A Network Architecture for Large Mobile Robotics Environments

Daniel Moraes, Paulo Coelho, Eleri Cardozo
Thienne Johnson, Fernanda Atizani
School of Electrical and Computer Enginneering
University of Campinas, Brazil
Email: dmoraes@dca.fee.unicamp.br

Eliane Guimarães
Information Technology Center Renato Archer
Robotics and Computer Vision Division
Campinas, Brazil
Email: eliane.guimaraes@cti.gov.br

***Abstract*—Mobile robotics environments must adopt networking solutions that provide secure and reliable communications for the mobile robots across wide areas such as hospitals, factories, farms, etc. This paper proposes a network architecture for large mobile robotic environments built above the existing networking infrastructures. The architecture builds an overlay network above the already deployed network. The overlay network must fulfill the requirements demanded by mobile robotic applications, mainly, communication continuity during handover, security, and quality of service. A prototype of this architecture was implemented and evaluated in a mobile robotic environment composed of Pioneer P3-DX mobile robots accessed through the Internet. Results from simulation show that the architecture scales well in larger networking scenarios.**

I. Introduction

As mobile robots become more and more integrated on internal and external environments, networking solutions for supporting control and communication with the mobile robots are of major concern. Buildings, factories, and hospitals, for instance, already have networking infrastructures deployed. Usually, these networks follow the common architecture where a backbone integrates a set of departmental subnetworks. Departmental subnetworks usually incorporate wireless access points for mobile clients.

Handover is the process by which a mobile node changes its network point of attachment in order to improve the signal-to-noise ratio of the wireless link. A layer 2 (L2) handover rebuilds the wireless link established with the previous access point to a closer access point. During this process the mobile node remains unreachable. The handover overhead (time to resume communication) varies from tens of milliseconds to seconds [1] depending on the wireless hardware and device drivers installed on the mobile nodes and access points. After the L2 handover completes, the layer 3 (L3) handover starts. L3 handover rebuilds the L3 parameters such as IP address, network prefix, and default router. If the L3 parameters of the mobile node change (i.e., the new access point belongs to a different subnetwork) the transport connections established on the previous subnetwork are broken. For mobile robots that commonly act as servers, changing L3 parameters is obviously unacceptable.

If the organization decides to deploy mobile robotics applications without constraining the mobile robots inside a subnetwork, a network architecture must be designed and deployed. This architecture must offer communication continuity during handover, security, and quality of service to the mobile robots. An expensive solution would deploy a separated network for the mobile robots. A more economical approach is to build a logical (overlay) network above the existing organizational network able to fulfill the mobile robotics requirements. Figure 1 illustrates this approach. The overlay network may need some low cost devices such as access points and PC-based servers that are connected to the existing network as any other devices. As such, the architecture makes use of the existing expensive devices such as routers, switches, and cabling without demanding any updating or reconfiguration on these devices.

The overlay network gives the mobile robots an homogeneous networking environment where parameters such as ESSID (Extended Service Set ID), network prefix, default router, and security keys remain unchanged. As a result, mobile robots can roam among access points preserving their network connections.

This paper is organized as follows. Section II presents the proposed network architecture for network robotics. Section III presents some implementation details of this architecture. Section IV describes an application on network robotics running above the architecture. Section V compares this work with some related ones. Finally, Section VI presents the concluding remarks.

II. A Network Architecture for Mobility

A network architecture addressing mobility must provide a set of mobility-related functions. The most important functions are:

- L3 addressing functions: functions that assign L3 parameters to the mobile nodes while they roam among access points.
- Location functions: functions that keep track of the mobile nodes and signal the network when they change their points of attachment.
- Mobile routing functions: functions that act on the network in order to deliver packets to the mobile nodes' actual locations.

