

# Standard Implementation Framework for Opportunistic Networks in Emergency Preparedness and Response Applications

Leszek Lilien,<sup>\*</sup> *Senior Member, IEEE*, Ajay Gupta, *Senior Member, IEEE*, Zijiang Yang, *Member, IEEE*

Wireless Sensornet Laboratory (WiSe Lab), Department of Computer Science  
Western Michigan University, Kalamazoo, MI 49008-5466

**Abstract**—We present a novel paradigm of *opportunistic networks* or *oppnets* in the context of *Emergency Preparedness and Response (EPR)*. Oppnets constitute the category of ad hoc networks where diverse systems, *not* employed originally as nodes of an oppnet, join it dynamically in order to perform certain tasks they have been called to participate in. After describing the oppnets and their operation, we discuss the *Oppnet Virtual Machine (OVM)*—a standard implementation framework for oppnet applications. We also signal the critical oppnet privacy issues. Oppnets can significantly improve effectiveness and efficiency of EPR—one of the six mission areas within the national strategy for Homeland Security. They can also improve diverse other applications, including agriculture, environment, healthcare, manufacturing, surveillance, and transportation. Oppnets should create new application niches as yet hard to imagine. To the best of our knowledge we have been the first to define and explore oppnets.

## I. INTRODUCTION

Homeland security is perhaps the most crucial challenge facing the United States today. The “National Strategy for Homeland Security,” published by the Office of Homeland Security [NSHS02], identifies *Emergency Preparedness and Response (EPR)* as one of its six mission areas. The goal of EPR is stated as preparing “to minimize the damage and recover from any future terrorist attacks that may occur despite our best efforts at prevention. An effective response to a major terrorist incident—as well as a natural disaster—depends on being prepared.”

We propose a new paradigm and a new technology, called *opportunistic networks*, or *oppnets*, that can make these two EPR initiatives more effective and efficient. In particular, it can provide a wealth of modes of communication, sensing devices, and other tools to the first responders and victims.

Oppnets are a new broad category of application-driven computer networks. Defining a new subarea has many precedents in the computer network area, the object of active research for decades. During this time, investigators devised

many new categories of networks including wireless, ad hoc, mobile, and sensor networks [AgZe02, IyBr03].

To the best of our knowledge, opportunistic networks as defined by us are the network subarea not studied by others.<sup>†</sup> An earlier paper co-authored by one of us [BLRW04] was the first to define opportunistic *sensor* networks, a subclass of oppnets. This paper, after describing the oppnets, shows applicability of oppnets for EPR applications.

Oppnets differ from traditional networks, in which the nodes of a single network are all deployed together, with the size of the network and locations of its nodes pre-designed. In oppnets, the initial *seed oppnet* grows into an *expanded oppnet* by taking in foreign nodes. In other words, oppnets constitute the category of networks where diverse devices, *not* employed originally as its nodes, join the original set of “seed” oppnet nodes to help the oppnet realize its goals. We say that the new nodes become *helpers* for their oppnet.

Oppnets deployed for EPR can count on free help, which provides a tremendous leverage of the oppnet capabilities. This is the main reason why oppnets can have a huge impact in numerous application domains.

For EPR, oppnets have a significant potential for reduction of human suffering and loss of life in natural and man-made disasters, and for improving effectiveness and efficiency. For example, by enabling improved communications and monitoring of people and infrastructure, they can contribute to the safety and security of first responders and victims within possibly damaged elements of the infrastructure.

Oppnets will have a strong impact on domains other than EPR, both within Homeland Security applications, and outside. In addition to EPR, the former include: intelligence and warning, border and transportation security, domestic counterterrorism, protecting critical infrastructure, and defending against catastrophic terrorism [NSHS02]. The latter might include agriculture, environment, healthcare, manufacturing, surveillance, and transportation. Oppnets will lead network technology into new application niches as yet hard to imagine.

Oppnets inherit many capabilities and characteristics from ad hoc networks and P2P systems, in particular, node localiza-

This research was supported in part by the National Science Foundation under Grant IIS-0242840, and in part by the U.S. Department of Commerce under Grant BS123456. Any opinions, findings, conclusions or recommendations expressed in the paper are those of the authors and do not necessarily reflect the views of the funding agencies or institutions.

