Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 49 (2005) 449–464
Computer Networks
www.elsevier.com/locate/comnet

Performance analysis of a Denial of Service protection scheme for optimized and QoS-aware handover

Tianwei Chen [a], Michel Sortais [b], Günter Schäfer [a,*], Stefan Adams [c], Changpeng Fan [d], Adam Wolisz [a]

[a] *Fachgebiet Telekommunikationsnetze, Technische Universität Berlin, Germany*
[b] *Math. Institut, Fak. II MA 7-4, Technische Universität Berlin, Germany*
[c] *Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany*
[d] *ICM N PG SP RC PN, Siemens AG, Germany*

Available online 9 June 2005

Abstract

Quality of Service (QoS) mechanisms in networks supporting mobile Internet communications give rise to Denial of Service (DoS) threats: if the network cannot efficiently check the credibility of a QoS request during a handover process, malicious entities could flood the network with bogus QoS requests; if the authentication check is performed by means of an AAA protocol before the access network commits its resources, the authentication process may not only introduce a notable latency to the handover process, but also generate an extensive traffic in the presence of malicious requests, thus causing the network signaling capacity to degrade. In order to defend against these kinds of attacks and meet the low-latency micro-mobility handover requirement, we propose a preliminary authentication check with a cookie-based mechanism before processing the requests and performing authentication and authorization. Our performance evaluation shows that the cookie-based mechanism is efficient in dealing with the identified issues.

Keywords: QoS; Mobility; DoS; Cookie protection; Performance analysis

1. Introduction

In IP-based mobility scenarios, the handover signaling and *Quality of Service* (*QoS*) provisioning, which aim to guarantee certain service characteristics like end-to-end delay and jitter in a handover process, give rise to new threats that these mechanisms could be abused by malicious

* Corresponding author.
E-mail addresses: chen@tkn.tu-berlin.de (T. Chen), sortais@math.tu-berlin.de (M. Sortais), schaefer@tkn.tu-berlin.de (G. Schäfer), adams@mis.mpg.de (S. Adams), changpeng.fan@icn.siemens.de (C. Fan), wolisz@tkn.tu-berlin.de (A. Wolisz).

1389-1286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2005.05.015

entities to launch so-called *Denial of Service* (*DoS*) attacks, which aim at reducing the availability of services to legitimate users.

In an IP-based access network, a mobile node (MN) sends a request to an access router (AR) for a certain resource (see Fig. 1 for an illustration). If the network cannot check the credibility of a QoS request (i.e., whether the request originates from an MN that is actually authorized to use the services it is requesting), malicious entities could flood the network with bogus QoS requests in order to cause the exhaustion of the available resources through temporal reservations. This represents one specific DoS threat.

A potential solution consists of the following procedure: when an access router (AR) receives a QoS request, before starting the resource reservation process, the AR communicates with a security authority, e.g., an AAA (Authentication Authoriostic and Accounting) server, to authenticate the MN and authorize the QoS request [1]. AR continues with the reservation process only once this security check passes. However, the latency introduced by proceeding to security checks at the AAAL (Local AAA), which includes the contribution of the propagation delay and processing time at AAAL, is not desirable when low latency of the registration process is a major concern. Moreover, the same checks at an AAA server have to be performed on all the bogus requests from attackers. Thus all the security check signaling may degrade the performance of the access network substantially by depleting the signaling capacity of the path between the AR and the AAAL and exhausting the computing resource of the AAAL. This represents another specific DoS threat.

To defend against the identified DoS threats and meet the low-latency requirement in intra-domain handovers, we propose a two-step procedure comprising of one preliminary credibility check, after which processing of the signaling request is either aborted or continued, and the second definitive authentication check as described above. The credibility check should have the following properties: performing the first check must be a quick operation, and ARs must not keep per-session or per-user state until the verification is complete.

Up to now, two principal approaches for checking the credibility of a request have been proposed: *exchanging* "*cookies*" and *solving* "*client puzzles*". The concept of exchanging "cookies" has been introduced in the context of transactions between Web servers and browsers, where cookies are pieces of information generated by a Web server and stored in the user's computer, ready for future access. This idea has been adopted to provide protection against resource exhaustion DoS attacks in IPSec's key exchange protocol ISAKMP and the mobility support in IPv6 design [10]. In the "client puzzle" approach [3], a client is asked to solve a cryptographic puzzle and the server stores the protocol state and executes expensive operations only

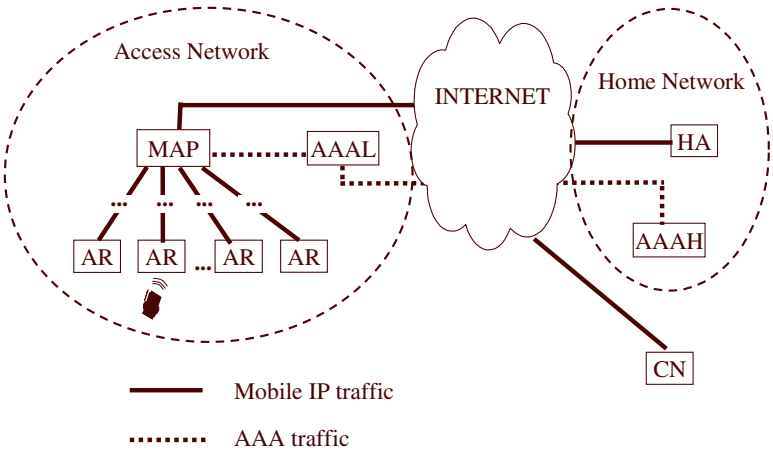


Fig. 1. An overview of the involved entities.

after it has verified the client's solution. In this way, the puzzle can prevent intensive connection initiations from attackers, thus enhancing the DoS-resistance of a server. However, solving cryptographic puzzles imposes a computational burden on all legitimate clients, as well as requiring additional message elements to be exchanged. It would add a non-negligible latency to the establishment of a connection between client and server.

In this paper, we therefore propose to use a cookie-based mechanism as the first credibility check. A cookie is verified by an AR to ensure that the QoS request sender is a credible registered user before processing the requests and performing authentication and authorization. This preliminary check enables us to prevent DoS attacks, both in the form of resource reservations along a path or keeping the AR busy through the processing of malicious QoS requests, with the help of an AAAL or possibly the home AAA server (AAAH).

Our use of the cookie idea as a first step of authentication—a preliminary check of a request's credibility—is completely new in IP-based mobile networks. After AR filters out most of the bogus requests without cookie or with false cookies by means of cookie verification, the access network has a lower burden in the authentication of QoS requests—the second step of authentication. Since it cannot be completely ruled out that an attacker gains access in the network with an eavesdropped cookie, it is necessary to perform the second step authentication with the help of the local AAA server.

The remainder of the paper is organized as follows: Section 2 introduces related work; Section 3 describes the cookie-based mechanism in detail; Section 4 contains a comparative performance analysis using queueing theory. An overview of the corresponding results is given in Section 5, and Section 6 summarizes our contribution.

2. Related work

So far the most common form of DoS is to cause excessive bogus traffic to a particular server so as to prevent legitimate users from getting services. To deal with this kind of DoS attack, several defense techniques have been published, see, e.g., [3].

As mentioned above, the basic idea of cookies is adopted in the mobility support in IPv6 design. The correspondent nodes (CNs) do not have to retain any state about individual MNs until an authentic binding update (BU) arrives. When receiving a BU message, the CN includes a cookie in a message sent back to the sender according to the source address in the in-coming BU message. If the source address is not a bogus one from an attacker, the genuine sender will include the cookie in its following messages to the CN. It is stated that the cookie mechanism can protect the CN against memory exhaustion attacks except where on-path attackers are concerned. In contrast, our scheme is dealing with DoS attacks on the signaling capacity of the access network, a problem which has not been addressed previously.

