Reliable UDP over the air transfer in digital radio system

Juan Zhang*, Hesong Jiang, Hong Jiang, Chunmei Chen

The Open Fund of Robot Technology Used for Special Environment Key Laboratory of Sichuan Province , School of Information Engineering, Southwest University of Science and Technology, Mianyang, China

Received 5 May 2014, www.tsi.lv

Abstract

Reliability is very important in digital radio point-to-point transmission system, especially for bulk data transfer in narrow band channel. Currently most applications are based on raw UDP service, which does not guarantee the reliability, and existing reliable UDP transfer protocols do not satisfy the performance expectations. The article presents R2UDP(Reduced Reliable UDP) over the air transfer suitable to radio system, SBACK (selective bundled ACK) and smart probe improves the transfer efficiency and saves the bandwidth, and also minimizes the impact of bulk data transfer to other traffic on the shared channel.

Keywords: R^2UDP, bundled ACK, smart probe, digital radio, narrow band channel

1 Introduction

In digital radio system, some data application needs to transfer bulk data over the air in low speed underlying channel. This application does not occur often to warrant having a dedicated reversion channel. And it has the lowest priority and shares channel resource with other traffic loading. As a result, it will take a long duration for whole transfer and occupy the bandwidth for a long time, not only because the channel speed is low and packet loss probability is high, but also the transfer is interrupted often by other higher priority traffic occurred on the shared channel.

Currently methods of the data transmission mainly has TCP (transmission control protocol), SCTP (Stream Control Transmission Protocol), and the UDP (user datagram protocol). TCP and SCTP protocols are connection oriented, ensure the reliability of the data, but the processing is complex, efficiency is not high, occupy more resources, unable to support the massive concurrent connections; UDP protocol adopted for non-connection transmission, fast speed, high efficiency, and can support massive concurrent connections, but there are many shortcomings of poor reliability [1-2]. Through the comparison, the UDP protocol is a more appropriate choice to further improve the speed of data transmission.

To study and improve the reliability of UDP protocol has become a hot issue in current. The literature [3, 4] proposed the RUDP protocol for the environment of a large number of communication terminal frequently sending small size message to the dedicated server. The protocol is very similar with the TCP timeout mechanism. Although compared with TCP, the protocol is much simplified, the kinds of delay is not suitable for fast data transmission of wireless narrow-band system. The literature [5-7] proposed the RUDP that sends a message into waiting queue and continues to send the next packet, do not receive confirmation of the message and then send a message. Compared with the literature [3-4], RUDP saves a lot of waiting time. However, it is still to confirm each message and spend a lot of resources. Lack of RUDP based on [5-7], the literature [8-13] put forward the concept of batch confirmation or timing validation BA-RUDP (Bulk Ack-Reliable UDP). The BA-RUDP algorithm is better than the previous RUDP. It has saved a lot of time and resources, but it still exists some problems: the sending end of the sending pointer is only one. Namely, the sending end each received confirmation; the sending pointer need temporarily stop the sending operation if the sending end has the need to retransmit a packet. This process is still a stop waiting process, spend considerable time.

This paper presents R2UDP (Reduced Reliable UDP) protocol for the specific application of digital wireless narrow-band system of point-to-point file transfer. The protocol creatively adds selective BACK (Bundled ACK) and smart probe, improves the transmission efficiency and save the bandwidth, but also reduces the influence of mass data on the shared channel transmission.

2 Problem statement

In digital radio system, some data application needs to transfer bulk data over the air in low speed underlying channel. This application does not occur often to warrant having a dedicated reversion channel. And it has the lowest priority and shares channel resource with other traffic loading.

As a result, it will take a long duration for whole transfer and occupy the bandwidth for a long time, not only because the channel speed is low and packet loss

**Corresponding author* e-mail: zhangjuanr@swust.edu.cn

probability is high, but also the transfer is interrupted often by other higher priority traffic occurred on the shared channel.

For example, as shown in Figure 1, the PCR OTAP (Over The Air Programming) application deployed in remote PC server and subscriber exchanges large size configuration data through the DMR (Digital Mobile Radio) radio system, such as IPSC (IP Site Connect), single site. The system context diagram is illustrated below.

