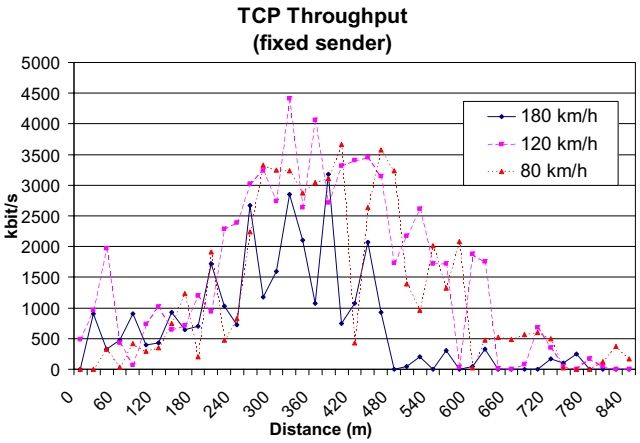

Fig. 3. TCP throughput from a mobile sender at different speeds

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 as depicted by figure 3 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 has shown a good overall performance and has been able to adapt to the varying transmission characteristics sufficiently well.

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 areas with a diameter of at least 2.5 kilometers. In addition, the increase in transmission rates and throughput has been more 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. The highest transmission rate of 54 Mbit/s has been obtained for several seconds at 80 km/h [1].

Consequently, we have observed a significant higher maximum throughput of some 15 Mbit/s and cumulative throughput of 110 MBytes. 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, our IEEE 802.11g measurements also confirm the three phase model described above, with even much larger ranges. When entering a connectivity island, the mobile device associates with the access point and completes DHCP-based autoconfiguration early during the entry phase, leaving sufficient room for subsequent authentication and connection establishment before the production phase is entered. Individual mobile nodes may take advantage of the full reach of a connectivity cloud and still obtain significant throughput even at large distances in the exit phase – but with multiple mobile nodes, we expect a more conversative behavior to be advised.

Summarizing, both the IEEE 802.11b and IEEE 802.11g tests have validated the general idea of IEEE 802.11 based Drive-thru networking at higher speeds. Even with IEEE 802.11b and less elaborated radio hardware we have been able to transmit a significant amount of data in a single Drive-thru cloud, deploying only one access point. The second measurement series has exposed the real potential of IEEE 802.11 technologies for our scenario. With inexpensive off-the-shelf equipment and IEEE 802.11g hardware, we have been able to increase the range and the throughput significantly.

Especially the IEEE 802.11g results indicate a need for quick adaptation to the varying transmission rates (and packet loss rates). TCP has shown to perform sufficiently well, however work on further optimizations is still on-going.

III. Related Work

Our measurements have validated the concept of WLAN-based communication at higher speeds. Interestingly, we can currently note an increasing deployment of WLAN technologies in the transportation environment. The deployment of WLAN hot spots has reached the road (and the rail). Truck stops, travel plazas and gas stations begin to provide (stationary) wireless access to customers. E.g., NATSO, the US-based National Association of Truck Stop Operators has founded *Truckstop.net* [9], a WLAN Internet service provider offering services for the transportation industry; and in several European countries, gas station chains have announced the introduction of WLAN hot spot services. Architectures for WLAN hot spots and roaming infrastructures [2] have been developed and commercial deployment of WLAN hot spots has been started by telecommunication operators. In addition, first vehicle-WLAN prototypes have been demonstrated such as the Citroën C3 Pluriel with WLAN capabilities [10].

While these developments are mainly targeted at providing WLAN hot spot access to stationary clients, some research projects have addressed mobile scenarios as well: IP communications on the road has also independently been studied by the FleetNet project [5], with a different focus and slightly different goals though: FleetNet primarily targets inter-vehicle communications in wireless ad-hoc networks for traffic-related control information using addressing geo-based routing. Similarly to other projects such as the Hocman project [11],

[1]Traffic conditions did not allow for higher speeds at this particular experiment. Tests at higher speeds are currently being conducted.