Currently QoS, DoS and low-latency handovers have been dealt with separately. Koodli proposed a fast handover approach to achieve seamless mobility [8]. It can also be integrated with the Context Transfer (CT) protocol for QoS support [11]. However, it did not address the DoS threat.

In principle, in order to address the first aforementioned DoS threat, an authentication must be performed when the access network receives a QoS request. In addition to the fact that the authentication check is performed by an AAA protocol [5], which might post the second aforementioned DoS threat, the authentication check can be performed by the Context Transfer protocol [12]: the authentication data is transferred from the old AR to the new AR during a handover. According to a performance analysis work in [7], the authentication process is not optimized in non-predictive handovers since the new AR needs to communicate with the old AR for the authentication data. Moreover, CT cannot be useful in establishing a new QoS path.

In the signaling design efforts of the Next Steps in Signaling (NSIS) working group in IETF, DoS attacks have been identified [6,15]. However, currently no scheme for QoS path establishment in mobility scenarios has addressed the DoS and handover optimization issues efficiently and practically.

3. A cookie-based mechanism

Fig. 1 shows an overview of a network with a hierarchical Mobile IPv6 (HMIPv6) [14] and Authentication, Authorization and Accounting (AAA) [5] joint architecture.

During the intra-domain handover procedure, the MN needs to find out whether the path from the new AR to MAP can meet its QoS request. Each router along this path in the access network must determine whether it has sufficient resources to satisfy the per-hop QoS requirement of an MN's session. If the path from the new AR to MAP can satisfy the QoS request, it reserves the resource for the related session. Before the resource availability check and reservation, an efficient authentication check is mandatory to prevent DoS attacks and achieve low-latency handovers.

The cookie-based mechanism is designed as an efficient authentication scheme meeting the following requirements: an AR can verify a cookie immediately upon receiving a QoS request; the cookie generation and verification must be lightweight; the entities in the access network are free from keeping per-client states until the cookie check passes. In the following, we first describe the data structure of a cookie and the mechanism operations, then we give a discussion and a summary.

3.1. The data structure of a cookie

Fig. 2 shows the data fields of a cookie. The purpose of each field may be listed as follows:

- **MN_ID** is the MN's unique identifier. This can be a local unique identifier the MN gets after its first registration.
- **Gen_ID** identifies the cookie generator which is always an AR. It can be the AR's IP address or another unique identifier acceptable in the access network.

Fig. 2. Cookie data fields.

- **Creation_T** is the timestamp marking when the cookie was generated. It is used to limit the cookie's period of validity.
- **Random_Nr** is used to distinguish two cookies which are generated at the same time.
- **CookieHash**: The hash code is a message digest of the cookie information and a cookie key. The hash function could, for example, be of either HMAC-MD5 or HMAC-SHA1 (note that collision resistance [13, Section 9.2.2] is not required for the creation and verification of cookies). The cookie key could be distributed from the MAP to each AR and updated by the MAP periodically. For example, a new cookie key could be distributed by the MAP every hour or day.

In summary, a cookie is defined according to the following formula:

$$\begin{aligned} CookieInfo &:= (Identity_{\mathrm{MN}},\ Identity_{\mathrm{AR}i},\\ &\qquad Timestamp_{\mathrm{AR}i},\ RandomNumber_{\mathrm{AR}i}),\\ CookieHash &:= HMAC(CookieKey,\ CookieInfo),\\ Cookie &:= (CookieInfo,\ CookieHash). \end{aligned}$$

3.2. Mechanism operations

The cookie-based mechanism is described in an access network which is based on a HMIPv6 and AAA joint architecture. The architecture includes an AAAL, a mobility anchor point (MAP) and ARs positioned linearly as shown in Fig. 4.

- *First cookie generation*: When a mobile user enters an access network (e.g., it performs a global movement or powers up), the authentication on its first QoS request must involve a trusted network entity (e.g., in the mobile user's home network) because the user is unknown to the access network at the moment.

As shown in Fig. 3, when an AR receives a QoS request from an MN, it first sends a message to the AAAL server for a security check. Since the MN is unknown in the access network, AAAL sends a message to the MN's AAAH for the authentication check [4]. After the authentication is done successfully at AAAH, the access network knows that the user is credible; AAAL caches the MN's

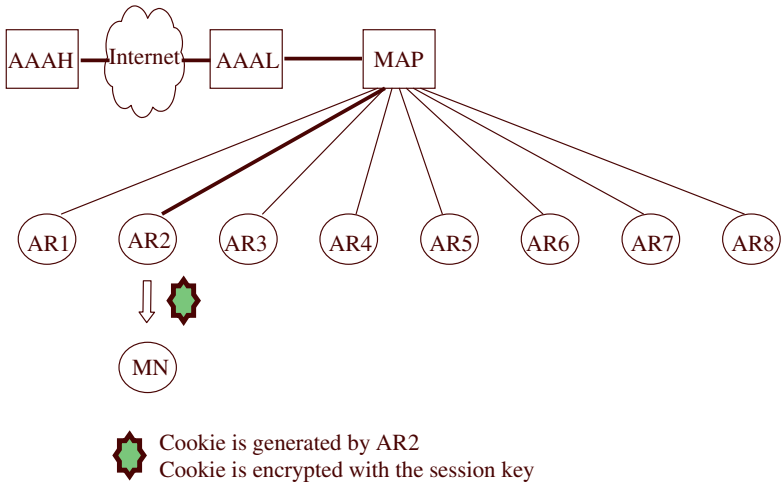


Fig. 3. MN obtains its first cookie from the access network.

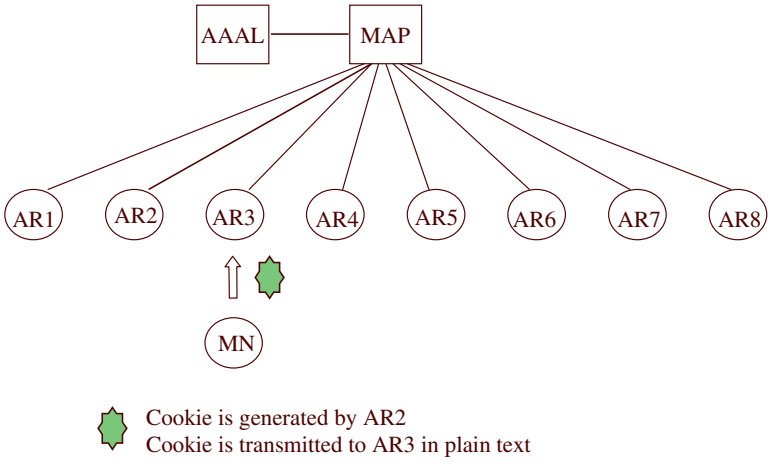


Fig. 4. AR3 verifies the cookie presented in plain text from the MN.

authorization information and MAP, AR2 (taken to be the associated AR in our example, see Fig. 4) and MN now know the session key which was generated by the MN's home domain.

AR2 generates a cookie, encrypts the cookie with the session key, inserts the encrypted cookie in the BU acknowledgement (BU ACK) message which is generated by MAP and destined to MN. The MN can get its first cookie in the access network since the MN can derive the session key due to its long term trust relationship with its home domain. Thus, MN gets its first cookie in the access network.

