# NanoPeer Networks and P2P Worlds

P. Triantafillou N. Ntarmos S. Nikoletseas P. Spirakis
{*peter, ntarmos*}*@ceid.upatras.gr* {*nikole, spirakis*}*@cti.gr*
Computer Technology Institute
Patras, Greece

## Abstract

*We present the NanoPeers architecture paradigm, a peer-to-peer network of lightweight devices, lacking all or most of the capabilities of their computer-world counterparts. We identify the problems arising when we apply current routing and searching methods to this nano-world, and present some initial solutions, using a case study of a sensor network instance; Smart Dust. Furthermore, we propose the P2P Worlds framework as a hybrid P2P architecture paradigm, consisting of cooperating layers of P2P networks, populated by computing entities with escalating capabilities. Our position is that (i) experience gained through research and experimentation in the field of P2P computing, can be indispensable when moving down the stair of computing capabilities, and that (ii) the proposed framework can be the basis of numerous real-world applications, opening up several challenging research problems.*

## 1 Introduction

The peer-to-peer (P2P) paradigm has emerged to be one of the hottest subjects of research and development of computer science during the last few years. After the advent of Napster ([17]) and Seti@Home ([20]), researchers have focused on the fields of content and resource sharing P2P systems, usually dealing with such tasks as indexing and searching, routing, security and anonymity, resource exploitation and load balancing, etc. This is a natural consequence of the widespread use of such systems[1].

However, what researchers usually take for granted (i.e. average processing/storage/network capacities[2] and power supply of modern computers) may not exist when we take a step further and deal with devices other than personal computers. Such restrictions may include little or no storage capacity or memory at the peers, highly unstable communication links, and power consumption issues, usually inherent in the fields of embedded devices, sensor networks, and ubiquitous computing in general.

We examine Smart Dust systems ([6]) - an inherently pure P2P system - and present NanoPeer Networks: an approach to P2P networks comprised of micro-devices acting as lightweight peers in a P2P overlay, with restricted computing and energy capabilities. We try to identify the problems arising when applying computer-world techniques to this nano-world, attempt to locate the cause of such discrepancies, and propose outlines of relevant solutions.

We further argue that, due to the analogy between sensor networks and pure P2P systems, experience gathered through research and experimentation in the P2P field, can be indispensable when dealing with real-world problems in the nano-level. As we'll see, many of the issues arising when dealing with NanoPeers, have a computer-world counterpart which has already been dealt with by P2P scientists, thus making computer-world P2P systems a first-class testbench for nano-level solutions.

The rest of the paper will proceed as follows. In the next section we discuss related work in both the fields of peer-to-peer and sensor networks. The next section (sec. 3) introduces us to the world of Smart Dust and presents two simple protocols for local detection and propagation of information in this context. We then (sec. 4) argue that Smart Dust networks are in perfect analogy with pure P2P networks and attempt a side-by-side comparison of these two P2P worlds. Section 5 discusses various extensions to the Smart Dust paradigm, inspired from experience gathered from the fields of peer-to-peer computing. Section 6 concludes the paper and proposes various open problems.

## 2 Related work

Relevant literature categorizes P2P systems in three major categories: *Pure*, *Centralized*, and *Hybrid* P2P systems.

[1] For example, the FastTrack-based P2P overlay ([11]) had a whooping 2.6 million users and 3.6 petabytes of data online at the time of writing).

[2] We'll collectively denote processing, storage, and network communication capabilities by the term "computing capabilities".

*Pure* P2P systems (such as Gnutella [12]) are systems in which all peers are of the same stature and execute the same algorithms; no peer exposes any special functionality and the operation of the system is fully decentralized. In *Centralized* control systems, peers are of the same stature and execute the same algorithms, like in the Pure P2P case. However, specific operations (e.g. authentication, indexing, searching, etc.) are executed in a centralized manner (e.g. central authentication, indexing and searching server ala Napster[17], AudioGalaxy[3]).

