Groupe de travail Réseau
Request for Comments: 5531
RFC rendue obsolète : 1831
Catégorie : Sur la voie de la normalisation

R. Thurlow, Sun Microsystems
mai 2009

Traduction Claude Brière de L'Isle

Spécification du protocole d'appel de procédure à distance version 2 (RPCv2)

Statut de ce mémoire

Le présent document spécifie un protocole sur la voie de la normalisation de l'Internet pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Le présent document décrit la version 2 du protocole d'appel de procédure à distance (RPC, *Remote Procedure Call*) de calcul de réseau ouvert (ONC, *Open Network Computing*) version 2 comme il est actuellement déployé et accepté. Ce document rend obsolète la RFC 1831.

Table des matières

1. Introduction

Le présent document spécifie la version 2 du protocole de message utilisé dans l'appel de procédure distante (RPC) d'ONC. Le protocole de message est spécifié avec le langage de représentation de données externes (XDR, *eXternal Data Representation*) [RFC4506]. Le présent document suppose que le lecteur est familiarisé avec XDR. Il ne tente pas de justifier les systèmes d'appel de procédure à distance ni de décrire leur usage. L'article de Birrell et Nelson [XRPC] est recommandé comme une excellente introduction au concept d'appel de procédure distante.

1.1 Langage des exigences

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119],.

2. Changements par rapport à la RFC 1831

Le présent document rend obsolète la [RFC1831] comme document d'autorité pour décrire RPC, sans introduire de changement au protocole sur le réseau. Les principaux changements par rapport à la RFC 1831 sont :

- Ajout d'un Appendice qui décrit comment une mise en œuvre peut demander de nouveaux numéros de programme RPC, numéros de nuance d'authentification, et numéros d'état d'authentification à l'IANA, plutôt qu'à Sun Microsystems

- Ajout d'une section de "Considérations relatives à l'IANA" qui décrit la politique passée d'allocation des numéros et comment l'IANA est destinée à les allouer à l'avenir.

- Précisions à la spécification de langage RPC pour correspondre à l'usage courant.

- Amélioration de la section des "Considérations sur la sécurité" pour refléter l'expérience de nuances de sécurité fortes.

- Spécification de nouvelles erreurs d'authentification qui sont d'usage courant dans les mises en œuvre modernes de RPC.

- Mise à jour avec les dernières déclarations de propriété intellectuelle de l'IETF.

3. Terminologie

Le présent document discute des clients, appels, serveurs, réponses, services, programmes, procédures, et versions. Chaque appel de procédure distant a deux côtés : un côté client actif qui fait l'appel à un côté serveur qui renvoie une réponse. Un service réseau est une collection d'un ou plusieurs programmes distants. Un programme distant met en œuvre une ou plusieurs procédures distantes ; les procédures, leurs paramètres, et les résultats sont documentés dans les spécifications de protocole spécifiques du programme. Un serveur peut prendre en charge plus d'une version d'un programme distant afin d'être compatible avec des changements de protocoles.

Par exemple, un service de fichiers réseau peut être composé de deux programmes. Un programme peut traiter des applications de haut niveau comme un système de contrôle d'accès et de verrouillage de fichiers. L'autre peut traiter des entrées-sorties de fichier de niveau inférieur et avoir des procédures comme "lire" et "écrire". Un client du service de fichiers réseau va appeler les procédures associées aux deux programmes du service au nom du client.

Les termes "client" et "serveur" s'appliquent seulement à une transaction particulière ; une entité matérielle particulière

(hôte) ou une entité logicielle (processus ou programme) pourrait opérer dans les deux rôles à des moments différents. Par exemple, un programme qui fournit un service d'exécution à distance pourrait aussi être un client d'un service de fichiers réseau.

4. Modèle RPC

Le protocole RPC ONC est fondé sur le modèle de procédure d'appel à distance, qui est similaire au modèle d'appel de procédure locale. Dans le cas local, l'appelant place des arguments à une procédure dans une localisation bien spécifiée (comme une fenêtre d'un registre). Il transfère alors le contrôle à la procédure, et finalement reprend le contrôle. À ce point, le résultat de la procédure est extrait de la localisation bien spécifiée, et l'appelant continue l'exécution.

Le modèle de procédure d'appel à distance est similaire. Un fil de contrôle passe logiquement à travers deux processus : le processus de l'appelant et un processus de serveur. L'appelant envoie d'abord un message d'appel au processus de serveur et attend (se bloque) pour un message de réponse. Le message d'appel inclut les paramètres de procédure, et le message de réponse inclut les résultats de la procédure. Une fois que le message de réponse est reçu, les résultats de la procédure sont extraits, et l'exécution de l'appelant est reprise.

Du côté du serveur, un processus est dormant en attendant l'arrivée d'un message d'appel. Quand il en arrive un, le processus de serveur extrait les paramètres de procédure, calcule les résultats, envoie un message de réponse, et ensuite attend le prochain message d'appel.

Dans ce modèle, seulement un des deux processus est actif à un moment donné. Cependant, ce modèle est seulement donné comme exemple. Le protocole RPC ONC ne fait pas de restriction sur le modèle mis en œuvre, et d'autres sont possibles. Par exemple, une mise en œuvre peut choisir d'avoir des appels de RPC asynchrones de sorte que le client peut faire des travaux utiles tout en attendant la réponse du serveur. Une autre possibilité est de faire que le serveur crée une tâche séparée pour traiter un appel entrant afin que le serveur d'origine puisse être libre de recevoir d'autres demandes.

Il y a quelques différences importantes entre les appels de procédure distants et ceux de procédure locale.

- Traitement d'erreur : les défaillances du serveur ou réseau distant doivent être traitées quand on utilise des appels de procédure distants.

- Variables globales et effets collatéraux : comme le serveur n'a pas accès à l'espace d'adresses du client, des arguments cachés ne peuvent pas être passés comme des variables globales ou retournés comme effets collatéraux.

- Performances : les procédures distantes opèrent généralement à un ou plusieurs ordres de grandeurs plus lentement que les appels de procédure locaux.

- Authentification : comme les appels de procédure distants peuvent être transportés sur des réseaux non sécurisés, l'authentification peut être nécessaire. L'authentification empêche une entité de se faire passer pour une autre entité.

La conclusion est que même si il y a des outils pour générer automatiquement des bibliothèques de clients et serveurs pour un service donné, les protocoles doivent quand même être conçus avec soin.

5. Transports et sémantique

Le protocole RPC peut être mis en œuvre sur plusieurs protocoles de transport différents. La portée de la définition du protocole RPC exclut comment un message est passé d'un processus à un autre, et inclut seulement la spécification et l'interprétation des messages. Cependant, l'application peut souhaiter obtenir des informations sur (et peut-être le contrôle sur) la couche de transport à travers une interface non spécifiée dans ce document. Par exemple, le protocole de transport peut imposer une restriction à la taille maximum des messages RPC, ou il peut être en mode flux comme TCP [RFC0793] sans limite de taille. Le client et le serveur doit s'accorder sur le choix de leur protocole de transport.

Il est important de souligner que RPC n'essaye pas de mettre en œuvre une forme de fiabilité et que l'application peut avoir besoin de connaître le type de protocole de transport en dessous de RPC. Si il est connu qu'il fonctionne par dessus un transport fiable comme TCP, alors la plus grande partie du travail est déjà faite pour cela. Par ailleurs, si il fonctionne par dessus un transport non fiable comme UDP [RFC0768], il doit mettre en œuvre ses propres politiques de temporisation,

retransmission, et détection de dupliqués car le protocole RPC ne fournit pas ces services.

À cause de l'indépendance au transport, le protocole RPC n'attache pas une signification spécifique aux procédures distantes ou leurs exigences d'exécution. La sémantique peut être déduite (mais devrait être explicitement spécifiéer) du protocole de transport sous-jacent. Par exemple, si on considère RPC fonctionnant sur un transport non fiable comme UDP. Si une application retransmet des messages d'appel RPC après des fins de temporisation, et ne reçoit pas de réponse, elle ne peut rien déduire sur le nombre de fois que la procédure a été exécutée. Si elle ne reçoit pas de réponse, elle peut alors déduire que la procédure a été exécutée au moins une fois.

Un serveur peut souhaiter se souvenir des demandes acceptées précédemment d'un client et ne pas les accorder à nouveau, afin d'assurer un certain degré de sémantique d'exécution au plus une fois. Un serveur peut faire cela en tirant parti de l'identifiant de transaction qui est associé à chaque message RPC. Le principal usage de cet identifiant de transaction est la confrontation des réponses aux appels par l'entité de client RPC. Cependant, une application de client peut choisir de réutiliser son identifiant de transaction précédent quand elle retransmet un appel. Le serveur peut choisir de se souvenir de cet identifiant après l'exécution d'un appel et ne pas exécuter les appels avec le même identifiant, afin de réaliser un certain degré de sémantique d'exécution au plus une fois. Il n'est pas permis au serveur d'examiner cet identifiant d'une autre façon que d'un essai d'égalité.

Par ailleurs, si en utilisant un transport "fiable" comme TCP, l'application peut déduire d'un message de réponse que la procédure a été exécutée exactement une fois, mais si elle ne reçoit pas de message de réponse, elle ne peut pas supposer que la procédure distante n'a pas été exécutée. Noter que même si un protocole en mode connexion comme TCP est utilisé, une application a quand même besoin de temporisations et de reconnexions pour traiter les pannes de serveur.

Il y a d'autres possibilités pour les transports au delà des protocoles en mode datagramme ou connexion. Par exemple, un protocole de demande-réponse comme [VMTP] est peut-être un transport naturel pour RPC. RPC ONC utilise actuellement à la fois les protocoles de transport TCP et UDP. La Section 11 ("Norme de marquage d'enregistrement") décrit le mécanisme employé par RPC ONC pour utiliser un transport en mode connexion fondé sur le flux comme TCP. Le mécanisme par lequel de futurs transports ayant des caractéristiques structurelles différentes devraient être utilisés pour transférer les messages ONC RPC devrait être spécifié par une RFC sur la voie de la normalisation, quand de tels transports supplémentaires seront définis.

6. Indépendance de liaison et de rendez vous

L'acte de lier un client particulier à des paramètres de service et transport particuliers NE FAIT PAS partie de cette spécification de protocole RPC. Cette fonction importante et nécessaire est laissée à des logiciels de niveau supérieur.

Les utilisateurs pourraient voir le protocole RPC comme l'instruction "sauter au sous-programme" (JSR, *jump-subroutine*) d'un réseau ; le chargeur (lieur) rend JSR utile, et le chargeur lui-même utilise JSR pour accomplir cette tâche. De même, le logiciel de lien rend RPC utile, éventuellement en utilisant RPC pour accomplir cette tâche.

7. Authentification

Le protocole RPC fournit les champs nécessaires pour qu'un client s'identifie à un service, et vice-versa, dans chaque appel et message de réponse. Les mécanismes de sécurité et de contrôle d'accès peuvent être construits par dessus cette authentification de message. Plusieurs différents protocoles d'authentification peuvent être pris en charge. Un champ dans l'en-tête RPC indique quel protocole est utilisé. Plus d'informations sur des protocoles d'authentification spécifiques sont au paragraphe 8.2, "Authentification, intégrité et confidentialité".

8. Exigences du protocole RPC

Le protocole RPC doit fournir ce qui suit :
- o la spécification unique d'une procédure à invoquer ;
- o des dispositions pour confronter les messages de réponse aux messages de demande ;
- o des dispositions pour authentifier l'appelant au service et vice-versa.

À côté de ces exigences, des caractéristiques qui détectent ce qui suit valent la peine d'être prises en charge à cause des erreurs de changement de protocole, des erreurs de la mise en œuvre, de l'utilisateur, et de l'administration du réseau :
- o les discordances de protocole RPC ;
- o les discordances de version de protocole de programme distant ;
- o les erreurs de protocole (comme une mauvaise spécification de paramètres d'une procédure) ;
- o les raisons de l'échec de l'authentification distante ;
- o toutes autres raisons pour lesquelles la procédure désirée n'a pas été invoquée.

8.1 Programmes et procédures de RPC

Le message d'appel RPC a trois champs d'entier non signé- -- le numéro de programme distant, le numéro de version de programme distant, et le numéro de procédure distante -- qui identifient de manière univoque la procédure à invoquer. Les numéros de programme sont administrés par une autorité centrale (IANA). Une fois que les utilisateurs ont un numéro de programme, ils peuvent mettre en œuvre leur programme distant ; la première mise en œuvre va très probablement avoir le numéro de version 1 mais elle NE DOIT PAS être le numéro zéro. Parce que la plupart des nouveaux protocoles évoluent, un champ "version" du message d'appel identifie quelle version du protocole l'appelant utilise. Les numéros de version permettent la prise en charge des protocoles anciens et nouveaux à travers les mêmes processus de serveur.

