Design and Implementation of Real-Time Video Transmission on Ad Hoc Network

Jian He[1], Shuo Shi[1,2(✉)], Zhong Zheng[3], and Cong Zhou[1]

[1] School of Electronic and Information Engineering,
Harbin Institute of Technology, Harbin 150001, Heilongjiang, China
crcss@hit.edu.com
[2] Peng Cheng Laboratory, Network Communication Research Centre, Shenzhen 518052, Guangdong, China
[3] International Innovation Institute of HIT in Huizhou, Huizhou 516000, Guangdong, China

Abstract. Video transmission technology has been widely used in all aspects of life, but the video transmission that we usually see is transmitted on a network with fixed base station support, which is not suitable for earthquakes, fire rescue, and detection without fixed Scenes such as areas covered by base station signals. This article uses Raspberry Pi 3B+ equipped with wireless Ad-Hoc network to realize the function of real-time video transmission, and realizes the push flow control of the central control terminal to the video capture end, which proves the feasibility and practicality of real-time video transmission in the wireless Ad-Hoc network. This article runs the Ubuntu 16.04 system on the Raspberry Pi 3B+, with external physical transmission equipment and cameras, a large number of audio and video related libraries are installed as a dependency of real-time video transmission, and a high-performance SRS streaming media server is deployed on the central control end. Each network node has installed the OLSR routing protocol and carried out related network configuration.

Keywords: Ad-Hoc · Real-time video transmission · SRS streaming server

1 Introduction

In areas that don't have base station, it is more difficult to communicate. Since the end of the 20th century, the development of wireless Ad-Hoc network has solved this demand of people. However, with the improvement of people's quality of life and demand, there is an urgent need for a technology to solve the problem that video cannot be transmitted in areas without base station for signal coverage.

The Ad-Hoc has become a kind of dedicated network with high research value due to its features such as not relying on ground infrastructure, strong node mobility, and nodes in the network that can both forward tasks and actively publish tasks. It is mainly used to provide communication capabilities between nodes in special scenarios such as earthquakes, combat, and detection [1].

S. Shi et al. (Eds.): 6GN 2021, LNICST 439, pp. 51–60, 2022.
https://doi.org/10.1007/978-3-031-04245-4_5

The Ad-Hoc network has the characteristics of high flexibility, low cost, and strong integration. It is one of the key technologies in forest fire prevention, agricultural pest detection, remote identification of objects, and emergency rescue. It is obtained in many neighborhoods and has a high degree of attention and has great advantages when performing tasks such as image acquisition and video transmission. However, at the same time, it is limited by the limited network bandwidth of the Ad-Hoc network and the link is easily interrupted [2]. When the video is transmitted on the Ad-Hoc network, The quality of the video will be affected badly, and the performance degradation will be more obvious when the number of network hops increases [3].

(1) Non-centrality
Each node in the Ad-Hoc network is equal to each other. As long as the relevant network configuration information matches, each node can enter and exit the network by itself. The failure of each node will only change the structure of the network, and will not cause The entire Ad-Hoc network was devastated. Therefore, the wireless mobile Ad-Hoc network does not have a network center.

(2) High dynamic topology
In a mobile wireless Ad-Hoc network, in addition to being free to join and exit the network at any time, each node can move and change its geographic location at will, and each node's signal transmission.
power and received signal gain compensation, channel fading, etc. The factors are unpredictable, which will cause the change in the topology of the Ad-Hoc network to become very rapid and difficult to estimate [4]. Therefore, in the face of rapidly changing Ad-Hoc network topology, it is necessary to design or select a suitable routing protocol to adapt to different scenarios.

(3) Multi-hop data transmission
Because the nodes in the Ad-Hoc network may be out of the communication range of one node due to the low signal transmission power, or the MAC address of the node is blocked by another node for various reasons, this will cause two nodes No direct communication between them. However, each node in the Ad-Hoc network can forward data [5]. Therefore, the node of the wireless mobile Ad-Hoc network can communicate with nodes outside the communication range of its own through the forwarding mechanism of the intermediate node, which means that the Ad-Hoc network has the multi-hop nature of data transmission.

