Available online at www.elixirpublishers.com (Elixir International Journal)

Information Technology

Elixir Inform. Tech. 79 (2015) 30642-30645

ISSN: 2229-712X

Channel maintenance in Push-To-Talk Service with high redundancy level

Meenakshi Sinha and Tanmaya
Information Science, R V College of Engineering, Bangalore, India.

ARTICLE INFO

Article history:
Received: 7 September 2014;
Received in revised form:
15 February 2015;
Accepted: 24 February 2015;

Keywords
Push To Talk (PTT),
PoC, SIP Clients,
High Availability,
Fault Tolerance in network,
Redundancy.

ABSTRACT
Push-to-Talk (PTT) is a useful capability for rapidly deployable wireless mesh networks used by first responders. PTT allows several users to speak with each other while using a single, half-duplex, communication channel, such that only one user speaks at a time while all other users listen. This paper presents the architecture and protocol of a fault tolerant PTT service for PTT in all IP Networks.

Introduction

Push-to-Talk (PTT) is a well known service in the law enforcement and public safety communities, where coordination and spectral efficiency are key for efficient communication. Some cell phone companies offer a similar service in the commercial world. However, core differences in motivation drive these two sectors. Cellular phone systems are designed for the busiest hour, as outages impact revenue, while public safety systems are designed for worst case scenarios, as outages impact lives.

Unfortunately, first responders cannot always rely on preexisting ground communication infrastructure. For example, the White House report on hurricane Katrina **[1]** states that 1,477 cell towers were incapacitated, leaving millions unable to communicate.

A PTT system requires an arbitration mechanism (also known as floor control) which determines the order in which participants speak. All participants that wish to communicate with each other form a PTT group. As the name suggests, they request to talk by pressing a button. In contrast to peer-to-peer VoIP systems, data must be disseminated from the speaker to all the participants in a given PTT group.

Since PTT is a complex service involving session controllers, media gateways, failure of any component will result in disruption of service. Previous solutions address using cold standby components so in case of failure backup systems will start, but the existing sessions will be destroyed. In the paper we propose a fault tolerant mechanism so no session is lost and on failure the existing user sessions are seamlessly migrated without any loss of sessions.

Literature Survey

In the literature survey, we explore PTT service in IP networks and the existing fault tolerance mechanism.

PTT allows half-duplex communication between multiple participants which request to speak by pressing a button. On a PTT group only one user is granted Permission-to-Speak at a time, while all the other users listen. DaSilva et al. [7] provide a good survey about PTT technologies. Floor control, an integral part of PTT, has been studied extensively over the years [8]–[10]. Some approaches to decentralized floor control are presented in [11]. A basic level of fault tolerance is built into some of these protocols to enable crash recovery.

PTT is commonly used by law enforcement and public safety communities to efficiently communicate between multiple users. Public safety agencies usually rely on trunked networks, known as Land Mobile Radio (LMR) systems, for voice and data communication [12]. The two major LMR systems are Project-25 [13], which is deployed over North America, and Terrestrial Trunked Radio (TETRA), which is deployed over Europe. Stringent guidelines for PTT, such as 500 ms one-way delay for voice packets to all listeners of a group, ensure that the system operates with acceptable performance.

Cell phone users also benefit from PTT type services that are now offered by telecommunication companies. A common standard, known as Push-to-Talk over Cellular (PoC) [14], allows PTT from different cellular network carriers to interoperate with one another. PoC uses VoIP protocols (SIP, RTP, etc) between clients and the PoC server. A floor control mechanism, referred to as Talk Burst Control Protocol, arbitrates communication in each group. The performance requirements of PoC are less demanding than those in LMR systems. For example, the standard specifies that end-to-end delay should typically be no more than 1.6 seconds and that the turnaround time from the time a user releases the floor until it hears another user speak should be no longer than 4 seconds. An initial evaluation on a GPRS cellular network is shown in [15].