*Hybrid* P2P systems ([27]) are a median between these two architectures. Peers are not all equal; a subset of the peers' population (e.g. Super-peers in FastTrack-based[11] implementations, Ultra-peers in LimeWire[15], etc.) is assigned special tasks (e.g. indexing and searching). Selection of such peers is usually based on processing capabilities and available network bandwidth.

One of the main issues in P2P systems is the routing of (search, join, leave, etc.) messages. Pure P2P systems usually deploy simple implementations for the routing of their protocol-related messages, often relying on some form of broadcasting or network flooding. This, however, imposes a huge load on the network due to the traffic caused by the great amount of packets circulating around the net.

On the other hand, centralized and hybrid P2P systems rely solely on the TCP/IP protocol stack for the routing of messages. Since searching is executed at a central server (or specific super-node), clients only contact a single server/node, get multiple <*file-id, node-id*> pairs in response to their queries, and directly contact other nodes for downloading. This solution removes almost all search-related traffic from the network, but introduces single-points-of-failure, while lacking an intelligent routing scheme.

Bleeding-edge routing schemes are based on the notion of *distributed hash tables - DHTs* (e.g. Tapestry[29] (using a variation of Plaxton trees [18]), Pastry[9], Chord[23], and CAN[19]). These schemes provably scale better than mere flooding; for $n$ nodes in the system, [18, 29, 23] all maintain $O(log(n))$ neighbors and route in $O(log(n))$ hops, [9] needs approximately $(2^b - 1) * log_{2^b}(n)$ neighbors and routes in $log_{2^b}(n)$ hops (where $b$ is a configuration parameter with typical value 4), while [19] maintains $2d$ neighbors and routes in $(\frac{d}{4}) * (n^{\frac{1}{d}})$ hops (where $d$ is the dimensionality of the d-torus used). Moreover, [24] proposes a layered paradigm for architecting P2P networks that routes in $O(log(log(N)))$ hops and is very efficient with regards to the dynamics of contemporary peer-to-peer networks.

A rather different approach has been proposed in [25]: peers are assigned into clusters, according to the content they contribute to the system. The system's architecture ensures that (a) load is fairly distributed across and within clusters, and that (b) routing can be performed utilizing routing indices (e.g., for neighboring searches) and/or metadata to route directly from cluster to cluster, thus guaranteeing low hop-counts and short response times. This is also one of the first works to deal with the issue of load balancing in highly dynamic P2P networks.

Keep in mind, though, that all solutions presented so far depend on the computing and power consumption capabilities of current computer systems. We'll show that (i) in the nano-world of P2P in ubiquitous computing and sensor networks, such methods are not directly applicable, due to the restrictions imposed by the very nature of the participating devices, although (ii) many of the problems inherent in the computer-world solutions can be leveraged to simulate issues arising when dwelling in the nano-world.

In the field of Sensor Networks, there has been significant research in the last few years, focusing on power-aware routing protocols (for a survey on sensor networks, consult [2]). [13] proposes SPIN, a family of information dissemination protocols that try to avoid flooding by using negotiations between sensors to ensure that data is transmitted only when necessary. [14] presents a data dissemination paradigm (called directed diffusion) where sensor data is sent towards nodes explicitly requesting it along paths of a low-latency tree.

At the MAC level, [21] presents a contention-based protocol that avoids overhearing among neighboring nodes, thus reducing node idle listening. Recently, [28] further avoids any out-of-channel signaling, and also introduces trade-offs of fairness for energy savings. [7] presents a set of smart dust information propagation protocols that are both energy and time efficient, along with a rigorous analysis of their performance.

Note, however, that all or most of these works focus mainly on energy consumption and not on time efficiency issues, and describe mainly protocol design and technical specifications, lacking any theoretical analysis. Furthermore, the routing solutions we discuss later (i) don't assume global network topology information or geolocation information, and (ii) use very limited control message exchange, thus having low communication overhead.

