A Method of Designing Seamless Connectivity Algorithm in Ubiquitous Networks

Utsav Sinha
Indian Institute of Science, Bangalore
Bangalore 560012, Karnataka, India
Email: sinha.ju@gmail.com

Pallapa Venkataram
Indian Institute of Science, Bangalore
Bangalore 560012, Karnataka, India
Email: pallapa@ece.iisc.ernet.in

***Abstract*—The idea of Ubiquitous computing is gaining much importance because of mobile devices with capabilities like multiple interfaces and more computational capabilities. Success of Ubiquitous applications depends on how effectively users are provided seamless connectivity. In Ubiquitous networks seamless connectivity encompasses providing seamless migration to ubiquitous nodes as well as locating and providing necessary resources of the networks for running these applications. In this paper we propose an agent based scheme of seamless connectivity in ubiquitous networks. One static cognitive agent registers, monitors and predicts ubiquitous node movements in the network. To facilitate in seamless migration of the nodes to any other network we deploy mobile cognitive agents to serve the purpose. Mobile cognitive agents also ensure that necessary resources required for running a particular ubiquitous application are reserved for the migrating node in foreign network. Prediction helps us deploy agents to networks prior to the node's migration. We consider Mobile SCTP as our underlying transport layer protocol, which facilitates seamless migration of the ubiquitous nodes from one network segment to another. Several simulations on ubiquitous node movements have shown that we can provide seamless connectivity to the nodes successfully with minimum hand-off delay and data loss if we can successfully predict the movement pattern of the ubiquitous node.**

I. Introduction

Ubiquitous computing involves the idea of thousands of devices being networked together distributed at all scales throughout our daily life with the user being unaware of that. This is why Ubiquitous computing is also known as pervasive or invisible computing. Ubiquitous computing also involves several heterogeneous networks and inter-networking between them. A mobile user using a ubiquitous application may keep moving in and out of several of these networks of different technologies. So one of the key challenges in providing ubiquitous application is seamless connectivity of the ubiquitous nodes (UNs). Seamless connectivity provides seamless and uninterrupted sessions to the nodes even as they move between different networks. Simultaneously, guaranteeing resources to the UNs for running ubiquitous applications as they migrate in between different network segments is also a challenge.

In this paper we propose a cognitive agent based solution for seamless connectivity of UNs using a ubiquitous application. This approach involves a $Ubiquitous\ System$, which runs a $static\ cognitive\ agent$ (SCA) & maintains a $User\ Database$. The SCA is responsible for registering a UN when it comes up for the first time in the network. It also monitors the movement pattern of the UN in the network. Based on the movement pattern the SCA predicts to which network the UN can migrate to. At this point a $mobile\ cognitive\ agent$ (MCA) is created & deployed by the SCA to the predicted network. The mobile agent carries credential user level data to the new network for registering the UN in the network even before the UN actually migrates to that network. It also interacts with the local administration to make sure that necessary resources are reserved for the UN so that it can seamlessly keep using the previous application . This approach considers mobile SCTP (mSCTP) as underlying transport layer protocol. The rest of the paper is organized as follows. In section 2 we discuss about the related works in seamless connectivity. Section 3 adds definitions of terminologies used in the work. In section 4 we discuss the proposed work in details. Section 5 and 6 include the simulation environment and simulation results respectively. In section 7 we add concluding remarks.

II. **Related works on Seamless Connectivity**

The work in [3] proposes the idea of seamless handover of content being viewed by the user even as the user switches between different terminal points and also changes network connections. The authors have developed a user environment meta data (UEM) server. This server describes terminal capabilities and status. It integrates CC/PP (Composite Capabilities/ Preference Profiles) as defined by W3C and presence information as defined by IETF, with some enhancements. CC/PP describes terminal capabilities such as hardware and software platform, browser characteristics and also user preferences in using them. Thus it provides a content adaptation scheme for seamless viewing of content data. A location resolution (LR) server resolves the location of the content that is compatible with target terminal capability. The work in [1] provides seamless connectivity to MNs by providing information on user movement. The Ubi-system model for providing seamless migration consists of Ubi-SubSystem and Soft switch. The Ubi-SubSystem registers a new user, provides him/her with information server URL based on user contexts and also challenges any new or suspicious user in the network. The Soft-switch is responsible for monitoring and predicting the movement of the user to a new network and also providing seamless migration based on the predictions. Mobile IP[2] relies on entities like FA(foreign agent) & a HA(home agent). When a MN goes to a new network it obtains either a Foreign Agent care-of-address or a collocated care-of-address. This new address is registered with the FA in the foreign network. The FA further forwards this to the HA in the original network. Any data packets destined to the MN from a correspondent node (CN) is directly tunneled to the MN by the HA, which intercepts any data packets meant for the MN. In addition to this, the first time a CN sends data to the MNs previous address, the HA sends a binding-update to the CN informing the CN about the MNs newly obtained IP address. Any further data from the CN are tunneled to the MN directly.

