Li *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:11
DOI 10.1186/s13638-015-0514-z

EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

AVATARS: a software-defined radio based teleoperating cyber-physical system for disaster environment exploration

Lian Li*, Cailian Chen, Yiyin Wang, Yong Cao and Xinping Guan

Abstract

In this paper, we develop a software-defined radio (SDR)-based teleoperating cyber-physical system (CPS) for disaster environment exploration, named AVATARS, which includes a telerobot and a teleoperator. The telerobot of AVATARS can be sent into some dangerous or inaccessible disaster environments for search and rescue tasks. Assisted by the environment information collected by the telerobot, the teleoperator can have a better understanding of the detected disaster environment, and the telerobot can be teleoperated more precisely to accomplish tasks effectively. Wireless communications play a vital role for AVATARS to guarantee its flexibility and scalability. To support reliable wireless transmission and real-time wireless teleoperation, we design an efficient and reliable wireless link to enhance the effective throughput of AVATARS. In the link, wireless communication efficiency is improved by using the packets pipelining and accumulative acknowledgement strategy based on the traditional stop-and-wait ARQ protocol. Sequentially, cognitive radio techniques are explored, and an adaptive channel switching mechanism is proposed to mitigate channel interferences in the crowded ISM band. Furthermore, three different kinds of control information are integrated into a single acknowledgement (ACK) packet to reduce communication overhead. The wireless link is implemented on SDR platforms, and extensive experiments under different conditions are carried out to evaluate its performance. The effective throughput of AVATARS can be enhanced more than four times compared with the traditional stop-and-wait ARQ protocol. We also present a video demo to show the powerful and interesting performance of AVATARS.

Keywords: Teleoperation, CPS, Cognitive radio, Effective throughput, SDR

1 Introduction

Teleoperating cyber-physical system (CPS) is a coordinated robotic system with autonomous motion control, local computations, and wireless communications [1]. This kind of system usually includes one or many telerobots and a teleoperator. The telerobot is a device that has autonomous work ability and can also be controlled remotely by a teleoperator through human operation. The teleoperating CPS is able to meet sudden and temporary requirements for exploration and rescue tasks in accidents and disasters [2, 3]. For example, three species of telerobots were used at the World Trade Center (WTC) 9/11 disaster to actively explore the rubble of the building and inspect the building foundations [4]. However, there is a gap between the wealth of distributed information captured by multiple robots and the understanding of the unknown physical situation of the disaster environment. Wireless communications within robots and between robots and teleoperator are essential to bridge this gap by locally carrying out computation/configuration and transmitting the required and partially processed data. Thus, being able to quickly, reliably, and cost effectively transmit the fused data of vision, infrared, sonar sensors, and other sensors is fundamental for disaster environment exploration.

In this paper, we develop a software-defined radio (SDR)-based teleoperating CPS, called AVATARS, for disaster environment exploration. Here, the name of AVATARS borrows from video games to represent the mobile agent (the robot) as an eyeball. We focus on

*Correspondence: li_li_an@sjtu.edu.cn
Department of Automation, Shanghai Jiao Tong University, 200240 Shanghai, China

© 2016 Li et al. **Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

designing an efficient and reliable wireless link to support real-time teleoperation and reliable wireless data transmission between the telerobot and the teleoperator. The telerobot is functioned with video sensing and sensors for environment detection and obstacle avoidance. It is remotely controlled by the teleoperator according to the analysis of real-time video and other sensor information. Based on SDR technologies and platforms, we can implement the wireless link with adaptive and dynamic adjustment ability by means of software rather than hardware. It is much flexible to update and upgrade the wireless communication link using SDR technologies according to the actual communication environment. Moreover, we can design and implement the wireless link to achieve the specific goals of our specific teleoperating CPS with SDR technologies.

Design Goals: The design of AVATARS focuses on an efficient and reliable wireless link between the telerobot and the teleoperator. AVATARS pursues the following three design goals: (1) AVATARS is aiming to enable real-time and reliable data transmission with low-cost SDR platforms and technologies; (2) AVATARS is trying to combat the dynamic and unpredictable channel interferences in the crowded ISM band; and (3) AVATARS is striving to reduce the communication overhead of control information induced by controlling robot and switching channel.

Challenges: Although SDR techniques are low cost and flexible, the proposing system still faces several challenges. First, the original data information has a high demand of communication throughput. However, the communication throughput of the low-cost SDR platforms is limited because of many kinds of latencies [5, 6]. Hence, packets pipelining transmission is proposed to improve the communication efficiency. Second, it is quite challenging for wireless communications under the circumstances of imperfect channels and unreliable links. As a result, retransmission and acknowledgement (ACK) are necessary. Moreover, wireless communications may be interfered by other wireless radios in the same crowded ISM band, and the channel reliability is dynamic and unpredictable. It is possible that even the best channel at this moment is less reliable than any other channels at the previous or the next moment. In this case, an observation on link quality is used to decide whether a channel switching will be taken. Last but not the least, the transmission of robot control information and channel switching information from the teleoperator to the telerobot can consume non-trivial communication throughput. Therefore, we consider to piggyback these information with each ACK packet in each round-trip communication.

Contributions: In this study, we present the design, implementation and evaluation of AVATARS that can be deployed quickly and flexibly into disaster environments to obtain different kinds of information. The main contributions of our work can be summarized as follows:

- We design and implement a teleoperating CPS (AVATARS) used for disaster environment exploration. It is developed based on SDR platforms (NI USRP [7] and NI PXI [8]) and a robot. To the best of our knowledge, this is the first work to rebuild the USRP with a robot and enhance its communication throughput for teleoperating CPS.
- In particular, to improve the effective throughput with low-cost SDR platforms and flexible SDR technologies, we design an efficient and reliable wireless link with three communication strategies for AVATARS. First, AVATARS employs the packets pipelining and accumulative acknowledgement strategy based on the traditional stop-and-wait automatic repeat-request (ARQ) protocol to improve the communication efficiency. Second, AVATARS makes use of cognitive radio techniques and adopts an adaptive channel switching mechanism to mitigate dynamic and unpredictable channel interferences. Third, AVATARS integrates three different kinds of control information into a single ACK packet to reduce the communication overhead of these information.
- We experimentally show significant performance improvement of the proposed communication strategies over traditional communication mechanisms. In different experiments, we vary the window size of packets pipelining, the interference level, and the different channel switching techniques. Our results demonstrate that the communication efficiency can be improved three times and the effective throughput is improved four times over the traditional static communication systems.

In the rest of the paper, Section 2 presents a brief background. The overview is introduced in Section 3, and the technical details of the proposed wireless link are given in Section 4. Section 5 provides implementation details and experimental results. A video demonstration is presented in Section 6, and conclusions are given in Section 7.

2 Background

In this section, we provide a brief background on our system, particularly on teleoperating CPS for disaster environment exploration, strategies about enhancing the efficiency of low-cost SDR platforms, and ISM band channel switching.