Balachandran et al. show a unifying system for bridging LMR and commercial wireless access technologies [16]. Both LMR and commercial PTT solutions (PoC) rely on a central point of arbitration and send a separate unicast voice stream to each member of the PTT group. On these networks, the inefficiency inherent in using multiple unicast streams is not that costly over the wired backbone medium. Such a design would yield a multi-hop wireless mesh network useless with just a few users, and therefore is not a good fit in our case.

A decentralized approach with a full-mesh conferencing model is presented by Lennox and Schulzrinne in [17]. Florian Maurer [18] shows a decentralized scheme for PTT. Both

Tele:
E-mail addresses: meenakshisinha92@gmail.com

approaches rely on all-to-all communication of control and voice packets between users. While adequate for small conferences or PTT sessions, this approach does not scale well and does not provide the robustness necessary to support node crashes and network partitions and merges, as presented in this paper.

Background

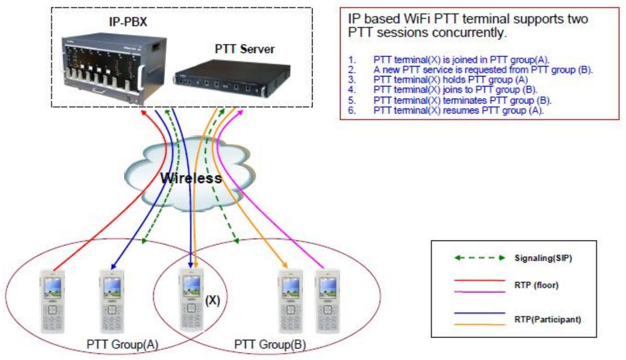
IP-PBX
PTT Server
IP based WiFi PTT terminal supports two PTT sessions concurrently.
1. PTT terminal(X) is joined in PTT group(A).
2. A new PTT service is requested from PTT group (B).
3. PTT terminal(X) holds PTT group (A)
4. PTT terminal(X) joins to PTT group (B).
5. PTT terminal(X) terminates PTT group (B).
6. PTT terminal(X) resumes PTT group (A).
Wireless
(X)
PTT Group(A)
PTT Group(B)
Signaling(SIP)
RTP (floor)
RTP(Participant)

Fig. 1. PTT System Architecture

The PTT system consists of two important sub systems as shown in Fig 1.

1. SIP Clients
2. PTT Servers

SIP Client connects to server for the PTT session. SIP is used for call control and media control messages. PTT Server System implements the IP exchange to divert all the messages from users to the PTT Server as shown in Fig 1. PTT server will do session control and floor management. PTT Server is functionally split into two systems

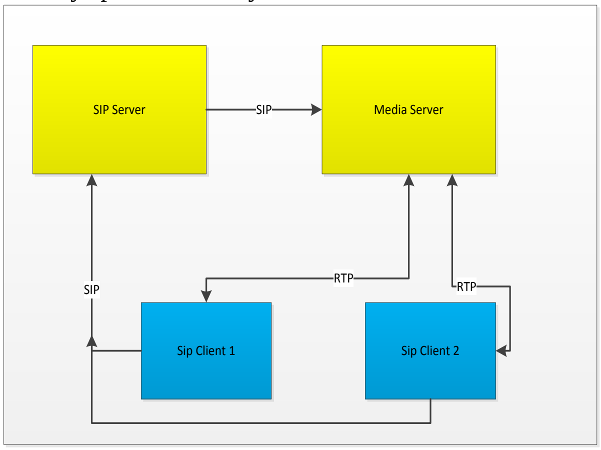

Fig. 2. PTT Server functionality

SIP Server will do the SIP Message handling for connecting the users in a PTT Call and the Media Server will do media delivery and handle the RTP Streams as shown in Fig 2.