<sup>\*</sup> Affiliated with Center for Education and Research in Information Assurance and Security (CERIAS), Purdue University, West Lafayette, IN 47907.

<sup>†</sup> The name “opportunistic” is used for networks other than our oppnets. However, their “opportunism” is quite restricted, e.g., limited to opportunistic communication, realized when devices are within each other’s range. In contrast, our oppnets realize an opportunistic growth and an opportunistic use of resources acquired by the opportunistic growth.

tion and self-organization qualities from ad hoc networks, and growth-by-joining abilities from P2P systems. (For more details on relationship of oppnets to P2P see [LiKG06].)

We begin this paper by describing in the next section the basic oppnet operations. Section III discusses the proposed standard implementation framework for oppnets. Section IV summarizes the critical privacy problems in oppnets. Section V includes a brief overview of related work. Section VI concludes the paper and sketches plans for future work.

## II. BASIC OPPNET OPERATIONS

This section shows basic oppnet activities and applications.

### A. Seed Oppnets and Oppnet Helpers

*Seed Oppnets:* As a realization of the first requirement, each oppnet starts as a *seed oppnet*, i.e., a set of nodes employed together at the time of the

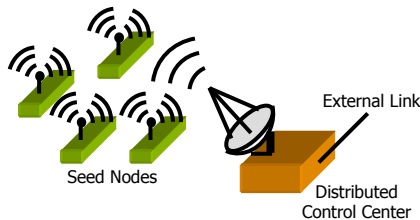


Figure 1. Seed oppnet.

initial network deployment (cf. Fig. 1). The seed is pre-designed (and can be viewed as a network in its own right). It might be very small, in the extreme consisting of a single node.

A subset of seed nodes constitutes a *distributed Control Center (CC)*. CC can grow by admitting other seed nodes or helpers, and shrink by expelling any of its nodes.

In addition to regular helpers, we can also have *lites* (i.e., “lightweight helpers” of limited capabilities). Helpers but not lites have the capability to discover and admit other helpers.

At any moment, a node belongs to only one of the four categories: (i) CC nodes; (ii) “seed nodes,” which really are the seed nodes that are not CC nodes; (iii) “helpers,” which really are the helpers that are not lightweight; and (iv) lites.

*Potential Helpers and Their Discovery:* In general, the set of *potential helpers* for oppnets is very broad, including communication, computing and sensor systems, both wired and wireless, both free-standing and embedded. Furthermore, as pervasive computing continues to progress, the pool of candidates will continue increasing dramatically: in infrastructures, buildings, vehicles, appliances, etc.

More densely populated areas will have, in general, a denser coverage by potential helpers. As a result, it will be easier to leverage capabilities of an oppnet in more densely populated areas. This is a desirable property, since more resources become available in areas with a possibility of more human victims and more property damage.

Before a seed oppnet can grow, it must discover its own set of *potential helpers* available to it. In addition to a mere lookup (of a previously prepared information, e.g., a directory), which is often referred to as “discovery,” we mean also much more challenging true discovery.

As an example of a true discovery, a PC can be discovered by an oppnet once the oppnet identifies a subset of IP ad-

resses located in its geographical area. Another example of true discovery could involve an oppnet node scanning the spectrum for radio signals or beacons, and collecting enough information to be able to contact their senders.

Discovery (including lookups) can be done by any means possible, both traditional and novel (cf. [GuAA05]). The former may include contacting via wired or wireless Internet, cellphone, Bluetooth, ham radio, microwave links, satellites, power grid, etc. The latter might exploit capabilities of Software Defined Radio [Abou06].

*Candidates, Helpers, and Utilizing Helpers:* Those of the potential helpers that are considered promising and are contacted by an oppnet, become its *candidate helpers* or *candidates*. Candidates admitted into an oppnet become its helpers.