(4) Limitations of nodes
The hardware implementation of the Ad-Hoc network needs to consider many factors. Generally speaking, the mobile terminal of the Ad-Hoc network needs to meet the conditions of portability, which will lead to the limitation of hardware functions [6].When designing a wireless Ad-Hoc network node, it is necessary to comprehensively consider multiple factors, including the weight, volume, endurance, communication range, and information processing speed of the node.

(5) Low security
In the Ad-Hoc network that does not use encryption technology, its nodes can join and exit at will. The information in the wireless Ad-Hoc network communication is

also easy to be intercepted and eavesdropped, thereby leaking the relevant information of the network; similarly, unfamiliar nodes can also easily join the established wireless Ad-Hoc network and disguise it, so Ad-Hoc Poor network security [7].

2 Video Transmission Scheme

2.1 Hardware Scheme

On the premise of ensuring certain performance and considering the convenience of Ad-Hoc network nodes, this article will choose Raspberry Pi 3B+ as the hardware platform to connect the camera and a large network card with higher performance to complete the function of Ad-Hoc network video transmission.

Raspberry Pi is a kind of micro-controller, which is equivalent to a reduced computer. The Raspberry Pi 3B+ is lighter in weight and smaller in size, making it easy to carry, and it can even be equipped with an aircraft to build a flying Ad-Hoc network in the air. At the same time, the performance of Raspberry Pi 3B+ is superior, and the processing speed it provides is sufficient to meet most needs.

The Raspberry Pi 3B+ provides a Micro-USB power interface, an HDMI interface,4 USB2.0 interfaces, 1 RJ45 interface and a camera interface, and the working frequency is 1.4 GHz. In this article, you can choose a USB2.0 camera to connect to the Raspberry Pi 3B+ to collect camera video data; select a signal transmission device to connect to the Raspberry Pi to send and receive data and build a wireless Ad-Hoc network.

Each wireless transmission device has its own address. Enter the address of the transmission device in the browser to enter the configuration interface. Before building a wireless Ad-Hoc network, the frequency of each transmission device must be set in the same frequency band; at the same time, in order to carry a specific routing protocol, the mesh function of the transmission device needs to be turned off. The transmission equipment used in this article works in Ad-Hoc mode by default, so there is no need to perform additional configuration on the network card, only the connection is normal, and the routing protocol suitable for Ad-Hoc networks can be used to complete the networking.

2.2 Software Scheme

The function of real-time video transmission in the Ad-Hoc network designed in this paper is mainly divided into two parts: the video capture terminal and the control terminal.

The video capture terminal needs to call the camera to capture video data, convert the video data into a video stream format and send it to the streaming media server; at the same time, the capture terminal sets aside a port to receive commands from the control terminal and can give a response to the received instructions.

Running the required routing protocol under the Ubuntu system can build a stable and efficient Ad- Hoc network to improve the quality of video transmission in the Ad-Hoc network. The commonly used routing protocols for wireless Ad-Hoc networks include AODV and OLSR [8]. This article uses the olsrd-0.6.8 version of the protocol to build the required wireless Ad-Hoc network. Of course, the “olsrd” protocol version

can also be updated or old as needed Version, the different version selection will not substantially affect the function of video transmission. The OLSR routing protocol is a representative of a priori routing protocol in the Ad-Hoc network. Under the rules of OLSR, each network node in the Ad-Hoc network will exchange packet information with its neighboring network nodes in a certain period. So as to realize the awareness of the link state, and then create its own topology. Each node in the network uses this topology to send and receive packet information, and then obtain the topology information of the entire Ad-Hoc network. Each network node can use the topology information of the Ad-Hoc network to calculate according to a certain algorithm The routing table from this node to every other node in the Ad-Hoc network.

The negative impact of the low route discovery and establishment delay is the increase in network overhead. Each node must periodically maintain its own routing information, which will make the entire network use too many resources, especially when the number of nodes in the Ad-Hoc network is increasing, it will produce unbearable network overhead. Therefore, when building a wireless Ad-Hoc network, this article uses only a few Raspberry Pi 3B+ as network nodes, and uses a flat structure to simplify the process of building a Ad-Hoc network, reduce the CPU occupancy rate, and better realize the function of video transmission.