ROBOCOMM 2009, 31st Mar – 2nd Apr 2009, Odense, Denmark.
 ISBN 978-963-9799-51-6
DOI 10.4108/ICST.ROBOCOMM2009.5831

- Forwarding functions: functions that allow special packet forwarding decisions such as packet filtering, address translation, tunneling, and proxying.
- Management functions: functions that allow routers to be configured to perform the mobility-related functions (e.g., the establishment of tunnels).
- Enhancing functions: additional functions that provide fast handover, quality of service, security, reliability, etc.

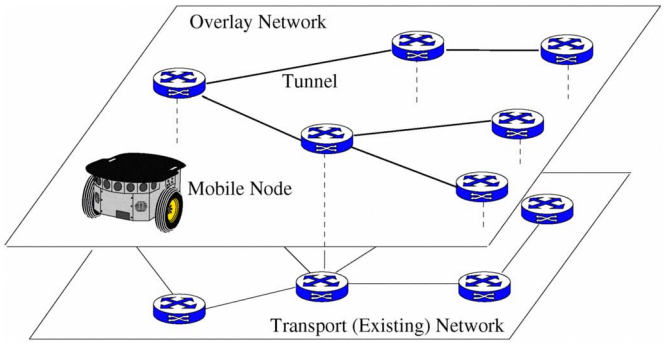

Fig. 1. An homogeneous overlay network built above an heterogeneous transport network.

The Mobility Plane Architecture (MPA) [2] is a network architecture for supporting mobility in IPv4, IPv6 and MPLS (Multiprotocol Label Switching) transport networks. In MPA, mobile routing functions are performed by the Resource Reservation Protocol (RSVP) with two extensions: Traffic Engineering extensions that allow constraint-based routing of tunnels (RSVP-TE) [3]; and point-to-multipoint (P2MP) extensions that allow the signaling of P2MP tunnels [4].

RSVP-TE nodes form a logical (overlay) network above the transport network. The overlay network is composed of a set of P2MP tunnels rooted on the ingress routers. These tree-structured tunnels are responsible for the distribution of traffic to the mobile nodes. The reverse traffic (generated by the mobile nodes) follows the paths given by the regular IP routing on the transport network.

MPA employs access subnetworks with the same network address prefix, usually in the private range. This means that the mobile nodes keep their IP addresses during handover. The wireless network is based on IEEE 802.11b/g configured with WPA2 (Wireless Protected Access 2) security employing PSK (Pre-Shared Key) or RADIUS (Remote Authentication Dial In User Service) authentication.

In MPA, the L3 addressing functions rely on the DHCP (Dynamic Host Configuration Protocol). On IPv4 networks, DHCPv4 supplies network prefix, IP address, and default router. On IPv6 networks, DHCPv6 does not supply network prefix and default router, a task left to the Neighbor Discovery (ND) protocol. ND causes long delays during handover as the mobile node must wait for ND Router Advertisement (RA) messages. There are two possible solutions for this shortcoming. The access router may send RA at a higher rate, or RA messages may be synchronized with node attachments. MPA implementation for IPv6 uses the latter approach.

MPA implements a location function on the access points via L2 triggers. An L2 trigger is a notification generated when a mobile node attaches to or detaches from an access point. The trigger is targeted to the RSVP-TE daemon running on the access router where the access point is connected to. Upon receiving the notification, the RSVP-TE daemon starts the mobile routing function.

Mobile routing in MPA employs an opaque RSVP-TE object carried on RESV (reservation) messages, the location object. In RSVP-TE, RESV messages are employed to refresh the soft state tunnels signaled with PATH messages. RESV messages follow the tunnel bottom-up, being processed by the routers along the path. The proposed networking architecture employs RESV messages for both refreshing the tunnels and signaling mobile node attachments. A RESV message carrying a location object is generated as soon as the RSVP-TE daemon receives an L2 trigger notification. The location object carries the identification (MAC address) and IP address of the mobile node.

Figure 2 shows an overlay network composed of a P2MP tunnel rooted on router R1. A mobile node is attached initially on an access point (not shown) connected to router R4. When the mobile node moves to a link served by R5, the route related to this node on the mobile routing table at R2 must be updated with a different tunnel segment (in this case, from segment C to D). If the mobile node roams to a link served by R7, the mobile routing table at R1 and R3 must be updated. Updatings on routing tables are performed as soon as the RESV message indicating the new point of attachment is processed by the routers.

