Journal of Mobile Multimedia, Vol. 12, No.1&2 (2016) 001-016

A FLEXIBLE ROUTER WITH TANGIBLE NETWORK INTERFACES FOR SHARING A LAST MILE AND ITS PERFORMANCE

Toshikazu NISHIMURA

College of Information Science and Engineering, Ritsumeikan University, Japan
tnt AT is.ritsumei.ac.jp

After the big earthquake in 2011 in Japan, the disaster survivors lost the means to acquire and convey information in the stricken area. In this paper, we propose a means to share seamless access to the Internet immediately among survivors using surviving access lines including ground infrastructures and satellite communications even if disaster exceeds assumption. We introduce the notion of TNI (Tangible Network Interface) that can easily configure small handy router named Flexible Router on site without configuring the settings of router from external console device. We also introduce Integrated Wireless Authentication System to share Internet access line among disaster survivors. We implemented Flexible Router and its TNIs and demonstrate its performance as well as the solution to improve its poor performance of Wi-Fi device and the intercontinental throughput measurement over VPN (Virtual Private Network).

Key words: Wi-Fi hotspot, disaster mitigation, last mile, Tangible User Interface, flexible router

1 Introduction

Catastrophic natural disasters, such as the 2011 Tohoku Earthquake and tsunami [1], cause crucial physical damage disabling us for using land communication infrastructure including communication base station and telephone networks in the stricken area. Without communications, it is difficult to grasp disaster situation and the request of relief goods. The damages by natural disaster vary in area and also do the relief that disaster survivors need since most of the disasters are heavily related to the geographical location. Without exact request, support from outside are apt to be sent to the most sensational stricken region by hearsay. That led to a support gap or a new type of digital divide for each shelter by communication system in the last disaster. For suitable disaster survivor support, it is necessary to solve such a divide by recovering the function of a communication promptly in the case of a catastrophic disaster. Therefore, in this paper, we propose a means to share seamless access to the Internet immediately among survivors using surviving access lines including ground infrastructures and satellite communications.

Supporting backup redundancy is one of the solutions to maintain networks after a natural disaster. Heavily equipped redundant network system might work as disaster countermeasure. Although this approach is important for trunk line, it is too expensive to provide against various types of natural disaster for access line or last mile. Also, this approach is powerless for a disaster beyond assumption like the 2011 Tohoku Earthquake and tsunami.

Mesh networking (multi-hop / mobile ad hoc networking) [2] and DTN (Delay / Disruption-Tolerant Networking) [3] are the powerful technologies to construct data link among survivors of disasters rapidly without depending on wired ground infrastructures. The media to carry data in these research works are weak radio wave among portable routers in mesh networking and physical transporter like automobile / pedestrian volunteer in DTN. It would be useful to share information among survivors of disasters as trunk lines. However, since the network protocol used in these technologies is not compatible to the ordinary one like IP (Internet Protocol), exclusive routers running unified routing protocol and special end nodes for user supports should be prepared in advance to utilize those technologies in wide area practically.

In our approach, we emphasize that flexible adaptation is desirable from this point of view. The small inexpensive network routers should be provided against natural disaster at every shelters or bases of rescue operations in order to provide network connection for every disaster survivor by sharing lasting access lines. In our proposal, a Flexible Router (hereafter referred to as FR) is the network router that implements such flexible adaptation. Since the damage of access lines varies according to the nature of disaster, it is difficult to predict in advance precisely what types of access lines would last. It might be redundant optical fibre networks to local government headquarters, expensive satellite communication system for disaster mitigation, cellular radio wave from neighbour base stations, or grass roots computer network using mesh networking or DTN technologies. In any case, complicated manual configuration on site for multiple types of NIs (Network Interfaces) and network protocol is needed if we apply these by a generic router. In this setting, flexible adaptation would be implemented by the efforts of expert professionals. However, it is doubtful to dispatch engineers to all shelters to set a router up. The FR itself should provide flexible adaptation.

Also, along with the spread of mobile computing, people come to possess their own Internet terminals of various forms. Therefore, it is desirable to open an Internet access to these personal terminals rather than to offer a limited number of special-purpose terminals connected to the Internet for public use. In this approach, the types of the user terminals that can be used in the stricken area may vary from smart phone / tablet PC (Personal Computer) to high-end PC. Multiple types of NIs are desirable for this heterogeneous user environment. We also have to prepare for on-demand configuration for a specific terminal that has special NI.

2 Flexible Router

2.1 Flexible Router and its Component

Therefore, we introduce the notion of TUI (Tangible User Interface) [4] into the configuration of FR for flexible adaptation of the lasting access lines and for providing suitable access interface in the heterogeneous user environment. A TUI is a UI (User Interface) where a user can interact with digital information through the direct physical environment rather than through indirect, virtual or computer graphic environment. In this paper, we introduce two types of abstraction of routing modules. One is the hardware NIs of our router called TNIs (Tangible Network Interfaces) named after TUI. A bare router should be armed with suitable TNIs that can be easily attached on it on site in order to adopt the lasting access lines and the terminal interfaces in our idea. The bare router is the body of what we call FR. This component virtually offers a universal bus that can connect multiple TNIs and that forwards transmitting packets among TNIs. The function of the bus is called FR bus apart from the FR body here after.