Teleoperating CPS: Teleoperating CPS is regarded as an extension of a human into a disaster environment, providing real-time video and other sensory data about the situation. Murphy et al. summarize the basic characteristics of disasters and their impact on robotic design and describe that the telerobots were actually used in disasters of past several years [3]. Most of the teleoperating CPS applications rely on real-time communications for teleoperation and for enabling the teleoperator to see what the robot is seeing immediately, although wired communications are more reliable and have been adopted for the ground robots [9–11]. However, it is not suitable for aerial and surface robots. Moreover, the wired communication can only be used between the telerobot and the teleoperator, and its application for multi-robot cooperation is limited. The wireless communication is a better choice due to its flexibility and broadcast property. In [12], Li et al. present a wireless robot teleoperation system based on the CDMA1X mobile network. The commercial mobile communication network gets rid of the spacial restriction and is favorable for wide-area tasks. However, disasters may degrade or destroy the fixed infrastructures, such as cellular base stations. Moreover, it is very expensive to use the dedicated commercial spectrum. In [13], Xue et al. propose a teleoperation system using Bluetooth for wireless communications in the ISM band. Frequency hopping is used by Bluetooth to avoid random interferences. For continuous channel interferences, frequency hopping is not reliable because it cannot avoid interferences smartly and dynamically. Moreover, the Bluetooth is used for short-range communications. In this paper, we do not use commercial networks, but design a reliable wireless link in the ISM band with the ability to avoid unpredictable interferences for the AVATARS teleoperating CPS.

SDR: SDR [14] is a technology which facilitates experimentations and practical realizations of cognitive radio techniques and novel wireless communication systems. Baldini et al. designed an emergency communication system based on SDR to support public protection and disaster relief operations [15]. Due to its flexibility, the SDR technology is also widely applied in wireless protocol design, practical performance evaluation, and prototype implementation [16].

Since the wireless link is unreliable, ACK and retransmission are necessary to guarantee the reliable data transmission. The stop-and-wait ARQ protocol is a popular method with ACK and retransmission. It has been widely used in the wireless standards such as 802.15.4, 802.11. For the stop-and-wait ARQ protocol, the round-trip time (RTT) is a key factor to measure the communication efficiency and effective throughput. Nevertheless, the latency of the USRP SDR platform is measured in [5], and its impact on throughput in modern wireless protocols is presented. Software signal processing, data queuing, and bus transfer delay easily add up to hundreds of milliseconds [6]. Therefore, the RTT usually costs tens to hundreds of milliseconds and leads to a low communication efficiency. Due to this fact, many works are striving to enhance the performance and efficiency of low-cost SDR platforms, but still keep their flexibility and simplicity.

Researchers try to shift lower layer (physical and MAC layers) implementation to FPGA hardware [17–19] to reduce latencies. However, this approach increases the complexity and limits the flexibility of developing and experimenting with new protocols. There are also some SDR platforms based on either programmable hardware such as field programmable gate arrays (FPGAs) [20] or embedded digital signal processors (DSPs) [21]. These platforms can meet the processing and timing requirements of high-speed wireless protocols; however, the implementation on FPGAs and DSPs are time-consuming. It is not suitable for fast prototyping. Moreover, such hardware platforms are much more expensive.

To reduce the RTT for the ARQ protocol using the low-cost SDR platform, an optimized frame transmission path is designed in [22] by pre-modulating the transmission frame. In [23], a dedicated spectrum sensing engine is designed to enhance the sensing speed of a pure software implementation of a CSMA-based MAC protocol. Even the dedicated hardware can improve the performance of SDR implementation, there are some additional hardware costs. In this paper, to overcome the processing delay of the low-cost USRP platform and improve the communication efficiency, we not only optimize the frame transmission path as in [22], we also employ the packets pipelining transmission. Moreover, the transmission of robot control information and channel switching information wastes communication throughput if using the existing Zigbee or WiFi protocols. However, based on the USRP platforms, we can piggyback the information with each ACK packet to improve the effective throughput of the wireless link further.

ISM band channel switching: Co-channel or adjacent-channel interference becomes more and more serious in the ISM band [24, 25] because of the proliferation of wireless devices and standardized protocols in this licensed free band. The conventional static wireless protocols, such as Zigbee, WiFi, may not guarantee the quality of service (QoS) requirements for some specific applications under the complex and dynamic spectrum environment. There are protocols such as Bluetooth [26] and WirelessHART [27]. Both adopt channel hopping to avoid channel interferences in the ISM band. Nevertheless, the hopping pattern is fixed and it cannot be updated according to the communication environment. Moreover, both protocols require critical time synchronization, which brings extra costs and overheads. An adaptive channel allocation scheme is proposed in [28], which allows Zigbee nodes under significant interferences to switch to new channels with less collisions. To minimize the effect of WiFi interferences to Zigbee nodes, an interference detection and avoidance strategy is proposed in [29], which selects the channel that is least likely to have interferences. Parameters, such as link quality indicator (LQI) [30] and packet deliver ratio (PDR) [31], are used as metrics for channel switching. However, they are not directly related to interferences. In this paper, combined with packets pipelining transmission, we measure the packet loss probability in each communication round to detect the interference. Only after the packet loss probability obviously increases, the system starts spectrum sensing and then dynamically accesses the best channel in its vicinity. We make use of the cognitive radio idea in the crowded ISM band, adopt an adaptive channel switching mechanism in the wireless link to avoid channel interferences, and improve communication reliability.

3 Overview of AVATARS

In this section, we give an overview of AVATARS, including system blocks, system states and system metrics.

3.1 System blocks

The overview of AVATARS is shown in Fig. 1. It consists of a telerobot and a teleoperator. The telerobot includes a mobile robot and a USRP. The onboard computer on the robot connects the USRP through a gigabit ethernet and operates the movement of robot through a serial port. The teleoperator implemented on NI PXI consists of three components including an RF signal analyzer (RFSA), an RF signal generator (RFSG), and an embedded controller, which is a general computer used for signal processing and user interaction.

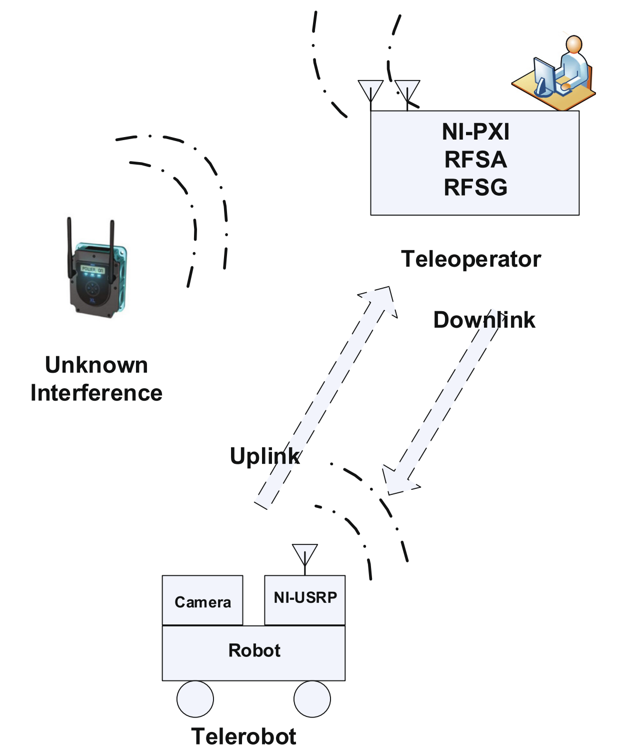


Fig. 1 The overview of AVATARS

The telerobot explores the unknown disaster environment, acquires the environment information, including video information, infrared sensor and ultrasonic sensor information. In this paper, only video information is transmitted to the teleoperator using our designed wireless link, since it is the most challenging one. However, it is easy to extend our system to transmit different kinds of information. The telerobot moves around according to the commands of the teleoperator. On the other side, the teleoperator is used to display the video information acquired by the telerobot to end users and send control commands to the telerobot. As shown in Fig. 1, wireless communications between the telerobot and teleoperator may face severe unknown interferences in the ISM band. Hence, it is challenging to transmit the information reliably and quickly, and it is also demanding to control the telerobot precisely and timely under the interferences using conventional static protocols.

