International Journal of Future Generation Communication and Networking
Vol. 9, No. 5 (2016), pp. 181-188
http://dx.doi.org/10.14257/ijfgcn.2016.9.5.18

Design, Extension and Implementation of RADIUS Client

Feng Jian and Nan Tian-zhu

College of Computer Science & Technology, Xi'an University of Science and Technology, Xi'an, China 710054
actour@163.com

Abstract

RADIUS is an authentication, authorization and accounting protocol being widely used in network environments. Safe, efficient, and scalable RADIUS client module is an important part for a network access server (NAS) to provide access services. Through describing working mechanism of RADIUS, the architecture and interaction model to the external modules of RADIUS client are given. Based on finite state machine (FSM) theory, states, events and actions of the protocol are analyzed, state transition mechanism is proposed, and then RADIUS client module is implemented. After that, the protocol is extended from aspects of secondary accounting, accounting update, users with no charge, forcing user offline, error status descriptions and so on. PPPoE test module in AX/4000 broadband test system is used for simulating access users to test functionality and performance of RADIUS client module on NAS, the test results show that NAS which realized the RADIUS client module can meet carrier-class functionality and performance requirements of access services.

Keywords*: RADIUS protocol, FSM, NAS, protocol extension, secondary accounting*

1. Introduction

AAA services, namely authentication, authorization and accounting services, are not only effective means used by network operators to ensure safe, efficient, reliable, and rational use of network resources, but also the core business of user access.

Currently mainstream protocol applied to support AAA services is RADIUS (Remote Authentication Dial In User Service). It is based on C/S model, in which Network Access Server (NAS) is working as client, interacting with RADIUS server to realize authentication, authorization and accounting services for remote access. With the popularity of network applications, NAS has become an important access gateway between access layer and edge layer in backbone to complete the user's high-speed data access. In practical applications, NAS must meet functionality and performance requirements of carrier-class broadband user access. The realization of RADIUS client module in NAS will affect directly to availability, reliability and response speed of NAS. Currently, there are some implementations of RADIUS client software, but they mostly failed to address problems of high load and high burst caused by the broadband network effectively, such as the problem that no guarantee to normal user access when a large number of users being on-line at the same time or network congestion, at the same time the consideration of scalable were inadequate. And because RADIUS protocol is complex and stateless, the design process is often prone to omissions, causing imperfect implementation and low performance. RADIUS client module designed in the paper is to address these problems and to support carrier-class access applications. FSM (Finite State Machine) is an important cornerstone of computer science, is a widely used software design pattern in the development of network protocol [1]. Based on protocol analysis, FSM state transition of RADIUS protocol is given, the capabilities are extended and eventually the implementation of RADIUS client module in NAS is completed.

ISSN: 2233-7857 IJFGCN

2. Overview of RADIUS Protocol

RADIUS is a protocol issued by IETF (Internet Engineering Task Force) to provide security, authentication and accounting management, and currently includes: RFC2865 [2], RFC2866 [3], *etc*. Security management includes setting shared key between RADIUS client and RADIUS server, encrypting sensitive information, and using authentication codes to test the integrity of the data packets. The main mechanism of the protocol is shown in Figure 1.

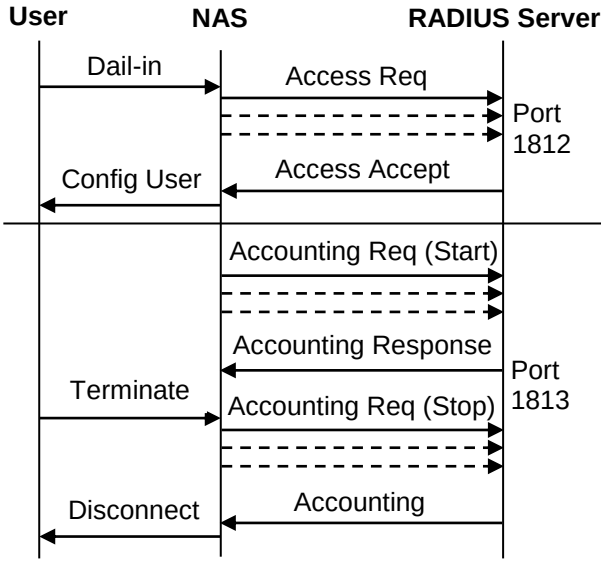


Figure 1. Working Mechanism of RADIUS Protocol

In Figure 1, NAS represents RADIUS client in RADIUS protocol workflow. The dashed line represents the retransmissions caused by timeout. For brevity, retransmissions of response message sent from RADIUS server to NAS are not shown.