## 3 Life at the nano-level – Smart Dust

We shall now proceed with a quick architectural overview of Smart Dust systems. We'll also present two simple energy- and hops- efficient protocols for local detection and propagation of information in such systems, along with an analysis of their performance. The goal of this section is to act as a quick introduction to the concepts and limitations inherent in sensor networks for readers not familiar with such issues, and to give a first glimpse of the things we can expect when we dwell in the nano-world of Smart Dust.

### 3.1 Overview of Smart Dust systems

Smart Dust ([6]) is comprised of a large amount of ultra-small sensors, called “grain” particles; homogeneous, fully autonomous devices – as far as computing and communication issues are concerned – characterized mainly by their available power supply (battery) and the energy consumption of computation and communication tasks.

Each such particle features a set of (light, pressure, humidity, etc.) sensors, two communication modes: a *broadcast beacon mode* (implemented using digital radio, for low energy - short signals) and a *directed to a point mode* (implemented using a laser beam), and two modes of operation: *awake* (normal operation) and *sleeping* mode (all sensors and communication links are off). We’ll assume particles have no storage capacity, although there exist variations with limited memory support.

Smart Dust particles monitor various events and send observations to a central location. This may be a “special” particle (called the “sink”), or a series of base stations–control centers (called the “wall”). Since particles may be far away from their “wall” or “sink”, they cooperate with each other in order to propagate information closer to the base stations. We adopt here the simple but realistic Smart Dust Cloud (*SDC*) model, first presented in [7]; an SDC is a set of particles spread in a two-dimensional plane, plus a “wall” $\mathcal{W}$; an infinite (or of appropriately big length) line in the smart-dust plane, having high computing capabilities and a constant power supply (see fig. 1). Note that this model can be considered a generalization of the “sink” model, since any particle close enough to the “wall” can function as a “sink”.

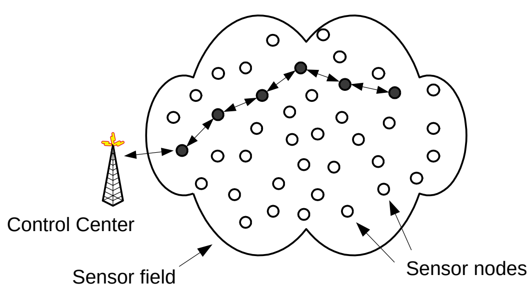


**Figure 1. Smart-Dust Cloud**

### 3.2 Some Results in NanoPeer Networks: Routing

As already stated, a hot-spot in NanoPeer networks is the problem of efficient propagation of information, on the realization of an event, detected locally by a particle. We shall now present, and analyze two simple protocols for local detection and propagation of information. The protocols presented have been experimentally evaluated ([7]) and the results indicate that they are very efficient in terms of both time and energy.

#### 3.2.1 The *Local Target* Protocol: A First Approach

The first information propagation protocol (called the “*local target*” protocol or *LTP*), also proposed in [7], basically consists of three phases: *searching*, *direct transmission*, and, optionally, *backtracking*, recursively repeated at every particle along the way to the wall. We assume that particles have a sense of direction and are therefore aware of the general location of the wall.

In the *search* phase, the particle having the information tries to discover, using the *broadcast beacon mode*, a particle nearer to the wall than itself[3]. The search is performed within a cyclic sector of radius $\mathcal{R}$ (the transmission range of particles) and of angle $\alpha$ (see fig. 2).

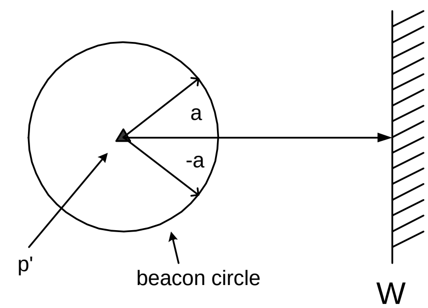


**Figure 2. LTP – Search Phase**

When an appropriate particle is found, information is passed to it using a *direct transmission*, using the particle’s laser beam. If, however, at some point, consecutive searches fail to discover such a particle, the protocol performs a *backtracking* operation (i.e. information is sent back to where it was received from), as do messages from single-neighbor nodes in a real-world P2P system (e.g. Gnutella)[4].