- *Cookie verification*: In a local movement, as shown in Fig. 4, MN presents the cookie to a neighboring AR server (say AR3). Because there is no security association between MN and AR3 so far, the cookie is transmitted in plain text.

When receiving the cookie, AR3 first performs a cookie verification. Each AR has a *trusted list*, a *trusting list* and a *notified cookie list*. The *trusted list* contains the ARs from which an AR can

accept cookies and verify them; the *trusting list* includes the ARs which can accept and verify its cookies. These notions will be discussed in more detail later.

The cookie verification includes the following operations:

- check the timestamp in the cookie to verify the cookie has not expired;
- check the identity of the cookie generator to verify the cookie was created by an AR on its trusted list;
- verify that the cookie is not on the notified cookie list (this list contains all the used cookies which may be accepted and verified by it);
- if the above checks pass, AR3 computes a key-hashed digest of the cookie information by using a cookie key, and compares the computation result with the hash digest contained in the cookie. If the two hash digests match, the cookie check verification is completed successfully.

After verifying a cookie successfully, AR3 informs AR2 who originally generated the cookie that it has the cookie. AR2 then notifies all other ARs on its trusting list to invalidate the cookie, preventing these ARs from accepting it again; indeed, an attacker could intercept the cookie from the open wireless interface and replay it to cheat these ARs for access. After the expiration of a cookie's lifetime, all ARs can delete it from the notified cookie list.

• *New cookie granting*: When the cookie verification is completed successfully, QoS + BU [9] and authorization processes start. The two processes can proceed in series or in parallel. If the two processes are successful, the AR3 can verify the authenticity of the registration request upon receiving the session key embedded in the BU ACK message from the MAP. When this check passes, the AR3 generates a new cookie, encrypts it with the session key and inserts it in the BU ACK message as shown in Fig. 5. The new cookie is used for its next registration and the old cookie is no longer valid in the access network.

3.3. Discussion

There are three main points in the design of this cookie mechanism: the presentation of a cookie in plain text, the notification of a used cookie, and the "area of validity".

- *Presenting a cookie in plain text*: Mobile nodes always send cookies in plain text to access routers, as in the case of a handover, since the MN does not yet share a session key with the new access router at the time it sends the cookie. Although the cookie might be intercepted by an attacker when being presented in plain text, the risk of DoS attacks is reduced sufficiently. The reason for this is that the cookie mechanism reduces the overall number of "credible looking" handover requests, since every cookie can only be presented once.
- *Notification of used cookies*: After verifying the cookie, AR3 notifies immediately AR2 about the cookie use and then AR2 notifies AR1, who is the remaining AR on AR2's trusting list, not to accept the cookie. All ARs are then free

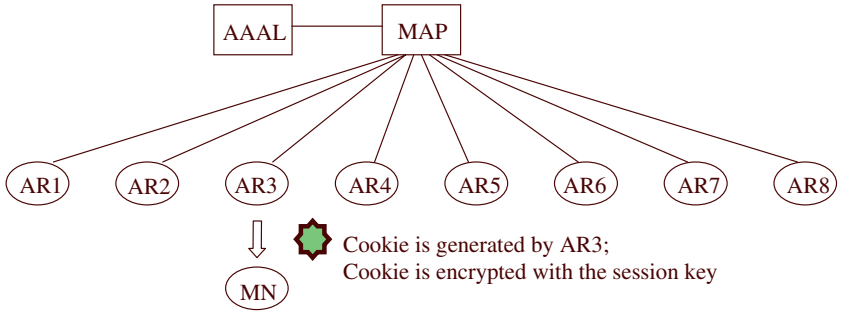


Fig. 5. MN obtains a new cookie after the re-registration procedure.

from replay attacks, provided that the notification messages propagate faster than the time needed for an attacker to intercept a cookie and replay it.

- *Area of Validity*: The "area of validity" is a group of ARs in which a cookie is valid. In other words, the "area of validity" corresponds to the trusting list of the cookie generator.
  When a cookie is presented in plain text to AR3 (see Fig. 4), if AR3 were not to notify other ARs about the use of the cookie, the cookie could be intercepted by an attacker in open air and replayed to other ARs. Consequently, the attacker could gain access at these ARs so as to make DoS attacks on the corresponding paths as shown in Fig. 6. On the other hand, if each AR were to notify the rest of the access network when receiving a cookie, the propagation of notification messages would generate substantial traffic in the access network.
  Therefore, we introduce a limited "area of validity" for each cookie, the nodes in this area being the only ones that can accept the cookie. For example, each AR could have its adjacent ARs only on its *trusting list* and *trusted list*. AR2 would put AR1, AR3 and itself on its *trusting list* to form the "area of validity", meaning that the cookies generated by AR2 are only accepted by AR1, AR2 and AR3, while being rejected by other ARs (see Fig. 7).

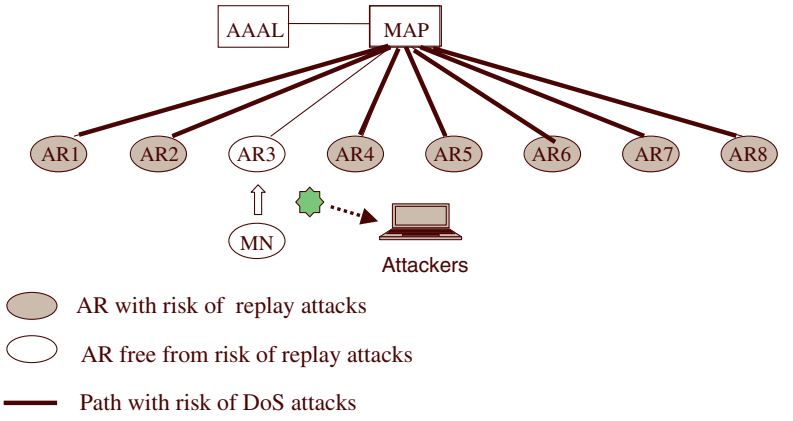


Fig. 6. Threat of replayed cookie without limited AOV and notification.

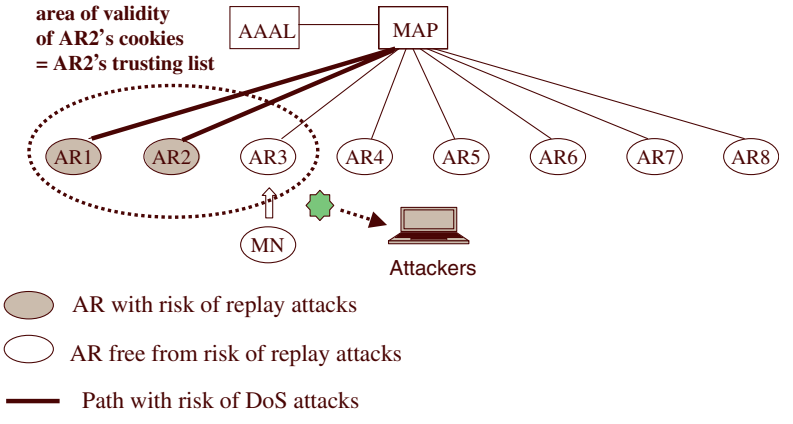


Fig. 7. Threat of replayed cookies with limited AOV and without notification.