Ordinary servers cannot meet the requirements of this design. In order to achieve video streaming, a streaming media server needs to be built to assist in real-time transmission of multi-hop video in a Ad-Hoc network. Streaming media technology can encode and compress continuous audio and video data and store it on a streaming media server. When the client is watching a media file, it can watch it without downloading the entire multimedia file. When using a streaming media server for video transmission, the basic logic diagram is as shown in the figure Fig. 1.

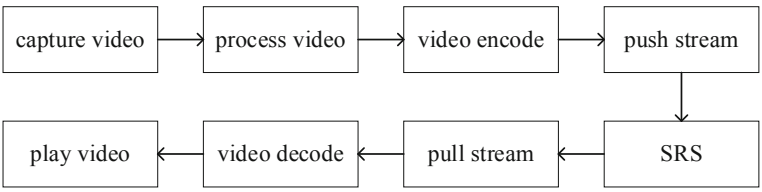

Fig. 1. Server logic diagram

First, the camera video information is collected on the collection side, and then the video is processed (such as adding watermarks, filters, etc.), and then sent to the encoder for video encoding; after the encoding is completed, the stream is pushed, and the real-time video stream is pushed to The streaming media server built. At the central control end, the multimedia stream can be obtained from the streaming media server. At this time, the video stream is encoded, so it needs to be decoded. After decoding, you can obtain the bare video stream that can be played directly. The naked video stream is sent to the player for playback and display, and the function of real-time video transmission can be completed. When the collecting end is pushing the stream, it will first send a request for room creation to the intermediate service server, and then the service server will send a live stream creation request to the streaming server, and the streaming server

will send a response to the service server after receiving the request. After that, the service server sends an address for live streaming to the video capture terminal, and the capture terminal pushes the stream to the address according to the live streaming address received in the response, and the entire process of streaming is completed.

The RTMP protocol at the application layer is needed to transmit audio and video streams to improve the quality of audio and video data information transmitted on the network. The RTMP protocol is created on the underlying transport layer protocol.

Generally speaking, the RTMP protocol will establish a connection to the application layer through a handshake based on the establishment of a connection at the TCP protocol layer. On the RTMP-based link, some data used for control will be transmitted, which can be used to transmit actual multimedia data, as well as some other command data for controlling these multimedia data, and so on. When the RTMP protocol is used for data transmission, the transmitted data will be converted into its own format, which is called the message of this protocol, but in fact, when the RTMP protocol transmits these messages, these messages will continue to be divided It is a data block to better realize the splitting of data packets, multiplexing and ensuring the fairness of information. Each data block has a message ID, and each data block has a probability of being a complete message or a part of the message. If the transmitted data block is not a complete message, the data receiving end is also It reads the information in the data block, obtains the data length and message of the data block, and restores the data block belonging to the same ID into a complete message.

3 Realization Process

3.1 Raspberry Pi Platform Construction

Connect a card reader to the Raspberry Pi to install the Ubuntu 16.04 operating system; after installing the operating system, use instructions to update some components of the operating system. In order not to be limited to only transmitting media video files and to increase the function of transmitting real-time video, it is also necessary to configure a camera. Connect the camera to the Raspberry Pi 3B+. When calling the camera, you need to find the folder location where the camera is located. You can use instructions to complete the function of finding the camera. The result may be "/dev/video0".

After the camera is configured, you need to configure the wireless network card. Connect the prepared transmission device to the power adapter to supply power, and use the network cable to connect the Raspberry Pi 3B+ to the transmission device. There is the initial IP address of the network card on the back of the transmission device. Generally, this address does not need to be modified. You can use the ifconfig command in the Raspberry Pi terminal to assign an IP address to each network node. Open the terminal on the Raspberry Pi 3B+ and use the ping command to check whether the wireless transmission device can be pinged (the command is ping 10.10.10.**). If the ping is successful, the wireless transmission device has been successfully connected to the Raspberry Pi.

