

Token-based authorization of Connection Oriented Network resources.

Leon Gommans, Franco Travostino, John Vollbrecht, Cees de Laat, Robert Meijer.

Abstract.

Authentication, Authorization and Accounting (AAA) mechanisms have an increasingly versatile role when performing access control on various types of network resources. Emerging data intensive grid applications generate new network requirements. These requirements call for (pre-) allocate-able data transport facilities serving specific user communities. The network specific requirements are characterized by a very limited need for connectivity, usage during specific periods and in many cases the need to span large distances at maximum available speed. This should all be possible at the lowest achievable cost involving different owners of the involved networks. Solutions are researched in the area of connection oriented networking using relative cheap, lower layer switches. This paper presents a novel usage of an existing model to grant access to connection oriented network resources based on an authorization message sequence involving a token. The presented approach makes use of specialized monitor hardware emerging in high capacity low-layer switches.

1.0 Introduction

The IRTF AAA Architecture Research Group [1] performed research into fundamental authorization sequences and described the outcome as models in RFC2904 [2]. Further study into these sequence models continued within the GGF AuthZ Working Group [3]. Both studies showed that there are at least three fundamental sequences that describe the interaction between three fundamental entities involved in an authorization. In this article we explore the mentioned sequence models within networking environments. We first show examples of the two common sequences. For authorization of access to lower layer, packet-based connection-oriented network transports, we subsequently focus on the third model called the “push sequence”. Although quite common in many applications, the usage within layer 1, 2 or 3 networking has not seen much attention. The authors believe that the push model, given recent developments in network packet monitoring technology, offers a number of benefits above the other two fundamental sequence models currently in use. In particular the model allows a chain of organizational entities to distribute tokens representing pre-allocated network bandwidth. Tokens issued by the proper authority allow acceptance and collection by multiple administrative network domains. We conclude by showing an example architecture.

2.0 Authorization message sequence models

In summary, chapter 3 of RFC2904 recognizes three fundamentally different message sequences involving:

- a. The User. The entity that sends an access request and obtains an authorization.
- b. The AAA Server. The entity that receives a request and makes a policy based authorization decision using information contained within the request, information concerning the resource and possibly information from the stakeholders. After an authorization decision has been made it replies to the entity that send the request. The AAA server is considered the single authority governing access to the underlying service equipment. The AAA Server and Service Equipment forms in this sense a unity called the “service provider”. An AAA server can also represent a User Home Organization containing user access rights.
- c. The Service Equipment. The entity that represents the service which needs information that authorizes the usage of the service offered by the equipment.

Between these entities one can imagine a number of message sequences aimed at *requesting* an authorization and subsequently *using* the authorization to gain access. To request an authorization the user can contact the AAA Server or the Service Equipment. In the latter case the Service Equipment will out-source the access decision to the AAA server. The AAA server will reply a decision. The Service Equipment can subsequently enforce the access. Fig. 1 shows this sequence as the *pull sequence*. In the second and third scenario the User will both send a request to the AAA server. In the second scenario, called the *agent sequence*, the AAA Server will act as an agent that will first take an authorization decision. It will then talk directly to the Service Equipment to permit or deny access. After a decision has been made in the third scenario, the AAA server will create a (secure) token that is handed back to the user. The user must then push this token to the Service Equipment hence its name: the *push sequence*.