#### 3.2.2 The *Min-Two Uniform Targets* Protocol

Assume now that the search phase always returns *two points* $p''$, $p'''$, each uniform in $(-\alpha, \alpha)$, and that we select the best of the two points, with respect to the local (vertical) progress. We call this second protocol the “min-two uniform targets” protocol (or *MTP*). We’ll show that, even with this small change, the gain in efficiency is significant.

[3] Note that, in the worst case, a particle initiating a transmission will be able to contact only one other particle, located in the opposite direction of the wall. In this case, we choose this only particle, since propagating information backwards is better than not propagating it at all...

[4] In case of a segmentation of the smart-dust plane in communicationally disjoint regions, propagated information is finally dropped, much in the way messages are lost in conventional P2P systems in the case of a network segmentation. We assume, however, that this situation is highly improbable in a real-world placement of particles.

#### 3.2.3 Analysis / Upper Bounds

In the analysis of our protocols, we'll use the following metric, called the "hops" efficiency, which characterizes both the energy consumption and the time needed to propagate information to the wall.

**Definition 1.** *Let $h_{opt}(p, \mathcal{W})$ be the (optimal) number of "hops" (direct, vertical to $\mathcal{W}$ transmissions) needed to reach the wall, in the* ideal *case in which particles always exist in pair-wise distances $\mathcal{R}$ in the vertical line from $p$ to $\mathcal{W}$. Let $\pi$ be* a smart-dust propagation protocol, *using* a transmission path *of length $L(\pi, p, \mathcal{W})$ to send info about event $\mathcal{E}$ to wall $\mathcal{W}$. Let $h(\pi, p, \mathcal{W})$ be the number of hops (transmissions) taken to reach $\mathcal{W}$. The* "hops" efficiency *of protocol $\pi$ is the ratio:*

$$C_h = \frac{h(\pi,p,\mathcal{W})}{h_{opt}(p,\mathcal{W})}$$

To enable a first step towards a rigorous analysis of smart dust protocols, we assume that: **(i)** the search phase always finds a $p''$ (of sufficiently high battery) in the semicircle of center $p$, in the direction towards $\mathcal{W}$, **(ii)** the position of $p''$ is uniform in the arc of angle $2\alpha$ around the direct line from $p'$ vertical to $\mathcal{W}$, and **(iii)** each target selection is random *independent* of the others, in the sense that it is always drawn uniformly in the arc $(-\alpha, \alpha)$. In a more general case, we assume that when $p$ searches in the sector $\mathcal{S}$ defined by angles $(-\alpha, \alpha)$ and radius $\mathcal{R}$, another particle $p'$ is returned *in the sector* with some probability density $f(\vec{p'})d\mathcal{A}$, where $\vec{p'} = (x_{p'}, y_{p'})$ is the position of $p'$ in $S$ and $d\mathcal{A}$ is an infinitesimal area around $p'$.

By considering the sequence of points towards the wall generated, by estimating the (vertical) progress done in each hop, and by using Wald's equation for the expectation of a sum of a random number of independent random variables, we get that:

**Lemma 1.** *The expected "hops" efficiency of the LTP in the $\alpha$-uniform case is $E(C_h) \simeq \frac{\alpha}{\sin \alpha}$, for large $h_{opt}$. Also $1 \leq E(C_h) \leq \frac{\pi}{2} \simeq 1.57$, for $0 \leq \alpha \leq \frac{\pi}{2}$.*

Moreover, in the case of MTP[5], we can prove that:

