# Ad Hoc Disposable Networks and Networked Applications

**Dr. Rick Schantz, Dr. Joe Loyall, Dr. Jason Redi**
**BBN Technologies**
**Cambridge, Massachusetts**
**{schantz, jloyall, jredi} @bbn.com**
**February 3, 2001**
**Author Point of Contact: Rick Schantz, BBN Technologies, 617-873-3550, schantz@bbn.com**

**Abstract:** The technology behind Ad Hoc Networks and Networked Applications represents a revolutionary leap forward in moving digital capabilities out of the "computer" and into the everyday world. We speculate on a concept where large numbers of digital nanonodes are inserted in common products during manufacture (such as medicines, building materials, seeds) to be activated and organized later during the lifecycle of the carrying entity. Some of the technology topics raised from this concept include miniaturization and environmental compatibility and safety, as well as new styles of networking and self-organizing behavior. We provide a forecast of available components ten years hence, speculate on a variety of possible application areas, and raise a number of technology barriers deriving from the concept, which are appropriate for future R&D activities.

## 1. Motivation and Vision

Imagine swallowing a liquid medication with embedded computational elements and having the "corpuscles" work together for disease recognition and fighting, controlled by interacting elements exchanging localized information. Imagine miniaturized computation and communication capability embedded in seeds, for use years later after planting in a field and after many have grown into the forms of life they represent. Imagine computational elements with built in localized communication capabilities implanted during manufacture in the concrete and steel and wood products which compose new construction, and which are later activated after construction is complete (and on occasion renovated) to perform a variety of integrated computations regarding the safety and features of the facility. These are all examples of one of the shapes for networked systems of the future. We call the technology concept underlying such configurations *Ad-hoc Disposable Networked Systems*, and the individual units *nanonodes*. They embody a new set of challenges to the technical community, as well as revisiting some old ones with new constraints. The issues are wide ranging from miniaturization, to manufacturing and environmental packaging, to new issues and constraints on the communications, software, software engineering, algorithms, designs and techniques for organizing the networked computations. The need also exists to revisit and extend or replace some of our current ideas in familiar, recurring topics such as distributed control, composition, security and validation.

In our view, the networking challenge which these Ad-hoc Disposable Networked Systems pose is concentrated on four areas:

- extensive miniaturization and compatibility with the specific environmental and manufacturing processes which deliver them
- self-organizing, specializing and composing at runtime large sets of general purpose elements many of which may be faulty or malfunctioning or being held in reserve
- long shelf life and changing requirements well after the manufacturing stage limitations have been set
- programmability and safety considerations, within the limited computational infrastructure available

## 2. Prevailing Underlying Technology Base Forecast and Trends

Designs of systems of very large numbers of interconnected, inexpensive nanonodes will be critically dependent on the individual node capabilities and characteristics available circa 2010. If we imagine capsule or grain sized nodes, we immediately imply non-rechargeable (primary) power sources, highly integrated circuits, and minimal requirements of off-chip components.

Power constraints will be the primary driver of future nanonode design. Battery technology is a wide and mature field, providing relatively small gains of only doubling capacity in the past 30 years. Only modest performance gains are expected in the future, and these will be dramatically determined by consumer demand for particular battery chemistries (especially Lithium based batteries due to their wide use in secondary cells). As described in the

National Research Council's Report on Energy-Efficient Technologies for the Dismounted Soldier, there are only a few chemistries of primary cell that meet the safety, temperature variation, and reliability of the soldier. These metrics are equally important for our tiny nodes, so the data provided is pertinent. The report mentions Zinc-air primary cells as having the greatest energy density for a technology where the theoretical limit is close to the effective, working limit. The theoretical limit for Zinc-air is 1,066 Wh/kg. If we assume that the battery available to our nodes can only be one gram in weight, this leaves us with about 1 Wh in power available.

In contrast to the battery industry, the microprocessor industry has been advancing at a dizzying pace. However, with the increase in processing power (via increased number of transistors) has come an increase in the power consumption of these devices. As an example, the International Technology Roadmap for Semiconductors (1999), predicts that in 2005, the minimum feature size in CMOS logic will be 100nm, allowing 190 million transistors on a typical chip, and using 160W for desktop chips and 2.4W for portable chips. This is in contrast to the 180nm, 100 million transistor, 90W desktop, 1.4W portable chips of today. Note that the desktop and laptop industries can accept the power requirements for the demands in processing due to the effectively infinite power sources available.