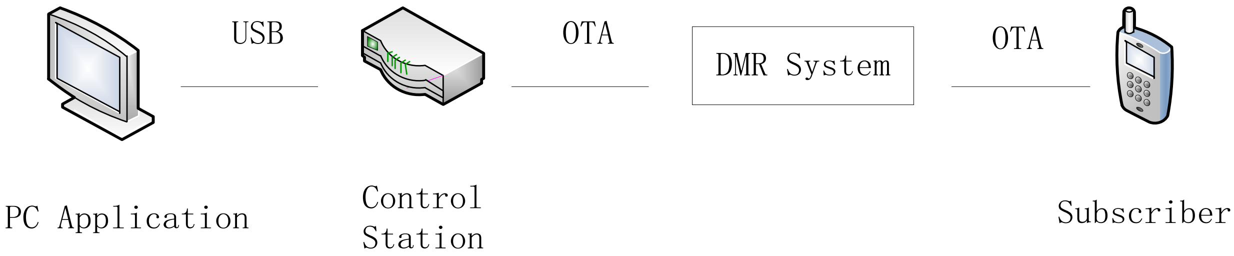


FIGURE 1 System Context Diagram

The PC application and the subscriber application are two end applications for bulk data transfer in the existing radio data application infrastructure. The control station is a L3 device to route the data between the RAN and the wired USB connection. It is transparent to application and does not expect any specific change. The channel in the RAN is low speed with less than 9600 bps data rate, it suffers from poor link quality and also shares with high priority traffic, e.g. voice call. The channel of USB connection is high speed and has good link quality.

In the radio system above, there is a critical parameter of 'time to transfer' between the two ends, and the bottle neck is the performance of the OTA link in the RAN. The 'time to transfer' over the air is impacted by many factors, besides the inherent wireless channel characteristic of high error rate due to RF impairment, the round trip delay varies in different system, and the transmission channel will become unavailable when the subscriber is engaged in voice call, or it is out of RF range, or radio user switches the channel, etc.. All of them worsen the transfer success rate.

In case of bulk data transfer, we are pursuing shorter transfer time with less bandwidth consumption, because we need to minimize the impact of long time channel occupation by this lowest priority data application. If the duration is too long, it will not meet the requirement of the application itself, and also it will impact other traffic transfer occurred on the channel.

To achieve this goal, in upper L7 application layer, the raw bulk data is compressed and fragmented, and the transfer is designed as resilience. It means that the application only needs to transfer those unsuccessful fragments in continuous sessions and does not transfer the data from the scratch, in this way the bandwidth resource is saved and the whole transfer time is shorter. Here the fragment size is balanced by the underlying channel speed and feature interaction, if it is too large, it will delay other higher priority traffic access on this channel, and this will be unacceptable.

Although the L7 handling gets benefits to some extent, the performance is still not desired. The main problem is the efficiency of reliable transfer for each fragment. So an effective transfer mechanism is expected to transport the data over the air faster and consume the bandwidth resource as less as possible.

Current known transfer mechanisms can be found in transport layer protocols, e.g. TCP, UDP and some standard file transfer protocols, e.g. TFTP. Since the UDP does not provide reliable transfer service, it is upper layer application's responsibility to retransmit the failed packet to guarantee the reliability. The UDP based transfer is the direction of over the air transfer in radio system because of its simplicity, lightweight, and some proprietary utility of UDP/IP header compression. The existing UDP based reliable transfer scheme is not desirable because the retransmission strategy is low efficient or aggressively clogs channel, or there is redundancy as it is heavy weight model based which is not suitable to the radio system. In consequence, the channel utilization and system throughput is decreased, and the user interface is hard to be utilized in special application scenario.

Accordingly, there is a need for a reliable UDP transfer, which is lightweight and efficiently transports the bulk data over the air in radio system.

3 R^2UDP mechanism analysis

This paper provides an efficient reliable UDP transfer over the air. It is a balance of efficiency and complexity in radio system. R^2UDP (Reduced Reliable UDP) is referred in following chapters. The security is not covered as it is supposed to be application layer's responsibility. It has the following goals.

1) It will guarantee the success rate of each data UDP packet transfer, but not allow the big size data packet is loaded on the channel too fast to clog the system.
2) It will reduce the reliability overhead as much as possible to decrease the overall transfer time and save the bandwidth, e.g. less cost on acknowledgement traffic and handshake.
3) It will be lightweight and simple in radio system model.
4) It will provide simple and general low level user interface, like socket interface.

3.1 GENERALS

R^2UDP is a ‘thin’ layer based on UDP and completely compatible with UDP packet format to embrace existing proprietary UDP/IP header compression. It attempts to provide only those services necessary, in order to be efficient in operation and small in size. It is efficient because essentially there is small overhead and retransmission intelligence.

R^2UDP is deployed as 2 ends: ‘Fat’ sender and ‘thin’ receiver. One device could be both sender and receiver for full duplex reliable channel. The sender initiates bulk data transmission with UDP packets. It is the key functional part to manage the pace of retransmission. The receiver receives the bulk data and returns ACK to indicate which packets are arrived successfully.

In the example below, the PC device is the data sender that sends bulk data over the air to the receiver of subscriber. As shown in Figure 2.

As shown in Figure 3, there are 2 symmetric data paths provided for full duplex reliable channel as below. Each path consists of bidirectional UDP packets (data and ACK) and is independent on each other. The following chapters only describe one path.

As shown in Figure4, R^2UDP encapsulates the payload data and send it with raw UDP service. The transfer unit of R^2UDP is called as segments as follows.

