

UFR d'INFORMATIQUE

DESS

Applications des Réseaux et de la Télématique

Aurélien Amacker

La multidiffusion (multicast) en IPv6 dans le cadre du téléenseignement interactif

Mémoire soutenu en septembre 2003

Directeur des études : Gilbert Sol
Directeur du mémoire : Yves Legrandgérard
Tuteur en entreprise : Jacques Prévost

*Deux exemplaires de ce mémoire sont déposés et librement consultables au secrétariat du DESS
« Applications des Réseaux et de la Télématique »
Toute reproduction en est interdite sans accord écrit de l'auteur et du directeur des études*

Résumé

Le projet ATHENA, initié par l'association Aristote, avait à l'origine pour but de favoriser les échanges de contenus académiques avec l'Afrique francophone en utilisant les réseaux de la recherche et s'est par la suite étendu au reste du monde avec des partenariats en Inde et au Mexique.

Par ailleurs l'association Aristote est impliquée dans la mise en œuvre et la retransmission en multicast d'évènements à caractère pédagogique tels que les Cours DIM, les Causeries de Renater et les séminaires X-Aristote.

L'emploi des logiciels de visioconférence en multicast dans le cadre de ces retransmissions a permis de révéler des besoins quant aux fonctionnalités offertes par ces outils.

J'ai donc procédé à des modifications du logiciel utilisé pour émettre et recevoir de la vidéo en multicast afin de lui permettre de transmettre les transparents de l'orateur lors d'un cours ou d'une présentation, mais aussi pour lui permettre d'émettre une même source vidéo à deux débits différents, ceci afin de proposer aux récepteurs un flux vidéo en adéquation avec leur bande passante.

D'autre part, le GIP Renater, partenaire de l'association Aristote, participe au processus de transition du protocole IPv4 vers IPv6 en proposant à la communauté académique un service de passerelle IPv4 – IPv6 multicast et c'est dans ce contexte que j'ai été chargé de la mise en œuvre et du fonctionnement de ce service.

En effet l'association Aristote est concernée par le développement du protocole IPv6 qui, outre les nombreuses fonctionnalités innovantes qu'il apporte, permettra à de nombreux pays oubliés par les nouvelles technologies de l'Information et de la Communication de bénéficier d'un nombre d'adresses IP nécessaire au déploiement de leurs réseaux et ainsi d'accéder aux ressources pédagogiques contenues sur le réseau Internet, c'est pourquoi l'association Aristote encourage ses partenaires du projet ATHENA à mettre en œuvre des réseaux IPv6 et IPv6 multicast.

Remerciements

Je tiens tout d'abord à remercier M. Jacques Prévost de m'avoir choisi parmi les différents candidats au poste d'apprenti au GIP Renater et de m'avoir accordé sa confiance.

Je tiens aussi à remercier M. Bernard Tuy pour ses précieux conseils ainsi que tout le personnel du GIP Renater avec en particulier les personnes suivantes : Cecile Marlet, Laurence Freyt-Caffin, Jérôme Durand, Pierre-Emmanuel Goiffon, Ahmed Sahnoun, Simon Muyal, Gilles Yonnet et Emmanuel Da Costa.

Je tiens à remercier également le directeur du GIP Renater, M. Dany Vandromme, d'avoir donné son autorisation à ma venue.

Je tiens à remercier tout particulièrement M. Gilbert Sol, pour ses enseignements mais aussi pour la disponibilité dont il a fait preuve tout au long de l'année pour répondre à nos questions.

Sur le plan personnel je tiens à remercier ma mère qui m'a toujours soutenu et Audrey qui a fait preuve de beaucoup de patience afin de m'accompagner tout au long de cette année.

Sommaire

I.	Introduction.....	6
II.	Présentation du contexte	8
A.	L'association Aristote	8
B.	Renater.....	9
1.	Le Réseau National des Télécommunications pour la Technologie, l'Enseignement et la Recherche	9
2.	Le GIP Renater	13
C.	LE GN6.....	14
D.	Des actions de téléenseignement	15
1.	Le projet ATHENA	15
	Actions pour des Transmissions Harmonieuses et des Echanges de Natures Académiques	15
2.	Les cours DIM	22
3.	Les séminaires X-Aristote	25
4.	Les Causeries de Renater	26
5.	La Grille de Conférences.....	26
E.	Objectifs de mon stage	28
III.	Les technologies employées	29
A.	IPv6.....	29
1.	Un plus grand nombre d'adresses IP	29
2.	Une meilleure hiérarchisation des adresses.....	30
B.	La multidiffusion (multicast) en IPv6	31
1.	Protocoles de routage au niveau local.....	32
2.	Protocoles de routage inter-liens.....	33
C.	Le M6Bone	35
1.	Présentation.....	35
2.	Topologie du M6Bone.....	35
D.	L'interface de programmation d'applications en multicast IPv6 en C/C++ sous Unix/Linux.....	38
1.	L'interface de programmation d'applications en IPv6.....	38
2.	L'interface de programmation d'applications en multicast IPv6.....	38
E.	Les protocoles de transport de trafic temps réel, RTP et RTCP	42
1.	RTP.....	42
2.	RTCP.....	44
F.	La vidéo numérique.....	46
1.	Codage de la couleur	46
2.	Les formats d'images vidéo	47
3.	Les formats de compression de signaux vidéo	48
G.	Les outils du M6Bone	49
1.	Vic (Video Conference).....	49
2.	Rat (<i>Robust Audio Tool</i> , outil audio robuste).....	55
IV.	Mises en oeuvre.....	57
A.	Poursuite du projet ATHENA.....	57
1.	Support technique auprès des partenaires du projet ATHENA	57

