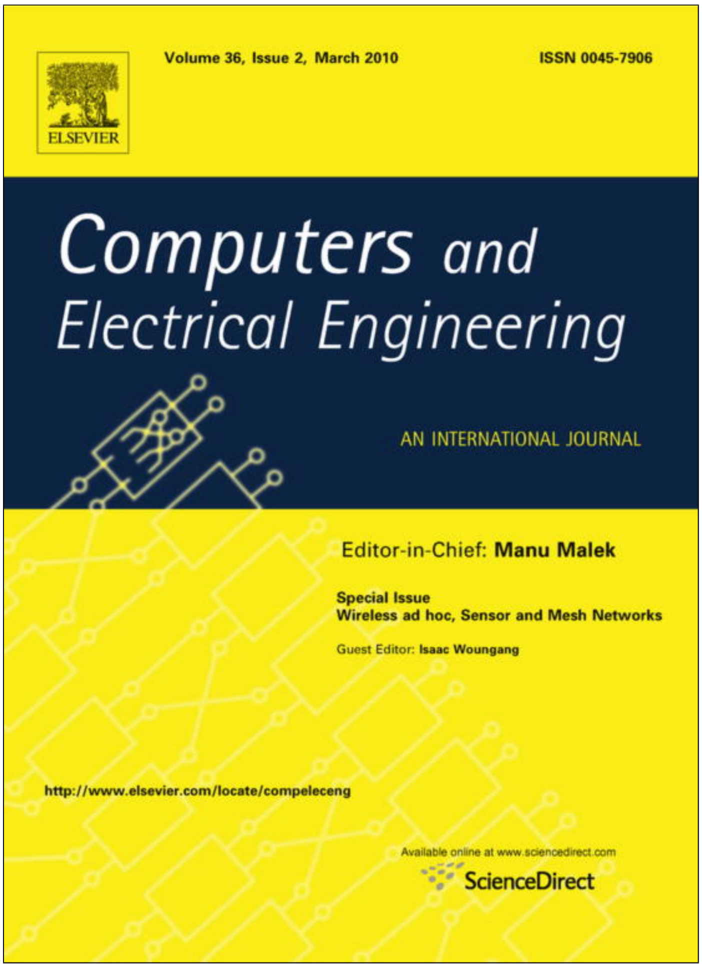


Volume 36, Issue 2, March 2010 ISSN 0045-7906

ELSEVIER

Computers and Electrical Engineering

AN INTERNATIONAL JOURNAL

Editor-in-Chief: Manu Malek

Special Issue
Wireless ad hoc, Sensor and Mesh Networks

Guest Editor: Isaac Woungang

http://www.elsevier.com/locate/compeleceng

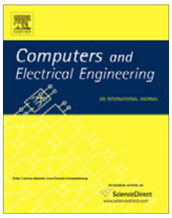
Available online at www.sciencedirect.com
ScienceDirect

Computers and Electrical Engineering 36 (2010) 328–340

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng



Opportunistic resource utilization networks— A new paradigm for specialized ad hoc networks

Leszek Lilien [a,b,*], Ajay Gupta [a], Zill-E-Huma Kamal [a], Zijiang Yang [a]

[a] Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008-5466, United States
[b] Affiliated with The Center for Education and Research in Information Assurance and Security (CERIAS), Purdue University, West Lafayette, IN 47907-2086, United States

ARTICLE INFO

Article history:
Available online 13 May 2009

Keywords:
Ad hoc networks
Oppnets
Opportunistic networks
Specialized ad hoc networks

ABSTRACT

We present *opportunistic resource utilization networks* or *oppnets*, a novel paradigm of specialized ad hoc networks. We believe that applications can benefit from using specialized ad hoc networks that provide a natural basis for them, the basis more efficient and effective than what general-purpose ad hoc networks can offer. Oppnets constitute the subcategory 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. Oppnets have a significant potential to improve a variety of applications, and to create new application niches. We categorize opportunistic networks currently known in the literature as *class 1 opportunistic networks* that use opportunistically only communication resources, and *class 2 opportunistic networks* or *oppnets* that use opportunistically all kinds of resources, including not only communication but also computation, sensing, actuation, storage, etc. After describing the oppnets and the basics of their operation, we discuss the *Oppnet Virtual Machine* (*OVM*)—a proposed standard implementation framework for oppnet applications. It is followed by a discussion of an example application scenario using the OVM primitives. Next, we discuss the design and operations of a small-scale oppnet, named MicroOppnet, originally developed as a proof of concept. MicroOppnet is now being extended to serve as a testbed for experimentation and pilot implementations of oppnet architectures and their components. We conclude with a summary and a list of some open issues for oppnets.

1. Introduction

An emerging technology can be defined [6] as: "One whose science, basic principles and theory are understood, and at least some useful applications are recognized. However, the potential is mostly unfulfilled as evidenced by a lack of significant products, possibly by the lack of market demand or need." Another definition [20] adds: "Emerging technologies are new and *potentially* [our emphasis] disruptive technologies which have the potential to significantly change everyday life in the near future. A disruptive technology supersedes or marginalizes an existing dominant technology or status quo product in the market." Of course, the positive effects of this disruptiveness must marginalize the negative ones for the technology to become successful. In the computing systems domain, the former may include reduction of time and costs of system development and easier application use, and the latter typically include dealing with legacy systems and "legacy thinking" of the software industry and users of systems based on new paradigms.

[*] Corresponding author. Address: Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008-5466, United States.
E-mail address: llilien@cs.wmich.edu (L. Lilien).

0045-7906/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2009.03.010

Wikipedia's *List of Emerging Technologies* [20] shows "Wireless communication" as an emerging technology that is set to disrupt "Wired communication." Many specialized or specific technology paradigms within the broad category of "Wireless communication"—such as ad hoc and mesh networks, mobile networks, and sensor networks—could as well be termed "emerging." They bring with them a promise of significant improvements in application development or use.

We present *opportunistic resource utilization networks* or *oppnets*—a novel paradigm for specialized ad hoc networks. Oppnets have all the traits of an emerging technology. (As explained below, they can be viewed as a generalization of the notion of *opportunistic networks* that are opportunistic just in their communications or data forwarding functions.) Oppnets can comprise both wired and wireless devices or subnetworks, so they can be viewed as another specialized solution within the "emerging" wireless communication category.

We believe that applications can benefit from using *specialized* ad hoc networks that can provide a natural basis for them, the basis more efficient and effective than what *general-purpose* ad hoc networks can offer. The practice of building specialized ad hoc networks is well established. (Mobile ad hoc networks, ad hoc wireless sensor networks, and ad hoc P2P systems are perhaps the best-known categories of specialized ad hoc networks. Another category—opportunistic networks—is less popular but gaining ground as well.) Yet, many proposed technical solutions are not generally applicable to all ad hoc networks and to systems based on them, being too inefficient for specific application classes or individual applications. This relative lack of application support (including the lack of sufficient quality of service or QoS guarantees) is perhaps the most neglected of today's crucial challenges facing ad hoc networks. Others—including variable topology, component heterogeneity, limited power supply and the need for effective energy-efficient designs—are investigated much more actively.