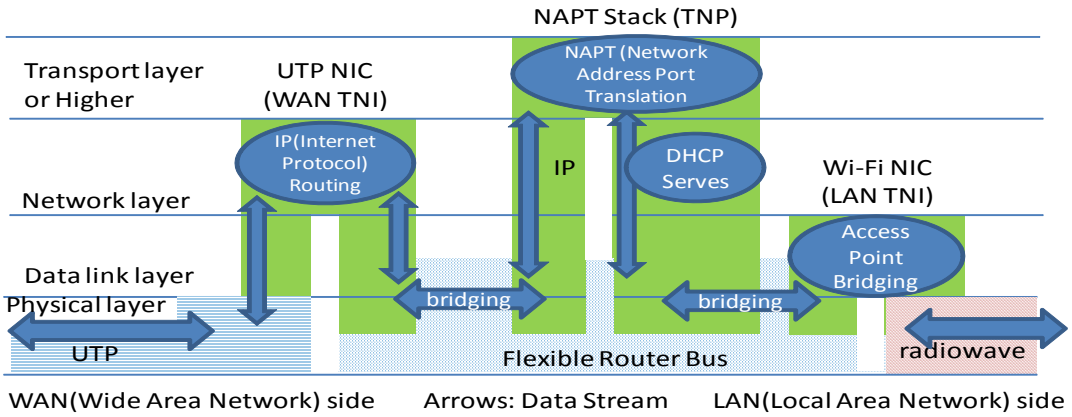

Fig. 1 The example: Wi-Fi Router using Flexible Router.

Each TNI is equipped with its network hardware port like RJ-45 (Registered Jack), that is, the 8P8C (eight positions eight contact) modular connector for UTP (Unshielded Twisted Pair) cable media or is equipped with the semiconductor chip that deals with its wireless network media like infrared or radio wave. They are to be used as converters between the specific network signals to packets on FR bus. Even if we have no console to maintain router body, we can manually configure the bare router on site by choosing the TNI with suitable network hardware port or network media when the bare router can appropriately configure the TNI automatically. If we provide several TNIs that contain specific default configurations for the same network media, we can choose suitable configuration by attaching proper TNI without changing the settings of bare router from console. So our FR with TNIs can be configured manually without hardware console device or software control panel for expert professionals.

Another type of abstraction of routing modules is software module including router configuration, TNI configuration or the configuration of special network protocols like authentication, security or tunnelling. We virtually detach them from TNI hardware and embody them as another hardware module that can be easily attached to the armed router. In this paper, it is called TNP (Tangible Network Protocol). Although TNP should be also attached to the bare router as TNI should, TNP has neither external hardware port nor chip that deals with network media. TNP could be attached to a universal bus as if they change the default configuration of specific TNI or bare router itself, or as if they convert the data between specific two TNIs with special network protocol. Thus TNP can be used for configuring the router in startup or enhancing the functions of the routers with special network protocols without configuring it manually from external console device by expert professionals.

2.2 TNP modals

The idea of TNI and TNP is simple in syntax and easy to operate, their function, implementation, and semantics are rather complicated. We discuss the problems of their semantics and modals here. Figure 1 shows the example configuration of broadband router with Wi-Fi AP (Access Point), that is, what we call Wi-Fi router based on a FR bus and two TNIs, one for WAN (Wide Area Network) side connected over UTP cable, another for LAN (Local Area Network) side acting as Wi-Fi AP with one TNP for NAPT (Network Address Port Translation) [5]. The TNI for WAN is a UTP NIC (Network Interface Card) with suitable default configuration to access to the Internet over IP. The data from the Internet are converted into the data stream over FR bus to the TNP. The

TNP consists of two services denoted as two ellipses in the figure. One is the NAPT that enables Internet sharing among multiple clients by NAPT software modules using specific private IP address space for LAN side. Another is the DHCP (Dynamic Host Configuration Protocol) [6] server that configures the clients that are connected to a LAN side. The translated data from the Internet are converted into the data stream over FR bus again to the TNI for LAN. This TNI is a Wi-Fi NIC with suitable firmware for Wi-Fi chip and suitable default configuration acting as Wi-Fi AP. Associated Wi-Fi clients can be connected the LAN side and be properly configured by the settings derived from DHCP server, then they can receive the translated data from the Internet via the bridging function of LAN TNI. The data from the Wi-Fi clients are sent to the Internet in opposite direction, via LAN TNI, TNP and WAN TNI.

This figure demonstrates how FR bus works in emulating Wi-Fi router and also shows the importance of the discussion on the modals of UI. A TNI or a TNP has only two states from the view of the function of the module. One is detached state in which every function of the module is lost. Another is attached state in which some function of the module is provided. This is a simple syntax if we regard the state as on-off switch. However, compared to this syntax, the semantics and their context are rather complex.

1) Asymmetric Flow

As we show in figure 1, The FR bus is divided into two bridges by the TNP. This means a TNP should distinguish at least two types of TNIs, WAN side and LAN side. We denote the former TNI as WAN TNI and denote the latter one as LAN TNI. This is mainly because our FR is not for trunk line but for the end node of the Internet. Moreover, NAPT is the protocol of asymmetric flows.

2) Scope of Hidden Variables

A routing table is a table that lists the routes to a specific network destination. This means only one routing table exists in a router and updating a routing table by a WAN TNI configuration affects the whole routing function. So it is undesirable to attach multiple WAN TNIs to a FR bus without using dynamic routing protocol that manages the routing table properly. A TNP module changing default settings of FR itself might lead to a similar problem. Through such a hidden global variable, modules may conflict with each other and should be automatically mediated by meta-configuration rules like priority control or dynamic routing protocol.

On the other hand, MAC (Media Access Control) address learning is a function of bridging that caches MAC addresses of peer nodes in the bridge in order to avoid unnecessary flooding of data frames. Although MAC address learning also needs a cache table, each entry contains bridge ID (Identification) and NI's ID. So module conflict is limited in a bridge group. Handling of hidden local variables is also needed.

3) Providing Independent LANs

Conversely, multiple LAN TNIs that create their own bridges with protocol translation can co-exist. This means each LAN TNI can provide independent LAN with specific service and policy connected to the Internet within the network resources provided by a WAN TNI. For example, bridging layer 2 VPN (Virtual Private Network) [7, 8] to a home network (an Intranet of one's organization) located outside of the stricken area and specific Wi-Fi configuration can provide a remote Wi-Fi AP that supports easy access to the home network at stricken area over the same configured Wi-Fi radio wave as at the home network.

