Computer Communications 32 (2009) 268–280

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Increasing wide-area download throughputs on the roads by trunking multiple cellular channels over a vehicular ad hoc network

S.Y. Wang *, C.L. Chou, C.C. Lin

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 25 October 2006
Received in revised form 25 October 2008
Accepted 26 October 2008
Available online 8 November 2008

Keywords:
GPRS
WLAN
MANET
Trunking

ABSTRACT

To obtain infotainment services (e.g., traffic, maps, accidents, weather, etc.) on the roads, vehicle drivers and back-seat passengers may want to use wide-area networks (e.g., GPRS) to download useful information from the Internet when needed. In this paper, we design and implement a scheme to increase GPRS download throughput for a vehicle user while he (she) is moving on the roads. This scheme integrates inter-vehicle communication with trunking mechanisms to achieve this goal. This scheme can be readily deployed for any real-world GPRS network or any 3G network without any support from the network operator. Our field trial results show that when this scheme is applied in a motorcade of four vehicles and all of the GPRS channel provided by each vehicle are used to download a file simultaneously, the download speed is 3.92 times faster than that achieved when only one GPRS channel is used.

1. Introduction

Intelligent transportation systems (ITS) is an important research topic. ITS aims to provide vehicle users (including drivers and back-seat passengers) with safer, more efficient, and more comfortable trips. For example, ITS aims to provide vehicle users with timely traffic congestion and road condition information so that they can avoid congested or dangerous areas. In addition, ITS aims to provide vehicle users with infotainment services so that, for example, they can download useful information from the Internet (e.g., navigation maps, weather, etc.) and send/receive emails while on the roads. To achieve these goals, providing vehicle users with a high-availability communication service is important.

Although recently 3G wide-area networking service has been launched, 3G networks are still not widely deployed yet. Due to its low availability, using 3G networks to provide vehicle users with infotainment services on the roads so far is still very limited. Recently, deploying many short-range IEEE 802.11(a/b/g) access points along highways to provide infotainment services for vehicle users has been proposed. However, due to the short range of these access points, a vehicle will suffer from excessive performance drops caused by constant handoff between these access points. In addition, when vehicles move on roads without such infrastructure supports (e.g., in rural areas), they cannot obtain infotainment services by this approach.

General packet radio service (GPRS, i.e., 2.5G), in contrast, is widely available with current GSM (i.e., 2G) networks. It is a reliable data service that allows information to be sent and received over distances up to 35 km. Due to its current high availability, some commercial vehicle on-board-units (OBU) are using it to provide wide-area infotainment services. For example, the TOBE OBUs manufactured by Yulon Motor Corporation in Taiwan use GPRS for communicating with TOBE traffic information center.

Although GPRS provides several advantages such as high availability and support for high mobility, a vehicle user, however, receives very little transfer throughput over a GPRS channel. In the real world, the maximum download/upload throughput achieved is only 36/12 Kbps using the common 3 + 1 package, where 3 and 1 time slots are allocated to downlink and uplink traffic, respectively. On such a low-bandwidth network, applications that demand high bandwidth may not perform satisfactorily.

In this paper, we design and implement an integrated scheme to increase the GPRS download throughput for a vehicle user moving on the roads. This scheme can be applied in a motorcade composed of vehicles driven by the people who know each other, such as family, friends, colleagues, etc. Usually, the vehicles within this kind of motorcade have the same moving path and follow one another closely. Thus, the topology of this small vehicular ad hoc network (VANET) changes infrequently and the wireless communication paths among vehicles (one-hop or multi-hop) are always available during the traveling time. The main idea of our scheme is to download a requested file over multiple GPRS channels in parallel. If a user (can be an OBU) wants to download a large file through the GPRS network (this user is called the request originator in the following description), the user asks for bandwidth support from his (her) neighboring users belonging to the same motorcade. The neighboring users who are not using their GPRS

* Corresponding author.
E-mail address: shieyuan@cs.nctu.edu.tw (S.Y. Wang).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.10.011

bandwidth then use their GPRS channels to help download parts of the file. The content of the requested file then is downloaded over these GPRS channels in parallel. When these helpers receive packets over their GPRS channels, they forward these packets to the request originator through a VANET using IEEE 802.11(b) WLAN interfaces. Since the bandwidth of an IEEE 802.11(b) interface is 11 Mbps, which is much larger than the bandwidth of a GPRS channel, the request originator can simultaneously receive these packets without congestion. By this scheme, the time required for downloading a large file over a GPRS network can be greatly reduced.

The proposed scheme can be readily deployed for any real-world GPRS network or any 3G network without any support from the network operator. The proposed scheme only involves two daemon programs and one slightly modified web proxy server (Apache). One daemon program (Mobile Node Daemon) is run on every mobile host. The other daemon program (Trunk Daemon) and the web proxy server are run on an Internet host, which can be located in any subnet of the Internet. According to our experimental results, throughput speedups ranging between 3.92 and 2.98 can be achieved by using four GPRS channels in indoor, outdoor, or mobile environments.

Recently, cellular devices equipped with GSM/GPRS and WLAN interfaces have been introduced to the market (e.g., Nokia 6136). Such devices have the required network interfaces to use the proposed scheme. Since the scheme can be readily deployed for any real-world GPRS network, before expensive 3G networks and services are widely deployed, it provides a low-cost software solution for GSM/GPRS network operators to use their current 2.5G networks to provide high-bandwidth applications for their customers.

This paper makes the following contributions. First, we design a novel scheme that integrates a cellular network and a VANET. This scheme can be readily deployed for any real-life GPRS network or 3G network without any support from the network operator. In addition, no matter whether the GPRS/3G network operator assigns a GPRS/3G device a public or private IP address, this scheme still works correctly. Second, we implement this scheme in a novel way as user-level cooperating daemons. This implementation allows the proposed scheme to be easily ported to different operating systems such as Linux and Windows XP. Besides, real-life network applications (e.g., Mozilla Web browser) running on a mobile node can readily benefit from this scheme without being modified. Third, this scheme is really implemented in Linux and extensively evaluated in field trials using moving vehicles. In contrast, most related work only provides simulation results. The performance data reported in this paper are realistic. They are more convincing and more applicable to ITS environments than the simulation results reported in most related work.

In the rest of the paper, related work is presented in Section 2. Next, we present the architecture, design, and implementation of the integrated scheme in Section 3. In Section 4, the detailed design of the proposed protocols is presented. In Section 5, we present the experimental settings and calibration test results. In Section 6, we present the performance of the proposed scheme in indoor, outdoor, and mobile environments. In Section 7, we discuss the issues affecting the performance of the proposed scheme. In Section 8, we discuss some future work. Finally, Section 9 concludes this paper.

2. Related work

In the literature, several approaches have been proposed to integrate WLAN and GPRS (or other cellular) networks to achieve better performance [1–5]. However, these approaches differ from our scheme in both design and implementation. In addition, our scheme has been implemented and evaluated on a commercial GPRS network in indoor, outdoor, and mobile environments. In contrast, most of these existing approaches were only evaluated by simulations.

UCAN [1] aimed to reduce the file download time over a 3G CDMA2000 HDR (high data rate) network. In UCAN, a download request originator, called the destination node, chooses a neighboring node whose cellular channel's quality is better than that of its own channel to be its proxy node. When the destination node wants to download a large file through the HDR network, it informs the base station of its proxy node. After receiving this information, the base station sends the requested file to the proxy node and then the proxy node forwards the packets of the file to the destination node through a mobile ad hoc network (MANET).

The motivation for proposing UCAN was that a low-throughput file transfer is mainly due to the use of a low-quality channel; therefore finding another channel with a better quality to download the file could improve the file transfer throughput for a destination node that has a low-quality channel. Although in UCAN both HDR and WLAN are used (which is similar to our scheme), only one HDR channel is used to download a requested file at any time. This is different from our multi-channel scheme and it limits the maximum achievable download throughput to be a HDR channel's bandwidth. Moreover, in UCAN one needs supports from the network operator to modify base stations whereas in our scheme this is unnecessary.

In [2], the authors proposed a mechanism for smooth handovers between WLAN and GPRS networks. The authors defined the area reachable from a base station in single hop as a cell. If the source node and the destination node stay in the same cell, the packets sent from the source node to the destination node need not pass through the base station. Instead, other mobile nodes located in the same cell can use a MANET to relay these packets. In this case, a multi-hop routing mechanism has to be involved. If the source node and destination node do not stay in the same cell, the source node first sends its packets to the base station with which it is associated. Then the packets are forwarded to the base station with which the destination node is associated through the GPRS backbone network. Finally, the destination node receives the packets through its GPRS channel. The transmission path between the source node and its associated base station may be one-hop (using GPRS) or multi-hop (using MANET), so is the path between the destination node and its associated base station. This approach aims to smoothly switch the transmission path between a WLAN and a GPRS networks. Its goal, design, and implementation are different from those of our scheme.