FleetNet also addresses data sharing across vehicles. FleetNet's Internet access is based on gateways for which service location features are provided [12]; the aforementioned routing provides continuous connectivity with optimal route selection where possible [13]. Other car communication environments assume hybrid networks, e.g. DVB-T peered with GSM or UMTS [14].

Providing enhanced mobility support for Internet protocol based systems is a research topic with many facets. While network layer approaches provide the fundamental mechanisms for mobile communications at the Internet layer, numerous research activities have addressed the issue of transport protocol performance in order to mitigate the effects of disruptive hand-overs and intermittent connectivity. `I-TCP` [15] is a *split-connection approach* that introduces a transport layer intermediary splitting a TCP connection between a fixed and a mobile host into two connections. The idea is to isolate the fixed host from communication anomalies such as packet loss due to hand overs and short periods of intermittent connectivity. `I-TCP` explicitly breaks the end-to-end semantics of the TCP connection, i.e., the TCP connections are terminated at the intermediary. In case of a hand-over, a state transfer from one `I-TCP` intermediary to another must be performed.

The `Snoop` protocol [16] provides a more transparent support and relies on a dedicated agent on the path between a mobile and a fixed station that *snoops* on the TCP communication, buffers TCP segments and transparently provides retransmissions. The TCP peers are thus shielded from segment loss that can be repaired by the `Snoop` agent. In case of a hand-over, a state transfer procedure is not necessarily required.

While `I-TCP` and `Snoop` represent optimizations for short-term communication problems caused by hand-over and transmission failures, the `Mobile TCP` approach [17] is additionally targeted at avoiding TCP connection termination due to longer connectivity blackout periods. A connection splitting mechanism, similar to the `I-TCP` approach is deployed, however, `M-TCP` generally maintains the end-to-end characteristics of the TCP connection as an intermediary does not buffer and retransmit segments but merely relays TCP acknowledgments. The mobile station and the `M-TCP` intermediary employ specialized TCP implementations that can accommodate connection interruptions that would normally lead to the termination of a corresponding TCP connection. When the connection to the mobile station is lost, an `M-TCP` intermediary sets the sending window size for the fixed station to zero, thus transitioning the TCP connection to *persistent mode*.

In addition to these optimization that are directly targeted at enhancing the performance of TCP in mobile wireless scenarios, other transport and application layer approaches have been developed. RFC 3135 [18] provides a survey of different types of performance enhancing proxies (PEPs).

Our approach differs from the aforementioned techniques because we do not address the problem of enhancing TCP performance for short-term communication interruptions in order to maintain a seamless, high-throughput TCP connection during hand-overs between different base stations. In the Drive-thru scenario, we assume intermittent connectivity with longer blackout phases to be the rule instead of the exception. A roaming user, e.g., in a car, may experience long periods without connectivity and even when connectivity is available, we do not assume support for Mobile-IP in visited networks (for IPv4). Different link layer technologies may be used, e.g., Ethernet, WLAN, 3G networks – with completely different characteristics with respect to performance, topology, and operational constraints.

IV. The Drive-thru Architecture

Based upon the findings from our measurements as well as on insights from related work, we have developed the Drive-thru architecture presented in this section. First, we outline our (user and application) requirements for Internet services in networks with intermittent connectivity. Next, we discuss the overall system architecture and introduce the individual constituents and their responsibilities. Finally, we present the protocol architecture and outline the characteristics of our Persistent Connection Management Protocol (PCMP) and briefly discuss the applicability of our approach.

A. Requirements

The main objective is to provide a solution that accommodates the distinct mobility and the intermittent nature of connectivity in Drive-thru environments. We have seen that the distribution of Drive-thru clouds (one or more access points covering a regional area) can be quite sparse, which results in long periods with no connectivity at all that are interrupted by rather short connectivity periods. During these periods, connectivity can vary from slow, unreliable IEEE 802.11b connectivity to almost "optimal" conditions that one would expect in non-mobile scenarios.

