

# 13<sup>th</sup> ICCRTS: C2 for Complex Endeavors

## Hastily-Formed Networks for First Responders

Topic 2: Networks and Networking

*Gurminder Singh*

Naval Postgraduate School  
Code CS/Rp, 1411 Cunningham Road  
Monterey CA 93943

(831) 656-3041  
gsingh@nps.edu

### Abstract

First responders who participate in humanitarian assistance and disaster relief missions have many special requirements which are not common in normal civilian operations. These include the ability to get going with their mission with minimal infrastructure, tight-loop and frequent communication, light-weight equipment, ability to scale-up the team when needed, and finally, the longest-running and lightest power source for their equipment. We present a system called TwiddleNet, which harnesses the power of mobile devices, primarily smart phones, to enable 1) instant content capture and publish, 2) full owner control of content, and 3) search, view and download of content. TwiddleNet is most useful for first-responder networking and information sharing tasks which require immediate content capture and dissemination. TwiddleNet can be scaled up or down depending on the needs of the mission. The entire system can be run on handheld devices to support a small first-responder team, or on a mix of handheld devices and server-class computers to link together a large number of smartphones sharing images, music, videos and mobile-blogs. TwiddleNet is designed to support the first 48-72 hours of first responder missions. As a result, TwiddleNet assumes little infrastructure and provides sufficient redundancy to operate on alternate mechanisms.

**Keywords:** handheld devices, smartphones, hastily-formed networks, first responders , humanitarian assistance, disaster response

### 1 Introduction

The humanitarian assistance and disaster relief missions are inherently distributed operations with many teams working on problems throughout an organization's area of responsibility. Even though such operations are centrally run, the work and the required information to accomplish the work are needed throughout the area assigned to the organization. A data dissemination architecture optimized for such operations can improve the speed of information flow from the teams to the headquarters and directly

## Report Documentation Page

*Form Approved  
OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE <b>JUN 2008</b>	2. REPORT TYPE	3. DATES COVERED <b>00-00-2008 to 00-00-2008</b>			
4. TITLE AND SUBTITLE <b>Hastily-Formed Networks for First Responders</b>		5a. CONTRACT NUMBER			
		5b. GRANT NUMBER			
		5c. PROGRAM ELEMENT NUMBER			
6. AUTHOR(S)		5d. PROJECT NUMBER			
		5e. TASK NUMBER			
		5f. WORK UNIT NUMBER			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Naval Postgraduate School, Code CS/Rp, 1411 Cunningham Road, Monterey, CA, 93943</b>		8. PERFORMING ORGANIZATION REPORT NUMBER			
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)			
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)			
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>13th International Command and Control Research and Technology Symposia (ICCRTS 2008), 17-19 Jun 2008, Seattle, WA</b>					
14. ABSTRACT <b>First responders who participate in humanitarian assistance and disaster relief missions have many special requirements which are not common in normal civilian operations. These include the ability to get going with their mission with minimal infrastructure tight-loop and frequent communication, light-weight equipment, ability to scale-up the team when needed, and finally, the longest-running and lightest power source for their equipment. We present a system called TwiddleNet, which harnesses the power of mobile devices, primarily smart phones, to enable 1) instant content capture and publish, 2) full owner control of content, and 3) search, view and download of content. TwiddleNet is most useful for first-responder networking and information sharing tasks which require immediate content capture and dissemination. TwiddleNet can be scaled up or down depending on the needs of the mission. The entire system can be run on handheld devices to support a small first-responder team, or on a mix of handheld devices and server-class computers to link together a large number of smartphones sharing images, music, videos and mobile-blogs. TwiddleNet is designed to support the first 48-72 hours of first responder missions. As a result, TwiddleNet assumes little infrastructure and provides sufficient redundancy to operate on alternate mechanisms.</b>					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>	<b>Same as Report (SAR)</b>	<b>24</b>	

from team to team. Receiving information faster can improve decision turn-around time, resource prioritization, and initiative on-site by team members.

We have developed a system called TwiddleNet, which gives an organization the ability to exploit the capabilities of the mobile devices that are in the best position to gather and distribute pertinent information at the right time. TwiddleNet harnesses the power of devices smart phones to enable 1) instant content capture and publish, 2) full owner control of content, and 3) search, view and download of content which was previously inaccessible. TwiddleNet is most useful for first-responder networking and information sharing tasks which require immediate content capture and dissemination. TwiddleNet can be scaled up or down depending on the need of the application. It can be run on a handheld device to support a small first-responder team and on large computers to link together millions of cell phones sharing images, music, videos and mobile-blogs.