Some APs provide multiple SSIDs (Service Set IDs) over Wi-Fi as if a single AP serves multiple APs. Each SSID provides the access to a specific VLAN (Virtual LAN) by specific authentication and encryption. This technology is mainly for Wi-Fi environment in limited area like enterprises or business.

3 Sharing Internet Access

3.1 Access through Open Wireless Local Area Network

A Wi-Fi hotspot [9] is a service that offers Internet access over Wi-Fi in a small limited area like fast food restaurant, café, airports, hotels and other public places. A hotspot or providing an open Wi-Fi without authentication nor security is one of the methods to share Internet access line among multiple users. After the 2011 Tohoku Earthquake, some commercial hotspot service providers disabled user authentication and made their hotspots open to public for free to provide Internet connection to the disaster survivors. One of the providers opened their all hotspots in Japan [10] for about four weeks and the hotspots around the stricken area for over one year. This is one of the great examples sharing a lasting internet connection among many disaster survivors.

However, this type of methods has several problems. It is hard to trace such abuse as attacking remote sites, or disclosing of private information and so on since it has no authentication. The traffics among clients and servers on the Internet, containing secrets, private information like plain password or credit card transaction may also be sniffed by third party since it has no security. The IWAS (Integrated Wireless Authentication System) [11] is a system that integrates multiple authentication services in order to allow the users of various organizations connecting Wi-Fi APs as Internet access method. In their framework, four basic ideas are discussed [12, 13]. To share Internet access line among disaster survivors, we focus on their idea.

The first one is redirecting packets from unauthenticated clients to external authentication server on the Internet via tunnelling or VPN rather than discarding them at AP. The packets from the clients that are not authenticated by AP, should be encapsulated by IP packets keeping them from reaching other LAN nodes directly.

The second idea is distinguishing suitable authentication / VPN server address from other servers within the authentication scheme. The combination of these two ideas means that multiple VPN servers and independent authentication servers of multiple organizations can coexist in order to authenticate users of various organizations in order to share one Internet access line.

The third idea is the source IP address of the redirected packets authenticated by external authentication server sent to the Internet. They focused on the Internet security, including the traceability of abuse by tracking the authenticated clients and prohibition of anonymous access and abuse as a springboard on the shared Wi-Fi and Internet access line.

The fourth idea is the table server on the Internet that serves the table of the IP address of authentication servers including VPN servers that can be temporally cached on AP. Combining these ideas, they discussed the multiple external authentication servers on the Internet for multiple groups or organizations.

3.2 Integrated Wireless Authentication System framework

Figure 2 shows our FR in IWAS framework. The implementation of the IWAS is rather simple. They introduced closed Ethernet link over Layer 2 VPN that conveys authenticated packets over

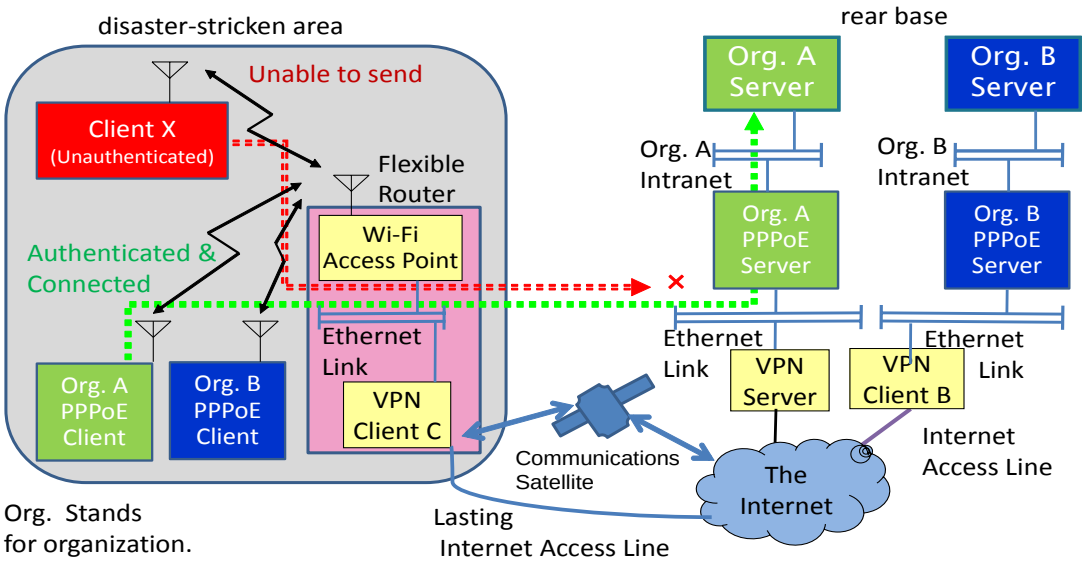

Fig. 2 Flexible Router in IWAS framework.

the Internet. We have three Ethernet links in the figure. Although the Ethernet link is to open to users through open Wi-Fi network, there is no gateway that admits anonymous access to the Internet. So it is hard to abuse it or to threaten Internet servers or users. In the figure, the unauthenticated client X is unable to connect to the outside.

They introduced networked authentication protocol like PPPoE (Point-to-Point Protocol over Ethernet) [14] or PPTP (Point-to-Point Tunneling Protocol) [15] for distributed authentication on the Wi-Fi-opened Ethernet link. We have two organizations A, B that have their own intranet and server in the figure. After the authentication server admits their genuine user account from the client PC, say, the PPPoE client of Org.A in the figure, the PC can access to the server on their intranet.

So in IWAS framework, the data packets from their client are encapsulated twice. First, they are encapsulated in authentication packets like PPPoE frame in the client PC to be sent to the Wi-Fi-opened Ethernet link. Though the link, the authentication packets are encapsulated in VPN frame in a VPN client to be sent to its VPN server over the Internet.