3.2 System states

The stop-and-wait ARQ protocol has been widely used in the wireless standards such as 802.11, 802.15.4. The wireless link of these standards cannot adaptively adjust the communication channel according to the channel reliability. Figure 2 shows an efficient and reliable wireless link with an adaptive adjustment ability. Different from the state transition of the traditional static wireless link, the yellow color states are added to analyze the communication environment and adjust the communication parameters according to the analysis results as shown in Fig. 2. Moreover, a synchronization state is necessary to avoid the link failure due to channel-switching mismatch.

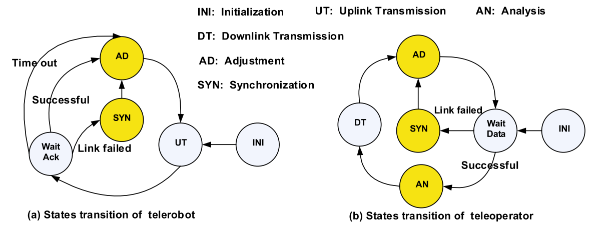


Fig. 2 State transition of AVATARS. **a** States transition of telerobot, **b** States transition of teleoperator

Figure 2(a) shows the five states of telerobot transmitter and state transition among them. The first state is INI (initialization). In this state, the telerobot initializes system parameters according to pre-configurations. After initialization, it transfers to the UT (uplink transmission) state, where the telerobot begins to acquire the video information and transmit the packets using packets pipelining strategy. After continuously transmitting several packets in a round, it enters into the Wait-ACK state to wait for the ACK from the teleoperator receiver. There are three different possibilities after the Wait-ACK state. If the ACK is received successfully during the given time, the telerobot will transfer to the AD (adjustment) state, where it adjusts the communication channel, prepares the lost packets, and makes movements according to the ACK packet. If the ACK times out, the AD state will form the retransmitted packets with the whole packets of the last round and ignore the channel adjustment and robot movement. For the third case, if the telerobot does not receive any ACK after retransmitting the packets R rounds, it will enter into the SYN (synchronization) state; this state will be explained later.

Along with the state transition of the telerobot, the six states and transition relationship of the teleoperator are shown in Fig. 2(b). In the wait-data state, if packets are successfully received, the teleoperator will enter into the AN (analysis) state, checking the received packets, monitoring the link quality, analyzing the communication channels, and generating the corresponding ACK packet. Then, the ACK is transmitted in the DT (downlink transmission) state. The teleoperator also adjusts the communication channel in the AD state. In the wait-data state, if the teleoperator does not receive any data for a time period T_s, it will also transfer to the SYN state.

Let us discuss the SYN state. Although we use the adaptive channel switching mechanism to avoid interferences, the link failure is sometimes unavoidable due to channel interferences or ACK lost. If the link failure happens, the teleoperator and telerobot will switch to different channels and fail to communicate with each other. To address this, AVATARS introduces a SYN state that devices enter into when they detect a link failure. In the SYN state, devices communicate on a dedicated narrow-band channel to exchange the channel switching information. Then both devices enter into the AD state to adjust the communication channel according to the shared information.

3.3 System metrics

In this paper, we focus on wireless communications between the telerobot and teleoperator, and try to improve the communication efficiency and effective throughput using different strategies in the wireless link. The communication efficiency of wireless link is defined similarly as in [22, 32] as

$$\eta = \frac{M * T_{packet}}{RTT} \tag{1}$$

where RTT is the round-trip time. It is also the total time cost from the beginning of M packets transmission to the end of decoding the ACK packet. M is the window size (number of packets) in each transmission round. For the traditional stop-and-wait ARQ protocol, $M = 1$. However, for the packets pipelining and accumulative acknowledgement strategy proposed in this paper, we can set M flexibly. T_{packet} is the transmission time of each packet and $T_{packet} = L/V$, where L is the length of each packet in bits and V is the data rate. Moreover, combined with the reliability of each packet transmission [33], the effective throughput of AVATARS is defined as follows

$$E_{put} = \frac{M * (1 - P_e) * L}{RTT} \tag{2}$$

where P_e is the lost probability of each packet. E_{put} can also be regarded as the average throughput of the M packets transmission in each round.

4 Wireless link design

In this section, we present the technical details of an efficient and reliable wireless link design. In the wireless link, the packets pipelining and accumulative acknowledgement strategy is proposed based on the traditional ARQ protocol to improve the communication efficiency. Second, the cognitive radio idea is utilized, and an adaptive channel switching mechanism is employed to mitigate dynamic interferences and improve the reliability of transmission. At last, the integration design of the ACK packet reduces the overhead of control information exchange and improves the efficiency and throughput further.

4.1 Packets pipelining and accumulative acknowledgement

As we know the wireless link is unreliable, the acknowledgement and retransmission are necessary to guarantee the communication quality. However, per-packet acknowledgement of the traditional stop-and-wait, ARQ mechanism reduces the efficiency of communication because of frequent acknowledgement overhead. As shown in Fig. 3(a), the RTT of the stop-and-wait ARQ mechanism consists of five components in each round-trip communication. The turnaround time (T_{td}) is the switching delay from the receiving to the transmitting state. The packet and the ACK time (T_{packet}, T_{ack}) are the transmission time of a packet and an ACK given a specified transmission speed, respectively. The LIFS time (T_{LIFS}) is the processing delay of each packet, and the SIFS time (T_{SIFS}) is the processing delay of the ACK.

Due to the switching and processing delay of the USRP platform [5, 6, 18, 22], the RTT for a single packet

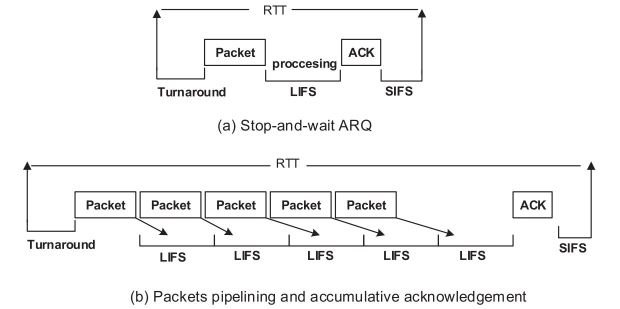


Fig. 3 Time cost of a round-trip communication. **a** Round-trip time constitution of the traditional stop-and-wait ARQ protocol. **b** Round-trip time constitution of the packets pipelining and accumulative acknowledgement strategy

transmission easily adds up to tens and hundreds of milliseconds, which lead to a low communication efficiency. Figure 3(b) shows details of the packets pipelining and accumulative acknowledgment strategy adopted in the wireless link. The telerobot transmitter continues to send a number of packets specified by the window size M in a pipelined way. The teleoperator receiver captures and processes the packets in a concurrent way to improve efficiency. After processing all packets in the round, the teleoperator will then acknowledge the telerobot with an accumulative ACK to reduce the overhead.