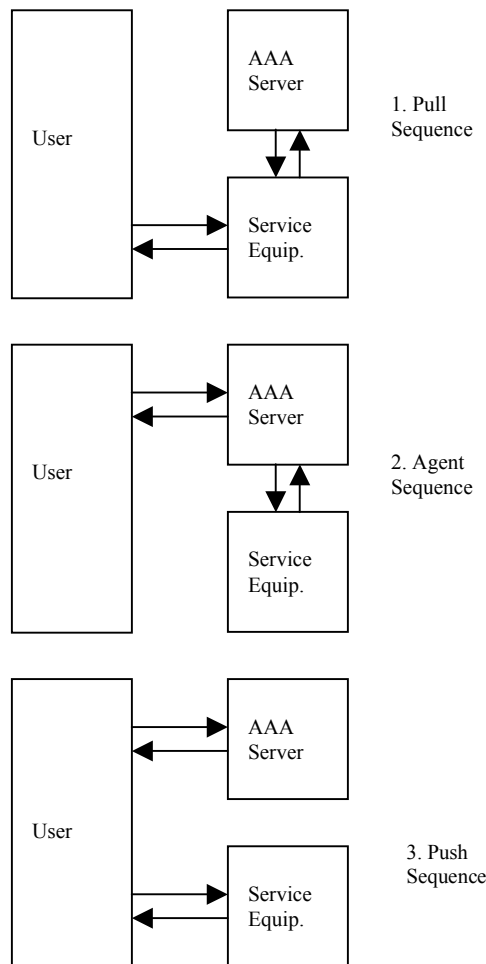


Fig 1: Fundamental authorization sequences

3.0 Authorization sequence models used within traditional networking.

We have seen 3 different fundamental models in the previous chapter. The pull- and agent models for authorization sequences are quite common within networking. In this chapter we will give some examples as they are being used for a number of different applications in the area of access control and QoS networking.

3.1 The pull sequence.

The pull sequence as described in chapter 2 is typically used whenever a user tries to gain access via some device that is offering service and is also capable of enforcing access. This device could be at the ingress of a network. As an example of such an enforcement device we describe a Network Access Server (NAS). A NAS is used to recognize a user dialing into an Internet Service Provider using a modem. Whenever the NAS receives a call on one of its phone lines, it will pick up the line and connect a modem. After the modems synchronize and line protocol has been established, an authentication protocol such as (CHAP [4] or PAP [5]) will exchange user-identifying information with the NAS. The NAS is configured to contact an AAA server in order to send it the user related information using a protocol such as RADIUS [6]. Multiple NASs may contact the same AAA server. One AAA server will typically serve a single domain of users. A so-called User Home Organization (UHO) could operate the AAA server independently. A university could administer such a server on behalf of their student community. If permitted, each service providers NAS could in theory access a university's AAA server. As this requires every NAS to know about an additional university, this model does not scale very well. One approach to the solution is to use proxy chaining as described in RFC 2607 [7].

3.2 The agent sequence.

The agent model is typically used whenever a user or a service does not have the knowledge and/or relationships to obtain a particular authorization. Agents abstract the underlying complexity and offer the authorization service in a simplified and/or easier to use way towards the user of the agent. Agents typically advertise themselves by a well-known mechanism. An example of an agent sequence is used in a

bandwidth broker [8] scenario within networks that deploy differentiated services [9]. A broker could keep track of the available capacity inside a domain. A chain of domains will implement a certain capacity between a source and destination. Bandwidth brokers will provision routers inside a domain with the proper queue parameters to implement a certain diffserv model. Bandwidth Brokers will communicate with neighboring Bandwidth Brokers to ensure a certain bandwidth is available when traffic traverses the underlying domains. A user will typically communicate with a Bandwidth Broker at the source domain. This bandwidth broker will effectively authorize traffic if conditions checked with neighboring Bandwidth Brokers so permit.

4.0 The use of the push model within new forms of networking.

The push model is used at the application layer predominantly. Many examples are found where signed tokens or certificates enable applications or system to recognize users, user privileges in order to provide access control functions. Access is gained after a user first obtains a token from the AAA server and subsequently presents this token to the Service Equipment. A certificate, in comparison with a token, suggests a particular format. A token is a more general type of trusted, cryptographically protected proof of authorization with less strict issue and usage policies. A token cryptographically binds attributes to an issuing attribute authority. Therefore a token could be acquired and subsequently used anonymously. In our case the AAA server acts as a kind of attribute authority that issues something like an Attribute Certificate. As the terms Attribute Certificate and Attribute Authority have already been defined in RFC3281 [10], we refer to our AAA server issued cryptographic entity more generally as a token without assuming any particular format like X.509 [11]. Within this context we assume that a token is a set of attributes that are cryptographically bound (signed) to an authority represented by the AAA server. Signing proofs the authenticity and integrity of a token. It neither prevents duplication nor ensures confidentiality. A token could be used in several ways: It could for example periodically handed in a secure fashion to the network or it could be used as key material with some message security method that is used along with every message (eg. some encryption or signing method).

