Cooperation Between Stations in Wireless Networks

Andrea G. Forte
Department of Computer Science
Columbia University
Email: andreaf@cs.columbia.edu

Henning Schulzrinne
Department of Computer Science
Columbia University
Email: hgs@cs.columbia.edu

***Abstract*— In a wireless network, mobile nodes (MNs) repeatedly perform tasks such as layer 2 (L2) handoff, layer 3 (L3) handoff and authentication. These tasks are critical, particularly for real-time applications such as VoIP. We propose a novel approach, namely Cooperative Roaming (CR), in which MNs can collaborate with each other and share useful information about the network in which they move.**

We show how we can achieve seamless L2 and L3 handoffs regardless of the authentication mechanism used and without any changes to either the infrastructure or the protocol. In particular, we provide a working implementation of CR and show how, with CR, MNs can achieve a total L2+L3 handoff time of less than 16 ms in an open network and of about 21 ms in an IEEE 802.11i network. We consider behaviors typical of IEEE 802.11 networks, although many of the concepts and problems addressed here apply to any kind of mobile network.

I. Introduction

Enabling VoIP services in wireless networks presents many challenges, including QoS, terminal mobility and congestion control. In this paper we focus on IEEE 802.11 wireless networks and address issues introduced by terminal mobility.

In general, a handoff happens when an MN moves out of the range of one Access Point (AP) and enters the range of a new one. We have two possible scenarios:

1) If the old AP and the new AP belong to the same subnet, the MN's IP address does not have to change at the new AP. The MN performs a L2 handoff.
2) If the old AP and the new AP belong to different subnets, the MN has to go through the normal L2 handoff procedure and also has to request a new IP address in the new subnet, that is, it has to perform a L3 handoff.

As we have shown in [31] and Mishra et al. have shown in [24], the time needed by an MN to perform a L2 handoff is usually on the order of a few hundred milliseconds, thus causing a noticeable interruption in any ongoing real-time multimedia session. In either open 802.11 networks or 802.11 networks with WEP enabled, the discovery phase constitutes more than 90% of the total handoff time [24], [31]. In 802.11 networks with either WPA or 802.11i enabled, the handoff delay is dominated by the authentication process that is performed after associating to the new AP. In particular, no data can be exchanged amongst MNs before the authentication process completes successfully. In the most general case, both authentication delay and scanning delay are present. These two delays are additive, so, in order to achieve seamless real-time multimedia sessions, both delays have to be addressed and, if possible, removed.

When a L3 handoff occurs, an MN has to perform a normal L2 handoff and update its IP address. We can break the L3 handoff into two logical steps: subnet change detection and new IP address acquisition via DHCP [14]. Each of these steps introduces a significant delay.

In this paper we focus on the use of station cooperation to achieve seamless L2 and L3 handoffs. We refer to this specific use of cooperation as Cooperative Roaming (CR). The basic idea behind CR is that MNs subscribe to the same multicast group creating a new plane for exchanging information about the network and help each other in different tasks. For example, an MN can discover surrounding APs and subnets by just asking to other MNs for this information. Similarly, an MN can ask another MN to acquire a new IP address on its behalf so that the first MN can get an IP address for the new subnet while still in the old subnet.

For brevity and clarity's sake, in this paper we do not consider handoffs between different administrative domains and AAA-related issues although CR could be easily extended to support them. Incentives for cooperation are also not considered since they are a standard problem for any system using some form of cooperation (e.g., file sharing) and represent a separate research topic [12], [15]. Bandwidth and energy usage in CR, application layer mobility, and load balancing have been omitted due to space constraints but can be found in [16].

The rest of the paper is organized as follows. In Section II we show the state of the art for handoffs in wireless networks, in Section III we briefly describe how IPv4 and IPv6 multicast addressing is used in the present context, Section IV describes how, with cooperation, MNs can achieve seamless L2 and L3 handoffs. Section V introduces cooperation in the L2 authentication process to achieve seamless handoffs regardless of the particular authentication mechanism used. Section VI considers security and Section VII shows how streaming media can be supported in CR. In Section VIII we present our experiments and results and finally, Section IX concludes the paper.

1-4244-1588-8/07/$25.00 ©2007 IEEE

II. Related Work

The network community has done a lot of work on L2 and L3 handoffs in wireless networks. As of the writing of this paper, many standards such as IEEE 802.11f [2] and IEEE 802.11e [4] have been ratified and others, such as IEEE 802.11k [1], IEEE 802.11r [8] and IEEE 802.21 [6] are emerging, trying to solve some of the problems a wireless environment introduces. All of these approaches, however, introduce significant changes in the infrastructure and in the protocol. In particular, they have always been structured thinking of each MN as a stand alone entity.

802.11f focuses on ways in which APs can share information among each other with the definition of an Inter Access Point Protocol (IAPP). This can be particularly useful for the transfer of users' credentials during handoffs, for example.

The 802.11e protocol addresses QoS problems in wireless Local Area Networks (LANs). In particular, different traffic classes are defined with their own medium access parameters, giving real-time traffic higher priority in accessing the wireless medium than best-effort traffic.

The 802.11k protocol utilizes MNs to collect topology information and other useful statistics about the network and conveys it back to the APs. The APs then build a neighbor report containing all the information about the various APs and their neighbors. These reports are then sent to the MNs so that each MN can have information about its neighboring APs. The way these reports are built is not specified and often involves each MN having to scan different channels.

802.11r addresses the need for fast L2 roaming in 802.11 networks considering different authentication mechanisms as well as QoS. In 802.11r, fast Basic Service Set (BSS) transitions can only take place between APs in the same mobility domain. A mobility domain is a set of BSSs in the same Extended Service Set (ESS). Within a mobility domain, APs can exchange key material and context using encapsulation over the distribution system. 802.11r does not specify how an MN discovers the best candidate AP to connect to next. Scanning, neighbor reports and other means can be used. 802.11r supports pre-keying and resource reservation between MN and AP and it defines a key hierarchy to extend Pairwise Master Keys (PMKs) to multiple APs.