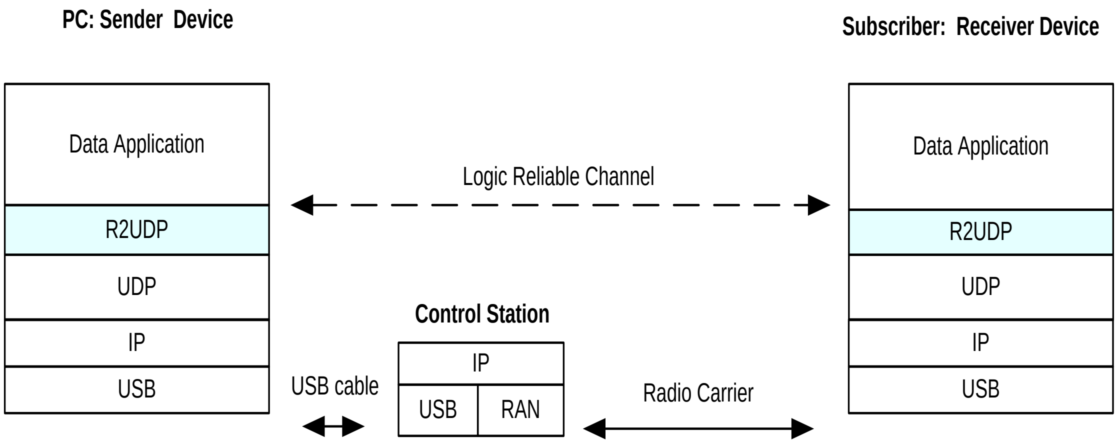


FIGURE 2 Protocol Stack Overview

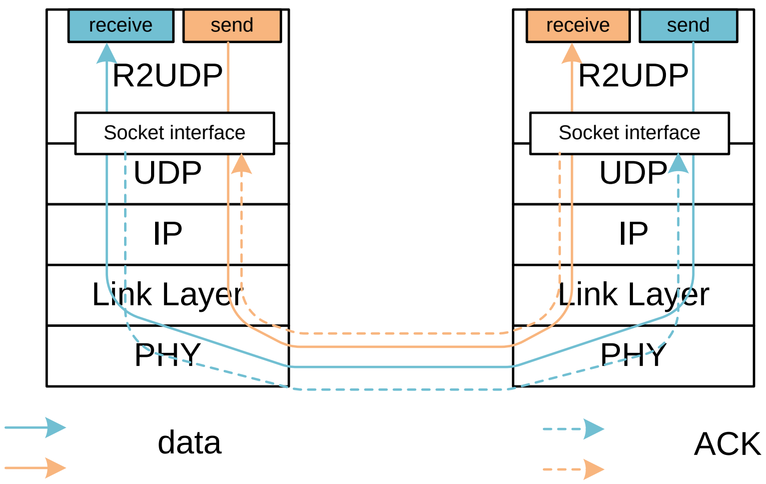


FIGURE 3 Logical Reliable Channels

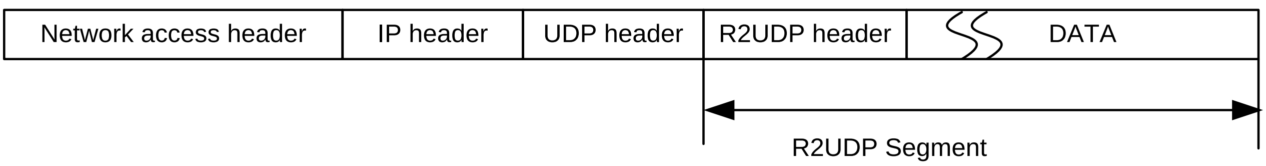


FIGURE 4 Segments in Frame

R^2UDP provides efficient reliable transfer through main mechanisms here.

1) Manage sending buffer with several windows and data blocks. The data block in next window is only transferred when the current window transferring completes. It essentially provides more patience to avoid channel clogging.
2) Bundle acknowledgement for cumulative ACK and selective retransmission. It essentially reduces the traffic and increase the data throughput.
3) Active probe by the sender to pull in the last delivery information. In this way, the sender is capable of pacing the flow with more patience instead of put bulk data aggressively.
4) Reduced header format and handshake overhead.

3.2 RELIABLE COMMUNICATION

3.2.1 Window and Block Number

As shown in Figure 5, a buffer from user will be split into several windows, and each window contains several blocks, which share one ACK. Each window engages one window number to differentiate it with others, and each block engages one block number to identify it among blocks in the window. The initial window number and block number are 0 after data path reset which happens prior to the beginning of first transfer. The window number is increased by 1 each time a window buffer is sent successfully.

Sender is responsible for filling in window and blocks number, and make sure that a window must be successfully received by receiver before a new window starts.

Window number and block number are filled in R2UDP header, which forms a data segment with data block.