Oppnets can utilize resources of helpers to significantly enhance their capabilities. This has the form of leveraging of all kinds of resources and “skills” (provided by smart or intelligent software) that new helpers bring with them. In this way oppnets obtain a lot of help effectively and efficiently (even for free in emergency situations as discussed later).

Oppnets are able to exploit *dormant capabilities* of their helpers. For instance, even entities with no obvious sensing capabilities can be used for sensing: (a) a desktop can “sense” its user’s presence at the keyboard; (b) a smart refrigerator (with an embedded processor) monitoring opening of its door can “sense” presence of potential victims at home in a disaster area. As another example, the water infrastructure sensor network with multisensor capabilities (and positioned near roads) can be directed to sense vehicular movement (or the lack thereof).

Use of helpers might include novel combinations of existing technologies, as illustrated by the following scenario. A surveillance system, serving as a helper, receives an image of an overturned car. The image is passed to a next helper that analyzes it to read the license plate. This information is used by another helper to check in a vehicle database if the car is equipped with a satellite communication system, e.g., OnStar™ [OnSt05]. If it is, the operator of the system can become a helper, and can contact the BANs (*body area networks*) or PANs (*personal area networks*) of car occupants.

### B. Growth of Seed Oppnet into Expanded Oppnet

A seed oppnet grows into an expanded oppnet after admitting new helpers. E.g., the expanded oppnet in Fig. 2 admitted the following helpers: (a) a computer network, contacted via a wired Internet link; (b) a cellphone infrastructure (represented by the cellphone tower), contacted via oppnet’s cellphone peripheral; (c) a satellite, contacted via a direct satellite link; (d) a *home area network*, contacted via an intelligent appliance (e.g., a refrigerator) with a wireless link; (e) a microwave network, contacted via a microwave relay; (f) BANs of occupants of an overturned car, contacted via OnStar.

Helpers are either *invited* or *ordered* to join. In the former case, contacted candidates can either volunteer or refuse the invitation. In the latter case, they must accept being conscripted in the spirit of citizens called to arms (or suffer the consequences of going AWOL).

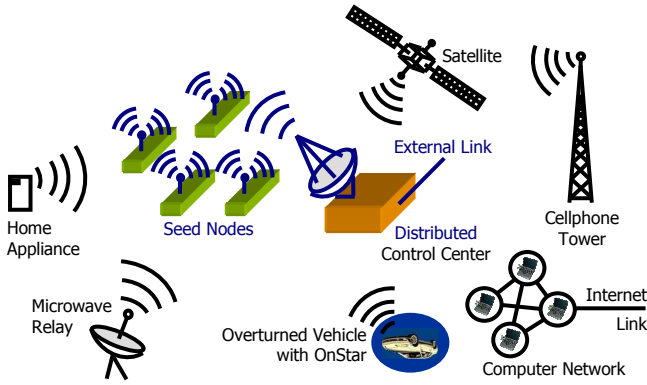


Figure 2. Expanded oppnet.

C. Asking or Ordering Helpers and Oppnet Reserve

The issue of ordering candidate helpers may seem controversial, and requires addressing. First, it is obvious that *any* candidate can be *asked* to join in any situation.

Second, *any* candidate can be *ordered* to join in *life-or-death* situations. It is an analogy to citizens being required by law to assist with their property (e.g., vehicles) and their labor in saving lives or critical resources.

Third, *some* candidates can *always* be *ordered* to become helpers in emergencies. Such helpers include many kinds of computing and communication systems serving police, firemen, National Guard, and military. Also the federal and local governments can make some of their systems available in emergencies upon order of an EPR oppnet.

The category of systems always available on an order of an EPR oppnet includes systems that volunteer—actually, “are volunteered” by their owners. In an obvious analogy to Army, Air Force, and other Reserves, they all can be named collectively as the *oppnet reserve*. Individually they are *oppnet reservists*. As in the case in the human reserves, volunteers sign up for oppnet reserve for some incentives, be they financial, moral, etc. Once they sign up, they are “trained” for an active duty: facilities assisting oppnets in their discovery and contacting them are installed on them. For example, a standard Oppnet Virtual Machine (OVM) software, matched to their capabilities—either heavy-, medium- or lightweight—is installed on them. (OVM is discussed in Section IV.) The “training” makes candidates highly prepared for their oppnet duties.

