

Multipath TCP in Vehicular to Infrastructure Communications

Nigel Williams, Prashan Abeysekera*, Nathan Dyer*, Hai Vu, Grenville Armitage
Centre for Advanced Internet Architectures, Technical Report 140828A
Swinburne University of Technology
Melbourne, Australia

njwilliams@swin.edu.au, 6653634@student.swin.edu.au, ndyer@swin.edu.au, hvu@swin.edu.au, garmitage@swin.edu.au

Abstract—This technical report describes some preliminary experiments investigating the use of Multipath TCP (MPTCP) to augment cellular 3G connections with roadside infrastructure based on emulated WAVE/DSRC transceivers. MPTCP is an extension to TCP that allows a multi-homed host to utilise multiple interfaces on a single TCP socket. The IEEE WAVE/DSRC stack defines standards for wireless vehicular communications. Experiments are performed on a physical testbed using the Linux MPTCP implementation. We first characterise the performance of MPTCP using a mix of wireless links. We then simulate MPTCP mobility using WiFi and 3G. Finally we conduct a small-scale, vehicle-based field test using 3G and a DSRC transceiver adapted from 802.11a hardware. We find that in cases of high path-diversity MPTCP can perform worse than standard TCP. When used in mobile scenarios (both simulated and in field trials), MPTCP is found to provide some benefit to existing 3G connections by detecting and then utilising roadside access points.

Index Terms—MPTCP, V2I, WAVE, DSRC, 802.11p, experiments, testbed

I. INTRODUCTION

Intelligent Transport Systems (ITS) are a broad set of traffic safety and management services accessed through vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communications. It is expected that in the coming years ITS will expand beyond current applications such as electronic tolling to include services such as collision warning systems and traffic update dissemination.

Supporting these vehicular-based ITS applications are the IEEE Dedicated Short-Range Communication (DSRC) and Wireless Access for Vehicular Environment (WAVE) suite of standards. The physical layer of DSRC

is defined by 802.11p, an amendment to 802.11 (Wi-Fi). Together these standards form a common network stack and communication channels for ITS access devices.

While most DSRC messaging is expected to be short-lived and delay-sensitive [1], there is also provision for delay-tolerant services running over TCP/IP. It is anticipated that a portion of ITS infrastructure deployment will be driven by the provision of such so-called *infotainment* services [2].

A potential use of roadside network infrastructure is to provide Internet access to passing vehicles [3], [4]. A limiting factor however has been the lack of session persistence between access points - a TCP connection must be broken once the access point is out of range, even if an alternate interface is available (E.g. cellular).

The development of Multipath TCP (MPTCP) [5] means that it is now possible to spread a single TCP connection over multiple interfaces, and perform handover as new interfaces appear [6]. Thus a moving vehicle could utilise both cellular and roadside infrastructure for a single TCP connection, scheduling data for the most appropriate path.

There are several reasons why using MPTCP for such connections would be beneficial. Cellular data connections feature high availability but are comparatively expensive, whereas roadside access points have limited coverage but may be cheaper. In cases where path cost is consideration, using a lower cost link might be preferred.

The throughput and RTT of a cellular connection may also vary while in motion due to the broader coverage area and changing in terrain. Roadside infrastructure will typically be tuned to the installation environment in order to provide consistent link quality over the coverage area.

In this report we study MPTCP in static and mobile scenarios using 3G and two Wi-Fi access modes:

*The work described in this report was done during the author's summer internship at CAIA in 2014

802.11n, and an emulation of DSRC/802.11p. We first test the performance of MPTCP using different combinations of these access modes when downloading files of different sizes. We then perform simulated mobility experiments using 3G and both Wi-Fi variants for several V2I scenarios. Lastly we conduct a field trial using a vehicle to drive past a DSRC/802.11p roadside access point while simultaneously using 3G. We use the Linux MPTCP v0.88 kernel [7] for our experiments.

We find the following:

- When using the default congestion control [8] over links of comparable performance, MPTCP is able to perform as good as, or better, than single-path TCP over a range of download sizes.
- When the available paths are asymmetric (in terms of RTT, bandwidth, loss rate), multi-subflow connections can perform worse than single-path TCP on the best link, for short flows (sub 1MB). In these instances, the rate at which a connection can be completed is bounded by the RTT of the slowest path. The impact of the slower path increases with smaller file sizes and greater path asymmetry.
- Use of IW10 (RFC6928) [9] can further extend download times, as a slower path must send a minimum of ten segments. IW10 will also contribute to application-level delay due to out-of-sequence delivery of data-level segments.
- The choice of default interface has a bearing on the time-to-completion for short flows. Opening a connection on a slower subflow with longer RTT will delay the addition of the faster link, as the address is not advertised until after the initial window has been received on the first subflow (up to ten segments if IW10 is used).
- Short-duration (less than 20s) subflow connections to roadside infrastructure are able to make a meaningful contribution, in terms of downloaded bytes, for long-lived delay-tolerant downloads. The short-term increase in throughput is able to compensate for the delay and data-level retransmissions required in the period after the RSU has moved out of range.

The remainder of the report is organised as follows: Section II provides background on MPTCP and ITS technologies. We discuss related work in Section III. The testbed and experimental methodology are described in Section IV and Section V. Section VI presents the baseline single-path TCP characteristics of each of the test paths. Results for MPTCP with stationary hosts are in Section VII, results for mobile scenarios are presented

in Sections VIII and IX, with brief discussion in Section IX-A. Section X concludes and Section XI outlines future work.

II. BACKGROUND

This section provides a brief overview of MPTCP and DSRC.

A. Multipath TCP

MPTCP [5] is an extension of TCP and allows a host to spread a single TCP connection across multiple network addresses. MPTCP is implemented within the kernel and is designed to be backwards compatible with existing TCP socket APIs, thus operates transparently from the perspective of the application layer. In use, MPTCP mimics the on-wire behaviour of standard TCP and uses TCP options for signaling. A new MPTCP option and subtypes have been defined to support this. The following sections briefly describe MPTCP session management and congestion control.

1) *MPTCP Session* : MPTCP connections consist of one or more *subflows* (with each subflow representing a unique 4-tuple of source/destination IP address and port) and are established using a standard TCP 3-way handshake over the default route. An *MP_CAPABLE* option is included in the initial SYN, which indicates compatibility with MPTCP. An MPTCP-capable server will respond by including an *MP_CAPABLE* option in the SYN/ACK.

Once the first subflow is established, either host may advertise additional interfaces using the *ADD_ADDR* option, transmitted over the already established subflow. On receiving an *ADD_ADDR*, a host can attempt to create a new subflow by sending a SYN to the advertised address and including the *MP_JOIN* option.

A *REMOVE_ADDR* option is also defined, and is transmitted by a host when an interface is no longer available for a connection. Un-acknowledged data on a removed subflow is re-injected into the remaining subflows.

2) *Coupled Congestion Control*: A basic multi-path connection may consist of several subflows behaving as independent TCP flows (such as might occur with parallel TCP implementations). This maximises use of each path but can result in undesirable situations, such as bottleneck links where a single MPTCP connection will consume more bandwidth than single-path TCP cross traffic. It is therefore beneficial to couple the congestion window behaviour of the subflows in order to maximise

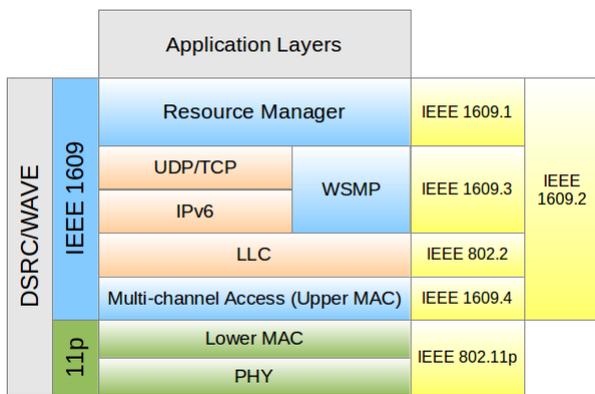


Figure 1. Simplified DSRC/WAVE stack. The IEEE 1609 standard has been designed specifically for DSRC, and integrates other existing IEEE and IETF standards. Note that the MAC layer consists of two sub-layers - the multi-channel access scheme (IEEE 1609.4) and 802.11p MAC (which is based on 802.11e EDCA).

use of the available capacity while ensuring fairness to other TCP flows.