III. Definitions

We give definitions for some of the terminologies used in this paper.

A. Mobile SCTP (mSCTP)

SCTP[4] (stream control transmission protocol) is a transport layer protocol in the IP protocol stack. A key feature of SCTP is multi-homing[4], where an SCTP enabled host can be configured with more than one IP address at the same time. mSCTP is a DAR (dynamic address reconfiguration) extension of ordinary SCTP and allows an endpoint to reconfigure its IP address dynamically by sending address configuration (ASCONF) messages to its peer. The sender host sends an add-IP message to correspondent node (CN). After receiving an ACK, the sender sends set-primary message to the CN. The CN sets the IP address as the primary address of the sender.

B. Cognitive Agent (CA)

Cognitive science is to design intelligent agents which are software artifacts, showing intelligent behavior in complex situations. A $Cognitive\ Agent$ is flexible and autonomous. It can learn, adapt and act according to changes in the environment. They can also interact with other agents and act collaboratively in a goal oriented manner. An MCA is such a

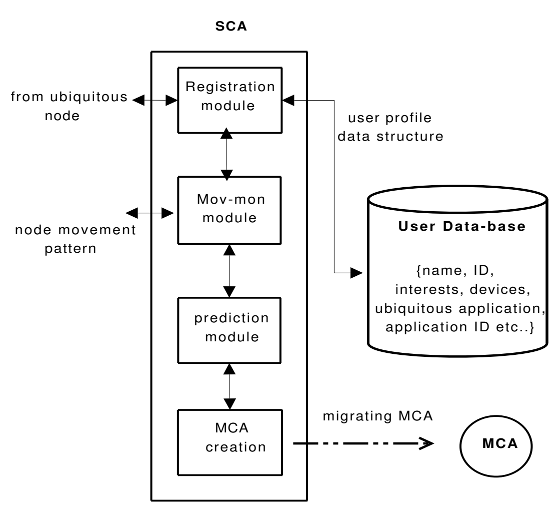


Fig. 1: Seamless Connectivity architecture

code that can migrate from one platform to another in between execution of a code. An SCA on the other hand is static in one platform.

C. User Database

The Ubiquitous System maintains a $User\ Database$ of all the UNs currently in that network. This database consists of all credential user data required for user registration & authentication in that network. A user data-structure is maintained for every UN. This data-structure holds information on user identity (name, age, occupation etc..), present purpose of visit, interests, the Ubiquitous application (name, type, application ID), device IP address etc. Whenever a new node comes to the network, the user profile data structure has to be stored in the User Database for node registration.

IV. Proposed work

A. Agent based seamless connectivity system architecture

The seamless connectivity system architecture consists of the following entities:

1) **User Database:** The Ubiquitous System maintains a $user\ database$ of all the UNs currently in that network. This database consists of all credential user data required for user registration and authentication in that network. Information on user identity (name, age, occupation etc.), present purpose of visit, interests, the Ubiquitous application, device IP address of every UN is stored as a user profile data structure.
2) **Static Cognitive Agent:** The Ubiquitous System also maintains one SCA for one particular ubiquitous application. The SCA consists of 3 modules. User $registration\ module$ is responsible for collecting user profile data from the new UNs, creating the user profile data-structure and handing it over to the user database for node authentication and registration. $Movement\ monitoring\ module$ monitors UN movement in a network. $Prediction\ module$ runs a prediction algorithm to predict the next network a UN can migrate to based on present and past movement patterns of the node.
   The Ubiquitous System model is shown in Fig.3.
3) **Mobile Cognitive Agent:** The SCA is capable of creating instance of an MCA, and deploying it in the predicted network prior to a node's migration to that network. The MCA carries the user profile data-structure of the UN with itself to predicted network. MCA also ensures that all necessary resources required for running a particular application are reserved for the UN prior to its migration to the foreign network.