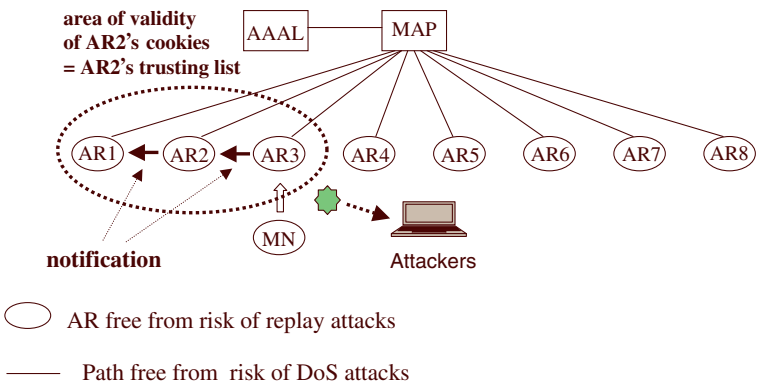


Fig. 8. Replay protection with limited "AOV" and cookie notification.

Necessarily, both AR1 and AR3 have AR2 on their own *trusted list* and they can accept cookies generated by AR2.

- *Notification of used cookies*: As shown in Fig. 8, after verifying the cookie, AR3 immediately notifies AR2 about the cookie use and then AR2 notifies AR1, who is the remaining AR on AR2's *trusting list*, not to accept the cookie. All ARs are then free from replay attacks, provided that the notification messages propagate faster than the time needed for an attacker to intercept a cookie and replay it.

4. Performance evaluation

In order to evaluate the performance of the cookie-based mechanism in reducing the risks of DoS attacks and in benefitting from the optimized intra-domain handovers, three parameters should be examined:

- *Mean response time*: The duration between the transmission of the first bit of a registration request of a legitimate MN and the arrival of the last bit of the corresponding registration response.
- *Mean queue length at AAAL*: How long in average is the queue of new jobs waiting for service at the AAAL server? In the absence of a cookie protection or with such a protection? This metric indicates that in the absence of a cookie protection, there are many conditions under which DoS occurs through an overflow of traffic at AAAL, whereas the use of a cookie mechanism enables one to prevent completely such DoS (under the same hypothesis regarding the attacking and the frequency of legitimate MN requests).
- *Size of the waiting room at AR*: Another disadvantage of the no-cookie scenario lies in the fact that the AR server has to store some (e.g., 500 bits long) jobs until receiving an answer from AAAL, and many of these jobs may actually correspond to false requests. Considering the AR server of a cell that is under attack, we use a (continuous time) Markov chain method in order to compute the mean value of the total queue length in such a "waiting room".

These parameters are examined in three different processing schemes as shown in Figs. 9–11, respectively:

- *Case 0. No cookie protection*: In a given cell (cell $\sharp m$) which has one AR, MNs and attackers

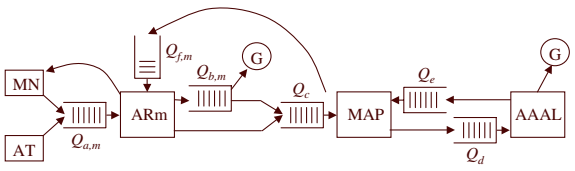


Fig. 9. Processing scheme in Case 0.

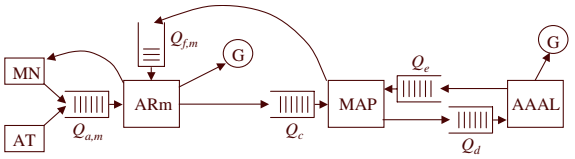


Fig. 10. Processing scheme in Case 1.

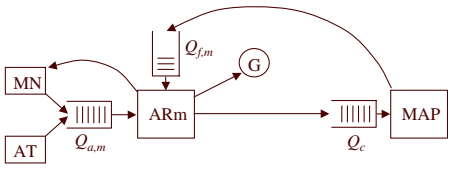


Fig. 11. Processing scheme in Case 2.

(AT) send QoS requests in registration messages to the AR (AR_m). The arrival messages are queued in $Q_{a,m}$ of AR_m if the processor is busy. Since (AA) checks need to be performed at AAAL before initiating the resource reservation process, AR_m caches temporarily the corresponding QoS request information in $Q_{b,m}$ (which serves as a "waiting room"). When a reply message for a genuine request arrives at AR_m, AR_m discards all the priors in $Q_{b,m}$ since these are regarded as bogus requests, provided all the (AA) checks at AAAL require the same time.

- *Case 1. Cookie protection and the resource reservation and authorization processes are in series*: In the AR_m's cell, MNs and ATs send QoS requests with cookies to $Q_{a,m}$ of AR_m. Before processing the requests, AR_m verifies the cookie. If the verification fails, AR_m drops the request in "G" (denoting garbage) silently. If the verification succeeds, AR_m sends the notification message and starts the resource reservation procedure immediately, checking its available bandwidth and sending the QoS request to Q_c of MAP. MAP performs the same check. Before MAP sends a reply destined to MN, including the check result and the session key, to $Q_{f,m}$ of AR_m, it generates a new message and sends it to Q_d of AAAL for the authorization check. When the authorization check passes, AAAL sends a message to Q_e of MAP. When AR_m receives the reply from MAP, it performs an authentication check to the QoS request with the session key. If the check passes, while removing the session key from the reply message, it generates a new cookie, encrypts it with the session key, inserts the encrypted cookie in the reply, and forwards it to the corresponding MN.
- *Case 2. Cookie protection and the resource reservation and authorization processes are in parallel*: AR_m performs the same cookie verification when receiving a QoS request from either a MN or an AT. If the verification fails, AR_m also drops the request to "G". If the verification passes, AR_m starts the two processes in parallel. Meanwhile, it sends the notification message. It is assumed that the result of the authorization process arrives earlier than that of the resource reservation process. Therefore, the time spent on the authorization process has no contribution to the response time of a registration process and the authorization process can be ignored from the analysis.

In all three cases, λ_{MN} denotes the Poisson rate of messages, sent by MNs in a given cell, whereas λ_{AT} denotes the Poisson rate of messages sent by Attackers in a cell that is currently under attack. We also let M denote the total number of cells, whereas N stands for the total number of cells that are currently under attack. The message sending process in a given cell (cell $\sharp m$, $1 \leqslant m \leqslant M$) is therefore a Poisson process with intensity λ that is either equal to λ_{MN} (no attack) or to $\lambda_{\mathrm{MN}} + \lambda_{\mathrm{AT}}$ (attack).

The asymptotic mean arrival rate at $Q_{a,m}$, namely $\lambda^*_{a,m}$, is defined as:

$$\lambda^*_{a,m} = \lim_{t \to +\infty} \times \frac{\mathbb{E}[\sharp(\text{arrivals at } Q_{a,m} \text{ during time } [0,t])]}{t},$$

and we let λ^*_c, λ^*_d, λ^*_e and $\lambda^*_{f,m}$ be the asymptotic mean arrival rates corresponding to Q_c, Q_d, Q_e and $Q_{f,m}$ respectively.

The asymptotic mean arrival rates at each of the remaining queues may then be determined as functions of M, N, λ_{MN} and λ_{AT}.

Indeed, in case 0:

$$\lambda_c^* = 2M\lambda_{\mathrm{MN}} + N\lambda_{\mathrm{AT}},$$

$$\lambda_d^* = N(\lambda_{\mathrm{MN}} + \lambda_{\mathrm{AT}}) + (M-N)(\lambda_{\mathrm{MN}}),$$

$$\lambda_e^* = M\lambda_{\mathrm{MN}}, \quad \lambda_{f,m}^* = 2\ \lambda_{\mathrm{MN}};$$

in case 1: $\lambda_c^* = M\lambda_{\mathrm{MN}} = \lambda_d^* = \lambda_e^*, \lambda_{f,m}^* = \lambda_{\mathrm{MN}}$, and in case 2: $\lambda_c^* = M\lambda_{\mathrm{MN}}, \lambda_{f,m}^* = \lambda_{\mathrm{MN}}$.