Due to the processing delay of software implementation on the SDR platform, T_{LIFS} is usually larger than T_{packet}. Thus based on the time components in Fig. 3, the RTT can be obtained as follows

$$RTT = T_{td} + T_{packet} + M * T_{LIFS} + T_{ack} + T_{SIFS} \quad (3)$$

Note that the communication efficiency η is calculated through (1) and is related to each parameter in (3). We can optimize the communication efficiency by transmitting packets with different sizes [34] or reducing the processing time of each packet with FPGA implementation. In this paper, we focus on increasing with the window size M to improve the communication efficiency. However, the window size cannot be set too large because of two reasons. First, additional buffers are needed at both transmitter and receiver. The telerobot transmitter requires a buffer of size M to store the previous round packets until they are successfully delivered, and the teleoperator receiver needs a buffer size larger than M to reorder the disordered receiving packets. Second, the retransmission delay of each lost packet is proportional to the window size M. Hence, the window size M should not be chosen arbitrarily due to the tradeoffs among the communication efficiency, the additional buffer cost, and the retransmission delay of the lost packets.

Since the lost packets cannot be retransmitted immediately, the sequences of receiving packets may be out of order. We design a special store-and-fetch method to reorder the receiving packets. As shown in Fig. 4, there is a circular buffer to store the receiving packets, and its length B_l is larger than window size M. Each element in the buffer consists of two parts, one part is a flag to indicate the validity of each packet and the other part is the effective data of each packet. The successfully received packets will be stored into the circular buffer with the corresponding flag being set true. However, the corrupted packets will be discarded and the corresponding buffer will be empty. Each packet has a unique number P_n, and the store position in the buffer is calculated by $P_n \mod B_l$, which is the remainder between the packet number and the buffer length.

The fetch process is conducted parallel with the store process. Each correct packet is fetched out of the buffer sequentially, and the flag is set to false after the corresponding packet being fetched out. Once the fetch process moves to a position with the false flag, it stops and waits for retransmission of the lost packets. The lost packets will be retransmitted at the beginning of the next communication round. Because of the unique packet number, the false packets will be stored into the right positions of the buffer, where the fetch process is waiting for them.

Figure 4 is a simple example; the window size $M = 5$ and the buffer length $B_l = 15$. The fourth packet (number 3 in Fig . 4) of the first communication round does not pass cyclic redundancy check (CRC). Hence in Fig. 4(a), the fourth flag is false and the flags of other four are true. Figure 4(b) is the fetch process, and the correct packets are fetched out until the fourth packet. After fetching the corresponding packet, the fetch process sets the flag to false and records the fetch position . In Fig. 4(c), there is a

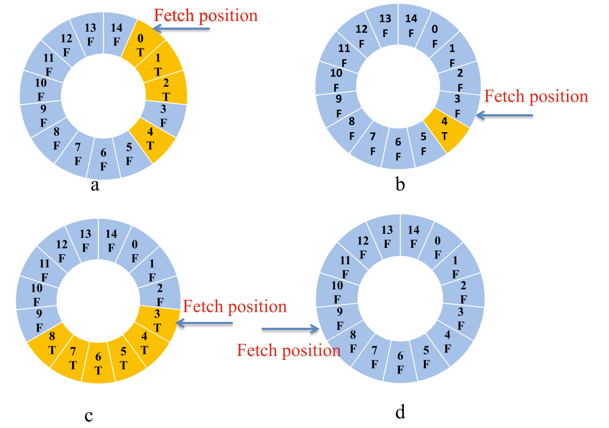


Fig. 4 Reorder the receiving packets. **a** Store process with one lost packet. **b** Fetch process with waiting for the lost packet. **c** Store process with the lost packet retransmission. **d** Fetch process with all correct packets fetched out

new transmission round with the false packet retransmission, and in Fig. 4(d), the fetch process fetches all correct packets.

4.2 Adaptive channel switching

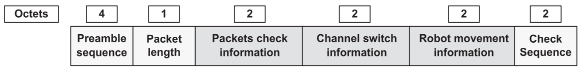
Co-channel or adjacent channel interference is a serious problem in the crowded ISM band [24, 25], and it reduces the effective throughput of the system severely. The wireless link may be blocked due to unpredictable channel interferences. Hence, we adopt an adaptive channel switching strategy to avoid dynamic and unpredictable interferences. Combining with the packets pipelining and accumulative acknowledgement strategy, the teleoperator will monitor the link quality and make a decision about channel switching based on packets check results P_e in each communication round. If $P_e < \epsilon$, here ϵ is a given threshold, the spectrum sensing and channel switching event will not be triggered. On the other hand, if $P_e > \epsilon$ in a transmission round, the teleoperator will scan each channel, measure the interference power in each channel, and then select the cleanest channel. Obviously, the design of the threshold for channel switching is very important. In this paper, we set the threshold according to the link reliability without channel interference. It can be obtained through the wireless link test and evaluation. The channel switching information will be integrated into the ACK packet and shared with the telerobot. The event triggered sensing and switching mechanism can also avoid unnecessary sensing and switching overheads.

Through the adaptive channel switching mechanism, the unwanted channel interference can be mitigated and the packet loss probability (P_e) can be reduced immediately. As a result, the communication effective throughput can be improved greatly.

4.3 Integration design of ACK

There are two types of control messages (packets check results and channel switching message) needed to be transmitted from the teleoperator to the telerobot. Moreover, the teleoperator needs to control the movements of the telerobot in an unknown environment. Hence, the movement message is also needed to be transmitted from the teleoperator to the telerobot. To reduce the transmission cost of these control information, three kinds of control messages are integrated into a single ACK packet.

The structure of the proposed ACK packet is shown in Fig. 5. The preamble sequence enables the telerobot to achieve symbol synchronization. The check sequence is used to verify the validity of the ACK packet. Each kind of control messages contains 16 bits. The packets check results bit is a boolean map to indicate the validity of each packet in a communication round. The first bit of the channel switching message indicates whether channel switching is necessary, and the remaining bits indicate the preferred switching channel. Similarly, the first bit of the robot movement message indicates whether there is a movement command sent by the teleoperator, and the remaining bits indicate the shared movement command. Using the integration design of the ACK packet, we not only reduce the exchange overheads of control information but also improve the effective throughput of the system.