Specialized ad hoc networks [13,32] provide application-oriented divide-and-conquer approach to ad hoc networking and system research and development—rather than one-size-fits-all *general* approach, maybe unattainable. Oppnets are specialized ad hoc networks and enable building specialized ad hoc systems.

We provide description of the paradigm and its pilot version. We illustrate its use with examples mostly from the domain of *emergency preparedness and response* (*EPR*) [25].

1.1. How oppnets alleviate resource utilization challenges and their communication underpinnings

Oppnets have an outstanding potential for a truly beneficially "disruptive" effect on existing technologies. They can make applications using them significantly more effective and efficient, in particular by providing these applications a wealth of communication modes, sensing devices, and other tools. By their very nature of relying on growth and expansion (discussed shortly), oppnets are highly adaptive, can be exploited for achieving highly reliable and dependable operation in highly dynamic and unforeseeable situations.

One of the reasons for outstanding potential of oppnets and oppnet-based applications—to be explained later—is their ability to contribute towards their objectives at a very low or no cost (the latter especially—but not only—in emergency situations) via employing armies of so called "helpers," mobilized by oppnets. This oppnet potential is realized by leveraging the wealth of pervasive resources and capabilities that are within an oppnet's reach. This is often a treasure that remains useless due to "linguistic" barriers. Different devices and systems are either unable to speak to each other, or do not even try to communicate. They remain on different wavelengths—sometimes literally, always at least metaphorically. This occurs despite systems quickly gaining ground in computing power and intelligence, allowing for autonomous behavior, self-organization abilities, adaptability, and even self-healing when faced with component failures or malicious attacks. It might be somewhat ironic to a person unaware of interoperability challenges that such powerful and intelligent entities are not making equally great strides in talking to each other.

Each oppnet serves a certain application. Oppnet goals can be realized by alleviating first of all the communication problems for the application—including bottlenecks and gaps—that are often the root causes of resource, service, or capability shortages (similarly as transportation inadequacies—not a lack of food in the world—are the root causes of famines). Once this is achieved, an oppnet can next assist its application in obtaining all kinds of other resources, services or capabilities (e.g., storage resources, format conversion services, and image processing capabilities).

1.2. Oppnets as a generalization of the opportunistic networks paradigm

To the best of our knowledge, oppnets or opportunistic *resource utilization* networks defined by us are a novel subarea of networks. An earlier paper co-authored by one of us [3] was the first to define opportunistic *sensor* networks (which we view now as a subclass of oppnets). We added the words "resource utilization" to the original full name of our paradigm (i.e., "opportunistic networks" [15,16]) to differentiate our "opportunistic resource utilization networks" (or *oppnets*) very clearly from the other "opportunistic networks."[1] The latter exhibit only quite restricted *class 1 opportunism* (as we call it) limited, e.g., to opportunistic communication when devices are within each other's range. In contrast, oppnets aim to fully exploit so called *class 2 opportunism* by relying on an opportunistic expansion and opportunistic utilization of resources gained by this expansion

[1] When our first papers mentioning oppnets [3,16] were being prepared for publication, the search for the term "opportunistic networks" had shown that it was unused, so we chose the term for our invention. In the meantime other networks using the name "opportunistic" have appeared.

[19]. Networks with built-in specialized facilities for opportunistic data forwarding or dissemination [28,30] might be considered *"class 1.5" opportunistic networks* [19].

Oppnets differ from traditional networks, in which the nodes of a single network are all deployed together, with the network size and locations of its nodes pre-designed. In oppnets, the initial *seed oppnet* grows into an *expanded oppnet* by taking in foreign nodes. Thus, oppnets constitute the category of networks where diverse devices, *not* employed initially as its nodes, join the original set of seed nodes to become *helpers* assisting the oppnet in realization of its goals.

Oppnets are an example of developing application-oriented specializations of ad hoc, P2P and other networks (cf. [18]). They provide application-oriented primitives to opportunistic *systems*, significantly reducing development efforts.

1.3. Oppnet characteristics and oppnet-based applications leveraging them

Due to their characteristics, use of oppnets is most beneficial for applications that are well supported by a network characterized by the following properties:

- Starts with a small network, known as a *seed network* or a *seed*.
- Requires high interoperability.
- Uses highly heterogeneous software and hardware components.
- Can provide access to diverse resources/services/capabilities obtained from *helpers* that are outside of the seed.
- Is able to maintain persistent connectivity with helpers once connections are established.

These characteristics make oppnets an excellent match, for instance, for *emergency preparedness and response* (*EPR*) applications [15]. Therefore, oppnets should be a much more natural foundation for EPR applications than many other systems currently used as a basis for EPR applications, starting from the classic ones (e.g., [12]) to the most advanced ones—mobile [22] and mesh networks [4,24].

Oppnets should have a strong impact on domains other than EPR or its sister homeland security applications, such as intelligence and warning, border and transportation security, domestic counterterrorism, protecting critical infrastructure, and defending against catastrophic terrorism [25]. We see room for oppnets, e.g., in precision agriculture; entertainment, especially computer gaming; environmental control; healthcare, including emergency care; manufacturing; military, including surveillance and battlespace information management; and transportation. As any emerging technology, oppnets will lead network-based systems into new application niches as yet hard to imagine.

Privacy and security issues are absolutely critical for oppnets (and for all pervasive computing systems). We refer the interested reader to our publications which discuss these issues [14,16,17]. Note that oppnets, as most non-trivial technologies, can be malevolent—deployed to harm humans, their artifacts, and their technical infrastructures [16].

1.4. Paper organization

This paper describes the oppnets, their implementation framework for oppnet-based applications, and their pilot version. The next section describes the basic oppnet operations. Section 3 discusses the Oppnet Virtual Machine (OVM)—a proposed standard implementation framework for oppnet applications, and shows standard OVM primitives. Section 4 presents the design and operations of a small-scale oppnet, named MicroOppnet, originally developed as a proof of concept. It is now being extended to serve as a testbed for experimentation and pilot implementations of oppnet architectures and their components. Section 5 concludes the paper with a summary of our work, and sketches plans for future investigations.

2. Basic oppnet operations

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

2.1. Seed oppnets and oppnet helpers

2.1.1. 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. 1a). 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*) or *Controller*. CC can grow the oppnet into the *expanded oppnet* by admitting other nodes (cf. Fig. 1b). It can also shrink the oppnet by releasing or expelling any of its nodes. Admitted nodes are called *helpers*. In addition to regular helpers, we can also have *lites* (which are "lightweight helpers" of limited capabilities, such as smoke or carbon dioxide detectors). Lites are oppnet-enabled, that is equipped with inexpensive, simple means of standard oppnet communications. In this way, even lites can be triggered to operate in the oppnet mode when needed and commanded to do so by a CC. Regular helpers, but not lites, can discover and may admit other helpers. Thus, an oppnet node can be: a CC node; a non-CC seed nodes; a regular (non-lite) helper; or a lite.