In order to support disaster survivors by IWAS framework, we should place the Wi-Fi-opened Ethernet link everywhere including in disaster-stricken area in order that disaster survivor can access open resources on the Internet or closed ones of his/her organization in their intranet. We introduce Wi-Fi-opened Ethernet link of IWAS framework in order to share Internet access lines among disaster survivors. That is, we should introduce TNI with Wi-Fi AP that is bridged to the Layer 2 VPN client. Thanks to the connection among VPN server and its clients, we can place authentication servers outside of the stricken area as supporting rear bases.

4 Prototyping

4.1 IWAS framework

We have set up two PPPoE server hosts with independent Service-Name Tag for simulating two organizations. The hosts are running FreeBSD 9.3-RELEASE *amd64* architecture (FB93amd64)

and we set up the server program in the OS (Operating System), `pppoed` utility including `pppd`, the daemon program for PPP (Point-to-Point Protocol) [16]. Each host is connected to both our laboratory's Intranet and an Ethernet link through independent NI cards.

We have set up bridged VPN (Layer 2 VPN) by the free and open source VPN software, OpenVPN 2.3.5. The VPN server host is running FB93amd64 on Intel Core i7-3770K CPU (Central Processing Unit) running at 3.5GHz (giga hertz) with 1GB (giga bytes) of memory and has enough computing power to serve as a VPN server. We built the software from source code. The NI card connected to the Ethernet link is bridged to the TAP interface, the software NI for Ethernet tunnel to which OpenVPN server accepts the connection from the clients.

4.2 Flexible Router Bus

In order to implement FR with TNIs, we focused on USB (Universal Serial Bus) as plug and play interfaces. Plug and play interfaces are used simply because they require no device configuration by user on site. The system administrator, that is, the manufacturer of FR, only has to install software for the self-configuring devices once before use. We choose USB because we have already several devices that can support Input / Output as NIs. USB communications device class is one of the classes of USB device that supports wired communication like Ethernet networking or serial communication including modem. Wireless Controller is one of the classes of USB device that supports radio wave communication like Wi-Fi or Bluetooth. Although PCI (Peripheral Component Interconnect) Express or other types of expansion slots are popular for desktop PC and have variety of interface cards, we rejected them since they are too refractory to be installed by novice user.

Since USB is also popular nowadays, almost all PC including handy video game or some smart phone can accept USB device. Although all the computers that have USB Host can be used as FR bus, we first introduce the compact network storage adaptor, IO-DATA USL-5P running OpenBSD 5.6/*landisk* as OS for the first prototype.

It has only four LEDs for user define output, and five USB 2.0 ports on the body so that USB devices can be attached as TNI and TNP. Its CPU is a Hitachi SH4 SH7751R processor running at 266MHz (mega hertz) with 64MB (mega bytes) of memory. Although its computing power is poor compared to a current PC or a smart phone, we believe it is enough for prototyping as a network routing equipment. Also, it is handy (150 grams in weight) and low power consumption (a maximum of five volts, two amperes). We can drive it for a long time by an external battery mainly made for recharging smart phone and note PC. The 18 ampere-hours battery of 3.7 volt is also portable (515 grams) and will last in seven hours at worst. We believe it is suitable for feasibility study.

We first introduce OpenBSD because of its rich support of network bridging (layer 2 forwarding) and encryption over Wi-Fi like WPA (Wi-Fi Protected Access) 2. Bridging is used to emulate the multiple buses discussed in section three. All the Wi-Fi settings in our later experiment use WPA2. Figure 3 shows the implementation of FR with a WAN TNI connected to UTP cable and a LAN TNI as Wi-Fi AP.

In the latter half of our experiment, we introduce laptop PC running FreeBSD RELEASE-10.2 / *amd64* architecture as OS for FR body of the second prototype to check the performance of our first prototype. Here after, we call the first prototype on USL-5P “thin FR” because of its poor limited computing resources. On the other hand, we call the second prototype on this laptop PC “thick FR” because of its rich computing ones. The laptop PC we use is rich in computing power

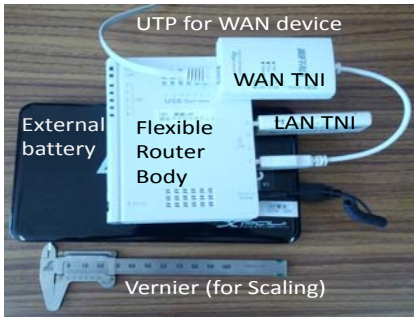

Fig. 3 The first prototype of Flexible Router and TNIs.

from an Intel Core i5-2467M CPU running at 1.6GHz with 4GB of memory. FreeBSD is one of the descendants of Berkeley Software Distribution (BSD) series of Unix variants as OpenBSD is. We do not introduce OpenBSD for the second prototype but FreeBSD because of its Wi-Fi software stacks. In OpenBSD, network drivers in kernel modules fully support WPA2 and host AP (Access Point) mode, while user-land supporting daemons are needed for WPA2 and host AP mode in FreeBSD. This means the network drivers of these OSs are not identical and might affect the performance of our prototyping. We will discuss it later.

4.3 Configuration of USB devices

The current version of OpenBSD supports ten types of wired NIs over USB and fourteen types of Wi-Fi ones by the native device drivers. Since the devices present different sets of software endpoints (called `ifmedia(4)` interfaces) in OpenBSD device drivers, we can distinguish which USB device is attached from user-land commands to check the `ifmedia` interface without developing special device drivers or kernel extension code for FR body. This means we can simply introduce shell script for describing the default configuration as a TNI after checking the type of USB device by the name of the `ifmedia` interface. So in our implementation, we describe the short shell script that configures default settings for each `ifmedia` in the built in file system and call them at start-up of the OS. The meta-configuration that automatically mediates the conflict among WAN TNIs is the calling sequence of each script in this implementation.

In the same scheme, TNP can be implemented as a mere USB flash drive. At start-up, FR mounts the drive in order to call setting shell script inside to activate special network protocol stack or to change the settings of specific TNI or router itself if any.

