A Tree Based Self-routing Scheme for Mobility Support in Wireless Sensor Networks

Young-Duk Kim[1], Yeon-Mo Yang[2], Won-Seok Kang[1], Jin-Wook Kim[1], and Jinung An[1,*]

[1] Mobile Robot Lab., Daegu Gyeongbuk Institute of Science and Technology
{ydkim,wskang,jwkim,robot}@dgist.ac.kr
[2] Dept. of Electronics Engineering, Kumoh National Institute of Technology
yangym@kumoh.ac.kr

Abstract. Recently, WSNs (Wireless Sensor Networks) with mobile robot is a growing technology that offer efficient communication services for anytime and anywhere applications. However, the tiny sensor node has very limited network resources due to its low battery power, low data rate, node mobility, and channel interference constraint between neighbors. Thus, in this paper, we proposed a tree based self-routing protocol for autonomous mobile robots based on beacon mode and implemented in real test-bed environments. The proposed scheme offers beacon based real-time scheduling for reliable association process between parent and child nodes. In addition, it supports smooth handover procedure by reducing flooding overhead of control packets. Throughout the performance evaluation by using a real test-bed system and simulation, we illustrate that our proposed scheme demonstrates promising performance for wireless sensor networks with mobile robots.

Keywords: Wireless Sensor Networks, Handover, Self-routing, Mobile Robots.

1 Introduction

Recently, WSN (Wireless Sensor Network) [1] and mobile robot technology are the one of the most popular technologies for realization of ubiquitous networks. WSN can be widely used such as military, medical and industrial purpose. However, when we deploy WSN in multi-hop environments, a number of open problems can be observed because of limited bandwidth capacity and significant packet collisions by channel interference, and so on. In order to tackle these problems, [5] [6] [7] are proposed with BOP (Beacon Only Period) and LAA (Last Address Assignment) mechanisms. However, they do neither consider nodes mobility nor smooth route recovery mechanisms during the communication session. In addition, when the network traffic is significantly congested, existing schemes suffer severe packet collisions between beacons and other control packets. In this paper, we designed and

[*] Corresponding author.

F. Granelli et al. (Eds.): MOBILIGHT 2009, LNICST 13, pp. 114–124, 2009.

developed an efficient wireless sensor network system with autonomous mobile robot for smooth mobility support. In order to reduce the handover overhead and the latency of mobile robots which role mobile nodes in WSNs, we propose a tree based self-routing scheme. All sensor nodes and sink nodes are implemented on the TinyOS [2] system which is based on NesC [3]. In addition, we also developed a monitoring system which is able to collect and process data packets from every sensor node including the mobile robot.

The rest of this paper is organized as the follows. In Section 2, we review TinyOS architecture for our operating system platform and IEEE 802.15.4 MAC protocol as well as its improved versions with beacon scheduling. In Section 3, we illustrate the detail design architecture and implementation issues of our tree based routing scheme. Performance evaluation by real test-bed and simulation study is presented in Section 4. Finally, concluding remarks with future works are given in Section 5.

2 Related Works

2.1 TinyOS

TinyOS is developed in U.C. Berkeley and designed for exclusive operating system in wireless sensor networks. Most applications based TinyOS can be compiled into very tiny volume under 30Kbytes which is the optimal size for general specifications of wireless sensor nodes such as a small hardware device, small memory size, low CPU performance, and limited wireless channel resources. In addition, since TinyOS excludes unnecessary libraries and components, it can reduce extra overhead of the source code and produce minimum sized programs. For the more convenient application development, TinyOS is written in NesC language which is a component based architecture. The components of each application are connected to each other by using interfaces during the compile procedure. Although the grammar of NesC is similar to traditional C language, there are several differences between them such as types, development scheme, code size and etc. The other main features of NesC are as follows. It offers very convenient environment for application programming and the final code size is small enough to install on tiny sensor motes. However, NesC does not support a dynamic memory allocation mechanism, which may disturb intelligent computing processes.

