

Groupe de travail Réseau
Request for Comments : 2653
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

J. Allen, WebTV Networks, Inc.
P. Leach, Microsoft
R. Hedberg, Catalogix
août 1999

Protocoles de transport pour CIP

Statut du présent mémoire

Le présent document spécifie un protocole en cours de 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.

Notice de Copyright

Copyright (C) The Internet Society (1999). Tous droits réservés.

Résumé

Le présent document spécifie trois protocoles pour transporter les demandes CIP, les réponses et les objets d'indices, en utilisant TCP, la messagerie, et HTTP. Les objets eux-mêmes sont définis dans la [RFC2652] ; l'architecture CIP globale est définie dans la [RFC2651].

1. Protocole

Dans cette section est présenté le protocole réel de transmission des objets d'indice CIP et de maintenance du maillage. Alors que les documents d'accompagnement ([RFC2651] et [RFC2652]) décrivent les concepts impliqués et les formats des objets MIME CIP, le présent document est la définition autorisée des formats de message et des mécanismes de transfert de CIP utilisés sur TCP, HTTP et la messagerie.

1.1 Philosophie

La philosophie de la conception du protocole CIP est un concept de bloc de construction. Au lieu de s'appuyer sur des outils de définition de protocole encombrants, ou des codages de texte ad hoc, CIP s'appuie sur des technologies existantes dans l'Internet, et bien comprises, comme MIME, la RFC-822, Whois++, FTP, et SMTP. On espère que cela va servir à faciliter la mise en œuvre et la construction du consensus. Cela devrait aussi servir d'exemple d'une façon simple de s'appuyer sur les technologies existantes dans l'Internet pour mettre facilement en œuvre de nouveaux services au niveau application.

1.2 Conventions

Les mots-clés "DOIT" et "PEUT" dans le présent document sont à interpréter comme décrit dans "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence" [RFC2119].

La syntaxe formelle est définie en utilisant l'ABNF [RFC2234].

Dans les exemples, les octets qui sont envoyés par l'envoyeur CIP sont précédés par ">>> " et ceux envoyés par le receveur CIP par "<<<< ".

2. Mécanismes d'échange de message MIME

CIP s'appuie sur l'échange de messages MIME standard pour toutes les demandes et réponses. Ces messages sont passés sur un système de transport bidirectionnel fiable. Le présent document définit le transport sur des flux réseau fiables (via TCP), via HTTP, et via l'infrastructure de messagerie Internet.

Le serveur CIP qui initie la connexion (auquel on se réfère conventionnellement comme à un client) sera appelé ci-dessous l'envoyeur CIP. Le serveur CIP qui accepte la connexion entrante d'un envoyeur CIP et répond aux demandes de l'envoyeur CIP est appelé un receveur CIP.

2.1 Transport des flux

Les messages CIP sont transmis sur des connexions TCP bidirectionnelles via un simple protocole de texte. La transaction peut avoir lieu sur tout accès TCP, comme spécifié par la configuration du maillage. Il n'y a pas "d'accès bien connu" pour les transactions CIP. Toutes les informations de configuration dans le système doivent inclure un nom d'hôte et un accès.

Toutes les actions de l'envoyeur CIP (y compris les demandes, l'initiation de connexion, et la finalisation de connexion) sont acquittées par le receveur CIP avec un code de réponse. Voir au paragraphe 2.1.1 le format de ces codes, une liste des réponses que peut générer un serveur CIP, et l'action attendue de l'envoyeur CIP pour chacune d'elles.

Afin d'assurer la rétrocompatibilité avec les serveurs Whois++ existants, les envoyeurs CIP en CIPv3 DOIVENT d'abord vérifier que le protocole le plus récent est pris en charge. Ils font cela en envoyant la commande système Whois++ illégale suivante : "# CIP-Version: 3<cr><lf>". Sur les serveurs Whois++ existants qui mettent en œuvre les versions 1 et 2 de CIP, il en résulte un code de réponse de la série 500, et le serveur termine la connexion. Si le serveur met en œuvre CIPv3, il DOIT plutôt répondre par un code de réponse 300. Les futures versions de CIP peuvent être correctement négociées en utilisant cette technique avec une chaîne différente (c'est-à-dire, "CIP-Version: 4"). Un exemple de ce bref échange est donné ci-dessous.