Le numéro de procédure identifie la procédure à invoquer. Ces numéros sont documentés dans la spécification de protocole du programme spécifique. Par exemple, la spécification de protocole d'un service de fichiers peut déclarer que son numéro de procédure 5 est "lire" et son numéro de procédure 12 est "écrire".

Tout comme les protocoles de programme distant peuvent changer sur plusieurs versions, le protocole réel de message RPC pourrait aussi changer. Donc, le message d'appel a aussi en lui le numéro de version RPC, qui est toujours égal à 2 pour la version de RPC décrite ici.

Le message de réponse à un message de demande a assez d'informations pour distinguer les conditions d'erreur suivantes :
- o La mise en œuvre distante de RPC ne prend pas en charge la version 2 du protocole. Le plus haut et le plus bas numéros de version RPC pris en charge sont retournés.
- o Le programme distant n'est pas disponible sur le système distant.
- o Le programme distant ne prend pas en charge le numéro de version demandé. Le plus haut et le plus bas numéros de version de programme sont retournés.
- o Le numéro de procédure demandé n'existe pas. (Ceci est généralement une erreur de protocole ou de programmation côté client.)
- o Les paramètres de la procédure distante apparaissent comme des déchets du point de vue du serveur. (Là encore, ceci est généralement causé par un désaccord sur le protocole entre client et service.)

8.2 Authentification, intégrité, et confidentialité

Les dispositions pour l'authentification de l'appelant auprès du service et vice-versa sont fournies au titre du protocole RPC. Le message d'appel a deux champs d'authentification : l'accréditif et le vérificateur. Le message de réponse a un champ d'authentification : le vérificateur de réponse. La spécification du protocole RPC définit ces trois champs comme étant du type opaque suivant (dans le langage de représentation de données externes (XDR, *eXternal Data Representation*) [RFC4506]) :

```
enum auth_nuance {
  AUTH_NONE   = 0,
  AUTH_SYS    = 1,
  AUTH_SHORT = 2,
  AUTH_DH     = 3,
  RPCSEC_GSS  = 6,
  /* et d'autres à définir */
};

struct opaque_auth {
  auth_nuance nuance;
  opaque body<400>;
};
```

En d'autres termes, toute structure "opaque_auth" est une énumération de "auth_nuance" suivie par jusqu'à 400 octets qui

sont opaques (non interprétés par) à la mise en œuvre de protocole RPC.

L'interprétation et la sémantique des données contenues dans les champs d'authentification sont spécifiées par les spécifications individuelles de protocole d'authentification indépendantes.

Si des paramètres d'authentification ont été rejetés, le message de réponse contient des informations déclarant pourquoi ils ont été rejetés.

Comme montré dans RPCSEC_GSS, il est possible à une "auth_nuance" de prendre aussi en charge l'intégrité et la confidentialité.

8.3 Allocation de numéro de programme

Les numéros de programmes sont attribués dans des groupes en accord avec le tableau suivant :

0x00000000 : réservé
0x00000001 - 0x1fffffff : à allouer par l'IANA
0x20000000 - 0x3fffffff : défini par l'administrateur local (certains blocs sont alloués ici)
0x40000000 - 0x5fffffff : transitoire
0x60000000 - 0x7effffff : réservé
0x7f000000 - 0x7fffffff : allocation en cours
0x80000000 - 0xffffffff : réservé

Le premier groupe est une gamme de numéros administrés par l'IANA et devrait être identique pour tous les sites. La seconde gamme est pour les applications particulières à un site particulier. Cette gamme est destinée principalement au débogage de nouveaux programmes. Quand un site développe une application qui pourrait être d'intérêt général, cette application devrait recevoir un numéro alloué dans la première gamme. Les développeurs d'applications peuvent demander des blocs de numéros de programme RPC dans la première gamme par les méthodes décrites à l'Appendice B. Le troisième groupe est pour les applications qui génèrent dynamiquement des numéros de programme. Les groupes de la fin sont réservés à de futures utilisations, et ne devraient pas être utilisés.

8.4 Autres usages du protocole RPC

L'intention de ce protocole est pour les procédures d'appel distant. Normalement, chaque message d'appel est confronté à un message de réponse. Cependant, le protocole lui-même est un protocole de passation de message avec lequel d'autres protocoles (non d'appel de procédure) peuvent être mis en œuvre.

8.4.1 Mise en lots

La mise en lots est utile quand un client souhaite envoyer une séquence arbitrairement grande de messages d'appel à un serveur. La mise en lots utilise normalement des protocoles de flux d'octets fiables (comme TCP) pour son transport. Dans le cas de mise en lots, le client n'attend jamais une réponse du serveur, et le serveur n'envoie pas de réponse aux appels en lots. Une séquence d'appels en lots est généralement terminée par une opération légitime d'appel de procédure distante afin de purger le canal et obtenir un accusé de réception positif.

8.4.2 Diffusion d'appels de procédure distants

Dans les protocoles de diffusion, le client envoie un appel en diffusion au réseau et attend de nombreuses réponses. Cela exige l'utilisation de protocoles de transport fondés sur le paquet (comme UDP). Les serveurs qui prennent en charge les protocoles de diffusion répondent génralement seulement quand l'appel est traité avec succès et sont silencieux en face d'erreurs, mais cela varie avec l'application.

Les principes de RPC en diffusion s'appliquent aussi à la diffusion groupée -- une demande RPC peut être envoyée à une adresse de diffusion groupée.

9. Protocole de message RPC

Cette section définit le protocole de message RPC dans le langage de description de données XDR [RFC4506].

```
enum msg_type {
   CALL  = 0,
   REPLY = 1
};
```

Une réponse à un message d'appel peut prendre deux formes : le message est accepté ou rejeté.

```
enum réponse_stat {
  MSG_ACCEPTED = 0,
  MSG_DENIED      = 1
};
```

Quand un message d'appel est accepté, ce qui suit est l'état d'une tentative d'appel de procédure distante.

```
enum accept_stat {
  SUCCESS  = 0,                 /* RPC exécuté avec succès */
  PROG_UNAVAIL = 1,             /* le côté distant n'a pas exporté le programme */
  PROG_MISMATCH = 2,            /* le côté distant ne peut pas prendre en charge le n° de version */
  PROC_UNAVAIL = 3,             /* le programme ne peut pas prendre en charge la procédure */
  GARBAGE_ARGS = 4,             /* la procédure ne peut pas décoder les paramètres */
  SYSTEM_ERR = 5                /* par exemple échec d'allocation de mémoire */
};
```

Raisons du rejet d'un message d'appel :

```
enum reject_stat {
  RPC_MISMATCH = 0,             /* RPC numéro de version != 2         */
  AUTH_ERROR = 1                /* remote ne peut pas authenticate appelant */
};
```

Pourquoi l'authentification a échoué :

```
enum auth_stat {
AUTH_OK = 0,                            /* succès */
/* Échec à l'extrémité distante */
AUTH_BADCRED = 1,                       /* mauvais accréditif (sceau brisé) */
AUTH_REJECTEDCRED = 2,                  /* le client doit commencer une nouvelle session */
AUTH_BADVERF = 3,                       /* mauvais vérificateur (sceau brisé) */
AUTH_REJECTEDVERF = 4,                  /* vérificateur expiré ou répété */
AUTH_TOOWEAK = 5,                       /* rejeté pour des raisons de sécurité */
/* Échec local */
AUTH_INVALIDRESP = 6,                   /* vérificateur de réponse bogué */
AUTH_FAILED = 7,                        /* raison inconnue */
/* Erreurs AUTH_KERB ; déconseillé. Voir la [RFC2695] */
AUTH_KERB_GENERIC = 8,                  /* erreur kerberos générique */
AUTH_TIMEEXPIRE = 9,                    /* temps d'accréditif expiré */
AUTH_TKT_FILE = 10,                     /* problème de ticket de fichier */
AUTH_DECODE = 11,                       * ne peut pas décoder l'authentificateur */
AUTH_NET_ADDR = 12,                     * adresse réseau erronnée dans le ticket */
/* Erreurs relatives à GSS dans RPCSEC_GSS GSS */
RPCSEC_GSS_CREDPROBLEM = 13,   /* pas d'accréditifs pour l'utilisateur */
RPCSEC_GSS_CTXPROBLEM = 14      /* problèmes avec le contexte */
};
```

Lorsque de nouveaux mécanismes d'authentification seront ajoutés, il pourrait y avoir besoin de plus de codes d'état pour les prendre en charge. L'IANA traitera les nouveaux numéros de auth_stat sur la simple base du premier arrivé, premier

servi, comme défini dans les "Considérations relatives à l'IANA" et à l'Appendice B.

Le message RPC :
Tous les messages commencent par un identifiant de transaction, xid, suivi par une union discriminante à deux branches. Le discriminant de l'union est un msg_type qui commute sur un des deux types de message. Le xid d'un message REPLY correspond toujours à celui du message CALL initiateur. Note : le champ "xid" est seulement utilisé pour les clients qui confrontent les messages de réponse avec des messages d'appel ou pour les serveurs qui détectent des retransmissions ; le côté service ne peut pas traiter cet identifiant comme un type de numéro de séquence.

```
struct rpc_msg {
  unsigned int xid;
  union switch (msg_type mtype) {
  cas CALL :
    call_body cbody;
  cas REPLY :
    reply_body rbody;
  } body;
};
```

Corps d'un appel RPC :
Dans la version 2 de la spécification du protocole RPC, rpcvers DOIT être égal à 2. Les champs "prog", "vers", et "proc" spécifient le programme distant, son numéro de version, et la procédure au sein du programme distant à invoquer. Après ces champs sont deux paramètres d'authentification : cred (accréditif d'authentification) et verf (vérificateur d'authentification). Les deux paramètres d'authentification sont suivis des paramètres de la procédure distante, qui sont spécifiés par le protocole de programme spécifique.

L'objet du vérificateur d'authentification est de valider l'accréditif d'authentification. Noter que ces deux éléments sont historiquement séparés, mais sont toujours utilisés ensemble comme une entité logique.

```
struct call_body {
  unsigned int rpcvers;                /* doit être égal à deux (2) */
  unsigned int prog;
  unsigned int vers;
  unsigned int proc;
  opaque_auth cred;
  opaque_auth verf;
  /* les paramètres spécifiques dc la procédure commencent ici */
};
```

Corps d'une réponse à un appel RPC :

```
union reply_body switch (reply_stat stat) {
cas  MSG_ACCEPTED:
  accepted_reply areply;
cas  MSG_DENIED:
  rejected_reply rreply;
} reply;
```

Réponse à un appel RPC accepté par le serveur :
Il pourrait y avoir une erreur même si l'appel a été accepté. Le premier champ est un vérificateur d'authentification que le serveur génère afin de se valider lui-même auprès du client. Il est suivi par une union dont le discriminant est une énumération accept_stat. La branche SUCCESS de l'union est spécifique du protocole. Les branches PROG_UNAVAIL, PROC_UNAVAIL, GARBAGE_ARGS, et SYSTEM_ERR de l'union sont vides. La branche PROG_MISMATCH spécifie le plus bas er le plus haut numéro de version du programme distant pris en charge par le serveur.

```
struct accepted_reply {
  opaque_auth verf;
  union switch (accept_stat stat) {
  cas SUCCESS:
   opaque results[0];
/* Les résultats spécifiques de la procédure commencent ici */
```

```
  cas PROG_MISMATCH:
    struct {
      unsigned int low;
      unsigned int high;
      } mismatch_info;
    default:
/* Vide. Les cas incluent PROG_UNAVAIL, PROC_UNAVAIL, GARBAGE_ARGS, et SYSTEM_ERR. */
   void;
   } reply_data;
};
```

Réponse à un appel RPC rejeté par le serveur :
L'appel peut être rejeté pour deux raisons : soit le serveur fonctionne sur une version non compatible du protocole RPC (RPC_MISMATCH) soit le serveur rejette l'identité de l'appelant (AUTH_ERROR). Dans le cas d'une version RPC discordante, le serveur retourne le plus bas et le plus haut numéro de version RPC pris en charge. En cas d'authentification invalide , l'état d'échec est retourné.

```
union rejected_reply switch (reject_stat stat) {
cas RPC_MISMATCH :
  struct {
    unsigned int low;
    unsigned int high;
  } mismatch_info;
cas AUTH_ERROR :
    auth_stat stat;
};
```

10. Protocoles d'authentification

Comme déclaré précédemment, les paramètres d'authentification sont opaques, mais ouverts au reste du protocole RPC. Cette section définit deux nuances standard d'authentification. Les mises en œuvre sont libres d'inventer de nouveaux types d'authentification, avec les mêmes règles d'allocation de numéro de nuance que pour les numéros de programme. La nuance d'un accréditif ou vérificateur se réfère à la valeur du champ "flavor" *(nuance)* dans la structure opaque_auth. Les numéros de nuance, comme les numéros de programme RPC, sont aussi administrés centralement, et les développeurs peuvent allouer de nouveaux numéros de nuance par les méthodes décrites dans l'Appendice B. Les accréditifs et les vérificateurs sont représentés comme des données opaques de longueur variable (le champ "body" dans la structure opaque_auth).

