

行政院國家科學委員會專題研究計畫 成果報告

子計畫二：即時應用之行動管理(I)

計畫類別：整合型計畫

計畫編號：NSC92-2213-E-002-092-

執行期間：92年08月01日至93年07月31日

執行單位：國立臺灣大學資訊工程學系暨研究所

計畫主持人：逢愛君

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 93 年 11 月 1 日

行政院國家科學委員會專題研究計畫成果報告

虛擬家網路環境：即時行動交談

子計畫二：即時應用之行動管理

計畫編號：NSC 92-2213-E-002-092

執行期限 92 年 08 月 01 日至 93 年 07 月 31 日

主持人：逢愛君 國立台灣大學 資訊工程系

一、摘要

【中文摘要】

行動設備上的省電設計已被廣為研究。本計劃探討兩個行動設備上省電的方法。一個是行動網際網路協定(Mobile IP)上省電的安全機制。此方法以改變路徑，來保護我們免於受到位址型攻擊。因為此法加強了網路層的安全保護，所以行動設備的密碼演算法就可採較省電的程序來達到與複雜程序相同效果。另一方法是多處理器上省電的任務排程法。我們對處理器速度不受限制且任務不可移轉的情況，提出的一個近似演算法；對處理器速度有上限且任務可移轉的情況提出了一個最佳演算法。

【英文摘要】

The energy-efficient design for mobile devices has been extensively studied. This project explores two energy-efficient approaches for mobile devices. One is the energy-efficient security mechanism of Mobile IP. This approach changes the routing paths so that it provides protection from IP-style attacks. Because of the enhancement of network side security, more energy-efficient encryption algorithms (i.e., less computation) of mobile devices can be adopted to achieve a good level of security. The other approach is the energy-efficient scheduling of tasks over multiple processors. We propose the approximation algorithm for multiprocessor scheduling with minimization of energy consumption without task migration. Also the optimal scheduling algorithm with task migration is presented.

Keywords: Mobile IP, Energy-Efficient Scheduling, Security, Power Management.

二、緣由與目的

The wireless technique progresses day by day. Users would like to be able to take their mobile devices, such as PDA or notebooks, from one network to another. Without plug-in power supply, mobile devices usually only use their on-board power to work until the power is out. Because the on-board power is limited, how to efficiently consume energy for mobile devices is a challenging issue. This project explores two energy-efficient approaches for mobile devices. One is the energy-efficient security mechanism for Mobile IP. The other is the energy-efficient task scheduling over multiple processors, where tasks share a common deadline.

For the first approach, each mobile node belongs to a home network in Mobile IP [1,2]. The home network assigns an IP address called "home address" to the mobile node. Home address is permanently assigned to the mobile node. The datagram destined to the mobile node must pass through a specific router, called "home agent (HA)", in the home network. Except the home network, other networks are regarded as foreign networks for the mobile node. When the mobile node roams to foreign networks, it will be assigned a temporary IP address, called "care-of-address (COA)". Foreign agent (FA) is a router on the visited network of mobile node, which provides routing

services to the mobile node while registered.

When corresponding nodes attempt to transmit a datagram to the mobile node, the datagram will firstly be sent to the home network. Then, HA will tunnel the datagram through FA to the mobile node. In hierarchical Mobile IP, the top layer of foreign agent is called “gateway foreign agent (GFA)”. Figure 1 illustrates the forwarding process.

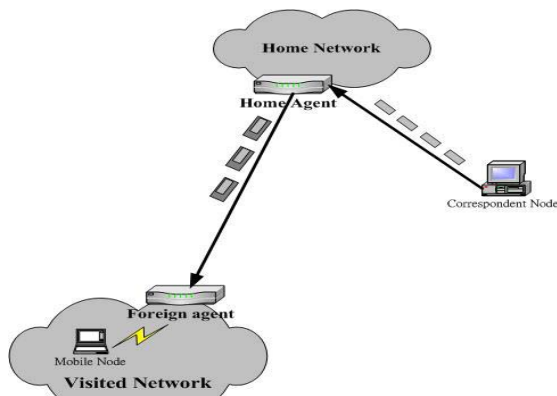


Figure 1: Forwarding Process.

In order to protect the datagram from attacking, the IETF working group [3] has proposed a scheme to apply IP security protocol (IPSec) to Mobile IP. However, both the complex computation and large key size of encryption algorithms in IPSec result in the rapid energy consumption for mobile nodes. Thus we propose a security-enhanced energy-efficient approach to provide network-side security so that less-computation encryption algorithms [4,5] of mobile nodes can be adopted. Our goal is to increase the variation of routing paths to protect the datagram from IP-style attacks. The detail will be described in section 3.