When a mobile node disconnects, the disassociated access point generates an L2 trigger identifying the disconnection. Upon receiving this notification, the RSVP-TE daemon at the access router generates a RESV message with a location object, but with a flag indicating disconnection. The processing of this message causes the removal of routes installed for this mobile node.

MPA supports micro-mobility, that is, mobility inside a potentially large domain. For mobility across domains (macro-mobility), MIPv6 [5] can be employed as described in [6].

A. Quality of Service Issues

Quality of service (QoS) consists of a set of control and management functions that allows the network to guarantee some end-to-end metrics such as delay and jitter for the traffic flows generated and consumed by the applications. QoS assumes resource reservation for a particular flow, an idea in line with the Integrated Service Architecture (IntServ). IntServ relies on RSVP for signaling resource reservation along a flow path. As we employ RSVP-TE in our architecture, resource reservation can be employed. Unfortunately, the management of resources in a per-flow basis is unfeasible for small-sized routers due to the processing power it demands. In addition, QoS demands that all routers on the transport network support RSVP-TE.

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5821
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5821

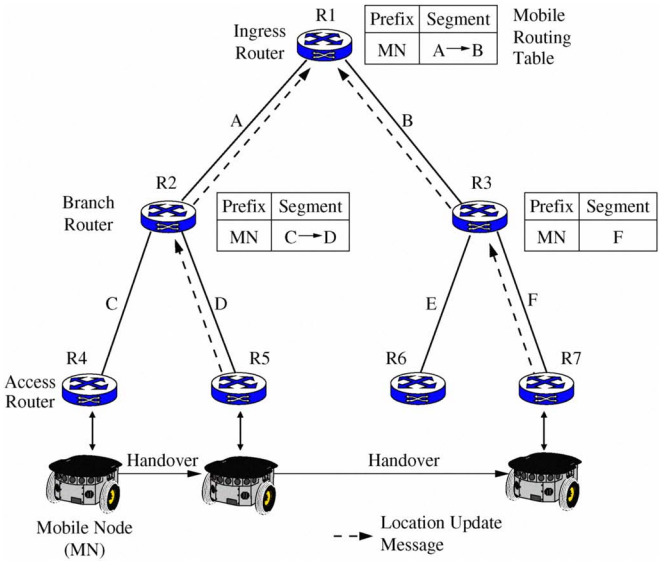

Fig. 2. Mobile routing process in MPA. Dashed arrows indicate the path of RSVP-TE RESV messages carrying location objects.

A simpler approach to QoS is traffic priorization where the network establishes relative priorities for the flows, without reserving resources for each particular flow. For example, telemetry flows have higher priority than audio and video flows. Classes of service (CoS) as defined by the Differentiated Services (DiffServ) architecture may be employed. DiffServ relies on packet markings and packet filters for traffic priorization on the routers along the flow path.

The current version of MPA employs DiffServ simply by configuring the routers to honor DiffServ markings. Packet filters responsible for marking packets are installed on ingress and access routers via network management configuration. For example, we can configure a robot's telemetry port to receive EF (Expedited Forward) service. DiffServ demands that all routers on the transport network honor DiffServ markings.

B. Security Issues

In MPA the access points can be configured to authenticate mobile nodes based on WPA2 employing pre-shared keys (PSK) or RADIUS. PSK is easy to configure but is not as secure as RADIUS-based authentication. RADIUS authentication can be strengthened by using certificates installed on the mobile nodes.