In iCAR [3], some ad hoc relay stations (ARSs) are placed at strategic locations so that a mobile host's packets can be relayed by these ARSs to a base station that is not the mobile node's associated base station. If a cell has a heavy traffic load and its adjacent cells do not have heavy loads, a mobile host's packets can be diverted from its cell to an adjacent cell so that its cell's congestion can be reduced. The iCAR approach could reduce the call blocking probability for circuit-like traffic by diverting traffic from heavily-loaded cells to nearby lightly-loaded cells. Since a mobile host still uses only a channel in iCAR, the goal, design, and implementation of iCAR are different from those of our scheme.

Another system is MADF [4]. The authors defined a cell with a heavy traffic load as a hot cell while a cell with a light traffic load as a cold cell. MADF uses a dynamic channel allocation scheme to assign more channels to hot cells. In MADF, ad hoc overlay devices are employed and located at the places between hot cells and cold cells. Users in hot cells can connect to adjacent cold cells through these ad hoc relay devices over specific channels. MADF prevents packet delay from increasing in a heavily-loaded cell. This approach attempts to overcome congestion within a heavily-loaded cell. It is different from our multi-channel scheme.

In [5], the authors attempted to improve the reliability of coverage in cellular networks. Its design extends a mobile phone call by employing an intermediary to carry this call. Thus, a mobile phone call may become a 2-hop call. The intermediary could be a mobile device or a vehicle with sufficient power resources. A MANET is employed to achieve better reliability in a cellular network; however, it may increase the intra-cell interference. The goals, design, and implementation of this scheme are different from those of our scheme.

In [6], the authors presented a framework for creating transport protocols for mobile hosts to simultaneously use multiple link layers for the same connection. In this study, each mobile host has multiple wireless interfaces and the traffic of a connection is striped over these interfaces. The striping algorithm is employed at the transport layer. This striping scheme does not integrate WLAN and GPRS and thus is different from our scheme.

In [7], the authors proposed MOPED (MObile groupPEd Device), which is a network model that treats a set of personal devices as a single virtual device. This virtual device dynamically aggregates available communication resources to provide network service. For example, a user may have a PDA with a cellular modem, a cell phone, and a laptop computer without Internet connectivity. These three devices can communicate with each other through some PAN (personal area network) technologies (e.g., Bluetooth, low-power IEEE 802.11). While the user is talking on the phone, Internet traffic for the laptop computer may be routed through the PDA. To support this, a MOPED proxy which routes flows for different endpoints is needed. In this study, the focus was on the routing approach within a dynamically formed MOPED. Its goals, design, and implementation are different from those of our scheme.

The integrated network architecture proposed in [8] shares the same goal with our scheme because it also uses a striping approach to improve file download throughput over GPRS links. However, there are some differences between our paper and that paper. First, only simulation studies were carried out in [8], however, in this paper we carried out field trials on commercial GPRS networks in indoor, outdoor, and mobile environments to obtain realistic experimental results. Second, the channel allocation algorithms used are different. In [8], application requirements and channel conditions are considered to perform channel allocation and reallocation. In our scheme, we monitor the packet delivery rate achieved on each channel to know its current quality. Traffic then is dynamically striped over multiple channels based on their channel quality. The packet delivery rate can better reflect the quality of GPRS and VANET links, which may fluctuate due to relative motions among mobile nodes and base stations.

In [9], the authors proposed MAR, a commuter mobile access router infrastructure that exploits wireless diversity to provide improved data performance for wireless data users. MAR is a wireless multi-homed device that can be placed in moving vehicles, such as cars, buses and trains. The architecture of MAR is different from that of our scheme. An MAR router, which has multiple different kinds of wireless interfaces, may be mounted on the top of a vehicle. It aggregates all the available bandwidth of different wireless interfaces to form a larger data transmission pipe for users inside a vehicle to connect to the Internet. In MAR, the wireless interfaces are all fixed in the vehicle and thus there is no mobility issue involved when striping/collecting traffic over these interfaces. In contrast, since our scheme allows GPRS users to move, our scheme can stripe/collect traffic over a dynamic VANET.

In [10], an end-to-end transport layer protocol, called pTCP, was proposed. The original TCP is designed for connections that traverse a single path. Sometimes, however, multiple paths may be used by a connection simultaneously to increase throughput. pTCP allows connections to exploit the aggregate bandwidths offered by multiple paths. It does not integrate WLAN and GPRS and thus is different from our scheme.

In [11], the authors studied how to stripe traffic over multiple IEEE 802.11(b) channels. The differences between that paper and our paper are: (1) The studied channels are different. The characteristics of low-delay and high-bandwidth IEEE 802.11(b) channels and high-delay, high-delay-jitter, and low-bandwidth GPRS channels are quite different. (2) In [11], the radios for these IEEE 802.11(b) channels are located on the fixed sending or receiving machine. However, in our scheme, the radios for these GPRS channels are separately located on different mobile nodes.

3. System design and implementation

3.1. System architecture

A high-level architecture of the proposed scheme is depicted in Fig. 1, which shows the components of the whole system. The functionalities of each component are described below:

- *Web server (WS)*: WS is a content provider that provides data for users. It can be any on-line information server on the Internet such as a ftp or a web server.
- *Proxy server (PS)*: PS is a slightly-modified Apache proxy server that retrieves a requested file from a WS on behalf of a user. In our scheme, a daemon program (called the trunk daemon) runs on the same machine as the PS. It intercepts the packets sent from the PS to the user and sends them over multiple GPRS channels to the DN and its helping RNs (see their definitions below).
- *Relay node (RN)*: RN is a node that does not use its GPRS channel for itself but instead uses its channel to help download data for the DN (see its definition below).
- *Destination node (DN)*: DN is a node that uses GPRS channels to download a file (e.g., a web page) from the Internet. The requested file is downloaded via the DN's own channel and the channels of its helping RNs. When receiving packets carrying the content of the requested file from GPRS channels, these RNs relay them to the DN through a VANET. A DN needs to learn the existence of its neighboring nodes. It periodically exchanges hello packets with its neighboring nodes to collect such information.

A simple example is given below to illustrate how the system works. In Fig. 1, when the DN wants to download a web file from the WS located on the Internet, it sends a HTTP web request to

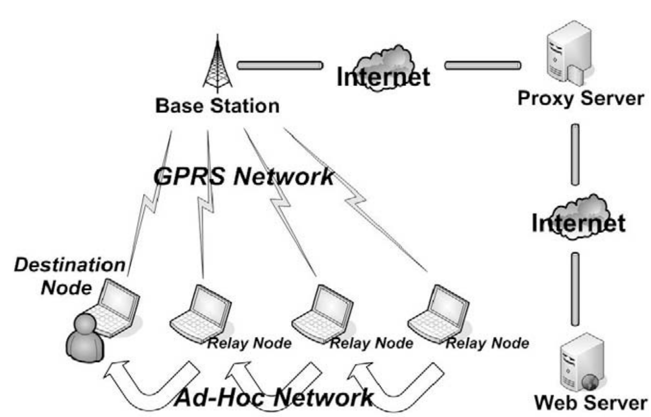

Fig. 1. The high-level system architecture of the proposed scheme.

the PS through its GPRS channel. At the same time, it sends the status of its neighboring nodes to the trunk daemon so that the trunk daemon can select some of them as the helping RNs for this DN. After the PS has retrieved the requested web file, it sends the web file to the DN. The trunk daemon intercepts these packets and then sends them over the GPRS channels of the DN and its helping RNs using a round-robin scheduling mechanism. To avoid sending packets over bad channels, the quality of each channel (packet delivery rate) is monitored every one second. Once a channel is detected as a bad channel, it is not used in the round-robin scheduling and retransmission mechanisms until its channel quality becomes good. When receiving packets from GPRS channels, these RNs then relay these packets to the DN through the mobile ad hoc network. Finally, the DN receives, re-sequences, and delivers all packets of the requested web file to the web browser, which then displays the downloaded file on the screen or saves it into a disk.

3.2. System implementation

3.2.1. Trunk daemon and web proxy server

In the following, a scenario is presented to illustrate how to download a requested file to a DN over multiple GPRS channels. In Fig. 2, the PS is a slightly modified apache server. When the PS receives the file request from the user, it informs the trunk daemon to install several firewall rules in the kernel. These firewall rules instruct the firewall facility in the kernel to intercept the data packets that will be sent from the PS to the DN. Packets captured by these rules are enqueued into the IP queue in the kernel. The trunk daemon then dequeues them from the IP queue and dispatches them to the DN and its helping RNs through their GPRS channels.

