DragonNet: A Robust Mobile Internet Service System for Long Distance Trains

Fung Po Tso*, Lin Cui*, Lizhuo Zhang*, Weijia Jia*, Di Yao*, Jin Teng†, Dong Xuan†,
*Future Networking Center & Department of Computer Science, City University of Hong Kong, HKSAR
†The Ohio-State University, Columbus, OH 43210

***Abstract*—Wide range wireless networks often suffer from annoying service deterioration due to fickle wireless environment. This is especially the case with passengers on long distance train (LDT) to connect onto the Internet. To improve the service quality of wide range wireless networks, we present the DragonNet protocol with its implementation. The DragonNet system is a chained gateway which consists of a group of interlinked DragonNet routers working specifically for mobile chain transport systems. The protocol makes use of the spatial diversity of wireless signals that not all spots on a surface see the same level of radio frequency radiation. In the case of a LDT of around 500 meters, it is highly possible that some of the spanning routers still see sound signal quality, when the LDT is partially blocked from wireless Internet. DragonNet protocol fully utilizes this feature to amortize single point router failure over the whole router chain by intelligently rerouting traffics on failed ones to sound ones. We have implemented the DragonNet system and tested it in real railways over a period of three months. Our results have pinpointed two fundamental contributions of DragonNet protocol. First, DragonNet significantly reduces average temporary communication blackout (i.e. no Internet connection) to 1.5 seconds compared with 6 seconds that without DragonNet protocol. Second, DragonNet efficiently doubles the aggregate throughput on average.**

I. Introduction

A. Motivation

Witnessing the tremendous popularity of WiFi capable devices such as laptops, netbooks and Smartphones, rail operators are rushing to deploy high-speed wireless in a bid to lure potential passengers to travel with railway [4]. A study revealed that 72% of business travelers were more likely to use trains than cars or airplanes if Wi-Fi access was available on trains. Among them, 78% of these business travelers would actually use Wi-Fi access if it was made available on trains [1].

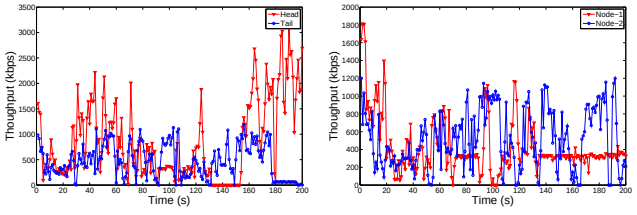
Existing infrastructures for providing Wi-Fi to Internet access are realized by relaying WLAN traffics via cellular network [3] [16], satellite [14], trackside WiMAX [11] or Leaky Coaxial Cable (LCX) [10] to the backbone network. However, there are still some barriers that hinder the use of these technologies. For example, satellite communications are not ideal for high-speed access to trains since satellite links have limited bandwidth and long round trip times (RTT). WiMAX access creates enormous financial burden for large scale installation of trackside WiMAX APs and equipment maintenance thereafter, so is LCX. On the contrary, the cellular-based infrastructure takes advantage of existing cellular architecture for reducing the deployment cost. However, handoffs between base stations or APs and drastic fading phenomena can easily cause severe deterioration in signal strength of a certain client device to an unacceptable level, resulting in degraded network performance.

Is there any method we can use to handle the downgraded cellular performance given the characteristics of LDT? First we may take notice of some everyday practice. Sometimes we may have very poor cellular signals in our office. So we get out of the office, walk some distance along the corridor, until we get perfect signal strength. The LDT is just like the corridor, except that the LDT is much longer. It is a corridor 500 meters long!

B. Contributions

In order to answer the question raised in the previous subsection, we have conducted experiments on LDTs to see whether there are much diversity in signal and networking performance at different spots. Fig. 1 shows the example of throughput diversity for two closely and distantly separated nodes from our field tests. Fig. 1a & Fig. 1b are instantaneous throughput variation for head-tail and adjacent nodes respectively. Despite the fact that all D-routers' cellular interfaces are usually attached to the same base station, diversity still exists among them because radio condition varies a lot with geographic location of which the larger in distance the greater difference they can have. As a consequence, channel quality seen by each cellular interface are very likely to be different from each other. It naturally follows that by exploiting the channel diversity along the LDT, those links suffering from deteriorated or failed link condition can be efficiently alleviated with help from neighboring nodes.

Along this line we present DragonNet, a gateway 500 meter which aims at handling single-point failure gracefully. DragonNet is formed by a DragonNet router (D-router, a



(a) Head & Tail Nodes (b) Adjacent Nodes

Fig. 1. Channel Diversity

gateway) chain running through the whole length of LDTs. The long D-router chain spanning the LDT makes single point failure manageable. With DragonNet, the failing D-routers can still connect with the Internet through D-routers located in areas of good signal quality. Therefore communication degradation or even failures at a certain section of LDT can be efficiently amortized. Based on thorough experimental tests, we show that DragonNet can derive significant benefits by exploiting the long stretching feature of LDT.

To sum up, our contributions are threefold:

- We have designed and implemented the DragonNet protocol to adaptively form a narrow and long stretching network and efficiently manages both internal and external (cellular) network link failures.
- We have built the D-router from scratch for running DragonNet protocol. Though it only accepts USB-interfaced UMTS/HSPA modems for cellular network access for now, it can be easily extended to future LTE access with only a small scale driver update.
- We have intensively tested the DragonNet prototype in two railways of Hong Kong. The performance results confirm that DragonNet can significantly amortize individual cellular link failure onto the whole chain and give higher aggregated system throughput.