The SIP Server and media server can fail. When the SIP Server fails, it will loss all the PTT Sessions in it and would lose control of the sessions. More severe is Media Server failure. If the Media Server fails the media broadcast cannot be made and all users in the session will tear down from the session. So in this project we deal with these two problems

Handling SIP Server failure

Handling Media Server failure

The core of our failure handling process is to shift the state information towards to the client side from the server side. The advantage of this scheme is that server becomes stateless so no need for complex failure recovery mechanism like hot standby mode. Also keeping the state information at client side makes the solution very scalable and less CAPEX and OPEX expenditure for the provider. Failure of a user client will affect only its operation but PTT session still continue.

Fault Tolerant PTT Solution

Handling SIP Server Failure

In the PTT service, the most important challenge in shifting the state from server side to client side is that, the current way of connecting call using INVITE transactions. This makes the server to maintain state information of the sip session and the transaction.

In usual PTT SIP session handling, SIP INVITE transaction is invoked to connect to the Call and SIP BYE transaction is required to disconnect the Call. Due to this way of connecting the call, SIP Server has to maintain lot of states which comes quite impossible and as a result hot standby is very difficult or not possible.

SIP Session details in terms of call-id, from, to, contact, route has to be maintained for each call. So when SIP server fails these session information are lost and this because difficult to restore the session and PTT session fails.

Instead of using the INVITE and BYE transactions, we propose a new mechanism using the SIP INFO message which is very light weight.

In the SIP INFO we propose a new body format for the purpose of PTT. Most of state information need to be maintained at server in existing solutions is pushed to keep at the SIP Client by transferring them via body in the INFO responses; this makes the SIP server stateless.

We design a new PPT profile. This profile is to be carried in the INFO message.

The flow of message in our fault tolerant solution is below

1. The PTT controller sends the INFO message with body. The body specifies the intent to create a PTT session.

The Body will be like this

```
 <CREATE PTT SESSION >
<SDP BODY>
…. Media description
</SDP BODY>
<JOIN LIST>
<Number> 8996899 </Number>
<Number> 8996810 </Number>
…….
</JOIN LIST>
</CREATE PTT SESSION>
```

SDP BODY will have the sdp description and the JOIN LIST will have the number which needs to connect to this session.

2. The Sip Server on receiving the INFO message, it will create a Bride in the Media Server, and send the Bride SDP to the Media Server. The body in the 200 OK response for the INFO is below

```
<CREATE PTT SUCCESS >
<SESSION ID> session id </SESSION ID>
<SDP BODY>
…. Media Bride SDP….
</SDP BODY>
 <BRIDE INFO>
 …. Bridge participant's info…
</BRIDE INFO>
</CREATE PTT SUCCESS>
```

3. Once the Controller User agent receives this media bridge SDP information add it to media bridge information and it will periodically send the INFO message

```
<PTT BRIDGE DETAIL >
<SESSION ID> session id </SESSION ID>
<JOIN LIST>
<Number> 89779 </Number>
……..
</JOIN LIST>
```

```
<BRIDE INFO>
….Bride participant's info …
</BRIDE INFO>
</PTT BRIDGE DETAIL>
```

4. On receiving the INFO body PTT BRIDGE DETAIL , the Sip Server query the Media server for the latest BRIDE INFO detail and send the INFO 200 Response with the latest bridge info details.
5. Sip Server on receiving the INFO with create PTT also sends the invitation to all the parties mentioned in join list to connect to the PTT session.

The body will be like this

```
<JOIN PTT SESSION >
<SESSION ID> session id </SESSION ID>
<SDP BODY>
…. Media Server SDP
</SDP BODY>
<JOIN LIST>
<Number> 8996899 </Number>
<Number> 8996810 </Number>
…….
</JOIN LIST>
</JOIN PTT SESSION>
```

6. On receiving the JOIN PTT Session, the participants will send a 200 OK response with its media description.

```
<JOIN PTT SUCCESS >
<SESSION ID> session id </SESSION ID>
<SDP BODY>
…. Media description….
</SDP BODY>
</ JOIN PTT SUCCESS >
```

7. On the 200 OK response with JOIN PTT SUCCESS, the sip server will send the media description to the Media Server. Media Server will update the Media Bridge.