In the mean time, the PS sends the web request to a WS to fetch the requested web file. After the requested file comes back from the specified WS over an Internet path shown on the left, the PS sends the packets carrying the file's content to the DN. It uses the DN's IP address on the GPRS network as the destination IP address for these packets. As described above, these packets will be intercepted by the trunk daemon and then sent over multiple GPRS channels to reach the DN.

The trunk daemon sends the captured TCP packets (web content is transferred using the TCP protocol) to the DN and its helping RNs through a UDP socket. This means that these packets are encapsulated as UDP packets when they travel on GPRS channels. When sending a UDP packet to the DN or a RN, the trunk daemon uses the IP address of the receiving machine on the GPRS network as the destination IP address for the packet. Doing so enables the packet to be sent over the receiving machine's GPRS channel. The trunk daemon retransmits a UDP packet if it detects that the packet is lost on a GPRS channel. It implements an acknowledgment scheme running between itself and a DN or a RN to detect packet losses. This acknowledgment scheme also helps monitoring the GPRS channel quality. If the trunk daemon still fails to send a UDP packet to the DN or a RN after retransmitting it three times, it will think the GPRS channel is under bad condition and stop pumping packets into this GPRS channel. Instead, the trunk daemon start sending out a probe packet through this GPRS channel every one second until a response packet is received. The DN or RN is responsible to send back a response packet to the trunk daemon if it receives the probe packet. Upon receiving the response packet, the trunk daemon reactivates the suspended GPRS channel.

When the requested file has been downloaded to the DN, the PS notifies the trunk daemon of the completion of the file transfer. The trunk daemon then removes the firewall rules that were previously installed in the kernel and releases used data structures and other resources.

3.2.2. Mobile node daemon and application

The mobile node daemon is executed on every mobile node equipped with a GPRS and an IEEE 802.11(b) WLAN interfaces. In the proposed scheme, at any time a mobile node daemon may perform the function of a DN or a RN, but not both. When a mobile node acts as a DN to download its own file, it will not agree to act as a RN to help another mobile node download its file. However, even if a mobile node is acting as a RN to help another mobile node, if it wants to download its own file, it can immediately become a DN for its own download and stop forwarding packets for the DN that it is helping (called previous DN below). If the new DN still holds some UDP packets for the previous DN, it sends all of them out together without waiting for any acknowledgment packet and performing any retransmission. If there is any packet loss among these packets, such loss will be recovered by the original TCP retransmission mechanism triggered on the TCP connection between the PS and the previous DN.

When a RN becomes a DN, it also stops acknowledging those UDP packets bound to the previous DN through its GPRS channel. Due to the GPRS channel monitoring design presented before, the trunk daemon will soon detect that this GPRS channel is no longer useful for the previous DN. As such, it will soon stop using this GPRS channel for the previous DN. Until the PS informs the trunk daemon about a new file-fetching request issued by the new DN, the trunk daemon stops sending probe packets into the new DN's GPRS channel.

Mobile node daemons periodically exchange hello packets so that each mobile node knows its neighboring nodes. This information helps a mobile node daemon choose its RN candidates when it becomes a DN.

On a mobile node, many real-world applications such as the Mozilla web browser can be used to download files. They function normally and readily take advantage of the proposed scheme without any modification. When the web browser initiates a web request, the mobile node becomes a DN and the request is sent over the DN's GPRS channel to the PS. Later on, the packets carrying the content of the requested file are sent back to the DN through its

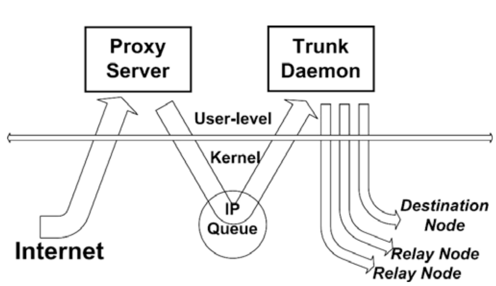

Fig. 2. The trunk daemon sets firewall rules to intercept and dispatch packets to the DN and its helping RNs.

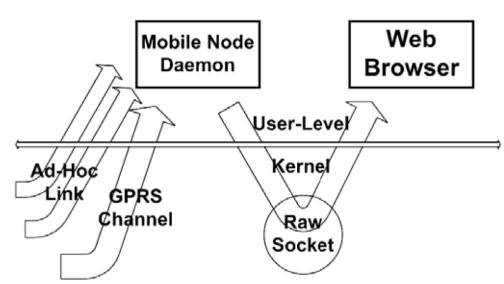

Fig. 3. The mobile node daemon reorders out-of-order packets and then sends them to the web browser through a raw socket.

own GPRS channel and some ad hoc links in the VANET. Fig. 3 shows the design.

If a mobile node daemon plays the role of a RN, it just needs to relay received packets sent from the trunk daemon to the DN. When relaying a packet to the DN, it sends out the packet through a UDP socket. The destination IP address of this packet is set to be the IP address of the DN on the VANET. As such, the packet is encapsulated as a UDP packet and travels on the VANET until it reaches the DN. The mobile node daemon implements an acknowledgment scheme running between itself and the DN to detect packet losses that may occur on the VANET. If any packet is lost on the VANET, it retransmits the packet until the DN eventually receives the packet. This network-layer retransmission design is more effective than the end-to-end transport-layer (TCP) retransmission design. This is because the round-trip time (RTT) between a RN and a DN on the VANET is only a few milliseconds for a 1500-byte packet and thus a lost packet can be retransmitted quickly. In contrast, due to the low GPRS channel bandwidth, the RTT of a GPRS channel is about 3 s for a 1500-byte packet. Since the end-to-end delay between the PS and the DN must be greater than the RTT of a GPRS channel, TCP will retransmit packets too slowly under this end-to-end delay and under-utilize the channel bandwidth.

If a mobile node daemon plays the role of a DN, it may need to reorder received packets, which may be out-of-order because they traverse on different GPRS channels and on different VANET paths. The mobile node daemon reorders these packets into the correct order and then writes them into the kernel through a raw socket. From the viewpoint of the kernel, these packets appear to arrive from a normal network interface such as an Ethernet interface. As such, these packets will pass through the TCP/IP protocol stack in the kernel and be enqueued into a TCP socket used by the web browser. Finally, the web browser reads the TCP socket to get the content carried by these packets. It may display the content on the screen or save the content into the disk.

4. Protocol designs

The protocols used in the proposed scheme include the initialization, data transfer, routing, and reset protocols. The first goal of the protocol design is that a real-world application, such as a web browser or a FTP client program, should be able to benefit from the scheme on a mobile node without any modification. The second goal is that the proposed scheme can be deployed for any real-world GPRS network without any support from the network operator, whether or not the GPRS network assigns a public or private IP address to a GPRS network interface. Our design can achieve both goals at the same time. In the rest of this section, we present the details of these protocol designs.

4.1. Initialization protocol

The initialization protocol coordinates different components of the system so that they work together to download a requested file over multiple channels. It consists of three phases. In the following, we use an example to illustrate the protocol. In Fig. 4, a web browser on the DN sends a request to the PS (apache server) (step 1). The PS then sends a notification packet to the mobile node daemon running on the DN (it is simply called the "DN" for brevity in the following description when there is no ambiguity) to inform it that a web browser executed on the same node has initiated a web transfer request (step 2). In the mean time, the PS sends the request to the specified WS on behalf of the DN to retrieve the requested file. Because each mobile node daemon maintains the information about its neighboring nodes, the DN sends a list of

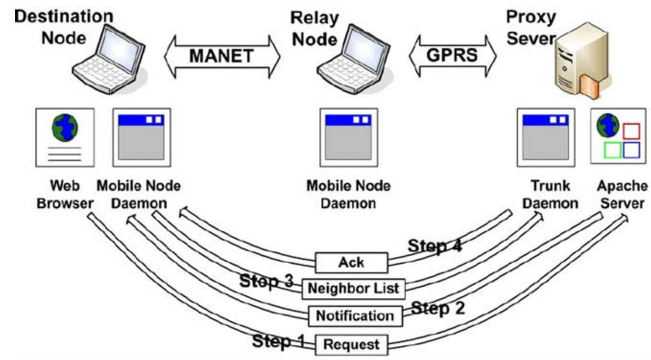

Fig. 4. The first phase of the initialization protocol.

its neighboring nodes to the trunk daemon (step 3). The nodes on the list are RN candidates for this DN and the trunk daemon may select some of them as the helping RNs for this DN based on their willingness and channel quality. The trunk daemon then sends an acknowledgment packet back to the DN (step 4).