As RADIUS transactions take long time (500ms in our testbed network), RADIUS-based authentication increases considerably the handover overhead. In order to speed up RADIUS-based authentication, a cache mechanism can be employed. In this mechanism, once a mobile node completes successfully a RADIUS transaction, the access point stores the Pairwise Master Key (PMK) supplied by the RADIUS server in the cache. When the mobile node connects to a new access point, the access point queries the cache (using the mobile node's MAC address as a search key) in order to recover the PMK assigned to the node. If found, the access point accepts the mobile node without the need of a RADIUS transaction. In this case, the PMK found on cache is used to secure the communication between the mobile node and the access point. We implemented a cache mechanism in order to speed up the RADIUS-based authentication. The details about this mechanism are outside the scope of this paper.

III. Implementation Details

MPA was implemented for Linux routers with extensions for IP/IP (IP over IP) tunnels and MPLS tunnels. The IP/IP extension was ported to the MikroTik [7] RoutingBOARD 133 running OpenWRT [8], a Linux-based operating system for network appliances. The MikroTik boxes act as both router and access point. The MPA implementation consists of a RSVP-TE daemon with P2MP extensions written in C and a management front-end written in Java. The RSVP-TE daemon was cross-compiled for the MikroTik boxes. The management front-end runs on a PC-based management station.

The management front-end interacts with the RSVP-TE daemon in order to manage P2MP tunnels. This interaction is based on the exchanging of XML messages over TCP (Transfer Control Protocol). The tool offers a menu bar with options for loading the physical network topology; discovering the logical topology of the network by polling the MPA routers for established tunnels; and managing (create, destroy, reroute, and monitor) P2MP tunnels. The front-end can run as a desktop application or as a Java applet on web browsers.

A. Results from Testbed

The testbed network consists of three MikroTik RoutingBOARD 133, one acting as ingress router and the remaining as egress routers. These routers were linked through three plain Linux routers emulating an existing networking infrastructure. RSVP-TE was installed on the MikroTik routers. The Linux routers were configured with static routes. The egress routers act also as access points configured with WPA-PSK.

The network services consist of a DHCP server from the Internet Software Consortium (ISC), a RADIUS server from the FreeRADIUS project, and an HTTP (Hypertext Transfer Protocol) server from the Apache Software Foundation. These servers run on a DELL PowerEdge 1900 server machine connected on the MikroTik ingress router. Figure 3 shows the physical and logical (overlay) topology of the testbed network.

The mobile nodes are Pioneer P3-DX mobile robots. Two robots have on-board processors running Linux (Debian and Xubuntu distributions). The robot without internal processor is fitted with a notebook DELL D430 running Windows Vista.

The egress routers were installed about 50m apart in order to force the mobile nodes to perform a handover when moving between them. For performing handover without user intervention, Linux relies on WPA-Supplicant while Windows has a built-in facility known as Wireless Zero Config (WZC, renamed to WLAN AutoConfig on Vista). These facilities keep scanning the air and select to the access point with better signal-to-noise ratio. An hysteresis mechanism prevents frequent switching among access points.

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5821
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5821

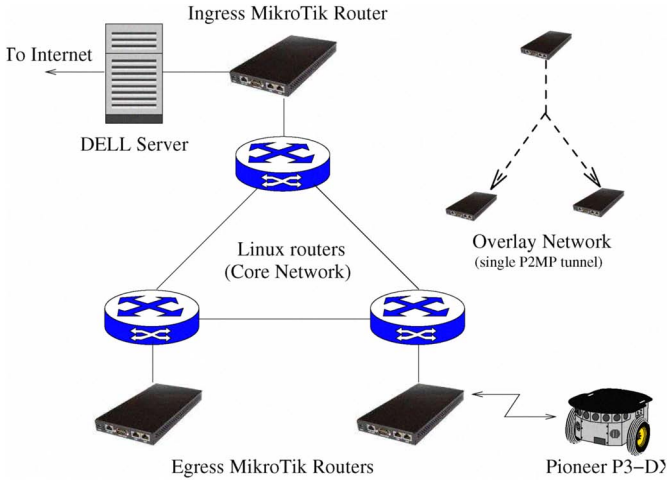

Fig. 3. Topology of the testbed network. MPA's P2MP tunnels are established among the MikroTik routers through the "core" network.