Octets	4	1	2	2	2	2
	Preamble sequence	Packet length	Packets check information	Channel switch information	Robot movement information	Check Sequence

Fig. 5 Structure of an ACK packet

4.4 A unified software program flow

Although three different strategies are designed to solve different challenges, we can combine these strategies into a unified software program flow as shown in Fig. 6. The yellow-colored blocks illustrate the packets pipelining transmission, the blue colored blocks show the adaptive channel switching strategy, and the purple colored blocks present the integration design of the ACK packet. The unified purpose of the three strategies is to increase the effective throughput of the wireless link. Each strategy is closely related with each other. From the different colored blocks in Fig. 6, we can see the correlations of the three strategies. First, the packets pipelining transmission strategy relies on the ACK packet to delivery the packets check results. Second, for the adaptive channel switching mechanism, the link quality is judged by the packets check results of the packets pipelining transmission in each communication round. Finally, the integration design of the ACK strategy includes the packets check results, the channel switching message, and the movement message in a single packet to reduce communication overheads.

5 Implementations and experimental results

We first implement a prototype based on the SDR platforms NI PXI and USRP to demonstrate the feasibility of our proposed wireless link with the packets pipelining and accumulative acknowledgement strategy and the adaptive channel switching mechanism. Moreover, we also implement the AVATARS with the SDR platforms and an autonomous robot, and all proposed strategies are implemented in the system. In this section, we will show our implementation details and experimental results.

5.1 Packets pipelining and accumulative acknowledgement

5.1.1 Implementation

As shown in Fig. 7, the prototype system consists of a USRP as the transmitter and a PXI platform with RFSA and RFSG as the receiver. The interference nodes are not

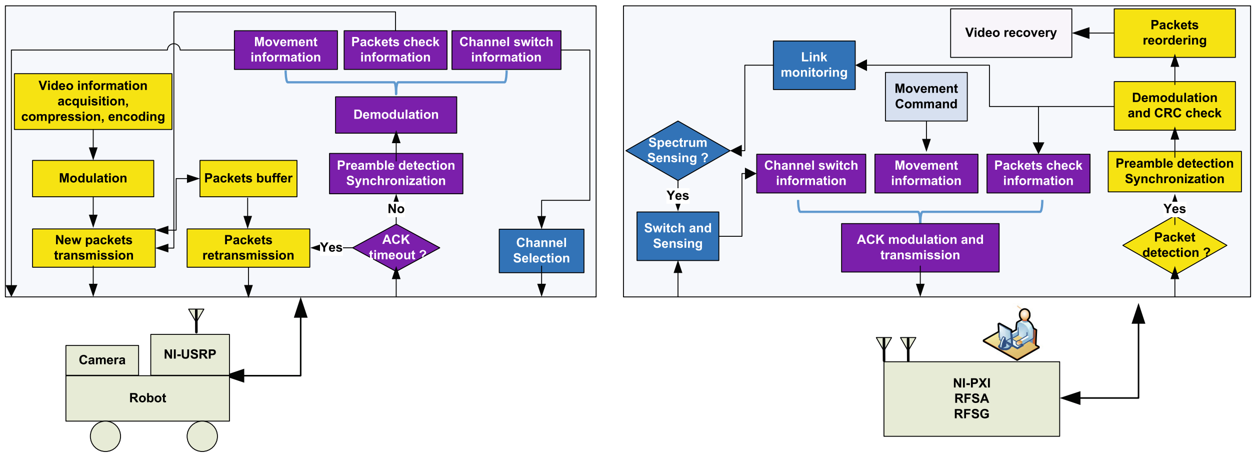


Fig. 6 Software program flow of AVATARS. The *left* side is the software program flow of telerobot, and the *right* side is the soft program flow of teleoperator

activated in this evaluation. The USRP device and PXI platform are connected to computers which run the program to evaluate the packets pipelining and accumulative acknowledgement strategy. We configure the power of the transmitter so that wireless link maintains 10 % or less packet loss when there is no channel interference present. The system parameters are shown in Table 1. The bit rate is configured as 250 kb/s with QPSK modulation, and the carrier frequency is selected as 2.45 GHz. The number of bits L in a packet is 4325 bits with 4096 bits payload length. We evaluate the communication efficiency η with different window size M. For each specific window size, each experiment takes 50 s and is repeated five times, thus we count successfully delivered packets during the 50 s to obtain the communication efficiency.

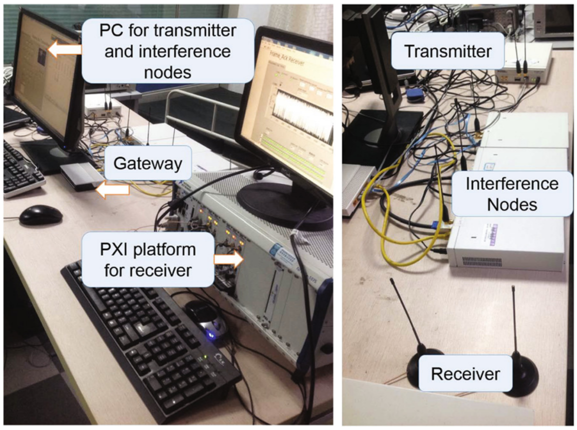

PC for transmitter and interference nodes
Gateway
PXI platform for receiver
Transmitter
Interference Nodes
Receiver

Fig. 7 Prototype system. A prototype system to evaluate the benefits of the packets pipelining and accumulative acknowledgement strategy and the adaptive channel switching mechanism

5.1.2 Experimental results

According to the bit rate and packet size, the packet transmission time (T_{packet}) is 17.3 ms, and the ACK transmission time (T_{ack}) is approximate 1 ms. Due to the processing delay of the SDR platform, the processing time of the packet (T_{LIFS}) and ACK (T_{SIFS}) are approximately 30 ms and 5 ms, respectively. However, the turnaround delay (T_{td}) is almost 55 ms because of the delay of the USRP driver.

Figure 8 shows the communication efficiency versus different window sizes in a round-trip communication. We

Table 1 Hardware configuration and link parameters

Parameter	Value
Carrier frequency	2.45 GHz
Sampling rate	2 Msample/s
Channel bandwidth	125 KHz
Bit rate (R)	250 bits/s
Modulation scheme	QPSK
Channel number	3
Packet size (L)	4325 bits
Max transmission power	10 dBm
Max range	50 m
T_{packet}	17.3 ms
T_{ack}	1 ms
T_{LIFS}	30 ms
T_{SIFS}	5 ms
T_{td}	55 ms
T_{switch}	10 ms
$T_{sensing}$	4.5 ms

