

Short Range Wireless Connectivity for Next Generation Architectures

Mandeep Singh, Santhosh Thampuran, Prashant Jain, Russell Tessier, Csaba Andras Moritz

Department of Electrical and Computer Engineering

University of Massachusetts, Amherst, MA 01003

E-mail: {msingh,sthampur,pjain,tessier,andras}@ecs.umass.edu

Abstract

This paper illustrates preliminary findings in our ongoing efforts on integrating Raw, a proposed scalable tiled processor architecture, with the robust wireless connectivity provided by BlueTooth (BT), one of the most popular short range wireless connectivity standards.

Our work of integrating Raw and BT is primarily motivated by two ideas. First, we are interested to evaluate the possible performance benefits – as providing a wireless interconnection for global on-chip messaging could improve network utilization on the wired networks, and improve the scalability of applications with global communications.

In an analytical study we have performed, we found that in a BT-enabled Raw, up to a factor of 5 performance improvement can be achieved for applications that require frequent global messages.

Ultimately, we are also motivated by the wide range of applications that a BT-enabled tiled processor could enable, and the opportunity to easily cooperate with other devices that are BT compatible in a future ubiquitous world.

1. Introduction

Performance and scalability concerns associated with current day architectures have led to the advent of parallel tiled architectures. New wireless standards are being proposed to satiate the demand for high bandwidth and noise immune communication between fixed and portable devices. It is not long before parallel tiled architectures and these standards cross paths. Raw [5, 9] is one of the most promising tiled architectures and BT is a popular evolving wireless standard. This paper illustrates the issues involved in integrating such a parallel and scalable architecture with a robust standard like BT.

Rapid advances in technology force a quest for computer architectures that exploit the new opportunities. Current architectures, such as hardware scheduled superscalars, are already hitting performance and complexity limits and cannot be scaled indefinitely. The Raw Architecture Workstation (Raw) is a simple, wire-efficient architecture that scales with increasing VLSI gate densities. The Raw architecture's goal is to provide performance that is at least comparable

to that provided by scaling an existing architecture, but that can achieve orders of magnitude higher performance for applications in which the compiler can discover and statically schedule fine-grain parallelism.

BlueTooth (BT) [1] is a technology intended to replace cable connecting portable or fixed electronic devices. Designed to operate in a noisy frequency environments, the BT radio uses fast acknowledgement and frequency hopping scheme to make it robust. BT radio modules operate in the unlicensed ISM band at 2.4GHz and avoid interference from other signals by hopping to a new frequency after transmitting or receiving a packet. Compared to other systems in the same frequency band the BT radio hops faster and uses shorter packets. It is foreseen that BT would soon be omniscient in mobile as well as desktop applications.

Raw answers the call of such applications requiring high performance levels that can't be reached by current day superscalar architectures. BT is foreseen to provide wireless connectivity between such Raw systems and other BT-enabled devices. This emphasizes the need to study the issues in integrating the high performance and scalability of Raw and the robust wireless connectivity of BT. The parallel and scalable nature of Raw offers numerous options to provide BT connectivity. This paper explores various possible implementations of a *BT-enabled Raw*. It also illustrates an application of such a BT-enabled parallel architecture.

It has been foreseen [9] that applications would require a sizeable amount of hardware to give reasonable performance on future architectures. This is feasible due to high integration densities achieved due to technology scaling. This raises network contention problems due to increased global communication inherent in some of the applications. A BT-enabled Raw eases network congestion thus improving the performance by a sizeable amount.

The paper is organized as follows. Section 2 introduces Raw and BT. Section 3 talks about various possible implementations integrating BT and Raw. Section 4 talks about preliminary experimental evaluation of the on-chip implementation of a BT-enabled Raw. Section 5 presents an application of a *BT-enabled Raw* system. Section 6 outlines the future work and concludes the paper.

2. Background

2.1. BlueTooth- A Cable Replacement Technology

Bluetooth is a wireless technology using short-range radio links, intended to replace cables for portable and fixed electronic devices [1]. The BT technology uses a fast acknowledgement and frequency hopping scheme enabling it to perform in noisy environments. BT radio modules operate in the unlicensed ISM band at 2.4GHz and avoid interference from other signals by hopping to a new frequency after transmitting or receiving a packet. Compared to other systems, BT radio hops faster and uses shorter packets.