We would like to determine exactly how much computation power can be expected to be available for a given amount of energy. Although the MIPS (millions of instructions per second) metric is a poor gauge of performance for a general purpose processor due to an informal definition of "instruction", we expect our nanonodes to be somewhat specialized and composed of a limited number of instructions, similar to today's DSPs. Texas Instruments is currently estimating that their DSP cores will require only 0.01 mW per MIP. As a sanity check on this value, The National Research Council Report states that 0.01mW per MIP will be reached by general purpose processors by 2015. If we assume as a starting point that we would like our nodes to run for a year and they will be in an active state for 25% of the time, and only use 5% of the power for the computation (the rest for communications), then we can expect a main processor of (0.05 Wh / (0.25 * 168 hours/week * 52 weeks * 0.00001 W/MIPS =) 2 MIPS, or approximately the computation power of a single Z-80 processor from 25 years past.

Using the computational abilities of any nanonode requires a mechanism for exchanging data with other nodes. Despite the vast expansion of the wireless industry within the past few years, the fundamental physics of antenna power requirements and amplifier efficiency are generally unchanged. Most modulation schemes rely on operating non-linear devices in a linear region and have an efficiency of 20 to 30%. As described in the National Research Council report, we may see an improvement to 50 percent or more, but the efficiency of 100% is probably unattainable. By 2015 we should see efficiencies of up to 60% overall. Today's most advanced, mid-cost, low-power short range transceivers are designed for the Bluetooth specification. Matsushita is producing a transceiver that uses 50mW in receive mode, 60mW in transmit, and 0.9mW in standby for a 1Mbps data rate designed to transmit over 1 meter (10^-6 Bit error rate). For a given distance and CDMA-like modulation scheme we can roughly assume that to transmit at half the data rate would require half the power. If we have a xmt / rcv / standby ratio of 5% / 20% / 75% when our processor is in the active state we will be able to support a data rate of about 20kbps.

In summary, by approximately 2010 we expect an achievable building block nanonode to be about a two gram package consisting of a battery, a Z-80-like capability processor and a 20kbps link that operates at 1 meter. This nanonode would be able to last a year assuming intelligent power cycling as described, with other tradeoffs possible (see below).

Predicting natural evolution of the software environment that these components will operate on and within is even more difficult because there are fewer physical processes to constrain the direction. Constraints often are of the human and market oriented variety. However, one can look back at the evolution of and the steady improvements to the operating environments for developing communication oriented system software and applications. Unlike hardware, however, this improvement comes with steadily increasing software (footprint) costs, requiring more capable engines to execute at these higher levels of abstraction and taking into account the ever increasing requirements and expectations. We expect to see scaled down versions of the various system and protocol layers now in use (many have withstood the test of time and remain largely unchanged in excess of 20 years already) plus additional "must have" layers oriented to new concerns such as safety and failsafe behavior absent from mainstream networking systems today.

Software has without a doubt become more complex and applications have become larger, in the sense of greater functionality, more features, and the amount of resources they consume. However, this has been accompanied in many cases with an accompanying decrease in the maintainability, reusability, flexibility, and security of the software. Many of these trends toward feature- and bug-filled software have been driven by time-to-market pressure and releasing new versions of software too frequently. In addition, greater hardware capabilities have enabled many significant but dormant software ideas to reappear as machines, environments, and domains emerged that could finally afford and effectively utilize them. The most recent example of this is Java, whose virtual machine bytecode technology, object oriented design support and garbage collection all existed in earlier versions in Pascal, Smalltalk, Lisp, and other languages. The earlier versions were hindered by their lack of performance. Java has not fixed the performance issues inherent in these earlier incarnations and is still suspect for use in constrained environments. However, the emergence of much more capable hardware, applications that are dominated by wide-area network latency, and the Internet environment's heterogeneous platforms have made the time right for the advantages that exist in Java and have lessened the performance implications, at least for high power environments.

Similarly, we can expect that in ten years, some, if not many, of the technologies that have been researched for years will reach a level of maturity and a form that makes them standardly applicable to the platforms and environments common at that time. We expect that this will include more advanced forms of higher level programming paradigms, such as distributed object computing; some form of abstraction and reuse capability, such as component and middleware technology; and highly optimizing compiler technology generating code optimized for specific application domains. We can also expect that research in other areas will find its way into these accepted technologies, such as software specification and analysis techniques that facilitate compilation and optimization or artificial intelligence techniques that facilitate middleware services and adaptation. But each of the improvements comes at an additional infrastructure cost, and moves the software developer further and further away from the actual platforms on which he is operating. This combination of expectations for higher levels of abstraction for common off the shelf software engineering support, alongside the constrained platform support due to environmental and physical limitations, are in direct conflict. This conflict serves as background to new R&D in the area of ad-hoc, disposable network technologies.