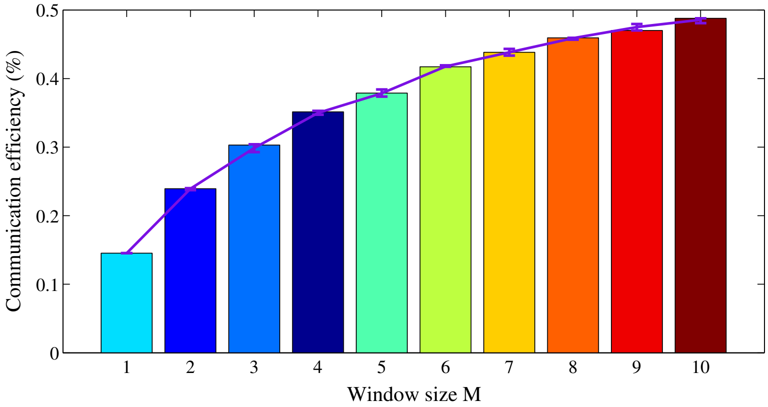


Fig. 8 Communication efficiency vs. window size in a round-trip communication

observe that the communication efficiency increases obviously with the window size augments. Especially, when the window size M is less than 5, the growth tends to be linear and rapid. However, when M is larger than 7, the growth speed is slowed down. To reduce additional buffer costs and the retransmission delay of lost packets, we set the maximum window size in our system to be 10. Based on Fig. 8, we can also observe that the communication efficiency is less than 0.5, which is limited by the turnaround time cost in a round-trip communication. For the traditional stop-and-wait ARQ protocol, the communication efficiency is approximately 0.15. By using the packets pipelining and accumulative acknowledgement strategy, the communication efficiency can be improved more than 3 times than the traditional stop-and-wait ARQ protocol.

5.2 Adaptive channel switching

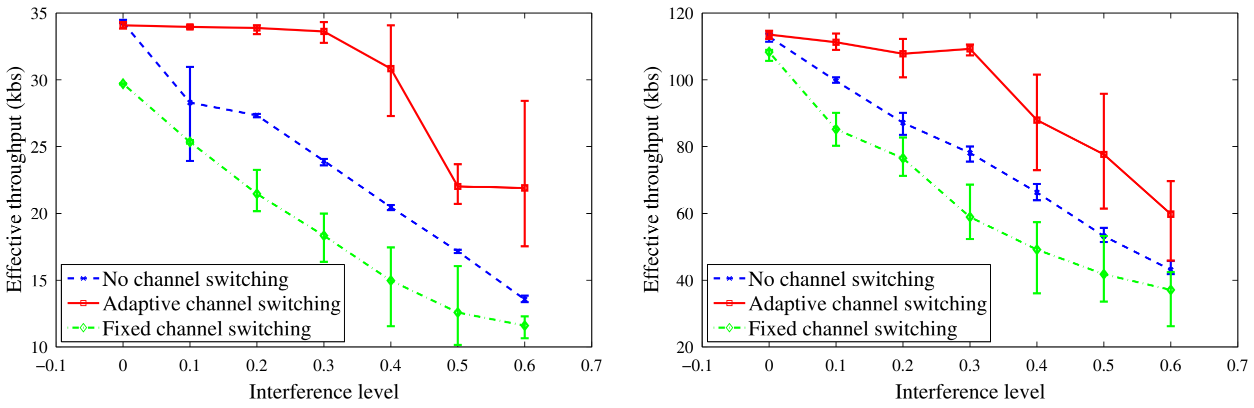
5.2.1 Implementation

The interference nodes are activated to evaluate the benefits of the adaptive channel switching strategy adopted in the system. The interference power is about 5 dBm, and the interference has an obvious effect on the packet loss probability of the link. Other two common strategies are also evaluated to compare the performance. One of them is the no-channel-switching strategy, which means the communication channel remains unchanged even under interferences. The other is the fixed-channel-switching strategy, which is employed by the Bluetooth protocol [26] and WirelessHART [27] standard. The communication channel will change according to a given pattern.

We evaluate the effective throughput of different channel switching strategies with different interference levels. When the interference is present, the packet loss probability (P_e) will obviously increase. The interference level is a ratio between the time duration that interference nodes transmit on the channel and the total time consumption of each experiment. There are three channels and three interference nodes in each experiment. Each interference node is set to interfere a dedicated channel. The time instant when the interference node appears on its channel is random and independent. Each experiment lasts 50 seconds and is repeated 5 times. We still count the correctly transmitted packets to obtain the effective throughput of different channel switching strategies. The transmitter and receiver may enter the SYN state due to channel interferences, we set the synchronization parameters $R = 3$ and $T_s = 500$ ms in these experiments.

5.2.2 Experimental results

Figure 9 shows effective throughput versus different interference levels with different channel switching strategies. We only present the performance of window size $M = 1$ and $M = 10$ for simplicity. At first, we observe that the effective throughput is consistent with the communication efficiency when there is no interference. When $M = 1$, the effective throughput is close to 0.15 times of the transmission bit rate, and for $M = 10$, it is close to 0.5. Second, by comparing the results in Fig. 9(a, b), the effective throughput greatly improves by using the packets pipelining and accumulative strategy. Third, the adaptive channel switching strategy always performs the best when dealing with channel interferences. Especially when the interference level at each channel is relatively low (less than 0.3), the effective throughput nearly keeps the same level as no channel interference exists. However, for the other two channel switching strategies, the effective throughput deceases linearly even if the interference level is low. Finally, it is quite interesting that the fixed-channel-switching strategy performs the worst among all. This is due to the following two reasons. First, the hardware takes several milliseconds to finish the channel-switching-process. Therefore, the fixed-channel-switching strategy performs worse than the no-channel-switching strategy



(a) Window size M = 1 (b) Window size M = 10

Fig. 9 Effective throughput versus interference level with different channel switching strategies. **a** Window size $M = 1$ in a communication round and it also represent traditional stop-and-wait ARQ protocol. **b** Window size $M = 10$ in a communication round

when no channel interference appears on each channel. Nevertheless, the adaptive channel-switching strategy does not bring any extra switching cost and performs as well as the no-channel-switching strategy when no interference exists. Second, the fixed-channel-switching policy may always experience channel interferences because the interference appears on each channel randomly and independently. It cannot avoid the channel interference intelligently as the adaptive channel switching strategy does.

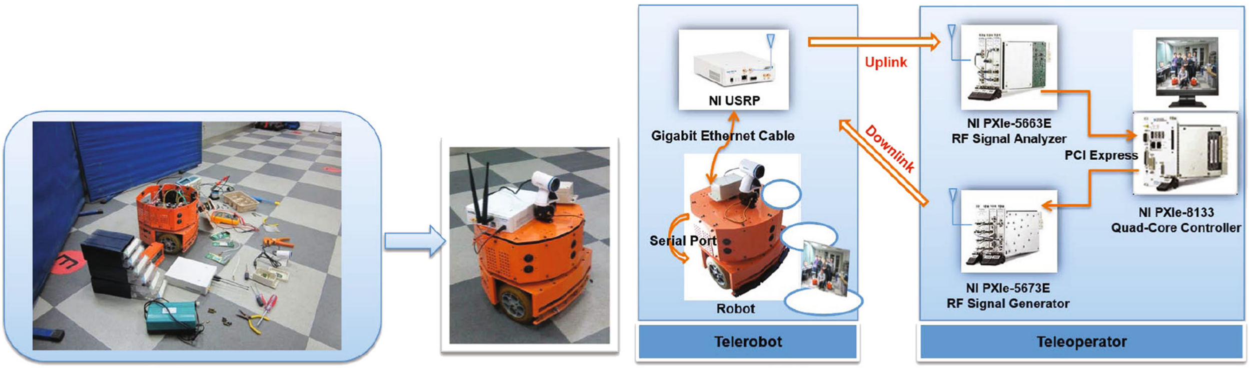
In summary, we prove that the adaptive channel-switching strategy not only avoids channel interferences smartly, but also avoids unnecessary channel-switching cost when no interference appears. Hence, the effective throughput can be improved greatly when we combined the packets pipelining and adaptive channel switching strategies together in the communications.