The protocol stack of the BT technology is shown in Figure 1. The BT radio is the lowest defined layer of the BT

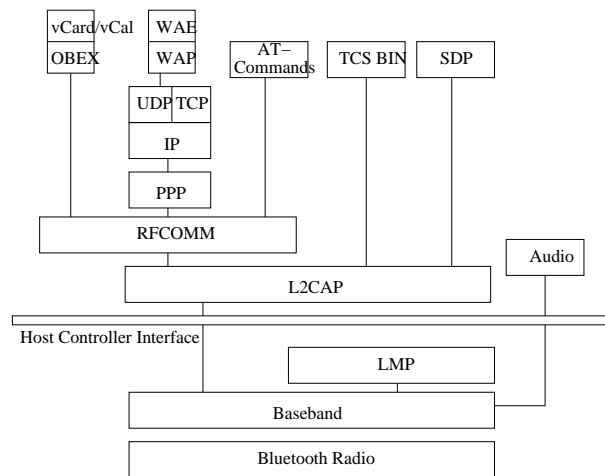


Figure 1: BlueTooth Protocol Stack.

specification and consists of the BT transceiver device operating in the 2.4GHz ISM band.

The Baseband is the physical layer of the Bluetooth and lies on top of the Radio layer. The Baseband layer manages physical channels and works with the Link manager for carrying out link connection and power control as well as services such as error correction, data whitening, hop selection and BT security. It also manages asynchronous and synchronous links, performing paging and inquiry to access and inquire other BT devices in the area. The Link manager carries out link setup, authentication, link configuration and other protocols. The Host Controller interface(HCI) provides a command interface to the baseband controller and link manager, providing a uniform method of accessing BT baseband capabilities.

Logical Link Control and Adaptation Protocol (L2CAP) provides connection-oriented and connectionless data ser-

vices to upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions. The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The Service Discovery Protocol (SDP) provides a means for applications to discover which services are available and to determine the characteristics of those available services.

2.2. Raw- A Parallel and Scalable Architecture

A Raw processor [4] is a chip containing a 2-D mesh of identical tiles (Figure 2), where the tiles connect to the nearest neighbors by the dynamic and static networks. The Raw architecture provides a raw, highly scalable, parallel interface to the application using every millimeter of the silicon. The Raw architecture is equipped with a parallel compiler and aims to maximally utilize it by fully exposing the hardware and by delegating the hardware's control completely to the software system.

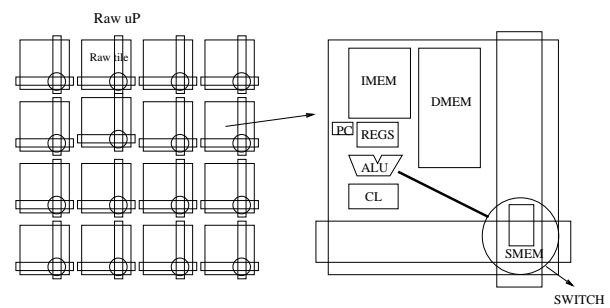


Figure 2: Raw Architecture.

Every tile in the 2-D mesh of tiles consists of a tile processor, a static switch processor and a dynamic router. The tile processor uses a 32-bit MIPS instructions set, while the switch processor uses a MIPS-like instruction set. The dynamic router runs independently, and is user control only indirectly. The tile processor also consists of an SRAM (data and instruction memory). The tile processor communicates with the switch processor and the dynamic router.

The parallel compiler [5] for the Raw architecture includes issues such as resource allocation, the exploitation of fine-grained parallelism, communication scheduling, the use of configurable logic and code generation. The parallel compiler maps different applications to different regions in the architecture, with each region consisting of a number of closely placed tiles. The number of tiles depend on the performance-criticality of the application. Program execution can be divided into a small number of large regions (fine-grained parallelism) or a large number of smaller regions (coarse-grained parallelism), thus exploiting parallelism in the applications mapped. Finally, the compiler schedules both instructions and communication events, with

the events being scheduled both temporally and spatially.

3. RawBite - A BT-enabled Raw

The highly parallel and software exposed scalable architecture of Raw offers numerous options to implement BT. The candidate implementations can be classified based on the locality of the implementation and based on how much of the BT protocol stack is implemented in software. Three primary implementations have been proposed

1. *The on-chip implementation*, involves the addition of a specialized BT tile in the Raw architecture.
2. *The off-chip implementation*, has BT as an off-chip component.
3. *The software-centric implementation*, aims to implement most of the BT protocol stack in software.