These capabilities are tuned to the needs of first responders. TwiddleNet is designed to support the first 48-72 hours of the first responder mission after a disaster has struck. As a result, it assumes little infrastructure and provides sufficient redundancy to operate on alternate mechanisms. After this initial period, it is assumed that there would be better infrastructure and more support available.

Key first responder requirements and how TwiddleNet addresses them is as follows:

- **Quick Set-Up.** A key requirement of first responders, especially immediately after a disaster has struck, is to get going with their mission at the fastest speed possible. This means little time to set-up. The entire TwiddleNet system is designed to work with light-weight and battery-powered equipment. TwiddleNet fly-away kits include everything that the first responder team will need to get started with their mission.
- **Tight-loop, Frequent Communication.** An important task of first responders is to convey the ground reality to their co-workers and the control room. This needs to be done frequently and in real-time but without taking too much of their time and attention lest it start affecting their mission performance. To support this requirement, TwiddleNet smartphones produce tagged, xml-based, Atom feeds automatically on content capture. The entire process of generating and attaching tags to content is automated. Once content has been captured and tagged, it is automatically disseminated so that those who need to access it, can pull it and others who need to notified, content is pushed to them.
- **Light-Weight Equipment.** Due to the nature of their work, first responders' equipment needs to be as light as possible. TwiddleNet uses smartphones which are light-weight and small but still provide the redundancy that is so critical to the mission success. In addition, the TwiddleNet portal itself can be run on handheld devices (PDAs or small handheld computers) to further reduce the weight that the first responder team needs to carry.
- **Scale-up as Team/Requirements Grow.** TwiddleNet is focused on the first 48-72 hours after a disaster. It is designed to hand-off to more robust infrastructure (powerful servers) when it becomes available.
- **Battery.** Due to their short charge life and weight, batteries that operate the smartphones are an important issue. Often first responders have to carry spare

batteries, which increase the weight they have to carry. To address this issue, TwiddleNet pays special attention to power management - it supports smart caching of popular content provided owner consents. It allows the first responder to offload some of its serving functions to the TwiddleNet portal. In addition, to further conserve power, TwiddleNet supports several user-selectable content dissemination schemes.

## **2 Related Work**

The dramatic increase in the compute power and miniaturization of electronic components has led to small, light-weight, battery-powered devices which can perform a significant amount of computing for the user. Want et al (2002) envisioned such devices without traditional I/O capabilities to act as user's mobile personal server. When needed, the user could use the I/O capabilities of external devices such as desk/laptop computers to get access to information in the personal server through a Bluetooth connection. Intel's personal media server (2004) project extends Want et al's (2002) original personal server concept by making it phone-based and using it as a personal media server.

BluOnyx ([www.bluonyx.com](http://www.bluonyx.com)) is a commercial realization of Want et al's (2002) vision of a personal server. It is a cell phone-sized mobile content server product from Agere Systems (McGrath 2006). The system provides network connectivity with Bluetooth and WiFi, and enables users to store pictures, videos, music, emails and other files. BluOnyx does not have a screen but the resident content can be viewed/played on the cell phone, TV or PC screens. In its capability and form factor, it is what Want et al (2002) had envisioned a personal server to be.

Byland and Segall (2004) investigate the use of personal servers running on smartphones to support seamless mobility. They investigate requirements from not only computing but network and user interface perspectives as well. In their analysis, in addition to mobile servers, they explore the use of remote personal servers to provide support for seamless mobility.

Barton, Zhai and Cousins (2006) propose the use of cell phones equipped with large storage, in the range of a terabyte, to support mobile users in a variety of work as well entertainment tasks. Following Want et al (2002) and Byland and Segall (2004), they envision these devices to use and take advantage of the large displays of PCs and televisions.

The common thread among all of the above efforts is that they all recognize that the user can carry significant compute and storage capability; that multiple networking modalities, many more than available in PCs, have become available in the small devices; and, that when possible mobile devices should use comfortable displays and keyboards of PCs. None of these efforts exploits the content capture capability of smartphones the way TwiddleNet does. Instead of viewing smartphone as compute devices with limited screens and no keyboards, TwiddleNet views them as tools for content capture and dissemination. Another difference between TwiddleNet and previous efforts is that TwiddleNet creates a complete network among personal server devices rather than as stand-alone, personal servers only.