The IEEE 802.21 (Media Independent Handover) standard [6] introduces link-layer enhancements for performing intelligent handoffs between heterogeneous networks such as IEEE 802.11 and cellular, including both wireless and wired networks. The handoff process can be initiated by either the client or the network, and just like in IEEE 802.11k, MNs provide information about available networks and other network statistics to the infrastructure by scanning. The infrastructure then builds and stores information such as neighborhood cell lists and available services, thus helping in the optimum cell selection. Furthermore, new link-layer primitives are defined in order to provide applications with consistent information regardless of the access technology used by the MN.

In all these approaches, MNs always behave as stand alone entities often having to scan the medium before handoffs, that is, causing interruptions in any ongoing multimedia session. Furthermore, seamless handoffs with these approaches, when possible, require changes in the network and in the clients. CR is a client-only approach and can represent either an alternative or a complement to the current standards.

More recently, cooperative approaches have been proposed in the network community. Liu et al. [22] show how cooperation amongst MNs can be beneficial for all the MNs in the network in terms of bit-rate, coverage and throughput. Each MN builds a table in which possible helpers for that MN are listed. If an MN has a poor link with the AP and its bit-rate is low, it sends packets to the helper who relays them to the AP. The advantage in doing this is that the link from the MN to the helper and from the helper to the AP is a high bit-rate link. In this way the MN can use two high bit-rate links via the helper instead of the low bit-rate one directly to the AP.

Aside from cooperation approaches and standardization efforts in the IEEE 802.11 working groups, many other approaches have been proposed in order to achieve fast handoffs in wireless networks. However, most of these approaches, such as [18] and [26], require changes to either the infrastructure or the protocol or both. One good example of such a situation is Mobile IP (MIP). MIP has been standardized for many years now, however, it has never had a significant deployment, in part because of the considerable changes required in the infrastructure. Fast handoff approaches in the MIP context usually require additional network elements [32], [33] and/or changes to the protocol [27].

In [28] Ramani et al. suggest an algorithm called syncscan which does not require changes to either the protocol or the infrastructure. It does require, however, that all the APs in the network are synchronized and only accelerates unauthenticated L2 handoffs.

In this paper we propose a novel approach that works in an already deployed wireless environment, an environment with heterogeneous networks, where new network elements cannot necessarily be introduced in the infrastructure, where all the APs are not necessarily synchronized amongst themselves, where any kind of authentication mechanism can be used and where different subnets may be present.

We use a cooperative approach amongst MNs for spreading information regarding the network topology without any infrastructure support. Our approach requires changes only to the wireless card driver, DHCP client and authentication supplicant; no changes to the infrastructure or the protocol are required. This allows us to solve many of the problems associated with terminal mobility, regardless of the network the user moves to.

III. IP Multicast Addressing

CR works for both IPv4 and IPv6. In IPv4, we make extensive use of UDP-over-IP multicast packets. Different values for time-to-live (TTL) are used according to how far we want multicast packets to reach into the IP network. This also depends on the density of MNs supporting the protocol.

For example, if an MN does not receive any response after sending a request with a TTL value of 1 (same subnet), it will send the same request again but with a TTL value of 2 (next subnet) and so on. We must note, however, that the probability for an MN to find the information it needs becomes smaller as the search moves to more distant subnets. On the other hand, a small TTL can be used to limit the propagation of CR multicast frames in very congested environments.

In IPv6, we would use multicast scopes instead of IPv4 multicast. No significant changes would be required.

IV. Cooperative Roaming

In this section we show how MNs can cooperate with each other in order to achieve seamless L2 and L3 handoffs.

A. Overview

In [31] we have introduced a fast MAC layer handoff mechanism for achieving seamless L2 handoffs in environments such as hospitals, schools, campuses, enterprises, and other places where MNs always encounter the same APs. Each MN saves information regarding the surrounding APs in a cache. When an MN needs to perform a handoff and it has valid entries in its cache, it will directly use the information in the cache without scanning. If it does not have any valid information in its cache, the MN will use an optimized scanning procedure called *selective scanning* to discover new APs and build the cache. In the cache, APs are ordered according to their signal strength that was registered when the scanning was performed, that is, right before changing AP. APs with stronger signal strength appear first. As mentioned in Section I, in open networks the scanning process is responsible for more than 90% of the total handoff time.

The cache reduces the L2 handoff time to only a few milliseconds and cache misses due to errors in movement prediction introduce only a few milliseconds of additional delay [31]. Such an approach, however, works only in open networks or networks with WEP enabled. Other forms of authentication are not supported.

Earlier, we had extended [17] the mechanism introduced in [31] to support L3 handoffs. MNs also cache L3 information such as their own IP address, default router's IP address and subnet identifier. A subnet identifier uniquely identifies a subnet. By caching the subnet identifier, a subnet change is detected much faster and L3 handoffs are triggered every time the new AP and old AP have different subnet identifiers. Faster L3 handoffs can be achieved since IP address and default router for the next AP and subnet are already known and can be immediately used. The approach in [17] achieves seamless handoffs in open networks only, it utilizes the default router's IP address as subnet identifier and it uses a suboptimal algorithm to acquire L3 information.

Here, we consider the same caching mechanism used in [17]. In order to support multi-homed routers, however, we use the subnet address as subnet identifier. By knowing the subnet mask and the default router's IP address we can calculate the network address of a certain subnet. Fig. 1 shows the structure of the cache. Additional information such as last IP address used by the MN, lease expiration time and default router's IP address can be extracted from the DHCP client lease file, available in each MN.

	Current AP	Next Best AP	Second Best AP
BSSID	MAC A	MAC B	MAC C
Channel	6	11	1
Subnet ID	160.39.5.0	160.39.10.0	160.39.10.0

Fig. 1. Example of MN's cache structure

In CR, an MN needs to acquire information about the network if it does not have any valid information in the cache or if it does not have L3 information available for a particular subnet. In such a case, the MN asks other MNs for the information it needs so that the MN does not have to find out about neighboring APs by scanning. In order to share information, in CR, all MNs subscribe to the same multicast group. We call an MN that needs to acquire information about its neighboring APs and subnets a requesting MN (R-MN). By using CR, an R-MN can ask other MNs if they have such information by sending an INFOREQ multicast frame. The MNs that receive such a frame check if they have the information the R-MN needs and if so, they send an INFORESP multicast frame back to the R-MN containing the information the R-MN needs.