The following sections talk in more detail about each of these implementations,

3.1. On-Chip Implementation

The motivation for implementing BT on chip stems from the time bound nature of communication links which emphasizes the need to have a high speed implementation of BT. This motivation is further augmented due to BTs omniscient nature to systems in the future. This approach involves the

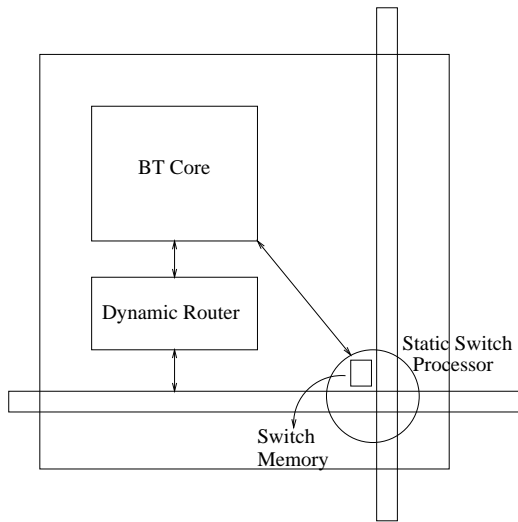


Figure 3: BT Tile in Raw.

design of a special purpose tile which implements the BT functionality (Figure 3). Therefore in addition to the replicatable tiles shown earlier, the Raw architecture will also include a BT tile. The latency encountered in communication can be further reduced if the software stack was executed in the tiles closer to the BT tile.

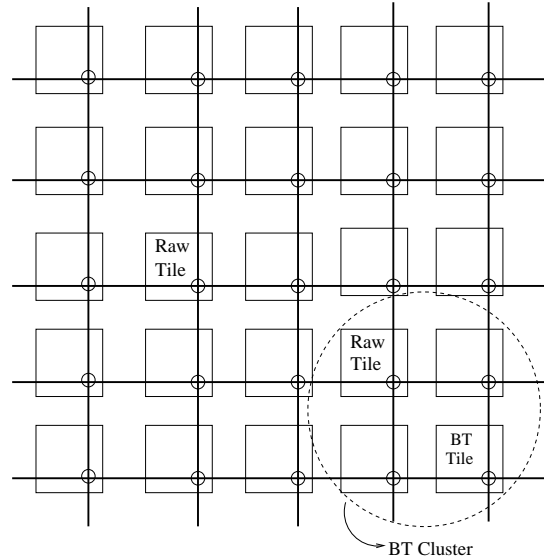


Figure 4: BT Cluster in Raw.

This can be achieved by allocating a group of tiles close to the BT tile for BT specific program execution. Such a group of processing units and the BT tile would form a BT cluster (Figure 4) which would handle the wireless traffic. As the amount of the BT protocol stack that has been implemented in hardware increases the number of tiles in the cluster reduces till we reach the extreme case wherein the whole of BT is implemented in hardware. With recent advances in fabrication technologies it is possible to realize all of the RF circuitry on chip thus reducing communication latencies even further.

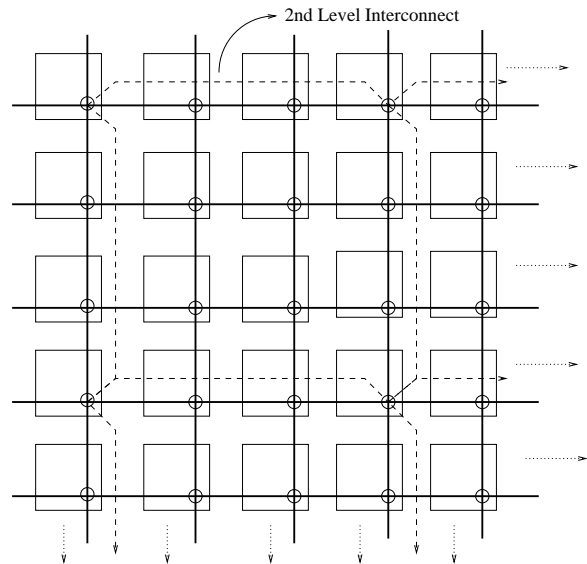


Figure 5: Second Level of Interconnect.