In order to support simple scheduling service, the process of TinyOS defines a 2-level scheduling scheme which consists of tasks and events. The task is a process which is used for computing operations and procedure call operations. All tasks run in a FIFO (First In First Out) queue. When all tasks in the queue finish their processing, they minimize the CPU power consumption to reduce limited energy until other tasks are activated. Although a task is not able to be preempted by other tasks, it is able to be preempted by events. The event is a kind of process which has higher priority than task and is invoked usually when a hardware interrupt occurs or certain conditions are satisfied. When an event is produced by the interrupt, the related component is called and the wiring component in upper layer is also called in succession if it is connected to each other. At the same time, the related functions are transformed into tasks and stored in the FIFO queue.

2.2 IEEE 802.15.4 and Beacon Based Protocols

IEEE 802.15.4 [4] is the one of most representative protocols to support the communication between sensor nodes in wireless PANs (Personal Area Networks). In the basic mode, IEEE 802.15.4 usually operates in star network topology and requires a coordinator node to control the whole communication procedures between nodes by using beacon frames. However, it has severe limitations that it supports only 1 hop distance nodes from the coordinator, which is not suitable for multi-hop environments or multi-beacon enabled mesh networks. If we adopt the legacy IEEE 802.15.4 in the wireless mesh network with multiple paths, the network may suffer from significant performance degradation such as beacon collisions, failures of routing path and etc.

In order to tackle these limitations of IEEE 802.15.4, [5] [6] [7] proposed the BOP (Beacon Only Period) and the LAA (Last Address Assignment) algorithm for dynamic mesh networks. However, these schemes were not implemented with autonomous mobile robots and sensor nodes did not support a stable operating system such as TinyOS. Another limitation of [5] [6] [7] is that they show poor network performance because they do not solve packet collision problems between flooding packets for route discovery and beacon frames. Moreover, they do not suggest actual solution of node mobility support when the application requires seamless data services. Thus, throughout this paper, we propose an efficient network architecture for smooth mobility support with tree based the self-routing scheme.

3 Proposed Scheme

3.1 Association Process

It is necessary that each end node starts an association process to participate in PAN communication when it hears beacon frames from the coordinator. Our network scheme also uses the beacon policy like [5] [6] [7]. However, most WSNs have many-to-one communication paradigm, which means that all nodes transmit their sensing data upload direction. Thus, in order to make hierarchy architecture for efficient association, we define three node types, which are WC (Wireless PAN Coordinator), WR (Wireless Router), and WED (Wireless End Device). WC plays a role of a sink node and gateway by transmitting periodic beacon frames and collects data from WRs and WEDs. The collected data is forwarded to monitoring server for more specific processing such as management of alert message to user terminals. WR also periodically transmits beacon frames to neighbors and executes scheduling process with neighbors by exchanging beacon frames. WEDs are logically located in the end of the network and generate packets containing sensing data. Each packet of WEDs is forwarded to WC via WRs in every wakeup time of the superframe. In general IEEE 802.15.4 networks, there are only FFD (Full Function Device) and RFD (Reduced Function Device). However, in our work, we assume that FFD is able to be a not only WR but also WC. In addition, FFD can manage the PAN or make its own network without participating in other PANs.

In order to organize and synchronize the network, WC and WR transmit beacon frames to neighbors periodically. At first, one of WRs becomes WC if it does not hear any beacon frame from neighbors. Then WC starts to beacon with its network information such as beacon interval, identification, and information of its neighbors. When other WRs or WEDs try to scan the channel, they executes MLME_SCAN_request() process which is a MAC layer management entity in order to associate with parent node. In this situation, WR also can associate with another WR node and it calculates its own beacon schedule within the BOP length, which is executed by using received BTTSL (Beacon Tx Time Slot Length) information from its neighbors or parent node [7]. The channel scanning process in MAC layer is invoked by calling Network_Discovery_request() command, then each node records beacon information of accessible channels which is between 11 and 26. The scanning information is delivered to upper layer by SCAN_Confirm() function. Finally, by using MLME_SCAN_request() and MLME_SCAN_confirm(), a node transmits Assocation_request() primitive to the parent node with maximum signal strength which is derived from scanned beacon. This Assocation_request() is called by Network_Discovery_confirm() from network layer. Figure 1 illustrates the overall procedure of association.