The window size can be modified during transfer according to the total data length. If the remained data is less than previous window size, e.g. there is only 4 data blocks, while the previous window size is 5, the sender updates the last window size as 4.E.g. transferring 2K bytes buffer from user, window size = 3, block size = 256 bytes. The window n+1 is only transferred upon completion of the window n.

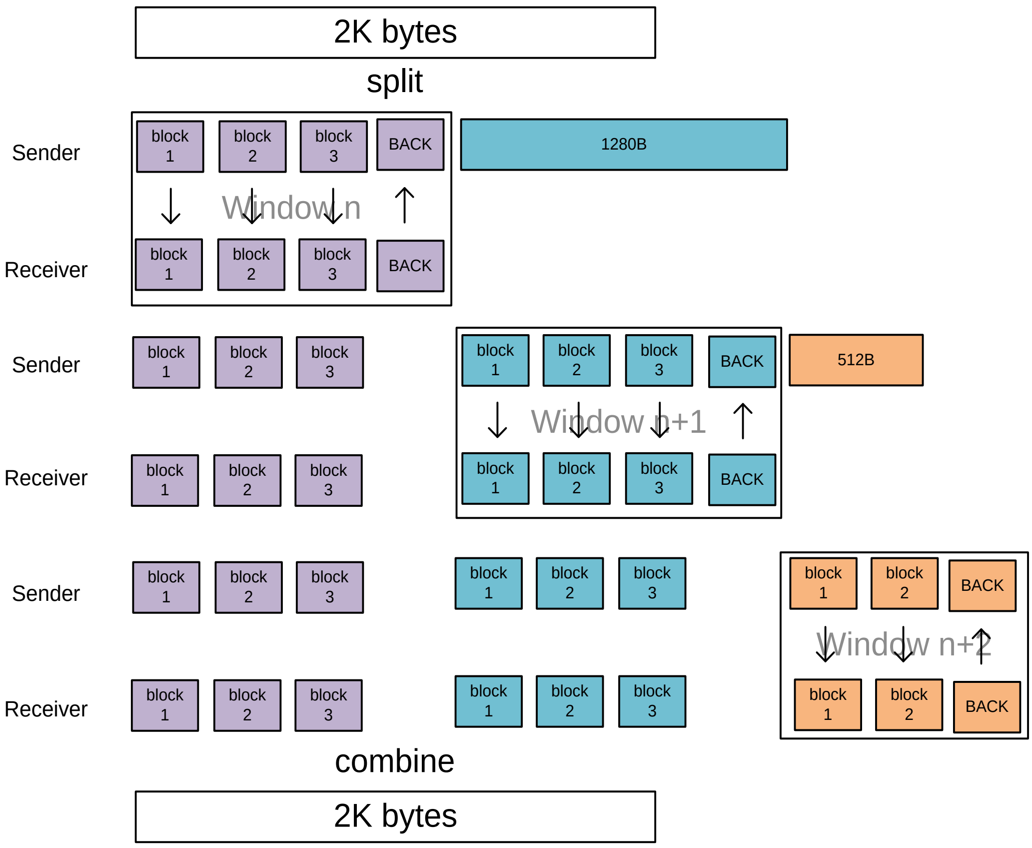


FIGURE 5 Window Management

3.2.2 Bundled Acknowledgement

R^2UDP assumes it has only an unreliable datagram service to deliver segments. To guarantee delivery of segments in this environment, R2UDP engages mechanism of ACK and selective retransmission.

As shown in Figure 6, here the ACK is a bundled acknowledgement for data segments belong to the same window. Receiver will update the received data block number in cumulative BACK when missing segment arrives. The ACK segment will not be acknowledged.

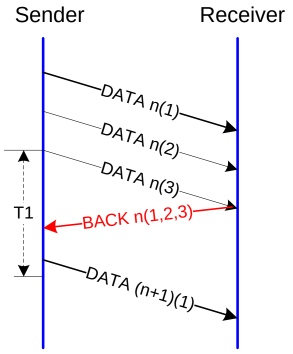


FIGURE 6 Window size = 3, normal case

3.2.3 Active Probe

To detect missing segments, the sender utilizes a retransmission timer for each window transmitted. As shown in Figure 7, the timer could be set according to the single segment transfer time in the network and the amount of segments. When an acknowledgement for a window is received, the timer for that window is cancelled. As shown in Figure8, if the timer for a window expires before an acknowledgement is received, a PROBE segment is transmitted. Receiver will reply a BACK to identify the window number and received data blocks.