B. Mobile Cognitive Agent based resource reservation

Here we adopt a modified version of an agent based resource reservation scheme proposed in [5]. The scheme proposed in [5] maintains a dedicated agent in every network for admission control in that network. But as we are concerned about ubiquitous environment we do not consider a centralized

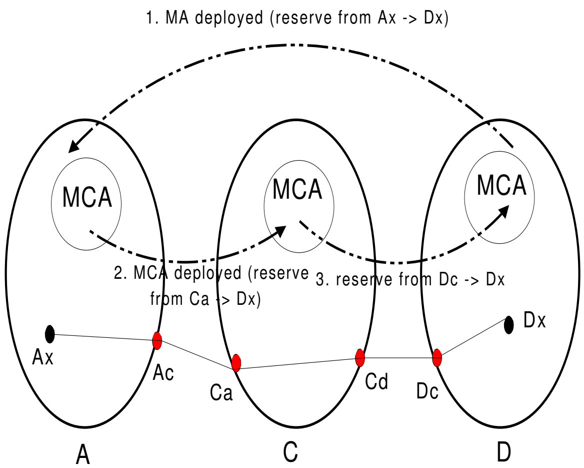


Fig. 2: Mobile Cognitive Agent based resource reservation

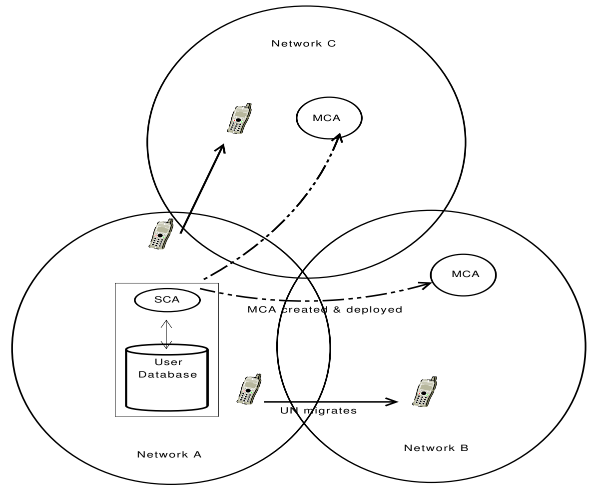


Fig. 3: Overview of Ubiquitous System model

agent based solution here. We consider MCAs doing the job of resource reservation on behalf of the migrating UN. When an MCA migrates to a foreign network it takes part in link-state routing passively as stated in [5]. It obtains a resource map of that domain. The MCA is aware of the starting point (the correspondent node/fixed server in this case)from where resource is to be reserved along the path. The MCA identifies if the starting point is in the same domain as itself. Once it finds the point in the same domain reservation and admission control is performed from that point to the destination point (the new address of the UN in the foreign network in this case) defined by the routing protocol. When the MCA realizes that the point is in a different domain, it deploys a copy of the MCA to that domain. The deployed MCA then determines if the destination point is in the same domain. If not it again sends out a copy of the MCA to the next domain defined by the routing protocol. It identifies the point where the traffic enters the next domain. It performs admission control from the starting point in its own domain to the ingress point in the next domain. It requests the MCA in the next domain to perform admission control from the ingress point to the destination point in that domain. Thus the admission control is performed in steps from the starting point to the destination point. MCAs in every domain can set up police points at the ingress and egress points to ensure that the rate of priority traffic stays within commitments. This has already been explained in [5]. The resource reservation scheme is shown in Fig.2.

C. Overview of hand-off scheme

After a node gets registered in a network, the mov-mon of SCA monitors the movement of the node in the network. At a certain point the pred-module predicts the next network. At this point the UN has crossed the $cross-over$ circle and is at the network boundary. At this point it is in the coverage area of both networks. So at this point the node can obtain a new IP address in the neighboring network (either from a DHCP server in case of WLAN or through an IP auto-configuration in MANET). At this point the MCA is created and deployed by the SCA to the predicted network. The MCA carries the user profile data-structure, and the newly obtained IP address of the node with itself. And as mentioned above, after the MCA ensures that the UN can migrate to the next network, it asks the node to undergo an ASCONF. The node undergoes an address configuration with its peer by sending ASCONF messages as discussed previously. Thus the UN migrates seamlessly from one network to next. In the foreign network the UN is monitored by the MCA. Here the MCA performs all tasks that are performed by the SCA in the original network. The MCA registers the UN in foreign network prior to the UN's migration. The MCA monitors node movement and also predicts where the UN can migrate to next. This MCA finally migrates to predicted network for providing seamless migration to the UN. The MCA thus accompanies the UN wherever it migrates. The delay in the whole scheme is,