Fig. 5 shows the second phase of the initialization protocol. After receiving the acknowledgment packet from the trunk daemon, the DN sends a "NAT Mapping Installation" packet to the trunk daemon (step 5). If the GPRS network uses a network address translator (NAT) and assigns private IP addresses to its GPRS users, when this packet passes through the GPRS network's NAT on its way to the PS, the NAT will install a mapping entry recording the mapping between the packet's private IP address and port number used inside the GPRS network and the public IP address and port number assigned to it. With this mapping entry, later on when the PS sends data packets to the DN from the Internet, these data packets can pass through the NAT and reach the DN successfully.

The DN also sends a relay-request packet to each of its neighboring nodes to ask it to help download data (step 6). When a mobile node receives a relay-request packet, it decides whether to help the requesting DN or not. If it is willing to help, it sends a "Willingness Indication" packet to the trunk daemon to express its willingness (step 7). The trunk daemon picks a mobile node as one helping RN for the DN only if it explicitly indicates its willingness to help. The "Willingness Indication" packet also serves the same function as the "NAT Mapping Installation" packet. With the transmission of this packet, later on when the PS sends data packets to a RN, they can pass through the NAT and reach the RN successfully.

This protocol design works successfully on a GPRS network that uses a NAT and assigns private IP addresses to its users (e.g., the FarEastone Telecom's GPRS network). It also works successfully for a GPRS network that does not use a NAT and assigns public IP addresses to its users (e.g., the ChungHwa Telcom's GPRS network).

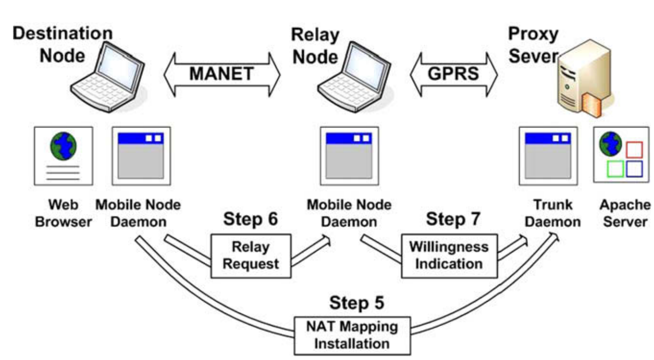

Fig. 5. The second phase of the initialization protocol.

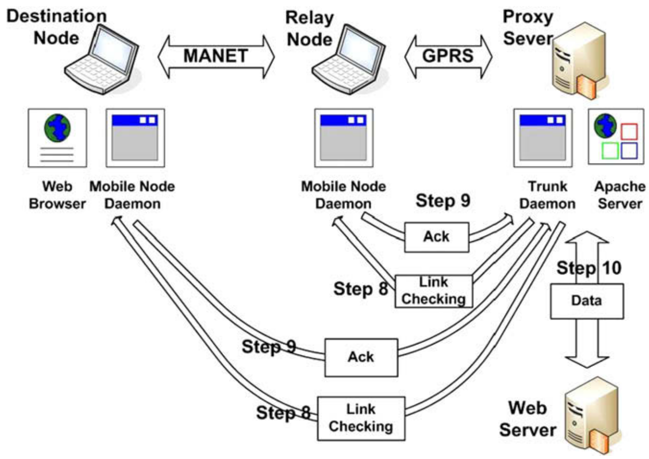

Fig. 6. The third phase of the initialization protocol.

Fig. 6 shows the third phase of this protocol. When the trunk daemon receives a "Willingness Indication" packet or a "NAT Mapping Installation" packet from a mobile node daemon, it sends a "Link Checking" packet to that mobile node daemon (step 8). The mobile node daemon replies an acknowledgment packet to the trunk daemon (step 9). When receiving the acknowledgment packet, the trunk daemon knows that the round-trip paths between the PS and the mobile node daemon are working well. If the delay between sending the "Link Checking" packet and receiving its acknowledgment packet is relatively small, which indicates good channel quality, the trunk daemon can choose the mobile node as a helping RN for this DN.

When these control packets are exchanged, the requested file can be transferred from the WS to the PS (step 10). In the mean time, the trunk daemon sets several firewall rules in the kernel to capture packets that will be sent from the PS to the DN. When such packets are captured, the trunk daemon sends them to the DN and its helping RNs in parallel. In case the packets sent from the WS to the PS arrives at the PS before these firewall rules are set, these early-arriving packets will be directly sent to the DN through the DN's own GPRS channel without using the proposed scheme. In this situation, the performance is not optimized; however, the function of the proposed scheme still works correctly.

4.2. Data transfer protocol

The data transfer protocol deals with packet delivery and retransmission. In Fig. 7, the trunk daemon sends data packets to the DN and its helping RNs through their GPRS channels (step 11-1). When the RN receives a packet, it forwards the packet to the DN through the VANET (step 11-2). The DN then sends an acknowledgment packet to the RN (step 12-2). The RN then sends an acknowledgment packet to the trunk daemon to inform it that the DN has successfully received the packet (step 12-3). When the DN receives a packet that is sent directly from the trunk daemon to itself, it sends an acknowledgment packet directly to the trunk daemon to acknowledge the receipt of the packet (step 12-1).

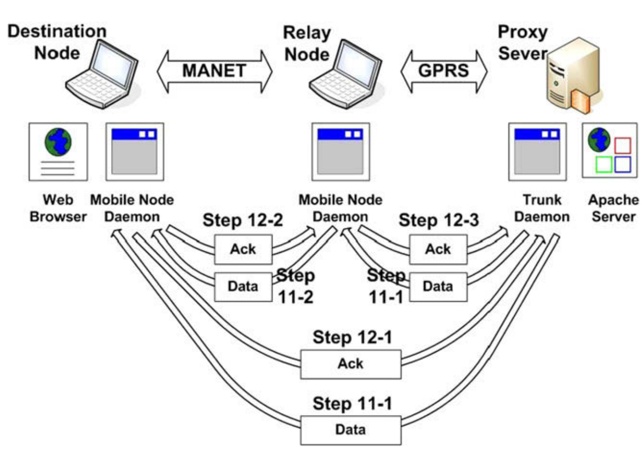

Fig. 7. The data transfer protocol.

The trunk daemon uses UDP to send captured packets to the DN and its helping RNs over their GPRS channels. RNs also use UDP to forward such packets over the VANET to the DN. Since UDP does not provide a reliable service, the trunk daemon employs an acknowledgment and retransmission scheme between itself and a mobile node (can be a RN or the DN) to quickly detect and retransmit a lost UDP packet over the GPRS channels. A RN also employs the same acknowledgment and retransmission scheme between itself and the DN over the VANET. The employed acknowledgment and retransmission scheme is similar to the famous TCP fast retransmission mechanism. When three duplicate acknowledgment packets are received at the sending node (can be the trunk daemon or a RN), the sending node quickly retransmits the lost packet pointed by these acknowledgment packets. If less than three acknowledgment packets can be received, the packet retransmission will be triggered by retransmission timer timeout.

Although one can rely on TCP to provide end-to-end lost packet detection and retransmission, TCP performs badly over varying-quality GPRS channels and highly-dynamic VANET. It is well known that TCP cannot detect packet losses caused by congestion, bit errors, or network topology changes and thus unnecessarily reduces its sending rate on every packet loss assuming that every packet loss is caused by congestion. Therefore, we employ an agile UDP-based fast retransmission mechanism on the varying-quality GPRS channels and highly-dynamic VANET to make these links reliable to TCP.

4.3. RN candidates finding protocol

As presented before, a mobile node daemon periodically exchanges hello packets with its neighboring nodes to know their existence. This information collection design adopts the design of a distance-vector routing protocol (e.g., RIP). As such, a mobile node daemon can collect the information of other mobile node daemons several hops away. Right now, the default value for this maximum hop count parameter is set to 3. It can be varied flexibly according to different policies or environments.

4.4. Routing protocol

Any routing protocol suitable for VANET can be independently used to set up routing paths in the VANET of the proposed scheme. In Section 6.2, we present the field trial results of the proposed scheme. In the field trials where mobile nodes are fast-moving vehicles, we used a flooding-based intelligent routing protocol [12] to increase the packet delivery ratio over the dynamic VANET. This routing protocol outperforms AODV greatly in a highly-dynamic VANET.