Note : Si un envoyeur CIP peut supposer en toute sécurité que le serveur met en œuvre CIPv3, il peut choisir d'envoyer la chaîne "# CIP-Version: 3" et la faire immédiatement suivre par la demande CIPv3. Cette optimisation, qui n'est utile que dans les maillages CIPv3 homogènes connus, évite d'attendre pendant le délai d'aller-retour inhérent à la négociation.

Une fois qu'un envoyeur CIP a réussi à vérifier que le serveur prend en charge les demandes CIPv3, il peut envoyer la demande, formatée comme un message MIME avec les en-têtes Mime-Version et Content-Type (seulement) en utilisant la terminaison de ligne standard du réseau : "<cr><lf>".

Cip-Req	= Req-Hdrs CRLF Req-Body
Req-Hdrs	= *(Version-Hdr Req-Cntnt-Hdr)
Req-Body	= Body ; format de corps de demande comme dans la [RFC2652]
Body	= Data CRLF "." CRLF
Data	= ; données avec CRLF "." CRLF remplacé par CRLF ".." CRLF
Version-Hdr	= "Mime-Version:" "1.0" CRLF
Req-Cntnt-Hdr	= "Content-Type:" Req-Content CRLF
Req-Content	= ; le format est spécifié dans la [RFC2652]
Cip-Rsp	= Rsp-Code CRLF [Rsp-Hdrs CRLF Rsp-Body] [Indx-Cntnt-Hdr CRLF Index-Body]
Rsp-Code	= DIGIT DIGIT DIGIT Comment
Comment	= ; tous caractères excepté CR et LF
Rsp-Hdrs	= *(Version-Hdr Rsp-Cntnt-Hdr)
Rsp-Cntnt-Hdr	= "Content-Type:" Rsp-Content CRLF
Rsp-Content	= ; le format est spécifié dans la [RFC2652]
Rsp-Body	= Body ; format du corps de réponse comme dans la [RFC2652]
Indx-Cntnt-Hdr	= "Content-Type:" Indx-Obj-Type CRLF
Indx-Obj-Type	= ; tout objet d'indice de type MIME enregistré ; format spécifié dans [RFC2045]
Index-Body	= Body ; format défini dans chaque spécification d'indice
CRLF	= CR LF ; nouvelle ligne Internet standard
CR	= %x0D ; retour chariot
LF	= %x0A ; saut à la ligne
DIGIT	= %x30-39 ; chiffres de 0 à 9

Le message est terminé en utilisant la terminaison de message de style SMTP. Les données sont envoyées octet par octet, excepté lorsque le schéma <cr><lf>1*["."]<cr><lf> est rencontré, auquel cas un point supplémentaire est ajouté.

Lorsque les données sont finies, le schéma d'octet "<cr><lf>.<cr><lf>" est transmis au receveur CIP.

Du côté du receveur CIP, la transformation inverse est appliquée, et la lecture du message consiste en tous les octets jusqu'à, non inclus, le schéma de terminaison.

En réponse à la demande, le receveur CIP envoie un code de réponse, des séries 200, 400, ou 500. Le receveur CIP traite alors la demande et la réponse, si nécessaire, avec un message MIME. Cette réponse est aussi délimitée par une terminaison de message de style SMTP.

Après avoir répondu avec un code de réponse, le receveur CIP DOIT se préparer à lire un autre message de demande, remettant l'état au point où il était lorsque l'envoyeur CIP avait vérifié la version CIP. Si l'envoyeur CIP a terminé de faire des demandes, il peut clore la connexion. En réponse, le receveur CIP DOIT interrompre la lecture de message et se préparer à une nouvelle connexion d'envoyeur CIP (rétablissant complètement son état).