Oppnet reserve is not necessary for the oppnet paradigm but very helpful for at least two reasons. First, presence of oppnet reservists in an incident area increases the pool of candidates that can be ordered—rather than asked—by an oppnet to join it. Second, having “trained” reservists (e.g., OVM-equipped ones) significantly simplifies discovery of candidates. Specifically, it facilitates finding the very first oppnet’s contact in an incident area, which is always most difficult. Once a reservist reports for duty or is called to duty by an oppnet, reservists’ own contacts become easy next-wave contacts for the oppnet.

The above discussion assumes that at least one reservist survives an incident. With numerous reservists in practically every area of the country—the more reservists the more densely populated is an incident area—we are practically guaranteed that some reservists *will* survive. (In the same way, at least some of the reservists’ own contacts will survive.)

By employing helpers working for free (as volunteers or conscripts), opportunistic networks can be extremely competitive economically in their operation. Full realization of this crucial property requires determining the most appropriate incentives for volunteers and enforcements for conscripts.

III. OPPNET VIRTUAL MACHINE

In order to facilitate the implementation of oppnet applications, we are developing the standard implementation framework for oppnets, named *Oppnet Virtual Machine (OVM)*. By following this standard, implementations from different oppnet vendors will become interoperable.

A. OVM Primitives

The OVM primitives are intended for use by all those who want to write programs in C/Java/C++/C# for oppnet seeds or oppnet helpers. This includes individual application programmers, manufactures of hardware devices, and creators of environments and tools. Therefore, as a part of our research on oppnets we plan to achieve the following of hierarchy of goals:

- Design an application programming interface
  - Language-independent semantics of the interface
  - Convenient C/Java/C++/C# interface bindings
  - Extensions allowing greater flexibility
  - Implemented on many vendors’ platforms
  - Usable in heterogeneous environments
- Allow efficient communication
  - Uniform data/message formats

TABLE I: Partial List of OVM Primitives for Control Nodes

Name of the Primitive	Functions of the Primitive
CTRL_initiate	Initiate oppnet
CTRL_terminate	Terminate oppnet
CTRL_command	Send command to seed nodes

TABLE II: Partial List of OVM Primitives for Seed Nodes

Name of the Primitive	Functions of the Primitive
SEED_scan	Scan communication spectrum to detect devices that could become candidate helpers
SEED_discover	Discover candidate helpers with a specific communication mechanism
SEED_listen	Receive and save messages in buffer
SEED_validate	Verify the received command
SEED_isMember	Checks if a device is already an oppnet node (oppnet member)
SEED_evaluateAdmit	Evaluate a device and admit it into oppnet if the device meets criteria for admittance
SEED_sendTask	Send a task to other oppnet device
SEED_delegateTask	Delegate a task that requires a permission from the delegating entity

SEED_release	Release a helper when no longer needed
SEED_processMessage	Process a message from buffer
SEED_report	Report information to control center/coordinator

TABLE III: Partial List of OVM Primitives for Helpers

Name of the Primitive	Functions of the Primitive
HLPR_isMember	Test if a helper is already a member of oppnet
HLPR_joinOppnet	Join oppnet
HLPR_scan	Scan communication spectrum to detect devices that could become candidate helpers (regular or lites)
HLPR_discover	Discover candidate helpers with a specified communication mechanism
HLPR_validate	Verify the received command
HLPR_switchMode	Switch between helpers' regular application and oppnet application
HLPR_report	Send information/data to specified node
HLPR_selectTask	Select a task from the task queue to execute
HLPR_listen	Receive message and save it
HLPR_evaluateAdmit	Evaluate a candidate helper and admit it into oppnet if it meets criteria defined by oppnet
HLPR_runApplication	Execute application indicated by authorized oppnet seed or helper node
HLPR_release	Release a helper (unless delegated a release task, a helper H can release only helpers admitted by H)
HLPR_processMessage	Process a message from buffer
HLPR_sendData	Send information/data to specified authorized oppnet node
HLPR_leaveOppnet	Leave oppnet when released

Recall that at any moment a node belongs to only one of the four categories: CC nodes, seed nodes, helpers, and lites. Also, recall that lites are leaves in the oppnet node hierarchy that are unable to discover more helpers or lites.