4.5. Reset protocol

The reset protocol is used to inform the DN and its helping RNs that the file download is finished. Fig. 8 shows this protocol. When the PS finishes sending the file content to the DN, it sends an "Ending" UDP packet to the trunk daemon (step 13) to notify it of this event. The trunk daemon in turns sends an "Ending" packet to each

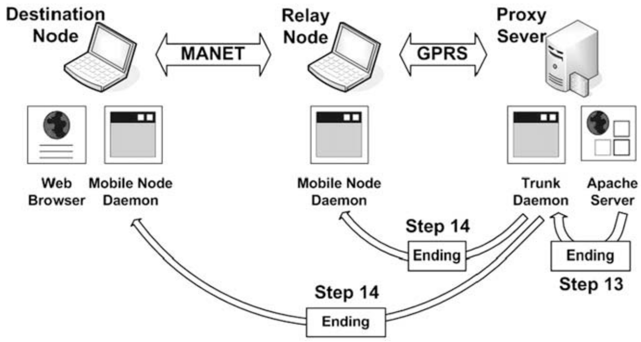

Fig. 8. The reset protocol used to release used resources.

mobile node daemon involved in this file download (step 14). It then removes the firewall rules that were previously installed in the kernel and releases all the data structures and other resources allocated for this file download. When a mobile node daemon receives an "Ending" packet, it releases the used data structures and other resources. It also resets its state to prepare for another file download. In case an "Ending" packet is lost, a mobile node daemon will automatically release the used data structures and other resource after a certain period of channel idle time.

5. Experimental settings and calibration tests

5.1. Experimental settings

Four IBM A31 notebook computers with 1.8 GHz CPU and 512 MB memory were used in the indoor, outdoor, and mobile experiments. One was used as the DN while the others were used as the DN's helping RNs. They are shown in Fig. 9. Each of these machines was equipped with an ASUS WL-14 IEEE 802.11(b) WLAN interface card and a Nokia D211 GPRS interface card, which is shown in Fig. 10. Because the two PCMCIA slots available on an IBM A31 notebook computer are physically too close, there is no way to use one PCMCIA IEEE 802.11(b) and one PCMCIA GPRS cards at the same time. To overcome this problem, the ASUS WL-14 IEEE 802.11(b) interface card was chosen because it uses the USB interface rather than the PCMCIA interface to connect to the notebook computer. Each of these machines ran the Red-Hat Linux operating system with the 2.2.6 kernel.

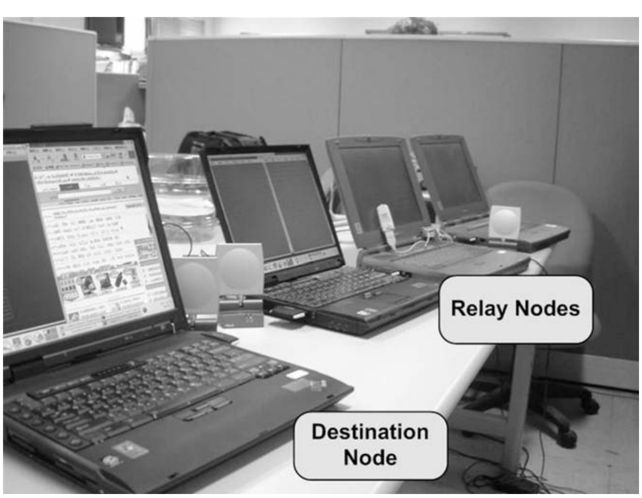

Fig. 9. Four notebook computers are used in experiments. One of them is used as the DN while the others are used as the DN's helping RNs.

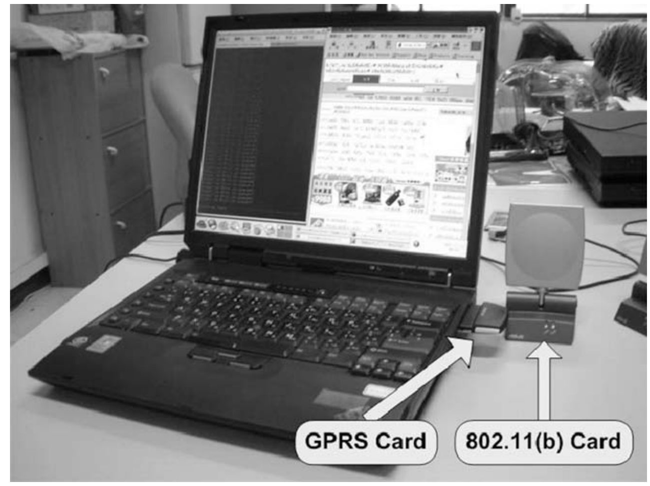

Fig. 10. Each notebook computer is equipped with an IEEE 802.11(b) WLAN interface card and a GPRS interface card.

One desktop host was used as the PS. It ran the Red-Hat Linux operating system with the 2.2.6 kernel. This host was located in our laboratory and connected to the Internet. A modified apache web server was run on this desktop host to provide proxy services for the DN.

The GPRS network used in the experiments was operated by ChungHwa Telecom Inc. When a GPRS network interface is attached to the GPRS network, ChungHwa Telecom Inc. automatically assigns a public IP address to it. As such, an Internet host (e.g., the PS) can actively send packets to an attached GPRS user.

The proposed scheme was also evaluated on a different GPRS network, which was operated by FarEastone Telecom Inc. Because the experimental results are similar to those on the ChungHwa Telecom GPRS network, they are not presented in this paper. On the FarEastone GPRS network, when a machine with a GPRS interface is attached to the network, it is assigned a private IP address. Such a machine can only actively exchange packets with a host on the Internet but a host on the Internet cannot actively exchange packets with it. This restriction, however, does not cause any problem for the proposed scheme. This is because before the PS sends packets to the DN, the DN has sent a "NAT Mapping Installation" packet to the PS.

5.2. Properties of the used GPRS network

To know the properties of the used GPRS network, we monitored the transfer throughput achievable over one GPRS channel when one, two, three, and four GPRS channels were simultaneously active to transfer data. The tested channels were set up between the PS and the DN, and between the PS and the three RNs. In these tests, the PS machine pumped UDP packets into a GPRS channel as fast as the channel could sustain, and the received throughput averaged over the past one second was reported every one second at the receiving machine (either a DN or a RN machine). In these tests, we used UDP traffic sources rather than TCP traffic sources. This is because the UDP protocol does not employ any congestion control while the TCP protocol does, which may thus reduce its sending rate when encountering packet losses. Since we are concerned about the maximum throughput provided by a GPRS channel rather than the maximum performance of a congestion control protocol on a GPRS channel, we used UDP traffic sources in the tests.

When UDP packets are continuously sent over a GPRS channel, we say that the channel is active. Fig. 11 shows the running

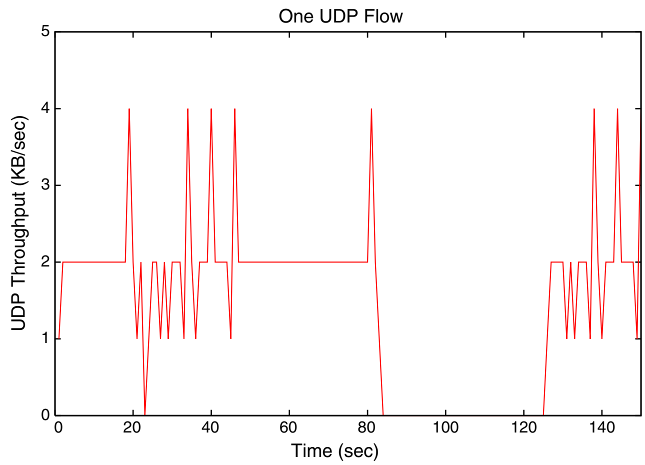

Fig. 11. The running throughput achieved on a single GPRS channel.

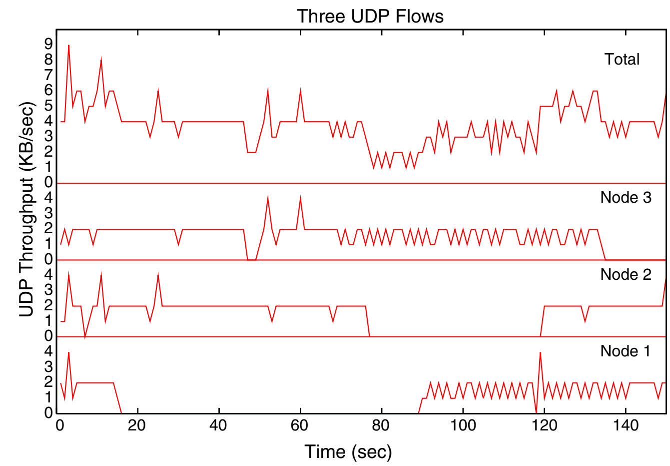

Fig. 13. The running throughputs achieved on three GPRS channels and their sum.

throughput achieved on a single channel when only one channel was active. Fig. 12 shows the running throughputs achieved on two channels and their sum when two channels were active at the same time. Fig. 13 shows the running throughputs achieved on three channels and their sum when three channels were active at the same time. Finally, Fig. 14 shows the running throughputs achieved on four channels and their sum when four channels were active at the same time.