**Lemma 2.** *The expected "hops" efficiency of the MTP, for large $h$ and for $0 \leq \alpha \leq \frac{\pi}{2}$, is $1 \leq E(C_h) \leq \frac{\pi^2}{8} \simeq 1.24$ (translating to a 21.5% gain w.r.t. the $E(C_h)$ of LTP).*

Note that both LTP and MTP require no memory at the particle (versus the quantities required by DHTs and other routing techniques), and route in at most 57% and 24% more hops respectively than in the ideal case[6]. Almost all computer-world routing schemes are inappropriate for this class of applications, since a hop in the overlay may be multiple hops at the physical level, translating to high power consumption in the nano-world.

[5] By stochastic dominance arguments, in [7] we estimate tight upper bounds to the hops distribution of a general target protocol.

[6] Experiments ([7]) have shown that $E(C_h)$ converges to 1 for more than 4 targets returned by the *search* phase of our protocol.

## 4 Smart Dust: a P2P world

We'll now argue that Smart Dust particles act as lightweight peers in a P2P overlay of their own and discuss the applicability of PC-world techniques and protocols in this context.

An SDC is, by definition, a pure P2P network; all "grain" particles are of the same stature, completely autonomous, and they all perform the same tasks (i.e. there is no specialized functionality at any particle). All communication is symmetric and the overall system operates in a completely self-organized, decentralized manner, since all operations are performed at every particle independently of the "wall" and the communication links are highly dynamic. The "wall" acts as an *information sink*, gathering observations from all particles. From a P2P point of view, we could model the "wall" as a *crawler* of the P2P overlay ([10]), visiting every and each node, or a set of super-peers, collecting all relevant information from the underlying nano-P2P layer and manipulating it appropriately.

This approach gives a new insight into Smart Dust systems and ubiquitous computing in general: we can simulate particles/entities with lightweight nodes in a P2P overlay, with arbitrarily restricted computing and power consumption capabilities. This, apart from providing us with a whole new testbench for protocols and techniques in the context of ubiquitous computing, allows us to reach a better understanding of novel P2P applications, utilizing lightweight and embedded devices.

### 4.1 Critique: intersection of P2P worlds

What happens if we try to apply computer-world methods and protocols in the P2P world of SDCs? As already stated, existing routing/communication protocols are *not* appropriate for such devices.

First of all, available processing capabilities (i.e. CPU, memory, etc.) are very restricted when it comes to "grain" particles. DHT-based routing protocols require a minimum of O(log n) neighbors, while flooding translates to high overall power consumption. Remember, moreover, that Smart Dust systems are usually at the extreme of having practically *no* memory at all.

Second, particles are highly dynamic; they may "sleep" or fail at will. This, coupled with the restricted available memory, gives rise to new problems. For example, in a SDC, neighbor discovery must be done at every search operation, also taking into consideration power consumption issues.

Last but not least, the connectivity of particles is also restricted; a particle can contact only particles lying within a specific area. For one of these NanoPeers to access another peer outside its area of coverage, relaying-like solutions are required ([7]).

### 4.2 Smart-Dust and P2P applications: parallel universes?

As pointed out in relevant literature ([2]), some of the main factors influencing the design and deployment of a sensor network, include fault-tolerance, scalability, hardware constraints, the network topology, the transmission media, and power consumption (as dictated by communication and data processing requirements). Sounds familiar? Let's take a step deeper and have a look at the dominant characteristics of SDCs and sensor networks (SNs) in general:

- The number of sensor nodes in a SN is very high – several orders of magnitude higher than the nodes in simple ad-hoc networks.
- Sensor nodes in a SN are deployed in a quite dense manner – as high as 20 sensor nodes per $m^3$. This, coupled with the fact that SN nodes mainly use broadcast communication, with a transmission range in the order of some meters, gives us a very dense, high-outdegree graph of sensor nodes and their interconnections.
- SN nodes are prone to failures of any kind, ranging from a simple power failure to the complete destruction of nodes by external factors (e.g. hostile action, dire environmental conditions, etc.).
- Sensor nodes have limited computing capabilities and power consumption capacity. For example, SDC particles may be at the extreme of having no memory at all.
- Bandwidth resources are also scarce. SDC particles are usually equipped with transmitters with transmission rates in the order of tens of kbps, although it's possible to use faster but greedier, with regards to energy consumption, transmitters (e.g. Bluetooth can achieve a 1Mbps transmission rate, for an energy consumption high enough to prevent it from being used in SDC).
- The topology of SNs may change very frequently, especially when sensor nodes are attached to mobile objects, or when they are deployed in an open environment, with multiple moving obstacles in the way. Node failures also result in topology changes, albeit these are permanent ones.

Compare the above to the status-quo of the Gnutella peer-to-peer network overlay:

- The number of nodes in the Gnutella network is very high – several orders of magnitude higher than nodes in traditional distributed systems (e.g. Mosix [16], Beowulf [5], etc.).
- Nodes in Gnutella are interconnected according to a power-law topology, following the small-world paradigm. Due to the quasi-complete connectivity of the underlying TCP/IP network, Gnutella nodes may have multiple neighbor nodes[7]. Moreover, Gnutella also uses broadcasting (flooding) techniques to propagate information through the overlay.
- Gnutella nodes are selfish ([1]); nearly 70% of the nodes share no files with the rest of the community. Without loss of generality, we can models such peers as computing entities of ultra-low computing capabilities, much like nodes in a sensor network.
- Following the above distribution, most Gnutella nodes have very low connection speeds (in the order of tens of kbps), while there do exist some (but few) nodes with high-bandwidth connections (e.g. 1Mbps lines).
- "Free-riders", are usually users entering the Gnutella overlay over dial-up modem lines, and may exit the overlay without prior notice (e.g. due to a modem hang-up). This makes such nodes highly volatile and results in a frequently changing network topology. For example, [26] has shown that the half-life – the time required for half of the peer population to be replaced due to joins and leaves – of MojoNation ([4]) was less than one hour!

The similarities of sensor networks and pure P2P computer-world systems extend well beyond the above mentioned characteristics. For example, Gnutella entered the realm of hybrid P2P systems, with the introduction of "UltraPeers" by LimeWire; in about the same time, researchers in the sensor networks' field proposed a "backbone"-based architecture ([8]), where some "higher-order" sensor nodes were injected into a sensor network and took over the communication tasks.

However, as already mentioned, there exist more than a handful of differences between the two worlds. For one, computers have no energy limitations (other than that they rely on the public electricity network's being stable) and their storage capacity and computing capabilities are growing at a quasi-exponential rate. SDC particles, bounded by the limitations in size and energy, don't (and probably will never) have access to such equipment as a multi-gigabyte hard disk or a multi-gigahertz CPU or a multi-megabyte RAM.

[7] A snapshot of the Gnutella network taken on October '00 showed that nodes had on average 5.5 neighbor nodes, while there also were nodes with as many as 136 neighbors!

## 5 Smart-Dust NanoPeers: one step further

So far, the SDC model features a single "wall" or "sink". We hereby propose some extensions to this model, borrowed from the computer-world P2P experience.

### 5.1 Multi-"wall"/"sink" Smart Dust systems

Computer-world P2P systems use a set of servers, scattered around the net, to handle authentication and bootstrapping of new nodes. Thus, we can imagine a SDC with multiple walls (e.g. particles "hovering" in a cube, whose all six sides are walls). Note that the SDC model assumes that all particles find out about the position of their wall during bootstrapping.

Multiple-sink systems have been also studied in [14], although their system is pull-based; the walls send out queries towards an area of interest – thus creating (possibly several) path(s) of hops on the smart dust plane – and replies are piggybacked on this very path. What happens, though, if we are interested in a push-based system (i.e. in a system were it's the particles that decide when they have something important to say to the rest of the world)?