Coupled Congestion Control for Multipath Protocols [10] is the default congestion controller for the Linux MPTCP implementation. The three primary design goals of the algorithm are:

- To provide aggregate throughput at least as good as single-path TCP on the best path.
- To not use more capacity on a shared resource than if using single-path TCP over the same path.
- Move traffic away from congested paths.

The behaviour of the algorithm is broadly described by the following:

- Each subflow maintains an independent congestion window.
- Slow start behaviour is the same as for TCP NewReno (congestion window increases by one segment for each ACK).
- During congestion avoidance, the additive increase mechanism (ACK) is coupled between the subflows.
- Multiplicative decrease (on loss) is handled per-subflow and behaves like TCP NewReno.

A full description of the algorithm and the design process are in [8]. The default congestion control algorithm has been found to be sub-optimal in certain areas [11] and several algorithms have been proposed (E.g. [12], [13]). We do not consider alternate congestion control implementations in this report.

B. DSRC/WAVE

The DSRC/WAVE (IEEE 1609.0-2013) stack (Figure 1) provides a common set of standards through which

V2V and V2I communications can occur. The two main devices within the WAVE framework are the On-Board Unit (OBU), installed in vehicles, and the Roadside Unit (RSU), which represents fixed roadside infrastructure.

1) *DSRC*: The DSRC standard specifies communications channels for vehicular ITS applications within a line-of-sight transmission range of 1000m. It has been designed with short-lived but delay sensitive communications in mind. In the United States it is based on the licensed 5.9Ghz band, and contains seven 10Mhz channels for V2V and V2I communications.

The specification mandates a single Control Channel (CCH) be used for safety and control messages. Up to six additional Service Channels (SCH) are available, and can be dedicated to a particular type of communication, such as V2V-only or infotainment services.

Typically a station will monitor the CCH for safety messages and to learn what services are available on the SCHs. The station will then periodically alternate between the CCH and a particular SCH. The multi-channel coordination sub-layer (IEEE 1609.4) provides a scheme by which DSRC devices can perform this channel switching. As an alternative to channel hopping, it is also possible to monitor CCH and SCH channels simultaneously using two transceivers [14], [15].

The US, European and Japanese DSRC standards differ in frequency, bandwidth and the number of channels available, hence regional standards are not compatible.

2) *WAVE IEEE 1609/802.11p*: The higher layers of WAVE cover multi-channel coordination (IEEE 1609.4), network protocols (IEEE 1609.3) and security (IEEE 1609.2).

Network services can be provided via TCP/IP (for non-safety applications) or through the IEEE WAVE Short Message Protocol (WSMP, from IEEE 1609.3). The WSMP has been developed for DSRC services and removes much of the overhead of IP. WSMP messages are used primarily for safety applications and for DSRC service advertisement (via WAVE Service Advertisements, WSA). 802.11p does not include authentication or security features - these are specified by IEEE 1609.2 and are performed in the upper layers.

The PHY and lower-MAC layers of WAVE are defined by the 802.11p standard. It is an amendment to IEEE 802.11-2007 and adapts existing 802.11 standards for use in DSRC environments.

The physical layer of 802.11p is based on 802.11a OFDM. It runs at half the clock rate of 802.11a, thus the channel bandwidth is 10Mhz (from 20Mhz) and supported data-rates range from 3-27Mbps (rather than

6-54Mbps). The half-rate operation doubles the time-domain characteristics of the signal and reduces errors caused by mobility (principally Doppler spread) and the environment. The specification also allows for much greater transmission power output compared with 802.11a (2-30W instead of less than 200mW).

The lower-MAC layer of 802.11p is based on 802.11e Enhanced Distributed Channel Access (EDCA). As in 802.11e EDCA, four access categories (ACs) are defined: Background (BK), Best Effort (BE), Voice (VO), Video (VI). Each access class is assigned a queue, and queues are serviced according to the priority of the AC. Contention parameters (Arbitration Inter-Frame Spacing, Contention Window Min/Max) are also set for each queue, with higher-priority ACs (VO, VI) having smaller values.

III. RELATED WORK

Previous studies have examined the use of Wi-Fi technologies in mobile environments using traditional TCP, while others have studied MPTCP in static (fixed) or limited mobility (e.g. walking speed) environments.

Chen et al. conduct a measurement-based survey of MPTCP across the Internet [16]. They use a multi-homed client with residential ADSL and 3G/LTE interfaces to download files from a single-homed server. File sizes are in the range of 64KB to 16MB. Measurements include download time and out-of-order delay.

They find that in most cases MPTCP using the default congestion control algorithm is able to download files at least as quickly as standard TCP. Longer flows benefit more than shorter flows, as there is more time over which each subflow can contribute to the connection.

For shorter flows, the duration of the download is determined by the largest RTT of the contributing links. Short flows are also shown to not make optimal use of capacity, as a single subflow must be established before additional subflows can be advertised and added to the connection (the processes of advertising and joining a subflow takes several RTTs, during which data could have been transferred). However once connected they did benefit from having multiple subflows in the slow start phase (all subflows experiencing exponential window growth). An enhancement is suggested where multiple subflows are initiated simultaneously at the beginning of the connection. This optimisation is applicable only in cases where all paths are available at the start of a connection.

They find out-of-order delay to be an issue when one of the subflows has a higher RTT or packet loss rate

than the other (e.g. using subflows of 50ms and 200ms together).

Raiciu et al. investigate the effect of mobility on the throughput of MPTCP connections [17]. A laptop equipped with Wi-Fi and 3G interfaces was moved on foot through indoor and outdoor environments while performing a long-lived download. They find MPTCP to be resilient to episodes of weak signal strength while moving between floors of a multi-story building. The laptop was also mounted in a backpack and moved between coverage areas of a Wi-Fi network. A connection established over 3G was able to utilise and benefit from the additional Wi-Fi network capacity when available.

Further research by Paasch et al. studied the handover phase of an MPTCP connection and its impact on TCP goodput and application delay [6]. Handover occurs when one interface is no longer available and is removed from the MPTCP connection. During this period an alternate subflow must take over the connection and re-transmit any outstanding data from the removed interface. They test handover in *Full-MPTCP* and *Backup-MPTCP* modes, and find that the impact on throughput and application-delay to be minimal.

Several studies ([3], [4]) have proposed using ISM-band Wi-Fi for V2I communications. These experiments used standard TCP, thus connections were not persistent and did not extend beyond the range of the roadside access point.

Bychkovsky et al. used public access points in the greater Boston area as RSUs for nine 802.11b-equipped vehicles [3]. The topography featured a mix of urban, suburban and highway environments. Measurements included TCP connection throughput (upload), access point coverage area and average connection time, and were conducted over the course of 12 months. They find that the access points provide a useful period of connectivity, with a mean connection time of 13 seconds during which a mean of 216KBytes of data could be uploaded.

Gass et al. conduct a controlled-environment test in which a single 802.11b access point was set-up in a desert location free of traffic and sources of radio interference [4]. A single vehicle was driven past at constant speeds ranging from 8-120km/h. They perform TCP and UDP bulk downloads, as well as fetching data via multiple http requests. Files were retrieved from a locally hosted server and from a remote server to simulate "real-world" network conditions.