$$t_{scheme} = t_{MCA} + t_{res} + t_{IP} + t_{hand-off}. \quad (1)$$

where t_{MCA} is time taken for creation and migration of the MCA, t_{res} is time taken for the agent based resource reservation, t_{IP} is time taken for obtaining a new IP address (from DHCP server in case of WLAN or address auto configuration in case of MANET)and $t_{hand-off}$ is the hand-off delay of an mSCTP based host. t_{scheme} can definitely be considered to be very less than the physical migration time of the UN from one network to a neighboring network. $t_{hand-off}$ is the time taken for the exchange of ASCONF messages with its peer i.e exchange of ADD-IP and Set-Primary messages, and also the hand-off delay in the underlying link layer $t_{link\_hand-off}$.

$$t_{hand-off} = t_{ASCONF} + t_{link\_hand-off}. \quad (2)$$

$$t_{ASCONF} = 2 * (L_{ASCONF} \div BW_{available} + t_{pr}).$$

where L_{ASCONF} is the size of an ASCONF chunk.

D. Prediction Algorithm

Next we discuss how the prediction algorithm works. The network model for the prediction algorithm is as shown in

Fig.4. We have assumed a 4-neighbor network model.We have divided a network into 4 equal quadrants. The center of the network has co-ordinates $(0, 0)$. Predictions take into account both the past movement patterns and the present movement patterns of the UN in the network. Past movement patterns include how many times the UN has been in a particular network and how many times from which quadrant the UN has migrated to which neighboring network. Present movement pattern is obtained from movement monitoring of the UN. When the UN cuts the $cross - over$ circle, we calculate the angle of the $cross - over$ point from the center of the network. For example, if this angle has a value in between $[-\frac{\Pi}{4}, \frac{\Pi}{4}]$ we can say that the UN is probably migrating to network 0 (refer Fig.4). We combine both these movement patterns and calculate the highest probable network the UN can migrate to. The prediction algorithm and calculation of the highest probable network is shown in algorithm 1.

Algorithm 1 Prediction algorithm

Begin
UN crosses $cross - over$ circle with radius R_1.
$get(x_{cross-over}, y_{cross-over})$ and calculate angle of movement $\theta(x_{cross-over}, y_{cross-over})$.
calculate $p_i^j\ \forall\ i, j\ from\ 0\ to\ 3$, where p_i^j is the probability that MN will migrate to network i from quadrant j based on $\theta\ (x_{cross-over}, y_{cross-over})$.
calculate $r_i^j\ \forall i, j\ from\ 0\ to\ 3$, where r_i^j is the probability that node has migrated to network i from quadrant j based on past movement patterns.
combining both probabilities: $P_i^j = a * p_i^j + (1 - a) * r_i^j\ \forall\ i\ 0\ to\ 3$ where P_i^j is the probability that MN will migrate to network i, and a is a pre-assigned weight.
if a$\leq$(1-a) **then**
  a=0.5
else
  a=a-ϵ
end if
where a is decremented by ϵ after every run
highest of $P_i^j\ \forall i, j\ 0\ to\ 3$ gives the predicted network i.
End

V. Simulation environment

The simulation is done using C language. We have considered a random movement of the node in a network. A node originates at a random point (X_0, Y_0). We can write,

$$X_n = X_0 + W_1 + W_2 + ... + W_n \tag{3}$$
$$Y_n = Y_0 + V_1 + V_2 + ... + V_n$$

where W_i and V_i are iid increments. (X_n, Y_n) will give the position of the node at the nth step. W_i, V_i take value $+1$ with probability $\frac{1}{2}$ and -1 with probability $\frac{1}{2}$. We have simulated node movement as above in a network for 5000 times. We predict the next network only when the UN cuts the $cross - over$ circle for first time. Simulation has been done considering two pairs of values for R_1 and radius of network R. We took the values for these as $13, 15$ and $8, 10$ respectively.