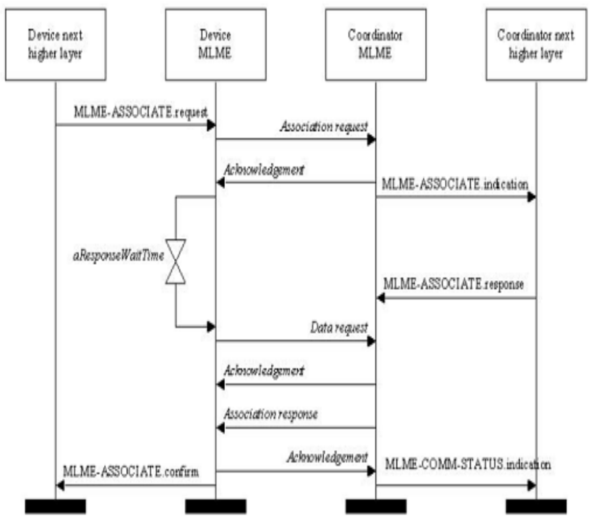

Fig. 1. Association process

3.2 Tree Based Self-routing Scheme

The traditional on demand routing protocols such as AODV (Adhoc On-demand Distance Vector routing protocol) [8] and DSR (Dynamic Source Routing protocol) [9] broadcast RREQ packets and receive RREP packets for route discovery. Even though the flooding scheme using these control packets is efficient for mobile ad hoc networks, it wastes network bandwidth and battery power in the wireless sensor network which consists of tiny sensors and motes. In addition, when relay nodes use the beacon frame for synchronization, it suffers significant packet collisions between beacons and other control packets. Moreover, when the duty cycle of each node increases, the flooding overhead also increases and it may result in network congestions. Thus, in order to reduce the control packet overhead for routing in network layer, we propose a self-routing approach by using association information between parents and child nodes in MAC layer.

When a node tries to participate in network communications, its parent node (WR) or coordinator (WC) may assign the address by using beacon frames with the LAA scheme. Therefore, after association procedure, the parent node can obtain the address of child node and the child node also can obtain the address of the parent node in tree based topology. The sharing address information between the parent and the child is stored in the simplified routing table which is described in table 1.

Table 1. An example of simplified routing table

Parent address		Child address	
Short address (16bit)	Long address (32bit)	Short address (16bit)	Long address (32bit)
7	0x0000000 000000007	8	0x0000000 000000008

Then, if a node receives incoming packets from the lower layer, it directly transmits to its parent node without using the RREQ flooding scheme. Consequently, the source node and relay node can guarantee rapid packet forwarding and reduce additional control overhead. In addition, since each node does not need to maintain and exchange the routing table information of whole network, it can not only resolve memory overhead but also accomplish self-routing.

3.3 Mobility Management and Route Recovery Process

When a node does not receive the expected beacon frame or a data packet in a certain interval, it believes that unexpected link failure or handover is taken placed in MAC layer. In this case, there are two desirable solutions to recover the routing path. The first one is that each node starts the process re-association to another parent node with our self-routing scheme mentioned in the previous section, which is simple and efficient approach from the fact that it does not require new route discovery process by using RREQ (Route Request) packet flooding. Consequently, this re-association scheme prevents unnecessary bandwidth wastes and prolongs the battery life time of each node. The other approach of path recovery is to use a route maintenance scheme by using network layer operation with RERR (Route Error) packet. Although this scheme is most common approach in mobile ad hoc networks, it significantly suffers from more route rediscovery delay and more bandwidth wastes.

Thus, we used the MAC layer re-association scheme and left the network layer approach as optional operation. After finishing the re-association procedure, MAC informs network layer with the updated route information and the node does not need to flood RREQ packets to the whole network. Since our network architecture intends to reduce the number of flooding of control packets, WR and WC execute the route discovery operation with association based tree routing. During the L2/L3 re-association procedure, WR acquires the relation information between child and parent. By using this information, the intermediate WR forwards the uplink data packet from the child node to the link of parent node. This relay process is continued until it arrives in WC. Consequently, the mobile node reduces handover latency and we can say that it is a self-routing scheme from the fact that the intermediate node does not depends on other routing information.