They find that the 802.11b access point provided usable capacity for downloads, although the introduction of longer end-to-end delay and application-layer proto-

cols with request-response behaviour (http) caused poor utilisation of this capacity. They observe that the MAC-layer authentication stages of 802.11b reduces the usable window during which data can be transferred.

The performance of 802.11p for V2V and V2I applications has been widely studied ([18], [19], [20], [15], [21], [22], [23]). However focus has primarily been on safety messaging and physical layer signal performance, and due to the difficulty of performing field trials, many of the studies are based on network simulations.

Gozalvez et al. perform an extensive survey of 802.11p V2I communications in 21 locations across the city of Bologna [18]. The study investigates the effect of urban topographic features such as bridges, trees, roundabouts and heavy vehicles on the connectivity range of RSUs. They transmit a periodic (100ms) packet containing time and location meta-data from the OBU and observe the *Packet Delivery Rate* (PDR) and received signal strength at the RSU. The PDR is used to determine the functional range of the RSU. They observe that the physical configuration of RSUs can have a large effect on link quality and the placement of such should take into consideration the environment within the proposed coverage area.

Mecklenbräuker et al provide analysis of the signal propagation properties of 802.11p channels in different environments [24]. A highway-based measurement survey is also performed. Using downstream broadcast traffic and a range of data-rates (3-27Mbps), they measure *frame success-ratio* (FSR, delivered frames against transmitted frames). FSR is shown to decrease with distance from the RSU, but also changes dramatically in the presence of environmental features such as bridges. RSU coverage area is found to decrease as data-rate increases, with rates in the region of 3-6Mbps providing the largest coverage area.

Fewer studies have examined the use of TCP over 802.11p networks. Wang et al. measure the throughput of UDP/TCP flows over 802.11p at 3Mbps with varying service channel slot times [25]. They propose two methods for reducing bandwidth wasted due to unused residual slot time (either by fragmenting queued frames or finding frames that can be transmitted within the remaining time). Although the proposed schemes are shown to improve TCP performance, a general assessment of TCP over 802.11p is not conducted. The effects of round-trip time, packet loss and TCP-level retransmission are not detailed in the paper.

Vandenbergh et al. have previously shown that 802.11p can be approximated using “off-the-shelf”

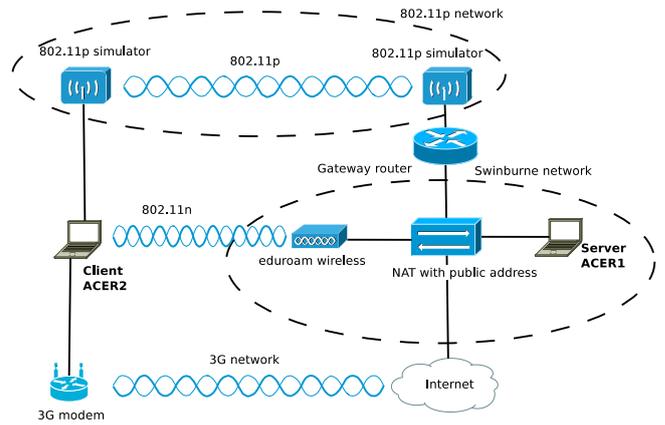


Figure 2. The experimental testbed. A public-facing server provided access via the Internet.

802.11a hardware [14]. Using only software modifications, they are able to disable beaconing and replicate the bandwidth, modulation and EDCA aspects of the 802.11p standard. They are not however able to set a 10MHz channel bandwidth, and the time-domain/bandwidth modifications are achieved by padding data frames, rather than re-clocking of the wireless cards. They do not implement channel hopping between the CCH and SCH, staying permanently tuned to the SCH.

IV. TESTBED

The testbed topology consists of a single-homed server and a multi-homed client (Figure 2). Both are Intel Atom N270 netbooks running Ubuntu Linux 12.10, patched with v0.88 of the MPTCP kernel [7].

The server is connected via Fast Ethernet to the Swinburne University network and is assigned a static IP address. Due to firewall restrictions, the address of the server is not directly accessible via the Internet, and another host is used as a gateway. The server and gateway were configured so that inbound/outbound traffic followed the same path.

The client has three interfaces: a Huawei E1750 3G USB modem, Fast Ethernet and an internal 802.11b/g/n Wi-Fi card, each connected to a different subnet.

The internal Wi-Fi was associated with the Swinburne wireless network at 39Mbps using 802.11n. This network is shared by staff and students and provides access to the Internet. The 3G modem has a maximum download throughput of 7.2Mbps and obtained Internet connectivity through mobile provider Vodafone. The Fast Ethernet port was connected to a 802.11p bridge. A corresponding

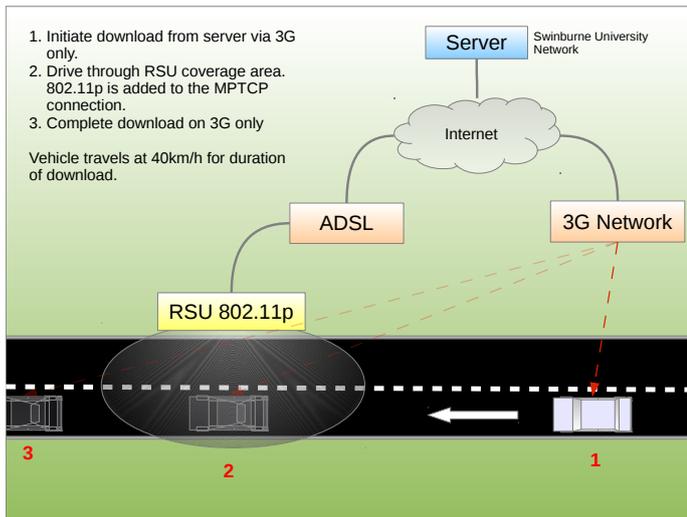


Figure 3. Field Trial setup. Connections to the server were established via residential ADSL and 3G Modem.

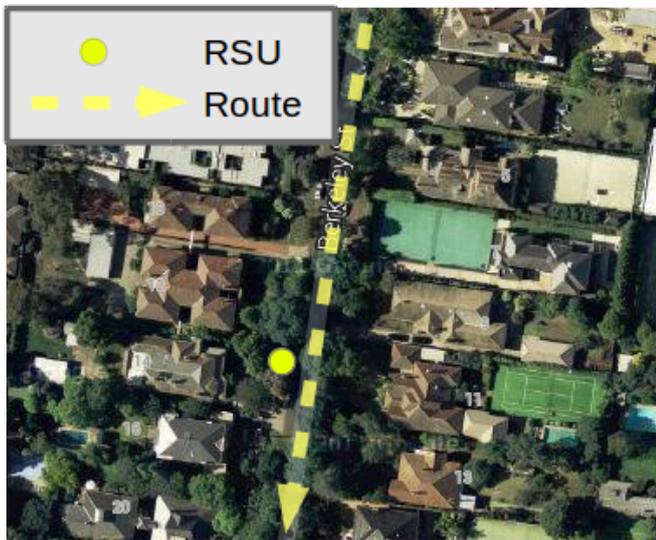


Figure 4. Field trial location and route. Source: Google Maps

802.11p unit was connected to a router on the Swinburne academic research network for Internet access.

Each of these interfaces accessed the server via the same public-facing IP address.

A. Field Trial

The topology of the field test is shown in Figure 3. The experiment was performed on a residential street featuring a mix of apartments and detached houses, with trees spaced every 10-15 metres along the roadside (Figure 4). The vehicle was driven north-to-south for each trial, at times when no other traffic was present.

An 802.11p bridge was mounted to the roof of a vehicle and connected to the client netbook. The 3G

modem was also connected to the client. The server was unchanged from the previous setup (Figure 2).