4.1. Waiting times at different queues, total response time

Consider the random variables $W_{a,m}^{(k)}$ and $R_{ARm}^{(k)}$ defined by:

$W_{a,m}^{(k)}$:= total time spent waiting in queue $Q_{a,m}$ by job $\sharp k$, and
$R_{ARm}^{(k)}$:= residual service time in AR_m upon arrival of kth job.

Define $W_{b,m}^{(k)}$, $W_c^{(k)}$, $R_{\mathrm{MAP}}^{(k)}$, $W_d^{(k)}$, $R_{\mathrm{AAAL}}^{(k)}$, $W_e^{(k)}$ and $W_{f,m}^{(k)}$ in a similar way (job $\sharp k$ in Q_c is the kth job having been stored in Q_c). We are interested in limits such as

$$\lim_{k\to+\infty} \mathbb{E}(W_{a,m}^{(k)}),$$

and we simply write $\mathbb{E}(W_{a,m})$ for it in the sequel, calling it the mean queue length or mean waiting time at $Q_{a,m}$ (refer to Table 2 for a list of processing time parameters).

It turns out that these mean waiting times may be evaluated by using two basic principles from Queueing Theory: *Little's theorem* and the *Pollaczek–Khinchine formula* (as applied to priority queueing systems, see, e.g., [2]). As an example, let us outline the computation of the mean waiting time at Q_d in case 0 and in case 1. First, according to Little's theorem, the mean number of jobs waiting for service in Q_d is given by

$$\mathbb{E}(L_d) = \lambda_d^* \mathbb{E}(W_d).$$

Secondly, denoting by $\mathbb{E}(\omega_d)$ the mean time needed for the execution of a single job queued through Q_d, we also know that $\mathbb{E}(W_d)$ relates to the mean residual service time $\mathbb{E}(R_{\mathrm{AAAL}})$ in such a way that

$$\mathbb{E}(W_d) = \mathbb{E}(L_d)\mathbb{E}(\omega_d) + \mathbb{E}(R_{\mathrm{AAAL}}).$$

In case 0, the time needed for the execution of a job depends on the authenticity of the corresponding request, so that

$$\mathbb{E}(\omega_d) = \frac{N\lambda_{\mathrm{AT}}}{M\lambda_{\mathrm{MN}} + N\lambda_{\mathrm{AT}}} T + \frac{M\lambda_{\mathrm{MN}}}{M\lambda_{\mathrm{MN}} + N\lambda_{\mathrm{AT}}} (t_3 + T),$$

whereas in case 1 we simply have $\mathbb{E}(\omega_d) = t_3$.

Using Little's theorem thus yields:

$$\mathbb{E}(W_d) = \frac{\mathbb{E}(R_{\mathrm{AAAL}})}{1 - \lambda_d^* \mathbb{E}(\omega_d)},$$

so that it just remains to compute $\mathbb{E}(R_{\mathrm{AAAL}})$, which is a simple application of the Pollaczek–Khinchine formula. Indeed, in case 0 we have

$$\mathbb{E}(R_d) = \frac{1}{2}\{(N\lambda_{\mathrm{AT}})T^2 + (M\lambda_{\mathrm{MN}})(T + t_3)^2\},$$

whereas in case 1:

$$\mathbb{E}(R_d) = \frac{(M\lambda_{\mathrm{MN}})^2}{2} t_3^2.$$

The total response time τ defined in the preceding subsection has an asymptotic mean value which may in turn be presented as a combination of mean waiting times and deterministic times. Indeed:

◊ In case 0, one has

$$\begin{aligned}\mathbb{E}(\tau) = {} & C_{1,2}^{(0)} + \mathbb{E}(W_{a,m}) + t_1 + C_{2,3}^{(0)} + \mathbb{E}(W_c) + C_{3,4}^{(0)} \\ & + \mathbb{E}(W_d) + t_3 + T + C_{4,3}^{(0)} + \mathbb{E}(W_e) + 2t_2 \\ & + C_{3,2}^{(0)} + \mathbb{E}(W_{f,m}) + 4t_4 + \hat{C}_{2,3}^{(0)} + \mathbb{E}(W_c) \\ & + \hat{C}_{3,2}^{(0)} + \mathbb{E}(W_{f,m}) + C_{2,1}^{(0)}.\end{aligned}$$

$C_{1,2}^{(0)}$ being the transmission time for the wireless up-link channel, the further $C_{i,j}^{(0)}$'s being transmission parameters for the further messages and links, and t_1, t_2, t_3, t_4, T denoting some fixed processing time parameters (see Tables 1 and 2).

◊ In case 1, we have

$$\begin{aligned}\mathbb{E}(\tau) = {} & C_{1,2}^{(1)} + \mathbb{E}(W_{a,m}) + 4t_4 + 3T + C_{2,3}^{(1)} + \mathbb{E}(W_c) \\ & + C_{3,4}^{(1)} + \mathbb{E}(W_d) + C_{4,3}^{(1)} + \mathbb{E}(W_e) + t_3 + C_{3,2}^{(1)} \\ & + \mathbb{E}(W_{f,m}) + C_{2,1}^{(1)}.\end{aligned}$$

Table 1
Link and message length parameters

Parameter	Value
Wireless link MN ↔ AR	11/54 Mbps
Wired link AR ↔ MAP ↔ AAAL	100 Mbps
Wireless PHY + MAC Header	58 bytes
Wired PHY + MAC Header	26 bytes
IPv6 Header	40 bytes
QoS Hop-By-Hop Option	82 bytes
Home Address Option	18 bytes
BU/BU ACK	6 bytes
ESP Header	8 bytes
ESP Authentication Extension	16 bytes
Authenticator	20 bytes
Cookie	32 bytes (HMAC-SHA)

Table 2
Processing time parameters

Symbol	Time[a] (μs)	Remark
t1	152	Generate an AA request/answer
t2	20	Forward an AA request/answer
t3	152	Perform an authorization check
t4	220	Check, reserve or confirm resources
T	40	Perform authentication check; Generate or verify a cookie

[a] The processing time values are obtained from measurements on Pentium III 600 machines.

◊ In case 2, we have

$$\mathbb{E}(\tau) = C^{(2)}_{1,2} + \mathbb{E}(W_{a,m}) + 3t_4 + 3T + C^{(1)}_{2,3} + \mathbb{E}(W_c) + C^{(1)}_{3,2} + \mathbb{E}(W_{f,m}) + C^{(2)}_{2,1}.$$

4.2. Queue length at AAAL

A computation of the mean waiting time at Q_d was already outlined in the preceding subsection; one thus obtains for case 0:

$$\mathbb{E}(W_d) = \frac{(N\lambda_{\mathrm{AT}} + M\lambda_{\mathrm{MN}})\{(N\lambda_{\mathrm{AT}})T^2 + (M\lambda_{\mathrm{MN}})(T+t_3)^2\}}{2(1-\{(N\lambda_{\mathrm{AT}})T + (M\lambda_{\mathrm{MN}})(T+t_3)\})}$$

and for case 1

$$\mathbb{E}(W_d) = \frac{\frac{(M\lambda_{\mathrm{MN}})^2}{2}t_3^2}{1 - M\lambda_{\mathrm{MN}}t_3},$$

and from there on, the mean queue length $\mathbb{E}(L_d)$ (mean number of tasks buffered in Q_d) may be easily derived using Little's theorem.