B. L2 Cooperation Protocol

In this section, we focus on the information exchange needed by a L2 handoff.

The information exchanged in the INFOREQ and INFORESP frames is a list of {BSSID, channel, subnet ID} entries, one for each AP in the MN's cache (see Fig. 1).

When an R-MN needs information about its neighboring APs and subnets, it sends an INFOREQ multicast frame. Such a frame contains the current content of the R-MN's cache, that is, all APs and subnets known to the R-MN. When an MN receives an INFOREQ frame, it checks if its own cache and the R-MN's cache have at least one AP in common. If the two caches have at least one AP in common and if the MN's cache has some APs that are not present in the R-MN's cache, the MN sends an INFORESP multicast frame containing the cache entries for the missing APs. MNs that have APs in common with the R-MN, have been in the same location of the R-MN and so have a higher probability of having the information the R-MN is looking for.

The MN sends the INFORESP frame after waiting for a random amount of time to be sure that no other MNs have already sent such information. In particular, the MN checks the information contained in INFORESP frames sent to the same R-MN by other MNs during the random waiting time. This prevents many MNs from sending the same information to the R-MN and all at the same time.

When an MN other than R-MN receives an INFORESP multicast frame, it performs two tasks. First, it checks if

someone is lying by providing the wrong information and if so, it tries to fix it (see Section VI-A); secondly, it records the cache information provided by such a frame in its cache even though the MN did not request such information. By collecting unsolicited information, each MN can build a bigger cache in less time and in a more efficient manner requiring fewer frame exchanges. This is very similar to what happens in software such as Bit-Torrent where the client downloads different parts of the file from different peers. Here, we collect different cache chunks from different MNs.

In order to improve efficiency and further minimize frame exchange, MNs can also decide to collect information contained in the INFOREQ frames.

C. L3 Cooperation Protocol

In a L3 handoff an MN has to detect a change in subnet and also has to acquire a new IP address. When a L2 handoff occurs, the MN compares the cached subnet identifiers for the old and new AP. If the two identifiers are different, then the subnet has changed. When a change in subnet is detected, the MN needs to acquire a new IP address for the new subnet. The new IP address is usually acquired by using the DHCP infrastructure. Unfortunately, the typical DHCP procedure can take up to one second [17].

CR can help MNs acquire a new IP address for the new subnet while still in the old subnet. When an R-MN needs to perform a L3 handoff, it needs to find out which other MNs in the new subnet can help. We call such MNs Assisting MNs (A-MNs). Once the R-MN knows the A-MNs for the new subnet, it asks one of them to acquire a new IP address on its behalf. At this point, the selected A-MN acquires the new IP address via DHCP and sends it to the R-MN which is then able to update its multimedia session before the actual L2 handoff and can start using the new IP address right after the L2 handoff, hence not incurring any additional delay.

We now show how A-MNs can be discovered and explain in detail how they can request an IP address on behalf of other MNs in a different subnet.

1) A-MNs Discovery: By using IP multicast, an MN can directly talk to different MNs in different subnets. In particular, the R-MN sends an AMN_DISCOVER multicast packet containing the new subnet ID. Other MNs receiving such a packet check the subnet ID to see if they are in the subnet specified in the AMN_DISCOVER. If so, they reply with an AMN_RESP unicast packet. This packet contains the A-MN's default router IP address, the A-MN's MAC and IP addresses. This information is then used by the R-MN to build a list of available A-MNs for that particular subnet.

Once the MN knows which A-MNs are available in the new subnet, it can cooperate with them in order to acquire the L3 information it needs (e.g., new IP address, router information), as described below.

2) Address Acquisition: When an R-MN needs to acquire a new IP address for a particular subnet, it sends a unicast IP_REQ packet to one of the available A-MNs for that subnet. Such packet contains the R-MN's MAC address. When an A-MN receives an IP_REQ packet, it extracts the R-MN's MAC address from the packet and starts the DHCP process by inserting the R-MN's MAC address in the CHaddr field of DHCP packets[1]. The A-MN will also have to set the broadcast bit in the DHCP packets in order for it to receive DHCP packets with a different MAC address other than its own in the CHaddr field. All of this allows the A-MN to acquire a new IP address on behalf of the R-MN. This procedure is completely transparent to the DHCP server. Once the DHCP process has been completed, the A-MN sends an IP_RESP multicast packet containing the default router's IP address for the new subnet, the R-MN's MAC address and the new IP address for the R-MN. The R-MN checks the MAC address in the IP_RESP packet to be sure that the packet is not for a different R-MN. Once it has verified that the IP_RESP is for itself, the R-MN saves the new IP address together with the new default router's IP address.

If the R-MN has more than one possible subnet to move to, it follows the same procedure for each subnet. In this way the R-MN builds a list of {router, new IP address} pairs, one pair for each one of the possible next subnets. After moving to the new subnet the R-MN renews the lease for the new IP address. The R-MN can start this process at any time before the L2 handoff, keeping in mind that the whole process might take one second or more to complete and that lease times of IP addresses are usually on the order of tens of minutes or more[2].

By reserving IP addresses before moving to the new subnet, we could waste IP addresses and exhaust the available IP pool. Usually, however, the lease time in a mobile environment is short enough to guarantee a sufficient re-use of IP addresses.

Acquiring an IP address from a different subnet other than the one the IP is for could also be achieved by introducing a new DHCP option. Using this option, the MN could ask the DHCP server for an IP address for a specific subnet. This would however, require changes to the DHCP protocol.

V. Cooperative Authentication

In this section we propose a cooperative approach for authentication in wireless networks. The proposed approach is independent of the particular authentication mechanism used. It can be used for VPN, IPsec, 802.1x or any other kind of authentication. We focus on the 802.1x framework used in Wi-Fi Protected Access (WPA) and IEEE 802.11i [3].

A. IEEE 802.1x Overview

The IEEE 802.1x standard defines a way to perform access control and authentication in IEEE 802 LANs and in particular in IEEE 802.11 wireless LANs using three main entities: supplicant, authenticator and authentication server[3]. The supplicant is the client that has to perform the authentication

[1] If supported, the client-ID field must be used instead [21].

[2] The DHCP client lease file can provide information on current lease times.