The second approach in this project is the energy-efficient task scheduling. Energy-efficient scheduling has been an active research topic in the past decade. In recent years, energy-efficient design has been outlined as a critical issue by the industry in business operations, e.g., [8], where various configurations of server farms are adopted. Some results [6,7] were proposed for uniprocessor energy-efficient scheduling.

This project will propose multiprocessor energy-efficient scheduling. Multiprocessor energy-efficient scheduling is often NP-hard under various application constraints [9]. Although some work [9,10,11] has been done, many previous results are mainly on heuristics-based energy-efficient scheduling. Distinct from many research results on heuristics-based scheduling, the goal of our approach to propose the approximation algorithm for multiprocessor scheduling with minimization of energy consumption without task migration and the optimal scheduling algorithm with task migration. The algorithms will be described in section 4.

三、Mobile IP 上省電的安全機制

Our energy-efficient security approach for Mobile IP increases the variation of routing paths to provide IP-layer protection. By using our proposed security-enhanced routing approach, less-computation encryption algorithms of mobile nodes can be adopted to support energy-efficiency. Three scales of the change of routing paths are proposed.

Small Scale Change

A mobile node adopts the foreign agent's IP address as its COA, and it will not change the foreign agent until it leaves the old foreign agent, and roams to other foreign agent. In our proposed mechanism, an assumption is made that there are at least two foreign agents in the same subnet. This is because the operator can set up several foreign agents if they would like to achieve robustness and well-performance. Since the foreign domain often faces unanticipated numbers of roaming mobile hosts, multiple foreign agents benefit to load balancing.

When the changing path mechanism starts, the mobile node tries to send Mobile IP registration request message to switch from the current FA to a new one. Figure 2 shows the process of small-scale (inter-FA) change for the mobile node. With performing the inter-FA switch for a mobile node, most

malicious attacks can be avoided since the routing path between the mobile node and FA is changed.

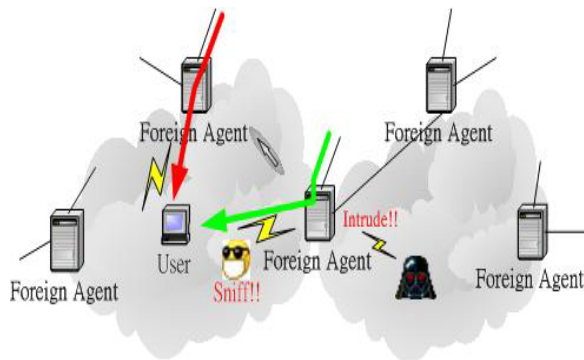


Figure 2: Changing Process for Small Scale.

Once the new FA receives the registration request message, it will forward the message to a FA with lower load. As shown in Figure 3, each FA maintains a table with all foreign agents' status and their queue sizes. If the queue size of a foreign agent exceeds the high threshold, a mobile node registered in this foreign agent will change the foreign agent to the other.

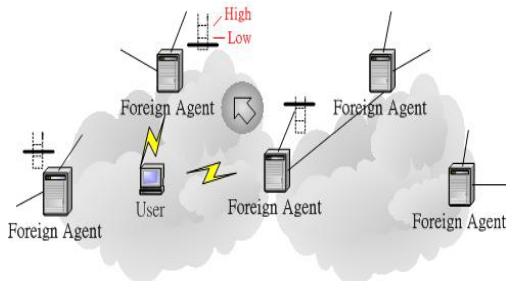


Figure 3: Small-Scale Change with Load Balancing

Medium Scale Change

The routing path's variation in a medium scale will be larger than one in a small scale. This is because we can change gateway foreign agent (GFA) arbitrarily in medium scale change. By adopting the medium-scale change of the routing path, it is hard for hackers sitting between GFA and FA to estimate the next hop for the delivered datagram. Figure 4 shows that each FA will periodically send "trace route" message to each GFA of upper layer to collect information shown as follows.

- Common hop number between the old and new paths : C
- Increasing hop number : I
- Each GFA's load : L
- A uniform random number from 0 to 1 : σ_i

Then we have the formula

$$(1) \quad abs(I * \alpha) + C * \beta + L * \gamma + \sigma_i$$

Based on (1), each FA chooses a new GFA with a smallest value.