In order to estimate the handover overhead a constant bit rate traffic generator/receiver tool was installed on the robots and on the DELL server. The rate was set to 20 packets/s. The tool prints the number of packets lost during handover. The handover overhead is computed by multiplying the number of packets lost by 50ms. During this time, the mobile node becomes unreachable.

The Windows Vista produced a handover overhead between 300 and 500ms for download traffic and between 200 to 400ms for upload traffic. The difference between download and upload overheads (about 100ms) can be indebted to MPA signaling and route redirection for download traffic (upload traffic follows fixed routes established by regular IP routing).

WPA-Supplicant performed much worse, producing handover overheads between 5 and 7 seconds. This high overhead can be reduced to 1 second if we force a handover manually using a WPA client tool. We can reach the Windows good handover performance on Linux by disabling WPA security and employing a shell script based on Linux Wireless Tools to perform automatic handover.

With the results obtained from the testbed network, it is imperative to upgrade WPA-Supplicant or replace it to more efficient handover assisting tools. WPA-Supplicant was designed to mobile nodes that stay connected for long periods of time on the same access point. Nodes with high mobility such as mobile robots must rely on better handover assisting tools at least as efficient as the one provided by the Windows operating system.

B. Results from Simulation

A simulation model from MPA described in [9] was tuned with the parameters obtained from the MikroTik-based testbed network with mobile nodes running Windows Vista. The objective of the simulation was to obtain the average throughput as the handover rate completed successfully as a function of two parameters. The first parameter is the size of the network given by number of routers (hops) between the ingress and egress routers. The second parameter is arrival rate of mobile nodes (number of handovers requested per second), up to 2 handovers/s. Figure 4 shows that the proposed architecture scales appropriately for larger network topologies and mobile node dynamics (arrival rate) than employed on our testbed network, since no request has being rejected by the network.

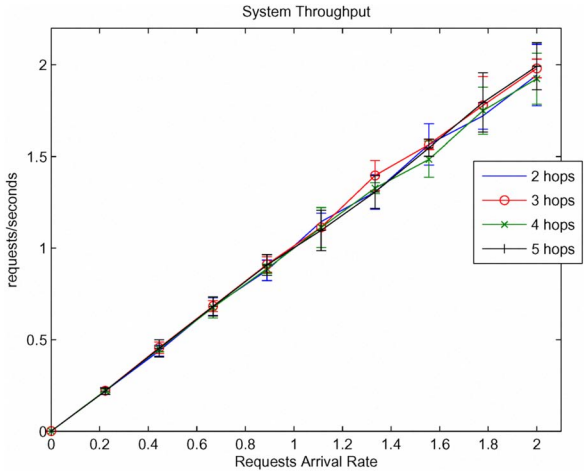

Fig. 4. Throughput of MPA running on MikroTik boxes as a function of network size (hops) and node arrival rate.

Another simulation result is the overhead imposed by MPA as a function of the network size (number of hops between the ingress and the egress routers) and node arrival rate. Figure 5 shows the MPA response time. The value around 100ms per hop for the response time obtained from simulation is coherent with the response times obtained from the testbed network. The simulation results show that the proposed mobility architecture scales fairly for larger network environments, since the delays are not strongly affected by the increasing of requests arrival rate. Delays of 300ms are considered an upper limit for high quality interactive applications.

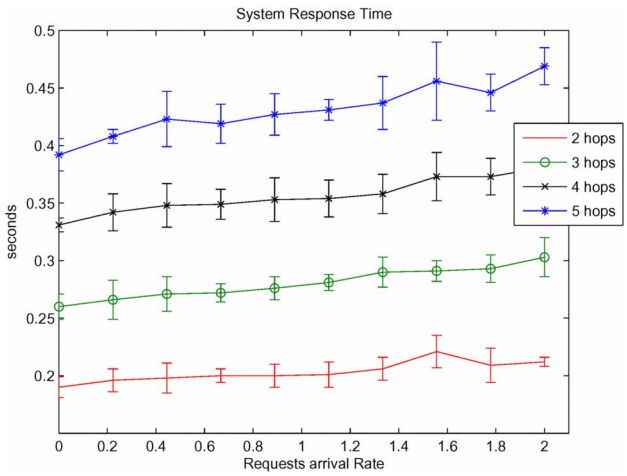

Fig. 5. Delay imposed by MPA as a function of network size (hops) and node arrival rate.