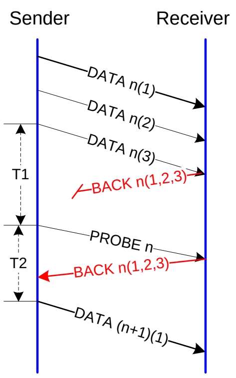


FIGURE 7 Window size = 3, BACK loss

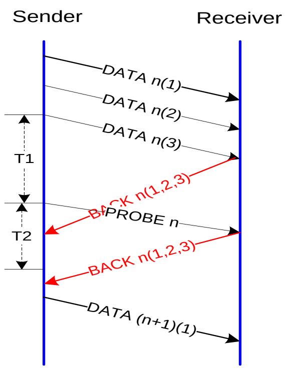


FIGURE 8 Window size = 3, BACK delay

3.2.4 Selective Retransmission

As shown in Figure 9, sender sends out data blocks of a window. Receiver will reply a BACK once all blocks in the window are received. Otherwise if not all blocks are received, receiver will wait for the blocks not arrived, or reply a BACK for blocks arrived on receiving a PROBE. Sender retransmits only those missing data blocks and waits for the positive BACK.

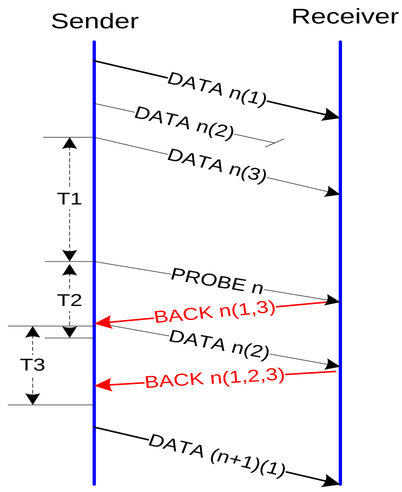


FIGURE 9 Window size = 3, 1 data segment loss

3.2.5 Reset Path

As shown in Figure 10, the data path is to reset before the first transfer. Sender is responsible to send a RST upon reset. On receiving a RST segment, receiver will discard incomplete window, and then reply a RACK to indicate its availability. Receiver will update its window number on receiving a DATA segment.

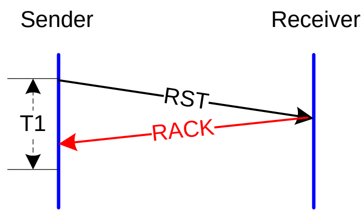


FIGURE 10 Reset a data path

4 Simulation results

We analysed the mechanism and performance of the R2UDP protocol through the Motorola radio to test and NS-2 network simulation.

4.1 TESTING ON VOICE INTERFERENCE

The testing conditions are as follows: the radio is Motorola Mag One A8, the channel bandwidth is

12.5KHZ, and the use of transferring the file is bandwidth is 9600bps, the file is10M between the radios. Comparison of the protocol is TCP, TFTP (the bottom for the UDP protocol), and R2UDP. The total test time is 10 minutes and 5 seconds of the phone every 30 seconds in Table 1. The total test time is 10 minutes and 5 seconds of the phone every 20 seconds in Table 2.

TABLE 1 Interference 30s

Protocol	Data
TCP	4.054M
TFTP	4.163M
R^2UDP	4.769M

TABLE 2 Interference 20s

Protocol	Data
TCP	3.292M
TFTP	3.856M
R^2UDP	4.293M

From Table 1 and Table 2 shows that, even in the frequent case of a voice interrupted, R2UDP showed better transmission performance. TCP and TFTP transmission performance dramatically decrease with frequent access interference.

4.2 SIMULATION ANALYSIS ON NO VOICEINTERFERENCE

Through the NS-2 simulation comparative analysis of TCP, UDP, R2UDP, the specific simulation environment is shown in table 3. Given the error rate of lrate=4%, different bandwidth situation, compare the average throughput as shown in Figure 11. Given the bandwidth of 9.6K, different error rate comparison of throughput is shown in Figure 12.

As shown in Figure 11, compared to Vegas and Reno, R2UDP showed good performance in a higher rate of error (4%), a narrow bandwidth (less than 12K). As shown in Figure 12, compared to Vegas and Reno, R2UDP showed good performance with the increased error rates in a narrowband system (9.6K), while Reno and Vegas respectively in the error rate is 10% and 45% in the case of a throughput of 0.