Since this method only distinguishes not an individual USB device but the types of device driver, the number of the type of TNI in our system is limited to the number of supported types of interfaces utmost. However, we believe it is enough for prototyping and feasibility study because a TNP can be used to override the default configuration of any TNIs.

Table 1 The Implemented TNIs and TNP.

`ifmedia` name	TNI type	Function
re0	WAN /LAN	Wired, configured by DHCP, falling back to LAN type if no carrier. (After falling back to LAN type) providing NAT & DHCP service for sharing WAN access.
aue0	WAN	Wired, authenticated and configured by PPPoE over satellite phone.
axe0	WAN	Wired, configured by DHCP.
rum0	LAN	Wi-Fi AP providing IWAS open access
rum1	WAN	Wi-Fi client of home network (an example of Wi-Fi closed network) configured by DHCP.
ural0	LAN	Wi-Fi AP providing lab. Network (closed Wi-Fi access by WPA2).
ural1	WAN	Wi-Fi client of tethering of 3rd generation mobile communication.
run0	WAN	Wi-Fi client of open Wi-Fi
urtwn0	WAN	Wi-Fi client of home network (an example of Wi-Fi closed network) configured by DHCP (used if no rum0 device)
(sd0)	TNP	Configuring urtwn0 as rum1 / Configuring rum0 as ural0
(sd0)	TNP	Configuring rum0 as Wi-Fi AP

The kernel interface of network interface in FreeBSD is similar to the one in OpenBSD. Although it is called `ifnet(9)` in FreeBSD, the simple short shell script can also configure the default settings of `ifnet` in our second implementation. Porting the shell scripts of thin FR on OpenBSD to FreeBSD was easy and the most of the changes for the porting were depended on the virtual network interfaces for VPN, ones for the Wi-Fi link layer, and the user-land daemons for Wi-Fi connectivity.

4.4 Implemented TNPs

We have implemented one TNP and nine types of TNIs including on-board wired NIC for thin FR bus. Table 1 shows the list of the TNIs and the TNPs. In order to show the flexibility for WAN media, we prepared five WAN TNPs for four types of lasting Internet access methods, PPPoE over satellite phone, DHCP configured LAN, Wi-Fi closed home network and tethering of 3rd generation communication network over Wi-Fi although they cannot be used simultaneously as we discussed in section three. The Internet access methods are limited to the commercial based ones just to show the performance of the implemented prototype. Later, we will discuss the possibility of applying our FR to another type of networking technology for emergency like mobile mesh network. In order to show the flexibility for the independent LAN TNPs, we prepared three LAN TNPs for the access from clients, sharing WAN access by NAPT configured by DHCP (open access through UTP), by providing closed Wi-Fi access, and by providing open access through IWAS framework. We measured the performance speed via *rum* device supporting IEEE 802.11g standard and host AP mode in order to unify conditions of experiment. Although other Wi-Fi devices, *ran* and *urtwn* support faster standard IEEE 802.11n, they were not used in this experiment since the OS we introduced do not support host AP mode of these devices.

Table 2 shows the performance of our first implementation, thin FR by comparing the RTT (Round-Trip Time), download rate and upload one measured on Speedtest.net by Ookla three times. We believe that this experiment is enough for comparisons among communication devices since the bit rate in this measurement is not mere the average rate of one transfer but the average rate of multiple sampling data during each trial [17].

Table 2 The performance of implemented FR.

Type of Internet access line	Connected via	RTT (ms) Min./Max.	Download (Mbps): Max./Min.	Upload (Mbps): Max./Min.
UTP (G Ether)	Direct PC	22/31	17.63/8.87	52.22/11.75
	FR over UTP	26/31	15.12/8.74	10.97/7.92
3rd generation mobile communication	Direct PC as Wi-Fi tethering client	66/73	7.61/3.34	0.94/0.91
	FR on Wi-Fi / UTP between FR & PC	73/78	6.78/4.82	0.93/0.75
Satellite phone via UTP interface	Direct PC as PPPoE client	932/951	0.24/0.18	0.03/0.02
	FR over UTP	933/943	0.13/0.03	0.03/0.02
UTP (G Ether)	FR with NAPT & Wi-Fi WPA2	30/31	0.86/0.79	7.91/7.57
	FR over closed Wi-Fi WPA2	36/37	0.80/0.78	3.78/3.62
	FR over IWAS framework	38/38	0.76/0.69	3.46/2.91

RTT: Round-Trip Time, ms: millisecond, Mbps: megabits per second.

We show four types of comparison here. The first three comparisons show the effects of the overhead of FR on performance. The first pair labelled “UTP (G Ether)” shows the great decrease of the bandwidth especially in uploading direction because of the limited computing resource. Since the uploading transmission rate without FR is over the 10 % of the upper limit of USB 2.0, 480 Mbps (bit per second), the aggravation of performance is unavoidable in our implementation. We will directly show the throughput of the UTP device of the first prototype in the later experiment.

As we describe previous subsection 4.1, almost all we did to implement our system are start-up script for hardware configurations related to the modal of TNI and TNP. It mainly deals control plane of Software Defined Networking. Introducing hardware logic switching in forwarding plane may solve this low performance. Especially for VPN performance, we will show the improved result over the Internet later.

The second pair labelled “3rd generation mobile communication” is the experiment in which Wi-Fi tethering is possible. Compared with UTP high speed, aggravation of performance by FR is rather small. It seems computing resource is enough for this range of bandwidth.

In the experiment of the third pair labelled “Satellite phone via UTP interface,” we introduce commercial portable satellite phones in Japan that utilize a GSat (Geostationary Satellite). With a portable satellite phone, we can quickly connect to the Internet anywhere, since it is independent from the ground communication infrastructure. Even when a local telecommunication is unable to work under the influence of a natural disaster such as an earthquake, it is possible to connect the Internet through the satellite communication.

The communication over this satellite phone is long in latency since the GSat is in so high altitude above the Earth's equator that the slow speed of radio wave becomes a problem. It affects the low performance compared to the transfer rate of air interface, 384kbps for download and 144kbps for upload. Introducing special protocol for long latency [18] is desirable to solve this problem.