Next, you need to install a lot of libraries. The installed video library is mainly used to serve "ffmpeg" instructions. The installation video library is mostly installed in the form of a static library, the purpose is to make the compiled file and the static library

file link into an executable file, so that the executable file does not depend on the library file afterwards, and enhance the portability of the program. First, you need to install the two required assembly libraries NASM and YASM, and then you can download and install the following libraries: libx264, libx265, libvpx, libfdk-aac, libmp3lame, libopus. Subsequently, these libraries need to be adapted to ffmpeg so that ffmpeg can use these libraries. These libraries are installed using source code.

3.2 Construction of Streaming Media Server

SRS is an excellent streaming media server, open source on git, can be deployed under ubuntu16.04, it is simple and stable, can be used as the source server of RTMP, supports long-term push and pull, and can run on the server During the process of modifying part of the configuration file, functions such as changing the bit rate can be realized without interrupting the service. You need to download the source code of the SRS server. In order to increase the download speed, you can download the relevant code on the code cloud. Next, compile the source code. The third step is to write the SRS streaming media server configuration file as the Fig. 2.

Fig. 2. Configuration file

After configuring the file, you can start the streaming server. After starting the server, you can receive the video stream sent to this port. You can filter through the pipeline or directly query the port to check whether the streaming media server is running.

3.3 Ad-Hoc Network Construction

Connect the prepared big network card to the Raspberry Pi, and then use ifconfig to check whether the connection is successful and query the name of the network card. Since the network card used works in Ad-Hoc mode by default, and all nodes connected to the same network card are in the same network segment, there is no need to perform additional configuration on the network card, just ensure that all communication nodes in the network work at the same frequency Just download. After finding the name of the network card, use the command to assign an IP address of the Ad-Hoc network to the node. After allocating the IP, you need to modify the “olsrd” configuration file; enter the

"/etc./olsrd" directory under the root user, use the command "gedit olsrd.conf" to open the authorized "olsrd.conf" file, and delete it using the library olsrd_txtinfo.so.0.1.Then change the network card that the protocol works to the connected big network card as the Fig. 3.

```
Interface "ens33"
{
    # Interface Mode is used to prevent unnecessary
    # packet forwarding on switched ethernet interfaces
    # valid Modes are "mesh" and "ether"
    # (default is "mesh")

    # Mode "mesh"
}
```

Fig. 3. Set interface

If the subnet masks of different nodes are different, the nodes cannot be connected to the Ad-Hoc network. Use the command to set the subnet mask of each node to 255.0.0.0. When the format of the address is qualified and the subnet mask is the same, you can run "olsrd" under the root user or add "sudo" before the command. The result of the operation is shown in Fig. 4.

```
root@ubuntu: /etc/olsrd
--- 01:59:34.336462 ------------------------------------------------- LINKS

IP address       hyst         LQ       ETX
10.10.10.88      0.000  1.000/1.000    1.000
10.10.10.26      0.000  1.000/1.000    1.000
10.10.10.44      0.000  1.000/1.000    1.000

--- 01:59:34.336492 ---------------------------------------------- NEIGHBORS

    IP address Hyst    LQ      ETX     SYM   MPR   MPRS  will
10.10.10.88     0.000   1.000/1.000     1.000   YES   NO    NO    3
10.10.10.26     0.000   1.000/1.000     1.000   YES   NO    NO    3
10.10.10.44     0.000   1.000/1.000     1.000   YES   NO    NO    3

--- 01:59:34.336525 ---------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
10.10.10.88      10.10.10.44      2.000
                 10.10.10.26      2.000
10.10.10.26      10.10.10.44      2.000
                 10.10.10.88      2.000
10.10.10.44      10.10.10.88      2.000
                 10.10.10.26      2.000
/
```

Fig. 4. Nodes in Ad-Hoc

It can be seen from the figure that there are three nodes in the Ad-Hoc network,10.10.10.88, 10.10.10.26, and 10.10.10.44. Since the geographic distances of the three nodes are in their respective communication ranges, both nodes can pass through one. Communication is performed in a hop mode. In particular, LQ in the figure represents the network state of each node in the network. The closer the value of LQ is

to 1, the more stable the node is in the Ad-Hoc network. Each data information in the graph changes dynamically, that is, the state of the entire Ad-Hoc network changes. For example, when a new node joins, the print information of the terminal will be refreshed.