IBM's Infinity (Cheung et al, 2007) is a middleware framework for linking heterogeneous mobile devices. By doing so, it enables access to content resident on mobile devices. At a high-level the goals of TwiddleNet are similar to that of Infinity – both focus on enabling access to content on mobile devices. The most significant difference between the two systems is that TwiddleNet treats smartphones as personal servers and provides a portal front-end to enable content access whereas Infinity is middleware to develop general data sharing applications for smartphones. Infinity will be good toolkit to develop TwiddleNet.

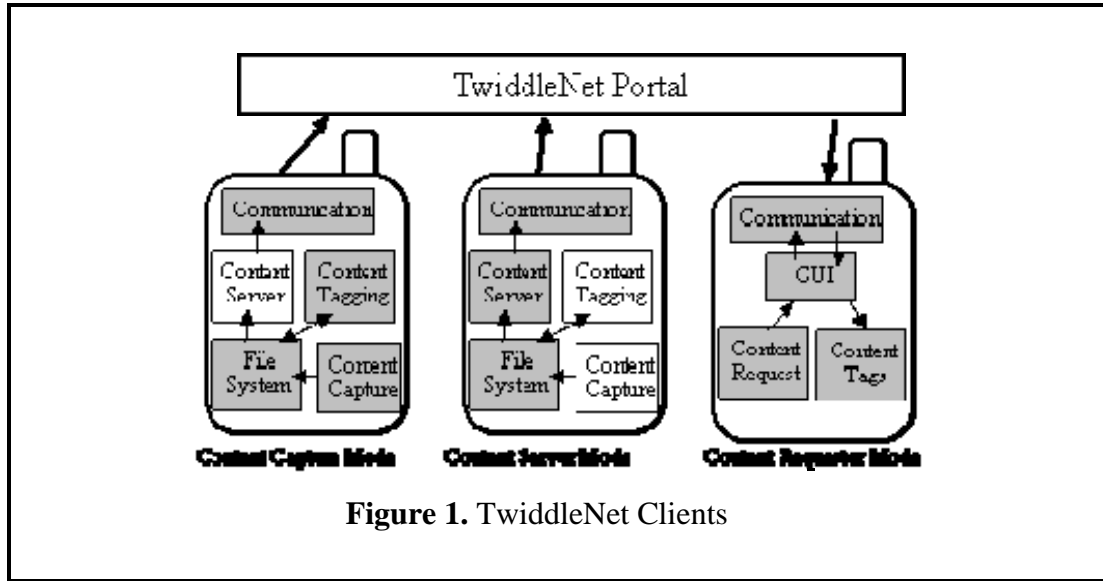
Q121 ([www.q121.com](http://www.q121.com)), a social networking and media-sharing service, allows people who register with the service to upload their favorite songs, videos, and photos to the site and then send them to the cell phones of other registered users (Roush 2006). In many respects Q121 is similar to other popular services such as MySpace ([www.myspace.com](http://www.myspace.com)) and FaceBook ([www.facebook.com](http://www.facebook.com)) which also allow people to upload their content and share with others with one significant difference – Q121 focuses on cell phones rather than everyday computers. While Q121 and TwiddleNet both focus on cell phones, TwiddleNet treats cell phones as personal content servers rather than content capture and upload devices for the web.

### **3 TwiddleNet Architecture**

In TwiddleNet, all client devices can work either personal content servers or as content requesters. They can switch roles rather transparently to the user. In the personal content server mode, the client device captures new content, tags it and lets the TwiddleNet portal know that the content is available. Content tags tell the portal, and through it the rest of user population, what is available on a particular personal server.

In the content requester mode, the client device gets updates from the TwiddleNet portal and request select content from the other users' client devices operating in the personal content server mode.

The TwiddleNet portal is a central repository for content created meta-data (or feeds) generated by clients. It allows for centralized searching for desired content as well as for sending alerts when a match occurs. The portal also acts as a cache for frequently accessed content.



In the following sections, we provide an overview of the significant modules of the TwiddleNet architecture.

## 4 Client

A typical TwiddleNet client is a smartphones but it is possible to use PDAs equipped with required networking and content capture capabilities as well.

