ISSN 2229-5984

# Bluetooth - Smart Nodes for Mobile Ad-hoc Networks


**Banta Singh Jangra**

***Assistant Professor***

*Deptt. of Computer Science,*

*HIT, Bahadurgarh (HR)*

***Correspondence to:***

***Banta Singh Jangra***

*bsjangra@gmail.com*



## ABSTRACT

The implementation and operation of wireless distributed, self-organized, large scale, mobile communication and information systems poses many interesting research problems such as blue tooth technology. While a lot of questions devoted to algorithmic and architectural aspects are already being pursued, few have actually deployed such systems to the extents envisioned. We can say Bluetooth Smart Nodes, each of which can store information, compute and communicate using standard wireless networks on a limited resource platform. Such types of wireless enabled small devices can interact in a heterogeneous environment consisting of different types of networking nodes as well as with other wireless enabled appliances. Important requirements and design tradeoffs to be able to support multiple communication interfaces, handle limited resources and power aware operation. The BTnodes are integrated into our MANET application and networking framework. Demo applications give an insight into usage scenarios envisioned for future architectural explorations.




**Introduction**

Mobile ad-hoc networks are an infrastructure less based networks and it have many attractive applications as computer systems and many communication devices. Handheld devices are no longer emerging technology but widespread. Especially mobile communication & wireless devices have opened new opportunities with applications like the SMS generating billions of interactions per year between independent devices.

Now the days the trends in price, performance and ever higher levels of system integration are pushing electronic devices into regimes that exceed visions and expectations alike. In contrast to the user centric approach of Computers, and infrastructure based systems like cellular telephony in the past years, the class of networked sensors are becoming a reality today that are actually empowering the environment to become "Best". And the Internet becomes ubiquitous, not the devices themselves will be dominating the field, but the network, applications and interactions. Interoperation between the most heterogeneous devices is thus an absolute necessity.

This requires infrastructure that can provide peer to peer services across many hierarchies of devices and in a distributed and heterogeneous world. In order to scale to populations magnitudes higher than present today, flexibility, compatibility, self assembly and a timely deployment are as important as low power and small size.

Nevertheless, to date only very select deployments of these ideas have been realized. Only very limited and actually few solutions for the implementation of ad-hoc networking system today are available. They are usually restricted in flexibility and the number of nodes in a single deployment that can actually interact with each other. Heterogeneity in existing solutions usually is limited to a maximum of a mobile unit and a base station. Interaction between hierarchies of devices can only be made possible if certain commitments are made and compatible standards are used, albeit their known deficiencies.

In order to be able to implement and analyze algorithms, networking principles and applications on a great number of mobile autonomous nodes the wireless prototyping testbed based on the miniaturized Bluetooth Smart Nodes (BTnodes) discussed in this paper has been developed based on standardized technology and prefabricated OEM macro modules. This makes it possible to focus on the higher levels of abstraction such as network, transport and session layer as well as applications and services. The utility of ubiquitous technology like Bluetooth and W-LAN is dominating clearly over the negative aspects such as the unacceptably high power consumption, considerable bulk and complexity of the implementations available today. To achieve the goal of large heterogeneous MANETs with all types of "smart" devices valuable insights gained from the deployment are going to have to drive the future development issues in lower level radio and base band hardware.

## 1.1 Design Ad-hoc Networking

In Ad-hoc networks configurations of hundreds and thousands of nodes certain requirements have to be considered. Mobility, scalability and most important self-organization are the most important of these requirements, while flexibility, low power consumption and small size are optimization goals in the design space as a smart ad-hoc networks devices.

Such networking nodes need to contain storage, computation and communication resources to account for the necessary flexibility and scalability in such an environment. Communication should not only be possible through one single type of interface with one front end per node but over multiple front ends per node enabling the exploration of network topologies at a higher degree of freedom without network arrangement limitations and interference. Many MANET applications, which are often embedded applications, stringent resource constraints apply. An optimal use of the available real estate in respect to system size, CPU cycles, and wireless communication bandwidth and battery capacity efficiently is crucial.