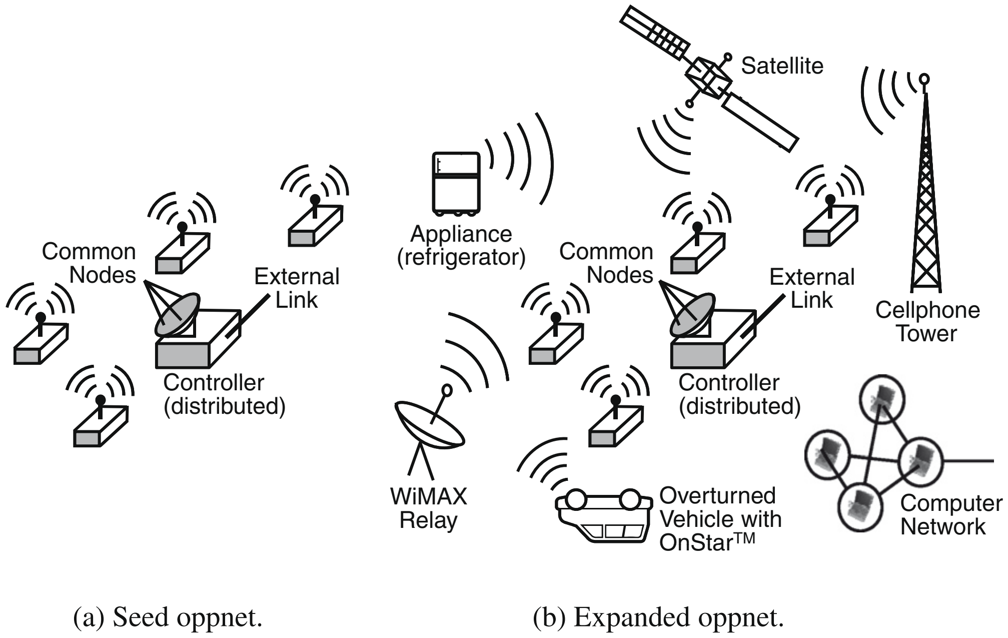


Fig. 1. Seed oppnet and expanded oppnet.

2.1.2. Potential helpers and their discovery

The set of *potential helpers* for oppnets is very broad, including communication, computing and sensor systems, wired and wireless, 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.

2.1.3. Candidates, helpers, and utilizing helpers

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).

Use of helper functionalities can be innovative in at least two ways. First, 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 monitoring opening of its door can "sense" presence of potential victims at home in a disaster area. As another example, the water infrastructure *sensornet* (sensor network) with multi-modal sensor capabilities, which is positioned near roads, can be directed to sense vehicular movement, or the lack thereof.

Second, helpers might be used in novel combinations of existing technologies, as in the following scenario. A seed oppnet is deployed in a metropolitan area after an earthquake. It finds many potential helpers, and integrates some of them into an *expanded oppnet*. One of the nodes of the expanded oppnet, a surveillance system, "looks" at a public area scene with many objects. The image is passed to an oppnet node that analyzes it, and recognizes one of the objects as an overturned car (cf. Fig. 1b). Another node decides that the license plate of the car should be read. As the oppnet currently includes no image analysis specialist, a helper with such capabilities is found and integrated into the oppnet. It reads the license plate number. The license plate number is used by another newly integrated helper to check in a vehicle database whether the car is equipped with a *vehicular communications* (*VC*) system, such as the OnStar™ system [27]. If it is, the appropriate VC center facility is contacted, becomes a helper, and obtains a connection with the VC device in the car. The VC device in the car becomes a helper and is asked to contact *body area networks* (*BANs*) on and within bodies of car occupants, or their *personal area networks* (*PANs*). Each BAN or PAN available in the car becomes a helper and reports on the vital signs or other characteristics of its owner. The reports from BANs and PANs are analyzed by dispatching nodes that schedule the responder teams to ensure that people in the most serious condition are rescued sooner than the ones that can wait for help longer. (Please note that with the exception of the BAN link that is just a bit futuristic—its widespread availability could be measured in years not in decades—all other node and helper capabilities used in the scenario are already quite common.)

2.2. Growth of seed oppnet into expanded oppnet

A seed oppnet grows into an expanded oppnet after admitting new helpers. For example, the expanded oppnet in Fig. 1b 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* (*HAN*), contacted via an intelligent appliance (e.g., a refrigerator) with a wireless link; (e) a Wi-MAX network, contacted via a WiMAX relay; (f) BANs of occupants of an overturned car, contacted via a vehicular communications system, such as OnStar.

For predictable disasters (like hurricanes), a seed oppnet can be put into action and its buildup into an expanded oppnet started (or even completed) *before* the disaster, when it is still much easier to locate and invite other nodes and clusters into the oppnet. As an example, the first invited helpers could be the sensornets deployed for structural damage monitoring and assessment in buildings, roads, and bridges.

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 emergency oppnet.

The category of systems always available on an order of an emergency or other critical-application 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 reservists, 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 (cf. Section 3) is installed on them. 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.

For EPR applications, at least one reservist must survive an incident. There will be numerous reservists in practically every area of the country, the more reservists the more densely populated is an area. Thus, we are practically guaranteed—for all but the most cataclysmic disasters—that some reservists *will* survive.

By employing helpers working for free as volunteers or conscripts, oppnets 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.

2.4. Dealing with unintended consequences of employing helpers

Integrating helpers by oppnets could have dangerous unintended consequences such as disruptions of operations of life-support and life-saving systems, traffic lights, utilities, wireline and wireless phones, the Internet, etc. To protect critical operations of oppnets and helpers joining an oppnet, oppnets must obey the following principles:

- Oppnets must *not disrupt critical operations* of potential helpers. In particular, they must not take over any resources of life-support and life-saving systems.
- For potential helpers running non-critical services, *risk evaluation* must be performed by an oppnet before they are asked or ordered to join the oppnet. This task may be simplified by potential helpers identifying their own risk levels, according to a standard risk level classification.
- *Privacy and security* of oppnets and helpers must be assured, especially in the oppnet growth process.

2.5. The basic sequence of oppnet operations

The basic sequence of oppnet operations is shown in Fig. 2. Let us note that upon accepting an invitation a candidate is admitted into the oppnet, becoming its helper. Oppnet's tasks can be offloaded to or distributed among all helpers (*Collaborative processing* in Fig. 2).

The Command Center—either augmenting human operators or fully autonomous—presides over the operations of the oppnet throughout its lifetime. If the oppnet needs more resources for to achieve its goal, the process repeats, and once the goal of the oppnet has been achieved, the helpers are released.

3. Oppnet Virtual Machine (OVM)

The Oppnet Virtual Machine (OVM) is a part of the oppnet project. The goal of OVM work is proposing a standard to facilitate oppnet implementations by different software vendors. The OVM standards will also assure interoperability among these different oppnet implementations and third-party oppnet products. OVM will assist in developing and marketing standard library routines and APIs for all kinds of applications.