Though DragonNet is proposed for LDT initially, it can be generically applied to other mobile chained transport systems, such as chained vehicles (buses, trucks) and chained ships, etc. However, some may argue that the research issues solved in this paper are not novel because the problem is merely a standard mesh routing problem. In fact, the routing problem DragonNet intends to solve is essentially different from that of mesh network. The mesh network focuses more on network formation, while the DragonNet concerns itself more about failure recovery. The difference in their starting points dictates that they will employ different approaches and implementations. Though mesh networks are capable of certain failure recovery, the mesh routing protocols deal with link failures among mesh routers and mesh clients such that new paths can be established between source and destination. In addition to handling such internal dynamics, the DragonNet primarily tackles with external cellular link failures so as to maximize the Internet connection time. With strong adaptivity to both internal and external failures, DragonNet can adaptively choose the optimal basestations to connect with and avoid pingpong phenomenon as a whole.

The remainder of this paper is organized as follows. We present the design rationale and constraints in Section II. In Section III, we give an introduction of DragonNet architecture. We then describe the DragonNet protocol and its operation in Section IV. Implementation details are presented in Section V. In Section VI, we extensively evaluate the performance of DragonNet and its supporting protocols. We present the results of a real DragonNet for different applications in various scenarios. Section VII presents a survey of related work. Section VIII concludes this paper.

II. Design Rationale

Having a long stretching gateway chain, the patterns of connection failures appearing during mobility become totally different from that of a single gateway. We thus classify the failures into two types: random failures and cascading failures. Random failure means a node is temporarily screened from cellular wireless signal in an unorganized manner. Cascading failure is defined as many D-routers close in location are temporarily out of coverage consecutively. Therefore, the primary challenge of DragonNet is to efficiently manage these failures so they will not affect ongoing traffics associated with failed cellular connections. On the other hand, the bandwidth may be largely underused if all users' traffics are herded to only a few D-routers in the DragonNet since all users' sessions contend for bandwidth on the target D-routers while many others are left idle. Thus another challenge of DragonNet is to evenly distribute users' sessions to all D-routers to improve bandwidth utilization for better system throughput. To overcome the challenges, we devised the DragonNet protocol with session-based rerouting algorithm to support DragonNet.

NAT transparency is another problem faced by DragonNet. In DragonNet, D-router creates a NAT for all associated wireless clients and HSPA network creates another NAT for all D-routers. To this extent, all wireless clients have to traverse at least two layers NATs in order to reach servers. Under rerouting, if failed D-routers divert their sessions to other D-routers, servers would simply reset all service sessions because they are perceived as new service requests. This example can be visualized in Fig. 2a where traffic leaving interface 10.13.1.9, for instance, is switched to interface 10.13.1.10 but the server treats NATed IP address of interface 10.13.1.10 as a new comer and thus creates new service sessions for it. To solve this problem, we introduce the proxy for guaranteeing the service transparency for end users and servers.

Beside technical challenges, we also list the practical constraints that DragonNet has to satisfy in order to be practically feasible for LDT. Those are:

- C1: Client devices are off-the-shelf and cannot be modified.
- C2: Any operation should be transparent to users so that they are not aware of having service interruptions due to temporary network blackout.
- C3: Algorithms & protocols should be feasible to be implemented into conventional WiFi APs.

Constraint C1 is critical to ensure user-friendliness and cost-efficiency. Users are generally reluctant to replace their mobile devices because they have already spent a lot of money to get one. Meanwhile, users are usually loath to install extra softwares and then go through a complicated configuration procedure for receiving Internet services. DragonNet makes sure users can gain access to mobile Internet as easily as conventional WiFi. Constraint C2 underscores the key idea of DragonNet, transparency. General speaking, it means users carry out their Internet related tasks or services as usual during

temporary services blackout of the associated APs, and DragonNet takes care of service recovery without user intervention. To be more specific, transparency guarantees that both client devices and servers need not modify their service protocols and are blind to DragonNet operations, but users should still get served as usual. Constraint C3 talks about feasibility of the solutions in real world networking elements. DragonNet's core protocol runs in a set of conventional APs, i.e. D-routers, which collaborate closely to facilitate DragonNet services.

III. DragonNet Architecture

In this section we present architecture and main components of the DragonNet system along with its supporting protocols. The DragonNet architecture is illustrated in Fig. 2. The DragonNet system consists primarily of a chain of the D-routers which bridges local WiFi LAN to outside cellular wireless network. The bus-like structured topology can be perceived like a live Dragon. It coordinates D-routers to perform traffic rerouting, if necessary, across the D-router chain (i.e. DragonNet) by exploiting available D-router's interface to amortize connection degradation or failure at certain point to provide a faster, smoother, and more reliable wireless service. To support such architecture, DragonNet protocol is realized in D-router as a core. Similar with conventional routing protocol, DragonNet protocol collects and exchanges local D-router's status with neighbors regularly. Routing decision is made upon receiving status metrics from neighboring D-router, particularly when failure report is received.

The D-router provides local (both wired and wireless) interfaces as well as a wide area wireless interface. Local interfaces provide wireless access to local mobile users using their WiFi capable devices such as Smartphones, Netbooks and other mobile devices. D-router is able to associate with a variety of wide area wireless technologies, e.g., UMTS, HSPA, WiMAX and future LTE, such that DragonNet can be attached to either one or mixture of these technologies. We primarily choose HSPA as it is currently the most dominant wide area broadband technology. We also refer wide area interfaces to HSPA modems thereafter in this paper.

In practice, IP addresses of wide area interface assigned by wide area operator in many cases are in a private range of addresses (e.g. 10.13.1.1). Requests from local users are directed to the D-router and further to the Internet. Based on a particular scheduling policy, D-router selects a given interface, for example, neighboring D-router or internet, for each packet or request. Once the wide area network interface is selected, client originated packets are source NATed, i.e. both IP address and port are mapped, using the IP address and port of the wide area interface. To the external world, the D-router appears as a NAT box. Therefore, remote server sees requests or packets from mobile user as from D-router. D-router deNATs packets returning from remote server and forwards them to the appropriate mobile user.