5.3 AVATARS

5.3.1 Implementation

Figure 10 shows the hardware structure of the telerobot and the whole AVATARS. As shown in Fig. 10(a), the telerobot consists of a USRP and an autonomous robot with a camera on it. In this paper, we only transmit the video information to the teleoperator. An onboard computer on the robot connects the USRP through a gigabit ethernet and operates the movements of the robot through a serial port.

According to Fig. 10(b), the system includes a telerobot and a teleoperator. The teleoperator is the PXI platform with three components. The RF signal analyzer is used for receiving signal, and the RF signal generator is used for transmitting signal. The Quad-core embedded controller is a general computer used for signal processing and user interaction.



(a) Implementation of telerobot **(b)** Implementation of AVATARS

Fig. 10 Hardware implementation of telerobot and AVATARS. **a** Hardware implementation of telerobot. **b** Hardware implementation of the whole AVATARS

The AVATARS experiments are conducted in the building of SEIEE of Shanghai Jiao Tong University. We put the teleoperator and the telerobot in two different offices and the maximum communication range is about 50 m using the transmission power 10 dBm. In the two offices, there are lots of obstacles to mimic the disaster environment and the wireless communication is non line of sight. There are three interference nodes in the two offices to transmit signals randomly and independently.

The AVATARS works in the following steps. First, the teleoperator and the telerobot initialize the system parameters according to the configuration. The major parameters are the window size M, bit rate, carrier frequency and channel list. Then the telerobot begins to acquire the video frame through the camera, and encode the data into bits stream. The bits of each frame are attached with head bits and tail bits to synchronize each frame. Each frame is packed into several packets. After the QPSK modulation, the packets will be transmitted through the USRP using the packets pipelining and accumulative acknowledgement strategy in each communication round. The teleoperator receives and processes the packets, generates the packets check results and monitors link quality. The adaptive channel switching mechanism is used to determine the channel switching message. The teleoperator can also generate the robot movement message according to the user input. A single ACK packet integrating three kinds of control information is transmitted to the telerobot at the end of each communication round. The teleoperator and telerobot adjust the communication channel according to the channel switching message. The telerobot also adjusts the movements of robot and the new transmission packets according to the ACK packet.

By using the hardware as shown in Fig. 10 and implementing all software algorithms on the SDR platforms, we evaluate the video transmission rate of the AVATARS versus different interference levels with different strategies.

5.3.2 Experimental results

Figure 11 shows the video transmission rate versus different interference levels with different strategies. The size of each video frame is approximately 15 kbits. The telerobot generates a steady stream of traffic according to the throughput of the wireless link. More video frames will be captured and encoded if the effective throughput of the wireless link is high. Otherwise, the telerobot reduces the captured frames. We can see the video transmission rate is improved more than three times by using the packets pipelining and accumulative acknowledgement strategy. However, without the adaptive channel switching strategy, the video transmission rate decreases obviously even with a low level of interferences. By combining these two strategies, the video transmission rate increases almost four times even with heavy interferences. Moreover, the integration design of the ACK packet also contributes a lot to increase the transmission speed of AVATARS.

6 Video demo of AVATARS

We demonstrate AVATARS in a video. There are three parts in the demo[12]. The first part is the background introduction. In the second part, we implement the AVATARS using a simulated robot of LABVIEW Robotics. As shown in the demo, the simulated robot can be controlled by the PXI teleoperator to accomplish dedicated movements or automatic movements. The video information of unknown environments can be captured by the camera on the robot and be transmitted through the USRP to the PXI teleoperator. The received wireless signal is processed to recover the video image to end users. There are two interference nodes in the demo. When the communication channel of the system is jammed, the link is interrupted and video transmission is blocked. After enabling the adaptive channel switching strategy, the video transmission link establishes again immediately. When we set up the other interference node, the system

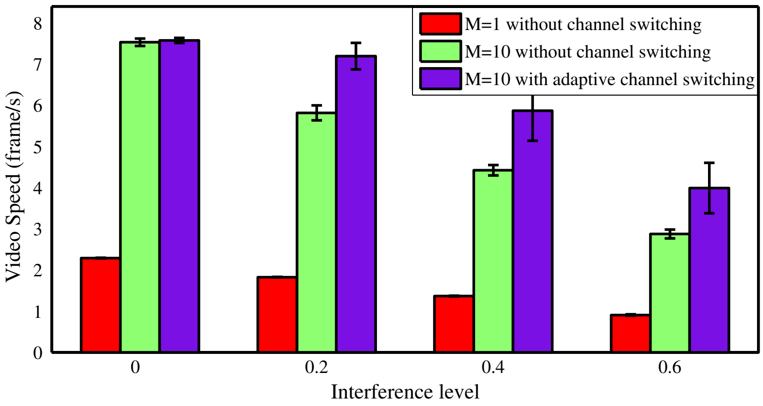


Fig. 11 Video transmission rate versus interference level with different strategies

can switch to another clean channel smoothly and reliably. The last part of the demo is a practical implementation of the AVATARS. We adopt a practical robot and rebuilt it with the USRP. In the demo, the telerobot can be controlled by the PXI teleoperator and the video information can be transmitted to display to the end users.

7 Conclusions

In this paper, we design and implement a SDR-based teleoperating CPS (AVATARS). The AVATARS is flexible and can be deployed quickly to obtain the environment information of emergent and dangerous environments. The detected information is useful to achieve precise perception of the unknown disaster environment. An efficient and reliable wireless link is designed and three strategies are proposed in the link to support reliable wireless communications and real-time wireless teleoperations. First, the packets pipelining and accumulative acknowledgement strategy is employed to improve the communication efficiency. Second, an adaptive channel switching mechanism is adopted to mitigate channel interferences and enhance the effective throughput. Finally, we integrate three types of control information into a single ACK packet to reduce the communication overhead. A prototype system is developed to evaluate the performance of the proposed strategies. Different results are obtained through many experiments, and the results demonstrate the efficiency and benefits of our proposed mechanism. Convincing performance and vivid demonstration of the AVATARS are presented through a video demo.

Endnotes

[1] http://youtu.be/kGrZHHHfwaY?list=UUlaPNFMocZmqY9Wa-7SBeCA

[2] http://v.youku.com/v_show/id_XNzgxNzM2NTYw.html

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The work was partially supported by NSF of China under U1405251, 61273181, 61221003, 61290322, 61301223 and 61174127, by Ministry of Education of China under NCET-13-0358 and 20110073120025, by Science and Technology Commission of Shanghai Municipality (STCSM), China under 13QA1401900, 13Z111050008, by Research Found for the Doctoral Program of Higher Education under 13Z102090106.