RADIUS is a scalable protocol, its attributes have variable length and can carry authentication, accounting and detailed configuration information. In implementation, new attributes can be added to extend the protocol easily [4].

3. Design and Implementation of RADIUS Client Module

3.1. Overall Design

As a secure authentication and accounting management module in NAS, on the one hand, RADIUS client should be able to receive access requests delivered by user module in NAS, encapsulate users' identity information to RADIUS messages as initiator authentication requests to RADIUS server; on the other hand, RADIUS client should be able to receive the authentication response from the server, grant user's access privileges based on the response. For legitimate users, the RADIUS client should also pass accounting start requests and accounting stop requests to the RADIUS server, including start on-line time and off-line time for user charging, as well as the total flow during the period of accessing to the network, to support the RADIUS server to deal with accounting information by various means.

Figure 2 shows architecture of RADIUS client module and information exchange between the module and its external components. In the figure, RADIUS server and user module are external components, in which the user module is usually PPP or PPPoE module, it is always a component of NAS.

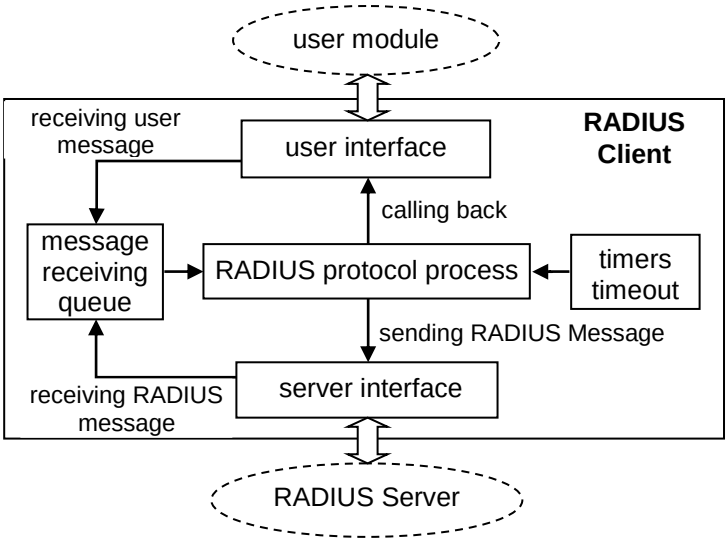


Figure 2. Architecture of RADIUS Client Module and External Components

RADIUS client module consists of two tasks (RADIUS protocol processing task and timer task), two interfaces (user interface and server interface) and one message queue (message receiving queue), functions of each part are as follows:

- RADIUS protocol processing task

Receives messages from message receiving queue; parses the packets and deals with the messages according to designed RADIUS FSM, including changing conversation state and taking appropriate actions. The messages received include user request messages sent by the user module and RADIUS protocol packets sent by the RADIUS server.

Timer timeout events need to be processed simultaneously, including checking the timer trigger mark, retransmitting packet, and cleaning session resources.

- Timer task

Provide timing functions. After timer timeouts, sets trigger tag of the timer.

- User interface

Used for communication with the user module on NAS, its function is to pass user requests from the user module to the message receiving queue, or return access control instruction back to the user module, i.e., whether allow a user to access network resources and how to access network resources and other information.

- Server interface

Be responsible for communication with the RADIUS server. Sends RADIUS packets to the RADIUS server and receives RADIUS response packets from the RADIUS server, and then puts the messages into the message receiving queue.

- Message receiving queue

Message receiving queue is a FIFO circular queue, stores request messages from the users and response message from the RADIUS server and waits for RADIUS protocol processing task to deal with.