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5821
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5821

IV. A Network Robotics Application

A mobile robotics WebLab (REALab) developed by the authors and reported in [10] was employed to evaluate from the user's standpoint the proposed network architecture. WebLabs allow laboratorial equipments be operated in real time from remote sites through the public Internet or private high speed networks. The WebLab operates four Pioneer P3-DX with different configurations, high quality network cameras, and a set of servers. Mobile robotic experiments can run on the server or on the user's computer.

We choose the teleoperation experiment to assess how the user perceives the handover overheads imposed by our network architecture. From a remote operating station equiped with a joystick, the operator conducts a mobile robot inside a room in order to inspect its interior.

The operator receives real time video from the on-board camera and a 3D image constructed from the laser rangefinder. Video frame rate is set to 5 frames/s and laser telemetry is acquired in intervals of 500ms. As the robot approaches to the room it changes access points as the access point inside the room provides a much better signal to noise ratio.

For Linux-based mobile robots with WPA-Supplicant, the high handover overhead compromises teleoperation as the user looses control for 6 seconds average. The robot maintains the velocity set by the joystick before the handover took place. Considering a velocity of 250mm/s, the robot moves 1.5m without any control. The images from the on-board camera and from the laser rangefinder freeze for 6 seconds, a period of time long enough to be perceived by the operator.

For a handover performance similar to Windows Vista (overhead averaging 400ms), the distance the robot moves without operator control drops to only 10cm. The video flow looses just 2 frames, a discontinuity not perceived by the human operator.

Teleoperation is more tolerant to handover overheads as an human operator is on the control loop. For autonomous navigation algorithms, mainly those running on remote servers, high handover overheads can lead to instabilities and inefficiencies. Application demanding precise sensing and actuaction (e.g., navigation on narrow spaces) or demanding high telemetry transferring rates (e.g., vision-based navigation at high speeds) will degrade in presence of handover. In the first case, communication disruptions during handover will delay sensing and actuaction. In the second case, disruptions cause TCP connections operating at high data rates to drop their rates significantly (due to the TCP's slow start algorithm). In both cases, the communication disruption during handover can result in shocks against obstacles, unprecise trajectories, and speed reduction in mobile robot autonomous navigation.

In order to illustrate the influence of handover on autonomous navigation we run the Potential Fields [11] experiment offered by the REALab WebLab. The Potential Fields algorithm runs on the user's computer and controls the robot over the network. We forced handover in the middle of the trajectory by killing the process responsible for managing the aerial interface on one MicroTik access router. The experiment consists of setting a goal to a point 4.5 meters from the robot's initial position. Between the robot and the goal there is an obstacle. The experiment consists in executing the Potential Fields algorithm two times, one to move to the goal and one to return to the origin. We performed the experiment four times, with two speeds and with and without handover. The robot runs Linux with WPA-Suppliant. The trajectories with low speed (100mm/s maximum) with and without handover are shown in Figure 6. The trajectories are near the same as at low speeds the robot runs a short distance without control even with long handover delays.

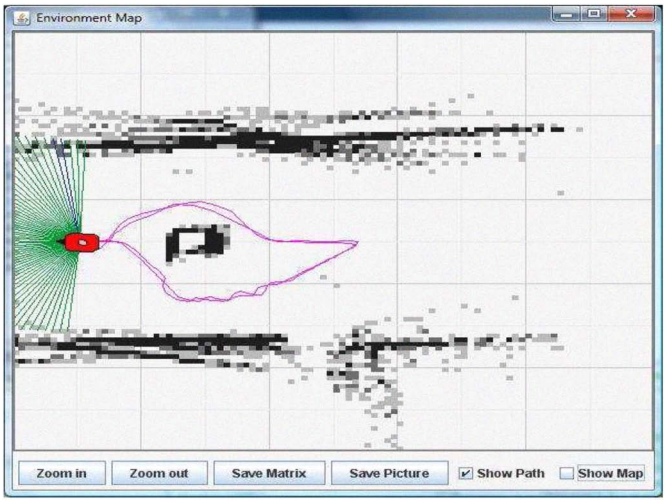

Fig. 6. Potential fields trajectories at low speed without and with handover.