The 0 approach results in reduced communication latencies at a cost of uneven usage of resources. This is because the tiles in the BT *cluster* would remain unused when the wireless link is idle. A more efficient approach would be to assign the BT specific program execution to the nearest free tile. This would ensure an even resource allocation and at the same time giving the optimum latency under the given constraints. Since it is impossible to predict BT communication at compile time the BT traffic is handled using the dynamic network in Raw. The components of a BT tile in the Raw architecture is shown in Figure 3.

In a scalable architecture like Raw where integration of thousands of replicated tiles is envisioned, a need for a second level of interconnect will soon be felt. The second level of interconnect in addition to reducing communication latencies between tiles spaced far apart, will also ease congestion on the first level network.(Figure 5) There are three possible options for implementing a second level interconnect

1. *Metal Interconnect*

Raws replicated tiles are connected to its nearest neighbors by short wires hence providing high performance due to short interconnect lengths. In such a billion transistor design with thousands of tiles the interconnect routing would be complex. Having another level of interconnect would further complicate the routing problem. In addition the second level of interconnect would incur huge delays due to long wires, hence reducing the performance.

2. *Optical Interconnect*

High delays encountered in long metal interconnects have motivated the development of alternate methods for signal routing. Notable among the alternatives is optical interconnections. This type of an interconnection involves the routing of optical signals converted from electrical through waveguides. This kind of interconnect suffers from some drawbacks. Major issues involve problems due to the optical signal getting absorbed at sharp bends in the waveguides. Curvaceous waveguides have been proposed for a clock distribution application in [6]. But it is difficult to fabricate such precision waveguides.

3. *BT enabled wireless interconnect*

Improved RF capability and projected increase in die sizes have led to the concept of wireless interconnects. The effectiveness and practicality of wireless interconnects has been demonstrated in [7]. In addition to reducing the delays and power consumption due to long interconnects, wireless interconnects also allow the formation of adhoc networks at no extra cost. In an architecture like Raw which exploits the ILP in the

software, this kind of a flexibility allows a more efficient static scheduling by the compiler.

Traditional on-chip wireless interconnects have been used for clock distribution [7]. The main concern areas of such an application has been noise and synchronization. The robust nature of BT alleviates both of these problems. The fast frequency hopping scheme ensures that signal integrity issues are minimal.

The 0 factors bring out the BT-enabled wireless interconnect as the most attractive option. Using BT within the Raw chip at a coarser level will solve the long interconnect problem. In such a scheme the Raw architecture would be divided into a number of clusters (consisting of Raw tiles) comprising of geographically distant tiles. The possible communication paths between two tiles would be either through short, multiple metal interconnects or through the wireless interconnect using BT. A cost function would be used to evaluate which of the two possible interconnects are faster. A BT tile in addition to performing the BT related functions can also be used for routing signals akin to other Raw tiles.

3.2. Software-Centric implementation

The demands of applications requiring high performance computations have compelled designers to always go for hardware-centric implementations of designs. But the advent of architectures like Raw which exploit Instruction Level Parallelism (ILP) present in the software to the maximal extent, has provided the designer with a more attractive option of a software-centric design. In addition to giving comparable levels of performance with hardware implementation it also enables us to implement complex algorithms as simple chunks of code. Earlier this would have resulted in a complex hardware solution requiring a very high design time. In applications like BT which are continuously evolving due to innovations in data processing algorithms and wireless networking, the flexibility provided by such an implementation is desirable.

Current BT implementations have the Host Controller Interface (HCI) and higher layers in software while the baseband is implemented as a dedicated off-chip add-on. This is due to the time critical functions handled by the baseband which mandate a high speed solution. These functions include access code correlation used for identification of the destination of a packet, decryption and header and data error checks among others. The comparable performance offered by Raw enables us to think about implementing these in software. Such an implementation would comprise of a large chunk of the protocol being implemented in software and just the RF circuitry implemented in hardware. Furthermore the advances in VLSI provides the option of im-

plementing the RF block as a on-chip component.

Noise considerations in wireless standards prompt the development of more complex data encoding and processing algorithms. A software implementation enables facilitates the use of such algorithms. Most of the traditional algorithms are sequential in nature. Hence there is a scope for further performance gains by modifying sequential algorithms for parallel distributed memory architectures like Raw. These modifications ensure that the resulting software exhibits a high degree of parallelism. With parallel tiled architectures being proposed as the architectures of the future, there is a need to study how applications can be modified so as to enable a more optimal execution on these architectures.