As discussed in section II, such connectivity islands are able to provide link layer connectivity for some 500m to 2500m or more, equivalent to periods of some five seconds to almost two minutes at various highway and autobahn speeds. For some significant fraction (some 25-40% of the period), mobile devices can obtain goodput rates close to laboratory settings and exchange data volumes of more than 100 MBytes in total while passing a single access point.[2]

These findings indicate that, for each individual connectivity island, autoconfiguration is feasible for a mobile device (which we have validated by further experiments for DHCP) and there should be sufficient time for automated authentication with a wireless service provider. Therefore, we may assume that a communication path can be established in each connectivity island that the mobile user is authorized to access so that we can concentrate on the architectural requirements from the user's and applications' perspectives.

Moreover, we cannot assume all Drive-thru clouds to be operated by the same service provider. Consequently, we have to accommodate different topological points of attachments,

[2]Those values hold when a mobile device uses the respective access point by itself. Measurements for simultanous access from multiple cars are presently in progress.

different IP and security configurations and different AAA infrastructures and requirements for mobile nodes. For example, WLAN hot spots today, provide different user authentication mechanisms, e.g., Web-based authentication by relying on a transparent HTTP-redirector as opposed to being EAP [19] based.

These restriction make the deployment of mobility support on the network layer, i.e., Mobile IP, difficult. But even if Mobile IP could be deployed in a subset of Drive-thru clouds, the strong intermittent nature of Drive-thru connectivity requires additional support for transport protocols and applications. Most current transport and application layer protocols are not designed to work well in such an environment, e.g., TCP connections will typically not survive the transition from one Drive-thru cloud to another with a longer period of interrupted connectivity in between.

While Internet applications can be classified in a number of ways, their information exchange pattern (bursty as opposed to continuous communication) is the most relevant for the Drive-thru architecture. For example, on one hand, interactive real-time communication applications (such as telephony and synchronous multimedia conferencing) are infeasible for Drive-thru networks. On the other hand, more transaction-oriented applications (such as e-mail, file transfer, and database synchronization) – that usually support some mode of offline operation – are almost workable by themselves under these conditions. The latter, however, would still benefit from automated triggers whenever connectivity becomes available. Yet other applications (such as the most desirable web browsing or even personal presence) may become workable to some degree in a Drive-thru environment given proper system and network infrastructure support. Finally, applications that are aware of and actively deal with intermittent connectivity may be designed, of course.

With these considerations in mind, we can identify the following requirements on the Drive-thru architecture:

- the architecture must provide persistent connectivity that is useful for a range of existing applications such as Web and email access, file transfer, etc.;
- the architecture should enable new applications that are aware of the intermittent nature of connectivity
- the architecture should not be dependent on a specific WLAN hot spot architecture and it must be applicable to different WLAN authentication technologies;
- no changes to existing operating systems and applications should be required; the architecture must not require the usage of specialized mobile devices but must support existing user equipment (laptops, inbuilt computers in cars etc.);
- the architecture must accommodate the specific characteristics of Drive-thru WLAN clouds as described in section II but must also accommodate other access network technologies such as Ethernet, dial-up connections and 3G networks;
- it must be possible to include performance enhancing proxy elements that operate on the network and transport layer, e.g. those described in section III; and
- the Drive-thru architecture must allow for operator-independent deployment, i.e. everyone with an Internet connection such as a DSL link should be able to provide Drive-thru access services.

B. System 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 on the network layer but takes interruptions, mobility, IP stack reconfiguration, etc. into account and introduces a connectivity management service *above* the transport layer. The 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* at the mobile node and *Drive-thru proxies* in the network. They operate peerwise 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.

