Opportunistic Networks for Emergency Applications and Their Standard Implementation Framework

Leszek Lilien,* Ajay Gupta, and Zijiang Yang
WiSe (Wireless Sensornet) Laboratory, Department of Computer Science
Western Michigan University, Kalamazoo, MI 49008-5466, USA
{llilien, gupta, zijiang}@cs.wmich.edu

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. 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 other diverse 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 work on oppnets.*

1. Introduction

Homeland security is perhaps the most crucial challenge facing the United States today. The "National Strategy for Homeland Security," published in 2002 by the Office of Homeland Security, 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 by providing a wealth of communication modes, sensing devices, and other tools.

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. Many new categories of networks devised during this time include wireless, ad hoc, mobile, and sensor networks.

To the best of our knowledge, opportunistic networks defined by us are the subarea of networks not studied by others.[1] An earlier paper co-authored by one of us [1] was the first to define opportunistic *sensor* networks, a subclass of oppnets. This paper, after describing the oppnets, discusses a standard implementation framework for oppnet 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 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

* Affiliated with the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University.

[1] The term "opportunistic" is used for other networks. Their "class 1 opportunism" is quite restricted, e.g., limited to opportunistic communication when devices are within each other's range. In contrast, our "class 2 opportunism" relies on an opportunistic growth and opportunistic use of resources gained by this growth.

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. 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 localization 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 [8].) Due to space limitations, we must forgo any discussion of work on general system research relevant to oppnets or to EPR systems.

The same limitations force us to just mention related research on EPR systems themselves. Many alternative approaches need be explored, evaluated, and have their best features extracted to provide winning solutions. Proposals for EPR systems range from the classic ones [6] to current research projects [17, 2, 12, 3, 14, 16], and to industry solutions [5, 11] and deployments of their products by municipalities, including NYC [10].

Privacy and security issues are absolutely critical for oppnets (and for all pervasive computing systems). Due to space limitations, we can only refer the reader to our publications which discuss these issues [9, 7].

The next section describes the basic oppnet operations. Section 3 discusses the proposed standard implementation framework for oppnets. Section 4 concludes the paper and sketches plans for future work.

2. Basic oppnet operations

This section shows basic oppnet activities and basic oppnet application scenarios.

2.1. Seed oppnets and oppnet helpers

Seed oppnets: Each oppnet starts as a *seed oppnet*—a set of nodes employed together at the time of the initial network deployment (cf. Fig. 1). The seed is predesigned, 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 admitting other nodes, and can shrink expelling any of its nodes.

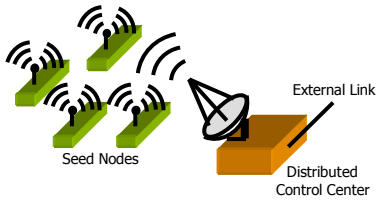

Figure 1. Seed oppnet

In addition to regular helpers, we can also have *lites* ("lightweight helpers" of limited capabilities). Helpers, but not lites, can discover and may 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 lites; 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. As pervasive computing progresses, the candidate pool will continue increasing dramatically: in infrastructures, buildings, vehicles, appliances, etc.

More densely populated areas will have, in general, a denser coverage by potential helpers. Thus, it will be easier to leverage capabilities of an oppnet in such areas. This is a desirable property: 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. True discovery could involve an oppnet node scanning the spectrum for signals or beacons, and collecting enough information to contact their senders.

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 actual 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. E.g., a water infrastructure sensornet with

multisensors, positioned near roads, can be told to sense vehicular movement (or the lack thereof).

Use of helpers can include novel combinations of existing technologies, as in 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™ [13]. If it is, the operator of the system can become a helper and contact the BANs (*body area networks*) or PANs (*personal area networks*) of car occupants.

2.2. 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 these 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 are free to either join 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).

2.3. Asking or ordering helpers and oppnet reserve

Ordering candidate helpers to join 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 labor in saving lives or critical resources.