Dans ce document, deux nuances d'authentification sont décrites. Parmi elles, l'authentification Nulle (décrite au paragraphe suivant) est obligatoire -- elle DOIT être disponible dans toutes les mises en œuvre. L'authentification de système (AUTH_SYS) est décrite dans l'Appendice A. Les mises en œuvre PEUVENT inclure AUTH_SYS dans leur développement pour prendre en charge les applications existantes. Voir les "Considérations sur la sécurité" pour des informations sur les autres nuances d'authentification, plus sûres.

10.1 Authentification nulle

Souvent, les appels doivent être faits alors que client ne se soucie pas de son identité ou que le serveur ne se soucie pas de qui est le client. Dans ce cas, la nuance de l'accréditif, vérificateur, et vérificateur de réponse du message RPC est "AUTH_NONE". Les données opaques associées au "AUTH_NONE" sont indéfinies. Il est recommandé que la longueur des données opaque soit zéro.

11. Norme de marquage d'enregistrement

Quand les messages RPC sont passés par dessus un protocole de transport de flux d'octets (comme TCP) il est nécessaire de délimiter les messages les uns par rapport aux autres afin de détecter et éventuellement récupérer d'erreurs de protocole. C'est ce qui est appelé le marquage d'enregistrements (RM, *Record Marking*). Un message RPC tient dans un RM.

Un enregistrement est composé d'un ou plusieurs fragments d'enregistrement. Un fragment d'enregistrement est un en-tête

de quatre octets suivis de 0 à (2**31) - 1 octets de données de fragment. Les octets codent un nombre binaire non signé ; comme avec les entiers XDR, l'ordre des octets est du plus fort poids au moindre. Le nombre code deux valeurs -- un booléen qui indique si le fragment est le dernier fragment de l'enregistrement (le bit de valeur 1 implique que le fragment est le dernier fragment) et une valeur binaire non signée de 31 bits qui est la longueur en octets des données du fragment. La valeur booléenne est le bit de poids fort de l'en-tête ; la longueur est les 31 bits de moindre poids. (Noter que cette spécification d'enregistrement N'est Pas dans le format de la norme XDR !)

12. Langage RPC

Tout comme il était nécessaire de décrire les types de données XDR dans un langage formel, il est aussi nécessaire de décrire les procédures qui opèrent sur ces types de données XDR dans un langage formel. Le langage RPC est une extension du langage XDR, avec l'ajout des déclarations "program", "procedure", et "version". Les mots-clés "program" et "version" sont réservés dans le langage RPC, et les mises en œuvre de compilateurs XDR PEUVENT réserver ces mots-clés même quand ils sont fournis avec de pures descriptions XDR, non RPC. L'exemple suivant est utilisé pour décrire l'essence du langage.

12.1 Exemple de service décrit en langage RPC

Voici un exemple de la spécification d'un simple programme de ping.

```
program PING_PROG {
                                        /* Dernière et plus haute version */
  version PING_VERS_PINGBACK {
     void
      PINGPROC_NULL(void) = 0;
/* Ping le client, retourne le temps d'aller retour (en microsecondes). Retourne -1 si l'opération est expirée. */
     int
     PINGPROC_PINGBACK(void) = 1;
} = 2;

/* Version originale /
    version PING_VERS_ORIG {
       void
       PINGPROC_NULL(void) = 0;
       } = 1;
} = 1;

const PING_VERS = 2;                    /* dernière version */
```

La première version décrite est PING_VERS_PINGBACK avec deux procédures : PINGPROC_NULL et PINGPROC_PINGBACK. PINGPROC_NULL ne prend pas d'arguments et ne retourne pas de résultats, mais elle est utile pour calculer les délais d'aller-retour du client au serveuret vice-versa. Par convention, la procédure 0 de tout protocole RPC devrait avoir la même sémantique et ne jamais exiger d'authentification. La seconde procédure est utilisée pour que le client obtienne du serveur qu'il fasse une opération de ping inverse vers le client, et qu'il retourne le temps (en microsecondes) que l'opération a utilisé. La version suivante, PING_VERS_ORIG, est la version originale du protocole, et elle ne contient pas de procédure PINGPROC_PINGBACK. Elle est utile pour la compatibilité avec les vieux programmes de client, et lorsque ce programme sera mûr, elle pourra être éliminée entièrement du protocole.

12.2 Spécification du langage RPC

Le langage RPC est identique au language XDR défini dans la RFC 4506, excepté l'ajout de la définition d'une "program-def", décrite ci(-dessous.

```
program-def:
  "program" identifier "{"
    version-def
    version-def *
  "}" "=" constant ";"
```

```
version-def:
  "version" identifier "{"
     procedure-def
     procedure-def *
  "}" "=" constant ";"

procedure-def:
  proc-return identifier "(" proc-firstarg
    "," type-specifier )* ")" "=" constant ";"

proc-return: "void" | type-specifier

proc-firstarg: "void" | type-specifier
```

12.3 Notes de syntaxe

- Les mots-clés suivants sont ajoutés et ne peuvent pas être utilisés comme identifiants : "program" et "version".
- Un nom de version ne peut pas se produire plus d'une fois dans la portée d'une définition de programme. Un numéro de version ne peut pas non plus se produire plus d'une fois dans la portée d'une définition de programme.
- Un nom de procédure ne peut pas se produire plus d'une fois dans la portée d'une définition de version. Un numéro de procédure ne peut pas non plus se produire plus d'une fois dans la portée d'une définition de version.
- Les identifiants de programme sont dans le même espace de nom que les constantes et identifiants de type.
- Seules des constantes non signées peuvent être allouées aux programmes, versions, et procédures.
- Les compilateurs de langage RPC actuels ne prennent généralement pas en charge plus d'un spécificateur de type dans les listes d'arguments de procédure ; la pratique usuelle est d'envelopper les arguments dans une structure.

13. Considérations relatives à l'IANA

L'allocation de numéros de programme RPC, de numéros de nuance d'authentification, et de numéros d'état d'authentification a dans le passé été effectuée par Sun Microsystems, Inc (Sun). Ceci est inapproprié pour un protocole sur la voie de la normalisation de l'IETF, car ce travail est bien fait par l'autorité d'allocation des numéros de l'Internet (IANA). Le présent document propose le transfert de l'autorité sur les numéros de programme RPC, les numéros de nuance d'authentification, et les numéros d'état d'authentification décrits ici de Sun Microsystems, Inc. à l'IANA et décrit comment l'IANA va tenir et allouer ces numéros. Les utilisateurs des protocoles RPC bénéficieront d'avoir un corps indépendant responsable de ces allocations de numéros.

13.1 Demandes de numéros à l'IANA

L'appendice B de ce document décrit les informations à envoyer à l'IANA pour demander un ou plusieurs numéros RPC et les règles qui s'appliquent. L'IANA va mémoriser la demande à des fins documentaires et mettra les informations suivantes dans le registre public :

- La description courte et l'utilisation de l'objet
- Le ou les numéros de programme à allouer
- La ou les chaînes d'identifiants courts.

13.2 Protection des allocations passées

Sun a fait des allocations dans les deux espaces de numéro de programme RPC et d'espace de numéros de nuance d'authentification RPC depuis le déploiement original de RPC. Les allocations faites par Sun Microsystems sont toujours valides, et seront préservées. Sun a communiqué à l'IANA toutes les allocations courantes dans les deux espaces de numéros et le traitement final des allocations de numéros est achevé. Les allocations courantes de numéro de programme et d'authentification sont fournies à l'Appendice C. La liste des numéros d'état d'authentification courants est donnée à la

Section 9 de ce document dans la définition de "enum auth_stat".

13.3 Allocation de numéro RPC

La pratique future de l'IANA pour traiter le partage ultérieur de l'espace de numéros de 32 bits est comme indiqué au paragraphe 8.3. Des informations détaillées pour l'administration du partage des blocs du paragraphe 8.3 sont données ci-dessous.

13.3.1 À allouer par l'IANA

Le premier bloc sera administré par l'IANA, avec la protection des allocations précédentes de Sun. Les allocations précédentes étaient restreintes à la gamme décimale 100000 à 399999 (0x000186a0 à 0x00061a7f) ; donc, l'IANA commencera les allocations au décimal 400000. Les numéros individuels devraient être accordés sur la base du premier arrivé, premier servi, et les blocs devraient être accordés selon les règles relatives à la taille de bloc.

13.3.2 Défini par l'administrateur local

Le bloc "Défini par l'administrateur local" est disponible pour tout domaine administratif local, de manière similaire aux gammes d'adresses IP réservées pour utilisation privée. L'utilisation attendue est par l'établissement d'une "autorité" de domaine local pour allouer les numéros dans cette gamme. Cette autorité va établir des politiques ou procédures à utiliser dans ce domaine local pour utiliser ou allouer des numéros RPC dans la gamme. Le domaine local devrait être suffisamment isolé pour qu'il y ait peu de chances que les applications RPC développées par d'autres domaines locaux puissent communiquer avec le domaine. Cela pourrait résulter en des conflits de numéros RPC, qui causerait l'échec d'une des applications. En l'absence d'un administrateur local, ce bloc peut être utilisé comme "utilisation privée" selon la [RFC5226].

13.3.3 Bloc transitoire

Le bloc "Transitoire" peut être utilisé par toute application RPC "comme disponible". Cette gamme est destinée aux services qui peuvent communiquer un numéro de programme RPC choisi de façon dynamique aux clients du service. Tout mécanisme peut être utilisé pour communiquer le numéro. Par exemple, une mémoire partagée quand le client et le serveur sont situés sur le même système ou un message réseau (RPC ou autre) qui dissémine le numéro choisi, peut être utilisé.

Le bloc transitoire n'est pas administré. Un service RPC utilise cette gamme en choisissant un numéro dans la gamme transitoire et tente d'enregistrer ce numéro auprès du lien RPC du système local (voir les procédures RPCBPROC_SET ou PMAPPROC_SET dans "Protocoles de lien pour ONC RPC Version 2", [RFC1833]). Si cela réussit, aucun autre service RPC n'utilise ce numéro et le lien RPC a alloué ce numéro à l'application RPC demandeuse. L'enregistrement est valide jusqu'à ce que le lien RPC se termine, ce qui ne va normalement arriver que si le système réamorce, causant la fin de toutes les applications, y compris le service RPC qui utilise le numéro transitoire. Si l'enregistrement du numéro transitoire échoue, une autre application RPC utilise le numéro et le demandeur doit choisir un autre numéro et essayer à nouveau. Pour éviter des conflits, la méthode recommandée est de choisir un nombre au hasard dans la gamme transitoire.

13.3.4 Bloc réservé

Les blocs "Réservé" sont disponibles pour une utilisation future. Les applications RPC ne doivent pas utiliser les numéros dans ces gammes sauf si leur utilisation est permise par une future action de l'IESG.

13.3.5 Sous blocs de numéro RPC

Les numéros RPC sont généralement alloués pour des services RPC spécifiques. Certaines applications exigent cependant plusieurs numéros RPC pour un service. L'exemple le plus courant est un service RPC qui a besoin d'avoir plusieurs instances du service actives simultanément sur un site spécifique. RPC n'a pas un "identifiant d'instance" dans le protocole, donc un mécanisme doit être mis en œuvre pour multiplexer les demandes RPC parmi diverses instances du service ou des numéros RPC uniques doivent être utilisés par chaque instance.