4.1 Content Monitoring and Action Device.

Lower layer network transports functions work in a connection-oriented way. These functions do typically not recognize tokens inside signaling messages or as part of the data stream. In this document we intend to drive network provision functions based on its recognition. We make the assumption that a token can be recognized even at the lowest network layer possible. This implies recognition at layer 1 or 2 switches. As part of elaborate network intelligence gathering capabilities, switch manufacturers develop hardware functions that are capable of recognizing network content at wire speed. This hardware includes programming capabilities that can subsequently trigger and execute actions. We will refer to this kind of device as a Content Monitor and Action Device (CMAD). Although one can imagine that such device could be implemented on photon-based switches, engineering efforts currently focus on switches that perform their functions using electrons.

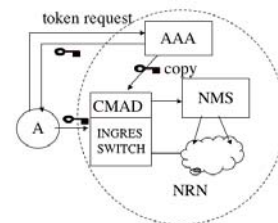


Fig 2: A possible position of the Content Monitoring and Action Device

Assuming the existence of such device, this article considers possible application of a CMAD with regards to the authorization of special network resources. Fig. 2 shows a possible position of a CMAD as part of an ingress switch at the edge of a network. The token, represented by the key symbol, is handed to user A as a result of the push sequence. The token or a derivate of this token is subsequently recognized by the CMAD when it is handed to the network via the ingress switch.

When a token is handed, some method must be used to avoid replay attacks. If the token is used The AAA server may already have provided a copy of the token to the CMAD as to ensure fast recognition. The CMAD could signal the network's Network Management System (NMS) to implement the desired connection inside the network cloud of the NRN. There are many more sequences and software components such as a

resource manager that will play a role in the implementation of such a network. However, for simplicity reasons, those sequences and components have been omitted in Fig. 2.

4.2 Rationale.

Let us first consider the rationale behind a token-based network. There are some fundamental differences between the push model and the pull or agent model. First, the push model allows a separation in time between the issuing of an authorization token and the usage of the token. The pull and agent model assume an authorization to be taken at the time of sending a request. The push model assumes that the issued authorization token will be used at a later point in time. Second, if the issuing policy so permits, the entity requesting the authorization token may be different from the entity using the token. Third, a token from a central authority may autonomously be recognized by a number of different services or service domains without the need for further communication. This approach fits well in a centralized authorization model where each domain is still able to take autonomous decisions on the presented token. For example, a policy may decide that a token may only be presented at selected ports. As tokens can be carried along with an application, a token-based approach could for example simplify operational aspects when considering pre-allocation requirements in grid environments.

4.3 Goals.

The goals of using a push model can be summarized as:

- a. Allow a token to represent a pre-allocated network connection that spans multiple domains for a specified amount of time between two locations.
- b. Allow the creation of various token distribution models where a token can be requested and subsequently be handled by (a chain of-) organizations that act on behalf of the ultimate user.
- c. Allow fast creation of a connection based on pre-configured information that is referred to by the token. This information is already established at token setup-time.

4.4 Example use case.

4.4.1 Definition of the network

Consider a federation of NRNs that are served for their interconnectivity by an organization that involves a number of Global Network Carrier (GNCs). The Federative Network Organization (FNO) is responsible for maintaining contracts with individual NRNs and with GNC. The FNO could operate its own carrier network but for simplicity reasons the FNO and GNCs are considered separate functional entities. The FNO is also responsible for the financial clearing and settlement between its members.

4.4.2 Maximum bandwidth connection

Federation members offer both best effort and special *maximum bandwidth connections* between a set of well-defined and static source and destination locations that serve a certain community. Maximum bandwidth connections do not experience any packet loss either from sharing network resources with other users or by any rate-limiting mechanism. A maximum bandwidth connection allows usage of aggressive protocols intended to maximize the bandwidth utilization. We assume that grid applications at the endpoints require either pre-allocated or on-demand maximum bandwidth connections. The rationale behind this assumption is described in [12].