3.2. FSM of RADIUS Client

According to [5], RADIUS system should be able to handle a large number of user accesses simultaneously. In this case there are multiple user access requests at the same time, each request is subjected to a plurality of different processing stages from beginning to the end, and different stages of different user sessions intertwine together resulting in a large number of intermediate states at the same time. As RADIUS basis protocol, [2] and

[3] did not give a description of the state machine, implementation of stateless protocol causes complex program logic in the case of processing a large number of users simultaneously and is difficult to debug, and is more difficult to meet the performance requirements of carrier-class access services. In order to provide carrier-class access service, ensure clear protocol control structure and efficient codes, it's essential to build model and give formal description for the problem.

FSM is a tool for object behavior modeling, and its main functions are to describe the sequence of states in life cycle of an object and how an object to respond to a variety of events from the outside world. In the process of RADIUS protocol, each and every process for user access is a state transition driving by external events, so FSM can be used to describe these state transition processes. In this paper FSM for RADIUS protocol is given, below are events, actions, and state transition analysis. Usually, FSM can be defined as a five-tuple $M = (Q, \Sigma, \delta, q_0, F)$. Among them:

- $Q=\{q_0, q_1,\ldots,q_n\}$ consists of a set of states (including the initial state). At a determined time, FSM will be in a determined state q_i. In RADIUS client, there are 5 states, shown in Table 1.

Table 1. States of Radius Client

State	Definition
down	initial/ terminal state
iar	initialize authentication request
iacr	initialize accounting request
aae	authentication/accounting end (normal-using internet)
act	accounting terminate

- $\Sigma=\{\sigma_1,\sigma_2,\ldots,\sigma_m\}$ is a set of input events. The state machine usually switches from one state to another state prompted by events. At a determined time, FSM can only receive one determined event σ_j. In RADIUS client, there are events including receive authentication request from user, timers timeout, receive response from RADIUS server, and so on, shown in Table 2.

Table 2. Input Events of RADIUS Client

Event	Definition
rar	receive authentication request from user
raf	receive authentication fail message from RADIUS server
ras	receive authentication success message from RADIUS server
ract	receive accounting term request from user
racs	receive accounting success message from RADIUS server
rtr	receive abnormal terminate request
tto＋	timer timeouts one time
tto－	timer timeouts all the time
unk	unrecognized RADIUS packets or attributes

- $\delta: Q \times \Sigma \rightarrow Q$ is a transition function that maps input events and current states to a next state. The function takes the current state $q_i \in Q$ and an input event $\sigma_j \in \Sigma$ and introduces a set of actions and the next state $q' = \delta(q_i,\sigma_j) \in Q$. Table 3 shows the actions caused by input events.

Table 3. Actions of RADIUS Client

Action	Definition
sar	send authentication request to RADIUS server
itt	initialize timer
stt	stop timer
gaf	generate authentication fail event to user
gas	generate authentication success event to user
sas	send accounting start request to RADIUS server
sat	send accounting stop request to RADIUS server
gacf	generate accounting fail event to user
gacs	generate accounting success event to user

- $q_0 \in Q$ is the initial state, computation begins in the start state with a input event. In RADIUS client, it is down state.
- $F \subseteq Q$ is the end state, after reaching end of the state, the FSM will no longer handle events. In RADIUS client, it is down state too.

FSM of RADIUS client is shown in Figure 3.