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

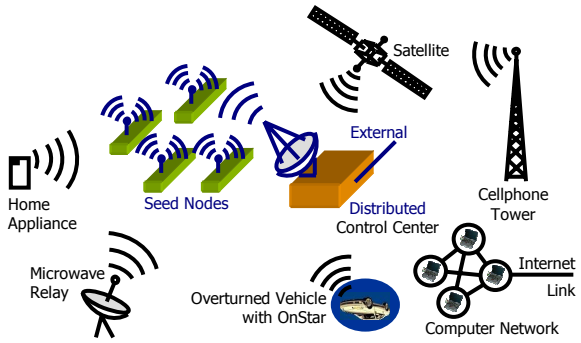

Figure 2. Expanded oppnet

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 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 is installed on them (cf. Section 3.) The "training" makes reservists highly prepared for their oppnet duties.

Oppnet reserve is not necessary for the oppnet paradigm but very helpful for at least two reasons. First, oppnet reservists in an incident area increase 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 by an oppnet the very first contact in an incident area, which is always most difficult. Once a reservist joins an oppnet, reservist's own contacts become easy next-wave contacts for the oppnet.

We have assumed 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 (and some of the reservists' 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.

3. Oppnet Virtual Machine (OVM)

To facilitate implementations of oppnet applications, we are developing the standard implementation framework for oppnets: the *Oppnet Virtual Machine* (*OVM*). Implementations from different oppnet vendors using this standard will become interoperable.

3.1. 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 interface semantics
    - Convenient C/Java/C++/C# interface bindings
  - — Extensions allowing greater flexibility
  - — Implemented on platforms of many vendors
  - — Usable in heterogeneous environments
- Allow efficient communication
  - — Uniform data/message formats

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

Tables 1, 2 and 3 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 3 except HLPR_scan, HLPR_discover, HLPR_evaluateAdmit and HLPR_releaseHelper.

Table 1. Partial list of OVM primitives for CC nodes

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

Separate primitives for the four node classes help preventing situations when a node attempts to plays a role of a node from another node class. The two main advantages of having distinct primitive classes are:

- 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 them with different programming languages and implement OVM libraries. This development follows the models of PVM [15] and MPI [4] used in grid computing.

Table 2. 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

3.2. Example oppnet EPR application scenario

The following simple scenario illustrates an oppnet EPR 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 expands by admitting helpers and lites. E.g., Bluetooth-equipped smoke and

motion detectors become lites. Lites will detect any motion and will transmit data to oppnet coordinators.

Table 3. 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

The sequence chart of the scenario is displayed in Fig. 3. (Messages are labeled with the names of primitives sending them. Reliable message delivery is assumed.) The chart shows how seed nodes obtain information from a lite via a helper. The lite runs a small motion detection application.

The pseudocodes for oppnet seed nodes and oppnet helper from Fig. 3 are shown in Figures 4 and 5. Oppnet CC nodes (Control Center nodes) are reactive systems that respond to human commands and process data reported from oppnet seeds. Since their main activities do not require oppnet primitives, we omit their pseudocode. Due to space limitations, pseudocode for lites is omitted as well, and all pseudocodes are significantly simplified, serving illustrative purposes only.

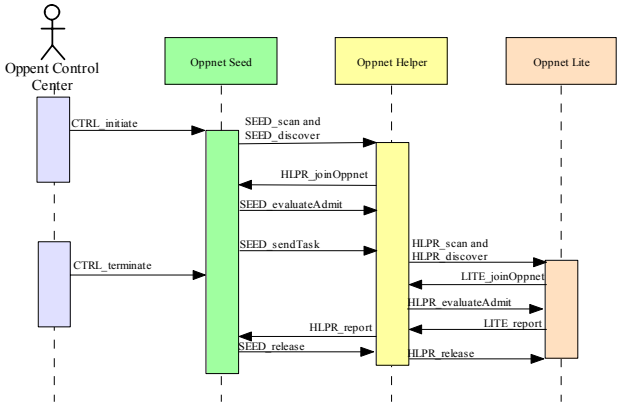

Figure 3. Sequence chart of an example oppnet application scenario

The nodes of an expanded oppnet—incl. seed nodes, helpers, and lites—keep listening to commands from the oppnet's CC or other authorized nodes (e.g., a helper can accept tasks form another helper). When a command *C* is received by a node, the node first verifies *C*. The verification may include: (i) checking sender's access rights; (ii) checking security level of *C*; (iii) estimating resources needed to carry out *C*. The command will be executed if it passes the checks.