Dans ces cas, le protocole RPC utilisé avec les divers numéros peut être différent ou le même. Les numéros peuvent être alloués dynamiquement par l'application, ou au titre d'une décision administrative spécifique d'un site. Si possible, les

services RPC qui allouent dynamiquement les numéros RPC devraient utiliser le bloc de numéros RPC "Transitoire" défini au paragraphe 13.3.3. Si ce n'est pas possible, des sous blocs de numéros RPC peuvent être demandés.

L'allocation des sous blocs de numéros RPC est contrôlée par la taille du sous bloc demandé. "Spécification Rexigée" et "Approbation de l'IESG" sont utilisées comme défini au paragraphe 4.1 de la [RFC5226].

Taille de sous bloc	**Méthode d'allocation**	**Autorité**
Jusqu'à 100 numéros	Premier arrivé, premier servi	IANA
Jusqu'à 1000 numéros	Spécification exigée	IANA
Plus de 1000 numéros	Approbation de l'IESG requise	IESG

Note : les sous blocs peuvent être de toute taille. Les limites donnés ci-dessus sont des maximums, et de plus petites tailles de sous blocs sont permises.

Des sous blocs de jusqu'à 100 numéros peuvent être alloués par l'IANA sur la base du premier arrivé, premier servi. La description de service RPC incluse dans la gamme doit inclure une indication de comment le sous bloc est géré. Au minimum, la déclaration devrait indiquer si le sous bloc est utilisé avec un seul protocole RPC ou plusieurs protocoles RPC, et si les numéros sont alloués dynamiquement ou statiquement (par une action administrative).

Les sous blocs de jusqu'à 1000 numéros doivent être documentés en détails. La documentation doit décrire le ou les protocoles RPC qui sont utilisés dans la gamme. Elle doit aussi décrire comment les numéros dans le sous bloc sont alloués ou utilisés.

Les sous blocs de plus de 1000 numéros doivent être documentés comme décrit ci-dessus, et l'allocation doit être approuvée par l'IESG. Il est supposé que cela va être rare.

Afin d'éviter des demandes multiples de grands blocs de numéros, la règle suivante est proposée.

Les demandes jusque et incluant 100 numéros RPC sont traitées via la méthode d'allocation au premier arrivé, premier servi. Ce seuil de 100 numéros s'applique au nombre total de numéros RPC alloués à un individu ou entité. Par exemple, si un individu ou entité demande d'abord, disons, 70 numéros, et ensuite demande 40 numéros, alors la demande de 40 numéros va être allouée via la méthode de la spécification exigée. Tant que le nombre total de numéros a allouer n'excède pas 1000, l'IANA est libre de suivre l'allocation par spécification exigée pour les demandes incrémentaires de moins de 100 numéros.

Si un individu ou entité a moins de 1000 numéros et demande ultérieurement un ensemble supplémentaire de numéros tel que l'individu ou entité va recevoir plus de 1000 numéros, alors la demande supplémentaire va exiger l'approbation de l'IESG.

13.4 Allocation de numéro de nuance d'authentification RPC

Le second espace de numéros est l'identifiant de mécanisme d'authentification, ou numéro de "nuance". Ce numéro est utilisé pour distinguer entre les divers mécanismes d'authentification qui peuvent être facultativement utilisés avec un message RPC. Un identifiant d'authentification est utilisé dans le champ "nuance" de la structure "opaque_auth".

13.4.1 Politique d'allocation

L'Appendice B de ce document décrit les informations à envoyer à l'IANA pour demander un ou plusieurs numéros d'authentification RPC et les règles qui s'appliquent. L'IANA va mémoriser la demande à des fins de documentation et mettre les informations suivantes dans le registre public :

- La chaîne d'identifiant courte
- Le numéro d'authentification alloué
- La description courte de l'objet et de l'utilisation.

13.4.2 Nuances d'authentification contre pseudo nuances

Les récents progrès de la sécurité de RPC se sont éloignés des nouvelles nuances d'authentification telles qu'utilisées par AUTH_DH [DH], et se sont concentrées sur l'utilisation de la nuance existante RPCSEC_GSS [RFC2203] et ont inventé un nouveau mécanisme d'interface de programmation d'application de services génériques de sécurité (GSS-API, *Generic*

Security Services Application Programming Interface) qui peut être utilisé avec lui. Bien que le RPCSEC_GSS soit une nuance d'authentification allouée, l'utilisation d'un nouveau mécanisme RPCSEC_GSS avec le système de fichier réseau (NFS, *Network File System*) ([RFC1094], [RFC1813], et [RFC3530]) va exiger l'enregistrement de "pseudo nuances" qui sont utilisées pour négocier les mécanismes de sécurité de façon non ambiguë, comme défini dans la [RFC2623]. Les pseudo nuances existantes ont été allouées dans la gamme décimale 390 000 à 390 255. Les nouvelles demandes de pseudo-nuances seront accordées par l'IANA dans ce bloc sur la base du premier arrivé, premier servi.

Pour les demande non de pseudo nuances, l'IANA commencera par accorder les numéros de nuance d'authentification RPC à 400 000 sur la base du premier arrivé, premier servi pour éviter des conflits avec les numéros actuellement accordés.

Pour les nuances d'authentification ou les mécanismes RPCSEC_GSS à utiliser dans l'Internet, il est fortement conseillé qu'une RFC pour information ou sur la voie de la normalisation soit publiée pour décrire le comportement du mécanisme d'authentification et ses paramètres.

13.5 Allocation de numéro d'état d'authentification

L'espace de numéros final est la valeur d'état d'authentification ou "auth_stat" qui décrit la nature d'un problème trouvé durant une tentative pour authentifier ou valider l'authentification. La liste initiale complète de ces valeurs est à la Section 9 de ce document, dans l'énumération de "auth_stat". Il est supposé qu'il va être rare d'ajouter des valeurs, mais qu'un petit nombre de nouvelles valeurs peut être ajouté de temps en temps lorsque de nouvelles nuances d'authentification introduisent de nouvelles possibilités. Les numéros devraient être accordés sur la base du premier arrivé, premier servi pour éviter des conflits avec les numéros déjà attribués.

13.5.1 Politique d'allocation

L'Appendice B de ce document décrit les informations à envoyer à l'IANA pour demander une ou plusieurs valeurs de auth_stat et les règles qui s'appliquent. L'IANA va mémoriser la demande à des fins de documentation et mettre les informations suivantes dans le registre public :

- La chaîne d'identifiant courte
- Le numéro de auth_stat alloué
- La description courte et l'utilisation de l'objet.

14. Considérations sur la sécurité

AUTH_SYS comme décrit à l'Appendice A est connu pour n'être pas sûr du fait de l'absence de vérificateur pour permettre la validation de l'accréditif. AUTH_SYS NE DEVRAIT PAS être utilisé pour des services qui permettent aux clients de modifier les données. AUTH_SYS NE DOIT PAS être spécifié comme RECOMMANDÉ ou EXIGÉ dans un service RPC sur la voie de la normalisation.

Comme mentionné aux paragraphes 8.2 et 13.4.2, AUTH_DH est considéré comme obsolète et non sûr ; voir la [RFC2695]. AUTH_DH NE DEVRAIT PAS être utilisé pour les services qui permettent aux clients de modifier les données. AUTH_DH NE DOIT PAS être spécifié comme RECOMMANDÉ ou EXIGÉ pour un service RPC sur la voie de la normalisation.

La [RFC2203] définit une nouvelle nuance de sécurité, RPCSEC_GSS, qui permet aux mécanismes GSS-API [RFC2743] d'être utilisés pour sécuriser RPC. Tous les programmes RPC non triviaux développés à l'avenir devraient mettre en œuvre la sécurité fondée sur RPCSEC_GSS de façon appropriée. La [RFC2623] décrit comment cela a été fait pour un programme de RPC largement déployé.

Les services RPC sur la voie de la normalisation DOIVENT rendre obligatoire la prise en charge de RPCSEC_GSS, et DOIVENT rendre obligatoire la prise en charge d'une pseudo-nuance d'authentification avec les niveaux appropriés de sécurité, selon le besoin de simple authentification, d'intégrité (autrement dit, de non répudiation) ou de confidentialité des données.

Appendice A. Authentification du système

Le client peut souhaiter s'identifier, par exemple, comme il est identifié sur un système UNIX(tm). La nuance de l'accréditif de client est "AUTH_SYS". Les données opaques qui constituent l'accréditif codent la structure suivante :

```
struct authsys_parms {
  unsigned int stamp;
  string machinename<255>;
  unsigned int uid;
  unsigned int gid;
  unsigned int gids<16>;
};
```

"stamp" est un identifiant arbitraire que la machine appelante peut générer. "machinename" est le nom de la machine de l'appelant (comme "krypton"). "uid" est l'identifiant effectif de l'appelant. "gid" est l'identifiant de groupe effectif de l'appelant. Les "gid" sont un dispositif compté de groupes qui contiennent l'appelant comme membre. Le vérificateur qui accompagne l'accréditif devrait avoir la valeur de nuance "AUTH_NONE" (définie plus haut). Noter que cet accréditif est seulement unique au sein d'un domaine particulier de noms de machines, uid, et gid.

La valeur de nuance du vérificateur reçue dans le message de réponse provenant du serveur peut être "AUTH_NONE" ou "AUTH_SHORT". Dans le cas de "AUTH_SHORT", les octets de la chaîne de réponse du vérificateur codent une structure opaque. Cette nouvelle structure opaque peut maintenant être passée au serveur au lieu de la nuance d'accréditif original "AUTH_SYS". Le serveur peut tenir une antémémoire qui transpose les structures opaques abrégées (repassées au moyen d'un vérificateur de réponse de style "AUTH_SHORT") en les accréditifs originaux de l'appelant. L'appelant peut économiser la bande passante du réseau et les cycles de cpu du serveur en utilisant l'accréditif abrégé.

Le serveur peut purger la structure opaque abrégée à tout moment. Si cela arrive, le message d'appel de procédure distante va être rejeté à cause d'une erreur d'authentification. La raison de l'échec va être "AUTH_REJECTEDCRED". À ce point, le client peut souhaiter essayer le style d'accréditif "AUTH_SYS" original.

On notera que l'utilisation de cette nuance d'authentification ne garantit par elle-même aucune sécurité aux utilisateurs ou fournisseurs d'un service. L'authentification fournie par ce schéma peut être considérée comme légitime seulement quand les applications qui utilisent ce schéma et le réseau peuvent être sécurisés en externe, et que des adresses de transport privilégiées sont utilisées pour les points d'extrémité communicants (un exemple en est l'utilisation d'accès TCP/UDP privilégiés dans les systèmes UNIX -- noter que tous les systèmes n'appliquent pas des mécanismes d'adresse de transport privilégiée).

Appendice B. Demande de numéros relatifs à RPC à l'IANA

Les numéros de programme RPC, numéros de nuance d'authentification, et numéros d'état d'authentification qui doivent être uniques à travers tous les réseaux sont alloués par l'IANA. Pour demander un seul numéro ou bloc de numéros, un message électronique doit être envoyé à l'IANA <iana@iana.org> avec les informations suivantes :

- le type de numéro) (numéro de programme ou numéro de nuance d'authentification ou numéro d'état d'authentification) recherché,
- combien de numéros sont recherchés,
- le nom de la personne ou société qui va utiliser le numéro,
- une "chaîne d'identifiant" qui associe le numéro à un service,
- l"adresse de messagerie de la personne de contact pour le service qui va utiliser le numéro,
- une brève description de l'objet et de l'utilisation du numéro,
- si un numéro de nuance d'authentification est recherché et si le numéro va être une "pseudo-nuance" destinée à être utilisée avec RPCSEC_GSS et NFS, des transpositions analogues à celles du paragraphe 4.2 de la [RFC2623]

Des numéros spécifiques ne peuvent pas être demandés. Les numéros sont alloués au premier qui les demande.

Pour tous numéros de nuance d'authentification RPC et d'état d'authentification à utiliser sur l'Internet, il est fortement conseillé qu'une RFC pour information ou sur la voie de la normalisation soit publiée, décrivant le comportement et les paramètres du mécanisme d'authentification.