[3] The authentication server is not required in all authentication mechanisms.

in order to gain access to the network; the authenticator, among other things, relays packets between supplicant and authentication server; the authentication server, typically a RADIUS server [29], performs the authentication process with the supplicant by exchanging and validating the supplicant's credentials. The critical point, in terms of handoff time in the 802.1x architecture, is that during the authentication process the authenticator allows only EAP Over LAN (EAPOL) traffic to be exchanged with the supplicant. No other kind of traffic is allowed.

B. Cooperation in the Authentication Process

A well-known property of the wireless medium in IEEE 802.11 networks is that the medium is shared and therefore every MN can hear packets that other stations (STAs) send and receive. This is true when MN and STAs are connected to the same AP - that is, are on the same channel. In [22] Liu et al. make use of this particular characteristic and show how MNs can cooperate with each other by relaying each other's packets so to achieve the optimum bit-rate. In this section we show how a similar approach can be used for authentication purposes.

For simplicity, in the following discussion we suppose that one authenticator manages one whole subnet, so that authentication is required after each L3 handoff. In such a scenario and in this context, we also refer to a subnet as an Authentication Domain (AD). In general, an MN can share the information about ADs in the same way it shares information about subnets. In doing so, an MN knows whether the next AP belongs to the same AD of the current AP or not. In a L2 or L3 handoff we have an MN which performs handoff and authentication, a Correspondent Node (CN) which has an established multimedia session with the MN and a Relay Node (RN) which relays packets to and from the MN. Available RNs for a particular AD can be discovered following a similar procedure to the one described earlier for the discovery of A-MNs (see Section IV-C.1). The difference here is that RN and MN have to be connected to the same AP after the handoff. In this scenario, we assume that RNs are a subset of the available A-MNs. The basic idea is that while the MN is authenticating in the new AD, it can still communicate with the CN via the RN which relays packets to and from the MN (see Fig. 2). Let us look at this mechanism in more detail. Before the MN changes AD/AP, it selects an RN from the list of available RNs for the new AD/AP and sends a RELAY_REQ multicast frame to the multicast group. The RELAY_REQ frame contains the MN's MAC and IP addresses, the CN's IP address and the selected RN's MAC and IP addresses. The RELAY_REQ will be received by all the STAs subscribed to the multicast group and, in particular, it will be received by both the CN[4] and the RN. The RN will relay packets for the MN identified by the MAC address received in the RELAY_REQ frame. After performing the handoff, the MN needs to authenticate before it can resume any communication via the AP. However, because of the shared nature of the medium, the MN will start

[4]In congested environments, where smaller TTL values may be preferred, a separate unicast RELAY_REQ frame can be sent to the CN.

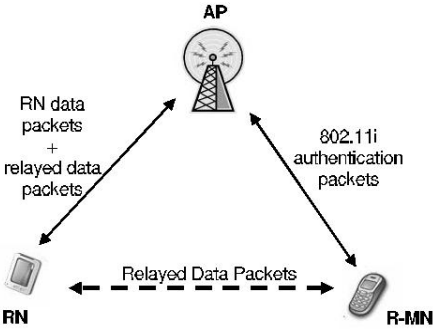

Fig. 2. Layer 2 handoff with authentication in CR

sending packets to the RN as if it was already authenticated. The authenticator will drop the packets, but the RN can hear the packets on the medium and relay them to the CN using its own encryption keys, that is, using its secure connection with the AP. The CN is aware of the relaying because of the RELAY_REQ, and so it will start sending packets for the MN to the RN as well. While the RN is relaying packets to and from the MN, the MN will perform its authentication via 802.1x or any other mechanism. Once the authentication process is over and the MN has access to the infrastructure, it can stop the relaying and resume normal communication via the AP. When this happens and the CN starts receiving packets from the MN via the AP, it will stop sending packets to the RN and will resume normal communication with the MN. The RN will detect that it does not need to relay any packet for the MN any longer and will return to normal operation.

In order for this relaying mechanism to work with WPA and 802.11i, MN and RN have to exchange unencrypted L2 data packets for the duration of the relay process. These packets are then encrypted by the RN by using its own encryption keys and are sent to the AP. By responding to an RN discovery, RNs implicitly agree to providing relay for such frames. Such an exchange of unencrypted L2 frames does not represent a security concern since packets can still be encrypted at higher layers and since the relaying happens for a very limited amount of time (see Section VI-B).

One last thing worth mentioning is that by using a relay, we remove the bridging delay in the L2 handoff [24], [31]. Usually, after an MN changes AP, the switch continues sending packets for the MN to the old AP until it updates the information regarding the new AP on its ports. The bridging delay is the amount of time needed by the switch to update this information on its ports. When we use a relay node in the new AP, this relay node is already registered to the correct port on the switch, therefore no update is required on the switch side and the MN can immediately receive packets via the RN.

VI. Security

Security is a major concern in wireless environments. In this section we address some of the problems encountered in a cooperative environment, focusing on CR.

A. Roaming Security Issues

In this particular context, a malicious user might try to propagate false information among the cooperating MNs. In particular, we have to worry about three main vulnerabilities:

1) A malicious user might want to re-direct STAs to fake APs where their traffic can be sniffed and private information can be compromised.
2) A malicious user might try to perform DoS attacks by re-directing STAs to far or non-existing APs. This would cause the STAs to fail the association to the next AP during the handoff process. The STA would then have to rely on the legacy scanning process to re-establish network connectivity.
3) At L3, a malicious user might behave as an A-MN and try to disrupt a STA' service by providing invalid IP addresses.

In general, we have to remember that the cooperative mechanism described here works on top of any other security mechanism that has been deployed in the wireless network (e.g., 802.11i, WPA). In order for a malicious user to send and receive packets from and to the multicast group, it has to have, first of all, access to the network and thus be authenticated. In such a scenario, a malicious user is a STA with legal access to the network. This means that MAC spoofing attacks are not possible as a change in MAC address would require a new authentication handshake with the network. This also means that once the malicious user has been identified, it can be isolated.