A corresponding 802.11p unit acted as the RSU transceiver and was mounted at a height of 2m, approximately 2m from the roadside. It was connected via Ethernet to a router in a nearby apartment. The data-rate was configured at 6Mbps. RSU back-haul was provided via a residential ADSL connection of 16Mbps downstream with a mean RTT of 16ms to the Swinburne network.

B. IEEE 802.11p Approximation

We approximate 802.11p using off-the-shelf 802.11a WLAN cards. As the 802.11p PHY layer is derived from 802.11a, it is possible to adapt 802.11a WLAN cards to act like 802.11p devices [14]. We do not emulate the upper layers of the IEEE 1609 standard (1609.1-4). We use Atheros AR5004-based Wistron CM9 802.11a/b/g miniPCI cards. These are installed in AMD Geode-based embedded computers running NanoBSD (built from FreeBSD-10). The regulatory domain, operating channel and EDCA were adjusted to match 802.11p specifications:

- Channel Bandwidth: The channel bandwidth is set at 10MHz, half the bandwidth of a standard 802.11a channel. The bandwidth is set by changing the clock rate of the WLAN card. This also doubles the time domain behaviour (E.g. symbol rate) to match that of 802.11p.
- Operating Channel: The interface is configured to use channel 165 (5.825 GHz), the highest 802.11a channel in ISM band. The actual 802.11p standard uses channels between 5.850-5.925 GHz, however a license is required to operate transceivers in this band. The difference in frequency is assumed to be negligible in this case.
- Service Channel only: As in [14], we do not implement channel hopping between the CCH and SCH. As a result we do not limit transmissions to the channel slot time as per IEEE 1609.4 (typically 100ms). We use only two nodes (a single OBU and RSU) so contention on the channel is between these transmitters only.
- Data-rate: A bit-rate of 6Mbps is used. [24] showed this to be a suitable data-rate for maximising range in V2I scenarios.
- Service set: As V2I connections can be short-lived and safety applications are delay sensitive, association, authentication and beaconing are not used at the 802.11p MAC layer. We configure the radios

to operate in *Adoc-hoc demo* mode, which does not use authentication or beaconing, and allows wireless cards on the same channel to transmit packets as long as they are within range.

- Lower MAC: EDCA contention parameters are set for each AC using the Wireless Multi-media Extensions (WME) in FreeBSD. As we are interested in non-safety applications, we configure our traffic class to match that of 802.11p *Best Effort*. The Best Effort traffic class, as defined in the 802.11p standard, is less aggressive in its use of the radio channel, with a longer AIFS period and disabled TXOP/bursting.

C. Connection parameters

We have changed the default TCP parameters of both client and server laptops, consistent with the settings used in [16]. TCP parameter caching is disabled (*no_metric_save=1*) and each new connection begins with a slow-start threshold of 64KB. The maximum received memory allocation (*rmem*) is set to 8MB to allow for buffering of out-of-sequence data.

V. EXPERIMENTAL METHODOLOGY

We investigate three scenarios for MPTCP: 3G with multiple Wi-Fi variants, simulated mobility with 3G and Wi-Fi, and a mobility field trial using 3G and 802.11p. In each of these experiments *wget*¹ is used to download files from the server via HTTP. File sizes range from 64KB to 64MB.

A. 3G with Wi-Fi Variants

As shown in Figure 2, the client has three physical interfaces through which it can connect to the single-homed server. We conduct measurements with following configurations:

- Single-path TCP: Client connects to the server using only one interface (802.11p or Wi-Fi or 3G).
- 2-path MPTCP: Client uses two interfaces to connect to the server. There are three possible configurations {(802.11n, 802.11p), (802.11n, 3G), (802.11p, 3G)}. We also change the default interface (i.e., the interface that initiates the connection and forms the first subflow).
- 3-path MPTCP: All three interfaces are used to connect to the server (802.11p, Wi-Fi, 3G).

¹A “non-interactive network downloader” provided as standard on many Linux distributions

Table I
MEAN RATE OF RETRANSMISSION FOR SINGLE INTERFACE TESTS

	File size			
Medium	64KB	512KB	4MB	16MB
3G	0.37%	1.08%	0.12%	1.84%
802.11n	0.00%	0.00%	0.02%	0.04%
802.11p	0.00%	0.00%	1.34%	0.64%

Table II
MEAN THROUGHPUT FOR SINGLE INTERFACE TESTS (Mbps)

	File size			
Medium	64KB	512KB	4MB	16MB
3G	1.47	2.61	3.77	3.65
802.11n	7.79	20.41	31.65	38.89
802.11p	2.89	3.34	3.40	3.41

B. Simulated mobility with 3G and Wi-Fi

We created a mobile vehicle scenario using a combination of 3G and a single Wi-Fi interface (802.11n or 802.11p). The 3G interface emulates “high-availability” network access while Wi-Fi provides intermittent connectivity.

We simulate a vehicle passing through the coverage area of an access point while downloading a 16MB or 64MB file. We choose a 10-second window of Wi-Fi availability, based on findings in [3].

The client starts downloading a file using 3G. The Wi-Fi interface is brought up after 5 seconds and is active for approximately 10 seconds before being brought down. The connection continues using 3G until the file transfer is complete. We test 802.11p and 802.11n as both could potentially see wider deployment in the future (although 802.11n is not a DSRC standard, [3] has shown that public Wi-Fi access points are accessible from vehicles).

The 802.11p link operates without association, as per WAVE/DSRC specifications. The 802.11n link operates in an infrastructure mode BSS, and must perform Layer 2 authentication and obtain an IP address. Association can take several seconds, which is consistent with the association time when moving in a vehicle at 50km/h and above [4].

We focus on whether the Wi-Fi interface is able to contribute bandwidth to the connection, and the trade off between any contribution and the costs of adding/removing the Wi-Fi subflow (E.g. data re-ordered, retransmissions).

VI. PATH CHARACTERISTICS

We obtain baseline measurements of each link in the testbed by performing downloads using single-path TCP.

Table III
MEAN RTT FOR SINGLE INTERFACE TESTS (MS)

	File size			
	64KB	512KB	4MB	16MB
3G	68.54	121.29	233.68	448.03
802.11n	12.81	13.96	20.52	42.09
802.11p	44.93	100.58	357.22	453.29

Each file was downloaded ten times and the results are aggregated.

A. Loss

Table I shows the mean percentage loss for each interface. Loss is calculated as the number of data packets requiring retransmission over the total number of data packets transmitted.

The 3G interface has a loss rate that is more consistent, and generally higher, than the Wi-Fi interfaces. The Wi-Fi interfaces experience loss only for larger flows, where there is time for the congestion window to grow large enough to trigger a loss.

The 802.11p interface shows unusually high loss when downloading the 4MB file. In this case there appears to be a correlation between the link rate, file size and the time at which the first losses occur due to congestion window growth.

Figures 5 and 6 show the throughput and RTT of the 802.11p link for a typical 16MB and 4MB transfer. We can see an increase in RTT due to queuing and congestion window growth for the first 8 seconds, after which time it appears the first loss has occurred and the RTT shrinks. The time taken for this loss to occur appears to be slightly less than that required to transfer a 4MB file. As the loss occurs near the end of the file transfer, the average throughput of the connection (throughput calculation is described in Section VI-B below) is comparable to that seen for the 16MB file transfer. Larger files (Figure 5) are able to average this error out as more data is transferred.

B. Throughput

Table II shows the mean throughput for each interface. Throughput is calculated as the number of data-carrying segments received divided by the download time. The download time is defined as the time between the *first SYN* of the TCP handshake and the *last ACK* of TCP teardown.