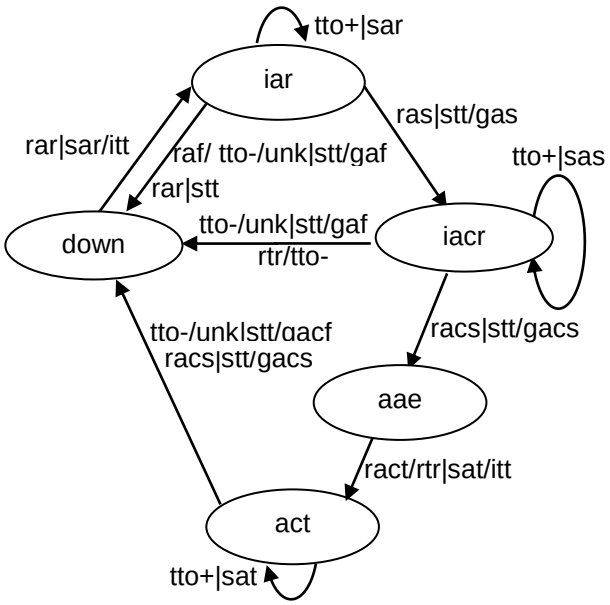


Figure 3. FSM in RADIUS Client

In Figure 3, states of the FSM are in the circle; arrows indicate direction of state migrations; events that trigger the actions are before "|"; only one event will be taken place in practice at one time, so a variety of events which will trigger the same action sequence are separated by "/"; action sequence triggered by one event is after "|" and separated by "/". Action sequence will execute in the order after the event happened.

Before entering FSM processing, consistency examination between current state and pending events should be done. And when FSM is running, received user messages and RADIUS packets need to be checked first. If the validation fails then the message should be discarded; if unexpected events occur, then release resources of the session.

3.3. Protocol Extensions

(1) Secondary accounting

For convenience of user reconciliation and secondary operations for hotels and other large customers, RADIUS client module is designed to support secondary accounting, namely passing same accounting information to two RADIUS servers. FSM described in 3.2. is state migration instruction for single authentication and accounting process. In order to support secondary accounting, communications with two servers require to be controlled. To do this, states of accounting Session Are Added, Shown In Table 4:

Table 4. States of Accounting Session

State	Definition
init	the initial state
mid	intermediate state, one of the two RADIUS server responds
end	successful state, two RADIUS servers both respond

(2) Attributes extension

RADIUS protocol exchanges information via attached attributes. In addition to the defined attributes in the protocol, developers can also customize attributes to store customize information such as user data, network configuration, control information and so on for meeting new business requirements. The RADIUS client module implements business expansions through extended attributes listed below.

- Accounting update

To avoid abnormal bill caused by loss of accounting stop messages, attribute of RADIUS accounting update is enabled. The attribute is Acct-Interim-Interval, its attribute type is 85, and it will be brought back in Access Accept message from RADIUS server, indicating accounting update interval. The client will set accounting update period according to the interval and send an accounting update message after each period.

- User with no accounting

On business needs users who are charged on a monthly basis or on a yearly basis usually pay a fixed amount of fees at one time. When such users access the network, they only need to be authenticated and need not to be charged every time they access the Internet according to time-span or traffic. A custom attribute No-Acct-User is designed for this, its attribute type is 100, bringing to the RADIUS server in Access Request message, to notice the RADIUS server that accounting message of the user will not be needed.

- Forcing users offline

In RADIUS protocol, servers do not take the initiative to send control messages to clients, they only answer to the clients. However, in practical applications, servers sometimes need to take the initiative to indicate the clients to break users' connection under certain conditions, such as user balance is insufficient, or the servers need to be restarted. To do this, standard RADIUS protocol attribute Session-Timeout is enabled, it can set users' timer, and when the timer timeouts the client should let the user offline. The attribute type is 27, carried in Access Accept response from RADIUS server.

- Error status description

Error messages returned by RADIUS protocol are very limited, and the client usually cannot get the exact reason for the failure. To solve this problem, a custom attribute Error-

Info is added. Its attribute type is 101, carried in the Access-Reject response from RADIUS server, used to give detailed error information.

In addition, other attributes can also be customized according to the needs, such as adding VLAN attribute to indicate a user belongs to a particular VLAN, *etc*. For normal use of new client attributes, the same attributes must be added in attribute dictionary on the RADIUS server.

4. Testing

In testing, Spirent AX/4000 test system is used to test the NAS which implemented the proposed RADIUS client module. The Spirent AX/4000 broadband test system is basically the most powerful system ever developed for testing performance and Quality of Service (QoS) of broadband networks. It can generate a variety of PPPoE sessions at one time and send these requests to user module in the NAS, and the NAS worked as a RADIUS client, interacted with AX/4000 and RADIUS servers. The testing process is designed in accordance with [5]. Functionality tests and their results are shown in Table 5.