Appendice C. Allocations actuelles de numéros

Numéros RPC alloués par Sun

Description/propriétaire	Numéro de programme RPC	Nom abrégé
portmapper	100000	pmapprog portmap rpcbind
remote stats	100001	rstatprog
remote utilisateurs	100002	rusersprog
nfs	100003	nfs
yellow pages (NIS)	100004	ypprog ypserv
mount demon	100005	mountprog
remote dbx	100006	dbxprog
yp binder (NIS)	100007	ypbindprog ypbind
shutdown msg	100008	wall
yppasswd serveur	100009	yppasswdprog yppasswdd
ether stats	100010	etherstatprog
disk quotas	100011	rquota
spray packets	100012	spray
3270 mapper	100013	ibm3270prog
RJE mapper	100014	ibmrjeprog
selection service	100015	selnsvcprog
remote database access	100016	rdatabaseprog
remote execution	100017	rexec
Alice Office Automation	100018	aliceprog
scheduling service	100019	schedprog
local lock manager	100020	lockprog llockmgr
network lock manager	100021	netlockprog nlockmgr
x.25 inr protocol	100022	x25prog
status monitor 1	100023	statmon1
status monitor 2	100024	statmon2
selection library	100025	selnlibprog
boot paramètres service	100026	bootparam
mazewars game	100027	mazeprog
yp update (NIS)	100028	ypupdateprog ypupdate
key serveur	100029	keyserveprog
secure login	100030	securecmdprog
nfs net forwarder init	100031	netfwdiprog
nfs net forwarder trans	100032	netfwdtprog
sunlink MAP	100033	sunlinkmap
network monitor	100034	netmonprog
lightweight database	100035	dbaseprog
password authorization	100036	pwdauthprog
translucent file svc	100037	tfsprog
nse serveur	100038	nseprog
nse activate daemon	100039	nse_activate_prog
sunview help	100040	sunview_help_prog
pnp install	100041	pnp_prog
ip addr allocator	100042	ipaddr_alloc_prog
show filehandle	100043	filehandle
MVS NFS mount	100044	mvsnfsprog
remote user file operations	100045	rem_fileop_user_prog
batched ypupdate	100046	batch_ypupdateprog
network execution mgr	100047	nem_prog
raytrace/mandelbrot remote daemon	100048	raytrace_rd_prog
raytrace/mandelbrot local daemon	100049	raytrace_ld_prog
remote group file operations	100050	rem_fileop_group_prog
remote system file operations	100051	rem_fileop_system_prog
remote system role operations	100052	rem_system_role_prog
gpd lego fb simulator	100053	[inconnu]
gpd simulator interface	100054	[inconnu]
ioadmd	100055	ioadmd

filemerge	100056	filemerge_prog
Name Binding Program	100057	namebind_prog
sunlink NJE	100058	njeprog
MVSNFS get attribute service	100059	mvsattrprog
SunAccess/SunLink resource manager	100060	rmgrprog
UID allocation service	100061	uidallocprog
license broker	100062	lbserverprog
NETlicense client binder	100063	lbbinderprog
GID allocation service	100064	gidallocprog
SunIsam	100065	sunisamprog
Remote Debug Server	100066	rdbsrvprog
Network Directory Daemon	100067	[inconnu]
Network Calendar Program	100068	cmsd cm
ypxfrd	100069	ypxfrd
rpc.timed	100070	timedprog
bugtraqd	100071	bugtraqd
	100072	[inconnu]
Connectathon Billboard – NFS	100073	[inconnu]
Connectathon Billboard – X	100074	[inconnu]
Sun tool for scheduling rooms	100075	schedroom
Authentication Negotiation	100076	authnegotiate_prog
Database manipulation	100077	attribute_prog
Kerberos authentification daemon	100078	kerbprog
Internal testing product (no name)	100079	[inconnu]
Sun Consulting Special	100080	autodump_prog
Event protocol	100081	event_svc
bugtraq_qd	100082	bugtraq_qd
ToolTalk et Link Service Project	100083	database service
Consulting Services	100084	[inconnu]
Consulting Services	100085	[inconnu]
Consulting Services	100086	[inconnu]
Jupiter Administration	100087	adm_agent admind
	100088	[inconnu]
	100089	[inconnu]
Dual Disk support	100090	libdsd/dsd
DocViewer 1.1	100091	[inconnu]
ToolTalk	100092	remote_activation_svc
Consulting Services	100093	host_checking
SNA peer-to-peer	100094	[inconnu]
Roger Riggs	100095	searchit
Robert Allen	100096	mesgtool
SNA	100097	[inconnu]
SISU	100098	networked version of CS5
NFS Automount File System	100099	autofs
	100100	msgboard
event dispatching agent [eventd]	100101	netmgt_eventd_prog
statistics/event logger [netlogd]	100102	netmgt_netlogd_prog
topology display manager [topology]	100103	netmgt_topology_prog
syncstat agent [syncstatd]	100104	netmgt_syncstatd_prog
ip packet stats agent [ippktd]	100105	netmgt_ippktd_prog
netmgt config agent [configd]	100106	netmgt_configd_prog
restat agent [restatd]	100107	netmgt_restatd_prog
lpq agent [lprstatd]	100108	netmgt_lprstatd_prog
netmgt activity agent [mgtlogd]	100109	netmgt_mgtlogd_prog
proxy DECnet NCP agent [proxydni]	100110	netmgt_proxydni_prog
topology mapper agent [mapperd]	100111	netmgt_mapperd_prog
netstat agent [netstatd]	100112	netmgt_netstatd_prog
sample netmgt agent [sampled]	100113	netmgt_sampled_prog
X.25 statistics agent [vcstatd]	100114	netmgt_vcstatd_prog
Frame Relay	100128	[inconnu]
PPP agent	100129	[inconnu]

localhad	100130	rpc.localhad
layers2	100131	na.layers2
token ring agent	100132	na.tr
related to lockd et statd	100133	nsm_addr
Kerberos project	100134	kwarn
ertherif2	100135	na.etherif2
hostmem2	100136	na.hostmem2
iostat2	100137	na.iostat2
snmpv2	100138	na.snmpv2
Cooperative Console	100139	cc_sender
na.cpustat	100140	na.cpustat
Sun Cluster SC3.0	100141	rgmd_receptionist
	100142	fed
Network Storage	100143	rdc
Sun Cluster products	100144	nafo
SunCluster 3.0	100145	scadmd
ASN.1	100146	amiserv
	100147	amiaux # code et décode BER et DER
Delegate Management Server	100148	dm
	100149	rkstat
	100150	ocfserv
	100151	sccheckd
	100152	autoclientd
	100153	sunvts
	100154	ssmond
	100155	smserverd
	100156	test1
	100157	test2
	100158	test3
	100159	test4
	100160	test5
	100161	test6
	100162	test7
	100163	test8
	100164	test9
	100165	test10
	100166	nfsmapid
	100167	SUN_WBEM_C_CIMON_HANDLE
	100168	sacmmd
	100169	fmd_adm
	100170	fmd_api
	100171	[inconnu]
	100172	idmapd
non alloués	100173 - 100174	
snmptrap	100175	na.snmptrap
non alloués	100176-100199	
non alloué	100200	
MVS/NFS Memory usage stats serveur	100201	[inconnu]
Netapp	100202-100207	
non alloués	100208-100210	
8.0 SunLink SNA RJE	100211	[inconnu]
8.0 SunLink SNA RJE	100212	[inconnu]
	100213	ShowMe
	100214	[inconnu]
	100215	[inconnu]
AUTH_RSA Key service	100216	keyrsa
SunSelect PC license service	100217	[inconnu]
WWCS (Corporate)	100218	sunsolve
	100219	cstatd
X/Open Federated Naming	100220	xfn_server_prog
Kodak Color Management System	100221	kcs_network_io kcs

HA-DBMS	100222	ha_dbms_serv
	100223-100225	[inconnu]
	100226	hafaultd
NFS ACL Service	100227	nfs_acl
distributed lock manager	100228	dlmd
	100229	metad
	100230	metamhd
	100231	nfsauth
	100232	sadmind
	100233	ufsd
	100234	grpservd
	100235	cachefsd
	100236	msmprog Media_Server
	100237	ihnamed
	100238	ihnetd
	100239	ihsecured
	100240	ihclassmgrd
	100241	ihrepositoryd
	100242	metamedd rpc.metamedd
	100243	contentmanager cm
	100244	symon
	100245	pld genesil
	100246	ctid cluster_transport_interface
	100247	ccd cluster_configuration_db
	100248	pmfd
	100249	dmi2_client
	100250	mfs_admin
	100251	ndshared_unlink
	100252	ndshared_touch
	100253	ndshared_slink
	100254	cbs control_board_serveur
	100255	skiserv
	100256	nfsxa nfsxattr
	100257	ndshared_disable
	100258	ndshared_enable
	100259	sms_account_admin
	100260	sms_modem_admin
	100261	sms_r_login
	100262	sms_r_subaccount_mgt
	100263	sms_service_admin
	100264	session_admin
	100265	canci_ancs_program
	100266	canci_sms_program
	100267	msmp
	100268	halck
	100269	halogmsg
	100270	nfs_id_map
	100271	ncall
	100272	hmip
	100273	repl_mig
	100274	repl_mig_cb
NIS+	100300	nisplus
NIS+	100301	nis_cachemgr
NIS+ call back protocol	100302	[inconnu]
NIS+ Password Update Daemon	100303	nispasswdd
FNS context update in NIS	100304	fnsypd
	100305	[inconnu]
	100306	[inconnu]
	100307	[inconnu]
	100308	[inconnu]
	100309	[inconnu]

non alloués 100310 - 100398
nfscksum 100399 nfscksum
network utilization agent 100400 netmgt_netu_prog
network rpc ping agent 100401 netmgt_rping_prog
100402 na.shell
picsprint 100403 na.picslp
100404 traps
100405 - 100409 [inconnu]
100410 jdsagent
100411 na.haconfig
100412 na.halhost
100413 na.hadtsrvc
100414 na.hamdstat
100415 na.neoadmin
100416 ex1048prog
rdmaconfig 100417 rpc.rdmaconfig
IETF NFSv4 Working Group - FedFS 100418 - 100421
100422 mdcommd
100423 kiprop krb5_iprop
100424 stsf
non alloués 100425 - 100499
Sun Microsystems 100500 - 100531 [inconnu]
100532 ucmmstate
100533 scrcmd
non alloués 100534 - 100999
nse link daemon 101002 nselinktool
nse link application 101003 nselinkapp
non alloués 101004 - 101900
101901 [inconnu]
non alloués 101902 - 101999
AssetLite 102000 [inconnu]
PagerTool 102001 [inconnu]
Discover 102002 [inconnu]
non alloués 102003 - 105000
ShowMe 105001 sharedapp
Registry 105002 REGISTRY_PROG
Print-serveur 105003 print-server
Proto-serveur 105004 proto-server
Notification-serveur 105005 notification-server
Transfer-agent-serveur 105006 transfer-agent-server
non alloués 105007 - 110000
110001 tsolrpcb
110002 tsolpeerinfo
110003 tsolboot
120001 cmip na.cmip
120002 na.osidiscover
120003 cmiptrap
non alloués 120004 - 120099
120100 eserver
120101 repserver
120102 swserver
120103 dmd
120104 ca
non alloués 120105 - 120125
120126 nf_fddi
120127 nf_fddismt7_2
non alloués 120128 - 150000
pc passwd authorization 150001 pcnfsdprog
TOPS name mapping 150002 [inconnu]
TOPS external attribute storage 150003 [inconnu]
TOPS hierarchical file system 150004 [inconnu]

TOPS NFS transparency extensions 150005 [inconnu]
PC NFS License 150006 pcnfslicense
RDA 150007 rdaprog
WabiServer 150008 wsprog
WabiServer 150009 wsrlprog
non alloués 150010 - 160000
160001 nihon-cm
160002 nihon-ce
non alloués 160003 - 170099
170100 domf_daemon0
170101 domf_daemon1
170102 domf_daemon2
170103 domf_daemon3
170104 domf_daemon4
170105 domf_daemon5
non alloués 170106 - 179999
180000 cecprog
180001 cecsysprog
180002 cec2cecprog
180003 cesprog
180004 ces2cesprog
180005 cet2cetprog
180006 cet2cetdoneprog
180007 cetcomprog
180008 cetsysprog
180009 cghapresenceprog
180010 cgdmsyncprog
180011 cgdmcnscliprog
180012 cgdmcrcscliprog
180013 cgdmcrcssvcproG
180014 chmprog
180015 chmsysprog
180016 crcsapiprog
180017 ckptmprog
180018 crimcomponentprog
180019 crimqueryprog
180020 crimsecondaryprog
180021 crimservicesprog
180022 crimsyscomponentprog
180023 crimsysservicesprog
180024 csmagtapiprog
180025 csmagtcallbackprog
180026 csmreplicaprog
180027 csmsrvprog
180028 csscltprog
180029 csscsvrprog
180030 csscopresultprog
non alloués 180031 - 199999
200000 pyramid_nfs
200001 pyramid_reserved
200002 cadds_image
200003 stellar_name_prog
200004 [inconnu]
200005 [inconnu]
200006 pacl
200007 lookupids
200008 ax_statd_prog
200009 ax_statd2_prog
200010 edm
200011 dtedirwd
200012 [inconnu]