The throughputs were as expected. The 802.11n link, which is negotiated at 39Mbps, is able to approach this speed when downloading larger files. The 802.11p link,

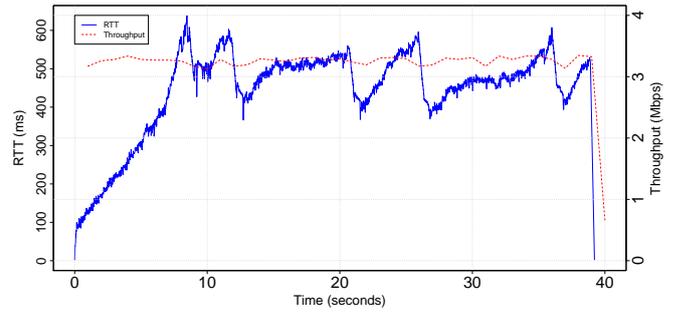


Figure 5. RTT and throughput for 802.11p interface over time, 16MB file

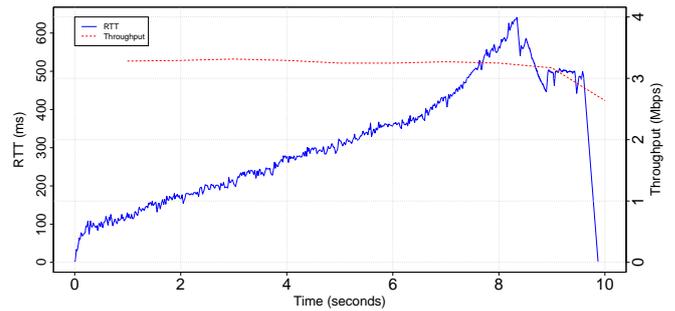


Figure 6. RTT and throughput for 802.11p interface over time, 4MB file

with a data-rate of 6Mbps and low-priority traffic class, achieves a similar throughput to that of the 3G modem (7.2Mbps).

C. RTT

Table III shows the mean RTT for each interface. We calculated RTT using the *Synthetic Packet Pair* (SPP) [26] utility.

The 802.11n interface has a lower overall RTT, as the faster transmission rate results in less queuing. The bit-rate, low loss and conservative transmission strategy of the 802.11p bridge contribute to queuing and as a result the RTTs for the link are much higher than that of 802.11n, but comparable to that of 3G.

VII. MULTIPATH PERFORMANCE

A. 3G and 802.11n (high asymmetry)

Figures 7 - 10 show the download times for MPTCP with 802.11n and 3G interfaces, compared with single-path 802.11n.

As observed in Section VI, 3G has significantly lower throughput and a higher RTT than the 802.11n path. This appears to negatively impact small file downloads (Figures 7, 8), as MPTCP-enabled flows take longer to

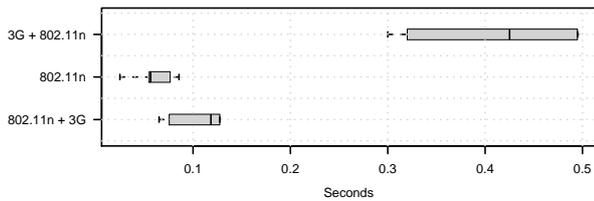


Figure 7. Download time for 64KB file, 802.11n and 3G. Single-path TCP over 802.11n is able to download the file in a shorter time period.

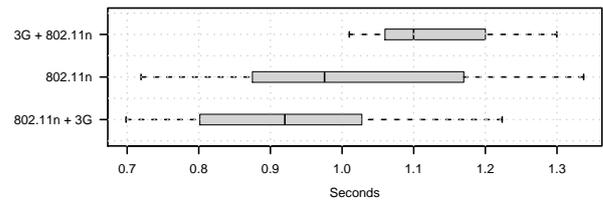


Figure 9. Download time for 4MB file, 802.11n and 3G. Multi-path TCP performs as good as single-path TCP, when the default interface is 802.11n.

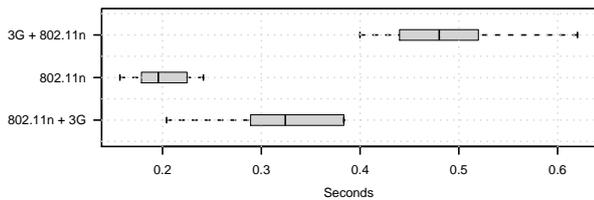


Figure 8. Download time for 512KB file, 802.11n and 3G. Single-path TCP over 802.11n is able to download the file in a shorter time period.

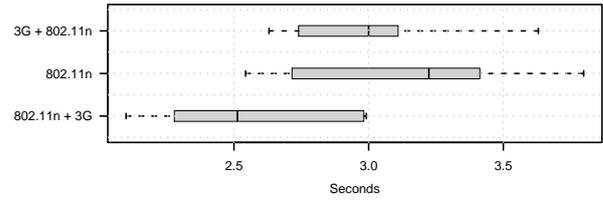


Figure 10. Download time for 16MB file, 802.11n and 3G. Multi-path TCP performs at least as good as single-path TCP.

complete than a single-path flow using 802.11n. In these cases, the rate at which the connection can be completed is bounded by the path with the longest RTT, and as IW10 is enabled by default, each path must send at least 10 segments.

As a result, although the 802.11n path is able to rapidly expand window size during slow start, receiving most of the data, the connection must wait for the segments allocated to the 3G path to arrive before completing. This is shown in Figure 11, which plots data-sequence numbers received against time. We can see data arriving over the 802.11n path, before head-of-line blocking occurs and the connection must wait for segments that have been sent on the 3G path.

The default interface also influences download time. Multi-path connections that started on the 3G path took longer to complete than those that started on the 802.11n path. As the second interface is only advertised after the first subflow becomes established, and must then go through a 3-way handshake, there is a delay before the capacity of the second interface can be used. As the RTT of the 3G path was multiple times greater than the 802.11n path, the delay in establishing the 802.11n subflow represents a time frame during which a larger portion of the file could have been transmitted.

A secondary effect of using 3G as the default interface is that the 3G subflow is able to spend more time in slow start and is thus allocated more data segments, which

contributes to head-of-line blocking. Figure 12 shows the data-sequence numbers over time when the 3G path was the default interface. Data segments arrive later on the 3G path, causing head-of-line blocking and delaying the progression of the data sequence.

For larger files (Figures 9 and 10), using multiple paths trended towards decreased download times, if only by marginal amounts. The mean download time for a 16MB file decreased by roughly 700ms, while the mean download time of a 4MB file decreased by roughly 50ms.

The portion of data transmitted on the 3G link is a small (approximately 5%, see Figures 13 and 14).

Although download times did decrease with MPTCP enabled, the difference would likely not be noticed by the user, and the range of download times across single-path and multi-path connections had substantial overlap. It should be noted however that this is not the case when considering mobility, and using MPTCP with mobile hosts that have diverse paths can in fact be highly beneficial (see Section VIII).

B. 3G and 802.11p (low asymmetry)

Section VI showed the 802.11p and 3G paths to be comparable in terms of RTT, with 3G achieving slightly higher throughputs over longer file downloads. The 3G path did however have a higher loss rate. Figures 15 - 18 show file download times when using MPTCP over 802.11p and 3G interfaces, compared with single-path 802.11p and single-path 3G.

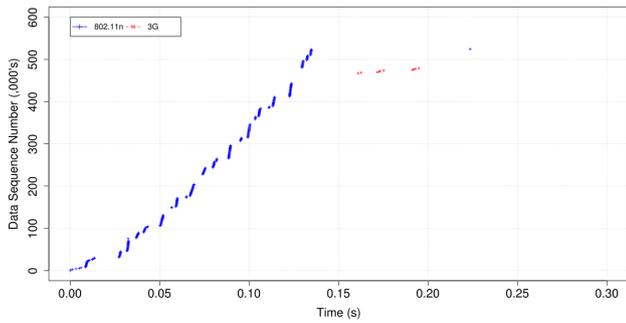


Figure 11. Client-side Data Sequence Number Vs Time for a 512KB file using 802.11n and 3G. The segments are transmitted across both links simultaneously from the server, but are received at different times due to the delay of the 3G path. This causes head-of-line blocking and means the completion time is longer compared to single-path TCP over 802.11n.