The mSCTP based UN sends 5,000 bytes of data to a correspondent node. The node undergoes a soft-hand-off as it changes its primary address from IP_1 to IP_2. The parameters considered for simulations are the following. 1 MTU is 2,500 bytes long, initial slow start threshold $ss_{threshold}$ is set to $4 * MTU$, congestion window $cwnd$ is set to $2 * MTU$, size of an ASCONF chunk is also set to $1 * MTU$. And the default values for timeout ($t_{timeout}$), round trip time (t_{rtt}) and propagation delay (t_{pr}) are set to $20ms,\ 1ms\ and\ 0.1ms$ respectively.

VI. Simulation results

Fig.5a and Fig.5b show the graph for false alarm probability p_{fa} vs n and prediction probability p_{err} vs n for both the cases mentioned beforehand. $False\ alarm$ occurs when we make a prediction but the UN does not move out of present network. Prediction error occurs when we make a wrong prediction. From Fig.5a we can see that p_{fa} decreases with higher values of n. This is because the more we increase the value of (X_0, Y_0), higher is the probability that the node will actually migrate out of the present network. and Fig.5b shows that p_{err} increases with higher values of n. This is because with higher values of n the node movement becomes even more random and results in higher prediction errors. Fig.6a shows the graph for p_{fa} vs the error probability p_{err} of the prediction algorithm, which shows that the false alarm probability decreases as the prediction error probability increases and vice versa.

Fig.6b shows graph of time (seconds) taken for sending data vs the sequence number of packet successfully sent. The sequence number of packets having multiple plots along the time axis are the sequence numbers that either suffer a packet drop in the channel or when there is an mSCTP hand-off. Here handover occurs at 10.557983 seconds, and handover delay is 0.001840 seconds. Fig.7a and Fig.8b show the results for the mSCTP based handoff delay $t_{hand-off}$ and data loss for different instants of handoff. Fig.7b shows the total delay required in the seamless connectivity scheme t_{scheme} for different handoff instants. and Fig.8a shows results for the delay in seamless connectivity scheme vs the delay entailed in resource reservation scheme t_{res}. It is clear that assuming the MCA migration time t_{MCA} and IP address configuration time t_{IP} as constants, the delay in the whole scheme depends on the hand-off delay and resource reservation delay.

VII. Conclusion

An efficient scheme for seamless service to nodes in Ubiquitous environment is very important. Apart from providing a scheme for seamless soft-hand-off, it is also very important to ensure that enough resources are provided to the UNs for seamless service delivery. In this paper we adopted an agent based seamless handoff scheme for Ubiquitous application. We also provide a modified agent based resource reservation scheme already done in [5] to reserve resources required for

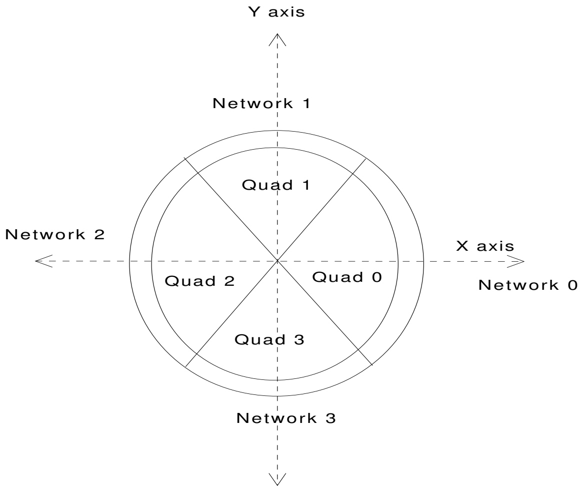
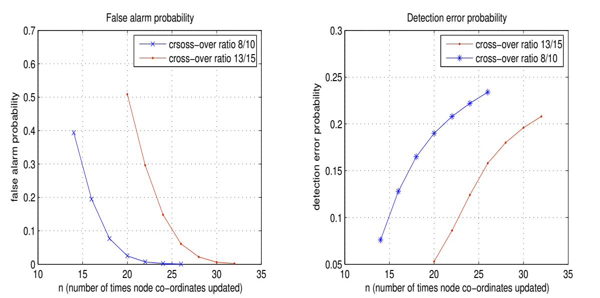