How can we attempt to isolate a malicious node? Since the INFORESP frame is multicast, each MN that has the same information than the one contained in such a frame, can check that the information in such a frame is correct and that no one is lying. If it finds out that the INFORESP frame contains the wrong information, it immediately sends an INFOALERT multicast frame. Such a frame contains the MAC address of the suspicious STA. This frame is also sent by an R-MN that has received a wrong IP address and contains the MAC address of the A-MN that provided that IP address. If *more than one* alert for the *same* suspicious node, is triggered by *different* nodes, the suspicious node is considered malicious and the information it provides is ignored. Let us look at this last point in more detail.

One single INFOALERT does not trigger anything. In order for an MN to be categorized as bad, there has to be a certain number of INFOALERT multicast frames sent by *different* nodes, all regarding the *same* suspicious MN. This certain number can be configured according to how paranoid someone is about security but, regardless, it has to be more than one. Let us assume this number to be five. If a node receives five INFOALERT multicast frames from five different nodes regarding the same MN, then it marks such an MN as bad. This mechanism could be compromised if either a malicious user can spoof five different MAC addresses (and this is not likely for the reasons we have explained earlier) or if there are five different malicious users that are correctly authenticated in the wireless network and that can coordinate their attacks. If this last situation occurs, then there are bigger problems in the network to worry about than handoff policies. Choosing the number of INFOALERT frames required to mark a node as malicious to be very large would have advantages and disadvantages. It would give more protection against the exploitation of this mechanism for DoS attacks as the number of malicious users trying to exploit INFOALERT frames would have to be high. On the other hand, it would also make the mechanism less sensitive to detect a malicious node as the number of INFOALERT frames required to mark the node as bad might never be reached or it might take too long to reach. So, there is clearly a trade-off.

Regardless, in either one of the three situations described at the beginning of this section, the MN targeted by the malicious user would be able to easily recover from an attack by using legacy mechanisms such as active scanning and DHCP address acquisition, typically used in non-cooperative environments.

B. Cooperative Authentication and Security

In order to improve security in the relay process, we introduce some countermeasures that nodes can use to prevent exploitation of the relay mechanism. The main concern in having a STA relay packets for an unauthenticated MN is that such an MN might try to repeatedly use the relay mechanism and never authenticate to the network. In order to prevent this, we introduce the following countermeasures:

1) Each RELAY_REQ frame allows an RN to relay packets for a limited amount of time. After this time has passed, the relaying stops. The relaying of packets is required only for the time needed by the MN to perform the normal authentication process.
2) An RN relays packets only for those nodes which have sent a RELAY_REQ packet to it while still connected to their previous AP.
3) RELAY_REQ packets are multicast. All the nodes in the multicast group can help in detecting bad behaviors such as one node repeatedly sending RELAY_REQ frames.

All of the above countermeasures work if we can be sure of the identity of a node and, in general, this is not always the case as malicious users can perform MAC spoofing attacks, for example. However, as we have explained in Section VI-A, MAC spoofing attacks are not possible in the present framework.

This said, we have to remember that before an RN can relay packets for an MN, it has to receive the proper RELAY_REQ packet from the MN. Such a packet has to be sent by the MN while still connected to the old AP. This means that the MN has to be authenticated with the previous AP in order to send such a packet. Furthermore, once the relaying timeout has expired, the RN will stop relaying packets for that MN. At this point, even if the MN can change its MAC address, it would not be able to send a new RELAY_REQ as it has to first authenticate again with the network (e.g., using 802.11i) and therefore no relaying would take place. In the special case in which the old AP belongs to an open

network[5], a malicious node could perform MAC spoofing and exploit the relay mechanism in order to have access to the secure network. In this case, securing the multicast group by performing authentication and encryption at the multicast group level could prevent this kind of attacks although it may require infrastructure support.

In conclusion, we can consider the three countermeasures introduced at the beginning of this section, to be more than adequate in avoiding exploitation of the relaying mechanism.

VII. Streaming Media Support

SIP can be used, among other things, to update new and ongoing media sessions. In particular, the IP address of one or more of the participants to the media session can be updated. In general, after an MN performs a L3 handoff, a media session update is required to inform the various parties about the MN's new IP address [30].

If the CN does not support cooperation, the relay mechanism as described in Section V-B does not work and the CN keeps sending packets to the MN's old IP address, being unaware of the relay process. This is the case for example, of an MN establishing a streaming video session with a stream media server. In this particular case, assuming that the media server supports SIP, a SIP session update is performed to inform the media server that the MN's IP address has changed. The MN sends a re-INVITE to the media server updating its IP address to the RN's IP address. In this way, the media server starts sending packets to the RN and relay can take place as described earlier. Once the relaying is over, if the MN's authentication was successful, the MN sends a second re-INVITE including its new IP address, otherwise, once the timeout for relaying expires, the relaying process stops and the RN terminates the media session with the media server.

VIII. Experiments

In the present section we describe implementation details and measurement results for CR.

A. Environment

All the experiments were conducted at Columbia University on the 7th floor of the Schapiro building. We used four IBM Thinkpad laptops: three IBM T42 laptops using Intel Centrino Mobile technology with a 1.7 GHz Pentium processor and 1GB RAM and one IBM laptop with an 800 MHz Pentium III processor and 384 MB RAM. Linux kernel version 2.4.20 was installed on all the laptops. All the laptops were equipped with a Linksys PCMCIA Prism2 wireless card. Two of them were used as wireless sniffers, one of them was used as roaming client and one was used as "helper" to the roaming client, that is, it replied to INFOREQ frames and behaved as an A-MN. For cooperative authentication the A-MN was also used as RN. Two Dell Dimension 2400 desktops were used, one as CN and the other as RADIUS server [29]. The APs used for the experiments were a Cisco AP1231G which is an enterprise AP and a Netgear WG602 which is a SOHO/home AP.

[5]Under normal conditions this is very unluckily but it might happen for handoffs between different administration domains, for example.