Received: 27 November 2015 Accepted: 22 December 2015
Published online: 08 January 2016

References

1. J Sztipanovits, X Koutsoukos, G Karsai, N Kottenstette, P Antsaklis, V Gupta, B Goodwine, J Baras, S Wang, Toward a science of cyber–physical system integration. Proceedings of the IEEE. **100**(1), 29–44 (2012)
2. T Takahashi, S Tadokoro, Working with robots in disasters. IEEE Robot. Autom. Mag. **9**(3), 34–39 (2002)
3. RR Murphy, S Tadokoro, D Nardi, A Jacoff, P Fiorini, H Choset, AM Erkmen, in *Springer Handbook of Robotics*. Search and rescue robotics (Springer, 2008), pp. 1151–1173
4. RR Murphy, Trial by fire rescue robots. IEEE Robot. Autom. Mag. **11**(3), 50–61 (2004). IEEE
5. T Schmid, O Sekkat, MB Srivastava, in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. An experimental study of network performance impact of increased latency in software defined radios (ACM, Canada, 2007), pp. 59–66
6. A Puschmann, MA Kalil, A Mitschele-Thiel, in *Proceedings of the 4th International Conference on Cognitive Radio and Advanced Spectrum Management*. Implementation and evaluation of a practical SDR testbed (ACM, Barcelona, Spain, 2011), pp. 15–19
7. National Instruments' Universal Software Radio Peripheral (NI USRP) (Nov. 2013). http://www.ni.com/usrp
8. National Instruments' PCI Extensions for Instrumentaion (NI PXI) (Nov. 2013). http://www.ni.com/pxi
9. RR Murphy, Trial by fire rescue robots. IEEE Robot. Autom. Mag. **11**(3), 50–61 (2004)
10. RR Murphy, S Stover, Rescue robots for mudslides: a descriptive study of the 2005 La Conchita mudslide response. J. Field Rob. **25**(1-2), 3–16 (2008)
11. RR Murphy, E Steimle, C Griffin, C Cullins, M Hall, K Pratt, Cooperative use of unmanned sea surface and micro aerial vehicles at Hurricane Wilma. J. Field Robot. **25**(3), 164–180 (2008)
12. Y-Y Li, WH Fan, YH Liu, X-P Cai, in *2005 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Teleoperation of robots via the mobile communication networks (IEEE, Hong Kong, China, 2005), pp. 670–675
13. X Xue, SX Yang, MQH Meng, in *2005 IEEE International Conference on Information Acquisition*. Remote sensing and teleoperation of a mobile robot via the internet (IEEE, Hong Kong, China, 2005), pp. 6–11
14. FK Jondral, Software-defined radio: basics and evolution to cognitive radio. EURASIP J. Wirel. Commun. Netw. **2005**(3), 275–283 (2005)
15. G Baldini, T Sturman, A Dalode, A Kropp, C Sacchi, An emergency communication system based on software-defined radio. EURASIP J. Wirel. Commun. Netw. **2014**(1), 169 (2014)
16. X Zhang, KG Shin, E-mili: energy-minimizing idle listening in wireless networks. IEEE Trans. Mob. Comput. **11**(9), 1441–1454 (2012)
17. J Ansari, X Zhang, A Achtzehn, M Petrova, P Mahonen, in *2011 IEEE Wireless Communications and Networking Conference (WCNC)*. A flexible MAC development framework for cognitive radio systems (IEEE, Cancun, Quintana Roo, 2011), pp. 156–161
18. G Nychis, T Hottelier, Z Yang, S Seshan, P Steenkiste, in *Proceedings of NSDI*. Enabling MAC protocol implementations on software-defined radios, vol. 9 (ACM, Boston, Massachusetts, USA, 2009), pp. 91–105
19. P Di Francesco, S McGettrick, UK Anyanwu, JC O'Sullivan, AB MacKenzie, LA DaSilva, A split mac approach for sdr platforms. IEEE Trans. Comput. **64**(4), 912–924 (2015)
20. WARP: Wireless Open Access Research Platform (Jun. 2014). http://warp.rice.edu/trac
21. Small Form Factor SDR Development Platform (Jul. 2014). http://www.xilinx.com/products/devkits/SFF-SDR-DP.htm
22. A Puschmann, P Di Francesco, MA Kalil, LA DaSilva, A Mitschele-Thiel, in *Proceedings of the 8th ACM international workshop on Wireless network test beds, experimental evaluation & characterization*. Enhancing the performance of random access MAC protocols for low-cost SDRs (ACM, Miami, USA, 2013), pp. 9–16
23. A Puschmann, MA Kalil, in *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS)*. The impact of a dedicated sensing engine on a sdr implementation of the csma protocol (ACM, Ilmenau, Germany, 2013), pp. 1–5
24. A Sikora, VF Groza, in *Proceedings of Instrumentation and Measurement Technology Conference*. Coexistence of IEEE 802. 15.4 with other systems in the 2.4 GHz-ISM-Band (IEEE, Ottawa, Canada, 2005), pp. 1786–1791
25. EG Villegas, E Lopez-Aguilera, R Vidal, J Paradells, in *2nd International Conference on CrownCom*. Effect of adjacent-channel interference in IEEE 802.11 WLANs (IEEE, Orlando, USA, 2007), pp. 118–125
26. JC Haartsen, The bluetooth radio system. IEEE Pers. Commun. **7**(1), 28–36 (2000)
27. J Song, S Han, AK Mok, D Chen, M Lucas, M Nixon, in *Proceedings of Real-Time and Embedded Technology and Applications Symposium*. WirelessHART: Applying wireless technology in real-time industrial process control (IEEE, St. Louis, MO, 2008), pp. 377–386

28. C Won, J Youn, H Ali, H Sharif, J Deogun, Adaptive radio channel allocation for supporting coexistence of 802.15. 4 and 802.11 b. IEEE Veh. Technol. Conference. **62**(4), 2522–2526 (2005)
29. R Musaloiu-E, A Terzis, Minimizing the effect of WiFi interference in 802.15. 4 wireless sensor networks. International Journal of Sensor Networks. **3**(1), 43–54 (2008)
30. S-U Yoon, R Murawski, E Ekici, S Park, ZH Mir, in *2010 IEEE International Conference on Communications (ICC)*. Adaptive channel hopping for interference robust wireless sensor networks, (Cape Town, South Africa, 2010), pp. 1–5
31. P Du, G Roussos, in *2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST)*. Adaptive channel hopping for wireless sensor networks, (Shanghai, China, 2011), pp. 19–23
32. M Heusse, F Rousseau, G Berger-Sabbatel, A Duda, in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Performance anomaly of 802.11 b, vol. 2 (IEEE, San Francisco, USA, 2003), pp. 836–843
33. D Shen, Z Pan, K-K Wong, VOK Li, in *Proceedings of Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*. Effective throughput: A unified benchmark for pilot-aided OFDM/SDMA wireless communication systems (IEEE, San Francisco, USA, 2003), pp. 1603–1613
34. S Choudhury, JD Gibson, in *IEEE 63rd Vehicular Technology Conference*. Payload length and rate adaptation for throughput optimization in wireless LANs, vol. 5 (IEEE, Melbourne, Australia, 2006), pp. 2444–2448

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Immediate publication on acceptance
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com