### 4.1 Content Tagging in Client

The TwiddleNet client relies on an XML provisioning document to build its metadata files. This provisioning document contains all of the tags that could possibly appear when providing metadata about shared content. Each base tag has attributes that the application reads as a set of instructions detailing how the tag is to be handled, i.e. how a tag value is to be generated, when a tag value is to be generated and if a tag is mandated to appear in the finished metadata document. There are additional tags that facilitate processing, but the key tag attributes are how (either automatic or userdefined), when (corresponding to the lifetime stages of the entry: predefined, onGeneration, and onSending) and authority (either mandatory or optional). The combination of these attributes yields 12 possible tag types, however for purposes of the TwiddleNet application two of those types will never be provisioned (userdefined/mandatory/onGeneration and userdefined/mandatory /onSend).

The act of sending the document does require some finalization. Since TwiddleNet is designed for mobility, it must be sensitive to the networks it has access to and the networks it is using to send information. The TwiddleNet application waits to assign any network specific information until the last moment prior to releasing the completed XML file. This insures that the portal has the most up to date information regarding the whereabouts of the device and reduces to small degree future transmissions resulting

from dynamic ip address changes. The document is also lightened before it is sent; any application specific attributes are removed as are any tags that were selected for use but whose values remain empty. This insures that bandwidth use is maximized, and reduces power consumption accordingly.

The TwiddleNet client builds metadata entries according to the Atom syndication standard. As metadata files are transmitted, they are added to an Atom feed maintained by the device.

#### *4.2 Content Dissemination from Client*

TwiddleNet provides the user with five different options for when they can send their updates. These options include sending on a timed interval, content generation, delayed content generation, sensing a connection and manual. The timed interval sending option allows the user to send updates at user set timer intervals such as every hour or every ten hours. The feature is adjustable from 1 minute to 999 hours. This feature allows the user to collect content during the time interval without having to worry about sending them. It also allows for the collection of many pieces of content to be sent at one time vice being sent one at a time. The on-generation sending feature allows for automatic sending of files as soon as they are either created or added to the shared folder. This feature allows the user to automatically update the portal with information as soon as it is created. Although this is not ideal for power conservation it is ideal in situations where timeliness is an issue. Sending content on delayed generation is an attempt to gain the power saving benefits of sending many pieces of content at once while still maintaining the real-time benefits of the application. The feature activates a timer when content is generated. During this period if another piece of content is generated the timer is reset to its original value, this continues until the timer expires at which time the new contents are all sent together. This feature is ideal when the user takes pictures in clusters. Say for instance a user saw a site of interest and is going to take many pictures a minute or two apart, the delayed sending feature would allow for all of the pictures to be collected and their documents to be sent after the picture taking was complete. The sending whenever a new connection is sensed by the device (useful when you have spotty coverage), and manually sending.

### **5 TwiddleNet Portal**

Figure 2 shows the key components of the TwiddleNet portal. For a small first responder team, the TwiddleNet portal is able to run on a smartphone Windows Mobile 5.0. With this implementation, there is little set-up required as everything runs on handheld devices. When the size of the team grows, the portal can be moved to the standard server hardware. Apache 2.2.4 has been used on the standard server implementation of the portal, while the smartphone implementation is based on the vxWeb server by Cambridge Computer Corporation.