On donne un exemple ci-dessous. Il est utile de répéter encore une fois que le format de commande est défini dans la [RFC2652] tandis que le corps de message est défini dans chaque définition d'objet d'indice. Dans cet exemple, la définition d'objet d'indice de la [RFC2654] sera utilisée. Les terminaisons de ligne sont explicitement montrées entre des crochets angulaires ; les sauts à la ligne dans ce texte ne sont faits que pour faciliter la lecture. Les commentaires sont inclus entre des accolades et décalés sur la droite.

```

                                     { L'envoyeur CIP se connecte au receveur CIP }
<<< % 220 Serveur CIP Exemple prêt<cr><lf>
>>> # CIP-Version: 3<cr><lf>
<<< % 300 CIPv3 OK!<cr><lf>
>>> Mime-Version: 1.0<cr><lf>
>>> Content-type: application/index.cmd.datachanged; type=
>>> x-tagged-index-1; dsi=1.2.752.17.5.10<cr><lf>
>>> <cr><lf>
>>> updatetype: incremental tagbased<cr><lf>
>>> thisupdate: 855938804<cr><lf>
>>> lastupdate: 855940000<cr><lf>
>>> .<cr><lf>
<<< % 200 Bon message MIME reçu
>>> MIME-Version: 1.0<cr><lf>
>>> Content-Type: application/index.obj.tagged;
>>> dsi=1.2.752.17.5.10;
>>> base-uri="ldap://ldap.umu.se/dc=umu,dc=se"<cr><lf>
>>> <cr><lf>
>>> version: x-tagged-index-1<cr><lf>
>>> updatetype: incremental<cr><lf>
>>> lastupdate: 855940000<cr><lf>
>>> thisupdate: 855938804<cr><lf>
>>> BEGIN IO-schema<cr><lf>
>>> cn: TOKEN<cr><lf>
>>> sn: FULL<cr><lf>
>>> title: FULL<cr><lf>
>>> END IO-Schema<cr><lf>
>>> BEGIN Bloc de mise à jour<cr><lf>
>>> BEGIN Old<cr><lf>
>>> title: 3/testpilot<cr><lf>
>>> END Old<cr><lf>
>>> BEGIN New<cr><lf>
>>> title: 3/chiefpilot<cr><lf>
>>> END New<cr><lf>
>>> END Bloc de mise à jour<cr><lf>
>>> .<cr><lf>
<<< % 200 Bon message MIME reçu
                                     { L'envoyeur CIP ferme la prise d'écriture }
<<< % 222 Clôture de la connexion en réponse à la fermeture de l'envoyeur CIP
                                     { Le receveur CIP clôt son côté, rétablit l'état et attend un nouvel envoyeur CIP }

```

Un exemple d'un échec de négociation de version ressemble à ceci :

```

                                     { L'envoyeur CIP se connecte au receveur CIP }
<<< % 220 Serveur Whois++ prêt<cr><lf>
>>> # CIP-Version: 3<cr><lf>
<<< % 500 Erreur de syntaxe<cr><lf>
                                     { Le serveur clôt la connexion }

```

L'envoyeur CIP peut tenter de ressayer d'utiliser la version 1 ou 2 du protocole. L'envoyeur CIP peut mettre en antémémoire les résultats de cette négociation infructueuse pour éviter des tentatives ultérieures.

2.1.1 Codes de réponse spécifiques du transport

Les codes de réponse suivants sont utilisés avec le transport de flux :

Code	Texte de description suggéré	Action de l'envoyeur CIP
200	Demande MIME reçue et traitée	N'attend pas de résultat, continue la session (ou la clôt)
201	Demande MIME reçue et traitée, résultat suit	Lire une réponse, délimitée dans le style de message SMTP.
220	Message bannière du serveur initial	Continuer avec une interaction Whois++, ou tenter la négociation de version CIP.
222	Fermeture de connexion (en réponse à la clôture de l'envoyeur CIP)	Transaction terminée.
300	Version CIP demandée acceptée	Continuation de la transaction CIP, dans la version spécifiée.
400	Temporairement incapable de traiter la demande	Réessayer plus tard. Peut être utilisé pour indiquer que le serveur n'a pas actuellement les ressources disponibles pour accepter un indice.
500	Mauvais format de message MIME	Réessayer avec un message MIME correctement formaté.
501	Demande inconnue ou manquante dans la commande application/indice	Réessayer avec une commande CIP correcte.
502	Il manque des attributs CIP exigés dans la demande	Réessayer avec les attributs CIP corrects.
520	Interruption de la connexion pour une raison inattendue	Alerter l'administrateur local.
530	La demande exige une signature valide	Signer la demande, si possible, et réessayer. Autrement, faire rapport du problème à l'administrateur.
531	La demande a une signature invalide	Faire rapport du problème à l'administrateur.
532	La signature ne peut pas être vérifiée	Alerter l'administrateur local, qui devrait coopérer avec l'administrateur distant pour diagnostiquer et résoudre le problème. (Il manque probablement une clé publique.)

2.2 Infrastructure de messagerie Internet comme transport

Comme solution de remplacement aux flux TCP, des transactions CIP peuvent avoir lieu sur l'infrastructure existante de messagerie Internet. Il y a deux raisons à cette disposition de CIP. D'abord, elle diminue les barrières à l'entrée des serveurs d'extrémité. Lorsque le besoin d'une mise en œuvre complète de TCP est faible, les nœuds d'extrémité (qui par définition n'envoient que des objets d'indice) peuvent consister en seulement une base de données et un programme d'indexation (éventuellement écrit dans un langage de très haut niveau) pour participer au maillage.