	200013	[inconnu]
	200014	[inconnu]
	200015	[inconnu]
	200016	easerpcd
	200017	rlxnfs
	200018	sascuiddprog
	200019	knfsd
	200020	ftnfsd ftnfsd_program
	200021	ftsyncd ftsyncd_program
	200022	ftstatd ftstatd_program
	200023	exportmap
	200024	nfs_metadata
non alloués	200025 - 200200	
	200201	ecoad
	200202	eamon
	200203	ecolic
	200204	cs_printstatus_svr
	200205	ecodisc
non alloués	200206 - 300000	
	300001	adt_rflockprog
	300002	columbine1
	300003	system33_prog
	300004	frame_prog1
	300005	uimxprog
	300006	rvd
	300007	entombing daemon
	300008	account mgmt system
	300009	frame_prog2
	300010	beeper access
	300011	dptuprog
	300012	mx-bcp
	300013	instrument-file-access
	300014	file-system-statistics
	300015	unify-database-server
	300016	tmd_msg
	300017	[inconnu]
	300018	[inconnu]
	300019	automounter access
	300020	lock server
	300021	[inconnu]
	300022	office-automation-1
	300023	office-automation-2
	300024	office-automation-3
	300025	office-automation-4
	300026	office-automation-5
	300027	office-automation-6
	300028	office-automation-7
	300029	local-data-manager
	300030	chide
	300031	csi_program
	300032	[inconnu]
	300033	online-help
	300034	case-tool
	300035	delta
	300036	rgi
	300037	instrument-config-server
	300038	[inconnu]
	300039	[inconnu]
	300040	dtia-rpc-server
	300041	cms
	300042	viewer

300043	aqm
300044	exclaim
300045	masterplan
300046	fig_tool
300047	[inconnu]
300048	[inconnu]
300049	[inconnu]
300050	remote-lock-manager
300051	[inconnu]
300052	gdebug
300053	ldebug
300054	rscanner
300055	[inconnu]
300056	[inconnu]
300057	[inconnu]
300058	[inconnu]
300059	[inconnu]
300060	[inconnu]
300061	[inconnu]
300062	[inconnu]
300063	[inconnu]
300064	[inconnu]
300065	[inconnu]
300066	nSERVER
300067	[inconnu]
300068	[inconnu]
300069	[inconnu]
300070	[inconnu]
300071	BioStation
300072	[inconnu]
300073	NetProb
300074	Logging
300075	Logging
300076	[inconnu]
300077	[inconnu]
300078	[inconnu]
300079	[inconnu]
300080	[inconnu]
300081	[inconnu]
300082	sw_twin
300083	remote_get_login
300084	odcprog
300085	[inconnu]
300086	[inconnu]
300087	[inconnu]
300088	[inconnu]
300089	[inconnu]
300090	[inconnu]
300091	smartdoc
300092	superping
300093	distributed-chembench
300094	uacman/alfil-uacman
300095	ait_rcagent_prog
300096	ait_rcagent_appl_prog
300097	smart
300098	ecoprog
300099	leonardo
300100	[inconnu]
300101	[inconnu]
300102	[inconnu]
300103	[inconnu]

300104 [inconnu]
300105 [inconnu]
300106 [inconnu]
300107 [inconnu]
300108 wingz
300109 teidan
300110 [inconnu]
300111 [inconnu]
300112 [inconnu]
300113 [inconnu]
300114 [inconnu]
300115 [inconnu]
300116 cadc_fhlockprog
300117 highscan
300118 [inconnu]
300119 [inconnu]
300120 [inconnu]
300121 opennavigator
300122 aarpcxfer
300123 [inconnu]
300124 [inconnu]
300125 [inconnu]
300126 groggs
300127 licsrv
300128 issdemon
300129 [inconnu]
300130 maximize
300131 cgm_serveur
300132 [inconnu]
300133 agent_rpc
300134 docmaker
300135 docmaker
300136 [inconnu]
300137 [inconnu]
300138 [inconnu]
300139 iesx
300140 [inconnu]
300141 [inconnu]
300142 [inconnu]
300143 [inconnu]
300144 smart-mbs
300145 [inconnu]
300146 [inconnu]
300147 docimage
300148 [inconnu]
300149 dmc-interface
300150 [inconnu]
300151 jss
300152 [inconnu]
300153 arimage
300154 xdb-workbench
300155 frontdesk
300156 dmc
300157 expressight-6000
300158 graph service program
300159 [inconnu]
300160 [inconnu]
300161 [inconnu]
300162 [inconnu]
300163 [inconnu]
300164 [inconnu]

300165 [inconnu]
300166 [inconnu]
300167 [inconnu]
300168 [inconnu]
300169 [inconnu]
300170 [inconnu]
300171 [inconnu]
300172 [inconnu]
300173 [inconnu]
300174 [inconnu]
300175 [inconnu]
300176 rlpr
300177 nx_hostdprog
300178 netuser-x
300179 rmntprog
300180 [inconnu]
300181 mipe
300182 [inconnu]
300183 collectorprog
300184 uslookup_PROG
300185 viewstation
300186 iate
300187 [inconnu]
300188 [inconnu]
300189 [inconnu]
300190 imsvtprog
300191 [inconnu]
300192 [inconnu]
300193 [inconnu]
300194 pmdb
300195 pmda
300196 [inconnu]
300197 [inconnu]
300198 trend_idbd
300199 rres
300200 sd.masterd
300201 sd.executiond
300202 sd.listend
300203 sd.reserve1
300204 sd.reserve2
300205 msbd
300206 stagedprog
300207 mountprog
300208 watchdprog
300209 pms
300210 [inconnu]
300211 session_server_program
300212 session_program
300213 debug_serverprog
300214 [inconnu]
300215 [inconnu]
300216 paceprog
300217 [inconnu]
300218 mbus
300219 aframes2ps
300220 npartprog
300221 cm1server
300222 cm1bridge
300223 sailfrogfaxprog
300224 sailfrogphoneprog
300225 sailfrogvmailprog

300226 wserviceprog arcstorm
300227 hld
300228 alive
300229 radsp
300230 radavx
300231 radview
300232 rsys_prog
300233 rsys_prog
300234 fm_rpc_prog
300235 aries
300236 uapman
300237 ddman
300238 top
300239 [inconnu]
300240 trendlink
300241 licenseprog
300242 statuslicenseprog
300243 oema_rmpf_svc
300244 oema_smpf_svc
300245 oema_rmsg_svc
300246 grapes-sd
300247 ds_master
300248 ds_transfer
300249 ds_logger
300250 ds_query
300251 [inconnu]
300252 [inconnu]
300253 nsd_prog
300254 browser
300255 epoch
300256 floorplanner
300257 reach
300258 tactic
300259 cachescientific1
300260 cachescientific2
300261 desksrc_prog
300262 photo3d1
300263 photo3d2
300264 [inconnu]
300265 soundmgr
300266 s6k
300267 aims_referenced_text_processor
300268 xess
300269 ds_queue
300270 [inconnu]
300271 orionscanplus
300272 openlink-xx
300273 kbmsprog
300274 [inconnu]
300275 futuresource
300276 the_xprt
300277 cmg_srvprog
300278 [inconnu]
300279 [inconnu]
300280 front
300281 [inconnu]
300282 [inconnu]
300283 [inconnu]
300284 conmanprog
300285 jincv2
300286 isls

300287 systemstatprog
300288 fxpsprog
300289 callpath
300290 axess
300291 armor_rpcd
300292 armor_dictionary_rpcd
300293 armor_miscd
300294 filetransfer_prog
300295 bl_swda
300296 bl_hwda
300297 [inconnu]
300298 [inconnu]
300299 [inconnu]
300300 filemon
300301 acunetprog
300302 rbuild
300303 assistprog
300304 tog
300305 [inconnu]
300306 sns7000
300307 igprog
300308 tgprog
300309 plc
300310 pxman pxlsprog
300311 hde_serveur hdeserver
300312 tsslicenseprog
300313 rpc.explorerd
300314 chrd
300315 tbisam
300316 tbis
300317 adsprog
300318 sponsorprog
300319 querycmprog
300320 [inconnu]
300321 [inconnu]
300322 mobil1
300323 sld service_locator_daemon
300324 linkprog
300325 codexdaemonprog
300326 drprog
300327 ressys_commands
300328 stamp
300329 matlab
300330 sched1d
300331 upcprog
300332 xferbkch
300333 xfer
300334 qbthd
300335 qbabort
300336 lsd
300337 geomgrd
300338 generic_fts
300339 ft_ack
300340 lymb
300341 vantage
300342 cltstd clooptstdprog
300343 clui clui_prog
300344 testerd tstdprog
300345 extsim
300346 cmd_dispatch maxm_ems
300347 callpath_receive_program

300348 x3270prog
300349 sbc_lag
300350 sbc_frsa
300351 sbc_frs
300352 atommgr
300353 geostrat
300354 dbvialu6.2
300355 [inconnu]
300356 fxncprog
300357 infopolic
300358 [inconnu]
300359 aagns
300360 aagms
300361 [inconnu]
300362 clariion_mgr
300363 setcimrpc
300364 virtual_protocol_adapter
300365 unibart
300366 uniarch
300367 unifile
300368 unisrex
300369 uniscmd
300370 rsc
300371 set
300372 desaf-ws/key
300373 reeldb
300374 nl
300375 rmd
300376 agcd
300377 rsynd
300378 rcnlib
300379 rcnlib_attach
300380 evergreen_mgmt_agent
300381 fx104prog
300382 rui remote_user_interface
300383 ovomd
300384 [inconnu]
300385 [inconnu]
300386 system_server
300387 pipecs cs_pipeprog ppktrpc
300388 uv-net univision
300389 auexe
300390 audip
300391 mqi
300392 eva
300393 eeei_réservé_1
300394 eeei_réservé_2
300395 eeei_réservé_3
300396 eeei_réservé_4
300397 eeei_réservé_5
300398 eeei_réservé_6
300399 eeei_réservé_7
300400 eeei_réservé_8
300401 cprlm
300402 wg_idms_manager
300403 timequota
300404 spiff
300405-300414 ov_oem_svc
300415 ov_msg_ctlg_svc
300416 ov_advt_reg_svc
300417-300424 showkron

	300425	daatd
	300426	swiftnet
	300427	ovomdel
	300428	ovomreq
	300429	msg_dispatcher
	300430	pcshare serveur
	300431	rcvs
	300432	fdfserver
	300433	bssd
	300434	drdd
	300435	mif_gutsprog
	300436	mif_guiprog
	300437	twolfd
	300438	twscd
	300439	nwsbumv
	300440	dgux_mgr
	300441	pfxd
	300442	tds
	300443	ovomadmind
	300444	ovomgate
	300445	omadmind
	300446	nps
	300447	npd
	300448	tsa
	300449	cdaimc
non alloués	300450-300452	
	300453	ckt_implementation
	300454	mda-tactical
non alloués	300455-300458	
	300459	atrrun
	300460	RoadRunner
	300461	nas
	300462	undelete
	300463	ovacadd
	300464	tbdesmai
	300465	arguslm
	300466	dmd
	300467	drd
	300468	fm_help
	300469	ftransrpc_prog
	300470	finrisk
	300471	dg_pc_idisched
	300472	dg_pc_idiserv
	300473	apd
	300474	ap_sspd
	300475	callpatheventrecorder
	300476	flc
	300477	dg_osm
	300478	dspnamed
	300479	iqddsrv
	300480	iqjobsrv
	300481	tacosxx
	300482	wheeldbmg
	300483	cnxmgr_nm_prog
	300484	cnxmgr_cfg_prog
	300485	3dsmapper
	300486	ids
	300487	imagine_rpc_svc
	300488	lfn
	300489	salesnet
	300490	defaxo