TABLE 3 Parameter settings

Type	Value
Link delay	10ms
Packet Size	256Bye
Window size	3

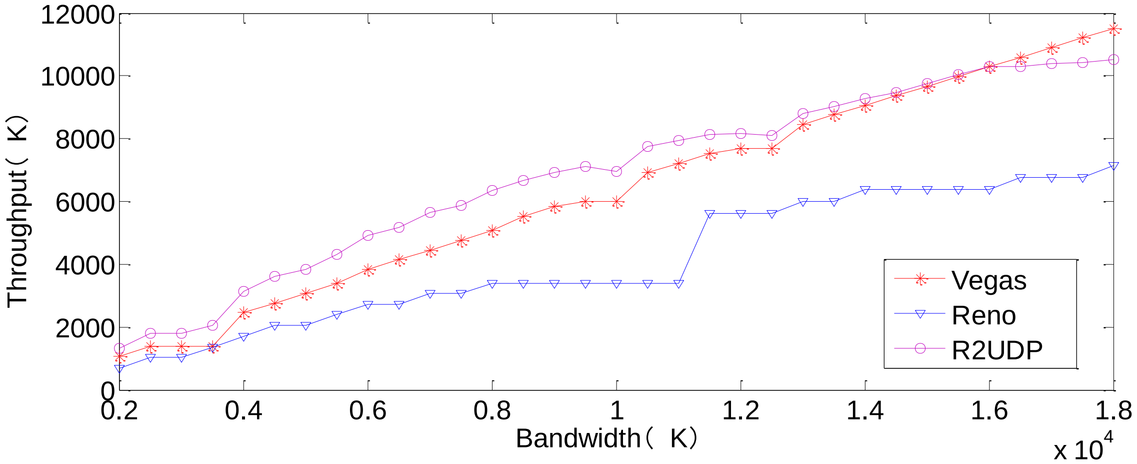


FIGURE 11 Throughput at 4% error rate

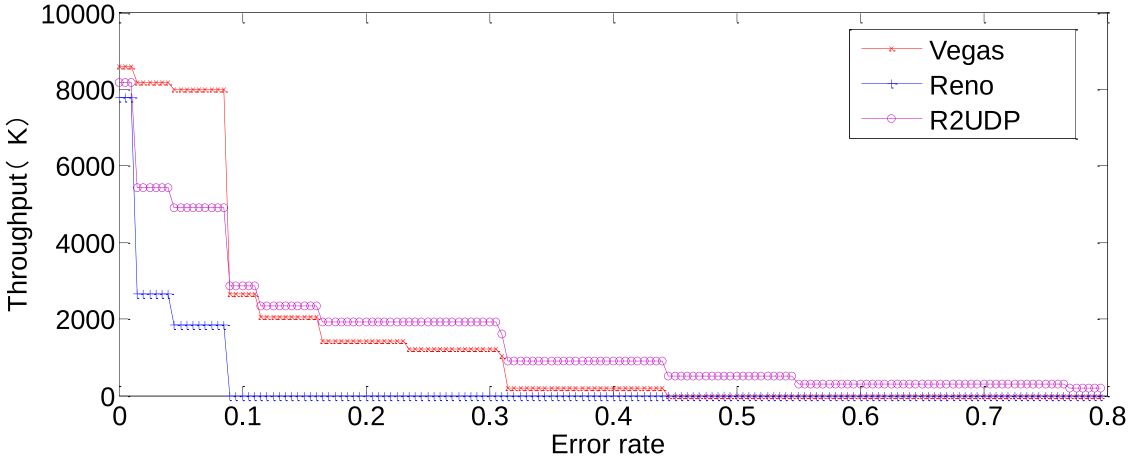


FIGURE 12 Throughput at 9.6K bandwidth

5 Conclusions

This paper designs a simple and reliable R^2UDP protocol in order to adapt to the digital narrowband radio systems. This protocol adds Selective BACK and Smart Probe Frame. According to the measured data of Motorola Digital Radio experiments indicate that: in the case of frequent voice interrupted, R^2UDP showed better transmission performance; and through the analysis of NS2 network simulation proves that, even in a higher rate of error, the narrow bandwidth cases still showed a better performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NO.61072138, No. 61379005), and Southwest University of Science and Technology (12zx7127).

Appendix Examples of operation

A1 No Segment Loss and Delay

In this case, the receiver receives all segments of the window and replies a BACK in time. It is the simplest case.

```
Time   Sender                                Receiver
     +----+---+------+---+-----~~-----+
1    |DATA| 2 |  n   | 0 |  Payload   | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
2    |DATA| 2 |  n   | 1 |  Payload   | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
3    |DATA| 2 |  n   | 2 |  Payload   | -->
     +----+---+------+---+-----~~-----+

                         +----+---+------+---+--------+
4                    <-- |BACK| 2 |  n   | 0 |00000111|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
5    |DATA| 2 | n+1  | 0 |  Payload   | -->(next window
starts)
     +----+---+------+---+-----~~-----+
```

A2 Data Segments Loss

In this case, the receiver replies a BACK indicate which segments are received, and then the sender retransmits those lost segments only.

```
Time   Sender                                Receiver
     +----+---+------+---+-----~~-----+
1    |DATA| 2 |  n   | 0 |  Payload   | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
2    |DATA| 2 |  n   | 1 |  Payload   | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
3    |DATA| 2 |  n   | 2 |  Payload   | -->    Lost
     +----+---+------+---+-----~~-----+
        4    Wait because not all segments of window 1
             are received
5    Time out
     +----+---+-+
6    |PROB| 2 |1| -->
     +----+---+-+
                         +----+---+------+---+--------+
7                    <-- |BACK| 2 |  n   | 0 |00000011|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
8    |DATA| 2 |  n   | 2 |  Payload   | -->
     +----+---+------+---+-----~~-----+
                         +----+---+------+---+--------+
9                    <-- |BACK| 2 |  n   | 0 |00000111|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
10   |DATA| 2 | n+1  | 0 |  Payload   | -->(next window
starts)
     +----+---+------+---+-----~~-----+
```