Tables I, II and III show partial lists of the primitives offered by OVM for the oppnet's CC nodes, seed nodes and helpers. The OVM primitives for these classes of nodes have prefixes CTRL\_, SEED\_ and HLPR\_. The primitives for lites have prefix "LITE\_" and include all the primitives from Table III except HLPR\_scan, HLPR\_discover, HLPR\_evaluateAdmit and HLPR\_releaseHelper.

Defining separate primitives for the four node classes facilitates preventing situations when a node attempts to play a role of a node from another node class. There are two main advantages of having distinct primitive classes:

- Better security. Seed nodes have higher clearance level than helpers, which in turn have higher clearance level than lites. (Within each class, clearance sublevels can be defined.) Extra class-based layers in security mechanisms facilitate addressing security concerns more efficiently.
- Resource savings. Most helpers and lites have quite limited resources. By knowing the limitations of the roles they can play, we can install on them only the relevant *partial* virtual machines. For example, a lite will not be burdened with the tasks of discovering other helpers or lites, thus eliminating the need to install on it OVM components needed for scanning, discovery, etc.

We are working on the primitives, defining their arguments, messages, etc. We plan to bind the primitives with different programming languages and implement OVM libraries. Development of the primitives follows the model of PVM [SGDM94] and MPI [GrLS94] used in grid computing.

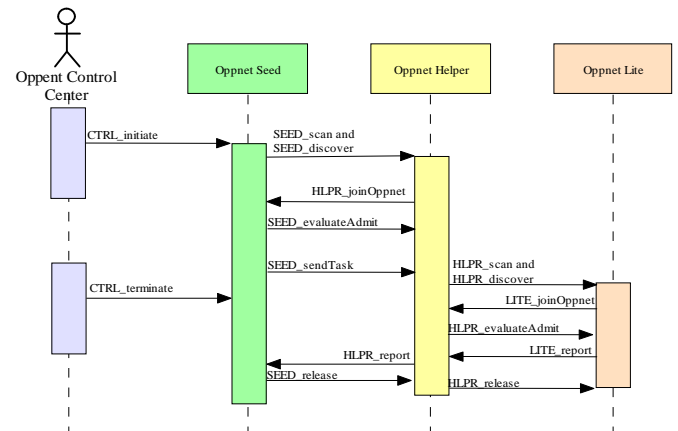


Figure 3. Sequence chart of an example oppnet application scenario. Messages are labeled with the names of primitives sending them. Reliable message delivery mechanisms are assumed.

### B. Example Oppnet Application Scenario

The following simple scenario illustrates an oppnet application. In a natural disaster area, one priority is to find survivors caged in houses and cut off by earthquake, hurricane, or flooding. After the oppnet seed is deployed, the oppnet will expand by admitting helpers and lites. Among others, the motion sensors embedded in Bluetooth-equipped smoke detectors will become lites. If any such lite detects any movement, data will be transmitted to oppnet coordinators.

The sequence chart of such a scenario is displayed in Fig. 3. It shows how seed nodes obtain information from a lite via a helper. (The lite runs a small motion detection application.)

```

repeat on command received from control center or
other authorized device
  SEED_validate(command);
  switch (command)
  case "scan":
    SEED_scan(...);
  case "BT (Bluetooth) discover":
    SEED_discover(BT,...);
    SEED_listen(...);
    for each responding BT device D do
      if (not SEED_isMember(D,...))
        SEED_evaluateAdmit(D,...);
    if need more BT helpers
      for each H in subset of regular
        helpers do
          SEED_delegateTask(H,
            "get BT helpers",...);
  case "send tasks":
    for each H in subset of helpers do
      SEED_sendTask(H, command,...);
  case "report":
    for each message M in buffer do
      SEED_processMessage(M);
    SEED_report(...);
  ...
end_switch
end_repeat