A first naive solution to the multiple-wall problem is for the wall(s) to inform particles of their existence, in a P2P recursive manner: every wall registers with all particles within reach, and registration information is propagated recursively, using the inexpensive digital radio transceivers. Particles are then responsible for selecting the wall that is closer to them.

### 5.2 P2P Worlds: A Hybrid model

A better and more scalable solution to the problems that may arise in the SDC P2P world, would be to have a Hybrid P2P system, consisting of heterogeneous particles, with escalating processing capabilities, network bandwidth, area of coverage, and power supply, allowing for multiple levels of peers.

In this scenario, every set of homogeneous particles would form a separate *Smart Dust Layer* (or *SDL*). Higher order SDLs[8] would then act as "walls" for lower order SDLs, with the actual wall(s) being seamlessly incorporated in this model. Imagine such a world were micro-peers coordinate the operation of nano-peers within their area of coverage, milli-peers coordinate micro-peers etc.

Particles would then contact the higher-order particle that is closer to them (discovered via the *broadcast beacon* mode). To go one step further, we can have particles use only the low-consumption digital radio transceiver to broadcast observations, under the virtual "umbrella" of one or more higher-order particles, much in the way overlapping GSM cells operate.

[8] That is, SDLs consisting of particles with higher computing and power consumption capabilities than lower order SDLs.

### 5.3 An example application of P2P worlds

**Parcicles: Trails on Smart-Dust**

Suppose that $Very\ Important\ Parcels^{TM}$ ($VIP^{TM}$), a (quite imaginary) major postal delivery firm, implants in every parcel shipped a *parcicle* - a low-cost, cut-down version of a conventional particle (i.e. a NanoPeer), with all sensors stripped-off, featuring only the communication equipment, plus a unique id number (given to customers along with their receipts)[9].

Imagine that each box carrying parcels features a "higher-order" particle (i.e. an MicroPeer), or a set of such particles, equipped with a better battery and some memory. Let's assume that more such particles (i.e. MilliPeers) are spread around $VIP^{TM}$'s warehouses, with each warehouse having a set of servers (e.g. one per department), and with all of these servers across all warehouses being interconnected in a P2P network. Assume that *parcicles* periodically register with their box's particle(s), with registration information being propagated and cached all the way up to the local servers.

Suppose now that a customer of $VIP^{TM}$, wishes to track down a parcel's current position. She would then utilize the all-secure $VIP - Tracker^{TM}$ P2P software to query the servers' P2P overlay for the one having last seen her parcel. Verification of the parcel's actual position could be done on-the-fly, with this last server sending an "are-you-there" query all the way down to the parcel's *parcicle* (or to the particle caching the *parcicle*'s registration, should the latter be asleep). Note that there is both horizontal and vertical communication, within and across the layers of the architecture, in order to register information and to answer queries.

## 6 Conclusions and Open problems

We have presented NanoPeer networks: a novel and highly interesting P2P application domain. We have argued for the similarities and differences between nano-world nodes and computer-world peers. We have included some relevant research results for the problem of routing at the nano-level, which helps us better understand this largely unstudied P2P environment. We have analyzed the fundamentals of the intended functionality, looking at them from the point of view of P2P computing, and have proposed some initial (albeit naive) solutions to the problem of building a set of SDCs (or SDLs), following the paradigm of

[9] Credits for the original *Inventory Control* idea go to [22].

Hybrid P2P systems, and of locating higher-order SDC particles.

The proposed framework offers several formidable challenges in tackling open research problems. Significant open problems include:

1. Fault-tolerance & reliability of current routing algorithms in the NanoPeer P2P world should be further examined and improved, in order to provide the infrastructure for the implementation of the proposed extensions.

2. The efficiency of current routing schemes must also be improved. Analysis is required to define the improvements in order for it to be applicable at the nano-level, paving the way for a whole new class of P2P applications.