Some observations can be made from these figures. First, the average throughput that can be achieved over one GPRS channel is only about 3 KB/s. Second, the quality of a GPRS channel is unstable. The received throughput on a GPRS channel often drops to zero, stays at zero for a while, and then rises again. Third, when multiple GPRS channels are actively used at the same time, it is common that at least one channel's throughput is very poor. (We conducted the same tests on both the ChungHwa and FarEastone GPRS networks and observed the same phenomenon.) These observations indicate that striping traffic over *N* GPRS channels to achieve *N* throughput speedup is almost impossible on current GPRS networks. We suspect that this phenomenon may be due to the fact that cellular network operators do not allocate enough time slots for GPRS traffic. In Section 7, we will discuss this issue in more details.

Since on the tested GPRS networks the best aggregate throughput achieved over N GPRS channels cannot be as high as *N* times of

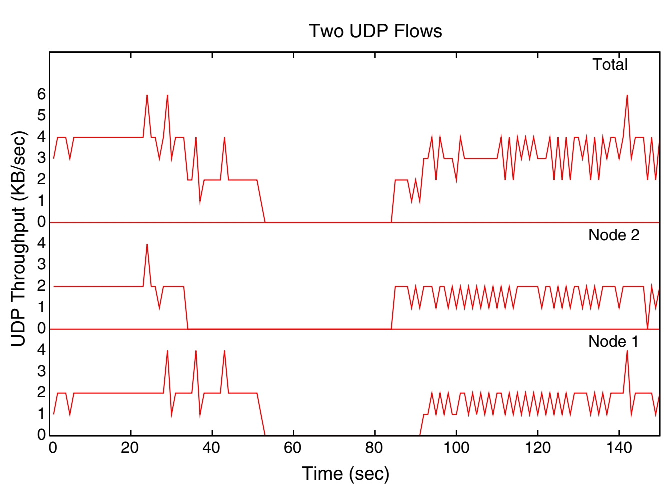

Fig. 12. The running throughputs achieved on two GPRS channels and their sum.

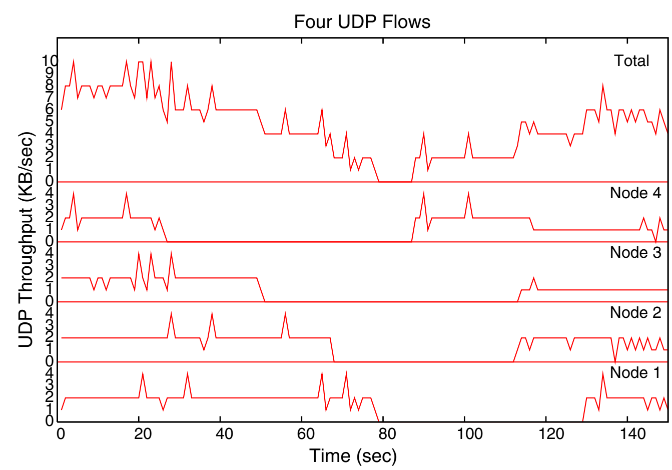

Fig. 14. The running throughputs achieved on four GPRS channels and their sum.

the throughput achieved over a single GPRS channel, when calculating the throughput speedup provided by the proposed scheme over *N* channels, we divide the achieved throughput of the proposed scheme by the single-channel throughput measured when *N* channels are simultaneously active. That is, let BAUDPTN denotes the best aggregate UDP throughput over *N* channels and TOSN denotes the achieved throughput of our scheme over *N* channels, then Speedup is calculated based on the following equation:

$$Speedup = \frac{TOSN}{\left(\frac{BAUDPTN}{N}\right)}. \quad (1)$$

Since Eq. 1 can be rewritten as the following equation:

$$Speedup = \left(\frac{TOSN}{BAUDPTN}\right)N, \quad (2)$$

it is clear that a higher speedup value indicates that the performance of the proposed scheme approaches more closely to the best aggregate throughput over *N* channels on the tested GPRS networks. For example, if the calculated speedup is 3.9 over four channels, it means that the performance of the proposed scheme approaches 3.9/4 = 97.5% of the best aggregate throughput achievable over four channels on the tested GPRS networks.

Note that even on a single GPRS channel, the throughput of a TCP connection can already suffer greatly from packet losses, large round-trip delays of the channel (3–4 s on a GPRS channel), and long channel blockage periods. When striping the traffic of a TCP connection over multiple GPRS channels, the proposed scheme has to overcome excessive packet reordering caused by the varying delay jitters of different GPRS channels. If this problem is not properly solved, excessive packet reordering can occur on the DN when packets arrive from multiple channels. This will trigger the TCP protocol stack on the DN to send back excessive duplicate TCP acknowledgment packets to the TCP protocol stack on the PS, causing the sending rate of the PS to be unnecessarily and severely reduced. Achieving a high aggregate TCP throughput over N unstable GPRS channels is a challenging task, especially when the quality of GPRS channels fluctuates greatly in mobile environments.

5.3. Calibration tests of the used GPRS network

The purpose of doing these tests is to measure the transfer throughput achievable over one GPRS channel when four GPRS channels are simultaneously active to transfer UDP packets (i.e., $\frac{BAUDPTN}{N}$). In these tests, the achieved throughput of a channel is defined to be the transferred file size (which is 500 KB) divided by the time required to finish the file transfer over the channel.

In the first test suite, only one GPRS channel was active and Table 1 shows the download throughputs of 10 tests. The average throughput of the 10 tests is 3.97 KB/s. In the second test suite, four GPRS channels were simultaneously active and Table 2 shows the aggregate download throughputs over the four channels of ten tests. The average aggregate throughput of the ten tests is 10.32 KB/s. This result shows that when four GPRS channels are simultaneously active, the average throughput achievable over one GPRS channel is only 2.58 (10.32/4) KB/s, which is lower than that in the first test suite (3.97), where only one channel is active. This phenomenon is the same as that observed in Section 5.2. As already explained in Section 5.2, we will use this number (2.58 KB/s) to calculate the throughput speedup provided by the proposed scheme.

6. Experimental results

6.1. Indoor experimental results

In the indoor experiments, we measured the web (TCP) download throughput of the proposed scheme when one, two, three, and four GPRS channels were used. In each of the experiment suite, the file transfer size was used as the system variable. For each file size, the experiment was repeated 10 times and the average and standard deviation of the 10 throughputs were reported. In the experiments, the four notebook computers (one DN and three RNs), the PS, and the web server hosting the web files all resided in our laboratory.

Table 1
The UDP download throughput over one GPRS channel.

Exp	Throughput (KB/s)
1	3.92
2	3.87
3	4.12
4	3.95
5	4.09
6	3.85
7	3.89
8	4.08
9	4.15
10	3.83
Avg	3.97

Table 2
The aggregate UDP download throughput over four GPRS channels

Exp	Throughput (KB/s)	Thput/channel (KB/s)
1	9.71	2.43
2	10.15	2.54
3	9.44	2.36
4	11.96	2.99
5	9.85	2.46
6	10.25	2.56
7	9.18	2.30
8	9.89	2.47
9	12.64	3.16
10	10.10	2.53
Avg	10.32	2.58

In the first experiment suite, no RN provided additional GPRS channel bandwidth to help the DN download its requested file. Thus, the packets carrying the file's content were transmitted on the DN's own GPRS channel. Fig. 15 shows the download throughput with different file sizes. We can see that the average throughput is about 3.3 KB/s when the file size is greater than 50 KB. For each throughput data point, the point above it is the average plus the standard deviation while the point below it is the average minus the standard deviation.

In the second experiment suite, one RN was employed and its channel and the DN's channel were used together to download the file in parallel. Fig. 16 shows the throughput of the proposed scheme using two channels. We see that the throughput is about 6 KB/s when the file size exceeds 90 KB.

In the third experiment suite, two RNs were employed and in total three GPRS channels were used to download the file in parallel. Fig. 17 shows that throughput of the proposed scheme using three channels. We see that the throughput is about 8 KB/s when the file size exceeds 550 KB.

Finally in the fourth experiment suite, three RNs were employed and in total four GPRS channels were used to download the requested file in parallel. Fig. 18 shows the throughput of the proposed scheme using four channels. We see that the throughput is about 9 KB/s when the file size exceeds 500 KB. This corresponds to a speedup of 3.49 (9/2.58).

The above figures show that the average file download throughput is low when the file size is small. This phenomenon can be explained as follows. After a TCP connection is set up, it immediately enters the TCP slow-start congestion control phase. In this phase,

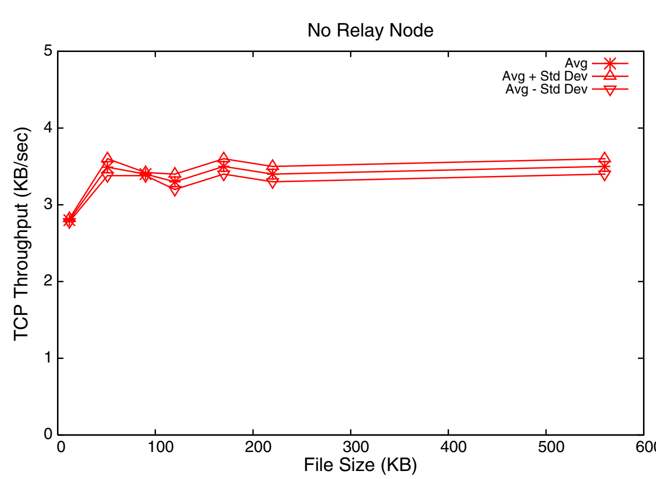

Fig. 15. The file download throughput with different sizes (through only one GPRS channel).