Since parallel tiled architectures are the favored architectures of the future it is essential that the applications running on them be designed keeping the underlying architecture in mind. As mentioned earlier sequential algorithms modified for highly parallel architectures have shown better performance [3]. Communication standards like BT can be further restructured for implementation on a parallel architecture like Raw.

3.3. Off-Chip Implementation

Another proposed implementation keeps the BT hardware external to the Raw chip. The software portion of the protocol stack is still executed using Raw. This implementation ensures minimal hardware changes to the Raw architecture at the same time utilizing the computational power offered by Raw. Existing implementations of BT implement the baseband part of the protocol stack on the off-chip hardware. This is due to the time critical nature of the functions handled by the baseband. The hardware can be implemented on a custom IC, an ASIC or a FPGA. Since the latencies in this kind of an implementation would be more, a faster off-chip BT implementation should be chosen. Thus a custom integrated circuit would give the best results. The RF circuitry can be added as a separate one chip solution to the baseband chip [8]. But advances in fabrication technologies allow integration of the RF circuitry and the baseband on a single chip.

4. Preliminary Experimental Evaluation

This section presents our preliminary findings regarding possible performance improvements due to improved network utilization on a BT-enabled Raw system. In our evaluation we used two analytical frameworks, the SimpleFit model [9] and the LoGPC model [10].

Our earlier studies presented SimpleFit [9], a novel analytical framework that designers can use to reason about the design space of RAW microprocessors in a billion transistor era. This model is also generalizable to other sin-

gle chip systems. Although the optimal machine configurations obtained vary for different applications, problem sizes and budgets, the general trends for various applications are similar. The applications studied are: Jacobi Relaxation, Dense Matrix Multiply, Nbody, FFT and Largest Common Subsequence. For the applications studied, assuming a 1 billion logic transistor equivalent area, we found that building a Raw chip with approximately 1000 tiles, 30 words/cycle global I/O, 20Kbytes of local memory per tile, 3-4 words/cycle local communication bandwidth, and single issue processors would give near optimal performance.

This configuration will give performance near the global optimum for most applications. Figures 6,7, 8 shows the optimal division of chip resources, namely number of processors, local and global communication bandwidth for the various applications as a function of the problem size (N).

Even though some of these benchmarks show good spatial locality, the same can't be assumed for most of the real application. In most of these applications, especially those with frequent and irregular communication patterns or those using large messages, network contention and contention for message processing resources can be a significant part of the total execution time. Our earlier studies on modeling network contention in message-passing programs presented a new cost model called LoGPC [10]. Based on that model we expect network contention to be more significant because of increased average distance traversed by a message.

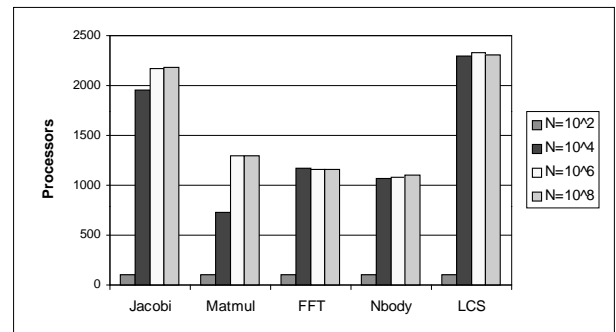


Figure 6: Number of Processors in optimal machine configurations for different problem sizes [9].

Using the LoGPC cost model, we performed preliminary experimental studies on the factor of performance degradation using wired interconnects in one case and using Bluetooth for short range wireless connectivity in the other. The parameters used for the computation involved message length of 100 bytes, network bandwidth of 4 bytes per cycle. The average distance a message travels in each network dimension (K_d) for the wired interconnect case is assumed to be 11 hops and the corresponding value for the model involving Bluetooth is assumed to be 5 hops. This has been

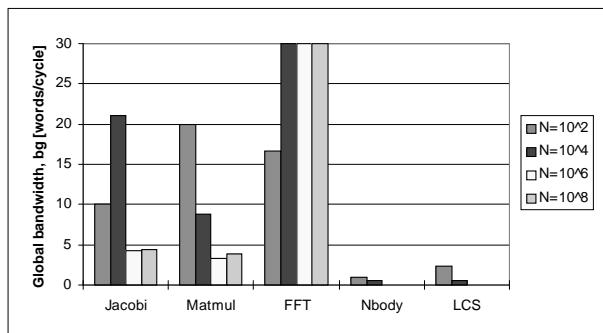


Figure 7: Global IO bandwidth in optimal machine configurations for different problem sizes [9].