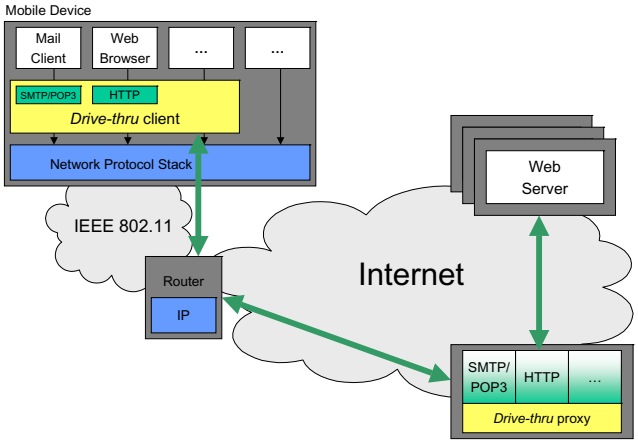

Fig. 4. Drive-thru architecture overview

Figure 4 shows a schematic overview of the Drive-thru architecture comprising the following elements:

- The *Drive-thru client* acts 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. The Drive-thru client is responsible for re-establishing underlying TCP connections (see next subsection) after a connectivity interruption. To facilitate this, it comprises a logical entity that provides indications from the link layer about the association with and signal strength of an access point.
  At the application layer, the Drive-thru client is capable of accumulating requests from the clients for transmission

in the next connectivity cloud as well as for caching information received from the Drive-thru proxy (via some push or prefetching mechanism) for later use by the applications.
The Drive-thru client uses standard protocols to communicate with its co-located applications entities and thus can either be run on a user's device (e.g. a laptop) or can be realized as part of a vehicle's communication platform.

- 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 proxies receives (batches of) requests from the Drive-thru client and executes those 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 those 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), proactively carry out those actions as far as possible (e.g. pre-fetching web pages), and then push the contents to the mobile node.
  Drive-thru proxies may be operated by ISPs, independent application service providers, or by individual users themselves. They may be located anywhere, provided that they are permanently reachable, globally addressable, and (for performance reasons) sufficiently well connected to the Internet.
- The mobile application peers (e.g. web browser or mail client) are the user's unmodified standard applications. They just need to support proxies or application layer gateways and need to be configurable accordingly.
  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.
- For the fixed application peers not even a re-configuration is needed; they remain entirely unchanged. They communicate with the Drive-thru server 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 email) as much as possible.

A final component has not been depicted in figure 4: the *Drive-thru PEP*. While our experiments have shown that TCP is well-suited for high performance data exchange with moving vehicles, those results apply to TCP connections terminated in the connectivity cloud. In real-world hot spot settings, however, we face an access link from the connectivity island to an ISP which is likely to become the bottleneck in terms of throughput limitations, congestion losses, and particularly latency. As TCP adjusts its transmission behavior at a temporal resolution in the order of RTT, long (and varying) RTTs across the access link and the backbone are expected to decrease TCP's reactivity to the quickly changing link layer characteristics observed by a moving vehicle.

Therefore, we envision another (optional) Drive-thru entity to be added: the *Drive-thru PEP*. This *performance enhancing proxy* may split the TCP connection between the Drive-thru client on the mobile node and the Drive-thru proxy in the fixed network. It does not perform any application layer functions but just buffers and forwards data.[3] If present, a Drive-thru PEP may announce its availability and its capabilities to Drive-thru clients so that those can actively leverage its functionality.

C. Protocol Architecture

In the Drive-thru architecture, the Drive-thru client and the Drive-thru proxy maintain a persistent relationship for managing *connections* initiated by the client, i.e., the mobile node. For this purpose, we have defined a *Persistent Connection Management Protocol (PCMP)* that allows a Drive-thru client to register with a Drive-thru proxy, to establish transport sessions[4], and to resume and terminate transport sessions. The Drive-thru proxy operates as an intermediary and sets up connections to communication peers in the public Internet corresponding to the connection setup request that are issued by the mobile node. Should the mobile node leave its current point of attachment, i.e., the current Drive-thru cloud, and is no longer reachable for a significant amount of time, the Drive-thru proxy maintains the established connections (as far as possible). When the mobile node enters the next connectivity cloud, the Drive-thru client may resume the connections and continue to communicate via the Drive-thru proxy. On top of PCMP, application-specific protocols may be extended for communication between Drive-thru client and proxy where necessary to improve mobility support in Drive-thru environments.