4 Performance Evaluation

4.1 Implementation of Test-Bed

As shown in figure 2, we developed sensor modules with CC2420 of TI Chipcon product as RF transceiver and ATMega128L as a main processor. The application was implemented for fire and atmosphere monitoring service such as temperature, gas, smoke, humidity and illumination. The gas information is classified into CO, CO_2, HCHO, SO_2, NO_2, and etc. This information is forwarded to WC in every seconds and KIP-AF (Knowledge Information Process Air/Fire data) shows the measured values in real time. If any emergent data arrive in KIP-AF, the server immediately transmits the alert message to the user terminal. For network entities, we used 1 coordinator, 8 relay WRs, and 50 WEDs with maximum 3hops. The transmission range was 30~40m and RF power control was set from 0 to -20dBm. The application used 40 bytes length packet and the duty cycle was maintained to 100%, which means all nodes transmit packets every interval in order to process real time data. All sensor nodes transmit their sensing data to coordinator and the gathered data is forwarded to KIP-AF which maintains the database for intelligent decision and further processing.

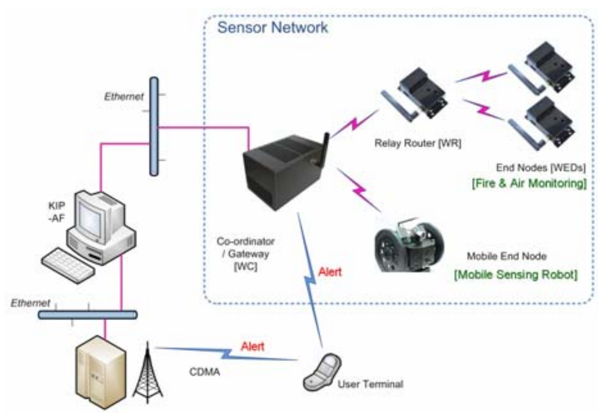

Fig. 2. Test-bed topology

For the mobility support, we used a mobile robot with sensing module which let the robot to be a mobile sensor node. The robot has 2 wheels and moved randomly with maximum 50cm/sec velocity. The robot platform is also designed and implemented on TinyOS and equipped with ATmega128L for the compatibility with sensing module. When the robot has mobility, there is an inevitable problem of link failure due to network handover or loss of LOS (Line of Sight). Then, the mobile robot tries to search another WR or WC with the best LQI value among the scanned candidates.

For the verification of reliable transmissions, we measured packet loss rate from end node to coordinator, which is logged and calculated in monitoring server. The measured results are shown in table 2 and the maximum loss rate is less than 4%. From the measurement our implemented network system is significantly reliable and the performance is well suitable for real time processing applications.

Table 2. Loss rate measurements

Performance Measurement			
Node ID	Loss Rate (%)	Node ID	Loss Rate (%)
A	3.38	L	2.94
B	1.88	M	3.71
C	2.93	N	0.91
D	2.62	O	1.91
E	2.63	P	1.97
F	1.87	Q	1.93
G	3.44	R	0.72
H	2.92	S	2.93
I	1.87	T	3.01
J	1.92	U	1.87
K	2.50	V	2.26

Fig. 3. Handover scenario for mobile robots

We also conducted the performance evaluation for mobility support by using “Sensor Network Analyzer (SNA)” of Daintree Networks [10], which is a commercial product for packet analysis. Figure 3 shows the route recovery scenario with mobile robots and the experiment is executed as follows. At first, we set up two WRs, named WR1 and WR2, which are associated to WC and two mobile WEDs associate with WR1. Then, WEDs move near to WR2, which means that they suffer the link failure.

After the re-association process, the mobile nodes have a new route to WC and they start to transmit data packets.

When the mobile node executes handover procedure in this experiment, the average handover latency, T_{HO}, is calculated as and figure 1 and expression (1).

$$T_{HO} = T_{beacon\_loss} + T_{asc\_req} + T_{ack} + T_{data\_req} + T_{ack} + T_{asc\_res} + T_{ack} \quad (1)$$

By using parameters as follow

T_{beacon_loss} : Interval of beacon loss due to handover
T_{asc_req} : Transmission time of association request frame
T_{asc_res} : Transmission time of association response frame
T_{data_req} : Transmission time of data request frame
T_{ack} : Transmission time of ACK frame

As shown in (1), since association duration is relatively short, the average handover latency is highly depends on beacon loss interval during the link failure. Hence, in order to minimize the beacon loss interval, we set the pending counter in MAC layer as 2. This means that after the mobile node does not hear beacon frame more than two times, it consider that the link is broken and executes the re-association procedure, immediately. In our implementation, we generated packets every second and set the beacon interval 1 sec. Thus, T_{beacon_loss} value is approximately 2 sec.