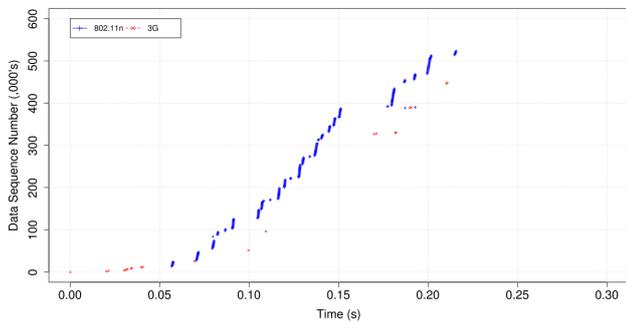


Figure 12. Client-side Data Sequence Number Vs Time for 512KB file. The connection was initiated over the 3G path. There is a delay in setting up the 802.11n subflow. Gaps along the x-axis indicate periods where no new data was received, an indication of head-of-line blocking.

Tests involving the use of 3G show a wider distribution of download times, which may be caused in part by the higher loss rate of the path (see Table I). These variations were less pronounced when using 3G together with 802.11n (Section VII-A), as the throughput and lower RTT of the 802.11n path was able to mask the effects.

For the 64KB download (Figure 15) there does not appear to be any benefit in using MPTCP over 802.11p alone. With a download size of 512KB (Figure 16) the download time improves when using MPTCP with 3G as the default interface (compared with 3G alone). Adding a 3G path to a connection initiated over 802.11p did not result in an improved download time.

Figure 19 shows the proportion of data downloaded on each of the paths, depending on the default interface. In both cases more data is transferred over the 802.11p path.

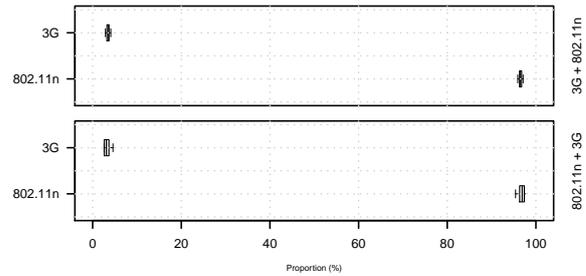


Figure 13. Proportion of data downloaded per interface, 4MB File. The 3G path contributes relatively little, given the longer RTT and lower bandwidth.

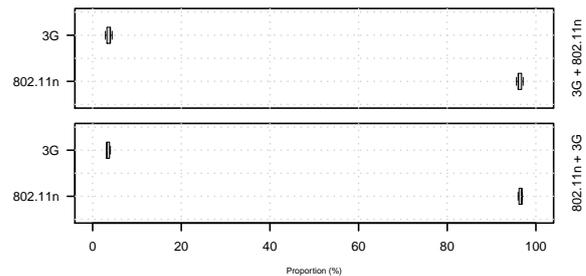


Figure 14. Proportion of data downloaded per interface, 16MB File. Even with the largest file, the proportion downloaded over the 3G path is relatively small.

Download times decrease for the 4MB and 16MB files when using multiple interfaces. The proportion of data transferred is relatively evenly weighted (Figure 20), though the 802.11p path is preferred. Comparing the figures for download times and proportion of data per interface, it appears that tests in which more data is transferred over 3G resulted in greater variation in download times. This may be related to the higher loss rate of the path, as losses increase the retransmissions required and are also a cause of head-of-line blocking at the data-sequence level.

As with the results in Section VII-A, the tests suggest that MPTCP (with the default congestion control algorithm) provides greater benefit for longer connections, as subflow RTT has proportionally less effect on the connection and congestion control is able to better react to events such as packet loss. However in this case the gains were larger due to the similar throughput of the links.

C. 802.11n, 802.11p and 3G

We also test MPTCP using all three interfaces with 802.11n as the default interface. Figure 21 shows the download times for each of the files.

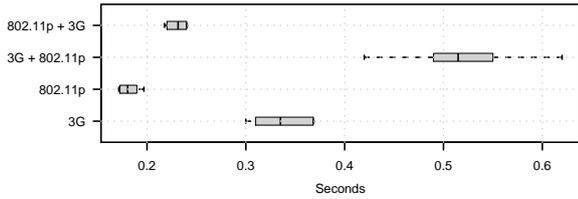


Figure 15. Download time for 64KB file, 802.11p and 3G. The higher RTT of the 3G path increases download times.

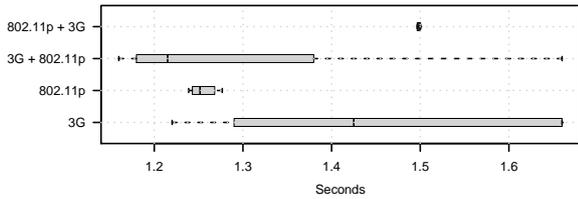


Figure 16. Download time for 512KB file, 802.11p and 3G. Using multiple paths did not make a significant impact.

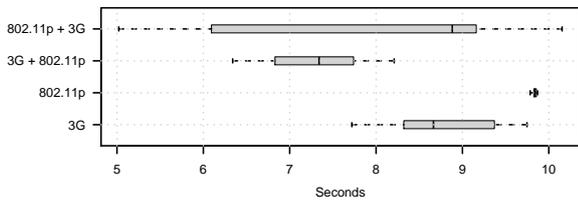


Figure 17. Download time for 4MB file, 802.11p and 3G. Some marginal improvements, however completion time has high variability.

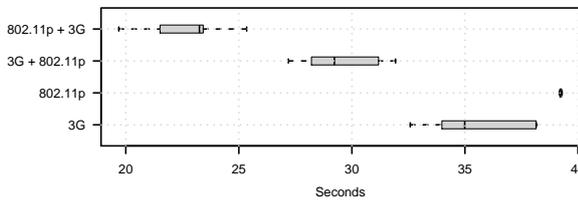


Figure 18. Download time for 16MB file, 802.11p and 3G. Downloading a larger file provided the biggest improvement when using multiple paths, with mean download time reductions in the range of 10s of seconds.

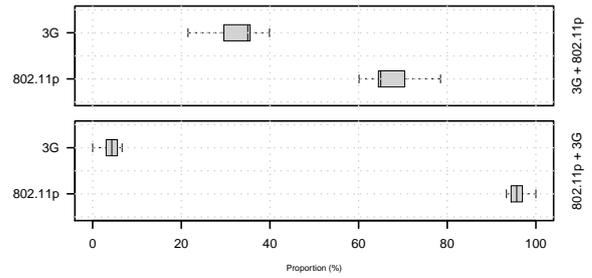


Figure 19. Proportion of data transferred, 802.11p and 3G. 512KB file. The lower RTT of the 802.11p path results in more data being downloaded over this link.

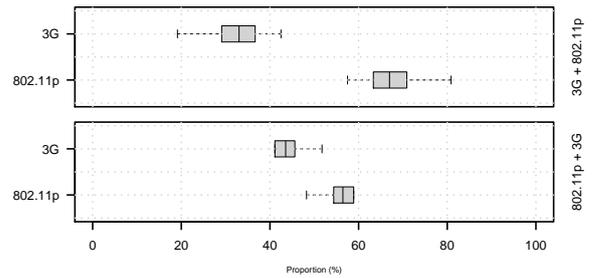


Figure 20. Proportion downloaded per interface, 802.11p and 3G. 4MB file. The download proportions are more even as the amount of data downloaded increases.

As in previous tests, there was little or no benefit in using MPTCP when downloading small files. There was however a slight benefit when downloading the 16MB file (a reduction of a little over a second). In each test the majority of data was transferred over the 802.11n link, with the 802.11p and 3G paths contributing a smaller, roughly equal volume of data.