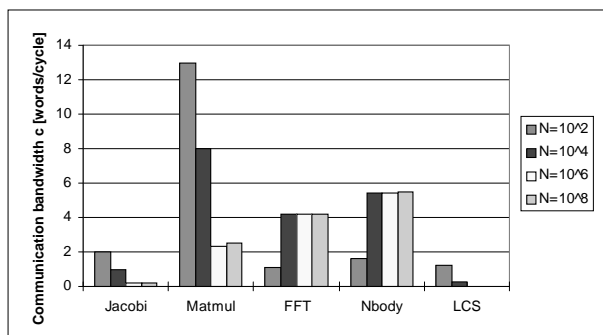


Figure 8: Local communication bandwidth in optimal machine configurations for different problem sizes [9].

defined assuming a bidirectional no wrap-around mesh network where the average distance a message travels (K_d) is given by,

$$K_d = \frac{k^2 - 1}{3k}, \quad (1)$$

on any dimension k . From these values, it was estimated using the expression from [10] that in the case of wired interconnects, the performance degradation due to network contention has an effect of a slowdown by a factor of 16. However, using Bluetooth this figure comes down to 10. As shown in Table 1, Raw with 9 BT nodes gives as much as a factor of 5 improvement in worst case execution time (E_t).

Parameters	No Contention	No BT	4 BT Nodes	9 BT Nodes
K_d	NA	10	5	2.2
E_t (incycles)	100	1600	1000	340

Table 1: Comparison of Worst case evaluation times of Raw with and without BT.

5. Application: Case of A Distributed Mobile Computing Environment

The increasing processing power of mobile devices (e.g. handheld computers, PDA, WAP phones) and their integration into network infrastructures (mobile internet and intranet access) lead to a wide range of new applications and services, extending distributed systems via wireless communication media. Traditional mobile devices handle specific applications. These mobile devices were interconnected through access points which had limited processing power and intelligence. (Figure 9(a)) Various mobile devices providing different kind of services have been developed. This has led to devices like the personal digital assistant, pagers and mobile phones. With the increase in the number of mobile device users, there is a demand for providing varied applications like web access, email and telecommunication on a single device. The traditional model uses the processing power provided by the mobile device for all the application related processing needs. Hence the user has to carry different devices for different applications. Though this model worked well with devices running a limited number of applications, it does not scale well for more complex applications. Such applications may require immense processing power which can't be satisfied due to the limited battery power available on the mobile device.

Hence an alternate distributed computing model is proposed. The main idea is to move processing and intelligence out of the portable device into fixed access points, used for connecting the device to the wired network infrastructure. (Figure 9(b)) Mobile devices poll for services provided by access points using BT. Services provided by service providers reside on servers which have a multitude of such requests coming in and thus require a high performance processor like Raw for servicing those requests. This in addition to satisfying the user demands of access to applications also provides a longer battery life.

6. Conclusion and Future Work

Various ideas for implementing a wireless standard like BT on a parallel tiled architecture like Raw were explored. The potential of Raw to offer levels of performance comparable to a hardware implementation enables a software centric implementation which in a continuously evolving standard like BT is favorable. Since parallel tiled architectures are favored as high performance architectures of the future it is necessary to think about restructuring standards for implementation on a parallel architecture. It was also seen that the BT-enabled Raw eases out congestion on the network thus improving performance by as much as 5 times.

Future work involves coming up with a restructured, parallel architecture friendly BT stack. Furthermore the BT stack as it is and after restructuring will be implemented in C and its performance on a Raw machine will be mea-