On the other end of DragonNet, there is a proxy serving the purpose of service transparency to servers. As packets originating from a certain client device usually reach servers through different paths, i.e., cellular interfaces, due to rerouting decisions made by individual D-router, this rerouting decision will force all packets, including TCP packets, to take another path to the destination servers while ongoing D-router is being blackout. To this extent, TCP session will be torn down and reset that eventually results in service interruptions to end users. To get rid of service suspensions, we introduce a proxy in between of DragonNet and servers. Under this circumstance, D-routers firstly redirect all packets to proxy which then re-encapsulates packets as if they are originated from proxy. Upon receiving response from the servers, the proxy incorporates packets with actual destination addresses and forwards them back to wireless clients.

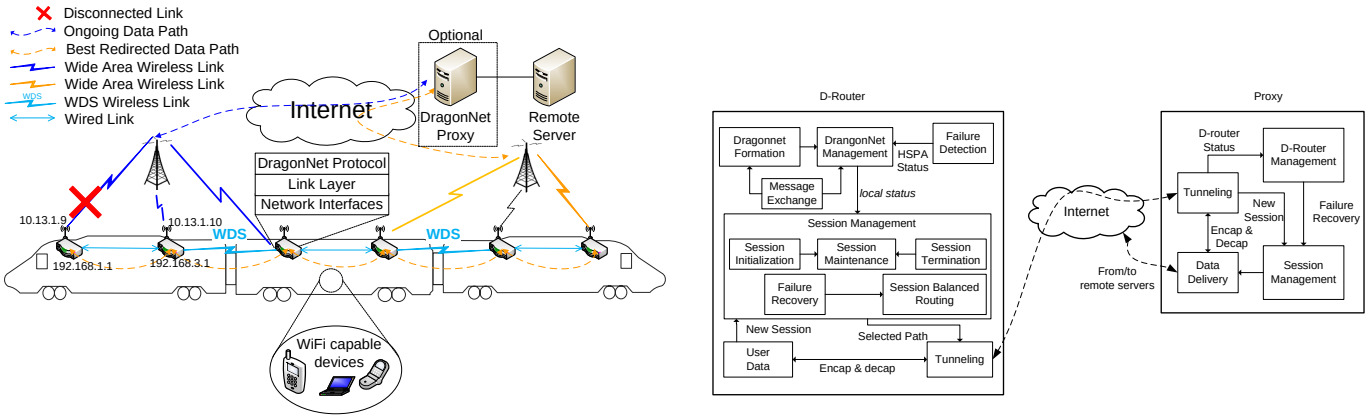
IV. DragonNet Protocol

As we discussed above, the DragonNet Protocol has two goals. First, it amortizes random and cascading D-router failure onto the whole D-router chain. Second, it aggregate bandwidth along the whole D-router chain. This section begins with detailed protocol operation description followed by rules and conditions governing the operations.

A. Operations

The individual D-router's hardware failure may divide the D-router chain into two or more smaller D-router chains. Such division will collapse the DragonNet if centralized management is adopted. But with decentralized management, smaller D-router chain will adaptively form smaller DragonNets to continue Internet services to end users. To support the decentralized operation, there are four types of control messages exchanging among D-routers & proxy in DragonNet protocol.

- HELLO Message: Carrying the *local status* information, this message is used for DragonNet formation during the initialization and connection maintenance phase. It is broadcast periodically but any change of the *local status* of the D-routers also triggers a immediate broadcast/unicast of this message. The Hello message is only exchanged among D-routers and restricted in one hop;
- NOTIFY Message: This message is used to notify proxy about the failure of D-routers' cellular interface. The failed D-router will broadcast the NOTIFY message to both previous and next neighbors. When this message is received, the neighbors would relay it to the proxy immediately if their cellular interfaces are available, otherwise they just forward it to the next D-routers. Upon receiving this NOTIFY, the proxy will update its local mapping table accordingly;
- CAscading FAilure (CAFA) Message: This message is sent out to notify the next D-Routers the presence of cascading failure when cascading failure is detected.
- Keep-ALIVE Message: This message is sent to proxy periodically when there is no traffic on the cellular connection. It acts as a heart-beat for D-router to notify the proxy of its aliveness and prevent timeout of possible network operator's NAT entries between D-routers and proxy, as discussed in Section II.



(a) Physical Components

(b) Logical Components

Fig. 2. DragonNet Architecture

On top of these messages, DragonNet protocol performs following operations to manage DragonNet and its failures:

- D-routers periodically broadcast HELLO message to discover their previous and next neighbors. HELLO message should only reach one hop away to avoid message flooding over the DragonNet. On the other hand, D-routers on the chain listen for HELLO messages from interconnecting interfaces. Once Hello messages are received, D-routers bind the arrival interfaces with previous and next D-routers using unique IDs extracted from received Hello messages.
- When a new session joins, the host D-router selects a route for it based on HELLO exchanged with neighbors. The decision can be either to tunnel this session to proxy directly or to forward this session to neighbor node. If latter option applies, the procedure should repeat until one of the D-routers tunnels the new session to the proxy. Once the route is identified, assuming all D-routers are healthy, all packets belong to this session should follow this route. The path selection processes will be discussed in greater details in IV-B.
- *Failure Detection & Prediction* produces HSPA status for each D-router. This is a parameter of *local status* and it is eventually carried by other message. D-router instantly broadcast HELLO and NOTIFY message upon detecting failures. Neighbors then relay the NOTIFY message proxy for updating reverse path. This is because when the HSPA interface of a D-router is down and sessions are rerouted to neighbors, the proxy will not know the happening of rerouting before data is being sent out from client through the affected sessions after rerouting. So the NOTIFY message is needed to let proxy know the changes of rerouting and make some coordination to the affected sessions in session table. Otherwise, packets from server to client on these sessions will be dropped. Neighboring D-routers examine received HELLO messages and determine which type of failure it should trigger. If the behavior of cascading failure is detected, a CAFA message should be sent out to notify the next D-router.
- All the user packets will be encapsulated at D-routers and forwarded to the HSPA interface through the path according to the session table. Therefore, on the proxy end, upon receiving a packet from a D-router, it decapsulates the packet. If the packet belongs to a new session, the proxy should assign a new free port to this session and then forward the packet to destination. Otherwise the proxy looks up and forwards the packet through in-use port number. For the reverse direction, the proxy encapsulates every packet and forwards it to specified D-router.
- When there is no data flowing through the HSPA interfaces, provided that they are still active and sound, the ALIVE message is sent from their D-router to the proxy periodically to keep the tunneling path alive.