3.4 Real-Time Video Play

When testing the video transmission function under the Ad-Hoc network of the OLSR protocol, the number of Raspberry Pi depends on the number of hops required during the test. In the single-point-to-single- point test, only two Raspberry Pi are needed for networking; when the hop count is greater than 1, more Raspberry Pi are needed for networking. Before playing the video stream transmitted in the Ad-Hoc network, it is necessary to check whether the "FFplay" player functions normally.

After building a Ad-Hoc network and carrying a routing protocol, there are two ways to deploy a streaming media server. One method is to deploy a streaming media server at each node of the video capture terminal. This method can make the central control terminal very convenient. Multi-threaded viewing of videos from multiple video capture terminals has the disadvantage of increasing the CPU occupancy rate of each video capture node and increasing overhead; the other method is to deploy only the streaming media server on the central control terminal, so that it can be passed through The stream key is used to separate the video streams of different nodes, but it will also cause the degradation of the video quality when playing a single video. Deploying the streaming media server on the central control terminal also has obvious advantages. Firstly, it can effectively reduce the resource consumption of the entire network. Secondly, only the configuration of the central node needs to be improved, which makes it easier to manage and control other video capture nodes.

Set the address of the control end to 10.10.10.31, after deploying the SRS server, set the default listening port to 1935, use the push instruction on the video capture node, and change the push target address to rtmp://10.10.10.31:1935/ live/livestream. This instruction is "ffmpeg 128 -f video4linux2 -r 15 -s 720 × 360 -i /dev/video0 -vcodec libx264 -acodec aac -f flv -y rtmp://10.10.10.31/live/livestream". The terminal prints as Fig. 5. This is the instruction in the "ffmpeg" library, where "-r 12" is to set the transmitted frame rate to 12fps, "-s 1280 × 720" is to set the resolution of the transmitted video to 1280 × 720, "-i /dev/video0" is to set the source of the video, you can set the media file or video capture device, here is set to the connected camera, that is, the camera video stream is transmitted, and the last transmitted address is the address of the server.

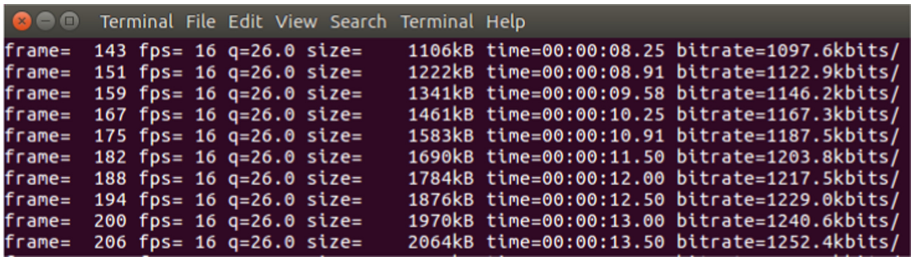

Fig. 5. Pushing stream

In order to reduce the transmission delay, the "-fflags nobuffer" parameter is added when the playback command is executed, that is, the video stream is transmitted in the form of no buffer. Run an order: "ffplay rtmp://10.10.10.31/live/livestream". If the video playback window can pop up normally as shown in Fig. 6, it means that the function of real-time video transmission has been successfully implemented.