```

Figure 4. Pseudocode for seed nodes.

```

repeat on command received from control center or
other authorized device
  HLPR_validate(command);
  switch (command)
  case "join oppnet":
    HLPR_switchMode(...);
    HLPR_joinOppnet(...);
  case "detect motion":
    HLPR_runApplication(motion,...);
    HLPR_sendData(...);
  case "get BT (Bluetooth) helpers":
    HLPR_scan(BT,...);
    HLPR_discover(BT,...);
    HLPR_listen(...);
    for each responding BT device D do
      if (not HLPR_isMember(D,...))
        HLPR_evaluateAdmit(D,...);
        HLPR_report(...,BT, D);
      end_if
  case "report":
    for each message M in buffer do
      HLPR_processMessage(M,...);
    HLPR_report(...);
  ...
  case "leave oppnet":
    HLPR_leaveOppnet(...);
    HLPR_switchMode(...);
  end_switch
end_repeat

```

Figure 5. Pseudocode for helpers.

The pseudocodes for Oppnet Seed, Oppnet Helper and Oppnet Lite from Fig. 3 are shown in Figures 4, 5 and 6, respectively. Oppnet Control Center nodes do not run an autonomous algorithm but respond to human commands. Therefore, it is not necessary to have pseudocode for them. (Due to space constraints, the pseudocodes have been significantly simplified to serve illustrative purposes only.)

The nodes of an expanded oppnet—incl. seed nodes, helpers, and lites—keep listening to commands from the oppnet’s (distributed) CC or other authorized nodes (e.g., a helper can

accept tasks from another helper). When a command  $C$  is received by a node, the node first verifies  $C$ . The verification may include (1) checking sender’s access rights; (2) checking security level of  $C$ ; (3) estimating resources needed to carry out  $C$ . The command will be executed if it passes the checks.

```

repeat on command received from control center or
other authorized device
  LITE_validate(command);
  switch (command)
  case "join oppnet":
    LITE_switchMode(...);
    LITE_joinOppnet(...);
  case "detect motion":
    LITE_runApplication(motion,...);
    LITE_sendData(...);
  case "report":
    for each message M in buffer do
      LITE_processMessage(M,...);
    LITE_report(...);
  ...
  case "leave oppnet":
    LITE_leaveOppnet(...);
    LITE_switchMode(...);
  end_switch
end_repeat

```

Figure 6. Pseudocode for lites.

Oppnet helpers and lites perform their daily activities until they are called upon to join an oppnet, in which case they switch to the defined emergency mode. When a command is received by a node, the node will also verify if it has the ability to execute the task. E.g., a command that requires connection via Wi-Fi cannot be run by a device with a Bluetooth connectivity only.

#### IV. OPPNET PRIVACY ISSUES

Due to space limitations, we can only signal privacy issues, and refer the reader to our other paper which discusses privacy and security issues for oppnets and proposes initial privacy and security solutions [LKBG06].

First, we need to note that *Pervasive Computing*, one of the ultimate goals of Information Technology, will surround people with countless computing devices of all kinds, sizes, and capabilities. Pervasive systems collectively are able to spy anywhere, anytime, on anybody and anything in their midst.

Second, since oppnets are a Pervasive Computing technology, the same privacy problems apply to them.

Third, without adequate privacy protections, the public will justifiably revolt against any pervasive systems, oppnets included. Privacy protection is the “make it or break it” issue for any pervasive computing technology [LiBh06].

Fourth, we strive to minimize intrusiveness of oppnets towards helpers in order to reduce privacy threats for helpers. Actually, our basic privacy protection goals in oppnets include not only protecting a helper from its oppnet, but also protecting an oppnet from its helpers, and protecting the environment privacy from oppnets (also from malevolent oppnets).

Fifth, controls are needed against malicious oppnet nodes or whole oppnets that can be deployed by adversaries. They could dupe honest helpers into joining them, and use them for

their covert malevolent goals. E.g., an attacker can create an apparently harmless weather monitoring sensornet, which actually is a tool for planning a chemical attack.