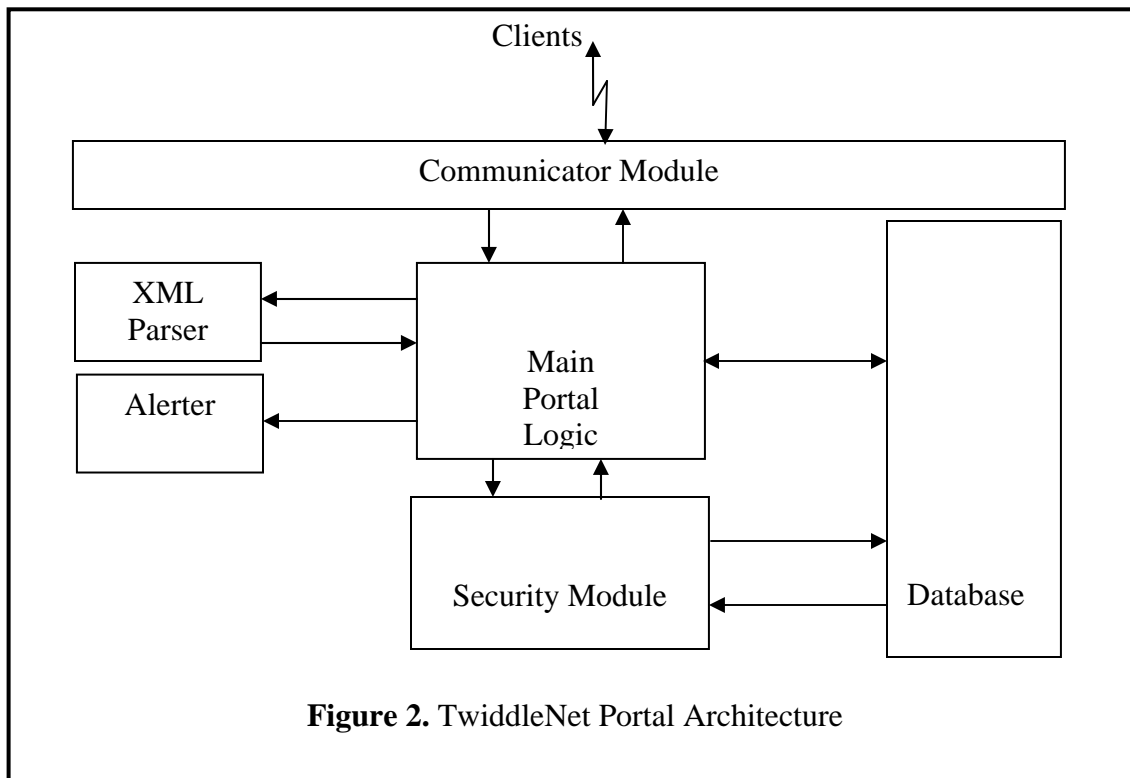
Portal allows for users to subscribe to desired content and receive alerts when new content, which was previously subscribed to, is created and made available. The web interface to the portal allows users to perform searches on all available content from all

registered TwiddleNet clients. The Portal supplies hyperlinks of desired content to TwiddleNet clients. A more robust Portal should allow for users to cache frequently requested data on the portal to ease demand placed on the small devices.

### 5.1 Communicator Module

The Communicator Module is responsible for providing the interface for the Main portal Logic to interact with the network(s). The current working implementation of this module is running as a CGI program through a web server. This requires the Communicator Module to extract all incoming client request data from the CGI interface implementation. With Apache server, the data is passed through operating system environment variables. The vxWeb server passes the CGI data through a file. The Communicator Module will extract the data and build up a “request” object that the Main Program can then use and retrieve the desired client information, i.e. perform a search or update content.

Future implementations of the TwiddleNet portal could be run as a standalone service. The only component that would be required to be changed/modified would be this module, the Communicator Module. This would provide a clean area for developers to implement their own protocol, if desired, while maintaining portability and ease of maintenance.



**Figure 2.** TwiddleNet Portal Architecture

## 5.2 XML Parser

This module handles the parsing and retrieving of all content meta-data to and from the Main Program. Current implementation makes use of the C# implementation of the XML Document Object Model and XPath to store and retrieve content meta-data. Any changes to how the content meta-data feeds are built or if more efficient means of handling the XML are desired, they can be easily implemented/modified here.

## 5.3 Alerter Module

The Alerter Module sole responsibility is to send an alert message of new content to the specified device. That device could be attached via any possible network interface: Wireless (802.11), Cellular network (GSM) or Landline (Ethernet). As long as the hardware device has the hardware and drivers to allow that system to communicate through it, code could easily be added to send any alert message to that device on that network interface.

For a proof of concept, the current implementation of the Alerter Module will send out an alert message through the wireless 802.11 to specified clients. Because the alert message may be sent through any possible network interface which may be size constrained, e.g. SMS, the message will contain a short, simple message stating that new content is available and the IP address where that content could be reached. Additional logic could be written into this module to make the alert message as detailed as desired depending on the subscriber location.

## 5.4 Database Connector

The Database Connector is responsible for storing user, content-meta data (feeds), and, potentially, cached content. The Database Connector will also handle the search queries that users submit to the Portal.

The current implementation uses Microsoft SQL Server Compact Edition. This provides a small footprint so that it can be run on a mobile device but also run on large servers. Any desired changes or modification to the search capabilities or data storage can be accomplished in this module with no changes required on other TwiddleNet Portal modules.

## 5.5 The Security Module

The Security Module is designed to handle all aspects of security. Current Portal version has the capability to handle registration of users, simple login and logout. In the future versions, we would like to use some kind of client state management, e.g., Cookies.