300491	dbqtsd	
300492	kms	
300493	rpc.iced	
300494	calc2s	
300495	ptouidprog	
300496	docsls	
300497	new	
300498	collagebdg	
300499	ars_serveur	
300500	ars_client	
300501	vr_catalog	
300502	vr_tdb	
300503	ama	
300504	evama	
300505	conama	
300506	service_process	
300507	reuse_proxy	
300508	mars_ctrl	
300509	mars_db	
300510	mars_com	
300511	mars_admch	
300512	tbpipcip	
300513	top_acs_svc	
300514	inout_svc	
300515	csoft_wp	
300516	mcfs	
300517	eventprog	
300518	dg_pc_idimsg	
300519	dg_pc_idiaux	
300520	atsr_gc	
300521	alarm alarm_prog	
300522	fts_prog	
300523	dcs_prog	
300524	ihb_prog	
300525	[inconnu]	
300526	[inconnu]	
300527	clu_info_prog	
300528	rmfm	
300529	c2sdocd	
300530	interahelp	
300531	callpathasyncmsghandler	
300532	optix_arc	
300533	optix_ts	
300534	optix_wf	
300535	maxopenc	
300536	cev cev_serveur	
300537	sitewideprog	
300538	drs	
300539	drsdm	
300540	dasgate	
300541	dcdbd	
300542	dcpsd	
300543	supportlink_prog	
300544	broker	
300545	listner	
300546	multiaccess	
300547	spai_interface	
300548	spai_adaption	
300549	chimera_ci	chimera_clientinterface
300550	chimera_pi	chimera_processinvoker
300551	teamware_fl	teamware_foundationlevel

```
300552  teamware_sl         teamware_systemlevel
300553  teamware_ui         teamware_userinterface
300554  lprm
300555  mpsprog   Mensuration_Proxy_Server
300556  mo_symdis
300557  retsideprog
300558  slp
300559  slm-api
300560  im_rpc teamconference
300561  license_prog license
300562  stuple stuple_prog
300563  upasswd_prog
300564  gentranmentorsecurity
300565  gentranmentorprovider
300566  latituded   latitude_license_server
300567  gentranmentorreq1
300568  gentranmentorreq2
300569  gentranmentorreq3
300570  rj_serveur
300571  gws-rdb
300572  gws-mpmd
300573  gws-spmd
300574  vwcalcd
300575  vworad
300576  vwsybd
300577  vwave
300578  online_assistant
300579  internet_assistant
300580  spawnd
300581  procmgrg
300582  cfgdbd
300583  logutild
300584  ibis
300585  ibisaux
300586  aapi
300587  rstrt
300588  hbeat
300589  pcspu
300590  empress
300591  sched_server        LiveScheduler
300592  path_server         LiveScheduler
300593  c2sdmd
300594  c2scf
300595  btsas
300596  sdtas
300597  appie
300598  dmi
300599  pscd        panther software corp daemon
300600  sisd
300601  cpwebserver
300602  wwcommo
300603  mx-mie
300604  mx-mie-debug
300605  idmn
300606  ssrv
300607  vpnserver
300608  samserver
300609  sams_serveur
300610  chrysalis
300611  ddm
300612  ddm-est
```

300613 mx-bcp-debug
300614 upmrd
300615 upmdsd
300616 res
300617 colortron
300618 zrs
300619 afpsrv
300620 apxft
300621 nrp
300622 hpid
300623 mailwatch
300624 fos bc_fcrb_receiver
300625 cs_sysadmin_svr
300626 cs_controller_svr
300627 nokia_nms_eai
300628 dbg
300629 remex
300630 cs_bind
300631 idm
300632 prpasswd
300633 iw-pw
300634 starrb
300635 Impress_Server
300636 colorstar
300637 gwugui
300638 gwsgui
300639 dai_command_proxy
300640 dai_alarm_serveur
300641 dai_fui_proxy
300642 spai_command_proxy
300643 spai_alarm_serveur
300644 iris
300645 hcxttp
300646 updatedb rsched
300647 urnd urn
300648 iqwpsrv
300649 dskutild
300650 online
300651 nlserv
300652 acsm
300653 dg_clar_sormsg
300654 wwpollerrpc
300655 wwmodelrpc
300656 nsprofd
300657 nsdistd
300658 recollect
300659 lssexecd lss_res
300660 lssagend lss_rea
300661 cdinfo
300662 sninsr_addon
300663 mm-sap
300664 ks
300665 psched
300666 tekdvfs
300667 storxll
300668 nisse
300669 lbadvise
300670 atcinstaller
300671 atntstarter
300672 NetML
300673 tdmesmge

300674 tdmesmgd
300675 tdmesmgt
300676 olm
300677 mediamanagement
300678 rdbprog fieldowsrv
300679 rpwdprog rpwd
300680 sapi-trace
300681 sapi-master-daemon
300682 omdcuprog om-dcu
300683 wwprocmon
300684 tndidprog
300685 rkey_setsecretprog
300686 asdu_serveur_prog
300687 pwrcntrl
300688 siunixd
300689 wmapi
300690 cross_reference_ole
300691 rtc
300692 disp
300693 sql_compilation_agent
300694 tnsysprog
300695 ius-sapimd
300696 apteam-dx
300697 rmsrpc
300698 seismic_system
300699 remote
300700 tt1_ts_event nokia_nms
300701 fxrs
300702 onlicense
300703 vxkey
300704 dinis
300705 sched2d schedule-2
300706 sched3d schedule-3
300707 sched4d schedule-4
300708 sched5d schedule-5
300709 sched6d schedule-6
300710 sched7d schedule-7
300711 sched8d schedule-8
300712 sched9d schedule-9
300713 adtsqry
300714 adserv
300715 adrepserv
300716 [inconnu]
300717 caad
300718 caaui
300719 cescda
300720 vcapiadmin
300721 vcapi20
300722 tcfs
300723 csed
300724 nothand
300725 hacb
300726 nfauth
300727 imlm
300728 bestcomm
300729 lprpasswd
300730 rprpasswd
300731 proplistd
300732 mikomomc
300733 arepa-cas
300734 [inconnu]

300735	[inconnu]
300736	ando_ts
300737	intermezzo
300738	ftel-sdh-request
300739	ftel-sdh-response
300740	[inconnu]
300741	[inconnu]
300742	[inconnu]
300743	[inconnu]
300744	[inconnu]
300745	vrc_abb
300746	vrc_comau
300747	vrc_fanuc
300748	vrc_kuka
300749	vrc_reis
300750	hp_sv6d
300751	correntmgr01
300752	correntike
300753	[inconnu]
300754	[inconnu]
300755	intransa_location
300756	intransa_management
300757	intransa_federation
300758	portprot
300759	ipmiprot
300760	aceapi
300761	f6000pss
300762	vsmapi_program
300763	ubertuple
300764	ctconcrpcif
300765	mfuadmin
300766	aiols
300767	dsmrootd
300768	htdl
300769	caba
300770	vrc_cosimir
300771	cmhelmd
300772	polynsm
300773	[inconnu]
300774	[inconnu]
300775	[inconnu]
300776	[inconnu]
300777	[inconnu]
300778	[inconnu]
300779	[inconnu]
300780	[inconnu]
300781	dsmrecalld
300782	[inconnu]
300783	[inconnu]
300784	twrgcontrol
300785	twrled
300786	twrcfgdb
BMC software	300787-300886
non alloués	300887 - 300999
Sun Microsystems	301000-302000 [2000 numéros]
non alloués	302001-349999
American Airlines	350000 - 350999
Acucobol Inc.	351000 - 351099
The Bristol Group	351100 - 351249
Amteva Technologies	351250 - 351349
351350	wfmMgmtApp

	351351	wfmMgmtDataSrv
	351352	wfmMgmtFut1
	351353	wfmMgmtFut1
	351354	wfmAPM
	351355	wfmIAMgr
	351356	wfmECMgr
	351357	wfmLookOut
	351358	wfmAgentFut1
	351359	wfmAgentFut2
non alloués	351360 - 351406	
Sterling Software ITD	351407	csed
	351360	sched10d
	351361	sched11d
	351362	sched12d
	351363	sched13d
	351364	sched14d
	351365	sched15d
	351366	sched16d
	351367	sched17d
	351368	sched18d
	351369	sched19d
	351370	sched20d
	351371	sched21d
	351372	sched22d
	351373	sched23d
	351374	sched24d
	351375	sched25d
	351376	sched26d
	351377	sched27d
	351378	sched28d
	351379	sched29d
	351380	sched30d
	351381	sched31d
	351382	sched32d
	351383	sched33d
	351384	sched34d
	351385	sched35d
	351386	sched36d
	351387	sched37d
	351388	sched38d
	351389	sched39d
	351390	consoleserver
	351391	scheduleserver
	351392	RDELIVER
	351393	REVENTPROG
	351394	RSENDEVENTPROG
	351395	snapp
	351396	snapad
	351397	sdsoodb
	351398	sdsmain
	351399	sdssrv
	351400	sdsclnt
	351401	sdsreg
	351402	fsbatch
	351403	fsmonitor
	351404	fsdisp
	351405	fssession
	351406	fslog
	351407	svdpappserv
	351408	gns
	351409	[inconnu]

351410 [inconnu]
351411 [inconnu]
351412 axi
351413 rpcxfr
351414 slm
351415 smbpasswdd
351416 tbdbserv
351417 tbprojserv
351418 genericserver
351419 dynarc_ds
351420 dnscmdr
351421 ipcmdr
351422 faild
351423 failmon
351424 faildebug
351425 [inconnu]
351426 [inconnu]
351427 siemens_srs
351428 bsproxy
351429 ifsrpc
351430 CesPvcSm
351431 FrPvcSm
351432 AtmPvcSm
351433 radius
351434 auditor
351435 sft
351436 voicemail
351437 kis
351438 SOFTSERV_NOTIFY
351439 dynarpc
351440 hc
351441 iopas
351442 iopcs
351443 iopss
351444 spcnfs
351445 spcvss
351446 matilda_sms
351447 matilda_brs
351448 matilda_dbs
351449 matilda_sps
351450 matilda_svs
351451 matilda_sds
351452 matilda_vvs
351453 matilda_stats
351454 xtrade
351455 mapsvr
351456 hp_graphicsd
351457 berkeley_db berkeley_db_svc
351458 io_serveur
351459 rpc.niod
351460 rpc.kill
351461 hmdisproxy
351462 smdisproxy
351463 avatard
351464 namu
351465 BMCSess
351466 FENS_Sport
351467 EM_CONFIG
351468 EM_CONFIG_RESP
351469 lodge_proof
351470 ARCserveIT-Queue

	351471	ARCserveIT-Device
	351472	ARCserveIT-Discover
	351473	ARCserveIT-Alert
	351474	ARCserveIT-Database
	351475	scand1
	351476	scand2
	351477	scand3
	351478	scand4
	351479	scand5
	351480	dscv
	351481	cb_svc
	351482	[inconnu]
	351483	iprobe
	351484	omniconf
	351485	isan
BG Partners	351486 - 351500	
	351501	mond
	351502	iqlremote
	351503	iqlalarm
non alloués	351504 - 351599	
Orion Multisystems	351600-351855	
non alloués	351856 - 351899	
NSP lab	351900 - 351999	
non alloués	351999 - 352232	
	352233	asautostart
	352234	asmediad1
	352235	asmediad2
	352236	asmediad3
	352237	asmediad4
	352238	asmediad5
	352239	asmediad6
	352240	asmediad7
	352241	asmediad8
	352242	asmediad9
	352243	asmediad10
	352244	asmediad11
	352245	asmediad12
	352246	asmediad13
	352247	asmediad14
	352248	asmediad15
	352249	asmediad16
	352250	waruser
	352251	warlogd
	352252	warsvrmgr
	352253	warvfsysd
	352254	warftpd
	352255	warnfsd
	352256	bofproxyc0
	352257	bofproxys0
	352258	bofproxyc1
	352259	bofproxys1
	352260	bofproxyc2
	352261	bofproxys2
	352262	bofproxyc3
	352263	bofproxys3
	352264	bofproxyc4
	352265	bofproxys4
	352266	bofproxyc5
	352267	bofproxys5
	352268	bofproxyc6
	352269	bofproxys6