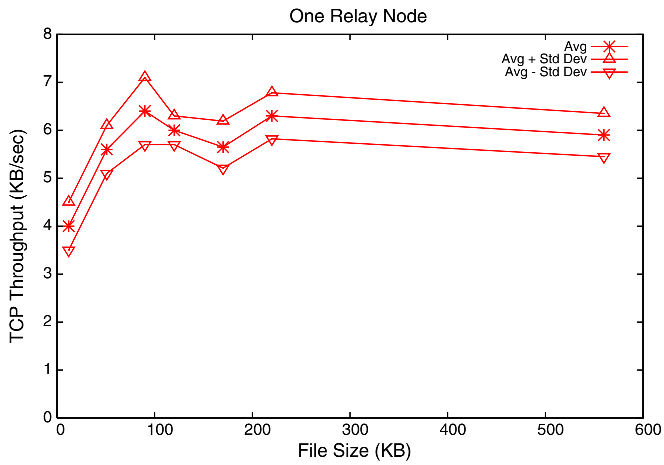

Fig. 16. The file download throughput with different sizes (through two GPRS channels).

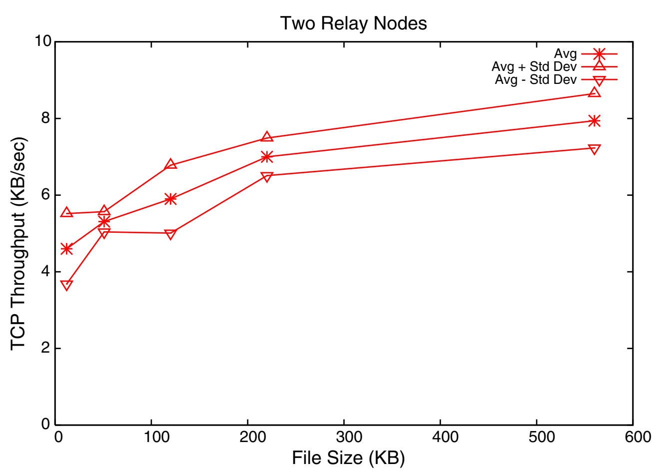

Fig. 17. The file download throughput with different sizes (through three GPRS channels).

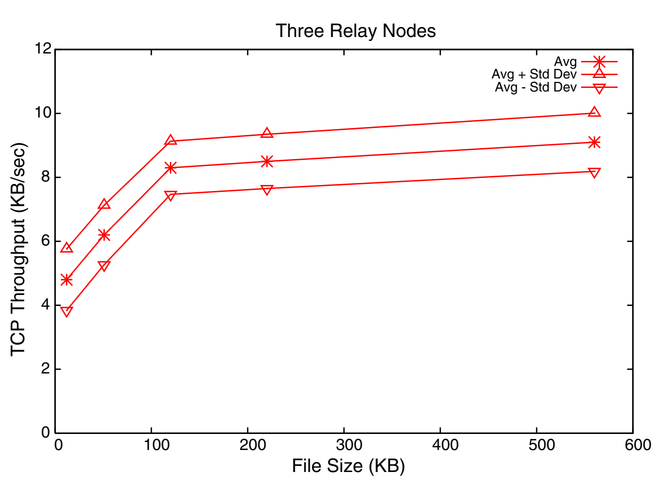

Fig. 18. The file download throughput with different sizes (through four GPRS channels).

the TCP sender exponentially increases its sending rate (its congestion window size – the number of unacknowledged packets that

Fig. 19. The external antenna, cable, and wireless NICs used in the field trials.

can be sent out per RTT) in each subsequent RTT, where RTT is the round trip time between the TCP sender and receiver. For example, a TCP sender sends out 1, 2, 4, 8, and 16 packets in the 1st, 2nd, 3rd, 4th, and 5th RTT, respectively. If the size of the requested file is small and thus only few packets are needed to carry its content, finishing the file transfer will use only the first few RTTs. Since a TCP sender's sending rate is its congestion window size divided by its RTT and the GPRS channel's RTT is very large (3–4 s for a standard-sized 1500-byte packet), the TCP sender's sending rates in the first few RTTs are the lowest. As such, TCP throughputs on GPRS channels are very low for small file transfers regardless whether the proposed scheme is used or not.

The results shown in these figures and the above explanation suggest that the proposed scheme is more suitable for downloading large files than downloading small files. Actually, if the file size is small, its transfer time over a single channel is already small and there is no need to use the proposed scheme to further reduce the small transfer time.

6.2. Outdoor and mobile experimental results

To evaluate the effects of outdoor environment and mobility on the performance of the proposed scheme, we repeated the above indoor experiments using four cars moving around our campus. Each of the four experimental notebook computers was moved into each of these four cars, respectively. The IEEE 802.11(b) WLAN interface card used by each notebook computer was connected to an external antenna (Fig. 19), which was mounted on the top of the car to get better signal quality (Fig. 20). Along the campus roads, there were roadside trees and buildings and we observed that these objects had spatial–temporal effects on the quality of the GPRS signal. Because the results of indoor experiments show that the proposed scheme is more suitable for downloading large files (Figs. 15–18), in all outdoor and mobility experiments we used 500 KB as the download file size.

6.2.1. Fixed and single-hop case

In the fixed and single-hop experiments, the four cars were parked at a parking lot. They were close to each other so that they can communicate directly with each other in VANET. Three relay cars (nodes) were employed to help download files for the destination car (node). Table 3 shows the download throughputs of the destination car in 10 different experiments. The average throughput speedup over four channels in fixed and single-hop outdoor experiments is 3.92 (10.12/2.58).

Fig. 20. Four cars were employed in outdoor and mobility experiments.

Table 3
The TCP download throughputs of 10 fixed and single-hop outdoor experiments.

Exp	Throughput (KB/s)	Speedup
1	9.09	
2	8.20	
3	11.9	
4	12.82	
5	8.62	
6	11.36	
7	8.93	
8	12.20	
9	9.43	
10	8.62	
Avg	10.12	3.92

6.2.2. Fixed and multi-hop case

In the fixed and multi-hop experiments, the four cars were parked along a straight road. The distance between each pair of adjacent cars was extended so that each car could only communicate with its two neighbors. As a result, the formed VANET is a 3-hop chain. The head car of the 3-hop chain was used as the destination car while the other cars were all used as relay cars. Table 4 shows the download throughputs of the destination car in 10 different experiments. The average throughput speedup over four channels in fixed and multi-hop outdoor experiments is 3.28 (8.45/2.58).

6.2.3. Mobile and multi-hop case

In the mobile and multi-hop experiments, the four cars were grouped into a motorcade and they moved around the campus. The head car of the motorcade kept its moving speed between 20 and 30 km/h and each of the other cars followed its preceding car to maintain a safe distance. Since there were bumpers on the roads, which made maintaining the VANET topology difficult, the packet forwarding path between the destination car and a relay car might be either single-hop or multi-hop during the period of an experiment. Table 5 shows the download throughputs of the destination car in 10 different experiments. The average throughput speedup over four channels in mobile and multi-hop outdoor experiments is 2.93 (7.55/2.58).

7. Discussions

7.1. Potential performance

The performance of the proposed scheme in the real world is greatly influenced by current operational GPRS base stations. According to our calibration test results, the throughput provided by a GPRS channel is both low and unstable on the current GPRS network.

Conceivably, if GPRS network operators allocate few time slots to GPRS traffic so that only three full-rate GPRS users can be active at the same time, using more than three channels at the same time will not improve the performance of the proposed scheme. Currently, the GPRS time slot allocation policy used in operational GPRS base stations is a black box and cannot be varied for our research purposes. Based on the experimental results, we suspect that the used GPRS network (ChungHwa Telecom Inc.) allocates few time slots to GPRS traffic so that only three or four full-rate GPRS users can be supported simultaneously. We also evaluated the scheme on another operational GPRS network (FarEastone Telecom Inc.) and observed the same phenomenon.

