Low-Complexity, High-Throughput Multiple-Access Wireless Protocol for Body Sensor Networks

Seung-mok Yoo[1], Chong-Jing Chen[2] and Pai H. Chou[2,3]
[1]Electronics and Telecommunication Research Institute, Daejeon, 305-700, Korea
[2]Center for Embedded Computer Systems, University of California, Irvine, CA 92697, USA
[3]Department of Computer Science, National Tsing Hua University , Hsinchu, Taiwan
Email : yoos@etri.re.kr, chongji@uci.edu and phchou@uci.edu

Abstract

Wireless systems that form a body-area network must be made small and low power without sacrificing performance. To achieve high-throughput communication in low-cost wireless body area networks, we propose a low-complexity, "pulling" MAC protocol. Such a network architecture consists of low-complexity nodes and moderate-complexity base stations, which act as clients to pull data from the nodes that act as thin servers. This organization achieves intra-network collision-free multiple access as in TDMA but without expensive time synchronization. It also achieves high utilization of air bandwidth and adaptivity of CSMA protocols but without collision or complexity on the node side. Experimental results show that our pulling protocol achieves better good channel utilization of up to 52.8% and high data throughput of up to 432Kbps. We also tested our MAC protocol on a wireless ECG system. Our MAC protocol can transmit ten simultaneous streams of data at 200 samples per second each from ten sensing devices.

1 Introduction

Body area networks have emerged as a distinct class of systems. Representative applications include personal health monitoring and human motion tracking [1,3,4,9,10]. Fig. 1 shows an example of a typical body area network. Unlike wireless sensor networks (WSNs) in general, body area networks have more specific requirements. First, the system should be small and lightweight to be wearable [9]. Second, these applications tend to demand much higher data throughput than traditional wireless sensor applications [5], and they tend to use single-hop star topology over a short distance rather than arbitrary mesh networks with relaying and routing. For example, electrocardiograms (EKG or ECG) for health monitoring can be sampled at 1000Hz per channel [3, 9]. Triaxial accelerometers for human motion tracking can be sampled at 1000 times per second per axis [4]. These requirements translate into a raw data bandwidth of 40K to 400K bits per second, even after compression.

We propose an architecture for high performance wireless body area networks based on the concept of pulling protocols. Unlike traditional WSNs that assume data pushing, our sensor nodes act as *thin servers*, and our base stations pull data from these thin servers on behalf of the host computer, which acts as a *fat-client* [16] . A base station may pull autonomously or may be transparent to the host by passing commands and data through. Depending on commands, sensor nodes can send multiple replies for a single command.

The contributions of this paper include both the protocol and the architecture that it enables. The protocol has the property of intra-network collision-freedom without requiring complex time synchronization or maintaining global time. The base station can also adapt its pulling schedule and even frequency hopping based on its global knowledge, without additional coordination from the distributed sensor nodes. Most importantly, all of these features can be implemented without modification to the firmware on the sensor node, all by changing the behavior on the base station side. This particular partitioning scheme enables us to keep the complexity of these body area networks very low while at the same time achieving higher performance than existing schemes. By our experiment, the channel utilization reaches 52.8% and the data throughput for the packet payload reaches 432 Kbps.

The paper is organized as follows. Section 3 describes the fat-client, thin-server architecture we propose. Section 3.1.2 describes how the base station reads sensor data, and Section 4 presents experimental results.