## 5.6 Main Portal Logic Module

The Main Portal Logic Module, as the name implies, handles the overall logic of the program. Depending on the requests received from the Communicator module, the Main Portal Logic performs a search, update content data, or login/logout. It also signals to the Alerter Module when and where to send an alert message.

## **6 TwiddleNet Fly-Away Kits**

A key feature of TwiddleNet is its ability to work with several different communication modalities as long as IP is supported. This enables us to provide a fail-safe system when needed, and if all network modalities are operational, some of the channels can be used for non-TwiddleNet communication, such as making voice calls, sending SMS etc. In order to deal with the eventuality that there is no Wireless WAN available, we have made a complete fly-away kit. This kit includes everything that a team needs in case absolutely no infrastructure exists at the scene of emergency. The kit includes several smartphones configured with TwiddleNet capability, a battery-powered BGAN (Broadband Global Area Network) unit, and a smartphone working as the TwiddleNet portal.

## **7 Current Status and Future Work**

We have developed an early prototype of TwiddleNet running on smartphones running Windows Mobile5.0. We have tested with TwiddleNet portal running on an iPAQ as well as on a standard PC server. We need to develop several additional modules as identified throughout this paper. We also need to develop TwiddleNet management tools so that user accounts and connections can be managed easily. Security is a very important concern for first responders. We need to devote significant effort to develop security policies for TwiddleNet and implement them.

We have done the first field test of TwiddleNet in Operation Golden Phoenix (OGP 2007) which simulated a 7.9 earthquake scenario in Los Angeles, California and had the participation of several agencies including LAPD, LAFD, LA Co. Sheriff, LA Co. Fire, California National Guard, MAG-46, NPS and SPAWAR. TwiddleNet was tested by soldiers from the US Marine Corps. Their overall feedback was very encouraging. In addition to ruggedizing the device, some of their other suggestions for improvements included a better user interface and more focus on privacy and security. Additional tests of TwiddleNet are planned as a part of the NPS COASTS field experiments to be held in January 2008, March 2008 and May 2008.

## **8 Conclusions**

First responders have to accomplish difficult tasks under high stress conditions. Technology must be built that allows them to focus on their task without burdening them with peculiarities of the communication devices. They need to communicate effectively, efficiently and frequently. This is often a critical component of their mission success. TwiddleNet aims to support their communication requirements without getting in the way of their work.

## 9 Acknowledgements

MAJ Jon Towle, CAPT Rob Myers, CAPT Eddie Zapata all of the US Marine Corps, and LT Chris Clotfelter of US Navy have contributed to the design and implementation of TwiddleNet.

## 10 References:

Markus Bylund and Zary Segall. Towards seamless mobility with personal servers. *Info – The Journal of policy, regulation and strategy for telecommunications*, 6(3): 172-179, 2004.

Trevor Pering. The Personal Server: Using a Commercially Available Cell-Phone As the Center of Your Personal Computing Experience, WMCSA 2006, *Proc. 7th IEEE Workshop on Mobile Computing Systems & Applications*, 2006, p. 48.

Darryl K. Taft (2006). Future Net: Expanding the Web from Pages to Data Sources. E-Week, Dec 7, 2006 <http://www.eweek.com/article2/0,1895,2069836,00.asp>

Alvin Cheung, Tyrone Grandison, Christopher Johnson, Stefan Schönauer, "Infinity: A Generic Platform for Application Development and Information Sharing on Mobile Devices", Sixth International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE) 2007, in conjunction with SIGMOD/PODS 2007, Beijing, China, June 2007.

Roush, Wade (2006). A New Platform for Social Computing: Cell Phones. *Technology Review*, July 3, 2006.  
[http://www.techreview.com/read\\_article.aspx?ch=specialsections&sc=telecom&id=17079](http://www.techreview.com/read_article.aspx?ch=specialsections&sc=telecom&id=17079)

Roy Want, Trevor Pering, Gunner Danneels, Muthu Kumar, Murali Sundar, and John Light. The Personal Server: Changing the Way We Think about Ubiquitous Computing. *Proc. Ubicomp 2002: 4th International Conference on Ubiquitous Computing*, Springer LNCS 2498, Goteborg, Sweden, Sept 30th-Oct 2nd, 2002, pp194-209

Barton, John J, Zhai, Shumin and Cousins, Steve B (2006) Mobile Phones Will Become The Primary Personal Computing Device. *Proc WMCSA 2006, Proc. 7th IEEE Workshop on Mobile Computing Systems & Applications*, 2006