Ensuite, elle reste en ligne avec la philosophie d'utilisation de la technologie existante de l'Internet. Les messages MIME utilisés pour les demandes et les réponses sont, par définition de la spécification MIME, convenables pour le transport via l'infrastructure de la messagerie Internet. Avec quelques règles simples, on ouvre une voie entièrement différente pour interagir avec les serveurs CIP qui choisissent de mettre en œuvre ce transport. Voir sous la section conformité du protocole les détails des options que les mises en œuvre de serveur ont pour prendre en charge les divers transports.

Le rythme de base des demandes/réponses est conservé lorsque on utilise le transport par messagerie. Les paragraphes qui suivent précisent quelques cas particuliers qui doivent être considérés pour le transport des objets CIP par messagerie. En général, toutes les spécifications de protocoles et de formats de messagerie (en particulier les multiparties de sécurité MIME) peuvent être utilisées avec le transport de CIP par messagerie.

2.2.1 Négociation de la version CIP

Comme aucune information sur la version CIP utilisée n'est présente dans le message MIME, cette information doit être portée dans l'en-tête du message. Donc les demandes CIP envoyées en utilisant le transport par messagerie DOIVENT inclure une ligne d'en-tête CIP-version, pour être enregistrée conformément à la [RFC3864]. Le format de cette ligne est :

```
DIGIT      = %x30-39
number     = 1*DIGIT
cipversion = "CIP-Version:" <sp> number["." number]
```

2.2.2 Chemin de retour

Lorsque des transactions CIP ont lieu sur un flux bidirectionnel, le chemin de retour pour les erreurs et les résultats est implicite. Utiliser la messagerie comme transport introduit des difficultés pour le receveur, parce qu'il n'est pas toujours

clair d'après les en-têtes où devrait exactement aller la réponse, bien qu'en pratique il y ait des heuristiques utilisées par les agents d'utilisateur de messagerie (MUA, *Mail User Agent*).

CIP résout ce problème par fiat. Les demandes CIP envoyées en utilisant le transport de messagerie DOIVENT inclure un en-tête Reply-To comme spécifié par la RFC-822. Tout message à traiter reçu par un serveur CIP qui met en œuvre le transport de messagerie sans en-tête Reply-To DOIT être ignoré, et un message devrait être enregistré pour l'administrateur local. Le receveur NE DOIT PAS tenter de répondre par une erreur à une adresse déduite du message entrant.

Si il n'y a aucune circonstance dans laquelle une réponse serait à renvoyer à une demande CIP, l'envoyeur devrait inclure un en-tête Reply-To contenant l'adresse "<>". Les receveurs NE DOIVENT jamais tenter d'envoyer des réponses à cette adresse, car elle est définie comme invalide (à la fois ici et dans la grammaire BNF de la RFC-822). On devrait noter qu'en général, c'est une mauvaise idée de supprimer le rapport d'erreur de cette façon. Cependant, dans le cas le plus simple d'un programme qui pousse les indices, cela PEUT être une simplification souhaitable.

2.3 Transport HTTP

HTTP PEUT aussi être utilisé pour transporter les objets CIP, car ce sont juste des objets MIME. Une transaction est effectuée en utilisant la méthode POST pour envoyer une commande application/index.cmd et retourner une application/index.response ou un application/index.obj dans la réponse HTTP. L'URL qui est la cible de l'envoi est un paramètre de configuration de la relation envoyeur CIP à receveur CIP.

Exemple :

```

                                     { Le client ouvre la connexion et envoie un POST }
>>> POST / HTTP/1.1<cr><lf>
>>> Host: cip.some.corp<cr><lf>
>>> Content-type: application/index.cmd.noop<cr><lf>
>>> Date: Thu, 6 Jun 1997 18:16:03 GMT<cr><lf>
>>> Content-Length: 2<cr><lf>
>>> Connection: close<cr><lf>
>>> <cr><lf>
                                     { Le serveur traite la demande }
<<< HTTP/1.1 204 No Content<cr><lf>
                                     { Le serveur ferme la connexion }
```

En plus d'avoir un effet de levier sur les capacités de sécurité qui découlent de HTTP, il y a d'autres caractéristiques de HTTP qui PEUVENT être utiles dans un contexte CIP. Un client CIP PEUT utiliser les en-têtes Accept-Charset et Accept-Language de HTTP pour exprimer le désir de restituer un indice dans un jeu de caractères ou un langage naturel particulier. Il PEUT utiliser l'en-tête Accept-Encoding pour (par exemple) indiquer qu'il peut traiter les réponses compressées, que le serveur CIP PEUT envoyer en conjonction avec l'en-tête Transfert-Encoding. Il PEUT utiliser l'en-tête If-Modified-Since pour empêcher la transmission inutile d'un indice qui n'a pas changé depuis la dernière interrogation. Un serveur CIP peut utiliser l'en-tête Retry-After pour demander que le client réessaie plus tard lorsque le serveur sera moins chargé.