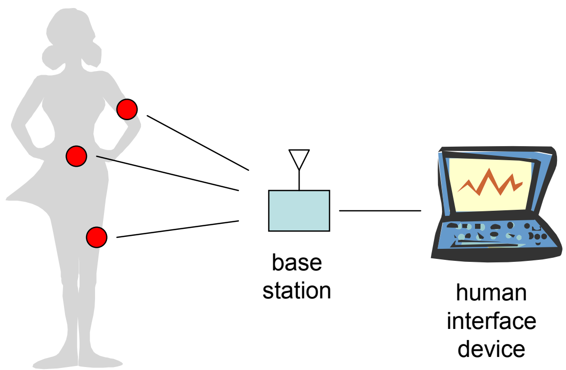

Figure 1. The topology of wireless body sensor networks

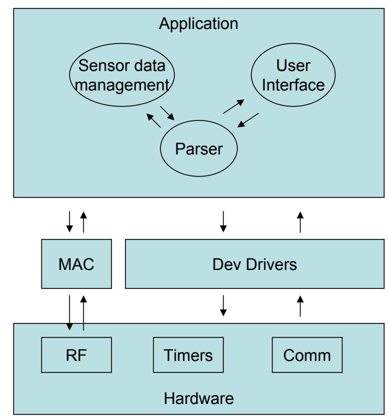

Figure 2. Base station architecture

2 Related work

Many MAC protocols for wireless sensor networks have been proposed. Contention-based protocols are compact and adaptive to environmental changes. However, it has been known that contention-based protocols are not suitable for dense networks because of their low channel utilization and data throughput. [14]. It is also difficult to predict the performance. TDMA-based protocols show good channel utilization and consistent data throughput in a dense network. However, they need slot scheduling, and they are not adaptive to environmental changes. Thus, pure contention-based and TDMA-based MAC protocols may not be a good solution for wireless body area networks.

To take advantages of both, many hybrid protocols have been proposed [2, 13, 17, 18]. At the same time, they also inherit disadvantages from both styles. Time synchronization overhead is one of them. Most protocols can work with local time synchronization, which incurs less overhead but does completely eliminate it. Moreover, fusing two different approaches may increase the code size and the execution overhead of the sensor node. These features may not work for some resource-constrained sensor node platforms.

3 Architecture Overview

The proposed architecture for body area networks consists of a set of base stations and a set of nodes, organized in a "fat-client, thin-server" style. A fat client includes a base station, which can be a local data aggregator in the form of a waist pack or a stationary one as part of the infrastructure. A base station is less physically size-constrained and thus can contain a larger battery or an AC adapter, a more powerful processor, and more memory. A thin client is in the form of a node, which can include sensors, actuators, and storage. The complexity of the node should be kept as low as possible for wearability, but at the same time wireless performance and adaptivity should not be sacrificed. A protocol that meets all of these requirements is data pulling, as opposed to data pushing in traditional approaches. By pulling, the base station serves the purpose of arbitration by eliminating intra-network packet collisions. Also, pulling can implement reliable communication with the same amount of traffic as ACK-based protocols, simply by re-pulling a lost packet. This section describes the distribution of responsibilities between the client and server in this architecture.

3.1 Base Station as a Client

In the proposed architecture, communications are all initiated by the base station as a client. Fig. 2 shows our base station architecture. Each base station can manage multiple servers as a group, and its responsibilities include

- Configuration
- Data Collection
- Management

Note that these responsibilities can be implemented by the list of commands shown in Table 1. These mechanisms can be used to implement a variety of policies dynamically, including RF channel arbitration, packet loss and remedies, and duty cycle management.

3.1.1 Configuration

To support in-field configuration of sensor nodes, the host typically pushes configuration commands with parameters to the nodes. This type of communication pattern actually matches well with our pulling protocol, since the host initiates the communication also, and the node replies with a

Table 1. Commands from the base station to sensor nodes

Commands	Type	Parameters
`get`	Configuration	dest, address
`set`	Configuration	dest, address, value
`xmit`	Data Collection	dest, count, period, offset
`sleep`	Management	dest, time
`ping`	Management	dest
`reset`	Management	dest

status code instead of data. Two types of messages are supported:

1. `get(dest, address)` is for the base station to read a parameter at address on the designated sensor node with ID dest.

2. `set(dest, address, value)` is to assign a value to the parameter address on the sensor node dest.

The base station and the nodes have a table in which an address is associated with its data size. If sensor nodes receive a get or set command from the base station, they extract address and look up the table to get the data size at the address. For `get`, they return the value at the adress back to the base station. For `set`, they update value at the address. These primitives can be used to configure RF transmission power, channel number, sampling rate, etc. Runtime reconfiguration may be done as a way to avoid interference or to compensate for low RF power. The meaning of the address and value may be user-defined.