The use of standardized communication systems such as W-LAN and Bluetooth allows testbed components to communicate with other wireless enabled appliances such as cell phones, headsets, PDAs, laptops, keyboards, game controllers, printers, video equipment, mobile storage, media players, cameras and many others through well defined APIs. Moreover as the Internet becomes ubiquitous, wireless access will be available virtually everywhere either through broad coverage of W-LAN, cellular services. Without connection to the outside world, the use of such a heterogeneous networked system would be very questionable and most tedious, the possible deployment limited to an artificial scenario at a single lab location.

In the wireless networks the best operating system (Linux) and ready to use development tools (gcc, gdb and Java) certainly are desirable when researchers that are not experts in embedded architectures want to profit from such a testbed for their own purposes. Here, where MANET applications, mechanisms in the network, transport layer and above as well as network services are the intended application of such a framework, it is important to provide this flexibility.



## 2. Mobile Ad-hoc Network Testbed Components

The wireless testbed described consists of three different classes of networking nodes: A Maxi class based on a laptop PC, a Mini class based on a PDA and a Micro class based on an embedded microcontroller, the BTnode. These networking nodes each feature multiple communication interfaces as well as computing and storage resources equivalent to their class specification.

| Class | Cost ($) | Size (Cm3) | CPU Type | Frequency (MHz) | Storage (Mbyte) | Memory (Mbyte) | Bandwidth (Mbit/sec) |
|---|---|---|---|---|---|---|---|
| Maxi | 3000 | 3200 | Pentium4 | 2000 | 40000 | 500 | 108 |
| Mini | 1000 | 300 | XScale | 400 | 1000 | 64 | 11 |
| Micro | 110 | 12 | ATmega1281 | 7.3 | 0.128 | 0.064 | 0.768 |

**Table 1. Testbed component features**

Depending on the applications and the scenarios surely the one or the other platform has it's advantages. In selecting such a heterogeneous group we are able to investigate into the interplay of the very small, numerous and resource limited devices such as smart tags and wireless sensor network nodes with larger already common but more bulky equipment incorporating a host of resources, extensive monitoring, debugging and development support and sophisticated user interfaces such as PCs and PDAs.

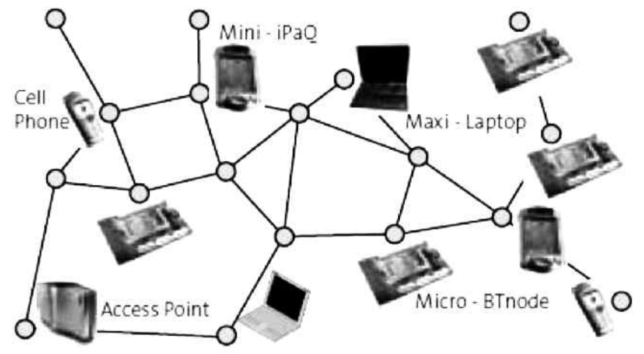


**Figure 1. Different scales of heterogeneous devices that can interact with each other**

The three classes of networking nodes can be distinguished according to cost, size, power and resource features. An overview of the features and classification is given in table 1. Furthermore, all classes have compatible interfaces, both to the user and to the developer.

### 2.1 Wireless Interfaces and Resources

Wireless networking systems available today range from W-LAN to cellular and low rate messaging and control technologies. Ideally such a medium would be required to be able to send a lot of data, very far, very fast, to many separate users and all at once. Unfortunately to achieve all of these objectives simultaneously is prohibited by laws of physics so a compromise has to be made in many cases.

Major differences can be identified in the spatial capacity, network topology, resource consumption, latency and throughput. Significant trends have been identified and are expected to sustain into the future: I. A growing demand for wireless data in portable devices, II. the crowding of spectrum, III. The growth of high speed wired Internet access, and IV. the shrinking semiconductor cost and power consumption for signal processing.