At a higher speeds (250mm/s maximum) the trajectories are shown in Figure 7. Trajectory A was obtained without handover while in trajectory B handovers occur when the robot is close to the obstacle. As expected, at higher speeds, the handover delay impacts negatively on the trajectories.

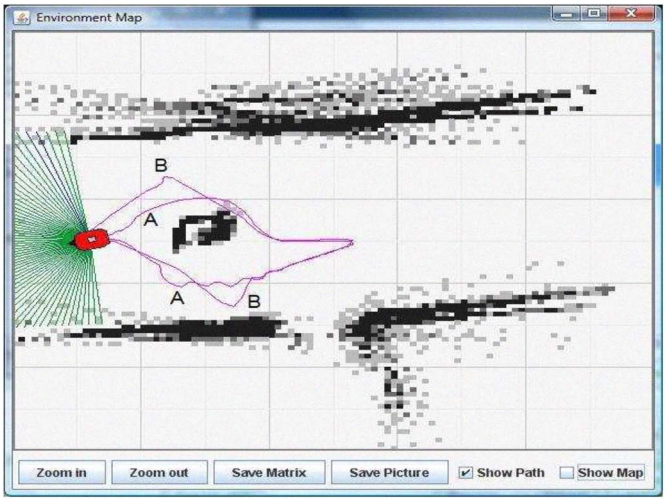

Fig. 7. Potential fields trajectories at higher speed without (A) and with (B) handover.

These experiments show that communication continuity

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5821
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5821

during handover as provided by MPA is fundamental for preserving control in network robotics environments. Handover efficiency depends on factors outside MPA such as how fast the mobile robots switch access points.

V. Related Works

Reference [6] proposes Mobile IPv6 (MIPv6) for supporting handover in network robotics environments. With MIPv6 there is no need of an overlay network as proposed by this paper as this protocol tolerates changes in layer 3 parameters such as network prefix and default router. This advantage, however, is shadowed by the disadvantages of MIPv6. Firstly, we can not see in a near future the deployment of IPv6 networks. Secondly, MIPv6 demands the installation of this protocol on the mobile robots. As these equipments usually lack processing power, MIPv6 becomes unfeasible in many situations. Finally, the handover delays observed (around four seconds) is unacceptable for many robotics applications (e.g., teleoperation). This delay is caused by the autoconfiguration process as employed in MIPv6. Surelly, the use of MIPv6 extensions such as FMIPv6 (Fast Handover MIPv6) or HMIPv6 (Hierarchical MIPv6) would improve handover overheads. Unfortunately, HMIPv6 and FMIPv6 implementations are not readily available as MIPv6.

Our solution places the mobility functions on the network, not on the mobile nodes. As such, mobile robots with very limited processing power can receive the same network service as the more equipped ones. Moreover, the native triggering process reduces handover delays to fraction of second, a figure much more realistic for network robotics applications.

Proposals addressing multicast communication for network robotics can be found in [12]. Multicast is limited to connectionless communications and demands specialized (and bandwidth consuming) routing protocols. Firewalls are usually configured to drop packets targeted to multicast (class D) addresses. In our network architecture the P2MP tunnels can be configured to replicate at forking points the packets targeted to a particular address or port (P2MP tunnels were originally designed for operating in this way). P2MP tunnels support multicast without the need of class D IP addresses (a "firewall friendly" solution). As multicast applications on the Internet did not widespread, the current version of our architecture does not support this form of communication.