3.1.2 Data Collection

Data is pulled by the client side by issuing the command `xmit(dest, count, period, offset)` to the node. As before, dest is the node ID and count is the number of packets requested. Sensor data collection can be easily implemented by calling the `xmit` function, which takes four parameters as shown in Table 1. By changing period, the third parameter, sensor nodes can send data in two different modes: burst and periodic modes. In the periodic mode, the base station has the sensor node send data periodically. The period is defined in period as follows.

`xmit(dest, count,` **period**`, offset)`

This call causes the sensor node to transmit sensor data every **period** time units. Burst mode means the sensor node transmits data continuously. This can be done by setting period to *0* as follows.

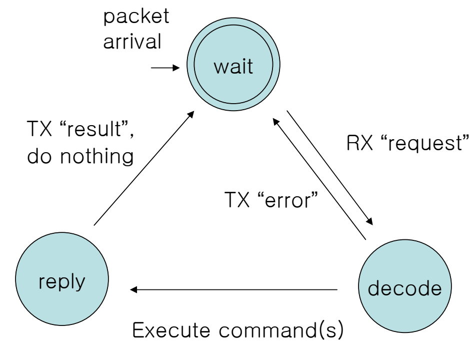

Figure 3. State diagram for processing requests on the server.

`xmit(dest, count,` **0**`, offset)`

That is, after the sensor node completes transmitting one packet, it transmits the next packet right away. The count parameter should be a positive number. The node transmits one packet per period for count consecutive periods, and then waits for the next command from the base station. In case count=1, the period parameter is ignored, as it starts waiting for the next command immediately. The maximum count can be decided by time synchronization period.

3.1.3 Management

Several commands can be issued by the client for control purposes: `ping(dest)`, `sleep(dest, time)`, and `reset(dest)` commands, to name a few. As commonly named, ping command is for a client to ask if a server with ID dest is reachable. The sleep command instructs the node to sleep for *time* milliseconds. Other control commands are for a node to reload its initial configuration, reboot, and load a new firmware.

3.2 Thin Server

The sensor nodes have minimal functions to manage resources and handle requests from the base station. They have a small execution engine to parse commands and minimal functions to execute the commands, which fit for the limited MCU capability and the small size memory. The engine processes requests from the base station and combines the functions to handle the requests.

The engine implements a state machine as shown in Fig. 3. Initially, the sensor node is in wait state and waits for packets from the base station. If the sensor node receives a packet, it moves to decode state. In decode state, if the packet is correct, then it extracts the command and its parameters from the packet and moves to reply state. Otherwise (the packet is not correct), it sends an error message to the base station and moves back to wait state. In reply state, it invokes the functions corresponding to the command with the parameters, sends the execution result to the base station, and moves to wait state.

4 Evaluations

4.1 Experimental Setup

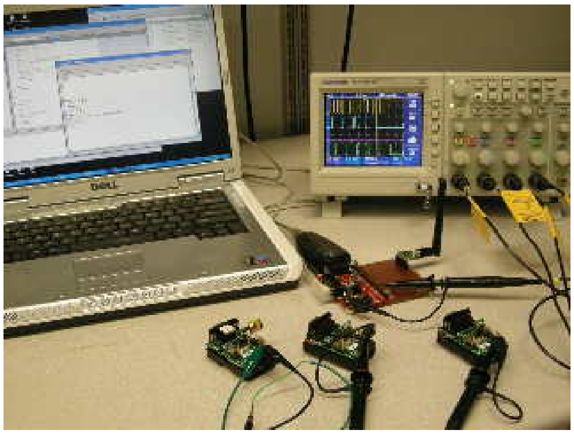

Figure 4. Lab setup for the performance evaluation

For the performance evaluation, we configure one MSP430F1611 evaluation board [8, 15] with a Nordic nRF24L01 RF transceiver [7] as the base station and three Eco nodes [11] as sensor nodes. Eco uses a Nordic nRF24E1 MCU [6], which contains an 8051-compatible core and a Nordic nRF2401 transceiver. Both the nRF24L01 and nRF2401 transceivers use the 2.4 – 2.527 GHz ISM band. The base station and Eco nodes transmit packets at 1 Mbps. On the base station, the MCU communicates with the transceiver over SPI off-chip at 4 MHz. On Eco nodes, the MCU core communicates with the transceiver over the internal SPI at 2 MHz.