VIII. SIMULATED MOBILITY

We simulate a mobile scenario in which an MPTCP-enabled host passes a WiFi (802.11n, 802.11p) access point after a connection is initiated over 3G. The WiFi coverage is available for a duration of 10 seconds and each test was repeated 10 times. Figures 22 and 23 plot the throughput of each subflow and the combined throughput, for a single trial of 802.11n and 802.11p.

Both the 802.11n and 802.11p paths are able to provide a short boost in overall download throughput. Table IV shows the average bytes contributed by the WiFi interfaces while they were associated, and maximum size data-level retransmit required over 3G once the WiFi path was removed. Retransmissions were calculated by matching DSNs transmitted over the 3G path that were

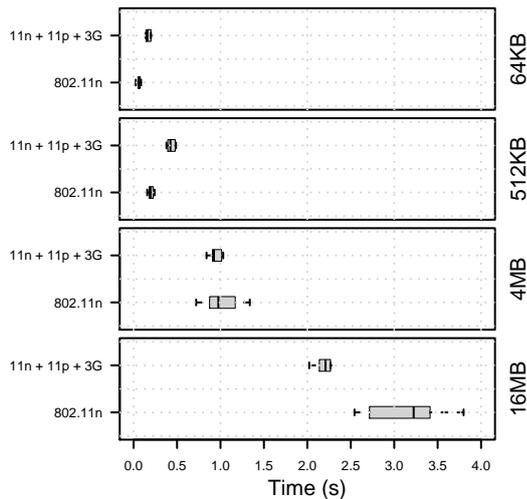


Figure 21. Download times using all interfaces, compared with 802.11n alone. As 802.11n has significantly higher bandwidth than the two additional paths, there is not a significant decrease in download times.

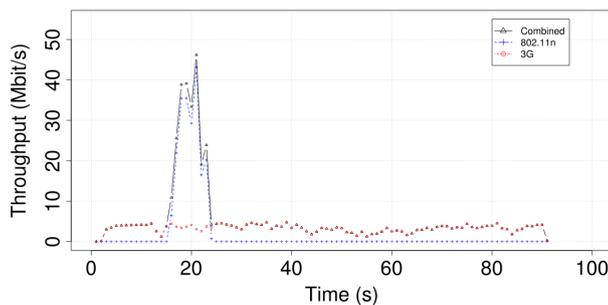


Figure 22. MPTCP handover with 802.11n. The 802.11n is able to contribute a significant boost in throughput over 10 seconds, which includes time taken for 802.11 authentication.

first transmitted over the WiFi path, after the WiFi path was removed.

	Mean contribution (MB)	Max retransmission (MB)
802.11n	26.6	0.37
802.11p	3.11	0.33

Table IV

MEAN CONTRIBUTION OF BYTES OVER THE 10 SECONDS OF WiFi AVAILABILITY, AND THE MAXIMUM DATA-LEVEL RETRANSMISSION SEEN ACROSS THE TESTS.

In both WiFi tests, the amount of data contributed was much greater than the retransmissions required once the path was removed. Although the sudden removal of a subflow causes subflow-level retransmission due to

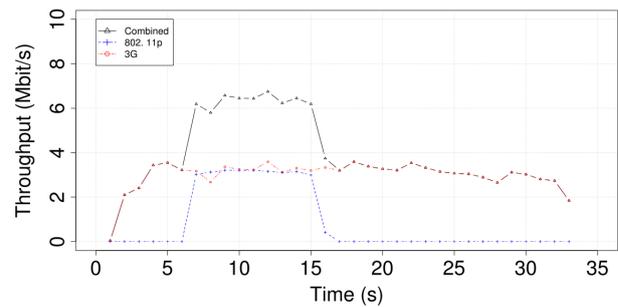


Figure 23. MPTCP handover with 802.11p (6Mbps).

undelivered ACKs, data-level ACKs can be transmitted on any subflow - thus the data that has been received at the subflow level can be acknowledged at the data-level on the remaining path (in this case 3G). This data does not need to be retransmitted. However out-of-sequence subflow-level data that has been buffered at the receiver does require retransmission. Data that was sent but not received must be retransmitted. The amount of data requiring retransmission is similar for both the 802.11n and 802.11p interfaces, even though the throughput of 802.11n is much higher. In this case it appears that the data-level retransmission timeout strategy is able to detect a lack of data-acknowledgments within a similar window.

Although the amount of data that requires retransmission is not large, there is an impact on the timeliness of application-level data delivery. During the time in which the connection is recovering from the removal of a path, there are typically out-of-order data-level segments that cannot be delivered to the application until retransmission occurs.

IX. FIELD TRIAL

We repeat the simulated scenario using an actual vehicle and an 802.11p access point acting as a RSU (see section IV-A). The test starts when the vehicle is out of the range of the RSU. The client begins downloading a 16MB file from the server using the 3G interface.

As it passes into range of the RSU it uses both the 3G and 802.11p interfaces to download the file. The 3G interface completes the file download once the RSU is out of range.

The car is driven at 40km/h for the duration of the connection, providing between 10-20 seconds of coverage. We were able to conduct six trials driving past the RSU. The throughput of one test is shown in Figure 24.

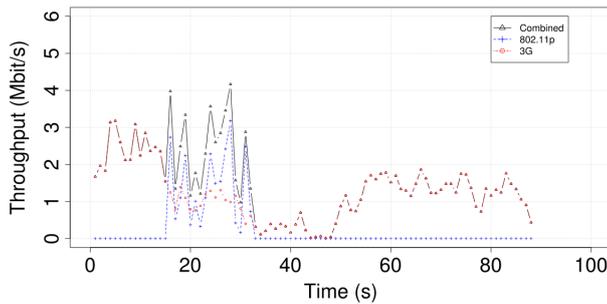


Figure 24. Throughput over time for field trial. The RSU was configured for 6Mbps operation. In this test the 802.11p RSU is able to provide ~1.5Mbps of additional throughput over a period of approximately 15 seconds.

The throughput on both channels is less consistent than seen in the simulated tests, as the changing terrain and mobility of the transceiver act as additional sources of channel fading. The 3G connection in particular varied considerably depending on the location along the street. However despite the variability the RSU was able to contribute an amount of data greater than the retransmission required once connectivity was lost (see Table V). The mean throughput achieved across the tests was approximately 1.5Mbps.

	Mean contribution (MB)	Max retransmission (MB)
802.11p	3.68	0.30

Table V
MEAN CONTRIBUTION OF ROADSIDE ACCESS POINT DURING CONNECTIVITY, AND THE MAXIMUM DATA-LEVEL RETRANSMISSION REQUIRED.

Although throughput was not equal to that in the static environment, it is likely that performance would improve with more optimal placement of the OBU and RSU antennae, and a higher transmission power (i.e. with planned, dedicated infrastructure and DSRC-band equipment). As such there appears to be some potential in using MPTCP over 802.11p-based roadside infrastructure to supplement cellular network connections.

A. Discussion

[16] found that using multiple subflows was beneficial for both short and long flows, and suggested an optimisation whereby additional subflows were joined prior to MPTCP-level connection establishment in order to further exploit the available paths. We found using MPTCP to not always be beneficial when available links

are highly diverse in terms of bandwidth, RTT or loss rate.

In such cases download times for short flows increased, and improvements gained on longer flows were marginal. Therefore, even though small gains are possible when using heterogeneous links, they should be weighed against the costs associated with adding an additional path (energy consumption, network costs and delay due to out-of-order packet arrival).

We found that MPTCP was able to function over 3G and limited-availability 802.11p when performing bulk data transfers. The costs of adding the V2I path to an existing connection, in terms of data-level retransmissions, were outweighed by the amount of data contributed during the connected period. Bulk-transfer applications are however more tolerant of the 'buffer-delay' caused by data-level retransmissions, and it is unclear what affect this would have on more delay-sensitive applications.