Table 3. Handover latency measurements

Trials	Handover starting time (min:sec:ms)	Handover finising time (min:sec:ms)	Handover latency (ms)
1	10:48:083	10:50:174	2,091
2	10:48:092	10:50:275	2,183
3	12:36:048	12:38:168	2,120
4	12:36:125	12:38:253	2,128
5	15:23:116	15:25:518	2,402
6	15:23:426	15:25:688	2,262

Table 3 shows the measurement results of handover latency of each trial. Around time 10:48:08 (min:sec:ms), both mobile robots which are associated to WR1, start to move to WR2. After the occurrence of route failure by handover, another route is established between WR2 and WEDs in 10:50:174 and 10:50:275, respectively. As shown in other results of trials, the average handover latency is under 2.5 sec. Thus we can say that it is possible to support mobility for communication between mobile nodes and the coordinator in wireless sensor networks.

4.2 Simulation Study

In addition to evaluation of our real test-bed, we also performed simulations to verify our proposed tree based self routing scheme comparing to original AODV protocol.

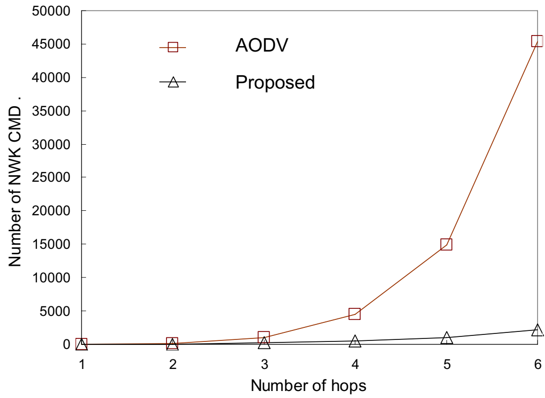

Fig. 4. The number of network commands

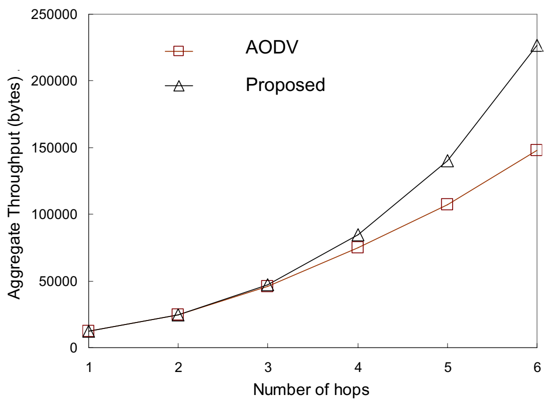

Fig. 5. Aggregated throughput

We used TOSSIM [11] with our beacon enabled MAC protocol and run the simulation for 1,000 seconds. All metrics are measured as a function of the number of hops and the network topology was the form of a perfect binary tree. For the traffic generation, we set the duty cycle at 50% with beacon order BO=8 and superframe order SO=7.

Figure 4 shows the number of network command packets as a function of the number of hops. When the topology is simple and the number of hops is smaller than 3, legacy AODV and our proposed scheme show similar performance. However, When the number of hops is higher than 4, our proposed scheme shows better performance because it does not need to flood the RREQ control packets and it only need to perform the association procedure. Hence, our proposed scheme has more opportunity to transmit data packets to neighbors with the limited wireless channel.