Nottingham, M. and Sayre, R. (eds) (2005) The Atom Syndication Format. IETF RFC4287, December 2005

McGrath, Dylan (2006) Agere 'card' server streams content to devices. *EE Times*. Dec 18, 2006.  
(<http://www.eetimes.com/news/semi/showArticle.jhtml;jsessionid=4DSCXNJPLTOY YQSNLPSKH0CJUNN2JVN?articleID=196700349> accessed on October 22, 2007)

Intel Personal Media Server (2004)  
([http://www.intel.com/research/exploratory/personal\\_server.htm](http://www.intel.com/research/exploratory/personal_server.htm) accessed October 22, 2007)

# Hastily-Formed Networks for First Responders



Presented by: Neil Rowe  
NPS/CS

[gsingh@nps.edu](mailto:gsingh@nps.edu)

[ncrowe@nps.edu](mailto:ncrowe@nps.edu)

# Introduction

- HA/DR missions are inherently distributed operations
  - Though often centrally run, data collection and use for data is distributed
  - Real-time data collection, dissemination and use
- HA/DR missions impose many other demanding requirements
- We present a smartphone based system to handle the requirements

# First Responder Needs (1/3)

- Get going with the mission at fast speed
  - TwiddleNet fly-away kits will require little set-up
- Tight-loop, frequent communication with team members
  - Smartphones produce RSS feeds automatically on content capture
  - Tagging is automated
  - Dissemination is automated

# First Responder Needs (2/3)

- Equipment needs to be as light as possible
  - The entire system is smartphone-based!
- Scale-up as team/requirements grow
  - TwiddleNet is focused on the first 48-72 hours after a disaster.
  - Can hand-off to more robust (read heavy metal) infrastructure when it becomes available

# First Responder Needs (3/3)

- And, don't forget the battery...
  - We pay special attention to power management, e.g.:
    - Smart caching of popular content (owner's consent is required)
    - Send content periodically (rather than as it is captured)
    - Don't allow access when the battery is critically low
    - Etc.

# What is TwiddleNet?

- Gateway to Mobile Personal Servers (which are twiddling most of the time)
- Mobile personal servers run on smartphones
- Mobile personal servers host user's content – images, videos, audio, other real-time sensor data

# TwiddleNet – Key Observation

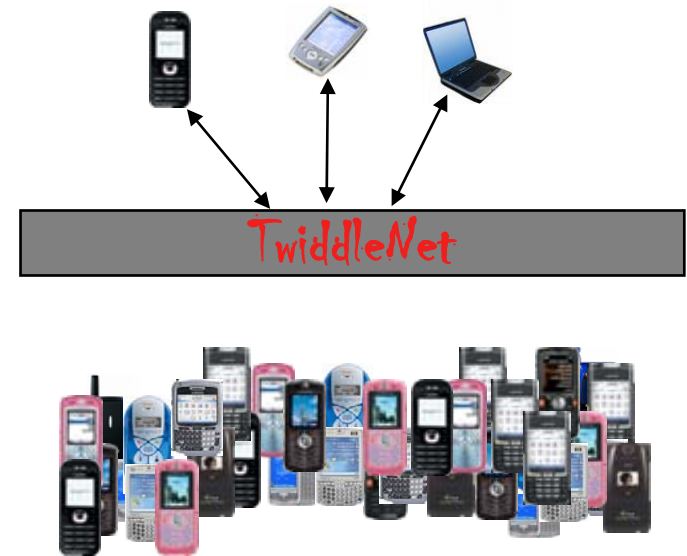
- Today's phones are more powerful than PCs of just a few years ago
  - Processors at 600MHz (1GHz coming!)
  - More than 100MB of RAM
  - More than 2GB of storage
  - And...
    - Wireless network – 2.5/3G, WiFi, Bluetooth etc.
    - Content capture capability – Photos, videos and sounds
- All this in a small form factor of a handheld device!!!