A3 BACK Segments Loss

In this case, the receiver replies a BACK indicate, which segments are received, and then the sender retransmits those lost segments only.

```
Time   Sender                                Receiver
     +----+---+------+---+-----~~-----+
1    |DATA| 2 |  n   | 0 |  Payload   | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
2    |DATA| 2 |  n   | 1 |  Payload   | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
3    |DATA| 2 |  n   | 2 |  Payload   | -->
     +----+---+------+---+-----~~-----+
                         +----+---+------+---+--------+
5               Lost <-- |BACK| 2 |  n   | 0 |00000111|
                         +----+---+------+---+--------+
6    Time out
     +----+---+-+
7    |PROB| 2 |0| -->
     +----+---+-+
                         +----+---+------+---+--------+
8                    <-- |BACK| 2 |  n   | 0 |00000111|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
9    |DATA| 2 | n+1  | 0 |  Payload   | --> (next window
starts)
     +----+---+------+---+-----~~-----+
```

A4 Communication Over Long Delay Path

In this case, the sender sends a PROBE segment to query the result of last transfer when timeout.

```
Time   Sender                                Receiver
```

```
     +----+---+------+---+-----~~-----+
1    |DATA| 2 |  n  | 0 |  Payload  | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
2    |DATA| 2 |  n  | 1 |  Payload  | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
3    |DATA| 2 |  n  | 2 |  Payload  | -->
     +----+---+------+---+-----~~-----+
                         +----+---+------+---+--------+
4            [Long Delay]   <-- |BACK| 2 |   n  | 0
|00000111|
                         +----+---+------+---+--------+
5    Time out
     +----+---+-+
6    |PROB| 2 |1| -->
     +----+---+-+
                         +----+---+------+---+--------+
7                        <-- |BACK| 2 |   n  | 0 |00000111|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
8    |DATA| 2 |  n+1 | 0 |   Payload  | --> (next window
starts)
     +----+---+------+---+-----~~-----+
```

The BACK at ‘7’ will be ignored by sender because it duplicates with ‘4’.

A5 Communication over Long Delay Path with Lost DATA Segments

In this case, the sender sends a PROBE segment to query the result of last transfer when timeout. After the BACK is received, it retransmits those lost segments according to the BACK.

```
Time     Sender                                    Receiver
     +----+---+------+---+-----~~-----+
1    |DATA| 2 |  n  | 0 |  Payload  | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
2    |DATA| 2 |  n  | 1 |  Payload  | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
3    |DATA| 2 |  n  | 2 |  Payload  | -->  Lost
     +----+---+------+---+-----~~-----+
         4    Wait because not all segments are received
5    Time out
     +----+---+-+
6    |PROB| 2 |0| -->
     +----+---+-+
                         +----+---+------+---+--------+
7                        <-- |BACK| 2 |   n  | 0 |00000111|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
8    |DATA| 2 |  n  | 2 |  Payload  | -->
     +----+---+------+---+-----~~-----+
                         +----+---+------+---+--------+
9            [Long Delay]   <-- |BACK| 2 |   n  | 0
|00000111|
                         +----+---+------+---+--------+
10   Time out
     +----+---+-+
11   |PROB| 2 |0| -->
     +----+---+-+
                         +----+---+------+---+--------+
12                       <-- |BACK| 2 |   n  | 0 |00000111|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
13   |DATA| 2 |  n+1 | 0 |   Payload  | --> (next window
starts)
     +----+---+------+---+-----~~-----+
```

The BACK at ‘12’ will be ignored by sender because it duplicates with ‘9’.

A6 Communication over Long Delay Path with Lost BACK Segments

In this case, the sender sends a PROBE segment to query the result of last transfer when timeout. After the BACK is received, it retransmits those lost segments according to the BACK.

```
Time     Sender                                    Receiver
     +----+---+------+---+-----~~-----+
1    |DATA| 2 |  n  | 0 |  Payload  | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
2    |DATA| 2 |  n  | 1 |  Payload  | -->
     +----+---+------+---+-----~~-----+
     +----+---+------+---+-----~~-----+
3    |DATA| 2 |  n  | 2 |  Payload  | -->
     +----+---+------+---+-----~~-----+
                         +----+---+------+---+--------+
4                Lost    <-- |BACK| 2 |   n  | 0 |00000111|
                         +----+---+------+---+--------+
5    Time out
     +----+---+-+
6    |PROB| 2 |1| -->
     +----+---+-+
                         +----+---+------+---+--------+
7            [Long Delay]   <-- |BACK| 2 |   n  | 0
|00000111|
                         +----+---+------+---+--------+
8    Time out
     +----+---+-+
9    |PROB| 2 |1| -->
     +----+---+-+
                         +----+---+------+---+--------+
10                       <-- |BACK| 2 |   n  | 0 |00000111|
                         +----+---+------+---+--------+
     +----+---+------+---+-----~~-----+
11   |DATA| 2 |  n+1 | 0 |   Payload  | --> (next window
starts)
     +----+---+------+---+-----~~-----+
```

The BACK at ‘10’ will be ignored by sender because it duplicates with ‘7’.