4.4.3 Using and obtaining a token

At the time the application needs the pre-allocated bandwidth, the user will insert a token in the network. This can be done either as a signal along with the data stream (in-band signaling), or the token can be handed to a special signaling interface (out-band signaling). In order to obtain tokens, the user or a representative user organization contacts the federation AAA server with an appropriate request. The information in the request is based on the advertised availability of specific connection that is available to the public or to a certain community. A request may be made for a specific period of time starting now or at some future point in time. After the request is received the FNO will contact the involved resource domains to find out if a reservation is still possible. If all involved resource managers of the domains agree, the FNO will subsequently allocate all resources both inside the NRN domains and also allocate an interconnecting link

from the GNC domain. In our example the FNO is responsible for the resource management of the GNC links. A GNC could also have a separate function for its resource management. Only if all underlying resources are available, the final allocation will be registered. The FNO is considered the authority that is contractually allowed to allocate the advertised network resources with NRNs and the GNC on behalf of a requesting NRN. A requesting NRN is contractually allowed to make further refinements or subdivisions to the offered service on behalf of its users. Users are contractually allowed to do the same. The GNC is in this sense the “manufacturer” of the bandwidth and the FNO and NRN’s are considered “distributors” of bandwidth. NRNs typically lease a set of connections for large periods of time with a GNC. The FNO is allowed to subdivide this bandwidth between NRNs. The GNC is therefore in our example not concerned with the actual usage.

4.4.4 A token request and service ID.

A source NRN is typically the requestor, but in theory this would not be a requirement. One can imagine applications need bandwidth between A and B and subsequently between B and C. Source NRN A is allowed to make reservations for bandwidth between B and C. A NRN does not need to know to whom this bandwidth has been allocated. A NRN will trust the FNO for this. The involved parties will however want to know about a unique service ID that the FNO has assigned to the allocated timeframe. The service ID may also have sub-ID’s that will point at a particular time-slot within an allocated timeframe. Based on this service information, the FNO will generate a number tokens. Each token will point at an individual time-slot and reference to a specific link within the allocated timeframe. The tokens receive a cryptographic proof of authenticity from the FNO that each NRN will recognize. A token will also point at a reservation for a specific link. When a token is received, it is the responsibility of the NRN to map this pointer to a particular a link.

4.4.5 Receiving a token.

Before receiving a token, a domain could have already prepared all the necessary control parameters for each individual switch component in advance. There will be some optimal time period for doing these preparations before a

token is expected. This time will for example depend on the likelihood of failure between the preparation and the actual usage of the link. One may also determine an earliest time to receive a token and assign significance to the token. Tokens received before such time should be ignored. As shown already in fig. 2, once the token is recognized, a domain specific network management system could provision all involved switches with the correct configuration information as to establish a connection. A NRN should also maintain a resource management system to be able to administer network allocations and this system should also be able to generate the necessary element configuration information so that a network path could be activated very rapidly. The FNO, who in our case does the resource management for the GNC links, should allocate an inter-NRN link and should identify the selected link to the corresponding NRNs such that the correct inter-NRN link is used.

4.4.6 Distribution of tokens.

If the NRN, on behalf of the user community, obtains a set of tokens from the FNO, the NRN has the right to distribute these tokens according to its own set of rules to one (or more) of its customers. The NRN however cannot further subdivide a token, as this would mean a security breach for the token. The NRN should distribute tokens by some secure means to its user, ensuring confidentiality and avoiding unwanted duplication of tokens. Intermediate parties could hand copies of the same token to more than one of its users. Such action would allow a connection to be shared. In our simple use case, a there is no binding of token to the identity of a certain user. Anybody in the possession of the token is therefore able to use a token. The token must however be used from the source domain. A policy may determine if the token is accepted on certain ports. Duplicate token usage from different ports may also be denied based on a policy to prevent sharing.

4.4 Token requirements

The described use case raises the question on the kind of information that should be contained within the token and how a token should be secured. It is not the objective of this paper to cover this topic in detail as much research is left to be carried out. Here is merely a group of

requirements towards such a token. At token must contain:

- some proof of authenticity that is recognized by multiple service domains.
- The validity period (start- and end-time) of the token.
- An optional (encrypted) list of service domains in which the token is valid.
- A unique reference number that can be used for accounting purposes and allow providers to collect this information.
- A reference to a pre-established service instance. This service instance references all necessary information to instantiate a connection.