The last triple labelled “UTP (G Ether)” shows the small problem of the first implementation. The download speed is rather slow compared to the upload one in Wi-Fi Access Point mode. The

Table 3 The throughput of TNI on two FRs.

Type of FR	Connected via	Download Max./Min. [Mbps] (File size: in seconds Min./Max.)	Upload Max./Min. [Mbps] (File size: in seconds Min./Max.)
Thin FR on USL-5P	100Base-TX	18.713/18.634 (500MB: 224.13/225.08)	32.773/32.246 (500MB: 127.98/130.07)
	Wi-Fi WPA2 (IEEE802.11g)	0.741/0.689 (20MB: 226.19/243.24)	5.319/5.216 (100MB: 157.71/160.81)
Thick FR on Laptop PC	100Base-TX	93.910/93.848 (1GB: 91.47/91.53)	94.146/94.013 (1GB: 91.24/91.37)
	Wi-Fi WPA2 (IEEE802.11g)	15.809/14.779 (400MB: 212.24/227.03)	18.998/17.761 (400MB: 176.62/188.92)

B: byte, G for file size: 1024^3, M for file size: 1024^2 , M for throughput: 1000^2

load average of CPU during uploading over Wi-Fi was about 1.03 while the one during down loading was about 0.22. This means the limitation of computing resource might not be the root cause of this problem.

4.5 Improving FR performance

To conquer the last problem regarding the poor performance of Wi-Fi Access Point, we conducted an extended experiment just for testing throughput of each TNI devices. In order to check the performance of Wi-Fi Access Point mode, we used the identical Wi-Fi TNI (*rum* device) and wired / Wi-Fi client PC. Since we already found that the transfer rate of the first prototype, the thin FR, is lower than the upper limit of 100Base-TX Fast Ethernet using UTP, we utilized 100Base-TX hub for wired case to unify the experiment conditions between the thin FR case and the thick FR one.

In order to measure the throughputs, FTP (File Transfer Protocol) server program was used for FR side. Download throughput was measured as the transfer rate of getting a zero-filled file on the FR's file system from the FTP client program on client PC over FTP binary mode. Uploading throughput was measured by the similar way. The little different point is to measure throughput not by just putting a file but by putting it to null device of the FR since the slow file writing on the FR might affect the transfer rate. Since this type of measurement has little disturbance compared to one over the Internet, we adaptively change the file size for each combinations of FR and TNI. In each measurement, we tried FTP transfer ten times and show the best throughput and the worst one. Although the throughput in this measurement is the average transfer rate during each trial and will not show the peak rate of each device, we believe that it is enough to show the difference between two prototypes.

Table 3 shows the results in four patterns of combination of two types of FR and TNI devices. Although the throughput of the wired interface of the thin FR is limited, the one of the thick FR is close to the upper limit of the wired media 100Base-TX, 100Mbps. This means we can improve the performance of our implementation in accordance with the recent refinement of the prototyping in this case. It is the same in the wireless media interface. The uploading throughput is poor in the thin FR environment as we noticed in the first performance measurement while the uploading one is much improved in the thick FR environment compared to the one in the thick one using the identical Wi-Fi TNI.

In the similar scheme, we compared the VPN throughput by FTP transfer over the Internet intercontinentally, *i.e.* between Japan and Italy. Two types of VPN server were used in this

Table 4 The throughput of VPNs.

Peer connection	Compression / cipher	Direction	Max. / Min. (MB/S)	deviation
Via the Internet	None	Downward	581.47 / 499.78	24.8
		Upward	453.62 / 425.97	8.13
Over OpenVPN	None	Downward	143.00 / 111.02	8.93
		Upward	205.73 / 180.77	6.41
	LZO / BF-CBC	Downward	251.98 / 190.20	16.7
		Upward	210.42 / 185.25	8.21
Over L2TP/IPsec	None	Downward	462.29 / 323.42	47.7
		Upward	299.49 / 281.37	5.04
	MPPC / AES 256	Downward	786.16 / 569.92	84.0
		Upward	290.78 / 278.41	3.61

MB/S: Mega (1000^2) bytes per Second, LZO C. (compression): Lempel-Ziv-Oberhumer C., BF-CBC: Blowfish in Cipher Block Chaining mode, MPPC: Microsoft Point-to-Point C., AES: Advanced Encryption Standard.

experiment. One is the same environment of the first implementation, OpenVPN, introduced in section 4.1. Another is L2TP (Layer Two Tunneling Protocol) [19] with IPsec in Windows 7 *x64* architecture on Intel Core i7-960 running at 3.2GHz with 24GB of memory. The reason why Windows OS is introduced here is similar to the one introducing FreeBSD as the second prototype of FR. Since these VPNs support ciphering and software compression of traffics, we chose two patterns of operating mode for measurement. One is the mode without ciphering or compression to show the overhead of VPN in communication channel. Another is the mode with ciphering and compression to show the influence of computing resource. The throughput in this mode might be better than the one in normal transfer if enough computing resource can effectively compress zero-filled file.

Both these VPN servers were located in Japan and corresponding VPN clients were located in Italy. Thus we denote the direction from Italy to Japan by “upward” and denote the opposite direction by “downward” hereafter. In the measurements, the thin FR with *urtwn* device was used as a client of local Wi-Fi infrastructure to unify the condition of last mile for VPN client. We did not use *ram* device but *urtwn* device here because this device is used as WAN TNI. The VPN client machines were wired to the thin FR as LAN clients through 100Base-TX interface. The thick FR was used as OpenVPN client machine and another *x64* architecture PC running Windows 8.1 on Intel Atom CPU Z3735F running at 1.33GHz with 2GB of memory was used as L2TP client machine. Ordinary throughput between the two countries was also measured at the thin FR. In this measurement, we unified the size of transfer file to 50MB where M for file size is 1024^2. Since this type of measurement has large disturbance compared to one in local environment, we also calculated deviation of the throughputs of ten trials.