The wireless communication interfaces used in this prototyping platform are standard interfaces. This allows for flexible interaction with other devices that incorporate these standards. The availability of standard interfaces and drivers is necessary for a fast deployment of the whole system to potential users.

W-LAN has shown to perform very poorly when used with single interfaces in multi hop scenarios. Multiple networking interfaces per nodes allow overcoming limitations imposed by standardized wireless systems at the cost of additional channels and of resources. Using multiple instances of both Bluetooth and W-LANs in a node allows to adapt to the necessary cell sizes (overlay networks) and to scale the available bandwidth in multiples of 0.768 (Bluetooth), 11 (IEEE 802.11b) and 54 (IEEE 802.11a), up to 108 Mbit/sec (proprietary IEEE 802.11a). Additionally, multiple Bluetooth systems per networking node make it possible to actively route across the very limited piconet borders, whereas Bluetooth and W-LAN in one node allow to bridge between the different standards.



### 2.2 MANET Testbed Classes

The Maxi class consists of laptops equipped with different wireless interfaces ranging from low rate, low range Bluetooth at 768 kbit/sec to high rate IEEE 802.11a at 108 Mbit/sec. Support for multiple interfaces is only limited by the overall size, mobility requirements and power consumption. Characteristics of this class are flexibility, a feature rich user interface and a wealth of support and debugging possibilities.

The Mini class is iPAQ type devices that are already a common handheld platform used by many researchers. The benefit of this device is the considerable reduction in form factor while still supporting a familiar and powerful operating environment, integrated Bluetooth and W-LAN and the broad availability of peripherals and documentation. The networking interfaces that can be supported simultaneously are somewhat more limited than on a laptop because only one expansion pack can be added at a time and the USB port is solely available to be used as a slave device.

At about 1/10 the cost and the size of a laptop it can be used in higher mobility settings with possibly more units than with PC based systems. The ratio of the operating frequencies, memory and network bandwidth are comparable at around 1/10, only the permanent storage capability is considerably reduced by 1/40. Most deployments today are known to have on the order of tens of these devices.

The characteristics of the Micro class devices are a competition-less small form factor and low component count while maintaining a standardized wireless interface. A detailed description of the system implementation is available in sections 3 and 4.

Fulfill the requirement of ubiquitous deployment of MANET capable devices, many more devices than feasible using the Maxi and Mini class need to be considered. When comparing the Micro class devices to the other two device classes the magnitude of reductions in cost, size, operating frequency and communication bandwidth are comparable, at about 1/50 for the Mini class and about 1/100 for the Maxi class resulting in a similar distribution of resources among these devices. The storage and memory is reduced drastically by factors of many thousands for nonvolatile storage and for system memory. This shows clearly that these devices are not suited for resident applications but more for networking and communication applications.

| Class | Operating System | API |
|---|---|---|
| Maxi | Multitasking OS | Java JDK/JRE |
| Mini | Downsized OS | Simple Java JRE |
| Micro | Limited OS | Simple drivers |

**Table 2. Software APIs**

For each networking interface the appropriate drivers are provided as part of the operating system. At a minimum a connection/disconnection state machine and means for exchanging addressing information are necessary in such a driver. Applications then can make use of the interfaces through these drivers.

### 2.3 Energy Aware Operation

Many developments concerning lowering the power consumption of very specific parts or of whole systems have been made. In principle, there are two ideas most noteworthy to our testbed approach: I. Wireless communication strongly dominates digital computation and II. If you really want to save power you have to turn o systems or parts thereof. In the sense of a whole architecture, to turn o a system really means to power o , not just to trigger a disable pin or go to a different power mode.

The well known numbers on the bitwise energy consumption for wireless communication and computation are important indicators. But they cater primarily to the development of semiconductor circuits. From a system perspective it is not just important to achieve a single most efficient power mode, but an overall optimized duty cycle for the system power consumption.