	352270	bofproxyc7
	352271	bofproxys7
	352272	bofproxyc8
	352273	bofproxys8
	352274	bofproxyc9
	352275	bofproxys9
	352276	bofproxyca
	352277	bofproxysa
	352278	bofproxycb
	352279	bofproxysb
	352280	bofproxycc
	352281	bofproxysc
	352282	bofproxycd
	352283	bofproxysd
	352284	bofproxyce
	352285	bofproxyse
	352286	bofproxycf
	352287	bofproxysf
	352288	bofproxypo0
	352289	bofproxypo1
	352290	bofproxypo2
	352291	bofproxypo3
	352292	bofproxypo4
non alloués	352293-370000	
	370001	[inconnu]
	370002	[inconnu]
	370003	[inconnu]
	370004	[inconnu]
	370005	[inconnu]
	370006	[inconnu]
	370007	[inconnu]
	370008	[inconnu]
	370009	[inconnu]
	370010	[inconnu]
	370011	[inconnu]
	370012	[inconnu]
	370013	[inconnu]
	370014	[inconnu]
	370015	[inconnu]
	370016	[inconnu]
	370017	[inconnu]
	370018	[inconnu]
	370019	[inconnu]
	370020	[inconnu]
	370021	[inconnu]
	370022	[inconnu]
	370023	[inconnu]
	370024	[inconnu]
	370025	[inconnu]
	370026	[inconnu]
	370027	[inconnu]
non alloués	370028 - 379999	
	380000	opensna
	380001	probenet
	380002	[inconnu]
	380003	license
	380004	na.3com-remote
	380005	na.ntp
	380006	probeutil
	380007	na.vlb
	380008	cds_mhs_agent

	380009 cds_x500_agent
	380010 cds_mailhub_agent
	380011 codex_6500_proxy
	380012 codex_6500_trapd
	380013 na.nm212
	380014 cds_mta_metrics_agent
	380015 [inconnu]
	380016 na.caple
	380017 codexcapletrap
Swiss Re	380018-380028
	380029 ncstat
	380030 ncnfsstat
	380031 ftams
	380032 na.isotp
	380033 na.rfc1006
non alloués	380034 - 389999
Epoch Systems	390000 - 390049
Quickturn Systems	390050 - 390065
Team One Systems	390066 - 390075
General Electric CRD	390076 - 390085
TSIG NFS subcommittee	390086 - 390089
SoftLab ab	390090 - 390099
Legato Network Services	390100 - 390115
	390116 cdsmonitor
	390117 cdslock
	390118 cdslicense
	390119 shm
	390120 rws
	390121 cdc
Data General	390122 - 390141
Perfect Byte	390142 - 390171
JTS Computer Systems	390172 - 390181
Parametric Technology	390182 - 390191
Voxem	390192 - 390199
Effix Systems	390200 - 390299
Motorola	390300 - 390309
Mobile Data Intl.	390310 - 390325
Physikalisches Institut	390326 - 390330
Ergon Informatik AG	390331 - 390340
Analog Devices Inc.	390341 - 390348
Interphase Corporation	390349 - 390358
NeWsware	390359 - 390374
Qualix Group	390375 - 390379
Xerox Imaging Systems	390380 - 390389
Noble Net	390390 - 390399
Legato Network Services	390400 - 390499
Client Server Tech.	390500 - 390511
Atria	390512 - 390517
GE NMR Instruments	390518 - 390525
Harris Corp.	390526 - 390530
Unisys	390531 - 390562
Aggregate Computing	390563 - 390572
Interactive Data	390573 - 390580
OKG AB	390581 - 390589
K2 Software	390591 - 390594
Collier Jackson	390595 - 390599
Remedy Corporation	390600 - 390699
Mentor Graphics	390700 - 390799
AT&T Bell Labs (Lucent)	390800 - 390899
Xerox	390900 - 390999
Silicon Graphics	391000 - 391063

Data General 391064 - 391095
Computer Support Corp. 391096 - 391099
Quorum Software Systems 391100 - 391199
InterLinear Technology 391200 - 391209
Highland Software 391210 - 391229
Boeing Comp. Svcs. 391230 - 391249
IBM Sweden 391250 - 391259
Signature Authority Svc 391260 - 391271
ZUMTOBEL Licht GmbH 391272 - 391283
NOAA/ERL 391284 - 391299
NCR Corp. 391300 - 391399
FTP Software 391400 - 391409
Cadre Technologies 391410 - 391433
Visionware Ltd (UK) 391434 - 391439
IBR-Partner AG 391440 - 391449
CAP Programator AB 391450 - 391459
Reichle+De-Massari AG 391460 - 391474
Swiss Bank Corp (London) 391475 - 391484
Unisys Enterprise Svr 391485 - 391489
Intel - Test Dev. Tech. 391490 - 391499
Ampex 391500 - 391755
391756 naas-spare
391757 naas-admin
391758 isps
391759 isps-admin
391760 mars
391761 mars-admin
391762 attcis_spare0
391763 attcis_spare1
391764 mail-serveur
391765 mail-serveur-spare
391766 attcis_spare2
391767 attcis_spare3
391768 attcis_spare4
391769 attcis_spare5
391770 attcis_spare6
391771 attcis_spare7
Integrated Systems, Inc. 391772 - 391779
Parametric Tech., Inc. 391780 - 391789
Ericsson Telecom AB 391790 - 391799
SLAC 391800 - 391849
391850 qhrdata
391851 qhrbackup
391852 minutedata
391853 prefecture
391854 supc
391855 suadmincrw
391856 suadminotas
391857 sumessage
391858 sublock
391859 sumotd
staffware dev. (uk) 391860 - 391869
Staffware Dev. (UK) 391870 - 391879
391880 namesrvr
391881 disksrvr
391882 tapesrvr
391883 migsrvr
391884 pdmsrvr
391885 pvrsrvr
391886 repacksrvr
391887 [inconnu]

Convex Computer Corp.	391888 - 391951
	391952 lookoutsrv
	391953 lookoutagnt
	391954 lookoutprxy
	391955 lookoutsnmp
	391956 lookoutrmon
	391957 lookoutfut1
	391958 lookoutfut2
windward	391959 - 391967
	391968 sra_legato
	391969 sra_legato_imgsvr
	391970 sra_legato_0
	391971 sra_legato_1
	391972 sra_legato_2
	391973 sra_legato_3
	391974 sra_legato_4
	391975 sra_legato_5
	391976 sra_legato_6
	391977 sra_legato_7
	391978 sra_legato_8
	391979 sra_legato_9
Brooktree Corp.	391980 - 391989
Cadence Design Systems	391990 - 391999
J. Frank & Associates	392000 - 392999
Cooperative Solutions	393000 - 393999
Xerox Corp.	394000 - 395023
	395024 odbc_sqlretriever
3M	395025 - 395091
Digital Zone Intl.	395092 - 395099
Software Professionals	395100 - 395159
Del Mar Solutions	395160 - 395164
	395165 ife-es
	395166 ife-resmgr
	395167 ife-aes
	395168 ife-bite
	395169 ife-loader
	395170 ife-satcom
	395171 ife-seat
	395172 ife-dbmgr
	395173 ife-testmgr
	395174 atrium_serveur
	395175 ase_director
	395176 ase_agent
	395177 ase_hsm
	395178 ase_mgr
	395179 ase_sim
Hewlett-Packard	395180 - 395194
XES, Inc.	395195 - 395199
Unitech Products	395200 - 395249
TransSys	395250 - 395505
Unisys Govt Systems	395506 - 395519
Bellcore	395520 - 395529
IBM	395530 - 395561
AT&T Network Services	395562 - 395571
Data General	395572 - 395577
Swiss Bank Corp	395578 - 395597
Swiss Bank Corp	395598 - 395637
Novell	395638 - 395643
Computer Associates	395644 - 395650
Omneon Video Networks	395651 - 395656
non alloués	395657 - 395908

```
UK Post Office                  395909 - 395924
AEROSPATIALE                    395925 - 395944
Result d.o.o.                   395945 - 395964
DataTools, Inc.                 395965 - 395980
CADIS, Inc.                     395981 - 395990
Cummings Group, Inc.            395991 - 395994
Cadre Technologies              395995 - 395999
American Airlines               396000 - 396999
Ericsson Telecom TM Div         397000 - 398023
IBM                             398024 - 398028
Toshiba OME Works               398029 - 398033
TUSC Computer Systems           398034 - 398289
AT&T                            398290 - 398320
Ontario Hydro                   398321 - 398346
Micrion Corporation             398347 - 398364
non alloués                     398365 - 398591
Pegasystems, Inc.               398592 - 399616
Spectra Securities Soft         399617 - 399850
QualCom                         399851 - 399866
non alloués                     399867 - 399884
Altris Software Ltd.            399885 - 399899
ISO/IEC WG11                    399900 - 399919
Parametric Technology           399920 - 399949
Dolby Laboratories              399950 - 399981
non alloués                     399982 - 399991
Xerox PARC                      399992 - 399999
#
Next Inc.                       200100000 - 200199999
Netwise (RPCtool)               200200000
Concurrent Computer Corp        200200001 - 200200007
AIM Technology                  200300000 - 200399999
TGV                             200400000 - 200499999

# Numéros de nuance d'authentification alloués par Sun

AUTH_NONE       0           /* pas d'authentification, voir la RFC 1831, c'est-à-dire AUTH_NULL */
AUTH_SYS        1           /* style unix (uid+gid) RFC 1831 , c'est-à-dire  AUTH_UNIX */
AUTH_SHORT      2           /* style unix abrégé, RFC 1831 */
AUTH_DH         3           /* style des (horodatage chiffré) c'est-à-dire AUTH_DES, voir la RFC 2695 */
AUTH_KERB       4           /* authentification kerberos, voir la RFC 2695 */
AUTH_RSA        5           /* authentification RSA */
RPCSEC_GSS      6           /* sécurité RPC fondée sur GSS pour authentification, intégrité et confidentialité, RFC 5403 */

AUTH_NW         30001       NETWARE
AUTH_SEC        200000      TSIG NFS subcommittee
AUTH_ESV        200004      SVr4 ES

AUTH_NQNFS      300000      Univ. of Guelph - Not Quite NFS
AUTH_GSSAPI     300001      OpenVision <john.linn@ov.com>
AUTH_ILU_UGEN   300002      Xerox <janssen@parc.xerox.com>          - Identité générique ILU non sécurisée
#
# De petits blocs sont alloués dans la série de numéros 39xxxx
#
AUTH_SPNEGO     390000
        390000 - 390255 "pseudo" nuances NFS pour RPCSEC_GSS
        390003 - authentification kerberos_v5, RFC 2623
        390004 - kerberos_v5 avec intégrité des données, RFC 2623
        390005 - kerberos_v5 avec confidentialité des données, RFC 2623
        200000000       Réservé
        200100000       NeXT Inc.
```

Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2203] M. Eisler, A. Chiu, L. Ling, "Spécification du protocole RPCSEC_GSS", septembre 1997. *(P.S.)*

[RFC4506] M. Eisler, éd., "XDR : norme de représentation des données externes", mai 2006. (STD0067)

Références pour information

[DH] Diffie & Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory IT-22, novembre 1976.

[RFC0768] J. Postel, "Protocole de datagramme d'utilisateur (UDP)", (STD 6), 28 août 1980.

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.

[RFC1094] Sun Microsystems, "NFS : Spécification du protocole de fichiers système réseau (NFS)", mars 1989.

[RFC1813] B. Callaghan, B. Pawloowski et P. Staubach, "Spécification du protocole NFS version 3", juin 1995. *(Information)*

[RFC1831] R. Srinivasan, "RPC : Spécification de la version 2 du protocole d'appel de procédure à distance", août 1995. *(Expérimentale, Obsolète, voir* RFC5531)

[RFC1833] R. Srinivasan, "Protocoles de liaison pour RPC ONC version 2", août 1995. *(P.S.) (MàJ par* RFC5665)

[RFC2623] M. Eisler, "Questions de sécurité de NFS versions 2 et 3 et utilisation de RPCSEC_GSS et Kerberos v5 par le protocole NFS", juin 1999. *(P.S.)*

[RFC2695] A. Chiu, "Mécanismes d'authentification pour RPC ONC", septembre 1999. *(Information)*

[RFC2743] J. Linn, "Interface générique de programme d'application de service de sécurité, version 2, mise à jour 1", janvier 2000. (*MàJ par* RFC5554)

[RFC3530] S. Shepler et autres, "Protocole de système de fichiers réseau (NFS) version 4", avril 2003. (*Obsolète, voir* RFC7530) *(P.S.)*

[VMTP] Cheriton, D., "VMTP: Versatile Message Transaction Protocol", Preliminary Version 0.3, Stanford University, janvier 1987.

[XRPC] Birrell, A. D. & B. J. Nelson, "Mise en œuvre des appels de procédure distants", XEROX CSL-83-7, octobre 1983.

Adresse de l'auteur

Robert Thurlow
Sun Microsystems, Inc.
500 Eldorado Boulevard, UBRM05-171
Broomfield, CO 80021
USA
téléphone : 877-718-3419
mél : robert.thurlow@sun.com