## V. RELATED WORK

Related work can be classified into research on EPR systems and work on system R&D, relevant for EPR systems.

### A. Related Work in the EPR System Dimension

We believe that many alternative approaches need be explored, evaluated, and have their best features extracted to provide a winning solutions. EMISARI [KuWi72, TCVY04] was the first group communication oriented crisis management system successfully since 1971 until the mid eighties.

Current system research projects devoted to EPR include Wireless Mesh Networks [BTMM06], MIKoBOS [MWPG06], a “ubiquitous mobile office” for disaster mitigation [GASD06], knowledge management systems [Otim06, MuJe06], and a wide array of work on multi-agent systems for EPR (e.g., [JRAD06, VLSD06, TaRP06]).

Industry solutions for EPR have been proposed by, e.g., IBM [IBM\_05] and Motorola [Moto05]. We already see examples of large-scale applications of these products by municipalities, including NYC [Menc06].

### B. Related Work in the System R&D Dimension

Due to space limitation, we don’t give any references here. Oppnets can adaptations localization and self-organization techniques from ad hoc networks (esp. from MANETs), growth-by-joining approaches from P2P systems (incl. searching for peers in unstructured systems), data aggregation algorithms from sensornets, and resource integration and management methods from grids. Useful results come also from research on interoperability in heterogeneous environments, and work on benevolent Trojans (for agents in search of helpers).

## VI. CONCLUSIONS AND FUTURE WORK

*Opportunistic networks* or *oppnets* presented by us are a new, specialized type of ad hoc networks. Oppnets provide an unprecedented leveraging potential for growing from a small seed network into a very powerful network with vast communication, computing, sensing and other capabilities.

We have identified many challenges for our future oppnet research [LiKG06]. We continue our investigation of oppnets, and designing oppnet architectures with their associated components: methods, protocols, and algorithms. The planned prototype oppnet will provide a proof of concept, as well as stimulation and a feedback necessary for fine-tuning oppnet architectures and their components.