Table 4 shows the result in four patterns of two VPN systems with/without compression and ciphering including ordinary transfer via the Internet. From the experiment in laboratory environment, the reference [20] reported that IPsec was faster than OpenVPN. It is also true in our running environment since even the maximum throughput over OpenVPN is slow compared to the minimum one over L2TP in each direction. Introducing L2TP as VPN system will increase the performance of our prototyping in throughput.

The maximum throughput over L2TP in downward direction is higher than the one in ordinary transfer. It shows computing resource for compressing/ciphering on server side is enough for this throughput. Without compression / ciphering, L2TP overhead decreases the performance.

However, the upward throughput over L2TP is limited and is 61~70% of ordinary one in spite of compressed tunnel. The difference between L2TP with compress/cipher and one without them is vague in upward direction and the effect of compression is doubtful. It is also true on the throughput over OpenVPN. The difference between OpenVPN with compress/cipher and one without them is also vague in upward direction while the minimum throughput over compressed tunnel is higher than the maximum one over non-compressed tunnel in downward direction. The computing resource on client side should be improved. It also needs farther analysis to choose better options of VPN tuning for the next prototyping.

The deviations of the downward throughput tend to be large in the ordinal transfer and in the transfer over IPsec compared to the ones in the opposite direction and compared to the ones in slower transfers over OpenVPN. This means the downward traffic is unstable compared to the opposite direction. The route between VPN client-server pair in this measurement is only in high-speed research networks including SINET (Science Information NETwork) by NII (National Institute of Informatics, Japan), GÉANT by DANTE (Delivery of Advanced Network Technology to Europe), and GARR (Gruppo per l'Armonizzazione delle Reti della Ricerca; the Italian Academic and Research telecommunication network), Italy. The upward route and the downward one are almost the same. The hop count between the VPN client-server pair was 18 at maximum while the one from an Italian ISP (Internet Service Provider) to a Japanese ISP was over 25. This means the route in this measurement is straightforward compared to the commercial use of the Internet. Apart from the implementation of FR, farther research on quality of services in those networks is also needed to clear up the root cause and to find better solution.

5 Discussions

In the reference [21], they discussed the five basic properties of TUI. They are as follows:

1. space-multiplex both input and output;
2. concurrent access and manipulation of interface components;
3. strong specific devices;
4. spatially aware computational devices; and,
5. spatial re-configurability of devices.

In our approach, we utilize mainly the 3rd property to configure FR by TNIs. A specific TNI consists of an NIC device and a special default configuration. We also utilize the 4th property and 5th one to configure FR by TNP if we introduce the spatial relationship putting TNP between WAN TNI and LAN TNI. Our approach lacks both the 1st property and 2nd one. The 1st property is lacking simply because we omitted the output device for feedback as UI to notice the result of the manipulation electronically. The small monitor screen that visualizes the status of FR might help.

The 2nd property is rather hard to implement because of the conflict between the components of TUI and the components of FR. The former is mainly coupled to underlying digital data structure itself, although the latter is not coupled to flow data but the flow itself and the function, that is, the algorithm of computation. We need a specific UI suitable for a chain of processing

elements if we focus on the algorithm type components rather than data type components. The description of a pipeline of multiple programs running on Unix-like OSs might help.

The reference [22] is the system that incorporated TUI IP network simulator. In their system, users can interact with the simulated IP network topology through TUI manner. The components of their system are mainly IP nodes and links. So they are the data type component. They can control some parameters of node and that of links, including flow parameters, but they did not focus on the components of the algorithm, protocol stack, or the combination of device in one router as we proposed here.

CrowdShare [23] is the framework that enables sharing network resources including Internet access line among nearby devices over mobile mesh network. Their important aims are communication security and access control based on the relationships among users. The system was implemented on Android-based devices that have little external expansion capability and network configuration support for end user is not discussed. So our FR will help in introducing Internet access gateway to/from such a mobile mesh network or grassroots network trials. We believe it is easy to adopt our implementation of FR to a running routing protocol for mobile mesh networks over Wi-Fi ad hoc mode.

Balloons, helicopters, drones, or other types of flying objects with wireless modules can quickly serve access to the Internet. For example, Project Loon [24] uses high-altitude balloons as the transponders of LTE (Long Term Evolution) telecommunication network for ruined area. Our FR would act as a converter from LTE to Wi-Fi in the stricken area on the ground.

6 Conclusions

We proposed the notion of FR with TNI and TNP for configuring manually on site without external console device or software control panel by expert professionals to share seamless access to the Internet immediately among survivors using surviving access lines. By combining our FR with IWAS framework, we can avoid the abuse of open Wi-Fi network like server attacking anonymously. We have implemented the prototypes especially for accessing intranet servers located in an organization from stricken area. We also showed the solutions to improve the poor performances of the first prototype especially in Wi-Fi access point mode and VPN in a running environment.

The most of the descriptions of our idea in this paper is for a feasibility study in a closed environment to demonstrate what our idea is. As our next step, a large-scale social experiment for mitigating real natural disaster is required to support the implementation of our idea as described in this paper, for the benefit of communication among human-networks, whereas the Information Communication Technology in the world remains an open matter to be constantly improved and developed.

Acknowledgements

The author would like to thank the anonymous reviewers of MoMM2015 for their valuable comments and suggestions to improve the quality of our paper. The author is also grateful to Dr. Paolo Bellavista, Associate Professor of Dept. Computer Science and Engineering, Bologna University, Italy for his support in using his research environment including the Internet in Italy.

References

[1] Mori, N., Takahashi, T., and the 2011 Tohoku earthquake tsunami joint survey group 2012. “Nationwide post event survey and analysis of the 2011 Tohoku earthquake tsunami,” Coastal Engineering Journal, **54**, 1-27.