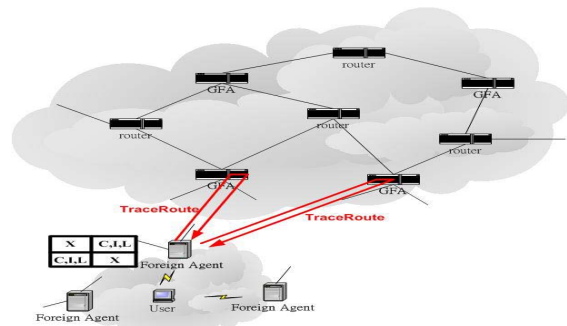


Figure 4: FA Periodically Send "trace route" Messages to Each GFA of Upper Layer

Large Scale Change

A large-scale change of the routing path involves with HA agent's switching. Large scale change makes our dynamic routing mechanism more powerful to provide a good level of security. Figure 5 shows the signaling procedure for HA reassignment.

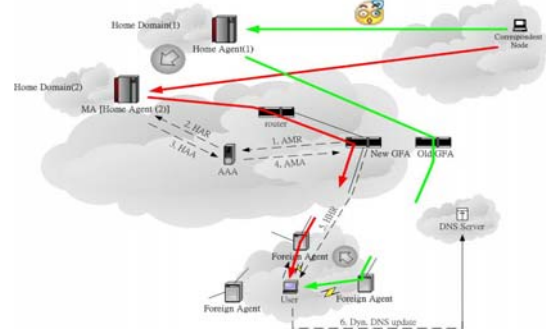


Figure 5: HA Reassignment Signaling Procedure.

四、多处理器上省电的任务排程法

We assume a homogeneous multiprocessor environment, where each of

the identical processors has the same energy consumption function of a given processor speed.

Consider a set T of independent tasks over M identical processors with an energy consumption function, where all tasks in T are ready at time 0 and share a common deadline D . Each task t_i in T is associated with a computation requirement equal to c_i CPU-cycles. The problem is to minimize the energy consumption in the scheduling of tasks in T without missing the common deadline D , where task migration is allowed.

The multiprocessor scheduling with the minimization of energy consumption with task migration is called MMEM. An efficient optimal algorithm for MMEM without constraints on the maximum processor speed is proposed in Figure 7. If $|T| \leq |M|$, it is clear that the executing of each task t_i on the processor i from time 0 to D at the speed c_i/D results in an optimal schedule. Our algorithm LTF-M adopts the Largest-Task-First strategy and time complexity $O(|T|\log|T|)$ come from the sorting of T in line 1.

The multiprocessor scheduling with the minimization of energy consumption without task migration is called MME. The input and output of the MME problem are as the same as their counterparts of the MMEM problem, where no task migration is allowed.

When the maximum available processor speed is infinite and the number of processors is greater than two, we present an algorithm with a 1.13-approximation ratio for MME problem. This algorithm, shown in Figure 8, adopts the largest-task-first (LTF) strategy.

Let p_m denote the load on the processor m . The load of a processor is defined as the total amount of the computation requirements of the tasks assigned to that processor. Let T_m denote the set of the tasks assigned to the processor m . Note that the task set T is a sorted set in a non-increasing order of the computation requirement of each task. Algorithm LTF assigns a task to the

processor with the smallest load by the task order in T . To achieve the minimal energy consumption, each task on the processor m

should be executed at the speed $\frac{\sum_{t_i \in T_m} c_i}{D}$. The time complexity of Algorithm LTF is $O(|T|\log|T|)$, which is dominated by the sorting of the tasks. Since each task is assigned to one processor without missing the common deadline, the correctness of Algorithm LTF is guaranteed. For the simplicity of representation, the schedule derived from Algorithm LTF is denoted as SC_{LTF} .

Algorithm: LTF-M

INPUT: (T, D, M) ;
OUTPUT: An optimal schedule with minimum energy consumption;