This means that one has to provide flexible methods for controlling all the resources and their modes of operation available on a networking node. Where shutdown or disable is not implemented in the hardware or in cases where these are not sufficient to reach power dissipation close to 0 when unused, a separate software controllable power supply should be implemented. The result achieved through this method is comparable to clock gating with a remaining power dissipation of only the standby current of the power supply which is typically < 6 µW. Good examples for power control are the PCMCIA specifications that have added power-management in a well-defined interface. Insertion and removal, suspend and resume as well as the time most appropriate to change to a new power state are detected by the drivers. When disabled, the driver interface can disconnect the card from it's power source completely.

Standby modes are not really an option when the goal is to investigate networking applications. Without improved energy saving mechanisms and standard rechargeable batteries the Maxi and Mini classes can be operated autonomously for only short periods. The Micro platform shows a tenfold increase in the operating time (see table 3).



| Class | Supply [Wh] | Power Drain [W] | Lifetime [h] |
|---|---|---|---|
| Maxi | 53.28 | 13.3-26.2 | 2-2 |
| Mini | 5.04 | 0.39-2.02 | 2.5-13 |
| Micro | 3.02 | 0.067-0.25 | 12-45 |

**Table 3. Power consumption under load**

## 3. Bluetooth Smart Nodes - The Embedded Micro Platform in Detail

The BTnode is an autonomous wireless communication and computing platform based on a Bluetooth radio module and a microcontroller. The benefit of this platform is the small form factor while still maintaining a standard wireless interface. With its general purpose interfaces the BTnode can be used with many peripherals, such as sensors, DSPs, serial devices (like GPS receivers, RFID readers, etc.) and user interface components.

Having networks comprising hundreds of autonomous nodes in mind, the following target features were considered in the design of the BTnodes:

- In-circuit programmable Bluetooth platform: Simple reprogramming of the system software as well as drivers is crucial to any development work. Although a very standard requirement a surprising amount of devices do not yet support this feature.
- Remote update of system software: When it comes to the deployment of the BTnodes there must be a way to reprogram the nodes over the network as they will be distributed in many different locations and not necessarily attached to a PC.
- Low component count: Cost effective, timely deployment and the reduction of design iterations both demand a fairly simple design consisting of standard components.
- Overall system size: One of the most apparent arguments for not deploying laptops and PDAs in large quantities besides the cost is the system size. They would interfere unduly with the environments that are intended for the use of our testbed components and sometimes cannot be placed exactly where necessary.
- Simple debugging capability: An experimental platform must support an easy to handle standard way of debugging. Many changes to the software and extensive monitoring are the main applications for this feature.
- Sensor and user interface: The BTnodes are targeted to be attached to sensor and UI peripherals in many usage cases. A standard interface is thus necessary.
- Single voltage design with power management: Supporting multiple operating voltages creates a much more complex design and limits the possible interactions with peripherals.

Most commercial Bluetooth solutions are available as fully self-contained transceiver modules that implement the lower layers of the protocol from RF front end to the generic Host Controller Interface (HCI). Representing a classical embedded system they contain an embedded CPU, different types of memory, baseband and radio circuits. They are designed to be used as add-on peripherals with the higher layers of the Bluetooth protocol and applications implemented on a host system. Due to system partitioning and commercial interests the in-system CPU and memory are usually not available to developers for user specific implementations.

## 4. System Software for the BTnodes

The BTnode system software is a lightweight operating system made up of low level drivers that are interrupt driven and a simple dispatcher for scheduling multiple threads. This OS is well suited for the applications of such small scale networking devices that will consist of mostly simple IO and monitoring tasks and communication.

There are two programming models: A sequential model and an event-driven model with cooperative multitasking. Different system software exists for the two models. The sequential model allows tight control of all resources but leaves more tasks to the application programmer, while the event-driven model provides convenient functions for resource management but is more restrictive when accessing hardware resources. This is explained in the following.

### 4.1 The Dispatcher

The dispatcher implements cooperative multitasking. Only one task (event handler) can be active at a time. Events are processed in the order they appear. Every event handler is always completely executed until the next is scheduled. So every event handler depends on the previous event handler to terminate in time.