[2] Marco, C. 2007. “Multihop Ad Hoc Networking: The Theory,” IEEE Communications Magazine, **45** (4), 78-86. DOI= http://dx.doi.org/10.1109/MCOM.2007.343616.

[3] McMahon, A. and Farrell, S. 2009. “Delay- and Disruption-Tolerant Networking", IEEE Internet Computing, **13 (**6), 82-87. DOI= http://dx.doi.org/10.1109/MIC.2009.127.

[4] Ishii, H. and Ullmer, B. 1997. “Tangible bits: towards seamless interfaces between people, bits and atoms,” In *Proc. of the ACM SIGCHI Conference on Human factors in computing systems* CHI ‘97. ACM, New York, NY, 234-241. DOI= http://dx.doi.org/10.1145/258549.258715.

[5] Srisuresh, P. and Egevang, K. 2001. “Traditional IP Network Address Translator (Traditional NAT),” RFC (Request for Comments) 3022, IETF (Internet Engineering Task Force).

[6] Droms, R. 1997. “Dynamic Host Configuration Protocol,” RFC 2131, IETF.

[7] Cohen, R. and Kaempfer, G. 2000. “On the cost of virtual private networks,” IEEE/ACM Trans. on Networking, **8** (6), IEEE Press Piscataway, NJ, 775-784. DOI=http://dx.doi.org/10.1109/90.893873.

[8] Knight, P. and Lewis, C. 2004. “Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts,” IEEE Communications Magazine, **42** (6), IEEE, 124-131. DOI=http://dx.doi.org/10.1109/MCOM.2004.1304248.

[9] Jamaluddin, J., Doherty, M., Edwards, R., and Coulton, P. 2004. “A hybrid operating model for wireless hotspot businesses,” In *Proc. of First IEEE Consumer Communications and Networking Conference* (Las Vegas, NV, Jan. 05-08, 2004). CCNC 2004. IEEE, 611-615. DOI=http://dx.doi.org/10.1109/CCNC.2004.1286931.

[10] SoftBank Mobile Corp. 2011. “SoftBank Wi-Fi Spots Available Free of Charge,” http://mb.softbank.jp/en/news_information/20110312_02.html (accessible on 11th, Mar. 2016).

[11] Ogawa, H., Nishimura, T., and Maeda, T. 2003. “Integrated Wireless Authentication System,” Japanese Patent 4332000.

[12] Nishimura, T. 2010. “A Distributed Authentication Mechanism for Sharing an Overlay Network among Multiple Organizations,” In *Proc. of the 12th International Conference on Information Integration and Web-based Applications & Services* (Paris, France, Nov. 08-10, 2010). iiWAS ’10. ACM, New York, NY, 811-815. DOI=http://dx.doi.org/10.1145/1967486.1967622.

[13] Nishimura, T. 2011. “Distributed Authentication for Sharing an Overlay Network Among Multiple Organization,” In *Proc. of Innovations in Information and Communication Science and Technology* (Tomsk, Russia, Oct. 03-07, 2011). IICST2011. Publishing Office of Tomsk University of Control Systems and Radio Electronics, Tomsk, Russia, 69-78.

[14] Mamakos, L., Lidl, K., Evarts, J., Carrel, D., Simone, D., and Wheeler, R. 1999. “A Method for Transmitting PPP Over Ethernet (PPPoE),” RFC 2516, IETF.

[15] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., and Zorn, G. 1999. “Point-to-Point Tunneling Protocol (PPTP),” RFC 2637, IETF.

[16] Simpson, W. 1994. “The Point-to-Point Protocol (PPP),” RFC 1661, IETF.

[17] Ookla 2012. “How does the test itself work? How is the result calculated? ,” https://support.speedtest.net/hc/en-us/articles/203845400-How-does-the-test-itself-work-How-is-the-result-calculated- (accessible on 11th, Mar. 2016).

[18] Nishimura, T. and Ogawa, H. 2013. “Integrated Wireless Authentication System: Sharing Satellite Communication among Multiple Organizations after Natural Disasters,” In *Proc. of the 11th International Conference on Advances in Mobile Computing & Multimedia* (Vienna, Austria Dec. 02-04, 2013). MoMM2013. ACM, New York, NY, 270-277. DOI=http://dx.doi.org/10.1145/2536853.2536884.

[19] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and Palter, B. 1999. “Layer Two Tunneling Protocol "L2TP",” RFC 2661, IETF.

[20] Kotuliak, I., Rybár, P., and Trúchly, P. 2011. “Performance comparison of IPsec and TLS based VPN technologies,” In *Proc. 9th International Conference on Emerging eLearning Technologies and Applications* (Stara Lesna, Slovakia Oct. 27-28, 2011). ICETA 2011. IEEE, 217-221. DOI=http://dx.doi.org/10.1109/ICETA.2011.6112567.

[21] Kim, M. J. and Maher, M. L. 2008. “The impact of tangible user interfaces on spatial cognition during collaborative design,’’ Design Studies, **29**, 3, 222-253. DOI=http://dx.doi.org/10.1016/j.destud.2007.12.006.

[22] Kobayashi, K., Hirano, M., Narita, A., and Ishii, H. 2003. “A Tangible Interface for IP Network Simulation,’’ In *Proc. of CHI ‘03 Extended Abstracts on Human Factors in Computing Systems*, ACM, New York, NY, 800-801. DOI=http://dx.doi.org/10.1145/765891.766000.

[23] Asokan, N., Dmitrienko, A., Nagy, M., Reshetova, E., Sadeghi, A. R., Schneider, T., and Stelle, S. 2013. “CrowdShare: Secure Mobile Resource Sharing, ’’ Applied Cryptography and Network Security, Lecture Notes in Computer Science 7954, Springer-Verlag, Berlin / Heidelberg, Germany, 432-440. DOI=http://dx.doi.org/10.1007/978-3-642-38980-1_27.

[24] Google Inc. 2013. “Loon for All – Project Loon – Google,” https://www.google.com/loon/ (accessible on 11th, Mar. 2016).