By pushing the bride info and participants information of floor control to the PTT controller user agent, we have made the sip server as stateless.

Handling Media Server Failure

The Media server is equipped with the mechanism to create the Media Bridge on fly with the media bridge information.

Sip Server on the receiving the PTT Bridge detail message will forward the Bridge Info in the body to the media server. Media server will create the Bridge based on the Bridge Info on fly. So even if Media server fails and restarts it will have least impact on the existing PTT Session

Performance Analysis

We implemented the proposed fault tolerance mechanism for a LAN network. We introduced random failures for the SIP server and the Media Server. 100 PTT sessions is created with 3 users for PTT session. We measure the service disruption time between our solution and hot standby solutions as shown in Fig 3. In the observation we found that the service disruption time in our approach is very less compared to hot standby solution.

Since our approach is stateless we also measure the number of PTT sessions the server is able to handle under different resource consumption. We found the number of PTT sessions the server is able to handle has increased a lot because of the stateless mechanism as shown in Fig 4.

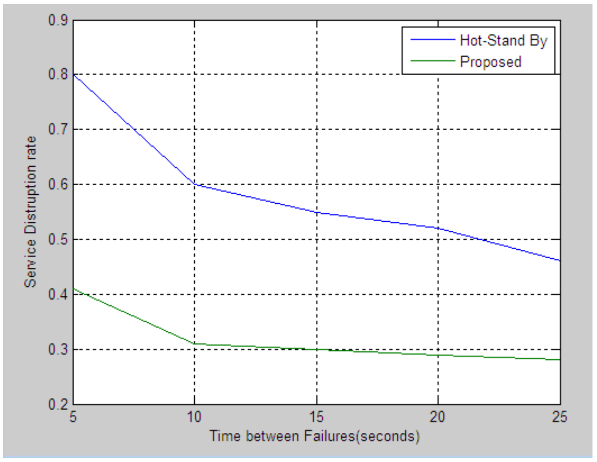

Fig. 3. Service Disruption time between improved and traditional solution

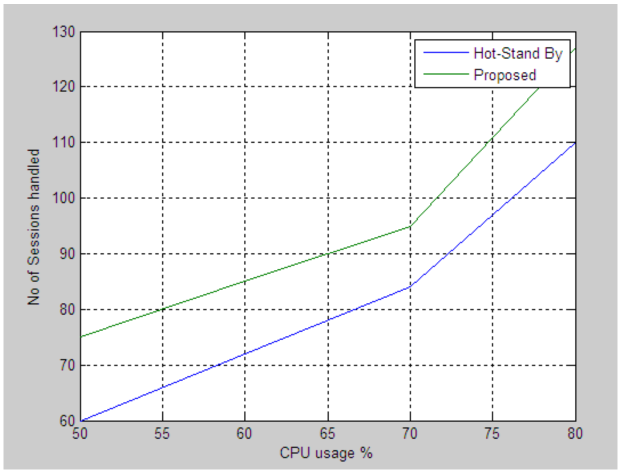

Fig. 4. Number of PTT sessions handled by the server

Conclusion And Enhancements

In this paper, we have implemented the Fault tolerant mechanism and proved that our solution works better than hot standby solution. Our solution also reduces the CAPEX and OPEX cost incurred if hot standby configuration is implemented.

We have left open the security part left untouched. Since body information in INFO and INFO Response goes as plain text any attacker can attack the messages. To avoid this we need to encrypt the message. This we plan to do as future work.