3.1. OVM primitives

The OVM primitives are intended for use by all those who want to write programs in C/C++/C# or Java for oppnets and any oppnet-enabled devices. This includes individual application programmers, manufactures of hardware devices, and creators of environments and tools. In order to be attractive to this wide audience, the standard must provide a simple, easy-to-use interface. As planned now, this standard will not specify program construction tools, debugging facilities, support for task management, or underlying mechanisms for communications. However, oppnet features that are not included in the OVM standard can always be offered as extensions by specific implementations.

Our standardization work follows the models of MPI [8] and PVM [31] used in grid computing.

Recall that at any moment a node belongs to only one of the four categories: CC nodes, seed nodes, helpers, and lites. Tables 1 and 2 show partial lists of the primitives offered by OVM for the oppnet's CC nodes and seed nodes. (Similar tables with primitives offered by OVM for the oppnet's helpers and lites are not included here due to space limitations but are available in Ref. [11].) The OVM primitives for the four categories of nodes have prefixes CTRL_, SEED_, HLPR_ and LITE. A description of all primitives is available elsewhere [11].

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 and resource savings [15].

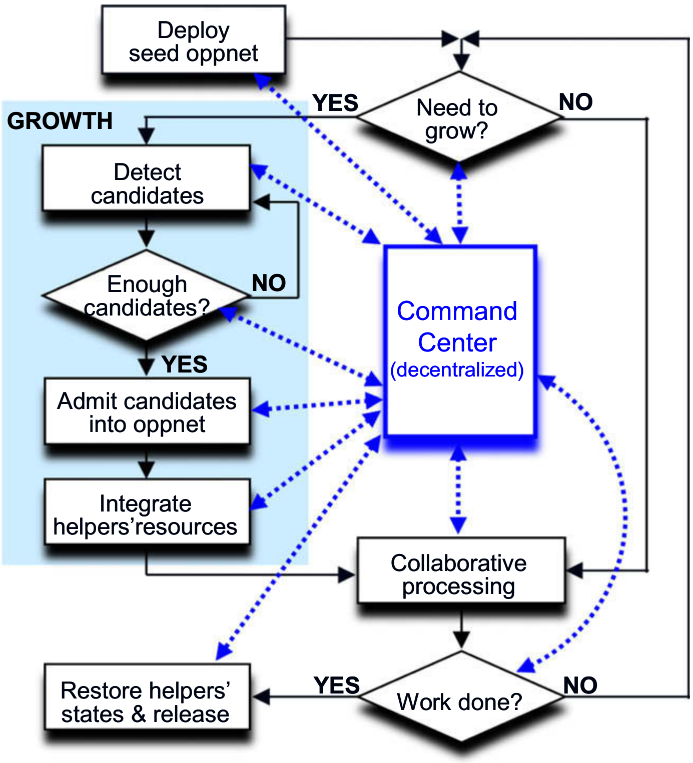


Fig. 2. The basic sequence of oppnet operations.

Table 1
Partial list of OVM primitives for CC nodes.

Name of the primitive	Functions of the primitive
`CTRL_start`	Initiate oppnet
`CTRL_end`	Terminate oppnet
`CTRL_cmd`	Send command to seed nodes

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 a buffer
`SEED_validate`	Verify the received command
`SEED_isMember`	Check if a device is already an oppnet node (oppnet member)
`SEED_evalAdmit`	Evaluate a device and admit it into the oppnet if the device meets criteria for admittance
`SEED_sendTask`	Send a task to another 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_processMsg`	Process a message from a buffer
`SEED_report`	Report information to the 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. Among others, the motion sensors embedded in Bluetooth-equipped smoke detectors will become oppnet lites. If a lite detects any movement, data will be transmitted to oppnet coordinators.

The sequence chart for such a 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 oppnet application.

The pseudocode for oppnet seed nodes is shown in Fig. 4. (Pseudocodes for helpers and lites are not included here due to space limitations but are available in Refs. [11,15]. CC 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. The pseudocodes are significantly simplified, serving illustrative purposes only.)

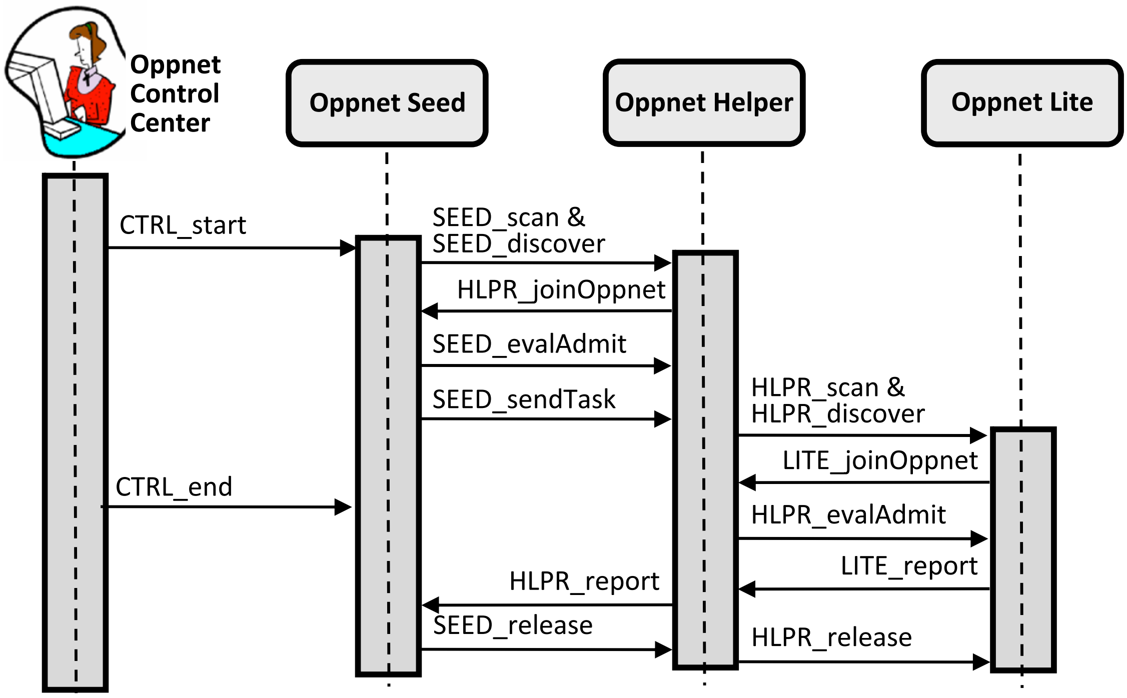


Fig. 3. Sequence chart for an example oppnet application scenario.

```
repeat on command from CC or 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_evalAdmit(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_processMsg(M);
         SEED_report(…);
      ...
   end_switch
end_repeat
```

Fig. 4. Pseudocode for seed nodes in the scenario.

The nodes of an expanded oppnet (all four categories) keep listening to commands from the oppnet's 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: (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. For instance, a command that requires connection via Wi-Fi will not pass Check (iii) if the device has a Bluetooth connectivity only.