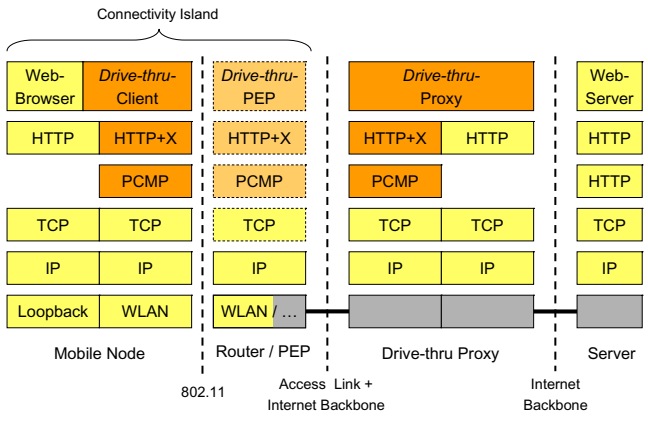

Fig. 5. Drive-thru protocol stacks

Figure 5 depicts the involved entities in this scenario from a protocol layering perspective for the use of HTTP as an application layer protocol. The mobile node provides unmodified applications such as a web browser that are configured

[3]Further research will investigate the performance gains achievable through Drive-thru PEPs in detail.

[4]Currently, we focus on the management of TCP sessions.

to employ the local client proxy. The Drive-thru client proxy supports PCMP and maps HTTP transactions to the corresponding PCMP connection setup requests and data messages that are sent to the Drive-thru proxy. The Drive-thru proxy again maps these requests to regular HTTP/TCP requests for communicating with a public, Drive-thru unaware web server.[5]

For PCMP, the communicating peers, i.e. the client proxy and the Drive-thru proxy, are identified by an IP address independent identifier allowing PCMP sessions to persist in spite of changing points of attachments and IP addresses. PCMP registration requires authentication as the applications may incur significant resources being tied up at a Drive-thru proxy.

Within a PCMP connection between the Drive-thru client and proxy, application session multiplexing (multiple sessions in a PCMP context) is supported. For each of these connections, persistent identifiers are used that allow their selective re-establishment in a subsequent connectivity cloud. PCMP transport sessions are synchronized upon each session resumption to ensure that no data gets lost in flight. PCMP transport sessions are individually flow-controlled at the PCMP layer while reliability and congestion control are provided by the underlying TCP.

It should be noted that the Drive-thru proxies themselves do not provide any TCP performance enhancing functions at the transport layer. Such functions may be added by a Drive-thru PEP, co-located with the access router (but usually not running on the same machine) within the connectivity island as discussed above. Beyond simple TCP connection splitting or similar functionality (as indicated by the dashed TCP protocol block in figure 5), the Drive-thru PEP may be involved in PCMP connection forwarding and buffering as well as selected application layer extensions (indicated by the dashed boxes for HTTP+X and PCMP).

On top of PCMP, application-specific protocols may need to be extended to better support the Drive-thru mobility management. In figure 5, the protocol blocks labeled HTTP+X indicate that HTTP-specific extensions are being used. Such extensions may e.g. support pipelining of requests and batching responses. Furthermore, HTTP extensions are needed to support pre-fetching and push-based distribution of web pages. Depending on the availability and complexity of a Drive-thru PEP in a particular connectivity island, these extensions may be used between Drive-thru client and proxy or between either and the Drive-thru PEP. If a Drive-thru PEP is available, it may be used to pre-provision information requested from the client to a connectivity island prior to the mobile node entering this cloud, assuming that predictions about the mobile node's arrival are possible.