The RF packet format is shown in Fig. 5. The packet size is configurable from 8 to 256 bits except for the preamble. The preamble is an 8-bit sequence, which is fixed in the transceiver hardware. The address field is configurable and the length is 8 to 40 bits. The packet checksum can be optionally enabled with either 8-bit or 16-bit CRC. Upon packet arrival, the hardware recognizes the preamble, matches the address, and performs the CRC check. If everything checks successfully, then the hardware extracts the payload to be transferred to the MCU. The maximum payload length is *(256 - Addr_length - CRC_length)* bits. In our experience, 3-byte address and 16-bit CRC yield the most robust results. For the experiment, we configure the transceiver for 33 bytes of packet size, which is divided into 1-byte preamble, 3-byte address, 27-byte payload, and 2-byte CRC.

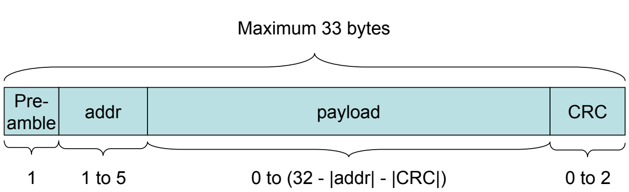

Figure 5. Nordic RF packet format for ShockBurst^{TM} **mode**

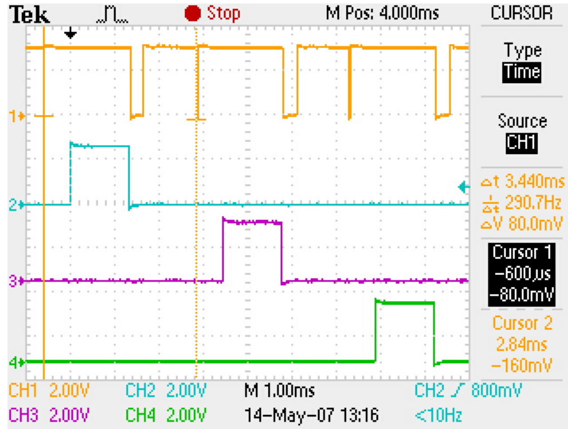

Figure 6. An oscilloscope screen-shot for base station's sequential access to three Eco nodes

4.2 Channel Utilization and Data Throughput

We measure the channel utilization under three different cases.

4.2.1 Sequential Transmission

The first case is that the base station sends xmit(1,1,0,0), xmit(2,1,0,0) and xmit(3,1,0,0) sequentially upon receiving the corresponding reply packets from Eco_1, Eco_2 and Eco_3. The base station repeats the above transmissions in order to measure the maximum throughput. Fig. 6 shows the transaction times measured with an oscilloscope. Channel 1 shows the interrupt signals from the RF transceiver. Narrow valleys indicate the *Tx-done* interrupts, while wide valleys indicate the *Rx-ready* interrupts. Channels 2, 3, and 4 are signals from the three Eco nodes. During a pulse period, an Eco reads the values from the triaxial accelerom-

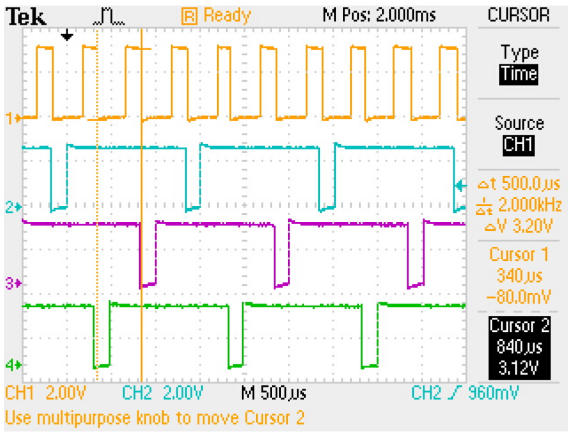

Figure 7. An oscilloscope screen-shot for periodic packet transmission from three Eco nodes to the base station