We note that this forecast of the hardware and software building block technology available for our future nanonode capability is based upon *expected* linear innovations in design, fabrication and off-the-shelf software infrastructure capabilities. An *unexpected* change in underlying primary cell or transistor capability, or even software engineering tools might make this level of capability available significantly sooner. However, since radical improvements will also require new methods and skills, with new and unknown constraints, and the time to field such changes into large scale production and common use is many, many years, we believe that using the constraints from our forecast is a safe estimate (it might be sooner or faster) and has the highest probability of yielding useful research results in this time frame.

## 3. Some Potential Examples

The following paragraphs describe uses for sets of tiny, programmable devices that construct into ad hoc networks to organize and control their behavior. These devices would need to have a reasonable shelf life and be programmable at the point that they are deployed or activated for a particular use, not at manufacture time. They would also need to have properties consistent with the physical environment of their manufacture and use.

### *Healthcare, medicine, battlefield medicine*

*Non-invasive discovery or treatment.* The patient swallows a set of tiny devices embedded in a solution. They establish a network and work their way through the digestive system (and, if tiny enough, into the bloodstream). Some of them can work as navigation or locating devices. Others work as camera devices sending images back to a doctor. The doctor can manipulate yet another set of them to perform stomach, intestinal, or urinary procedures.

*Directional, homing, and time-released medication.* There are already medications that use materials that react or dissolve at different rates or in contact with different natural substances, in order to target specific systems or to time-release medication. The technologies described in this paper are the digital equivalent, and would enable medications that could go well beyond these, to medication contained in intelligent delivery devices that would home in on particular organs or release medication based upon diagnosis that they conduct inside the targeted area. For example, consider a diabetes patient that has to inject insulin twice a day, each dose of which is based upon the blood sugar level at a given time. Now imagine that that same patient can inject or ingest a solution of insulin-delivery devices. Over the span of a week or a month, these networked devices check the blood sugar level at regular intervals and dispense the correct amount of insulin directly into the bloodstream. At the end of their useful life, the devices dissolve and are absorbed or expelled by the body.

*Medical diagnosis*. A surgeon performing complex surgery, such as open-heart surgery, joint restoration, etc. places a set of devices in the patient at the point of surgery. These devices monitor the healing process and alert the doctor of progress in healing. They would be able to gather and manipulate information about healing much better than external techniques, such as X-rays, MRIs, or EKGs. Some of these devices would need to degrade over time and be absorbed or expelled by the body, similar to the way dissolving sutures are used in certain surgeries. Others, similar to pacemakers, permanent sutures, and pins used in heart and bone surgeries, would need to be permanent or long-lived.

### ***Construction and manufacturing***

*Smart materials.* These types of devices can be installed or molded into the materials of buildings and structures. These devices can measure the stress or integrity of the material and aid in the construction process in several ways. First, they can measure, analyze, and communicate the strength and integrity of foundation parts of a building prior to the addition of floors. They can also provide information and analysis that currently are done manually by building inspectors and structural engineers, providing immediate feedback and/or approval of permits. They might even be used as part of a security or surveillance subsystem embedded in the construction material itself, and more difficult to avoid.

*Military construction.* These types of devices can also be molded into the materials used for the construction of tanks and aircraft. They can measure and communicate the integrity of the material, monitor external stresses, and serve as extra sensors for targeting or navigation. They can also interact with and control systems in the vehicle, e.g., deploying a fire-fighting system in response to a detected fire. These devices can also serve as sensors, routers, and hosts in a large vehicle-wide network, enabling the structure of the vehicle to become a large solar receptor or antenna in response to dynamic needs of the vehicle for additional power or communication.

### ***Agriculture***

Devices spread with seed or fertilizer can measure, monitor, and report on the health of a lawn or of crops. They can improve crop output by directing fertilization, irrigation, or reseeding to regions that need it the most, based upon analysis of the soil, the waterfall, or the crops. They can prevent catastrophic crop failure by detecting temperature and moisture fluctuations and the presence of destructive pests, and by dynamically deploying preventive measures, such as heat, moisture, and pesticides.

## 4. Some Important Technical Issues

### *Component design tradeoffs effecting approaches and techniques*

There are certain fundamental trade-offs discussed in our example nanonode design. Most significant is the percentage of power available to communication vs. computation. Communication is inherently a very inefficient operation due to the non-linearity of amplifiers and the typical $1/r^4$ or $1/r^2$ radio propagation losses. Although we cannot change the losses dictated by physics, one could imagine application solutions that require less power for communication. One example is the use of reflected power radiated by a nanonode “reader”, much the same way passive RF tags currently operate. This would require very little amounts of power at the individual node, though it would require a node “reader” to be within inches of the node itself. Since for most applications, we would often not know in advance which nanonodes were in which locations, we would require algorithms and data distribution that had vast amounts of redundancy. This redundancy could be in the form of similar sensing at each node, and little or no communication between the nodes themselves. For a relatively small reduction in communication capability, we can benefit from a very large increase in computational power. Alternatively, the gram-sized battery we hypothesize can be significantly reduced in size, particularly for applications where the nanonodes are needed to last for much less time.