Failure detection is achieved by looking at HSPA's physical and logical status. If the OS reports that the physical link status is set to *down*, this D-router will be omitted until its wide area interface is fixed or replaced. On the contrary, logical link quality measurement includes periodically collecting physical layer parameters RSSI (receiver signal strength) and EcIo (energy per chip over interference), both are good indicators, for estimating the dynamic wireless link. Besides, the predictability in LDT's route can greatly help in efficiently managing the wireless resources. The LDT route and the position of base stations are known to the DragonNet system. Even they are unknown, one or two trips are largely enough to feed the system with all necessary information. Given the predictable nature of routes, failure prediction is realized by examining current cell details and comparing them with historical details so as to get DragonNet ready for upcoming predictable failures such as handoffs and cell coverage black-holes.

B. Rules and Conditions for Path Selection

Assume there are n D-routers connected in a bus topology to form a DragonNet. For presentation simplicity, suppose there is no wide area interface failure initially. Let *loading factor* of *i-th* D-router be W_i, representing aggregated session weights

for all traffic sessions in this D-router. The bit rate of the session is quantized to (0,1] to become a weight, *w*, where 1 is referenced as the upper bound of certain HSPA interfaces of D-routers. The weights are recalculated at the same interval as broadcasting Hello message. Suppose the weight of *i-th* session is w_i , and there are n_i ongoing sessions on D-router i , so the *loading factor* W_i is defined as, $W_i = \sum_{j=1}^{n_i} w_j$. D-router *i* also keeps the aggregated *loading factors*, $WF_i = \sum_{j=1}^{i-1} W_j$ & $WB_i = \sum_{j=i+1}^{n} W_j$, and number of available D-routers, NF_i=(*i-1*) & NF_i=(*n-i*), in the forward and backward directions respectively.

Therefore, *unbalance factor*, the difference of maximum and minimum *loading factors* in DragonNet, of *i-th* D-router is expressed as, $d_i = \max(WF_i/NF_i, W_i, WB_i/NB_i) - \min(WF_i/NF_i, W_i, WB_i/NB_i)$. This factor is used for measuring the level of load difference for the DragonNet, the larger this value is means the higher level of load unbalance existing in the DragonNet.

During operation of the DragonNet protocol, the basic units, D-routers, have to exchange their *local status* regularly. The *local status* is defined as: (W_i, HSPA status, WF_i, NF_i, WB_i, NB_i).

Session Path Discovery: Based on above setup, DragonNet initializes every D-router accordingly at boot-up phase. But how does DragonNet accept a new session after then? For each new arrival session, a D-router has three choices: forward it to the previous or next node; or send out through the HSPA modem directly. The decision is certainly not made in unorganized fashion. Instead, the selection of outgoing node is made by router *i* based on a set of rules. First of all, let D_i denote the new *unbalance factor* for D-router *i* after a new session joined, so we have following rules for handling it:

- *Rule 1*: If HSPA interface of D-router *i* is available to handle the newly joined session and resulting *unbalance factor* for this D-router is less than or equal to the original *unbalance factor* d_i, i.e. $D_i \leq d_i$, D-router *i* takes the privilege to accept this session and tunnel all packets to the proxy.

- *Rule 2*: If D-router *i* can not satisfy rule 1, it also evaluate D_i of two interconnecting ports and choose the one producing minimal D_i among the three interfaces it has.

This path discovery process should continue in every upcoming D-router until a suitable HSPA interface is identified for the joining session.

Session-based Rerouting: Besides accepting new sessions, traffic rerouting also runs in an infinite loop in the background to periodically check every D-router's loading to determine whether or not to trigger a session balancing rerouting process. For instance, if HSPA interface of a D-router is temporarily blackout, its associated traffics have to be rescheduled. DragonNet should begin the rerouting process on router *i* as long as either one of following conditions is satisfied:

- *Condition 1*: If the difference of the loading factor for D-router *i* and the rest of D-routers, either forward or backward, is larger or equal to twice of a minimum session weight of D-router *i*, DragonNet will reroute some or all of D-router *i* sessions to the side with smaller loading factors. Larger or equal to twice of minimum session weight is required because we want to avoid unnecessary rerouting oscillation between two neighboring nodes.
- *Condition 2*: If condition 1 check is not satisfied, DragonNet continues with condition 2. This condition basically checks whether the loading difference between D-router *i* and its previous one, D-router *i-1* is larger or equal to twice of the minimum session weight of D-router *i*. DragonNet will reroute some or all of D-router *i* sessions to D-router *i-1*. Or
- *Condition 3*: Similar with condition 2, but this one primary concerns about loading difference between D-router *i* and its next one, D-router *i+1*. If it is larger or equal to twice of the minimum session weight of D-router *i*, DragonNet will reroute some or all of D-router *i* sessions to D-router *i+1* if this check is valid.