# Why TwiddleNet?

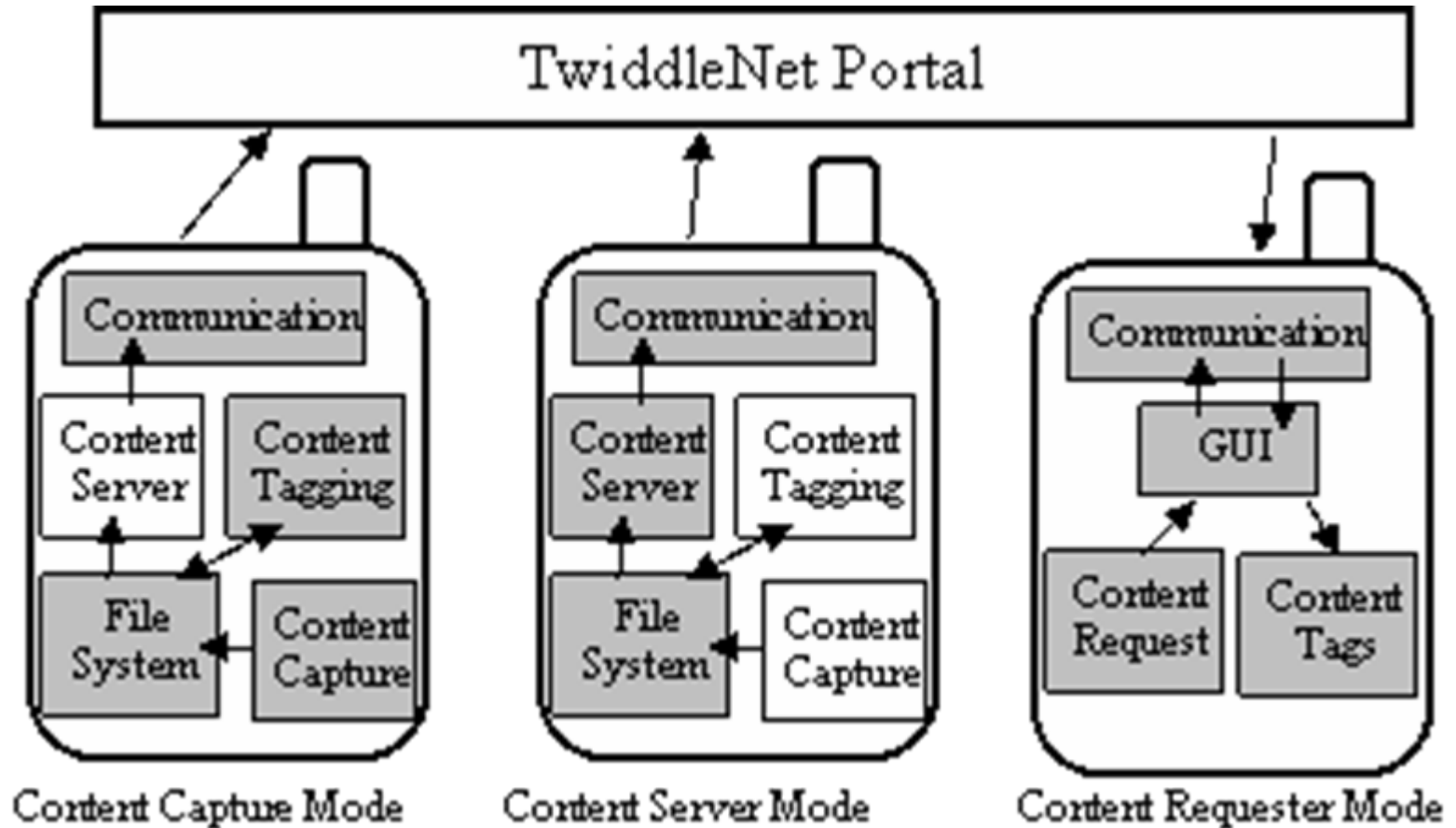
- Immediate content capture and publish
- Full owner control of content
- Harness the power of mobile devices twiddling most of the time
- Allow access to content which is otherwise inaccessible

# TwiddleNet Architecture

- Portal (but not a server) which is gateway to Mobile Personal Servers
- Allows search, viewing and download of content hosted on personal servers
- Content access statistics for smart caching
- Accessible from handhelds and desktops
- Match the end-user device capability



# Smartphone Modes in TwiddleNet



# TwiddleNet Applications

- First-responder hastily formed networks
- Social networking
- Other rapid content applications

# Current Status

- Tested an early version in Operation Golden Phoenix 07
- Planned test in COASTS Field Experiments in Ao Manao, Thailand – May '08

# Next Steps

- Continue R&D
  - Security
  - Availability
  - Content management
  - Scalability
  - Manageability
  - Server migration in the middle of operation
- Test in COASTS '08 – Thailand

Thank you.

[gsingh@nps.edu](mailto:gsingh@nps.edu)