D. Applicability

In summary, the Drive-thru architecture addresses the TCP, PCMP, and application protocol layers between the Drive-thru client and the Drive-thru proxy. (Re-)establishment of PCMP (and the underlying TCP) connections is up to the mobile node while multiplexed transport sessions within a PCMP connection can be initiated or resumed by either side. Drive-thru PEPs may be inserted in connectivity islands close to the wireless link and may operate at all three of the above protocol levels, possibly complemented by Drive-thru-independent performance enhancement mechanisms such as `Snoop`.

This approach allows Drive-thru services to deployed in arbitrary hot spots as the latter do not have to perform any Drive-thru specific functions, they do not even need to know about this use. They are transparent providers of IP connectivity while all the basic functions are carried out by the mobile user and her Drive-thru proxy (which may be run in her fixed network). The way the mobile node is configured[6] and how PCMP connections are established should work with regular hot spot architectures, even in the presence of Network Address Translators (NATs).

However, hot spots providers may decide to improve the Drive-thru performance by additionally deploying Drive-thru PEPs of varying functional range. And they may provide additional services to Drive-thru users including specific service announcements, coordination with neighboring connectivity clouds, and content caching and push. Furthermore, they may deploy applications specifically designed for the use in such intermittently connected environments.

V. Conclusions

The Drive-thru architecture that we have introduced in this paper is an approach to enable communication for existing and future applications in the presence of mobility and intermittent connectivity. The measurement results presented in section II and in [4] have validated the fundamental principle of applying IEEE 802.11 WLAN technologies to mobile communication to fast moving vehicles as mobile nodes. Especially our recent IEEE 802.11g measurements have shown that the combination of higher transmission rates and appropriate radio hardware can lead to a TCP goodput of more than 100 MBytes per Drive-thru cloud, using a single access point only.

In addition, the measurements have provided insights into the link characteristics of Drive-thru communications and on the performance of UDP and TCP. In particular, we have derived the three-phase model from the extreme variability in throughput and packet loss rates as a basis for further design considerations. Together with the distinct intermittent nature of connectivity, the Drive-thru Internet setting differs significantly from conventional environments employed for IP based communication. Consequently, many common assumptions that Internet protocols are based on simply do not hold: there is no seamless connectivity for mobile nodes, link layer characteristics are not predictable, and router congestion is not always the dominant cause for packet loss.

[5]The current version of our PCMP implementations are layered on top of TCP, i.e., the client proxy sets up a TCP connection to the Drive-thru proxy and routes all PCMP request over that connection. However, the use of other protocols such as SCTP and DCCP is feasible as well.

[6]The authors are well aware of the multiplicity of authentication schemes in use in today's hot-spot infrastructures. Dealing with this in an efficient manner, in particular to support automated authentication if so desired by the mobile user, is subject to ongoing research.

Drive-thru Internet is, similar to delay-tolerant networks [20], an environment where the naïve application of the end-to-end argument [8] as a design principle seems to be inadequate. The notion of a dumb network where all error recovery is performed by the end systems does not fit into an environment where one end of the path, the Drive-thru access network, shows these extremely varying characteristics. Dedicated support is needed to accommodate the Drive-thru network characteristics and to deal with intermittent connectivity.

We have introduced the Drive-thru intermediary model as a means to manage mobility and intermittent connectivity by placing intermediary systems around the problematic access network. This approach allows for isolating the Drive-thru specific support to a subsystem on the mobile node and to a dedicated Drive-thru proxy in the Internet, thus enabling *some* client and server applications to be used without modification. Typically, this enables us to use transaction-based application such as e-mail sending/retrieving and web browsing. Clearly, the Drive-thru architecture cannot provide support for interactive, real-time applications such as telephony – at least not across multiple Drive-thru clouds.