A7 Sender Reset

In this case, when the sender is reset, it sends a RST to reset the data path. Then the next window number will reset to 1.

```
Time    Sender                              Receiver
       +----+---+-+
1      | RST| 0 |0| -->
       +----+---+-+
                                           +----+---+-+
2                                      <-- |RACK| 0 |0|
                                           +----+---+-+
       +----+---+------+---+-----~~-----+
3      |DATA| 2 |  n   | 0 |  Payload   | --> (DATA starts)
       +----+---+------+---+-----~~-----+
```

References

[1] Partridge C, Pink S 1993 *IEEE/ACM Transactions on Networking* **1**(4) 429 - 40

[2] Zhang Yong-qiang, Gao Hong-bin 2010 Design and implementation of RUDP protocol for multiple mobile agent communication *International Conference on Computer Application and System Modeling (ICCASM), 22-24 Oct. (2010)* **8** 614-8

[3] Wang Jigang, Gu Guochang, Xie Shibo, Xu Lifeng 2006 Reliable and Efficient Data Transfer Protocol Based on UDP in Cluster System *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences* **1** 518-24

[4] Doan Thanh Tran, Eunmi Choi 2007 A reliable UDP for ubiquitous communication environments *Proceedings of the 2007 Annual Conference on International Conference on Computer Engineering and Applications* 1-6

[5] Saini T K, Kumar S, Dhaka M K 2014 Analysis of routing protocols using UDP traffic under dynamic network topology.*2014 IEEE International Advance Computing Conference (IACC),21-22 Feb 2014* 160-165

[6] Fang Jiaoli, Liu Ming 2011 Design and Implementation of Embedded RUDP *The 2nd International Conference on Networking and Distributed Computing (ICNDC), 21-24 Sep 2011* 7- 9

[7] Alrabaee S 2014 Concordia Inst. Using model checking for Trivial File Transfer Protocol validation *2014 International Conference on Communications and Networking ComNet 19-22 March 2014*, 1-7

[8] Ono F, Takizawa K, Tsuji H, Lin Shan, Kagawa T 2014 Measurement of TCP and UDP performance over UAS relay networks *2014 International Conference on Unmanned Aircraft Systems (ICUAS) 27-30 May(2014)* 389-94

[9] Manabe J, Funasaka J, Ishida K 2012 An improved UDP protocol for video transmission over Internet-to-wireless networks *The 3rd International Conference Networking and Computing (ICNC) 5-7 Dec 2012* 162 – 8

[10] Lam P P -K, Liew S 2004 CUDP-Liter: an improved UDP protocol for real-time multimedia applications over wireless links *International Symposium on Wireless Communication Systems, 20-22 Sep 2004* 314-8

[11] Finamore A, Mellia M, Meo M, Rossi D 2010 *IEEE/ACM Transactions on Networking* **18**(5) 1505 –15

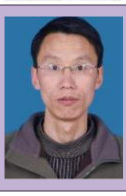
Authors

Juan Zhang, born in 1982,Mianyang, China

Current position, grades: associate professor of School of Information Engineering, Southwest University of Science and Technology, PhD
University study: Doctor of Engineering in Signal and Information Processing from University of Chinese Academy of Sciences (2012). Now she is.
Research interests: cognitive radio and intelligent learning.

Hesong Jiang, born in 1982,Mianyang, China

Current position, grades: lecturer of School of Information Engineering, Southwest University of Science and Technology
University study: Master of Engineering in Signal and Information Processing from University of Chinese Academy of Sciences (2009)
Research interests: cognitive radio and intelligent learning.



Hong Jiang, born in 1969, Mianyang, China

Current position, grades: The full professor of School of Information Engineering, Southwest University of Science and Technology
University study: Doctor of Engineering in Communications professional from University of Electronic School of Information Engineering (2004)
Research interests: cognitive radio and intelligent learning.

Chenmei Chen, born in 1977, Mianyang, China

Current position, grades: associate professor of School of Information Engineering, Southwest University of Science and Technology
University study: She received her Master of Engineering in Signal and Information Processing from University of Chinese Academy of Sciences (2008)
Research interests: cognitive radio and intelligent learning