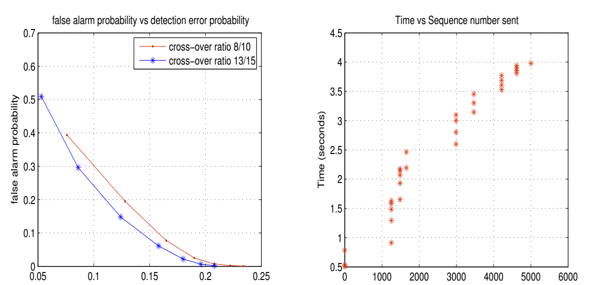
Fig. 4: A 4 neighbor network model



(a) False Alarm probability Vs. n (number of times node coordinates updated)

(b) Prediction Error probability Vs. n (number of times node coordinates updated)

Fig. 5: Simulation results

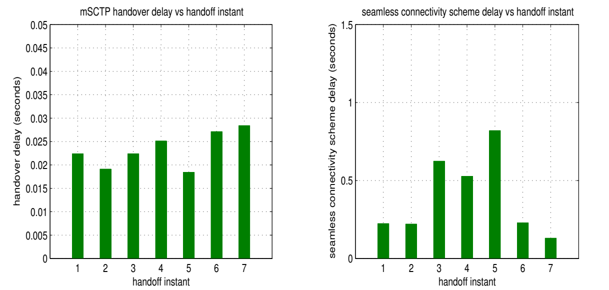


(a) False alarm vs Prediction error probability

(b) Time(seconds) vs sequence number of packets successfully sent

Fig. 6: Simulation results

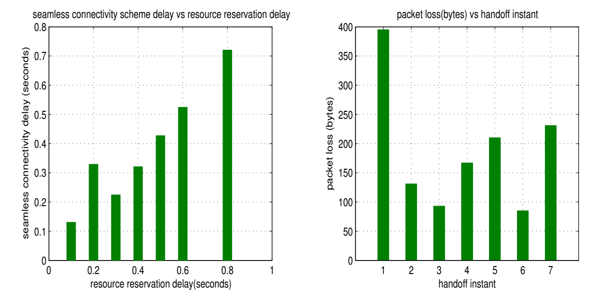
a Ubiquitous application. By developing the scheme based on user movements and by using MCAs we ensure better seamless service delivery to the nodes. Considering mSCTP as the transport layer protocol and also the MCA based pre-registration and resource reservation scheme we ensure a minimum percentage of data loss and a low latency hand-off.



(a) mSCTP handoff delay vs handoff instant

(b) Seamless connectivity scheme delay vs handoff instant

Fig. 7: Simulation results



(a) Seamless connectivity scheme delay vs resource reservation delay

(b) Data bytes lost vs handoff instant

Fig. 8: Simulation results

References

[1] P.K.Wali & P.Venkataram, An Effective Seamless Connectivity Scheme for Ubiquitous Applications, MoMM 2008, November 24-26, 2008, Linz, Austria.(c) 2008 ACM 978-1-60558-269-6/08/0011

[2] J.Redi & P.Bahl, Mobile IP: A Solution for Transparent, Seamless Mobile Computer Communications, Fuji-Keizais Report on Upcoming Trends in Mobile Computing and Communications, 1998

[3] Kawarasaki, M.; Ooto, K.; Nakanishi, T.; Suzuki, H.; , "Metadata driven seamless content handover in ubiquitous environment," Applications and the Internet, 2004. Proceedings. 2004 International Symposium on , vol., no., pp. 287- 290, 2004

[4] Siddiqui, F.; Zeadally, S.; , "SCTP multihoming support for handoffs across heterogeneous networks," Communication Networks and Services Research Conference, 2006. CNSR 2006. Proceedings of the 4th Annual , vol., no., pp.8 pp.-250, 24-25 May 2006

[5] Olov Schelen, Stephen Pink, Resource reservation agents in the internet, in Proc.International Workshop on network & Operating system support for digital audio & video,1998,153-156

[6] Ubiquitous computing, Mark Weiser, Xerox Corporation

[7] Danny B. Lange and Mitsuru Oshima. 1999. Seven good reasons for mobile agents. (em)Commun. ACM(/em) 42, 3 (March 1999), 88-89. DOI=10.1145/295685.298136 http://doi.acm.org/10.1145/295685.298136