The Drive-thru Internet project is an ongoing research activity and there are still some interesting questions to resolve: Our current measurements have essentially considered Drive-thru clouds that are served by exactly one IEEE 802.11 access point. First tests have shown that roaming between access points is difficult to achieve given the short connectivity periods. With high-gain omnidirectional antennas as we have used for our IEEE 802.11g tests, the distance between access points needs to be several hundreds meters in order to enforce roaming based on signal strength differences. We are currently investigating the effect of different access point configurations with respect to channel usage, beacon intervals and transmit power – not only to increase the coverage of a Drive-thru could but also to enable load-balancing, etc.

Another topic that we are currently investigating is the effect that authentication procedures (but also other operational aspects) impose on the available duration of connectivity and on the overall performance per Drive-thru cloud utilization. Clearly, we will have to deal with heterogeneous architectures and authentication procedures that are originally intended for rather stationary usage scenarios. A related activity is the investigation of mechanisms for detecting network access quickly in order to notify Drive-thru clients and applications when entering a Drive-thru cloud. Recent experiences with mobility have led to the development of a set of optimizations and heuristics that can be employed for this purpose [21].

After establishing and maximizing network access and persistent connections within and across connectivity islands, we can turn our attention towards evaluating the transport and application layer performance of our architecture. We will analyze the behavior of PCMP and the underlying TCP with different blackout times and particularly consider the performance impact of communications across access and backbone links. In this context, we will investigate the impact of Drive-thru PEPs operating at the transport layer and below as discussed in section III.

Ultimately, however, it is the application performance that matters to a user – surely in terms of throughput but even more so in terms of perceived (as opposed to real connectivity). As ubiquitous and performant connectivity is still far off, an important focus of our future work will be at the application layer in Drive-thru clients, proxies, and PEPs – enabling Drive-thru Internet services with existing applications to offer the familiar service model of permanent connectivity as far as possible.

REFERENCES

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

[2] Carsten Bormann and Niels Pollem, "Wbone: WLAN Roaming Based on Deep Security," TERENA Networking Conference (TNC), May 2003.

[3] Kaj Thomson, "Large Scale Deployment of Public Wireless LANs," M.S. thesis, Royal Institute of Technology (KTH), Stockholm, 2003.

[4] 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.

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

[6] Simon Byers and Dave Kormann, "802.11b Access Point Mapping," Communications of the ACM, Vol 46, No 5, 2003.

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

[8] Jerome H. Saltzer, David P. Reed, and David D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, nov 1984.

[9] "Homepage of Truckstop.net," http://www.truckstop.net/, 2003.

[10] Jason Ankeny, "Philippe Gaches, Director of Telematics and Multimedia Perspectives, Citroën," available online at http://wirelessreview.com/ar/wireless_philippe_gaches_director/index.htm, August 2003.

[11] Mattias Esbjörnsson, Oskar Juhlin, and Mattias Östergren, "The Hocman Prototype - Fast Motor Bikers and Ad-hoc Networking," Proceedings of MUM, 2002.

[12] Marc Bechler, Lars Wolf, Oliver Storz, and Walter J. Franz, "Efficient Discovery of Internet Gateways in Future Vehicular Communication Systems," in *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC 2003 Spring), Jeju, Korea*, April 2003.

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

[14] P. Christ, P. Krummenacher, P. Robertson, and E. Stare, "The Multimedia Car Platform," *IBC'2001, Amsterdam*, September 2001.

[15] Ajay Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," Tech. Rep., Department of Computer Science, Rutgers University, October 1994.

[16] 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.

[17] Kevin Brown and Suresh Singh, "M-TCP: TCP for mobile cellular networks," *ACM Computer Communication Review*, vol. 27, no. 5, 1997.

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

[19] L. Blunk and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)," RFC 2284, March 1998.

[20] Robert E. Filman, "End-to-end over interplanetary networks," *IEEE Internet Computing*, vol. 7, no. 5, pp. 4–5, September 2003.

[21] Bernard Aboba, "Detection of Network Attachment (DNA) in IPv4," Internet Draft draft-ietf-dhc-dna-ipv4-01.txt, Work in Progress, September 2003.