By following a set of rules and conditions we derived the DragonNet Protocol. Although the session-based rerouting is presented with assumption that there is no random or cascading failure, it can handle the failures with no or merely a little modification to the protocol. In fact, if a random failure occurs, DragonNet simply set such D-router's status to NULL and then shift suspended sessions to other node according the rules and conditions described above. Cascading failure is a very special case in DragonNet decision making rules. Recall that causes of cascading failure is due to a LDT traveling through a cellular coverage black-hole, such as tunnel. It can be observed that the last node should stay on for the longest period of time and the first node should resume connection at the earliest time. To this extent, DragonNet deals with this special case by pushing all data to the tail node while the LDT is entering a tunnel and then switching to the head node when the LDT is leaving the tunnel. Given that network connectivity can change quickly, including from being connected to disconnected and back, some may ask how does the rerouting protocol deal with such connection "oscillations"? As we have discussed above, rerouting can only be triggered under certain conditions. Consequently, although explicit cellular link failure should trigger an immediate rerouting, resuming from disconnection will not necessarily do this if conditions are not met. Thus, rerouting protocol can efficiently prevent "oscillation" from happening. Moreover, we think another factor that may affect protocol performance is latency for transfer of packets across the DragonNet chain. Nonetheless, we will show in Section VI that protocol message processing time and RTT for transferring packets along the chain are negligible so that they will not affect the system performance.

V. IMPLEMENTATION

We have built a prototype implementation of DragonNet on Linux OS platform. The DragonNet protocol glues and coordinates all components together to provide non-interruptive mobile Internet services on the LDT. The proxy is implemented as a user-level application that is hosted on a machine on the wired network. The DragonNet protocol on D-routers is implemented on OpenWrt, a Linux distribution for embedded devices. The programs run on D-routers and proxy are both written in C/C++.

We have built the D-router from scratch since building our own D-routers gives us adequate flexibility for constructing DragonNet and DragonNet protocol. We adopted Broadcom's BCM5354 as the core of the D-router because it is a 802.11b/g Router System-on-Chip. This chip is equipped with a processor that is powerful enough to accommodate a light weight OS. For this reason, we have ported a comprehensive OpenWrt embedded Linux OS to the D-router platform. In spite of the maturity of OS, it only enables the traditional TCP/IP capability and the routing core lags far behind our rerouting requirement. Traditionally, the routing table stores the routes, and in some cases, metrics associated with those routes, to particular network destinations. More specifically, the routing table maps an incoming interface (a subnet) to an outgoing interface (another subnet). In this circumstance, all packets arrived at the same incoming interfaces should only be routed to the same outgoing interfaces. However, the D-router needs to route packets of an incoming interface to multiple outgoing interfaces. This behavior has largely violated the operation and exceeded the scope of conventional routing algorithms. As a result, we need to overcome such the challenge by entirely replacing the routing core of IP layer in D-routers with our own one. Another reason for using this chip is this chip comes with a USB2.0 host controller, we take the liberty to integrate it with a USB HSPA modem (USB WiMAX modem was also driven and tested in our lab, but unfortunately there is no WiMAX infrastructure in the city) so as to bridge local wireless group to mobile Internet via this interface. The driver in use is *usbserial*, and it is hacked to periodically check the wide area interface's health status for every 500ms by sending Hayes AT commands to the HSPA modem.

Having at least one D-router in each compartment, all D-routers on LDT are interconnected into a cooperative chain to be deployed in such "dragon-like" infrastructure. We have investigated basically two types of D-routers' interconnections. We first consider Wireless Distribution System (WDS) [2] for wireless interconnection of D-routers. The prominent advantage of WDS over other similar solutions is that it preserves the MAC addresses of client packets across links between the APs. Connections between D-routers are made using MAC addresses rather than by specifying IP assignments.

The DragonNet's traffics are classified into either control or user traffics. DragonNet restricts control messages to be exchanged within the DragonNet such that no modification of client and server applications is required. The DragonNet performs two routine tasks on user traffics: tunneling between D-routers and proxy and route selection among D-routers. That says, first, DragonNet re-encapsulates packets with new source and destination address so that they can reach the proxy before going to servers and vise versa; second, DragonNet protocol monitors and changes routes of ongoing traffic periodically for failure recovery and load balance. To achieve the objectives, DragonNet includes Netfilter and iptables to intercept and modify packets for routings. Netfilter and iptables are building blocks of a framework inside the Linux 2.4.x and 2.6.x kernel series. This framework enables packet filtering, network address (and port) translation and other packet mangling.

Packet encapsulation for tunneling requires adding an extra TCP header to each packet. Therefore, the overall length of the packet always exceeds MTU (Maximum Transmission Unit, usually 1,500 bytes) that leads to packet being discarded by traditional network elements. In light of this, as long as MSS plus extra header is larger than MTU, DragonNet sets MSS for packets in between of wireless clients and D-routers, proxy and servers to as the length of MTU minuses an additional header (which is 1,420 bytes for our test cases). Consequently, packets exchanged between D-routers and proxy, though added an extra 40-bytes TCP header plus a 12-bytes proprietary header, would not longer than 1,500 bytes.