4.3. Queue length in the "waiting room"

In the analysis of the length of the "waiting room" $Q_{b,m}$ of case 0, the only jobs that will be further processed are the "good jobs" corresponding to genuine requests.

On the basis of the scenario given for case 0, the asymptotic mean total time elapsed between the storage of a "good job" in $Q_{b,m}$ and its marking as a "good job" by the AA acknowledgement message is given by

$$\mathbb{E}(W_{b,m;\mathrm{good}}) = t_1 + C^{(0)}_{2,3} + \mathbb{E}(W_c) + t_2 + C^{(0)}_{3,4} + \mathbb{E}(W_d) + t_3 + T + C^{(0)}_{4,3} + \mathbb{E}(W_e) + t_2 + C^{(0)}_{3,2} + \mathbb{E}(W_{f,m}) + t_4.$$

Via Little's Theorem, the asymptotic mean number of "good jobs" waiting for service in $Q_{b,m}$ is given by

$$\mathbb{E}(N_{b,m;\mathrm{good}}) = \lambda^*_{b,m;\mathrm{good}}\mathbb{E}(W_{b,m;\mathrm{good}}) = \lambda_{\mathrm{MN}}\mathbb{E}(W_{b,m;\mathrm{good}}).$$

Taking also the arrivals of "bad jobs" (i.e., fake requests) into account, one may then express the asymptotic total length of $Q_{b,m}$ as

$$\mathbb{E}(L_{b,m}) = \lambda_{\mathrm{MN}}\mathbb{E}(W_{b,m;\mathrm{good}})\left(1 + l^{(1)}_{b,m}\right) + l^{(2)}_{b,m}.$$

As shown in Fig. 12, $l^{(2)}_{b,m}$ above denotes the asymptotic mean number of "residual bad jobs" in $Q_{b,m}$ (those "bad jobs" that are located below the lowest "good job" in $Q_{b,m}$), whereas $l^{(1)}_{b,m}$ stands for the asymptotic mean number of "bad jobs" that are located in between two consecutive "good jobs" in $Q_{b,m}$ ("intermediate bad jobs"). In Fig. 12 the number of "residual bad jobs" is 3; when a "good job" is marked through an AA

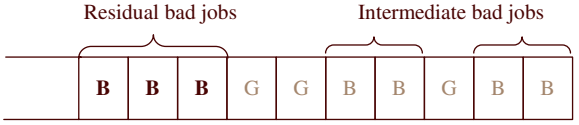


Fig. 12. Record of stored jobs in "waiting room".

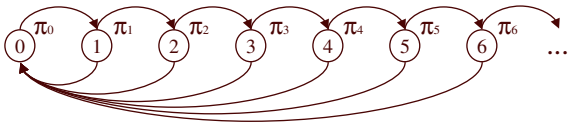


Fig. 13. The transition diagram for $\left(\rho_{b,m}^{(t)}\right)_{t\geqslant 0}$.

acknowledgement message, the corresponding data is taken away for further treatment, while the jobs prior to it are discarded.

The mean values $l_{b,m}^{(1)}$ and $l_{b,m}^{(2)}$ can be computed by introducing an appropriate continuous-time Markov chain $\left(\rho_{b,m}^{(t)}\right)_{t\geqslant 0}$ that is counting the "residual bad jobs" in the course of time. $\left(\rho_{b,m}^{(t)}\right)_{t\geqslant 0}$ has the transition diagram given in Fig. 13, and the corresponding transition rates are given by

$$\begin{aligned} r_{0,1} &= \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{P}\left(\rho_{b,m}^{(t+\mathrm{d}t)} = 1 \mid \rho_{b,m}^{(t)} = 0\right) \\ &= \lambda_{\mathrm{AT}} = r_{1,2} = r_{2,3} = \cdots, \\ r_{1,0} &= \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{P}\left(\rho_{b,m}^{(t+\mathrm{d}t)} = 0 \mid \rho_{b,m}^{(t)} = 1\right) \\ &= \lambda_{\mathrm{MN}} = r_{2,0} = r_{3,0} = \cdots. \end{aligned}$$

Let $\boldsymbol{\pi} = (\pi_0, \pi_1, \pi_2, \ldots)$ denote the invariant probability measure associated with this chain; according to the "global balance equations":

$$\pi_0 r_{0,1} = \pi_1 r_{1,0} + \pi_2 r_{2,0} + \cdots,$$

showing that

$$\pi_0 \lambda_{\mathrm{AT}} = (1-\pi_0)\lambda_{\mathrm{MN}}, \quad \pi_0 = \frac{\lambda_{\mathrm{MN}}}{\lambda_{\mathrm{MN}} + \lambda_{\mathrm{AT}}},$$

whereas

$$\pi_1(r_{1,0} + r_{1,2}) = \pi_0 r_{0,1}, \ \pi_2(r_{2,0} + r_{2,3}) = \pi_1 r_{1,2}, \ldots$$

so that altogether:

$$\pi_k = \frac{\lambda_{\mathrm{MN}}}{\lambda_{\mathrm{MN}} + \lambda_{\mathrm{AT}}}\left(\frac{\lambda_{\mathrm{AT}}}{\lambda_{\mathrm{MN}} + \lambda_{\mathrm{AT}}}\right)^k, \quad \forall k \geqslant 0.$$

The asymptotic mean number of "residual bad jobs" at $Q_{b,m}$ may thus be computed as

$$l_{b,m}^{(2)} = \lim_{t\to\infty}\mathbb{E}\left(\rho_{b,m}^{(t)}\right) = \sum_{k\geqslant 1} k\pi_k = \frac{\lambda_{\mathrm{AT}}}{\lambda_{\mathrm{MN}}}.$$

As for $l_{b,m}^{(1)}$, the asymptotic mean number of bad jobs separating two consecutive good jobs in $Q_{b,m}$, it may be seen to satisfy

$$l_{b,m}^{(1)} = \lim_{t\to\infty}\mathbb{E}\left(w_{b,m}^{(t)}\right),$$

where $\left(w_{b,m}^{(t)}\right)_{t\geqslant 0}$ is an integer-valued process such that

$$\mathbb{P}\left(w_{b,m}^{(t)} \geqslant k \mid \rho_{b,m}^{(t)} = l\right) = \begin{cases} \left(\frac{\lambda_{\mathrm{AT}}}{\lambda_{\mathrm{AT}}+\lambda_{\mathrm{MN}}}\right)^{k-l} & k \geqslant l \geqslant 0; \\ 1 & l > k \geqslant 0. \end{cases}$$

Sampling $\left(\rho_{b,m}^{(t)}\right)_{t\geqslant 0}$ from its equilibrium,

$$\begin{aligned} l_{b,m}^{(1)} &= \lim_{t\to\infty}\sum_{k=1}^{\infty}\mathbb{P}\left(w_{b,m}^{(t)} \geqslant k\right) \\ &= \frac{\lambda_{\mathrm{AT}} + \lambda_{\mathrm{MN}}}{\lambda_{\mathrm{MN}}} - \frac{\lambda_{\mathrm{MN}}}{\lambda_{\mathrm{AT}} + \lambda_{\mathrm{MN}}} + \frac{\lambda_{\mathrm{AT}}^2}{(\lambda_{\mathrm{AT}} + \lambda_{\mathrm{MN}})\lambda_{\mathrm{MN}}}. \end{aligned}$$

The mean asymptotic queue length at $Q_{b,m}$ is now given by

$$\mathbb{E}(L_{b,m}) = \left(1 + l_{b,m}^{(1)}\right)\lambda_{\mathrm{MN}}\mathbb{E}(W_{b,m;\mathrm{good}}) + l_{b,m}^{(2)}.$$

5. Results and interpretation

In this section we will evaluate the metrics derived in the previous section with actual parameters from existing technologies in order to obtain a performance assessment of our cookie protection scheme for realistic scenarios. Table 1 shows the assumed link speeds and the message lengths according to the relevant communication and signaling protocols, whereas Table 2 lists up the processing times of individual protocol steps obtained by measurements with a prototypical implementation.