Minimising buffer-delay depends not only on congestion control, but on the data-level retransmission strategy used in the implementation. For scenarios where link availability is expected to be transient, the data-level retransmission timeout would need to be adapted in order to quickly respond to the disappearance of a link and reduce delays in delivery to the receiving application.

X. CONCLUSIONS

In this report we studied the performance of MPTCP using WiFi (802.11n, 802.11p) and 3G with several static and mobile scenarios. Measurements were performed on live networks and across the Internet. We compare throughput and download time when using MPTCP against single-path TCP. We measure the throughput of an MPTCP connection in mobile scenarios using 3G and WiFi, and present results of a small-scale field trial using MPTCP over 3G/802.11p.

We find that when using MPTCP across paths of similar characteristics, download times are in most cases at least as good as single-path TCP over the best link. When paths are asymmetric (in terms of RTT and bandwidth), we show that MPTCP can perform worse than single-path TCP, particularly for flows transferring less than 1MB that do not exit slow-start.

We show that by using MPTCP with roadside infrastructure, it is possible to boost the throughput of a TCP connection initiated over 3G. The MPTCP connection was able to remain active and perform handover at 40km/h, without adversely affecting a delay-tolerate file download or causing excessive data-level retransmissions.

We find that using MPTCP for V2I TCP connections is feasible, and could provide connection persistence, additional bandwidth, or be used as an alternate low-cost path.

XI. FURTHER WORK

This report presents only a preliminary investigation into the use of MPTCP for V2I purposes. The experiments can be extended in a number of ways to more accurately reflect real-world 802.11p conditions:

- 1609.4 channel access mechanism: As defined in the WAVE upper layers, this would limit the transmission window on the service channel to a maximum of 100ms per window.
- Node contention: We have tested the best-case scenario of a single node accessing the RSU. 802.11p channel throughput is expected to change as the number of nodes increases.
- Access Class contention: We test a single, low-priority traffic class. Adding high-priority cross-traffic would further limit the transmission window for “best-effort” flows.
- Handover between RSUs: It is expected that RSUs would be deployed in relatively high density to provide consistent coverage, thus handover between neighbouring RSUs should be examined.
- Cross-channel interference: A RSU would typically be providing services on multiple adjacent channels, which may potentially interfere with one another.

XII. ACKNOWLEDGEMENTS

This project has been made possible by a gift from the Cisco University Research Program Fund, a corporate advised fund of the Silicon Valley Community Foundation.

REFERENCES

- [1] Y. J. Li, “An Overview of the DSRC/WAVE Technology,” in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 74, 2010.
- [2] M. Amadeo, C. Campolo, and A. Molinaro, “Enhancing IEEE 802.11P/WAVE to Provide Infotainment Applications in VANETs,” *Ad Hoc Networks*, no. 2, March 2012.
- [3] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden, “A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks,” in *12th ACM MobiCom Conference*, Los Angeles, USA, September 2006.
- [4] R. Gass, J. Scott, and C. Diot, “Measurements of In-Motion 802.11 Networking,” in *Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications*, Orcas Island, USA, 2006.
- [5] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “TCP Extensions for Multipath Operation with Multiple Addresses,” RFC 6824, Internet Engineering Task Force, January 2013. [Online]. Available: <http://tools.ietf.org/html/rfc6824>
- [6] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, “Exploring Mobile/WiFi Handover with Multipath TCP,” in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, Helsinki, Finland, 2012.
- [7] O. Bonaventure. (2013) MultiPath TCP Linux Kernel Implementation. [Online]. Available: <http://multipath-tcp.org/pmwiki.php>
- [8] D. Wischik, C. Raiciu, A. Greenhalgh and M. Handley, “Design, Implementation and Evaluation of Congestion Control for Multipath TCP,” in *USENIX Symposium of Networked Systems Design and Implementation (NSDI’11)*, Boston, MA, 2012.
- [9] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, “Increasing TCP’s Initial Window,” RFC 6928, Internet Engineering Task Force, April 2013. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6928.txt>
- [10] C. Raiciu, M. Handly, and D. Wischik, “Coupled Congestion Control for Multipath Transport Protocols,” RFC 6356, Internet Engineering Task Force, October 2011. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6356.txt>
- [11] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec, “MPTCP is Not Pareto-optimal: Performance Issues and a Possible Solution,” in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, Nice, France, 2012.
- [12] R. Khalili, N. Gast, M. Popovic, and J.-Y. L. Boudec, “Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP,” August 2013. [Online]. Available: <http://tools.ietf.org/html/draft-khalili-mptcp-congestion-control-04>
- [13] T. Le, “Improving the Performance of Multipath Congestion Control over Wireless Networks,” in *The 2013 International Conference on Advanced Technologies for Communications (ATC’13)*, Ho Chi Minh City, Vietnam, 2012.
- [14] W. Vandenberghe, I. Moerman, and P. Demeester, “Approximation if the IEEE 802.11p Standard Using Commercial Off-The-Shelf IEEE 802.11a Hardware,” in *2011 11th International conference on ITS telecommunications (ITST)*, St Petersburg, Russia, August 2011.
- [15] I. Msadaa and P. Cataldi and F. Filali, “A Comparative Study between 802.11p and Mobile WiMAX-based V2I Communication Networks,” in *Fourth International Conference on Next Generation Mobile Applications, Services and Technologies*, Amman, Jordan, July 2010.
- [16] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, “A Measurement-based Study of MultiPath TCP Performance over Wireless Networks,” in *Proceedings of the 2013 Conference on Internet Measurement Conference*, Barcelona, Spain, 2013.
- [17] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley, “Opportunistic Mobility with Multipath TCP,” in *Proceedings of the Sixth International Workshop on MobiArch*, Bethesda, Maryland, USA, 2011.
- [18] J. Gonzalez, M. Sepulcre, and R. Bauza, “IEEE 802.11p Vehicle to Infrastructure Communications in Urban Environments,” *IEEE Communications Magazine*, May 2012.
- [19] C. Han and M. Dianati and R. Tafazolli and R. Kernchen and X. Shen, “Analytical Study of the IEEE 802.11p MAC Sub-layer in Vehicular Networks,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, June 2012.

- [20] A. Jafari and S. Al-Khayatt and A. Dogman, "Performance Evaluation of IEEE 802.11p for Vehicular Communication Networks," in *8th IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing*, Poznan, Poland, 2012.
- [21] N. Sun, "Performance Study of IEEE 802.11p for Vehicle to Vehicle Communications Using OPNET," Master's thesis, Massey University, Auckland, New Zealand, 2011.
- [22] Y. Wang and A. Ahmed and B. Krishnamachari and K. Psounis, "IEEE 802.11p Performance Evaluation and Protocol Enhancement," in *IEEE International Conference on Vehicular Electronics and Safety*, Columbus, USA, 2008.
- [23] S. Eichler, "Performance Evaluation of the IEEE 802.11p WAVE Communication Standard," in *IEEE Vehicular Technology Conference*, Baltimore, USA, October 2007.
- [24] C. Mecklenbrauker and J. Karedal and A. Paier and T. Zeman and N. Czink, "Vehicular Channel Characterization and its Implications for Wireless System Design and Performance," vol. 99, no. 7, July 2011.
- [25] S. Wang, H. Chao, K. Liu, T. He, C. Lin, and C. Chou, "Evaluating and Improving the TCP/UDP Performances of IEEE 802.11(p)/1609 Networks," in *IEEE Symposium on Computers and Communications*, Marrakech, Morocco, July 2008.
- [26] Synthetic Packet Pairs (SPP). [Online]. Available: <http://caia.swin.edu.au/tools/spp/>