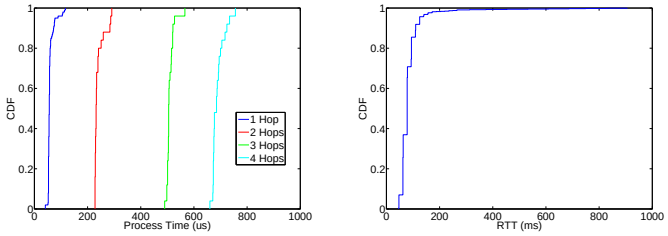
VI. EVALUATION

We have implemented the DragonNet system at the length of 25 nodes. But for the convenience of conducting tests on real railways, we have tested DragonNet with 4 D-routers in two railways. We have also carried out extensive tests for 25 nodes of which every one is embedded with a random and cascading failure generator. By comparing Fig. 4 & Fig. 5, we can see that processing delay for in-lab and field tests are highly correlated. Even in the most extreme case, the packet processing delay (including queuing delay) does not exceed 900us in both tests. Moreover, we also show in Fig. 5b the CDF of internal RTT for 25-hops DragonNet is less than 150ms, which is much smaller than values shown in Fig. 4b when RTT from D-router to server is also counted. Fig. 5b also evidences that propagation along the long stretching DragonNet will not deteriorate the network performance. Thus, we argue that test results for 4 nodes are still valid for 25 nodes. In this section, we will present the experimental evaluations of DragonNet. We stay focus on presenting on site test results due to lack of space.

A. Testbed & Setup

Fig. 3 describes the testbed utilizing East and West railway lines in Hong Kong for conducting on site tests. For the sake of easy coordination between testers, we have configured the DragonNet to have the length of four battery-powered D-routers across the first four compartments which stretches about 100m in length. The D-routers are interconnected with only WDS for these tests due to convenience of deployment. All D-routers were attached to HSPA network for providing mobile Internet services. Locations A, B, C and D marked

Fig. 3. East (blue line) and West (purple line) railways in Hong Kong (based on Google Maps). Locations A, B, C and D are cellular coverage black-holes



(a) (b)

Fig. 4. Field test performance: (a) CDF of processing delay; (b) CDF of round trip time (RTT) including processing delay introduced by DragonNet protocol and proxy

in Fig. 3 are cellular coverage black-holes where cascading failures can be easily observed.

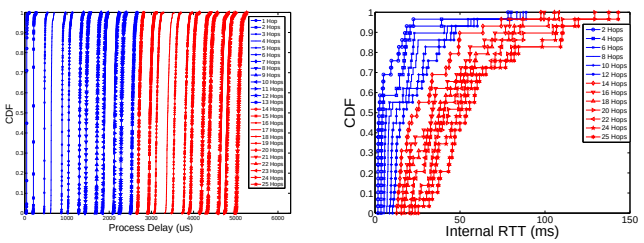
There are also two proxies involving in on site tests. Two identical proxies are separately hosted in our lab and a commercial data center to ensure the result will not be affected by anything specific to one proxy and its path. The client device can be any types of WiFi capable devices. We use netbooks running Ubuntu 9.04 and Windows XP for the tests to make sure OS dependent issues are dispelled. In mobile tests we placed a D-router in the middle part of each compartment and interconnect them with WDS as described above. At least one mobile client was associated with one D-router.

B. Protocol Performance

We are going to evaluate end-to-end delay, including processing delay, for individual D-router and proxy under both lightly and heavily loaded situations.

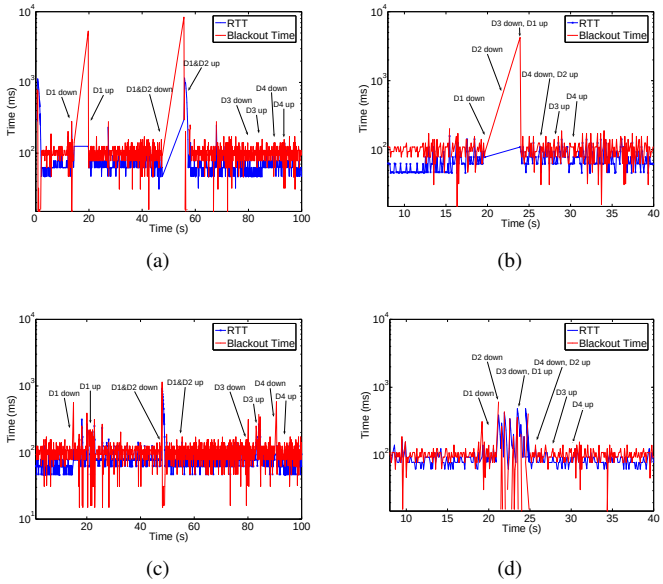
DragonNet Blackout Reduction: One of the main factors that affects end-user experience is the frequent presence of blackout periods in cellular networks. These blackout periods are normally due to interference problems, hardware failures, or loss of connectivity or coverage. We will quantify the performance of DragonNet under blackouts in this section.

These tests were done by sending bursts of data packets back-to-back from client to the server via HSPA links and measures throughput based on the inter-arrival times between packets in a burst. To measure the blackout periods we collected traces of packets in railway environments during a period of 3,600 seconds for each D-router and for the DragonNet. For each of these traces we identified those periods



(a) (b)

Fig. 5. In-lab experimental performance: (a) CDF of processing delay (b) CDF of internal round trip time (RTT) excluding RTT from DragonNet to servers



(a) (b) (c) (d)

Fig. 6. (a) & (b) are snapshots for random and cascading failure blackout respectively without DragonNet protocol; (c) & (d) are the corresponding snapshots (a) & (b) respectively with DragonNet protocol

of time where the inter-arrival time between packets was greater than a given blackout threshold. The snapshots for random and cascading failure caused blackout for D-routers D1, D2, D3 and D4 with and without DragonNet protocol in one of the traces are shown in Fig 6. It can be seen that snapshots, Fig. 6c & Fig. 6d, for the one with DragonNet protocol enabled give flatter and narrower curves than those without, Fig. 6a & Fig. 6b.