Towards usage the following requirements can be given:

- Tokens can be send one or multiple times. If received more then once, the additional usage re-asserts the usage right.
- If the service fails within the designated contract period, the token should be stored with this
- Token lifetime should be limited and granular enough to support application demand.

- Tokens should be allowed to be send some prior time to the validity of the token in order to maintain uninterrupted service.
- If no token is received, the connection between two domains may be based on best effort service.
- A token will always be received on a network ingress point that initially always connects to a best effort routed or VPN service. The token will cause the network to switch to a maximum bandwidth service.
- If the validity of the token is expired and no new token is received, the network will revert automatically to the best effort service.

4.5 Example network

Consider below network where end stations A and B want to communicate via a maximum bandwidth connection. A is connected to NRN-X via a single Gigabit Ethernet connection and station B is connected to NRN-Y via a similar connection. NRN-X and NRN-Y are connected with multiple connections via a GNC.

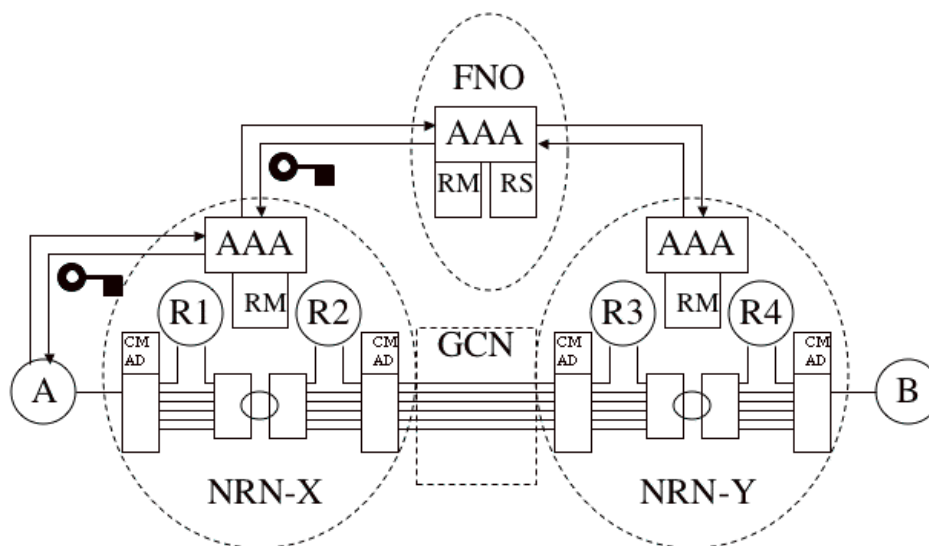


Fig 3: Example architecture featuring two NRN's interconnected with through a global network carrier network where the federative network organization is responsible for the resource management of the GNC.

Each NRN has a core transport network (e.g. a SDH ring). Switches at the ingress (client) and egress (GNC) side of the network determine the kind of connectivity offered. By default, the network offers a best effort services. The ingress/egress switches are also capable of accessing the core network directly. By default station A is connected via a VLAN to the router R1 of NRN-X. The NRNs transport infrastructure will interconnect R1 and R2. The egress switch of NRN-X and ingress switch of NRN-Y make sure that the two border routers R2 and R3 of each NRN are interconnected. The transport infrastructure of NRN-Y will interconnect R3 and R4 of NRN-Y and the ingress switch of B will connect R4 to B. Each ingress and egress switch of an NRN has a CMAD to monitor every port of the network for incoming tokens.

The architecture of an AAA server is subject of research by the AAA Arch RG. RFC2903 [13] describes such a architecture. An important aspect of an AAA server is the fact that it uses a driving policy to consider policy conditions and to take subsequent policy actions. The issuing of tokens is the responsibility of the FNO's AAA server. Upon receiving a request message from an NRN, the driving policy causes the AAA server to first contact a routing service (RS) that is capable of identifying one or more routes involving one or more (NRNs). Each NRN will advertise all available routes to this RS. Then the driving policy will contact all the involved resource managers (RM) to determine if a particular connection is available at the desired time.