5.1. Total response time

Fig. 14 shows the total response time in relation to the attacking rate per cell, when the wireless links are assumed optimistically to offer 54 MBit/s (physical layer according to IEEE 802.11a, leading to $C_{1,2} \approx 50$ μs), $M = 50$ cells are connected to one AAA-server, 10 cells are under attack and $x = 40$ [messages/s] are sent by genuine clients. As can be seen, a DoS situation occurs in case 0 but not

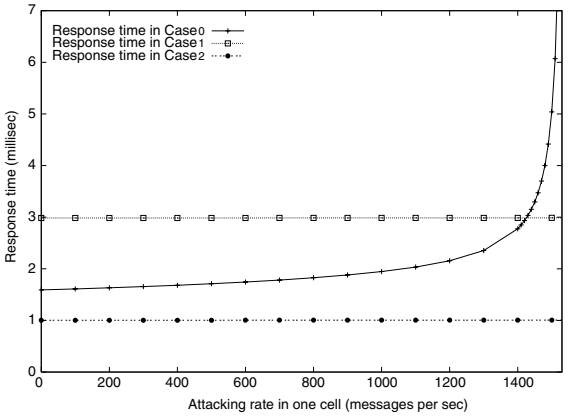


Fig. 14. Total response time for high speed uplink with $C_{1,2} \approx 50$ μs, $M = 50$ cells and $N = 10$ cells under attack for MN rate $x = 40$ [messages/s].

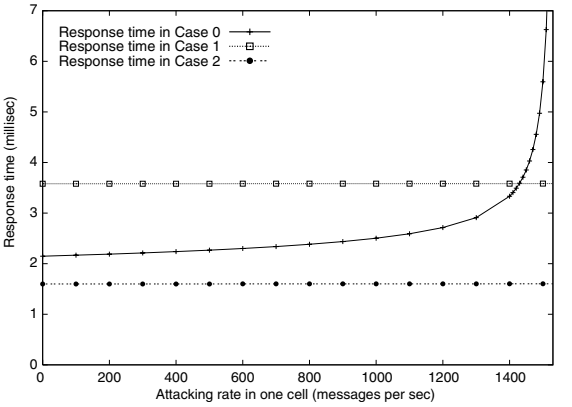


Fig. 15. Total response time for Slow Speed uplink with $C_{1,2} \approx 300$ μs, M = 50 cells and $N = 10$ cells under attack for MN rate $x = 40$ [messages/s].

in the cookie mechanism cases. In case 0 the total response time grows abruptly when the attacking rate approaches 1500 messages/s. The cookie mechanism requires around one additional millisecond of processing but the system is able to serve genuine clients up to any attacking rate, of course provided that there is still bandwidth left in the respective radio cell. Therefore, our DoS protection scheme offers a significant improvement over the unprotected case.

In case of a slower wireless uplink with $C_{1,2} \approx$ 300 μs (corresponding to an IEEE 802.11b WLAN operating at 11 MBit/s with long physical layer preamble) Fig. 15 again shows that a DoS situation would not occur before around 1500 messages/s are sent per cell. However, this attacking rate well exceeds the range in which it can be assumed that the attacker will be able to send his attacking packets over the wireless channel. Therefore, our DoS protection scheme does not offer a benefit in cases with rather low wireless channel capacity, or otherwise stated, more wireless cells have to be supported with one AAA server until our DoS protection scheme offers a significant improvement.

The cost of implementing the cookie mechanism corresponds to the discrepancy between the lines of response time in cases 0 and 1 (see Figs. 14 and 15). Before the attacking rate reaches a saturation point, case 0 performs slightly better than case 1; however, when the attacking rate is running over the saturation point, the cookie mechanism takes obvious effect in preventing a DoS attack, and at any rate, a cost of less than 2 ms is negligible.

5.2. Queue length at AAAL

Fig. 16 shows the queue length at the AAA server for case 0 under the same conditions as in Fig. 14. This graph clearly shows that the DoS situation is caused by the overloading of the AAA server and not by exceeding the transmission capacities of the access network. Under these conditions, the AAA server is not able to keep up

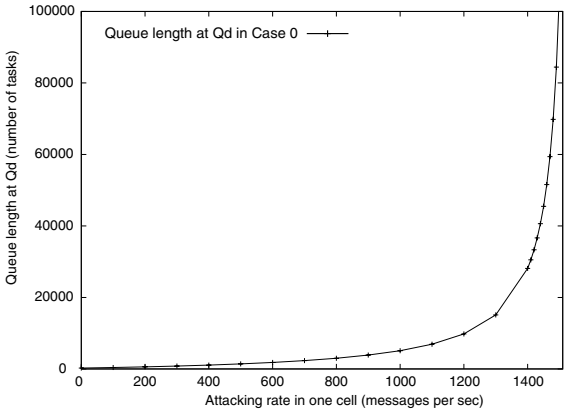


Fig. 16. Queue length at AAAL for high speed uplink $C_{1,2} \approx 50$ μs, $M = 50$ cells and $N = 10$ cells under attack for MN rate $x = 40$ [messages/s].

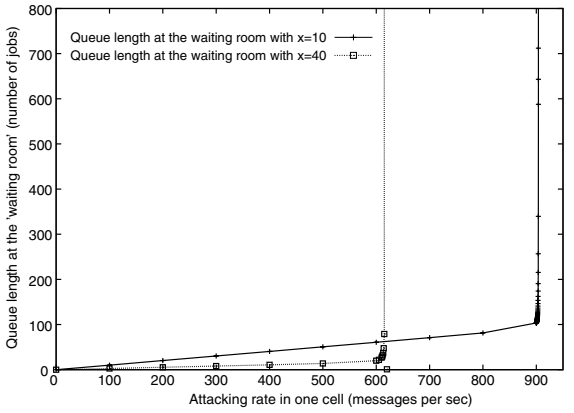


Fig. 17. Queue length in the waiting room for high speed uplink $C_{1,2} \approx 50$ μs, $M = 50$ cells and $N = 25$ cells under attack and different MN rates x.

with checking and discarding bogus messages being sent by attackers, so that the genuine requests from honest clients cannot be processed in time.

5.3. Queue length in the "waiting room"

Fig. 17 shows the queue length in the "waiting room" of an access router in a cell which is under attack (here we assume that 25 out of 50 cells are under attack). Depending on the rate of genuine requests by honest clients ($x = 10$ messages/s vs. $x = 40$ messages/s) a DoS situation will occur earlier. The reason for this behaviour lies in our strategy to silently discard bogus requests directly after they have been identified at the AAA server in order to save AAA processing capabilities. This implies that access routers need to receive a response to a genuine request, in order to be able to discard all bogus messages between two genuine requests. Furthermore, it can be seen from this graph that memory depletion situations can occur at access routers under attack. However, the most critical system in the access network infrastructure is the central AAA server.

6. Conclusions

In this paper, we described a cookie-based mechanism to protect against DoS attacks for optimized and QoS-aware handovers by performing a simple and preliminary check with a cookie before the QoS reservation and authorization processes begin. We also provided a performance evaluation which shows that the cookie mechanism is a method to protect against DoS attacks in the QoS reservation process in a distributed scenario, to speed up registration in the intra-domain handover case by paralleling the QoS reservation process and AA process without introducing additional DoS risks.