A software component, such as a driver, can generate an event to notify other components of the occurrence of a change in state that requires further action. For example a byte is received on the serial port. The serial driver now needs to notify an interested software component (e.g. an application or another driver) about this event so that it can read the byte received. This is done by inserting the corresponding event into the dispatcher's event queue. The dispatcher has a FIFO queue to hold events that have been triggered but not yet handled by the dispatcher:

Execution of event handlers can be delayed by other tasks. In that queue, however, there might be several other events already waiting for their event handlers being executed. So the delay by other tasks depends on the system load (tasks waiting for execution) and the time these tasks may take.

### 4.2 Reduced Bluetooth Protocol Stack

The Bluetooth radio module offer a generic host controller interface (HCI) to the lower layers of the Bluetooth protocol through a serial port while the higher layers of the protocol such as link and networking layer must be implemented on the host system. HCI commands and events are asynchronous which means that a state machine must be available to handle all these correctly.

The relevant commands and data structures from the HCI and L2CAP layers (see figure 5) have been implemented to support a basic connection manager at a very low level. The higher Bluetooth protocol layers required to provide connectivity to consumer devices are now part of the applications and are to be integrated in future system software.

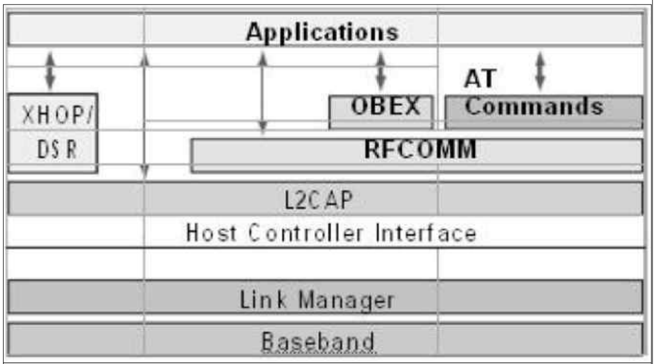


**Figure 5. Reduced Bluetooth protocol stack**

## 5. Ad-hoc Dynamic Source Routing on Bluetooth

A reduced version of the CMU Dynamic Source Routing protocol has been implemented as a packet forwarding mechanism on the wireless testbed. This XHOP/R-DSR protocol enables communication across piconet borders and hooks right into the L2CAP layer of the Bluetooth stack (see figure 5). It has been assumed that routes can be discovered and maintained separately from the packed forwarding mechanism. Due to the limitation of Bluetooth to master-slave piconets with at most 7 active slaves the capability of the multi-hop routing is reduced to a multi-hop message passing mechanism. Furthermore the long setup delays of Bluetooth are a drawback when standard BTnodes with one communication front end are used. On every hop a connection is opened by the source (master) and an XHOP packet is transmitted to the next destination (slave). This connection is then closed and the payload is evaluated. In order to forward a packet over further hops a new connection is opened and the XHOP packet is sent on until the final destination has been reached.

A simple script command language has been implemented in the payload of the XHOP packets to allow the execution of remote commands. For example a

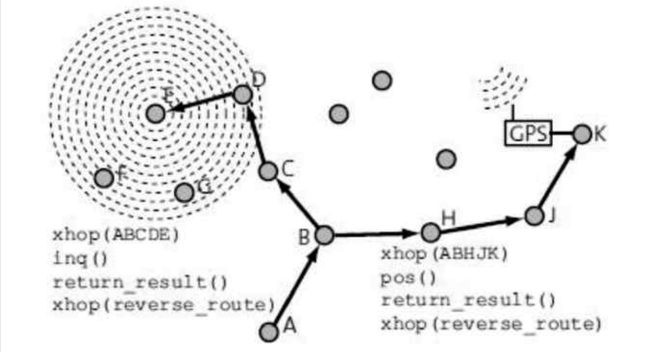


**Figure 7. Remote topology discovery in a heterogeneous environment using XHOP/R-DSR**


**Figure 8. Product monitoring using BTnodes as smart tags and SMS via mobile phones**

Similarly, the Bluetooth OBEX profile can be used to exchange and store data from BTnodes with mobile phones. Figure 8 shows an example of a product monitoring application where a shock sensor has been attached to the BTnode to detect motion patterns.