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.

4. Conclusions and future work

Opportunistic networks or *oppnets* constitute a new, highly specialized category of ad hoc networks. They 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 future oppnet research. We continue 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 feedback necessary for fine-tuning oppnet architectures and their performance.

```
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

Acknowledgments Supported in part by the NSF grant IIS-0242840, and the U.S. Department of Commerce grant BS123456.

5. References

[1] B. Bhargava, L. Lilien, A. Rosenthal, and M. Winslett, "Pervasive Trust," *IEEE Intelligent Systems*, Sep./Oct. 2004.

[2] B. Braunstein *at al*., "Challenges in Using Distributed Wireless Mesh Networks in Emergency Response," *Intl. Conf. on Information Systems for Crisis Response and Management* (*ISCRAM 2006*), May 2006.

[3] P. Gomez Bello *et al.*, "m-ARCE: Designing a Ubiquitous Mobile Office for Disaster Mitigation, Services and Configuration, "*Intl. Conf. on Information Systems for Crisis Response and Management* (*ISCRAM 2006*), May 2006.

[4] W. Gropp, E. Lusk, and A. Skjellum. "Using MPI: portable parallel programming with the message-passing-interface," *MIT Press*, 1994.

[5] IBM, "First Responder Interoperability Solution (FRIS)," 2005.

[6] R. Kupperman and R. Wilcox, "EMISARI - An On line Management System in a Dynamic Environment," *1st Intl. Conf. on Computer Communications*, IEEE, 1972.

[7] 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.

[8] L. Lilien, Z. H. Kamal, and A. Gupta, "Opportunistic Networks: Research Challenges in Specializing the P2P Paradigm," *Proc. 3rd Intl. Workshop on P2P Data Management, Security and Trust* (*PDMST'06*), Sept. 2006.

[9] L. Lilien, Z.H. Kamal, V. Bhuse, and A. Gupta, "Opportunistic Networks: The Concept and Research Challenges in Privacy and Security," in: "Mobile and Wireless Network Security and Privacy," ed. by K. Makki *et al*., Springer Science+Business Media, 2007 (to appear).

[10] G. Menchini, "Citywide IT Preparedness for Critical Events: Accomplishments and Challenges," keynote talk, *ISCRAM*, May 2006.

[11] Motorola, "Mesh Enabled Architecture (MEA®) Solutions for Emergency Response Agencies," 2005.

[12] A. Meissner, Z. Wang, W. Putz, and J. Grimmer, "MIKoBOS – A Mobile Information and Communication System for Emergency Response," *ISCRAM*, May 2006.

[13] OnStar Corp., "On Star Explained," 2006.

[14] S. Otim, "A Case-Based Knowledge Management System for Disaster Management: Fundamental Concepts," *ISCRAM*, May 2006.

[15] V. Sunderam, G. Geist, J. Dongarra, and R. Manchek. "The PVM concurrent computing system: Evolution, experiences, and trends," *Parallel Comp.*, *Vol. 20(4),* April 1994.

[16] B. Tatomir *et al.*, "Intelligent Systems for Exploring Dynamic Crisis Environments," *ISCRAM*, May 2006.

[17] M. Turoff *et al.*, "The Design of a Dynamic Emergency Response Management Information System (DERMIS)," *J. of Info Tech. Theory and Applic.* (*JITTA*), Vol. 5(4), 2004.