VI. Conclusions

Mobile robotics applications demand network solutions that provide secure and reliable communication with the robots independently of their current location. This paper described a network architecture that takes advantage of the existing indoor network deployments. An overlay network built with low cost access points running OpenWRT is the key element of the architecture. The overlay network addresses mobility, security, and class/quality of service. The low handover overhead allows strict teleoperation scenarios as the lack of communication during handover is very low compared with layer 2 handover overheads. The architecture demands no specialized software on the mobile robots, a key point for small mobile apparatus with reduced processing power.

The architecture makes no assumption about the security mechanism employed. We tested the architecture with WPA2 configured both with PSK and RADIUS. With RADIUS, authentication caching prevents the mobile node from performing a RADIUS transaction at each new attachment.

The architecture relies on DiffServ to offer traffic priorization. For instance, control traffic can have precedence over media traffic. As we employ RSVP-TE, resource reservation could be implemented straightforwardly if the existing transport network supports this protocol.

Currently we are investigating how to incorporate network robotics services (e.g., location) on our architecture, how to improve the handover performance on Linux, and how to add outdoor capabilities to the architecture by interworking with other external networks such sensor and 3G cellular networks.

Acknowledgment

The authors at the University of Campinas would like to thank Ericsson Telecommunications of Brazil and Fapesp for supporting this research. Thanks also to Fernando P. Neto, Alessandro Moretti, and Victor V. Pinto for helping the implementation of the REALlab WebLab.

References

[1] J. O. Vatn, "An Experimental Study of IEEE 80211b Handover Performance and its Effect on Voice Traffic," 2003, http://www.it.kth.se/~vatn/research/handover-perf.pdf.

[2] E. Zagari, R. Prado, E. Cardozo, M. Magalhães, T. Badan, J. Carrilho, R. Pinto, A. Berenguel, D. Barboza, D. Moraes, T. Johnson, and L. Westberg, "MPA: a Network-Centric Architecture for Micro-Mobility Support in IP and MPLS Networks," in *IEEE Sixth Annual Conference on Communication Networks and Services Research - CNSR'08*, Halifax, Canada, 2008.

[3] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," The Internet Engineering Task Force (IETF), RFC 3209, December 2001.

[4] S. Yasukawa, "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)," The Internet Engineering Task Force (IETF), RFC 4461, April 2006.

[5] D. Johnson, C. Perkins and J. Arkko, "Mobility support in IPv6," The Internet Engineering Task Force (IETF), RFC 3775, June 2004.

[6] C.-H. Ku and Y.-C. Cheng, "Remote Surveillance by Network Robot using WLAN and Mobile IPv6 techniques," in *TENCON 2007*, Tainan, Taiwan, 2007.

[7] MikroTik Routers & Wireless, 2008, http://www.mikrotik.com/.

[8] OpenWRT Project, 2008, http://openwrt.org/.

[9] T. Johnson, R. Prado, E. Zagari, T. Badan, E. Cardozo, and L. Westberg, "Performance Analysis of a New Architecture for Mobility Support in IP Networks," in *IEEE International Wireless Communications and Mobile Computing Conference - IWCMC'08*, Crete Island, Greece, 2008.

[10] P. R. Coelho, R. F. Sassi, E. Cardozo, E. Guimarães, L. F. Faina, R. P. Pinto, and A. Z. Lima, "A Web Lab for Mobile Robotics Education," in *IEEE International Conference on Robotics and Automation - ICRA'07*, Rome, Italy, 2007.

[11] R. Siegwart, *Introduction to Autonomous Mobile Robots*. MIT Press, 2005.

[12] A. Tiderko and T. Bachran, "A Service Oriented Framework for Wireless Communication in Mobile Multi Robot Systems," in *First International Conference on Robot Communication and Coordination - ROBOCOMM'07*, Athens, Greece, 2007.

Digital Object Identifier: 10.4108/ICST.ROBOCOMM2009.5821
http://dx.doi.org/10.4108/ICST.ROBOCOMM2009.5821