Figure 5 describes aggregated throughput during the simulation time. Since the intermediate node does not relay control packets such as RREQs and RREPs, it

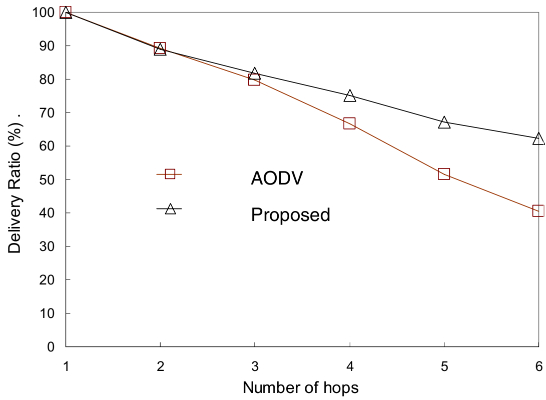

Fig. 6. Packet delivery ratio

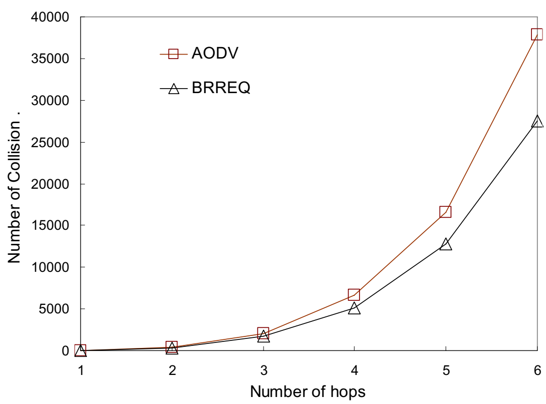

Fig. 7. The number of packet collisions

transmits more data packets during the channel access time. In case of AODV, it should wait another channel acquisition by the wireless contention after the route discovery procedure, which results in throughput performance degradation.

Figure 6 shows packet delivery ratios according to hop count and the result is correspondent to figure 5. This implies that when the number of hops increases, the performance gap between our scheme and AODV also increases. When AODV tries to establish the optimal route to the destination node, it should use RREQ flooding. These flooding packets may collide with data packets, which finally results in lower packet delivery ratio. Furthermore, the collision problem is more serious when the number of hops increases because the number of flooding increases exponentially in the tree based network topology. The number of packet collision is illustrated in figure 7. The collision performance shows a similar pattern with throughput results Therefore, in large wireless sensor networks, we can observe that the on-demand routing protocol like AODV is not suitable for communication with limited bandwidth.

5 Conclusion

In this paper, we have designed and implemented beacon mode based wireless sensor network system in TinyOS platform. For mobility support and smooth handover, we proposed a noble tree based self-routing scheme with association information between parent and child nodes. In other to verify the network performance, we implemented various sensing nodes as well as coordinator in real test-bed. Throughout performance evaluation with respect to loss rate, handover latency and several simulation studies, we showed that our network architecture accomplishes the reliable transmission for real-time processing service such as fire and emergency monitoring systems under heavy traffic environments.

As the future work, we plan to perform other extensive experiment to support QoS enabled packets such as voice and image. Then, we want to develop optimized and stable network architectures for WSN with mobility support.

Acknowledgement

This work is supported by DGIST and funded by MEST (Ministry of Education, Science and Technology) in Korea.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38 (March 2002)
2. Gay, D., Levis, P., Culler, D.: Software Design Patterns for TinyOS. LCTES (June 2005)
3. Gay, P.L., Culler, D.: The nesC Language: A Holistic Approach to networked Embedded Systems. In: ACM SIGPLAN 2003 (2003)
4. Draft IEEE Std. 802.15.4, Part 15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) (September 2006)
5. Jeon, H.-I., Kim, Y.S.: BOP and beacon scheduling for MEU devices. In: ICACT 2007 (Feburary 2007)
6. Jeon, H.-I., Kim, Y.S.: Efficient, Real-Time Short Address Allocations for USN Devices using LAA Algorithm. In: ICACT 2007 (Feburary 2007)
7. ISO/IEC JTC1/SC25, WiBEEM for Wireless Home Network Services - part 1, 2, 3, ISO standard (March 2008)
8. Perkins, C.E., Royer, E.: Ad-hoc on-demand Distance Vector Routing. In: Proc. 2nd IEEE Wksp. Mobile Comp. Sys. App. (February 1999)
9. Johnson, D.B., Maltz, D.A., Hu, Y.-C.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). In: Internet Draft, IETF Mobile Ad hoc Networks (MANET) Working Group
10. Sensor Network Analyzer (SNA), `http://www.daintree.net/products/sna.php`
11. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In: SenSys 2003 (November 2003)