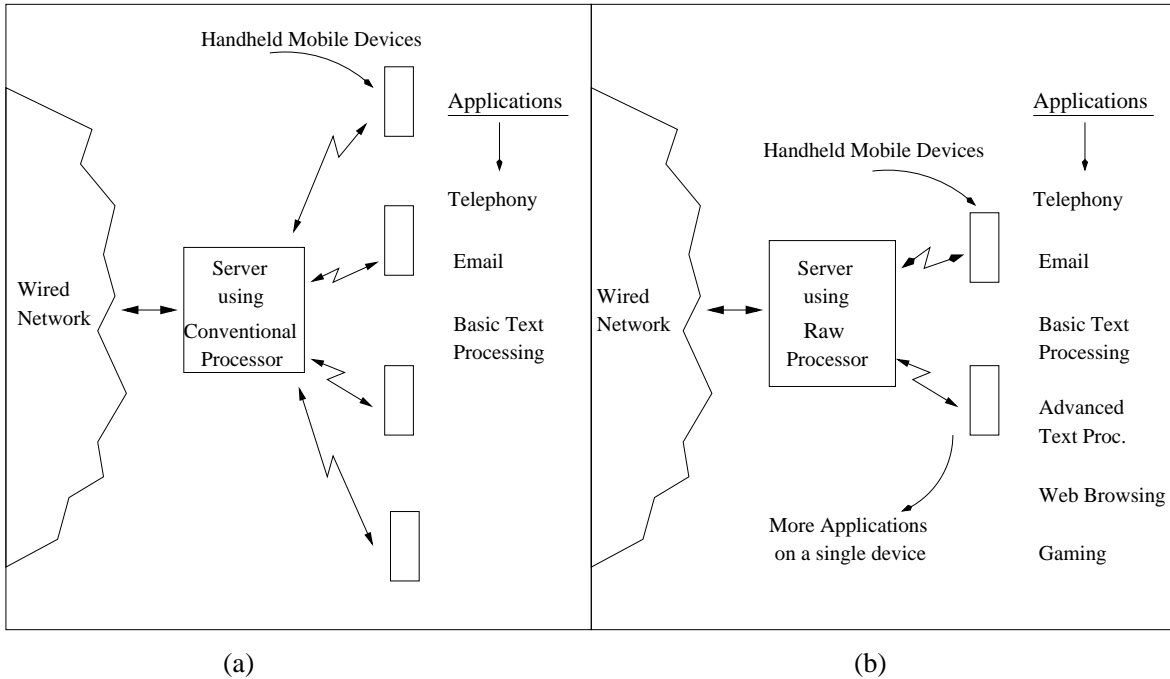


Figure 9: (a) Traditional mobile environment (b) Proposed mobile environment.

sured. Ultimately a working prototype of RawBite with a software-centric implementation will be demonstrated in the illustrated application.

7. References

- [1] "Bluetooth Specification," Ver. 1.0 B, www.bluetooth.com.
- [2] J. Flinn and M. Satyanarayanan, "Energy aware adaptation for mobile applications," *Proc. of the 17th ACM symposium on Operating Systems*, pp. 48-63, Dec. 1999.
- [3] H. Li and K. C. Seveik, "Parallel sorting by overpartitioning," *Proc. of the Symposium on Parallel Algorithms and Architectures*, pp. 46-56, 1994.
- [4] Elliot Waingold, Michael Taylor, Vivek Sarkar, Walter Lee, Victor Lee, Jang Kim, Matthew Frank, Peter Finch, Srikrishna Devabhaktuni, Rajeev Barua, Jonathan Babb, Saman Amarasinghe, and Anant Agarwal, "Baring it all to Software: The Raw Machine," *MIT/LCS Technical Report TR-709*, March 1997.
- [5] Walter Lee, Rajeev Barua, Matthew Frank, Devabhaktuni Srikrishna, Jonathan Babb, Vivek Sarkar, and Saman Amarasinghe, "Space-Time Scheduling of Instruction-Level Parallelism on a Raw Machine," *Proceedings of the Eighth International Conference on Architectural Support for Programming Languages and Operating Systems*, Oct. 4-7. 1998.
- [6] Anthony Mule, Stephen Schultz, Thomas K. Gaylord, and James D. Meindl, "An Optical Clock Distribution Network for Gigascale Integration," *IEEE International Interconnect Technology Conference*, pp. 6-8, June 2000.
- [7] Brian Floyd, Kihong Kim, and Kenneth O, "Wireless Interconnection in a CMOS IC with Integrated Antennas," *International Solid State Circuits Conference*, 2000.
- [8] "LMX5001 Dedicated Bluetooth Link Controller Data Sheet," National Semiconductor Corporation, www.national.com
- [9] Csaba Andras Moritz, Donald Young and Anant Agarwal, "SimpleFit: A Framework for Analyzing Design Tradeoffs in Raw Architectures," *accepted in IEEE Transactions on Parallel and Distributed Systems*,
- [10] Csaba Andras Moritz and Matthew I. Frank, "LoGPC: Modeling Network Contention in Message-Passing Programs," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 4, April 2001.