Table 5. Functionality Testing

Testing Items	Contents	Results
user access 1	use correct username and password	success
user access 2	Uuse wrong username or password	fail
secondary accounting	config secondary accounting, initiate PPPoE call. After the call is completed, check user's accounting information on both RADIUS servers	information are consistent
accounting update	set accounting update, initiate PPPoE call. After the call is completed, check user's accounting information	receive accounting update message periodically and correctly
forcing users offline	forcing online user to offline	user is offline and accounting information are correct
user with no accounting	set user to no accounting, initiate PPPoE call. After the call is completed, check user's accounting information	accounting information are not found

The system is designed to support 2048 active PPP sessions at the same time. The results of performance testing are as follows:

Table 6. Performance Testing

Testing Items	Contents	Results
call completing rate	initiate 21 calls per second and 210 calls totally	average call completing rate >99%
access error rate	initiate 21 calls per second and 210 calls totally	access error rate <0.1%
number of links	initiate 2000 calls per second	average number of links per second>200
connection time	initiate 2000 PPPoE calls totally	average connection time is 4s
accuracy of accounting information	after the call is completed, check user's accounting information	duration error in 12 hours <1s, error of traffic information <5%

The test results show that the realization of RADIUS client module complies with the requirements of standard RADIUS protocol, and can meet functionality and performance requirements listing in the literature [6]. In actual use, the NAS with the realization of

RADIUS client module can work properly and has stable performance under a large number of users access.

5. Conclusions

An implementation of RADIUS client based on FSM is presented in this paper, with consideration of protocol extensions. Functionality and performance of NAS implementing the RADIUS client module are tested via Spirent AX/4000 broadband test system, and the results show that both meet the requirements of carrier-class broadband access.

In the one hand, a further improvement of the system may also consider adding functions such as secondary authentication and accounting information stored in local, in order to meet the needs of users better. In the other hand, safety and suitability issues of RADIUS protocol become increasingly prominent, and there are some devices using next-generation AAA protocol - Diameter to achieve access management functions, so design and implementation of Diameter will be our next concern.

Acknowledgements

This work was supported in part by Shaanxi Provincial Natural Science Foundation Project (No. 2012JQ8030).

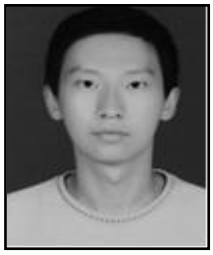
References

[1] H. X. Ying, X. Z. Hui and C. H. Yuan, “Software Development for Embedded System and Its Applications”, Techniques of Automation and Applications, vol. 23, no. 3, (**2004**), pp. 56-58.
[2] C. Rigney, A. Rubens, W. Simpson and S.Willens, “RFC 2865: Remote Authentication Dial In User Service (RADIUS)”, (**2000**).
[3] C. Rigney, “RFC 2866: RADIUS”, Accounting, (**2000**).
[4] W. Pan, H. Kai, F. Yang and X. Guilan, “RADIUS Client Timeout-processing Optimization in WLAN”, Study on Optical Communications, vol. 4, (**2013**), pp. 68-70.
[5] Telecommunications Industry Standards of the People’s Republic of China, Testing Methods of Network Access Server (NAS)-Broadband network access server, YD/T 1265-2003, (**2003**).
[6] Telecommunications Industry Standards of the People’s Republic of China, Technical Requirements of Network Access Server (NAS)-Broadband Network Server, YD/T 1148-2001, (**2001**).

Authors

Feng Jian, she was born on August 1973, she received her doctoral degree of Computer Software and Theory from Northwest University, Xi’an, China, in 2008. And research interests on computer network and communication, network security, distributed computing.

She is currently an Associate Professor with College of Computer Science & Technology, Xi’an University of Science and Technology, Xi’an, China.



Nan Tian-zhu, he was born on October 1991, he is now a master candidate of Computer Science and Technology from Xi’an University of Science and Technology, Xi’an, China. And research interests on computer networks and data analysis.