## 6 Conclusions and Future Work

In this paper we have reasoned that it is necessary to use standard wireless interfaces to be able to interact with wireless enabled consumer appliances. Moreover the design and implementation process of applications is simplified through the use of well defined standard APIs.

A prototyping testbed that is suitable for the deployment of large populations of heterogeneous distributed mobile communication and information systems to support a "smart" networked environment in everyday objects has been motivated and described. The centerpiece is the Bluetooth Smart Node, an embedded networking device that can store data, compute and communicate over a standard wireless interface. We believe that realizing strategies for the reduction of the overall system power consumption implies a good understanding and flexible control of all subsystems and respective operating modes involved. System centric power aware operation has been realized in the design of the hardware and the lean operating system.

The drawbacks of Bluetooth, namely the high power consumption and the rigid networking structure can be overcome by our approach of multiple front ends per node. They are outweighed by the fact that interaction can take place with almost any other wireless enabled device. The two examples and the successful deployment with different research groups show that our approach is feasible and extendable to larger systems and applications.

Future applications of the BTnode will be in the networking and ubiquitous computing application domain. Enabling technology like network topology management and control as well as positioning services will evolve to support applications



***References:***

1. B. Raman, S. Agarwal, Y. Chen, M. Caesar, W. Cui, P. Johansson, K. Lai, T. Lavian, S. Machiraju, Z.M. Mao, G. Porter, T. Roscoe, M. Seshadri, J. Shih, K. Sklower, L. Subramanian, T. Suzuki, S. Zhuang, A.D. Joseph, R.H. Katz, and I. Stoica, "The SAHARA Model for Service Composition Across Multiple Providers," Pervasive Computing, August 2002.
2. J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Next Century Challenges: Mobile Networking for Smart Dust," in Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking, 1999, pp. 271–278.
3. L. Doherty, B.A. Warneke, B.E. Boser, and K.S.J. Pister, "Energy and performance considerations for Smart Dust," International Journal of Parallel and Distributed Systems and Networks, vol. 4, no. 3, pp. 121–133, 2001.
4. J. Hill, R. Szewczyk, A. Woo, S Hollar, D.E. Culler, and K.S.J. Pister, "System architecture directions for networked sensors," in Architectural Support for Programming Languages and Operating Systems, 2000, pp. 93–104.
5. E. Shih, P. Bahl, and M. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in Proceedings of ACM MobiCom 2002, Atlanta, September 2002.
6. J. Elson, L. Girod, and D. Estrin, "Short Paper: A Wireless Time-Synchronized COTS Sensor Platform, Part I: System Architecture," in Proceedings of the IEEE CAS Workshop on Wireless Communications and Networking, September 2002.
7. M. Stemm, P. Gautier, D. Harada, and R. H. Katz, "Reducing power consumption of network interfaces in hand-held devices," in International Workshop on Mobile Multimedia Communications (MoMUC-3), Prince-ton, October 1996.
8. A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communications technology," Computer Communication Review, pp. 20–41, 2001.
9. L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001, vol. 3, pp. 1312–1320.

10. F. Bennett, D. Clarke, J.B. Evans, A. Hopper, A. Jones, and D. Leask, "Piconet - embedded mobile networking," IEEE Personal Communications, vol. 4, no. 5, pp. 8–15, October 1997.

11. J. Tourrilhes, Magalhaes L., and C. Carter, "On-Demand TCP: Transparent peer to peer TCP/IP over IrDA," in Proceedings of ICC 2002, 2002.

12. M.K. Marina and S.R. Das, "Routing performance in the presence of unidirectional links in multihop wireless networks," in Proceedings of the ACM MobiHoc 2002, 2002.

13. F. Siegemund, "Spontaneous interaction in ubiquitous computing settings using mobile phones and short text messages," in Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings, Ubicomp 2002, September 2002.