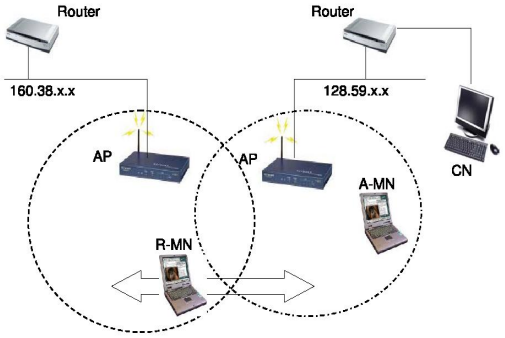

Fig. 3. L3 handoff environment

B. Implementation Details

In order to implement the cooperation protocol we modified the wireless card driver and the DHCP client. Furthermore, a cooperation manager was also created in order to preserve state information and coordinate wireless driver and DHCP client. For cooperative authentication, the WPA supplicant was also slightly modified to allow relay of unencrypted frames between MN and RN. The HostAP [23] wireless driver, an open-source WPA supplicant [19], and the ISC DHCP client [5] were chosen for the implementation. A UDP packet generator was also used to generate small packets with a packetization interval of 20 ms in order to simulate voice traffic. For the authentication measurements, we used FreeRADIUS [7] as RADIUS server.

C. Experimental Setup

For the experiments we used the Columbia University 802.11b wireless network which is organized as one single subnet. In order to test L3 handoff, we introduced another AP connected to a different subnet (Fig. 3). The two APs operated on two different non-overlapping channels.

The experiments were conducted by moving the roaming client between two APs belonging to different subnets, thus having the client perform L2 and L3 handoffs in either direction.

Packet exchanges and handoff events were recorded using the two wireless sniffers (kismet [20]), one per channel. The trace files generated by the wireless sniffer were later analyzed using Ethereal [13].

In the experimental set-up we do not consider a large[6] presence of other MNs under the same AP since air-link congestion is not relevant to the handoff measurements. Delays due to collisions, backoff, propagation delay and AP queuing delay are irrelevant since they usually are on the order of micro-seconds under normal conditions. However, even if we

[6]Other MNs were present in the Columbia wireless network during the experiments.

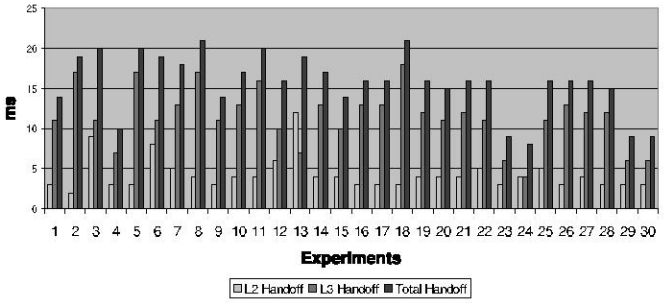

Fig. 4. Measured L2 and L3 handoff time with CR in an open network

consider these delays to be very high because of a high level of congestion, the MN should worry about not being able to make or continue a call as the AP has reached its maximum capacity. Handoff delay would, at this point, become a second order problem. Furthermore, in this last scenario, the MN should avoid to do handoff to a very congested AP in the first place as part of a good handoff policy.

Updating information at the Home Agent or SIP Registrar is trivial and does not have the same stringent delay requirements that mid-call mobility has, therefore it will not be considered.

D. Results

In this section we show the results obtained in our experiments. In Section VIII-D.1, we consider an open network with no authentication in order to show the gain of CR in an open network. In Section VIII-D.2, authentication is added and, in particular, we consider a wireless network with IEEE 802.11i enabled.

We define L2 handoff time as scanning time + open authentication and association time + IEEE 802.11i authentication time. The last contribution to the L2 handoff time is not present in open networks. Similarly, we define the L3 handoff time as subnet discovery time + IP address acquisition time.

In the following experiments we show the drastic improvement achieved by CR in terms of handoff time. At L2 such an improvement is possible because, as we have explained in Section IV-A, MNs build a cache of neighbor APs so that scanning for new APs is not required and the delay introduced by the scanning procedure during the L2 handoff is removed. Furthermore, by using relays (see Section V), an MN can send and receive data packets *during* the authentication process, thus eliminating the 802.11i authentication delay. At L3, MNs cache information about which AP belongs to which subnet, hence immediately detecting a change in subnet by comparing the subnet IDs of the old and new APs. This provides a way to detect a subnet change and at the same time makes the subnet discovery delay insignificant. Furthermore, with CR, the IP address acquisition delay is completely removed since each node can acquire a new IP address for the new subnet while still in the old subnet (see Section IV-C).

It is important to notice that in current networks[7] there is no standard way to detect a change in subnet in a timely manner[8]. Recently, DNA for IPv4 (DNAv4) [10] was standardized by the DHC working group within the IETF in order to detect a subnet change in IPv4 networks. This mechanism, however, works only for previously visited subnets for which the MN still has a valid IP address and can take up to hundreds of milliseconds to complete. Furthermore, if L2 authentication is used, a change in subnet can be detected only after the authentication process completes successfully. Because of this, in the handoff time measurements for the standard IEEE 802.11 handoff procedure, the delay introduced by subnet change discovery is not considered.

[7]Within the IETF, the DNA working group is standardizing the detection of network attachments for IPv6 networks only [25].

TABLE I

PERFORMANCE OVERVIEW FOR CR (AVERAGE)

IP_REQ - IP_RESP	867.0 ms
L2 handoff	4.2 ms
L3 handoff	11.4 ms
Total handoff	15.6 ms
Packet loss	1.3 packets

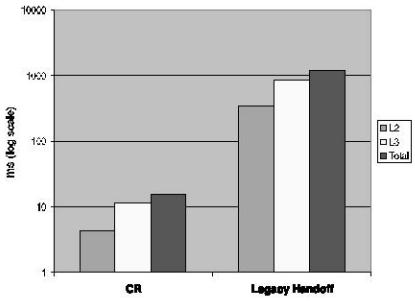

Fig. 5. Average handoff time for CR and IEEE 802.11b in an open network

To summarize, in theory by using CR the only contribution to the L2 handoff time is given by open authentication and association and there is no contribution to the L3 handoff time whatsoever, that is, the L3 handoff time is zero. In practice, this is not exactly true. Some other sources of delay have to be taken into consideration as we show in more detail in Section VIII-D.3.

1) L2 and L3 Roaming: We show the handoff time when an MN is performing a L2 and L3 handoff without any form of authentication, that is, the MN is moving in an open network. In such a scenario, before the L2 handoff occurs, the MN tries to build its L2 cache if it has not already done so. Furthermore, the MN also searches for any available A-MN that might help it in acquiring an IP address for the new subnet. The scenario is the same as the one depicted in Fig. 3.