Oppnet helpers and lites perform their regular activities until they are integrated into an oppnet, in which case they switch to a well-defined *emergency mode*.

4. Design and sample application scenario for MicroOppnet

This section discusses design and operations of MicroOppnet v.2.2., which was developed as a proof of concept and a testbed for oppnets. This small-scale prototype integrates Bluetooth, wireless Internet, and sensornets.

4.1. Overview of MicroOppnet

The *requirements* for MicroOppnet included project goals and evaluation criteria. Based on them, design choices were specified, and decisions made. The details, omitted here due to space restrictions, are available elsewhere [10].

The resulting current version of MicroOppnet built by us is both a small-scale proof of concept and a testbed for class 2 opportunistic networks (oppnets), since it not only allows opportunistic communications but also opportunistically accesses sensornet nodes to perform sensing. It is, though, rudimentary in its class 2 opportunism, hence the prefix "micro" in the name "MicroOppnet" ("micro" can also be understood in relation to its small size).

MicroOppnet is a platform on which functional parameters can be investigated in order to design, implement, test, and fine-tune oppnet components, such as OVM primitives, protocols, and architectures. Non-functional parameters—including quality of service (QoS) parameters (throughput, delay, reliability, accuracy, scalability, etc.)—can also be investigated.

The seed oppnet for MicroOppnet v.2.2 (cf. Fig. 5) consists of Workstation A with a Bluetooth (BT) adapter and a serial port connection to Sensornet Base Station BS1, and BS1. The seed searches for BT devices and initiates a connection with them. Alternatively, a BT-enabled device—a Victim cellphone in our example—can find the seed and initiate a connection. Once a connection has been established, the Victim cellphone can send a message to the seed, for example, the `help` message. This message is then forwarded via Base Station BS1, and then through the sensor network. In MicroOppnet, the sensornet consists of 10 Mica2 Motes and 6 Stargate sensornet gateways. Some of the gateways are also connected to Mica2 Motes.

Base Station BS2 at the other end of the sensornet is connected to Laptop B. Once the `help` message is propagated via BS2 to Laptop B, a Java TCP/IP client socket connection is initiated with Remote Java (Database) Server. The `help` message and the location of a device that sent it are logged on this server.

Remote Java (Database) Server can be queried by remote users employing either traditional computing devices or Java-enabled devices. In our example, the Responder cellphone employs T-Mobile™ Virtual Private Network (VPN).

The seed can broadcast to the sensornet a variety of messages in addition to the `help` message—e.g., `start_sensing`, `log_sensing`, `retrieve_log`. The messages can be used, for instance, to start temperature sensing, to log temperature

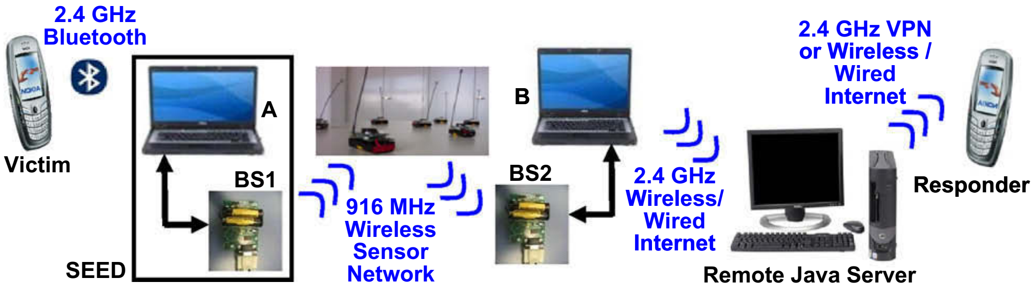

2.4 GHz Bluetooth
Victim
A
BS1
SEED
916 MHz Wireless Sensor Network
B
BS2
2.4 GHz Wireless/ Wired Internet
2.4 GHz VPN or Wireless / Wired Internet
Responder
Remote Java Server

Fig. 5. Structure of MicroOppnet v.2.2.

in the EEPROM of the sensor, or to retrieve the logged data from the sensor network. The retrieved temperature readings can be logged at Remote Java (Database) Server. Then, they can either be queried by remote users via wireless Internet, or be broadcast by the seed oppnet on the BT channels.

MicroOppnet v.2.2 integrates three communication media and frequency ranges: BT (2.4 GHz), a sensor network (916/896/433 MHz), and the wireless Internet using 802.11b/g (at 2.4 GHz, i.e., using the same frequency as BT).

4.2. Design of MicroOppnet

Fig. 6 shows the flow of control for MicroOppnet of Fig. 5 in terms of the OVM primitives. The flow can begin with: (a) an active discovering of candidates—using `SEED_discover`; (b) with a passive wait—using `SEED_listen`, when candidates search for and initiate connection with the seed; or (c) with dispatching a task for the sensornet—using `SEED_sendTask`. In MicroOppnet, communication for (a) and (b) is only over the Bluetooth (BT) medium.

Messages from nodes wishing to use MicroOppnet are processed. Tasks are delegated to the appropriate helpers. In MicroOppnet, there are only two sets of helpers: the nodes of the sensornet, and Remote Java (Database) Server.

Messages from a user such as Victim in Fig. 5 can be forwarded from the seed's Sensornet Base BS1 to helpers using the `SEED_sendTask` primitive. The nodes in the sensornet process the message using `HLPR_processMsg` and then perform the task using `HLPR_runApp`. If the task is sensing, then the Sensor Network Nodes will start or stop sensing as required. Otherwise, they will forward either the received message or their temperature sensor readings as directed. When a message is received by another sensornet gateway or another base station (e.g., by BS2), it is logged on Remote Java (Database) Server. If the task was to retrieve sensor-measured temperature, then BS2 aggregates sensornet readings and floods the result back through the sensornet to BS1.

Devices such as Responder (cf. Fig. 5) can send the message `retrieve_log` to Remote Java (Database) Server, which is a helper. This server uses the `HLPR_listen` primitive to be in a listening mode. This allows its log to be queried for specific tasks and retrieve the appropriate messages. The server can process any TCP/IP socket connection.

Summarizing, MicroOppnet supports the following tasks: (i) communication tasks: flooding messages and retrieving sensor readings; and (ii) sensing tasks: starting and stopping sensing. All these tasks rely on opportunism. In more detail, the following is the exhaustive list of all tasks using resources opportunistically:

- Communication over the BT medium.
- Communication over the sensornet medium.
- Communication using TCP/IP over a wired or wireless Internet.
- Temperature sensing using sensornet nodes.

The first three tasks use class 1 opportunism, and only the last task relies on class 2 opportunism—by leveraging the sensing resources of MicroOppnet helpers. Thanks to the last task, we can claim that MicroOppnet is a class 2 opportunistic network, albeit a rudimentary one (exploiting only one type of non-communication resources).

We conclude this subsection with two notes. First, MANET routing [29,34] is used in MicroOppnet, so every node is a router. Second, implementation details, not included due to space limitations, can be found in Ref. [10].