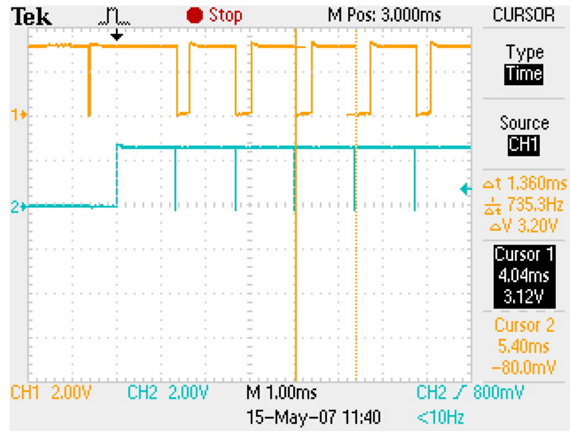

Figure 8. An oscilloscope screen-shot for burst packet transmission from an Eco node to the base station

eter, packetizes and transmits the data to the base station. It takes 3.44 ms for an Eco to finish one transaction, and it takes 10.32 ms for all three Ecos to finish the transactions. The sampling frequency is 96.9 Hz. The data throughput for 27-byte payload is 62.8 Kbps, and the channel utilization is 15.3%.

4.2.2 Interleaved Transmission

The second case is that these Eco nodes periodically transmit multiple packets for a single command from the base station. Transmissions are interleaved as shown in Fig. 7. The base station sends xmit(1,10,1.5ms,4ms) to Eco_1. After 2 ms, it sends xmit(2,10,1.5ms,3ms) to Eco_2. After 2 ms, it sends xmit(3, 10, 1.5ms, 2ms) to Eco_3. Sensor data packet is transmitted in the order of Eco_1, Eco_2, and Eco_3. Like Fig. 6, the first channel on the top shows the interrupt signals from the RF transceiver. The valleys are *Rx-ready* interrupts. The interrupt is raised in hardware when a packet arrives and is set off in software after copying the payload to the MCU. Channels 2, 3, and 4 are signals from the three Eco nodes. The length of the pulse plateau is 1340 μs. Each Eco sends sensor data packet every 1.5 ms. The base station receives a packet every 0.5 ms. The data sampling frequency is 666.7 Hz. The peak data throughput for 27-byte payload is 432 Kbps, and the channel utilization is 52.8%.

4.2.3 Burst Transmission

The last case is data transmission in burst mode. Fig. 8 shows data transmission from Eco_1 to the base station in burst mode. The base station sends xmit(1, 100, 0, 0) to Eco_1 and then Eco_1 transmits 100 data packets continuously. As described earlier, it takes 1340 μs to read the accelerations and transmit them. The decoding process takes another 20 μs. If the decoding process is counted, each sensor data packet can be transmitted every 1.36 ms. The peak data throughput for 27-byte payload is 159 Kbps, and the channel utilization is 19.4%.

In the above three cases, data transmission in periodic mode shows the best performance on the data throughput and the channel utilization. The first case shows the worst performance.

5 Application

We apply our pulling style MAC protocol to a wearable electrocardiograph (ECG) monitoring system. The innovative ECG bio-sensing devices from the Quasar company [12] shown in Fig. 9 are used as a testbed to evaluate our MAC protocol. The experimental prototype can make 10 ECG bio-sensing devices working simultaneously on the same wireless frequency channel and each of them can archive sampling rate at 200Hz.

6 Conclusions

We have proposed a low-complexity and high-throughput pulling-based MAC protocol for wireless body area networks. Unlike other traditional protocols, the base station as a fat client requests data from nodes as thin servers. The nodes passively reply to the request. There are several advantages of pulling. By base station's RF channel control, collision-free communication is guaranteed, which results in high channel utilization (52.8%) and high data