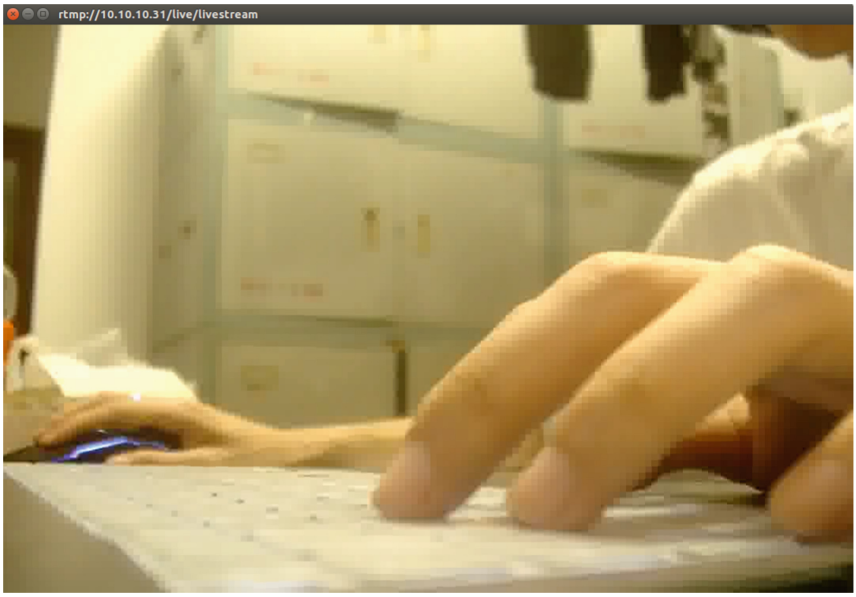

Fig. 6. Pulling stream

4 Conclusions

The focus of this article is to build a wireless Ad-Hoc network to realize the function of real-time video transmission. The overall approach is to use Raspberry Pi 3B+ as the hardware platform, run Ubuntu 16.04 system, build a wireless Ad-Hoc network, carry OLSR routing protocol, install and configure audio and video libraries, use SRS streaming media server, and use RTMP protocol transmission on the application layer for real-time video streaming, the TCP protocol is used on the transmission layer to send control commands. From the analysis of the real-time video transmission quality when the distance between the two nodes is fixed and relatively static, the quality of video transmission can also be improved from the following aspects:

(1) Use a more appropriate routing protocol to achieve higher video frame rate and resolution.
(2) Use physical transmission equipment with higher transmission performance to increase the data transmission rate and effectively improve the quality of video transmission.

(3) Optimize or rewrite the RTMP protocol, or use a streaming media server with higher performance to reduce video delay.

This article has well realized the video transmission function on the Ad-Hoc network, and can improve the transmission performance through cross-layer design in the future.

References

1. Al-Kharasani, N.M., Zukarnain, Z.A., Subramaniam, S.K., Hanapi, Z.M.: An adaptive relay selection scheme for enhancing network stability in VANETs. IEEE Access **8**, 128757–128765 (2020)
2. Rosário, D., Filho, J.A., Rosário, D., Santosy, A., Gerla, M.: A relay placement mechanism based on UAV mobility for satisfactory video transmissions. In: Annual Mediterranean Ad Hoc Networking Workshop, IEEE (2017)
3. Dan, R, Adrian, C., Camelia, A., Adina, A., Benoît, P.: Video content transmission in a public safety system model based on flying Ad-hoc networks. In: International Conference on Automation, Quality and Testing, Robotics, IEEE (2018)
4. Orozco, O.A., Llano Ramirez, G.: OSA: A VANET application focused in energy efficiency. In: 2014 IEEE COLCOM, pp. 1–9 (2014)
5. Rui, C., et al.: Towards a multilyered permission-based access control for extending Android security. Concurr. Comput. Pract. Exp. **30**, e4180 (2018)
6. Oubbati, O.S., Lakas, A., Zhou, F., Güneş, M., Yagoubi, M.B.: A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs). Veh. Commun. **10**, 29–56 (2017)
7. Bujari, A., Palazzi, C.E., Ronzani, D.: A comparison of stateless position-based packet routing algorithms for FANETs. IEEE Trans. Mob. Comput. **17**(11), 2468–2482 (2018)
8. Kang, M.-S., Kum, D.-W., Bae, J.-S., Cho, Y.-Z., Le, A.-N.: Mobility aware hybrid routing protocol for mobile ad hoc network. In: 26th International Conference on Information Networking, pp. 410–414. Bali (2012)