As shown in Fig. 7, we show the percentage of time spent in a blackout period, assuming a variable blackout threshold in the x-axis. From this figure we can see that the amount of time spent in a blackout can be quite significant for a given D-router. This is a natural effect of the fact that under mobile environments, mobile devices suffer frequent cell handoffs, loss of coverage, and sudden disconnections. When comparing the above results with the percentage of time spent by the DragonNet system in a blackout, we see that the DragonNet spends a much smaller portion of the time in a blackout. Thus, the probability that the DragonNet user cannot receive data from any of its D-routers for a period of 10 seconds or more

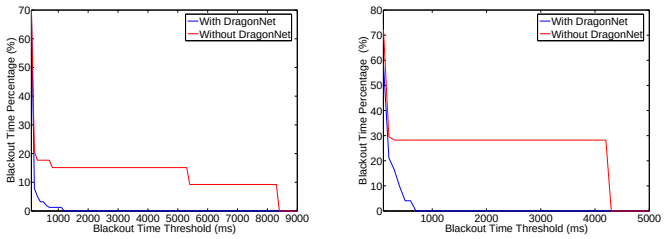


Fig. 7. Blackout time reduction. (a) Distribution of blackout time for random failure; (b) Distribution of blackout time for cascading failure

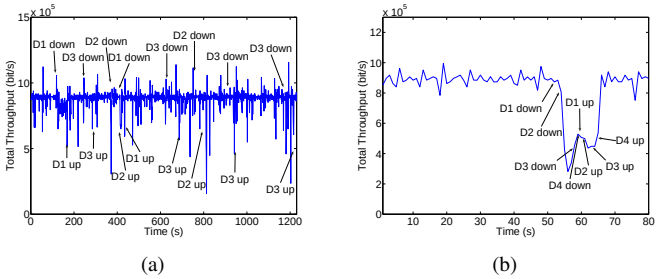


Fig. 9. A typical example of aggregated throughput for DragonNet with (a) random failure; (b) cascading failure

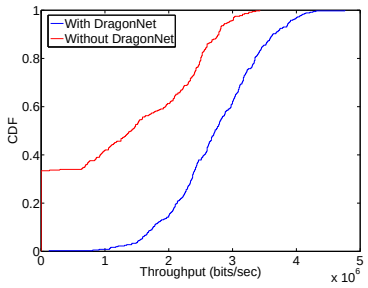


Fig. 8. CDF of system throughput with and without DragonNet

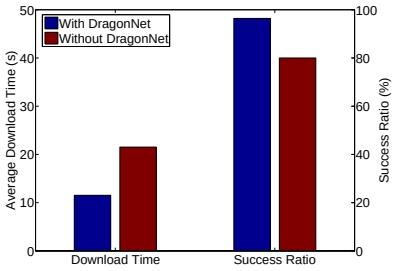


Fig. 10. CDF of system throughput with and without DragonNet

is almost negligible. Therefore the DragonNet significantly increases resilience against network failures due to its dynamic rerouting strategy.

DragonNet Throughput: Fig. 9 is a typical time-series example of individual and aggregated throughput for chained D-routers with and without DragonNet during random and cascading failures. Conventionally, for UDP traffics, all packets flowing through failed D-routers are lost but the session remains. On the contrary, TCP resets its connection if the blackout time is larger than predetermined timeout threshold. However, both are unwanted side effects for communications. In Fig. 9a, the HSPA interfaces of four D-routers failed randomly as the time passed by, every D-router involving in failures sees no throughput until its connection restores. This also applies for cascading failures as shown in Fig. 9b.

There are two sets of data samples, one is collected from DragonNet and another one is collected from chained D-router without running DragonNet protocol. The results, as illustrated in Fig. 8, shows that overall throughput of DragonNet is always higher than other networks. The DragonNet explicitly appears to have a factor of 2 times higher throughput than conventional network. Nonetheless, we are optimistic to foresee a greater improvement if more and more D-routers are included in the DragonNet. In fact, the percentage of improvement varies with respect to probability of actual link failure and connection blackout. In mobile scenarios, as we measured in Hong Kong over a period of time, each D-router (i.e. mobile interface) experience about 10% connection blackout time of total journey of 60 minutes. DragonNet takes advantage of channel diversity to relay traffic for broken links. Therefore, it is estimated that the more connection blackouts, the greater improvement DragonNet can have. Even in such a mobile environment, the impact of highly variable bit rates in the DragonNet is not as pronounced as for each individual gateway due to the fact that DragonNet exploits the benefits offered by long stretching geographic channel diversity.

C. System Performance

In this section we study the performance of a DragonNet client Web-browsing's session to illustrate a real user experience on presence of link failures.

Given the fact that web-browsing is the most common Internet service, we adopt web-browsing traffic to measure DragonNet's application performance. First of all, we replicated a copy of *cnn.com* front page to our test server. This is done to avoid official update so that we can keep our tests fair and consistent. The front page of CNN consists of 89 objects all together of size 1145 Kbytes. With average bit rate of 400 Kbps each, the Firefox web browser can often fully download the page within 20 seconds using HSPA access. In light of this, we implemented a Firefox web browser plugin application to randomly refresh it so as to repeatedly download *cnn.com* for these tests. The randomness is chosen in between 30 to 60 seconds to make sure that time interval is reasonably long to have concurrent access as well as individual access among clients during the tests. Caching is disabled to force web-browser downloading from server when each time "refresh" command is triggered. We believe the tests reflect actual behaviors of passenger inside a compartment of commuter train, where they access to the Internet in purely random-manner.

In Fig. 10 we show the average response time seen by the mobile browser for test cases. From this figure we can clearly see that the on average mobile users perceive nearly no disconnection even though their associated D-router has lost connection with wide area network.