Fig. 4 shows the handoff time when CR is used. In particular, we show the L2, L3 and total L2+L3 handoff times over 30 handoffs. As we can see, the total L2+L3 handoff time has a maximum value of 21 ms in experiment 18. Also, we can see how, even though the L3 handoff time is higher on average than the corresponding L2 handoff time, there are situations where these two become comparable. For example, we can see in experiment 24 how the L2 and L3 handoff times are

[8]Router advertisements are typically broadcast only every few minutes.

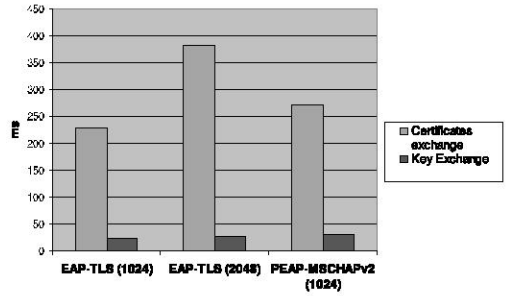

Fig. 6. Authentication delay in IEEE 802.11i

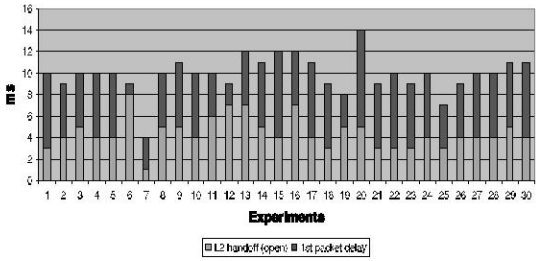

Fig. 8. CR L2 handoff time in IEEE 802.11i networks

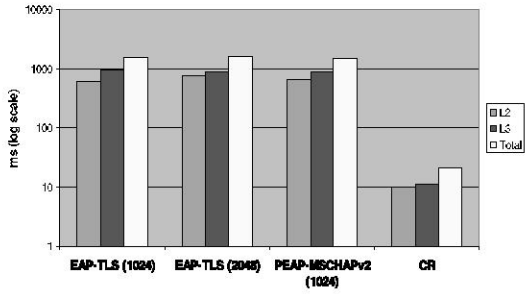

Fig. 7. Handoff time in IEEE 802.11i networks

equal and in experiment 13 how the L2 handoff time exceeds the corresponding L3 handoff time. The main causes for this variance will be presented in Section VIII-D.3.

Fig. 4 and Table I show how, on average, with CR the total L2+L3 handoff time is less than 16 ms, which is less than half of the 50 ms requirement for assuring a seamless handoff when real-time traffic is present.

Table I shows the average values of IP address acquisition time, handoff time, and packet loss during the handoff process. The time between IP_REQ and IP_RESP is the time needed by the A-MN to acquire a new IP address for the R-MN. This time can give a good approximation of the L3 handoff time that we would have without cooperation. As we can see, with cooperation we reduce the L3 handoff time to about 1.5% of what we would have without cooperation. Table I also shows that the packet loss experienced during a L2+L3 handoff is negligible when using CR.

Fig. 5 shows the average delay over 30 handoffs of L2, L3 and L2+L3 handoff times for CR and for the legacy 802.11 handoff mechanism. The total L2+L3 handoff time is less than 16 ms for CR while it is about 1210 ms for the legacy 802.11 handoff mechanism. CR has reduced the total handoff time to 1.3% of the handoff time introduced by the standard 802.11 handoff procedure.

2) L2 and L3 Roaming with Authentication: Here we show the handoff time when IEEE 802.11i is used together with EAP-TLS and PEAP/MSCHAPv2.

Fig. 6 shows the average over 30 handoffs of the delay introduced in a L2 handoff by the certificate/credentials exchange and the session key exchange. Different key lengths are also considered for the generation of the certificates[9]. As expected, the exchange of certificates takes most of the time. This is the reason why mechanisms such as fast-reconnect [9], [11] improve L2 handoff times considerably, although still on the order of hundreds of milliseconds.

Generally speaking, any authentication mechanism can be used together with CR. Fig. 7 shows the average over 35 handoffs of the total L2, L3 and L2+L3 handoff times. In particular, we show the handoff time for EAP-TLS with 1024 and 2048 bits key, PEAP/MSCHAPv2 with 1024 bits key and CR. The average L2+L3 handoff times are respectively 1580 ms, 1669 ms, 1531 ms and 21 ms. By using CR, we achieve a drastic improvement in the total handoff time. As we can see, CR reduces the handoff time to 1.4% or less of the handoff time introduced by the standard 802.11 mechanism. This significant improvement is possible because at L2 with CR we bypass the whole authentication handshake by relaying packets. At L3 we are able to detect a change in subnet in a timely manner and acquire a new IP address for the new subnet while still in the old subnet.

Fig. 8 shows in more detail the two main contributions to the L2 handoff time when a relay is used. We can see that, on average, the time needed for the first data packet to be transmitted after the handoff takes more than half of the total L2 handoff time. Here, with data packet we are referring to a packet sent by our UDP packet generator. By analyzing the wireless traces collected in our experiments, we found that the first data packet after the handoff is not transmitted immediately after the L2 handoff completes because the wireless driver needs to start the handshake for the authentication process. This means that the driver already has a few packets in the transmission queue that are waiting to be transmitted when our data packet enters the transmission queue. This, however, concerns only the first packet to be transmitted after the L2 handoff completes successfully. All subsequent data packets will not encounter any additional delay due to relay.

3) Measurement Variance: We have encountered a high variance in the L2 handoff time. In particular, most of the delay is between the authentication request and authentication response, before the association request. Within all the measurements taken, such behaviour appeared to be particularly prominent when moving from the Columbia AP to the Netgear

[9]The length of certificates affects the handoff time much more than the length of session keys.

AP. This behavior, together with the results shown by Mishra et al. in [24], have lead us to the conclusion that such a variance is caused by the cheap hardware used in the low-end Netgear AP.

At L3, ideally, the handoff time should be zero as we acquire all the required L3 information while still in the old subnet. The L3 handoff time shown in Fig. 4 can be roughly divided in two main components: *signaling delay* and *polling delay*. The signaling delay is due to various signaling messages exchanged among the different entities involved in setting up the new L3 information in the kernel (wireless driver and DHCP client); the polling delay is introduced by the polling of variables in between received-signal-strength samples[10], done in order to start the L3 handoff process in a timely manner with respect to the L2 handoff process.