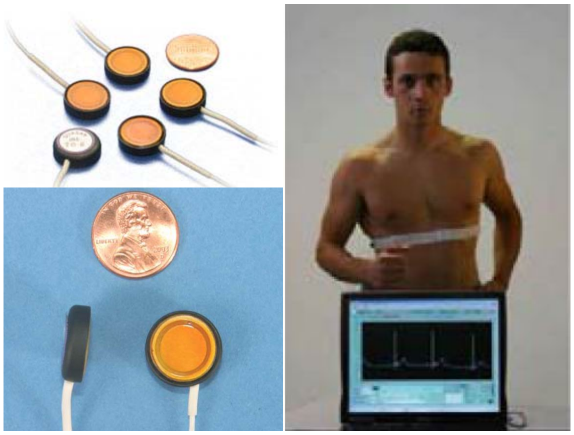

Figure 9. Quasar ECG Sensors

throughput (432Kbps). Sensor data is time-stamped without global time synchronization. When sensor nodes receive a request from the base station, they measure time difference between the request arrival time and the data sampling time, and send the data with the difference to the base station. The base station restores the sampling time from the difference. These prevent sensor nodes from wasting their memory and power to run a complicated MAC protocol and the time synchronization. The pulling enables wireless body area sensor applications to adapt to the environment dynamically after the deployment.

Acknowledgement

The authors would like to thank Jinsik Kim and Qiang Xie for their assistance with this work. The authors would also like to thank Ying Bai, Marty Steindorf, Robert Matthews, Andrew Hibbs and Kaichun Chanc at Quasar USA. This research project is sponsored in part by the National Science Foundation CAREER Grant CNS-0448668, UC Discovery Grant itl-com05-10154, the National Science Council (Taiwan) Grant NSC 96-2218-E-007-009, Information and Telecommunication National Scholarship (Korea IITA) , and Ministry of Economy (Taiwan) Grant 96-EC-17-A-04-S1-044.

References

[1] A. Y. Benbasat and J. A. Paradiso. A compact modular wireless sensor platform. In *IPSN '05*, pages 410– 415, Piscataway, NJ, USA, 2005. IEEE Press.

[2] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys '06*, pages 307–320, New York, NY, USA, 2006. ACM Press.

[3] D. Cypher, N. Chevrollier, N. Montavont, and N. Golmie. Prevailing over wires in healthcare environments: Benefits and challenges. *IEEE Communications Magazine*, 4:56–63, April 2006.

[4] E. Farella, A. Pieracci, D. Brunelli, L. Benini, B. Ricco, and A. Acquaviva. Design and implementation of wimoca node for a body area wireless sensor network. In *ICW '05*, pages 342–347, Washington, DC, USA, 2005. IEEE Computer Society.

[5] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02*, pages 88–97, New York, NY, USA, 2002. ACM Press.

[6] Nordic Semiconductor. nRF24E1: 2.4ghz rf transceiver with embedded 8051 compatible microcontroller and 9 input, 10 bit adc, March 2006.

[7] Nordic Semiconductor. nRF24L01: Single chip 2.4ghz transceiver, September 2006.

[8] Olimex Ltd. `http://www.olimex.com/`.

[9] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1(4):307–326, 2006.

[10] C. Park, P. H. Chou, and Y. Sun. A wearable wireless sensor platform for interactive art performance. In *PerCom 2006*, pages 52–59, Pisa, Italy, March 2006.

[11] C. Park, J. Liu, and P. H. Chou. Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring. In *IPSN '05*, pages 398– 403, Piscataway, NJ, USA, 2005. IEEE Press.

[12] Quasar. `http://quasarusa.com/usa/`.

[13] I. Rhee, A. Warrier, M. Aia, and J. Min. Z-mac: a hybrid mac for wireless sensor networks. In *SenSys '05*, pages 90–101, New York, NY, USA, 2005. ACM Press.

[14] A. S. Tanenbaum. *Computer Networks*. Prentice Hall PTR, Upper Saddle River, NJ, 4 edition, 2003.

[15] Texas Instruments Inc. MSP430x15x, MSP430x16x, MSP430x161x Mixed Signal Microcontroller Rev. E, August 2006.

[16] K. Whitehouse, G. Tolle, J. Taneja, C. Sharp, S. Kim, J. Jeong, J. Hui, P. Dutta, and D. Culler. Marionette: using rpc for interactive development and debugging of wireless embedded networks. In *IPSN '06*, pages 416–423, New York, NY, USA, 2006. ACM.

[17] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *IEEE INFOCOM*, pages 1567–1576, June 2002.

[18] C. Zhu and M. Corson. A five-phase reservation protocol (fprp) for mobile ad hoc networks. In *17th Annual Joint Conf. IEEE Comp. and Commun. Societies*, volume 1, pages 322–331, August 1998.