Furthermore, our scheme reduces the risk of replayed cookies by implementing an "*area of validity*" in which a cookie is acceptable, and by communicating cookies that have been used once at a particular AR to other ARs in the same area of validity.

The mechanisms of this solution can additionally protect against the following depletion threats (which exist when authentication and resource reservation are performed in parallel or sequentially): depletion of the memory of access routers, that would have to maintain state while the authentication of the MN is fetched from the AAAL, depletion of signaling capacity in the access network (by preventing signaling traffic for bogus requests which have not been verified before as being "credible"), and depletion of the resources of the AAAL (by shielding the AAAL server from authentication requests which result from bogus QoS requests).

Acknowledgements

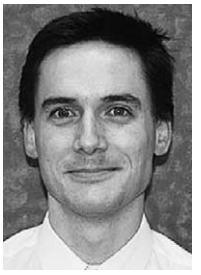
This work has been supported in part by a research contract with Information and Communication Mobile, Siemens AG. The work of S. Adams was supported by DFG Research Center "Mathematics for key technologies" (FZT 86) Berlin. M. Sortais was supported by the Graduiertenkolleg "Stochastische Prozesse und Probabilistische Analysis" Berlin and by the Swiss NSF.

References

[1] Mobility Architecture Implementation Report, D0302 IST-2000-25394 Moby Dick, December 2002.
[2] S. Asmussen, Applied Probability and Queues, second ed., Springer-Verlag, 2003.

[3] T. Aura, P. Nikander, J. Leiwo, DOS-resistant authentication with client puzzles, Lecture Notes in Computer Science 170 (2001) 2133.
[4] P. Calhoun, T. Johansson, C. Perkins, Diameter Mobile IPv4 Application, Internet Draft: draft-ietf-aaa-diameter-mobileip-20.txt, August 2004.
[5] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, Diameter Base Protocol, RFC 3588, September 2003.
[6] H. Chaskar (Ed.), Requirements of a Quality of Service (QoS) Solution for Mobile IP, RFC 3583, September 2003.
[7] T. Chen, M. Sortais, G. Schäfer, A. Wolisz, A performance study of session state re-establishment schemes in ip-based micro-mobility scenarios, in: Proceedings of the 12th Annual Meeting of the IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Volendam, Netherlands, October 2004, in press.
[8] R. Koodli (Ed.), Fast Handovers for Mobile IPv6, Internet Draft: draft-ietf-miopshop-fast-mipv6-03.txt, October 2004.
[9] X. Fu, H. Karl, C. Kappler, QoS-conditionalized Handoff for mobile IPv6, in: Proceedings of the 2nd IFIP-TC6 Networking Conference (Networking 2002), Springer-Verlag, Pisa, Italy, 2002, pp. 721–730.
[10] D. Johnson, C. Perkins, J. Arkko, Mobility Support in IPv6, Internet Draft: draft-ietf-mobileip-ipv6-24.txt, June 2003.
[11] R. Koodli, C. Perkins, Fast handovers and context transfers in mobile networks, ACM Computer Communication Review 31 (No. 5) (2001).
[12] J. Loughney, M. Nakhjiri, C. Perkins, R. Koodli, Context Transfer Protocol, Internet Draft: draft-ietf-seamoby-ctp-05.txt, October 2003.
[13] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press LLC, 1997.
[14] H. Soliman, C. Castelluccia, K. El-Malki, L. Bellier, Hierarchical Mobile IPv6 Mobility Management (HMIPv6), Internet-Draft: draft-ietf-mipshop-hmipv6-00.txt, June 2003.
[15] H. Tschofenig, D. Kroeselberg, Security Threats for NSIS, Internet Draft: draft-ietf-nsis-threats-06.txt, October 2004.

Tianwei Chen studied electronic engineering in the Electronic Engineering Department at Beijing Polytechnic University from 1988 to 1996. During this period, he obtained his bachelor and master degrees. Afterwards he worked at Hewlett Packard. In 2000 he started his Ph.D. study in Telecommunication Networks Group (TKN) at the Technical University in Berlin. His research interests include network security, Quality of Service (QoS) and mobile communications.



Michel Sortais studied Mathematics at the Swiss Federal Institute of Technology in Lausanne. Specializing in Probability Theory and Statistical Mechanics, he obtained the PhD degree from this institution in October 2001. Since then, he has been a teaching and research assistant at the Technical University in Berlin, joining the DFG Research Center "Mathematics for Key Technologies" in October 2002. His current interests include engineering applications of Probability such as Queuing Theory or Mobility Models.

Günter Schäfer received his diploma and Ph.D in computer science from the University of Karlsruhe in 1994 and 1998, respectively. Between February 1999 and July 2000 he took a post at the Ecole Nationale Supérieure des Télécommunications in Paris, France, where he focused on network security and access network performance of third-generation mobile communication networks. Since August 2000, he is working at the Technical University of Berlin where he is involved in research and lectures on the subject of telecommunications networks. His main subject areas are network security, mobile communication and active network technologies. He is a member of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE) and the German Gesellschaft für Informatik (Computer Science Society).

Stefan Adams is a research assistant at the Max-Planck-Institute for Mathematics in the Sciences, Leipzig, Germany. He received the Dipl.-Phys. degree from the University of Göttingen, Germany, in 1995, and the Dipl.-Math. degree from the University of Hagen, Germany, in 1998, both in theoretical physics and mathematics. He received his Ph.D. in mathematics from the University of Munich in 2000. He was a member of the new German Science Foundation research center for *mathematics in key technologies*, FZT86, Berlin. Since October 2004 he is a researcher of the Max-Planck-Institut for Mathematics in Science in Leipzig.

Changpeng Fan has been with Siemens AG since 2000, where he is currently a Senior Technology Manager at the Research and Concepts Department of Siemens Communications. Prior to this, he was a Senior Researcher at the GMD Research Institute for Open Communication Systems and a Senior Research Staff Member at the C&C Research Laboratories of NEC Europe. He has authored and co-authored several dozens of refereed papers in technical journals and conference proceedings, mainly on multimedia communications and mobile networks. He received his B.S. and M.S. degrees from Nanjing University, his Ph.D. degree from the Technical University of Berlin, all in Computer Science.

Adam Wolisz is currently a Professor of Electrical Engineering and Computer Science (secondary assignment) at the Technical University Berlin, where he is directing the Telecommunication Networks Group (TKN). He is also member of the Senior Board of GMD Fokus, being especially in charge of the Competence Centers GLONE and TIP. His teaching activities encompass courses on Communication Networks and Protocols, High Speed Networks, Wireless Networks and Performance Analysis of Communication Networks. He is acting as a member of the Steering Committee of the Computer Engineering Curriculum at the Technical University Berlin. He participates in the nationally (Deutsche Forschungsgemeinschaft) founded Graduate Course in Communication-Based Systems. His research interests are in architectures and protocols of communication networks as well as protocol engineering with impact on performance and QoS aspects. Recently he has been working mainly on mobile multimedia communication, with special regard to architectural aspects of network heterogeneity and integration of wireless networks in the Internet. The research topics are usually investigated by a combination of simulation studies and real experiments. He has authored two books and authored or co-authored over 100 papers in technical journals and conference proceedings. He is Senior Member of IEEE, IEEE Communications Society (including the TCCC and TCPC) as well as the GI/ITG Technical Committee on Communication and Distributed Systems.