4.3. Sample application scenario for MicroOppnet

To illustrate use of MicroOppnet, let us consider an emergency scenario: a fire in a large office building. Suppose that some workers were unable to evacuate. Most of them tried to use their cellphones to call for help. Many succeeded but many failed to get a connection since the cellphone infrastructure was overloaded at that time with calls being made by thousands of workers and passers-by outside of the building.

Firefighters can put a MicroOppnet to use. They deploy around the office building the MicroOppnet seed, consisting of laptops and networks connecting them. The Bluetooth (BT) Class 1 connectivity becomes an essential communications capability. (BT Class 1 has the range of approx. 100 meters. Note that BT Class 1 has nothing in common with class 1 or class 2 opportunistic networks—the similarity of names is coincidental.) MicroOppnet uses BT Class 1 connectivity to discover all

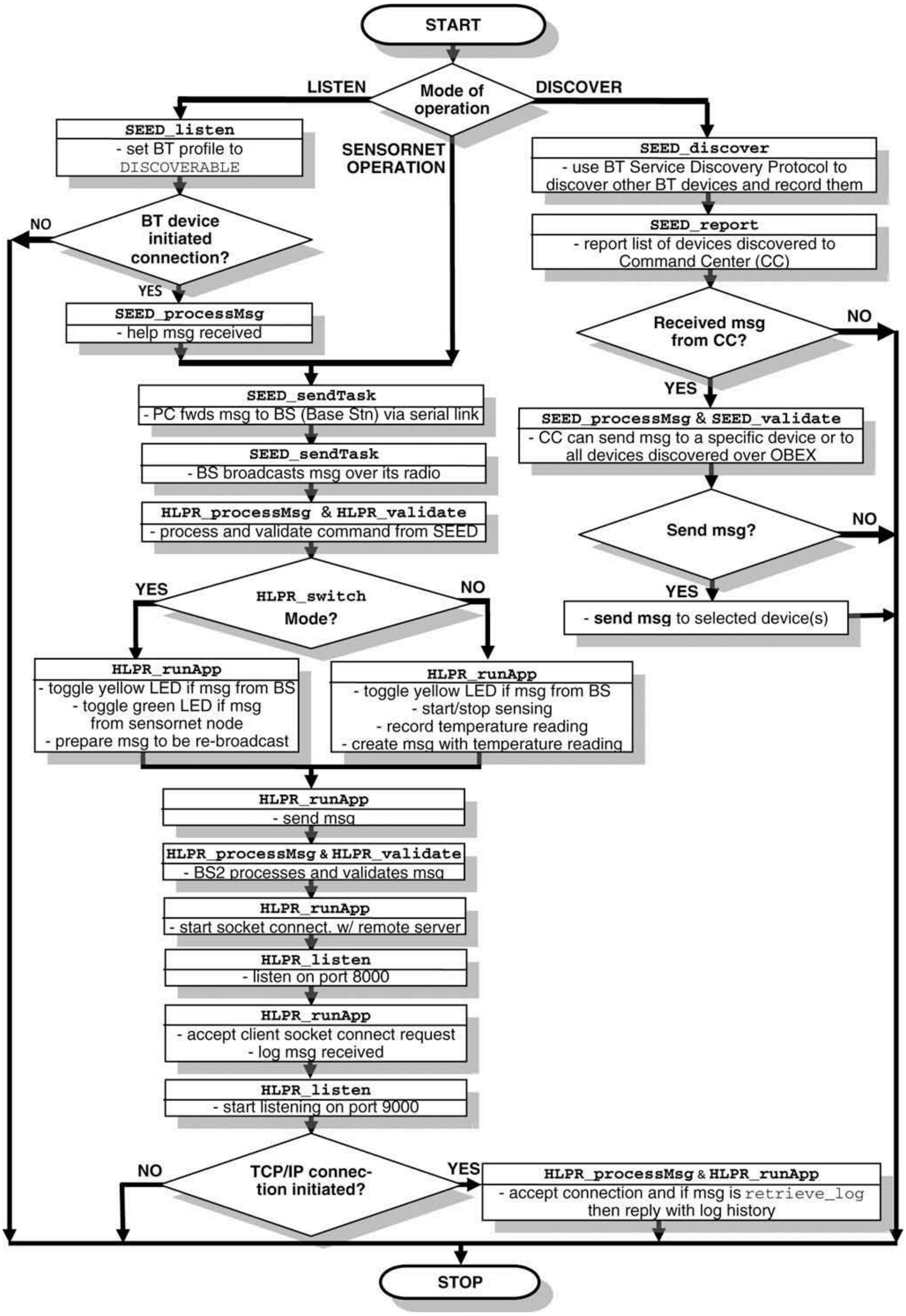

START
Mode of operation
LISTEN
DISCOVER
SENSORNET OPERATION
SEED_listen
- set BT profile to DISCOVERABLE
BT device initiated connection?
NO
YES
SEED_processMsg
- help msg received
SEED_discover
- use BT Service Discovery Protocol to discover other BT devices and record them
SEED_report
- report list of devices discovered to Command Center (CC)
Received msg from CC?
NO
YES
SEED_processMsg & SEED_validate
- CC can send msg to a specific device or to all devices discovered over OBEX
Send msg?
NO
YES
- send msg to selected device(s)
SEED_sendTask
- PC fwds msg to BS (Base Stn) via serial link
SEED_sendTask
- BS broadcasts msg over its radio
HLPR_processMsg & HLPR_validate
- process and validate command from SEED
HLPR_switch Mode?
YES
NO
HLPR_runApp
- toggle yellow LED if msg from BS
- toggle green LED if msg from sensornet node
- prepare msg to be re-broadcast
HLPR_runApp
- toggle yellow LED if msg from BS
- start/stop sensing
- record temperature reading
- create msg with temperature reading
HLPR_runApp
- send msg
HLPR_processMsg & HLPR_validate
- BS2 processes and validates msg
HLPR_runApp
- start socket connect. w/ remote server
HLPR_listen
- listen on port 8000
HLPR_runApp
- accept client socket connect request
- log msg received
HLPR_listen
- start listening on port 9000
TCP/IP connection initiated?
NO
YES
HLPR_processMsg & HLPR_runApp
- accept connection and if msg is retrieve_log then reply with log history
STOP

Fig. 6. Flow of control in MicroOppnet v.2.2.

kinds of BT-enabled helpers. An owner of any such helper can now communicate with the firefighters via the expanded MicroOppnet (consisting of the seed plus all helpers that joined it).

We used so far only class 1 opportunism in MicroOppnet. To show how class 2 opportunism can be used, suppose that MicroOppnet is now commanded to contact and query for temperature readings all sensing nodes within the building. They include a multitude of oppnet-enabled devices (in this example, oppnet-enabled means BT-enabled), including ubiquitous smoke and motion detectors with add-in multisensor capabilities. The obtained temperature readings, aggregated at the Remote Java (Database) Server, are used to plot the heat profile for the building. The profile, together with location information gathered by BT-equipped helpers before, can be used by the firefighters to find the best routes for reaching people trapped in the building by fire. Many other pervasive communications technologies could be used in parallel with BT (but our example is complete even without them).