If this conjecture is true, this few GPRS time slot allocation policy may be attributed to the current small GPRS market. Although GPRS has been introduced to the market since several years ago, it has few subscribers at present due to its expensive but low-throughput services. Since GSM and GPRS traffic compete for a limited number of time slots provided by a base station, to maximize the profit, GSM/GPRS cellular network operators are likely to allocate much more time slots to GSM traffic than to GPRS traffic. In addition to the biased static allocation policy, operators may also give a high priority to GSM voice traffic so that it can preempt and use the time slots originally allocated to low-priority GPRS data traffic.

To discover the real performance potential of the proposed scheme rather than its performance over a black-box operational GPRS network, simulations can be used. Thus, we plan to simulate and evaluate the performance of the proposed scheme in NCTUns

Table 4
The TCP download throughputs of 10 fixed and multi-hop outdoor experiments.

Exp	Throughput (KB/s)	Speedup
1	8.87	
2	8.76	
3	9.11	
4	8.60	
5	8.12	
6	8.10	
7	8.23	
8	8.15	
9	8.25	
10	8.34	
Avg	8.45	3.28

Table 5
The TCP download throughput of 10 mobile and multi-hop outdoor experiments.

Exp	Throughput (KB/s)	Speedup
1	7.69	
2	6.99	
3	7.75	
4	7.46	
5	8.55	
6	7.25	
7	7.75	
8	8.00	
9	6.45	
10	7.65	
Avg	7.55	2.93

[13]. In NCTUns, one can flexibly change the time slot allocation policy and scheduling algorithm used for GSM and GPRS traffic. If simulation results show that the proposed scheme can achieve good throughput speedup given enough GPRS time slots and there is a market for this GPRS + WLAN trunking service, GSM/GPRS network operators may be willing to allocate more time slots to GPRS traffic.

7.2. Market potential

Currently, devices with both GSM/GPRS and WLAN interfaces have been introduced to the market (e.g., Nokia 6136 and TOBE OBU manufactured by Yulon Motor Corporation). Such devices have the required network interfaces to use the proposed scheme to speed up their file transfer throughputs over GPRS networks. Using the proposed scheme, a typical web page containing many images or a large file can be downloaded quickly. Video conferencing is also feasible due to increased throughput. Because the proposed scheme does not need any support from a GPRS network operator, the GPRS + WLAN trunking service provided by it can be immediately deployed for any operational GPRS network. Currently, 3G cellular networks and services are still very expensive and not widely available. The GPRS + WLAN trunking service can use the current 2.5G networks to support high-bandwidth applications for such devices. The cost of the software-based proposed scheme is much lower than the cost incurred for replacing existing 2.5G networks with expensive 3G networks.

The proposed scheme can be applied to 3G networks as well, not just to 2.5G GPRS networks. Without any modification, this software-based scheme can be immediately used and deployed for any 3G network to further increase the download throughput over 3G or B3G networks.

7.3. Billing issues

A good billing policy is important to the success of the proposed scheme if it is to be deployed in the real world for commercial uses. In the proposed scheme, a mobile node (can be an OBU) that requests to download a file may borrow the GPRS channels of its neighbors. For a mobile node with limited power resource, receiving packets from a GPRS channel for another mobile node will consume its battery power and reduce its operation lifetime. However, this is not an issue in ITS applications as every vehicle has a long-lasting battery periodically charged by a generator. One left issue is that when a RN receives GPRS packets for the DN, the communication cost is charged to the RN rather than the DN. This billing problem can be solved by the following design with support from the network operator.

In this billing design, a GPRS network operator sets up the trunk daemon and web proxy server in the GPRS core network. The trunk daemon knows which users lend their GPRS channels to the requesting user for downloading data and how many packets are transferred over each of these channels. This borrow/lend usage statistics and accounting information can be transmitted to the GPRS network's billing service center to avoid the wrong-billing problem.

After eliminating the power and billing problems, one can further encourage GPRS users to help download other users' packets. To provide such incentives, a network operator may credit a certain number of points to a helping user's billing account for each relayed packet. Such points may be redeemed for gifts or be used to reduce a GPRS user's monthly payment. Although a network operator may lose some profit due to these credit points, the net profit of the network may increase due to increased GPRS usages (data services).

8. Future work

As mentioned in the previous section, we plan to simulate our scheme and study its performance over a simulated GPRS network. Using simulations, one can flexibly change the time slot allocation policy and scheduling algorithm used in base stations to observe their effects on the throughput speedup under various network conditions. In addition, a large network with many mobile nodes can be easily created and studied. This enables us to study the performance of the proposed scheme using more than four GPRS channels at a low cost. (The cost of doing GPRS experiments are very expensive and we have paid a lot of money for these experiments in the past one year.)

We will also study how to select appropriate RNs for a DN. The relationship between the achieved throughput speedup and the number of helping RNs may depend on several factors. Several important metrics need to be identified so that they can be used to select better RNs. Such metrics may include a RN's GPRS channel quality and the hop count between it and the DN. Moreover, it is desirable to study the effects of high node mobility on the performance of the proposed scheme. Such studies are difficult, expensive, and sometimes dangerous to be conducted in field trials.

9. Conclusions

In this paper, we design, implement, experiment, and evaluate the performances of a novel scheme, which enables a vehicle user (can be an OBU) to quickly download infotainment information from a GPRS network. This scheme integrates GPRS and VANET to achieve both of their respective advantages – large coverage of GPRS and high bandwidth of VANET.

Due to the novel design and implementation of the proposed scheme, it can be readily deployed for any real-world GPRS or 3G network without any support from the network operator. The performances of the proposed scheme have been evaluated on two real-world GPRS networks. Our indoor and outdoor experimental results show that the proposed scheme can achieve a speedup of 3.9 when using four GPRS channels. Currently, cellular devices and OBUs with GSM/GPRS and IEEE 802.11(b) interfaces are popular. The proposed scheme can immediately enable them to achieve a higher download throughput over any GPRS network.

Acknowledgements

We thank the anonymous reviewers for their valuable comments.

References

[1] Haiyun Luo, Ramachandran Ramjeey, Prasun Sinhaz, Li (Erran) Li, Songwu Lu, UCAN: a unified cellular and ad-hoc network architecture, ACM MOBICOM 2003, September 2003, San Diego, CA, USA.
[2] Ying-Dar Lin, Yu-Ching Hsu, Multihop cellular: a new architecture for wireless communications, IEEE INFOCOM 2000, March 2000, Tel Aviv, Israel.
[3] S. De, O. Tonguz, H. Wu, C. Qiao, Integrated cellular and ad hoc relay (iCAR) systems: pushing the performance limits of conventional wireless networks, in: 35th Annual Hawaii International Conference on System Sciences (HICSS'02), vol. 9, Januar 2002, Hawaii, USA.
[4] X. Wu, S. Chan, B. Mukherjee, MADF: a novel approach to add an ad-hoc overlay on a fixed cellular infrastructure, in: IEEE WCNC 2000, vol. 2, September 2000, Chicago, USA, pp. 549–554.
[5] Ansuya Negi, Suresh Singh, Improvement in reliability of coverage using 2-hop relaying in cellular networks, in: IEEE PIMRC 2003, vol. 3, September 2003, Beijing International, Convention Center, Beijing, China, pp. 2348–2352.
[6] L. Magalhaes, R. Kravets, End-to-end inverse multiplexing for mobile hosts, in: 19th Brazilian Symposium on Computer Networks (SBRC'01), 2001.
[7] Casey Carter, Robin Kravets, User devices cooperating to support resource aggregation, mobile computing systems and applications, in: Proceedings Fourth IEEE Workshop on 20–21 June 2002, pp. 59–69.

[8] P. Sharma, Sung-Ju Lee, J. Brassil, K.G. Shin, Handheld routers: intelligent bandwidth aggregation for mobile collaborative communities, Broadband Networks, 2004, BroadNets 2004, in: Proceedings, First International Conference on 25–29 Oct. 2004, pp. 537–547.
[9] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, S. Banerjee, MAR: a commuter router infrastructure for the mobile internet, ACM MOBISYS 2004, Boston, June 2004.
[10] Hung-Yun Hsieh, R. Sivakumar, pTCP: an end-to-end transport layer protocol for striped connections, in: Proceedings, 10th IEEE International Conference on Network Protocols, 12–15 Nov. 2002, pp. 24–33.
[11] S.Y. Wang, C.C. Hwang, C.L. Chou, On striping traffic over multiple IEEE 802.11(b) wireless channels, ACM Wireless Networks, 12:3, June 2006.
[12] S.Y. Wang, C.C. Lin, Y.W. Hwang, K.C. Tao, C.L. Chou, A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads, in: IEEE ITSC 2005, International Conference on Intelligent Transportation Systems, Sep. 13–16, 2005, Vienna, Austria.
[13] S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, C.C. Lin, The design and implementation of the NCTUns 1.0 network simulator, Computer Networks 42 (2) (2003) 175–197.