References

[1] "The federal response to hurricane katrina: Lessons Learned," February 2006, washington, DC: Office of the Assistant to the President for Homeland Security and Counterterrorism.
[2] M. R. Souryal, J. Geissbuehler, L. E. Miller, and N. Moayeri, "Realtime deployment of multihop relays for range extension," in *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*. NewYork,NY,USA:ACM,2007,pp.85–98.
[3] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. Das, "Performance optimizations for deploying voip services in mesh networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2147–2158, Nov. 2006.
[4] Y. Amir, C. Danilov, M. Hilsdale, R. Musaloiu-Elefteri, and N. Rivera, "Fast handoff for seamless wireless mesh networks," in *MobiSys 2006*. New York, NY, USA: ACM Press, 2006, pp. 83–95.
[5] M. Handley and V. Jacobson, "SDP: Session Description Protocol," *RFC 2327*, April 1998.

[6] “The SMesh wireless mesh network,” http://www.smesh.org. [Online]. Available: http://www.smesh.org

[7] L. DaSilva, G. Morgan, C. Bostian, D. Sweeney, S. Midkiff, J. Reed,C. Thompson, W. Newhall, and B. Woerner,
“The resurgence of pushto-talk technologies,” *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 48–55, Jan. 2006.

[8] P. Koskelainen, H. Schulzrinne, and X. Wu, “A sip-based conference control framework,” in *NOSSDAV ’02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*. NewYork,NY,USA:ACM,2002,pp.53–61.

[9] R. Malpani and L. A. Rowe, “Floor control for large-scale mbone seminars,” in *MULTIMEDIA ’97: Proceedings of the fifth ACM international conference on Multimedia*. NewYork,NY,USA:ACM,1997,pp.155–163.

[10] H.-P. Dommel and J. J. Garcia-Luna-Aceves, “Floor control for multimedia conferencing and collaboration,” *Multimedia Syst.*, vol. 5, no. 1, pp. 23–38, 1997.

[11] S. Banik, S. Radhakrishnan, T. Zheng, and C. Sekharan, “Distributed floor control protocols for computer collaborative applications on overlay networks,” *Collaborative Computing: Networking, Applications and Worksharing, 2005 International Conference on*, pp. 10 pp.–, 19-21 Dec.2005.

[12] D. Sharp, N. Cackov, N. Laskovic, Q. Shao, and L. Trajkovic, “Analysis of public safety traffic on trunked land mobile radio systems,” *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 7, pp. 1197– 1205, Sept. 2004.

[13] “Project 25, Association of Public-Safety Communications Officials,” http://www.apco911.org/frequency/project25.

[14] O. M. Alliance, “Push-to-talk over cellular (poc), release 1.0, 2005.”

[15] A. Balazs, “Push-to-talk performance over gprs,” in *MSWiM ’04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. NewYork, NY, USA:ACM, 2004, pp. 182–187.

[16] K. Balachandran, K. Budka, T. Chu, T. Doumi, and J. Kang, “Mobile responder communication networks for public safety,” *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 56–64, Jan. 2006.

[17] J. Lennox and H. Schulzrinne, “A protocol for reliable decentralized conferencing,” in *NOSSDAV ’03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*. NewYork, NY, USA:ACM,2003,pp.72–81.

[18] F. Maurer, “Push-2-talk decentralized,” 2004.

[19] Y. Kado, A. O. Lim, and B. Zhang, “A study of wireless mesh network routing protocols for push-to-talk traffic,” in *Proceedings of the 16th International Conference on Computer Communications and Networks, IEEE ICCCN 2007, Turtle Bay Resort, Honolulu, Hawaii, USA*, 2007, pp. 197–202.

[20] “The Spines Overlay Network,” http://www.spines.org.

[21] J. Moy, “Multicast extensions to OSPF,” IETF, RFC 1584, Mar. 1994. [Online]. Available: ftp://ftp.isi.edu/in-notes/rfc1584.txt

[22] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session Initiation Protocol,” *RFC 3261*, Jun. 2002.

[23] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, “Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP),” *RFC 3725*, April 2004.

[24] H. Schulzrinne and S. Petrack, “RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals,” *RFC 2833*, May 2000.