5. Conclusions and future work

We proposed categorization of opportunistic networks known from the literature into *class 1 opportunistic networks* and even "more opportunistic" *class 2 opportunistic networks* or *oppnets*. *Class 2 opportunistic networks* or *oppnets* constitute a new, highly specialized category of ad hoc networks, proposed by us. As shown, they provide an unprecedented leveraging potential for growing from a small seed network into a very powerful network, and using this growth to acquire at low or no cost vast communication, computing, sensing, storage, and other capabilities.

We introduced Oppnet Virtual Machine (OVM), a standard implementation framework for oppnets, and presented OVM primitives for four categories of oppnet nodes, namely control center nodes, seed nodes, helper nodes, and lites (i.e., lightweight nodes). We gave an overview of an example oppnet application scenario.

We described design and operations of MicroOppnet, a small-scale oppnet, which serves not only as a proof of concept but is currently being extended as a testbed providing stimulation and feedback needed for designing, implementing, testing and optimizing performance of: (a) oppnet primitives; (b) routing, privacy and security, and other oppnet protocols; (c) oppnet architectures (cf. architectures for other advanced networks [1,2,5,7,21,23,26]).

We see many challenges for future oppnet research. We continue investigation of oppnet fundamentals, including novel oppnet architectures with their associated components: methods, protocols, and algorithms. Examples of future research tasks—only the ones related to MicroOppnet implementation and investigation—include the following:

(1) Extending the rudimentary class 2 opportunism of MicroOppnet to a more substantial class 2 opportunism—by more fully implementing the opportunistic growth mechanisms of class 2 opportunistic networks.
(2) Including comprehensive privacy and security mechanism to assure confidentiality, integrity, availability, authentication, non-repudiation, etc.
(3) Implementing a specialized oppnet routing protocol (cf. [33]) to replace the current generic MANET-like routing (cf. [29,34]).
(4) Extending the range of communication media by using, for example, infrared spectrum, WiMAX, and RFIDs.
(5) Implementing detection of a wider variety of helpers via a broader range of communications technologies. The current version of MicroOppnet detects only Bluetooth devices. It is unable to detect, e.g., sensornet nodes, and incorporate them into the oppnet.
(6) Implementing Service Discovery Protocols, including protocols based on beacons, advertising, service location and planning [9], and Software Defined Radio or Cognitive Radio.
(7) Implementing release of helpers by an oppnet after they performed their "tour of duty" for the oppnet.

Broader open issues and challenges for the oppnet paradigm are discussed in some detail in Ref. [18]. Examples of such issues include: optimizing the seed oppnet infrastructure; developing methods for detecting useful helpers, inviting them, controlling them; developing methods for deciding which tasks should be "offloaded" by oppnet to which helpers; proposing techniques for coordinating helper tasks by oppnets; proposing ways of managing oppnets, including control of privacy and security in oppnets.

We should finally note that oppnets are a technology moving us towards the vision of pervasive computing.

Acknowledgements

This research was supported in part by the NSF under grants IIS-0242840 and IIS-0209059, and in part by the U.S. Department of Commerce under Grant BS123456. The authors would also like to acknowledge Computational Science Center and Information Technology and Image Analysis (ITIA) Center. Any opinions, finding, conclusions or recommendation expressed in the paper are those of the authors and do not necessarily reflect the views of the funding agencies or institutions.

References

[1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. Comput Networks 2002;38:393–422.
[2] Baker M, Buyya R, Laforenza D. Grids and Grid technologies for wide-area distributed computing. Software Pract Exp 2002;32(15):437–1466.
[3] Bhargava B, Lilien L, Rosenthal A, Winslett M. Pervasive Trust. IEEE Intel Sys 2004;19(5):74–7.
[4] Braunstein B et al. Challenges in using Distributed Wireless Mesh Networks in Emergency Response. In: 3rd international conference on information systems for crisis response and management (ISCRAM 2006), Newark, New Jersey; 2006.
[5] Cerf V, Burleigh S, Hooke A, Torgerson L, Durst R, Scott K, et al. Delay-Tolerant Network Architecture, DTN Research Group Internet Draft; 2003.
[6] Emerging Technologies, Bitpipe.com, <http://www.bitpipe.com/tlist/Emerging-Technologies.html> [accessed 15.06.08].
[7] Feeney LM, Ahlgren B, Westerlund A. Spontaneous Networking: an application-oriented approach to ad hoc networking. IEEE Commun 2001;39(6):176–81.
[8] Gropp W, Lusk E, Skjellum A. Using MPI: portable parallel programming with the message-passing interface. Cambridge, Massachusetts: MIT Press; 1994.
[9] Kamal ZH, Al-Fuqaha A, Gupta A. A service location problem with QoS constraints. In: International conference on wireless communications and mobile computing (IWCMC'07), Honolulu, Hawaii; 2007. p. 641–646.

[10] Kamal ZH, Gupta A, Lilien L, Yang Z. The MicroOppnet tool for collaboration experiments with class 2 opportunistic networks. In: Proceedings of the 3rd international conference on collaborative computing: networking, applications and worksharing (CollaborateCom 2007), White Plains, New York, CD-ROM (10 pages); 2007.

[11] Kamal ZH, Lilien L, Gupta A, Yang Z, Batsa M. New UMA Paradigm: class 2 opportunistic networks. In: Zhang Y, Yang LT, Ma J, editors. Unlicensed Mobile Access Technology: protocols, architectures, security, standards and applications. Boca Raton, Florida: Auerbach Publications, Taylor and Francis Group; 2008 [chapter 17].

[12] Kupperman R, Wilcox R. EMISARI – an on line management system in a dynamic environment. In: 1st international conference on computer communications, IEEE; 1972.

[13] Lilien L. A taxonomy of specialized ad hoc networks and systems for emergency applications. In: Proceedings of the 1st international workshop on mobile and ubiquitous context aware systems and applications (MUBICA 2007), Philadelphia, Pennsylvania, CD-ROM (8 pages); 2007.

[14] Lilien L, Bhargava B. A scheme for privacy-preserving data dissemination. IEEE Trans Sys Man Cybernet 2006;36(3):503–6.

[15] Lilien L, Gupta A, Yang Z. Opportunistic networks for emergency applications and their standard implementation framework. In: The 1st international workshop on next generation networks for first responders and critical infrastructure (NetCri'07), New Orleans, Louisiana; 2007. p. 588–93.

[16] Lilien L, Kamal ZH, Bhuse V, Gupta A. Opportunistic networks: the concept and research challenges in privacy and security. In: International workshop on research challenges in security and privacy for mobile and wireless networks (WSPWN 2006), Miami, Florida; 2006. p. 134–47.

[17] Lilien L, Kamal ZH, Bhuse V, Gupta A. The concept of opportunistic networks and their research challenges in privacy and security. In: Makki K et al., editors. Mobile and wireless network security and privacy. Norwell, Massachusetts: Springer Science+Business Media; 2007. p. 85–113 [chapter 5].