VII. Related Work

Much research work has been done on the integration of mobile networks and Internet. Hitoshi et al. [6] proposed utilizing heterogeneous wireless links which can provide both continuous slow connection and intermittent fast connection for broadband access service on train. Lannoo et al. [13], [12] proposed extensions to Gavrilovichs [8] moving base stations model. The authors argue, just as in [9], that frequent handoffs greatly reduce the bandwidth available to fast moving users. Consequently, they propose using radio-over-fiber, to feed base stations along the rail track. Unlike in Gavrilovichs model there are no moving base stations; instead there is a fiber-fed distributed antenna network. Ishizu et al. [10] observed that leaky coaxial cable (LCX) has been used throughout Japan for radio communications on trains, thus they propose an architecture for communications on "bullet trains" that consists of a base station with an Ethernet interface and mobile devices. Bianchi et al. [7] stated that it may be expensive to wire a train for network access and rewiring may be needed every time the train is reconfigured. Therefore, they proposed using IEEE 802.11 to construct a wireless network between the train cars. Trains can be connected to the Internet via a satellite link [14]. But satellite link has substantially high End to End delay. Aguado et al. [5] presented a network architecture based on WiMax for use in railway environments because it can provide mobility support at speeds up to 500 km/h. Kumar et al. [11] introduced an architecture called SWiFT. This architecture consists of IEEE 802.11e access points within train carriages for the on-train network, IEEE 802.16m base stations at the trackside for the access network, and an optical backbone for linking the IEEE 802.16m base stations to the global Internet. M. Luglio et al. [15] proposed the Noordwijk TCP protocol to resolve obstructions of the line of light in state of the broadband coverage over the railways. However, none of the published work so far provides solution on the entire infrastructure for network on train which can be flexibly implemented in real world.

VIII. Conclusion

In this paper, we discussed the limitations of the current wireless access systems in the commuter train due to dynamic nature of wireless communication. To that end, we made a case for exploiting the length of LDTs for networking opportunity. We introduced DragonNet, a unique system formed by a chain of D-routers that utilizes multiple healthy cellular wireless links to amortize deteriorated ones and thus provides local users with a more reliable access network than which can typically be provided by a single cellular gateway. As a result, the supporting protocol was devised to support DragonNet. DragonNet protocol manages the DragonNet, detects, predicts and reacts to failures. The benefits of employing a special protocol, along with field test analysis showing that DragonNet can provide more stable and sustainable data rates for applications such as web browsing. The benefits can all be achieved without requiring users to perform any software or configuration updates on their mobile devices.

IX. Acknowledgement

The paper is partially supported by CityU Strategic Research Grant No. 7008110; CityU Applied R & D Centre (ARD(Ctr)) No. 9681001; RGC General Research Fund (GRF), SAR HK, No. (CityU 114609); ShenZhen-HK Innovation Cycle Grant No. ZYB200907080078A; and China NSF No. 61070222/F020802. The above work is partially supported by the US National Science Foundation (NSF) CAREER Award CCF0546668, CNS0916584 and the Army Research Office (ARO) under grant AMSRD-ACC-R50521-CI. Any opinions, findings, conclusions, and recommendations in this paper arethose of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] http://news.bbc.co.uk/2/hi/technology/3729583.stm.
[2] http://www.dd-wrt.com/wiki/index.php?title=wds_linked_router_network.
[3] http://www.icomera.com.
[4] http://www.railway-technology.com/features/feature1150/.
[5] M. Aguado, O. Onandi, P. S. Agustin, M. Higuero, and E. J. Taquet. Wimax on rails. *IEEE Vehicular Technology Magazine*, 3(3):47–56, Sept 2008.
[6] H. Aida and S. Kambori. Effective use of heterogeneous wireless links in high speed railways by predictive scheduling. *International Symposium on Applications and the Internet*, pages 459–462, 2008.
[7] G. Bianchi, N. Blefari-Melazzi, E. Grazioni, S. Salsano, and V. Sangregorio. Internet access on fast trains: 802.11-based on-board wireless distribution network alternatives. *12th IST Mobile and Wireless Communications Summit*, pages 15–18, 6 2003.
[8] C. D. Gavrilovich. Broadband communication on the highways of tomorrow. *IEEE Communications Magazine*, 39(4):146–154, Apr 2001.
[9] F. D. Greve, B. Lannoo, L. Peters, T. V. Leeuwen, F. V. Quickenborne, D. Colle, F. D. Turck, I. Moerman, M. Pickavet, B. Dhoedt, and P. Demeester. Famous: A network architecture for delivering multimedia services to fast moving users. *Wireless Personal Communications*, 33(3-4):281–304, 2005.
[10] K. Ishizu, M. Kuroda, and H. Harada. Bullet-train network architecture for broadband and real-time access. *12th IEEE Symposium on Computers and Communications*, pages 241–248, July 2007.
[11] R. K. K, P. Angolkar, D. Das, and R. Ramalingam. Swift: A novel architecture for seamless wireless internet for fast trains. *Vehicular Technology Conference*, pages 3011–3015, 5 2008.
[12] B. Lannoo, D. Colle, M. Pickavet, and P. Demeester. Extension of the optical switching architecture to implement the moveable cell concept. *31st European Conference on Optical Communication*, 4:807–808, Sept 2005.
[13] B. Lannoo, D. Colle, M. Pickavet, and P. Demeester. Radio-over-fiber-based solution to provide broadband internet access to train passengers. *Communications Magazine*, 45(2):56–62, Feb 2007.
[14] X. Liang, F. Ong, P. Chan, R. Sheriff, and P. Conforto. Mobile internet access for high-speed trains via heterogeneous networks. *The 14th IEEE International Symposium on Persona1,lndoor and Mobile Radio Communication Proceedings*, 1:177–181, 2003.
[15] M. Luglio, C. Roseti, G. Savone, and F. Zampognaro. Tcp noordwijk for high-speed trains. *First International Conference on Advances in Satellite and Space Communications*, pages 102–106, July 2009.
[16] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee. MAR: a commuter router infrastructure for the mobile internet. *MobiSys '04*, pages 217–230, 2004.