These two delays are both implementation dependent and can be reduced by further optimizing the implementation.

IX. Conclusions and Future Work

In this paper we have defined the Cooperative Roaming protocol. Such a protocol allows MNs to perform L2 and L3 handoffs seamlessly, with an average total L2+L3 handoff time of about 16 ms in an open network and of about 21 ms in an IEEE 802.11i network without requiring any changes to either the protocol or the infrastructure. Each of these values is less than half of the 50 ms requirement for real-time applications such as VoIP to achieve seamless handoffs. Furthermore, we are able to provide such a fast handoff regardless of the particular authentication mechanisms used while still preserving security and privacy.

MN cooperating has many advantages and does not introduce any significant disadvantage as in the worst case scenario MNs can rely on the standard IEEE 802.11 mechanisms achieving performances similar to a scenario with no cooperation.

As future work, we will look in more detail at application layer mobility, load balancing and call admission control. We will investigate the possibility of having some network elements such as APs support A-MN and RN functionalities; this would be useful in scenarios where only few MNs support CR. Finally, we will look at how IEEE 802.21 [6] could integrate and extend CR.

X. Acknowledgements

This work was supported by Firsthand Technologies and by the National Science Foundation under grants CNS 0335244 and CNS 0202063.

References

[1] IEEE Draft Amendament to ieee std 802.11, 1999 edition. Specification for Radio Resource Measurement, July 2003.

[2] IEEE Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, 2003.

[3] Amendment to ieee std 802.11, 1999 edition (reaff 2003). medium access control (mac) security enhancements, 2004.

[4] Amendment to ieee std 802.11, 1999 edition. Medium Access Control (MAC) Quality of Service Enhancements, 2005.

[5] DHCP Client version 3, 2005.

[6] IEEE Draft STANDARD FOR Local and Metropolitan Area Networks: Media Independent Handover Services, 2005.

[7] FreeRADIUS, 2006.

[8] IEEE Draft Amendament to ieee std 802.11, 1999 edition. Fast BSS Transition, March 2006.

[9] B. Adoba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748, June 2004.

[10] B. Adoba, J. Carlson, and S. Cheshire. Detecting Network Attachment in IPv4 (DNAv4). RFC 4436, March 2006.

[11] B. Adoba, D. Simon, P. Eronen, and H. Levkowetz. Extensible Authentication Protocol (EAP) Key Management Framework. IETF Draft (work in progress), May 2006.

[12] C. Buragohain, D. Agrawal, and S. Suri. A game theoretic framework for incentives in p2p systems. In *Proceedings of P2P '03*, page 48, Washington, DC, USA, 2003. IEEE Computer Society.

[13] G. Combs et al. Ethereal: Network Protocol Analyzer, 2005.

[14] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997.

[15] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *Proceedings of EC '04*, pages 102–111, New York, NY, USA, 2004. ACM Press.

[16] A. G. Forte and H. Schulzrinne. Cooperation Between Stations in Wireless Networks. Technical Report cucs-044-06, Columbia University, December 2006.

[17] A. G. Forte, S. Shin, and H. Schulzrinne. Improving L3 handoff delay in IEEE 802.11 wireless networks. In *Proceedings of WICON '06*. ACM Press, August 2006.

[18] R. Hsieh, Z. G. Zhou, and A. Seneviratne. S-MIP: A seamless handoff architecture for Mobile IP. In *Proceedings of the IEEE INFOCOM conference*, 2003.

[19] j. Malinen. Linux WPA/WPA2/IEEE 802.1x supplicant, 2005.

[20] M. Kershaw et al. Kismet: 802.11 Layer 2 Wireless Network Sniffer, 2005.

[21] T. Lemon and B. Sommerfield. Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4). RFC 4361, February 2006.

[22] P. Liu, Z. Tao, and S. Panwar. A cooperative MAC protocol for wireless local area networks. In *Proceedings of ICC '05*, pages 2962–2968. IEEE Computer Society, May 2005.

[23] J. Malinen. Hostap: Wireless Driver for Intersil Prism2/2.5/3, 2005.

[24] A. Mishra, M. Shin, and W. Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *SIGCOMM Comput. Commun. Rev.*, 33(2):93–102, 2003.

[25] S. Narayanan. Detecting Network Attachment in IPv6 Networks (DNAv6). IETF Draft (work in progress), October 2006.

[26] D. V. Ote, S. Paskalis, A. Kaloxylos, and L. Merakos. A SIP-based method for intra-domain handoffs. In *Proceedings of VTC-Fall '03*, pages 2068–2072. IEEE Computer Society, 2003.

[27] C. E. Perkins and D. B. Johnson. Route Optimization for Mobile IP. *Cluster Computing*, 1(2):161–176, 1998.

[28] I. Ramani and S. Savage. Syncscan: Practical fast handoff for 802.11 infrastructure networks. In *Proceedings of the IEEE INFOCOM Conference*, 2005.

[29] C. Rigney, A. C. Rubens, W. A. Simpson, and S. Willens. Remote Authentication Dial In User Service (RADIUS). RFC 2865, June 2000.

[30] H. Schulzrinne and E. Wedlund. Application-layer mobility using SIP. *SIGMOBILE Mob. Comput. Commun. Rev.*, 4(3):47–57, 2000.

[31] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne. Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs. In *Proceedings of MobiWac '04*, pages 19–26, New York, NY, USA, 2004. ACM Press.

[32] C. H. Wu, A. T. Cheng, S. T. Lee, J. M. Ho, and D. T. Lee. Bi-directional Route Optimization in Mobile IP Over Wireless LAN. In *VTC-Fall 2002: Proceedings of the 56th Vehicular Technology Conference, 2002*, pages 1168–1172. IEEE Computer Society, 2002.

[33] H. Yokota, A. Idoue, T. Hasegawa, and T. Kato. Link layer assisted mobile IP fast handoff method over wireless LAN networks. In *Proceedings of MobiCom '02*, pages 131–139, New York, NY, USA, 2002. ACM Press.

[10] Received-signal-strength values are measured by the wireless card driver.