[18] Lilien L, Kamal ZH, Gupta A. Opportunistic networks: research challenges in specializing the P2P Paradigm. In: 3rd international workshop on P2P data management, security and trust (PDMST'06), Kraków, Poland; 2006. p. 722–6.

[19] Lilien L, Kamal ZH, Gupta A, Woungang I, Bonilla Tamez E. Quality of service in an opportunistic capability utilization network. In: Denko M et al., editors. Mobile opportunistic networks: architectures, protocols and applications. Boca Raton, Florida: Auerbach Publications, Taylor and Francis Group, in press.

[20] List of Emerging Technologies, Wikipedia, <http://en.wikipedia.org/wiki/List_of_emerging_technologies>, [accessed 15.06.08].

[21] MANET Implementations, <http://www.comnets.unibremen.de/~koo/manet-impl.html>; 2004.

[22] Meissner A, Wang Z, Putz W, Grimmer J. MIKoBOS – a mobile information and communication system for emergency response. In: 3rd international conference on information systems for crisis response and management (ISCRAM 2006), Newark, New Jersey; 2006.

[23] Milojicic DS, Kalogeraki V, Lukose R, Nagaraja K, Pruyne J, Richard B, et al. Peer-to-peer computing, report HPL-2002-57, HP Laboratories, Palo Alto, CA; 2002.

[24] Motorola, Mesh enabled architecture (MEA®) solutions for emergency response agencies; 2005.

[25] National strategy for homeland security, Office of Homeland Security; 2002.

[26] Niebert N, Schieder A, Abramowicz H, Malmgren G, Sachs J, Horn U, et al. Ambient networks – an architecture for communication networks beyond 3G, IEEE Wireless Communications (special issue on 4G mobile communications – towards open wireless architecture); 2004.

[27] On Star Explained, OnStar Corp., 2007, <http://www.onstar.com/us_english/jsp/explore/index.jsp> [accessed 23.06.07].

[28] Pelusi L, Passarella A, Conti M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. IEEE Commun 2006;44(11):134–41.

[29] Royer EM, Toh C-K. A review of current routing protocols for ad-hoc mobile wireless networks. IEEE Personal Commun 1999;6(2).

[30] Sistla P, Wolfson O, Xu B. Opportunistic data dissemination in mobile peer-to-peer networks. In: 9th international symposium on advances in spatial and temporal databases (SSTD 05), Angra dos Reis, Brazil; 2005.

[31] Sunderam V, Geist G, Dongarra J, Manchek R. The PVM Concurrent Computing System: evolution, experiences, and trends. Parallel Comp 1994;20(4).

[32] The First international workshop on specialized ad hoc networks and systems (SAHNS 2007), Toronto, <http://www.cs.wmich.edu/~alfuqaha/SAHNS>; 2007.

[33] Wang Y, Jain S, Martonosi M, Fall K. Erasure-coding based routing for opportunistic networks. In: ACM conference of the special interest group on data communication (SIGCOMM 2005), Philadelphia, PA; 2005.

[34] Zhou Z. A survey on routing protocols in MANETs. Technical reports in MSU-CSE-03-8, Department of Computer Science, Michigan State University, East Lansing, Michigan; 2003.

Leszek Lilien is an Assistant Professor of Computer Science at Western Michigan University, Kalamazoo, MI. He received Ph.D. and M.S. degrees in Computer Science from University of Pittsburgh, and his Master of Engineering degree in Electronics/Computer Engineering from Wrocław University of Technology, Wrocław, Poland.

His research focuses on *oppnets* – a class of opportunistic networks and a specialized kind of ad hoc networks; as well as trust, privacy and security in pervasive and open computing systems. He has published to date over 50 refereed journal and conference papers, and six book chapters.

Dr. Lilien serves on the editorial boards of the *International Journal of Communication Networks and Distributed Systems*, *The Open Cybernetics & Systemics Journal*, and *Recent Patents on Computer Science*. He was the main organizer and Chair of the *International Workshops on Specialized Ad Hoc Networks and Systems* (*SAHNS 2007 and SAHNS 2009*), held in conjunction with the *IEEE International Conferences on Distributed Computing Systems* (*ICDCS 2007 and ICDCS 2009*).

He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) and IEEE Computer Society.

Ajay Gupta is a Professor of Computer Science at Western Michigan University. From 1998 to 2002, he was the Chairman of the Computer Science Department at Western Michigan University. Dr. Gupta received his Ph.D. in Computer Science from the Purdue University in 1989, his M.S. in Mathematics and Statistics from the University of Cincinnati in 1984, and his B.E. (Honors) in Electrical and Electronics Engineering from the Birla Institute of Technology and Sciences, Pilani, India in 1982.

His research interests include sensor systems, mobile computing, web technologies, computer networks, evolutionary computation, scientific computing, and design and analysis of parallel and distributed algorithms. He has published numerous technical papers and book chapters in refereed conferences and journals in these areas. A paper he co-authored, "Adaptive Integration Using Evolutionary Strategies," won the Best Paper award in the International Conference on High Performance Computing in 1996. Another paper he co-authored, "Lightweight Intrusion Detection for Sensor Networks," received Honorable mention at the CERIAS Information Security Symposium in 2006. He also holds joint copyright for the parallel and distributed automatic numerical integration software package, ParInt 1.1.

He is a Senior Member of the IEEE and member of the IEEE Computer Society, the IEEE Communications Society, the ASEE and the ACM. He actively helps organize various ACM and IEEE conferences.

Zill-E-Huma Kamal is an Adjunct Professor of Computer Science at Colorado Technical University. She received her B.S. in Computer Science from Western Michigan University in 2002, and continued directly to pursue her Ph.D. degree. She received her Ph.D. degree in Computer Science in April 2008. Dr. Kamal has published numerous technical papers, journals, and book chapters and reviewed papers in the areas of service location and resource utilization in wireless networks and power consumption analysis in sensor networks. She has taught undergraduate level programming courses and graduate level courses in Modern Computer Architecture, Systems Engineering, Systems Integration and Testing. She has also given lectures in Design and Analysis of Algorithms, Computer Security and Theory of Automata/Computation. She has received teaching and research awards at department and university levels at the Western Michigan University in the 2004–2008 period.

Her research interest include analysis and design of parallel and distributed algorithms, Mobile Computing, Wireless Sensor Networks, Opportunistic Networks, Pervasive Computing, Resource Utilization with QoS, Optimization and Operational Research.

Zijiang Yang is an Associate Professor in the Department of Computer Science, Western Michigan University. He received his Ph.D. from the University of Pennsylvania in 2003. His primary research interest is computer system reliability, which spans the spectrum from hardware design automation to software engineering. He is particularly interested in model checking techniques to improve software reliability. His research is currently supported by National Science Foundation. He published over 30 research papers in international journals and conferences, and received the 2008 ACM TODAES best paper award. He is a Senior Member of IEEE.