## REFERENCES

- [Abou06] “About SDR Technology,” Software Defined Radio Forum, 2006.
- [AgZe02] D. P. Agrawal and Q. Zeng, *Introduction to Wireless and Mobile Systems*, Brooks/Cole Thompson Learning, 2002.
- [BLRW04] B. Bhargava, L. Lilien, A. Rosenthal, and M. Winslett, “Pervasive Trust,” *IEEE Intelligent Systems*, Sep./Oct. 2004, pp.74-77.
- [BTMM06] B. Braunstein, T. Trimble, R. Mishra, B.S. Manoj, L. Lenert, and R.R. Rao, “Challenges in Using Distributed Wireless Mesh Networks in Emergency Response,” *Intl. Conf. on Information Systems for Crisis Response and Management (ISCRAM 2006)*, Newark, NJ, May 2006.
- [GASD06] P. Gomez Bello, I. Aedo, F. Sainz, P. Diaz, and J. de Castro, “m-ARCE: Designing a Ubiquitous Mobile Office for Disaster Mitigation, Services and Configuration,” *Intl. Conf. on Information Systems for Crisis Response and Management (ISCRAM)*, Newark, NJ, May 2006.
- [GrLS94] W. Gropp, E. Lusk, and A. Skjellum. “Using MPI: portable parallel programming with the message-passing-interface,” *MIT Press*, 1994.
- [GuAA05] A. Gupta, D. Agrawal, and A.E. Abbadi, “Distributed Resource Discovery in Large Scale Computing,” *Intl. Symp. on Applications and the Internet (SAINT 2005)*, Trento, Italy, Jan.-Feb. 2005.
- [IBM\_05] IBM, “First Responder Interoperability Solution (FRIS),” 2005.
- [IyBr03] S. Iyenger and R. Brooks, *Distributed Sensor Networks*, CRC Press, 2003.
- [JRAD06] N.R. Jennings, S.D. Ramchurn, M. Allen-Williams, R. Dash, P. Dutta, A. Rogers, and I. Vetsikas, “The ALADDIN Project: Agent technology to the rescue,” *First Intl. Workshop on Agent Technology for Disaster Management (ATDM 2006)*, Hakodate, Japan, May 2006.
- [KuWi72] R. Kupperman and R. Wilcox, “EMISARI - An On line Management System in a Dynamic Environment,” *1st International Conference on Computer Communications*, 1972, IEEE, 117-120.
- [LiBh06] L. Lilien and B. Bhargava, “A Scheme for Privacy-preserving Data Dissemination,” *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 36(3), May 2006, pp. 503-506.
- [LiKG06] L. Lilien, Z. H. Kamal, and A. Gupta, “Opportunistic Networks: Research Challenges in Specializing the P2P Paradigm,” *Proc. 3rd International Workshop on P2P Data Management, Security and Trust (PDMST’06)*, Kraków, Poland, Sept. 2006.
- [LKBG06] L. Lilien, Z.H. Kamal, V. Bhuse, and A. Gupta, “Opportunistic Networks: The Concept and Research Challenges in Privacy and Security,” *Proc. Intl. Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN)*, Miami, FL, March 2006.
- [Menc06] G. Menchini, “Citywide IT Preparedness for Critical Events: Accomplishments and Challenges,” keynote talk, *3rd Intl. Conf. on Information Systems for Crisis Response and Management (ISCRAM 2006)*, Newark, NJ, May 2006.
- [Moto05] Motorola, “Mesh Enabled Architecture (MEA®) Solutions for Emergency Response Agencies,” 2005. Available at: [http://www.motorola.com/mesh/pdf/wp\\_mea\\_emergency\\_response.pdf](http://www.motorola.com/mesh/pdf/wp_mea_emergency_response.pdf)
- [MuJe06] T. Murphy and M.E. Jennex, “Knowledge Management Systems Developed for Hurricane Katrina Response,” *Intl. Conf. on Info Syst. for Crisis Response and Mgmt (ISCRAM)*, Newark, NJ, May 2006.
- [MWPG06] A. Meissner, Z. Wang, W. Putz, and J. Grimmer, “MIKoBOS – A Mobile Information and Communication System for Emergency Response,” *Intl. Conf. on Information Systems for Crisis Response and Management (ISCRAM 2006)*, Newark, NJ, May 2006.
- [NSHS02] *National Strategy for Homeland Security*, Office of Homeland Security, July 2002. At: <http://www.whitehouse.gov/homeland/book/>
- [OnSt05] OnStar Corp., “On Star Explained,” 2006.
- [Otim06] S. Otim, “A Case-Based Knowledge Management System for Disaster Management: Fundamental Concepts,” *Intl. Conf. on Info Syst. for Crisis Response and Mgmt (ISCRAM)*, Newark, NJ, May 2006.
- [SGDM94] V. Sunderam, G. Geist, J. Dongarra, and R. Manchek. “The PVM concurrent computing system: Evolution, experiences, and trends,” *Parallel Computing*, Vol. 20(4), April 1994, pp 531-547.
- [TaRP06] B. Tatomir, L. Rothkrantz, and M. Popa, “Intelligent Systems for Exploring Dynamic Crisis Environments,” *Intl. Conf. on Info Syst. for Crisis Response and Mgmt (ISCRAM 2006)*, Newark, NJ, May 2006.
- [TCVY04] M. Turoff, M. Chumer, B. Van de Walle, and X. Yao, “The Design of a Dynamic Emergency Response Management Information System (DERMIS),” *J. of Information Technology Theory and Application (JITTA)*, Vol. 5(4), Summer 2004, pp. 1-36. A version available at: <http://web.njit.edu/~turoff/Papers/dermis2004.htm>
- [VLSD06] J.R. Velasco, M.A. López-Carmona, M. Sedano, M. Garajo, D. Larrabeiti, and M. Calderón, “Role of Multiagent system on minimalist infrastructure for service provisioning in ad-hoc networks for emergencies,” *First Intl. Workshop on Agent Technology for Disaster Management (ATDM 2006)*, Hakodate, Japan, May 2006.