3. So far we have assumed that particles have no memory. Even with the addition of some memory, however, the memory requirements of protocols and techniques must be within the restrictions of NanoPeers. Thus, compact data structures must be developed for event memory/storage and communication for these memory-enabled NanoPeers.

Our position is that the proposed frameworks of NanoPeers and P2P worlds are worthy of further investigation. They offer several formidable challenges in tackling open research problems, such as the fault-tolerance & reliability of current routing algorithms in the NanoPeer P2P world, the efficiency of routing schemes for horizontal and vertical communication in P2P worlds, and compact data structures for memory-enabled NanoPeers. Moreover, due to their low deployment and maintenance cost, carefully-chosen computer-world P2P systems comprise a first-class testbench for nano-world techniques and solutions.

We intend to pursue the above issues, and in general the issues arising from the study of P2P worlds as a multi-layered architecture of P2P environments communicating with one another. We call on the community to join us in this endeavor.

## References


[1] E. Adar and B. Huberman. Free riding on Gnutella. Technical report, Xerox PARC, 2000.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[3] AudioGalaxy. http://www.audiogalaxy.com/.

[4] Autonomous Zone Industries. Mojonation. http://www.mojonation.com/.

[5] Beowulf. http://www.beowulf.org/.

[6] Berkeley Wireless Research Center. http://bwrc.eecs.berkeley.edu/.

[7] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. Smart Dust Protocols for Local Detection and Propagation. In *Proc. of ACM POMC '02*.

[8] I. Chatzigiannakis, S.Nikoletseas, and P.Spirakis. Distributed communication algorithms for ad-hoc mobile networks. *JPDC, Special Issue on Mobile Ad-hoc Networking and Computing*, 2002.

[9] P. Druschel and A. Rowstron. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of IFIP/ACM Middleware '01*.

[10] Endeavors Technology Inc. Magi P2P. http://endeavors.com/.

[11] FastTrack. http://www.fasttrack.nu/.

[12] Gnutella. http://gnutella.wego.com/.

[13] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. of ACM/IEEE MOBICOM '99*.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. of ACM/IEEE MOBICOM '00*.

[15] LimeWire. http://www.limewire.org/.

[16] Mosix. http://www.mosix.org/.

[17] Napster. http://www.napster.com/.

[18] C. Plaxton, R. Rajaraman, and A. Richa. Accessing nearby copies of replicated objects in a distributed environment. In *Proc. of ACM SPAA '97*.

[19] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of ACM SIGCOMM '01*.

[20] Seti@Home. http://setiathome.ssl.berkeley.edu/.

[21] S. Singh and C. Raghavendra. Pamas: Power aware multi-access protocol with signaling for ad hoc networks. In *ACM Computer Communication Review*, volume 28, no.3, pages 5–25, July 1998.

[22] SmartDust. robotics.eecs.berkeley.edu/˜pister/SmartDust/.

[23] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proc. of ACM SIGCOMM '01*.

[24] P. Triantafillou. Peer-to-peer network architectures: The next step. In *SIGCOMM FDNA'03 – Invited talk/presentation*. (http://www.ceid.upatras.gr/faculty/peter/papers/fdna03.pdf)

[25] P. Triantafillou, C. Xiruhaki, M. Koubarakis, and N. Ntarmos. Towards high performance peer-to-peer content and resource sharing systems. In *Proc. of ACM SIGMOD/VLDB CIDR '03*. (http://www.ceid.upatras.gr/faculty/peter/papers/cidr03.pdf)

[26] B. Wilcox-O'Hearn. Experiences deploying a large-scale emergent network. In *Proc. of IPTPS '02*.

[27] B. Yang and H. Garcia-Molina. Comparing hybrid peer-to-peer systems. In *The VLDB Journal*, Sept. 2001.

[28] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. of IEEE INFOCOM '02*.

[29] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, University of California at Berkeley, CSD, 2001.