3. Considérations pour la sécurité

Les informations d'indices peuvent être protégées à deux niveaux : le premier est par l'utilisation de la technologie disponible pour sécuriser les objets MIME [RFC1847], et le second en utilisant la technologie disponible pour sécuriser le transport.

Lorsque il en vient au transport, le transport du flux peut être protégé par l'utilisation de TLS [RFC2246]. Pour HTTP, la sécurité est traitée en utilisant l'authentification de base HTTP [RFC2616], l'authentification HTTP par résumé de message [RFC2617] ou SSL/TLS. Une protection supplémentaire pour l'échange SMTP peut être réalisée en utilisant SMTP sécurisé sur TLS [RFC2487].

4. Références

[RFC1847] J. Galvin, S. Murphy, S. Crocker, N. Freed, "[Sécurité multiparties pour MIME](#) : multipartie/signée et multipartie/chiffée", octobre 1995. (P.S.)

- [RFC2045] N. Freed et N. Borenstein, "[Extensions de messagerie Internet multi-objets \(MIME\) Partie 1 : Format des corps de message Internet](#)", novembre 1996. (*D. S., MàJ par 2184, 2231, 5335.*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence](#)", BCP 14, mars 1997.
- [RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir RFC5234*)
- [RFC2246] T. Dierks et C. Allen, "[Protocole TLS version 1.0](#)", janvier 1999.
- [RFC2487] P. Hoffman, "Extension de service SMTP pour SMTP sécurisé sur TLS", janvier 1999. (*Obsolète, voir RFC3207*) (P.S.)
- [RFC2616] R. Fielding et autres, "[Protocole de transfert hypertexte -- HTTP/1.1](#)", juin 1999. (*D.S., MàJ par 2817, 6585*)
- [RFC2617] J. Franks et autres, "Authentification HTTP : [Authentification d'accès de base et par résumé](#)", juin 1999. *DS.*)
- [RFC2651] J. Allen, M. Mealling, "Architecture du [protocole d'indexation commune](#) (CIP)", août 1999. (P.S.)
- [RFC2652] J. Allen, M. Mealling, "[Définitions d'objet MIME](#) pour le protocole d'indexation commune (CIP)", août 1999. (P.S.)
- [RFC2654] R. Hedberg et autres, "Objet d'index étiqueté à utiliser dans le protocole d'indexation commune", août 1999. (*Exp.*)
- [RFC3864] G. Klyne, M. Nottingham, J. Mogul, "Procédures d'[enregistrement pour les champs d'en-tête de message](#)", septembre 2004. ([BCP0090](#))

5. Adresse des auteurs

Jeff R. Allen
246 Hawthorne St.
Palo Alto, CA 94301

mél : jeff.allen@acm.org

Paul J. Leach
Microsoft
1 Microsoft Way
Redmond, WA 98052

mél : paulle@microsoft.com

Roland Hedberg
Catalogix
Dalsveien 53
0775 Oslo

Norway
mél : roland@catalogix.ac.se

6. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1999). Tous droits réservés.

Ce document et les traductions de celui-ci peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de droits de reproduction ci-dessus et ce paragraphe soient inclus sur toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant le droit d'auteur ou les références à l'Internet Society ou d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définis dans les procédures des normes d' l'Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet ou ses successeurs ou ayants droit.

Ce document et les renseignements qu'il contient sont fournis "TELS QUELS" et l'INTERNET SOCIETY et l'INTERNET ENGINEERING TASK FORCE déclinent toute garantie, expresse ou implicite, y compris mais sans s'y limiter, toute garantie que l'utilisation de l'information ici présente n'enfreindra aucun droit ou aucune garantie implicite de commercialisation ou d'adaptation à un objet particulier.

Remerciement

Le financement de la fonction d'éditeur des RFC est actuellement fourni par la Internet Society.