### *Long shelf life*

If we would like to design nanonodes that can operate for much longer periods of time, we can either increase the battery size, or we can reduce the cycle time. By reducing the amount of time that a node is in the “on” state we reduce its average power consumption, but we increase the response time for either computation or communication. The amount of power per transmission or computation effectively stays the same, but we stretch that operation over a longer period of time to extend the battery life (e.g., extending “shelf life”). This provides the mechanism for the classic application trade-off between delay and energy. Allowing increased delay in applications needs to be carefully considered since there is certainly a point where data may consistently arrive too late to be useful, or computation time may not keep pace with new incoming data, effectively making all computations wasted. Alternately, we need to consider time triggered use of only portions of massively redundant elements, and protocols for transferring important capabilities from dying elements to fresh ones. Organizing and programming large collections of components, only some of which are being used at various stages of the lifecycle while others are held in reserve, represents an approach to longer lifetimes but a significant challenge to seamless operation.

### *Throwaway low cost*

We are considering these nodes in an environment where they are mass-produced in the order of millions or tens of millions of units. To manufacture this many items and keep the costs down to the absolute minimum, no customization will be possible (i.e. all customization comes through software). This means that no node will have a pre-programmed unique node number. Delivering data across a network requires the concept of a “destination” and possibly a “source”. Selecting unique identifiers as part of a network establishment and discovery phase is today only possible with a fairly small and consistent set of nodes. Without unique identifiers nanonodes might need to resort to another naming technique that either includes context (“whoever has this type of data”), capability (“whoever has lots of battery power left”), or location (“whoever is far enough away to have a more useful reading”). This suggests a level of redundancy and tolerance to errors or losses that need to be supported by applications. Additionally, this data delivery and redundancy needs to be provided without the expense of blindly flooding to all neighboring nodes.

### *Laws of large numbers to get working subset and coverage*

For applications with large numbers of nodes, we require not only point to point communications, but a complete adaptive network association of elements. Traditional networks (both mobile and non-mobile) are designed for low-delay, high-throughput, and attempt to be scalable. For a nanonode based network, we will have very high delay, very low-throughput, but need to be highly redundant, and highly scalable. More importantly, we will have a limited amount of computation time and memory space, so the complexity of all algorithms and storage needs to be tightly controlled. Combined with large numbers of redundant nodes, not all of which are active at any time, we see a new type of communication infrastructure that can support scalability and very low data rates, by allowing redundant, duplicate nodes, highly suboptimal routes, high tolerance to losses and errors, and very low speed adaptation to changes. Communication, coordination and control structures that support these parameters are a radical departure from today’s environments and the trends (e.g., high-bandwidth, large computational elements, centralized operation, human configured). Programming techniques for large scale, distributed redundancy based computing paradigms are as yet unexplored.

### *Self-organizing behavior*

Organizing and controlling the ad-hoc networking needed for the highly redundant, overlapping, semi-autonomous environments envisioned, within limited packages, is a new research challenge. Our current offline and pre-configured, centralized control and static role assignment development constructs will not stand up in such environments. Major enhancements in capabilities for dynamic behavior in areas of discovery, role assignment, asymmetric capabilities as replacements are made over time, distributed control, overlapping regions, changing components as some die off and others are energized in an environment with many small parts, will all be needed as prerequisites to successful use of these ad-hoc network constructs. Modification to these strategies well after the time of manufacture, as well as the potential for organized self-repair (like white blood cells), may also be important new areas of applied research for these domains.

### *Safety issues*

Before any of these concepts can be realistically contemplated, there are serious safety, validation and external control issues which need to be engaged in parallel with the development of techniques and strategies for organizing the nanonodes together. In some ways, these usage patterns merely exacerbate the similar unmet challenges from our current repertoire of development activities in not nearly so hostile or demanding environments. Considering approaches which have capabilities for neutralizing the elements long after deployment or activation (or even anti-dotes to counter behavior begun in error) may be part of new pre-deployment requirements.

### *Tailored and domain sensitive communication environments*

There are significant conflicts among requirements for ease of development, small efficient components, parallel and redundant architectures, reusing existing infrastructure solutions and software, and new modes of communication appropriate for the environment embedding these ad-hoc networks. There are also the usual special purpose, higher cost vs. general purpose, wider use issues for cost sensitive applications, which must be addressed when considering possible approaches. There are questions of whether and how to achieve some degree of external, human in the loop operation and control. While it may be feasible in some contexts, it may prove difficult or impossible in others, and will surely influence the complexity of any proposed solutions.