1. sort T in a non-increasing order of the computation requirement of each task; let $C \leftarrow \sum_{t_i \in T} c_i$;
2. **if** $\frac{C}{MD} > U_{\max}$ or $\exists t_i \in T$ such that $\frac{c_i}{D} > U_{\max}$ **then**
3. return non-existence of any feasible schedule;
4. let $t \leftarrow 1$;
5. **while** $i < |T|$
6. **if** $c_i > \frac{C}{M}$ **then**
7. schedule t_i to be executed at the speed $\frac{c_i}{D}$ on the processor M from time 0 to D ;
8. $C \leftarrow C - c_i, i \leftarrow i + 1$, and $M \leftarrow M - 1$;
9. **else**
10. break
11. let $S \leftarrow \frac{C}{MD}$ and $t \leftarrow 0$;
12. **while** $i \leq |T|$ **do**
13. **if** $t + \frac{c_i}{S} > D$ **then**
14. schedule t_i to be executed at the speed S on the processor $M - 1$ from time 0 to $t + \frac{c_i}{S} - D$ and on the processor M from time t to D ; $M \leftarrow M - 1$;
15. **else**
16. schedule t_i to be executed at the speed M on the processor at the speed S from time t to $t + \frac{c_i}{S}$;
17. $i \leftarrow i + 1$ and $t \leftarrow (t + \frac{c_i}{S}) \bmod D$;
18. return the schedule of all tasks;

Figure 7: Efficient Optimal Algorithm with Task Migration

Algorithm : LTF

INPUT: (T, D, M) ;OUTPUT: A feasible schedule $SC_{T,LTF}$ with minimum energy consumption

1. sort all tasks in a non-increasing order of the computation requirement of each task;
2. set p_1, p_2, \dots, p_M to 0 and T_1, T_2, \dots, T_M to \emptyset
3. **for** $i=1$ to $|T|$ **do**
4. find the smallest p_m ;
5. $T_m \leftarrow T_m \cup \{t_i\}$ and $p_m \leftarrow p_m + c_i$;
6. return the schedule $SC_{T,LTF}$ which executes all of the tasks in T_m ($1 \leq m \leq M$) at the speed $\frac{p_m}{D}$ on the processor m ;

Figure 8: Largest-Task-First Algorithm without Task Migration.

五、結論

This project has explored two energy-efficient approaches for mobile devices. The first one is the energy-efficient security-enhanced routing mechanism based on Mobile IP. We have developed a dynamic routing approach to increase the variation of routing paths to provide IP-level protection. Then less-computation encryption algorithms with energy efficiency of mobile nodes can be adopted. The other approach is the energy-efficient scheduling of tasks over multiple processors. We have proposed the approximation algorithm for multiprocessor scheduling with minimization of energy consumption without task migration. Also the optimal scheduling algorithm with task migration has been presented.

As a final remark, the results of this project have been accepted by 16th Euromicro Conference on Real-Time Systems.

六、參考文獻

- [1] Wireless IP Network Standard. *3GPP2 P.S0001-A version 3.0.0*, Jul. 2001.
- [2] <ftp://ftp.rfc-editor.org/in-notes/rfc3846.txt> IETF RFC3846.
- [3] John K Zao, Matt Condell. *Use of IPSec*

in Mobile IP. Internet Draft, Internet Engineering Task Force, <draft-ietf-mobileip-ipsec-use-00.txt>, November 1997.

- [4] Charlie Kaufman, Radia Perlman and Mike Speciner. *Network Security: Private Communication in a Public World*, Second Edition. Prentice Hall PTR, April 15, 2002.
- [5] *Introduction to cryptography, Part 2: Symmetric cryptography* <http://www-106.ibm.com/developerworks/security/library/s-crypt02.html>
- [6] F. Yao, A. Demers and S. Shankar, A Scheduling Model for Reduced CPU Energy," *IEEE Annual Foundations of Computer Science*, pages 374-82, 1995.
- [7] Jian-Jia Chen, Tei-Wei Kuo, and Chia-Lin Yang, Profit-Driven Uniprocessor Scheduling with Timing and Energy Constraints, *ACM Symposium on Applied Computing*, 2004.
- [8] J. S. Chase, D. C. Anderson, P. N. Thakar, and A. M. Vahdat, Managing Energy and Server Resources in Hosting Centers, *ACM Symposium on Operating Systems Principles (SOSP)*, Oct. 2001.
- [9] F. Gruian and K. Kuchcinski, LEneS: Task scheduling for low energy systems using variable supply voltage processors, *Proc. Asia South Pacific Design Automation Conference*, pages 449-455, 2001.
- [10] F. Gruian, System-level design methods for low-energy architectures containing variable voltage processors, *Workshop on Power-Aware Computing Systems (PACS)*, Nov. 2000.
- [11] Yumin Zhang, Xiaobo Hu, and Danny Z. Chen, Task scheduling and voltage selection for energy minimization, *Annual ACM IEEE Design Automation Conference*, pages 183-188, 2002.