First it will check with its own resource manager. The FNO RM overlooks the usage of the GNC links between the NRNs. If a connection between the two NRNs is available, the driving policy will contact both NRN RMs of the source and destination domain via the NRNs AAA server. It will request if a transport connection is available from the egress switch connecting to the GNC to the ingress switch of A and B at the given time with the requested capacity. Each domain Resource Manager will provide this data, but a policy at the AAA server may still permit or deny a particular use. If both domains answer are positive, the FNO's AAA server will generate and sign the requested amount of tokens and send them to the AAA server of the requesting domain. It will also contact the RMs make an allocation of the appropriate resources with the proper ID and will

send a copy of signed the token which can be compared with the token received from the user. The requesting NRN's can store these tokens and issue them to a NRN user upon a request or the NRN can act on a users request directly depending on the model. The NRN could already have known the need based on a contract with the user or the user could have ad-hoc demands. If the user receives one or more tokens from the NRN it may further distribute the tokens to its applications, which will then insert the token into the network via an in- or out of band method.

4.6 Network considerations.

The pictured method of creating a Layer 1 or Layer 2 bypass connection after regular communication between station A and B goes via a routed connection is fairly trivial if one takes care of the proper end station configuration and the proper support at the router network. Firstly the end stations must think that they are always talking to each other via a L2 network. This means that the default gateway of a station must be set to its own address, causing the station to ARP [14] for every destination IP address. The routed network must therefore support proxy ARP, which is typically enabled by default. If the network bypass connection is created and put into effect, the end station ARP caches need to be flushed to re-learn the MAC – IP address association. This also needs to be repeated whenever the by-pass connection reverts.

5.0 Conclusions.

The token model allows control of network resources involving different domains. The token represents the right to use a pre-established network connection at a specific time domain. The generation of a token for a future usage right, allows trading of the token before between the time a token is generated and the time the token is used. This fact allows various trading models. One organization could order a bulk of tokens and resell and/or distribute the usage rights. As the token is not bound to a particular user, the user is responsible for maintain the security of the token. Modern hardware devices present in switches can be programmed to recognize tokens in the data stream. This may omit complex signaling interfaces. Switches at the ingress points or at a central point of a network could be holding such monitor and action device. The user must periodically insert a

valid token in the data-stream keeping the network by-pass connection alive at intervals. This user may or may not be the same user as the previous user inserting a token. Policies at individual domains may restrict the usage of a token. In this paper we have not tried to describe a detailed solution as this is for future study.

6.0 References

- [1] <http://www.aaaarch.org>
- [2] RFC2904, Vollbrecht J., Calhoun P., Farrell S., Gommans L., Gross G., de Bruijn B., de Laat C., Holdrege M., Spence D., "AAA Authorization Framework", August 2000.
- [3] <https://forge.gridforum.org/projects/authz-wg>
- [4] RFC1994, Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", August 1996.
- [5] RFC 1172, Perkins D., Hobby R."The Point-to-Point Protocol (PPP) Initial Configuration Options". July 1990.
- [6] RFC 2865, Rigney C., Willens S., Rubens A., Simpson W., "Remote Authentication Dial In User Service (RADIUS)", June 2000.
- [7] RFC 2607, Aboba B., Vollbrecht J., "Proxy Chaining and Policy Implementation in Roaming". June 1999.
- [8] Internet2 Qbone bandwidth broker project: <http://qbone.internet2.edu/bb>
- [9] RFC2474, Nichols K., Blake S., Baker F., Black D., "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", December 1998.
- [10] RFC3281, Farrell S., Housley R., "An Internet Attribute Certificate Profile for Authorization", April 2002.
- [11] Recommendation X.509, "Public-key and attribute certificate frameworks", www.itu.int.
- [12] de Laat C., Radius E., Wallace S., "The Rationale of the Current Optical Networking Initiatives", iGrid2002 special issue, Future Generation Computer Systems, volume 19 issue 6 (2003)
- [13] RFC2903, de Laat C., Gross G., Gommans L., Vollbrecht J., "Generic AAA Architecture", Augustus 2000.
- [14] RFC826, Plummer D., "An Ethernet Address Resolution Protocol", November 1982.