2.	Maintien du contact avec les partenaires du projet ATHENA	57
B.	Mise en oeuvre de la passerelle IPv4 multicast – IPv6 multicast.....	59
C.	Développement des fonctionnalités des outils de visioconférence pour le téléenseignement interactif	61
1.	Contexte	61
2.	Vic haute résolution (4CIF).....	61
3.	Vic à deux vitesses.....	72
D.	Transfert de connaissances	77
V.	Conclusion.....	78
VI.	Annexes.....	79
A.	Bibliographie – Références.....	79
B.	Références sur Internet	80
C.	Glossaire.....	81

I. Introduction

L'association Aristote, partenaire du GIP Renater, est impliquée dans la mise en œuvre et la retransmission en multicast d'un certain nombre d'évènements à caractère pédagogique comme par exemple les séminaires X-Aristote diffusés depuis l'école Polytechnique.

L'expérience acquise lors de ces retransmissions en multicast a conduit l'association Aristote à contribuer à l'amélioration de certaines des techniques utilisées pour la retransmission et l'enregistrement de ces séminaires et conférences ; amélioration de Vic, utilisation de IPv6, utilisation de SMIL, etc.

Ces techniques s'intègrent dans le projet Grille de Conférence, dont Aristote et Renater sont partenaires. Elles sont notamment utilisées par DIM (cours en visioconférence entre plusieurs DESS) et par les Causeries de Renater, des séminaires virtuels publics diffusés régulièrement sur Renater.

Le projet ATHENA, initié par l'association Aristote, a pour but de promouvoir les échanges académiques avec les pays de l'Afrique Francophone en utilisant les réseaux de la recherche.

La version actuelle du protocole IP utilisée sur Internet, IPv4, pose beaucoup de problèmes en ce qui concerne le nombre d'adresses disponibles et la taille des tables de routage, c'est pourquoi il y a dix ans a commencé le processus de normalisation de la prochaine norme du protocole IP, IPv6.

Cette norme devrait cohabiter plusieurs années avec le protocole IPv4 avant de le remplacer totalement, néanmoins Renater, le Réseau National des Télécommunications pour la Technologie, l'Enseignement et la Recherche, propose déjà une connectivité IPv6 en natif depuis la troisième version du réseau Renater inaugurée en octobre 2002.

L'association Aristote encourage ses partenaires à mettre en œuvre des réseaux IPv6 et IPv6 multicast afin de leur permettre de disposer d'un nombre d'adresses IP suffisant pour déployer leurs réseaux mais aussi afin de leur éviter d'avoir à effectuer plus tard la transition vers IPv6.

Le multicast est une technologie spécifiquement dédiée à la transmission de données multimédia sur le réseau Internet, elle est aussi tout particulièrement adaptée pour des applications de vidéoconférence.

Dans le cadre du projet européen 6Net, un réseau multicast IPv6 expérimental a été mis en place qui s'étend à l'ouest jusqu'au Mexique et à l'est en Corée du sud, il s'agit du M6Bone et le cœur de ce réseau est hébergé au GIP Renater. L'objet du déploiement de ce réseau est de proposer un service avancé du protocole IPv6 afin de participer à la promotion et à une plus large diffusion de ce protocole.

Cependant, en plus de la mise en œuvre de réseaux utilisant des technologies innovantes il est aussi nécessaire de développer les applications permettant d'utiliser pleinement toutes les fonctionnalités offertes par ces réseaux.

Mon travail a donc consisté d'une part à assurer un support technique auprès des partenaires du projet ATHENA en les aidant à mettre en œuvre des réseaux IPv6 multicast, et d'autre part à ajouter des fonctionnalités au logiciel de vidéoconférence Vic utilisé pour

émettre et recevoir de la vidéo en multicast afin d'améliorer le confort et l'interactivité des actions de téléenseignement dans lesquelles est impliquée l'association Aristote.

II. Présentation du contexte

A. L'association Aristote

Aristote est une association régie par la loi 1901. Elle a été créée en 1984 par l'INRIA¹, le CEA², EDF³ et le CNES⁴ puis formalisée en 1988.

L'association Aristote regroupe de grands organismes ou entreprises français intéressés comme acteurs ou utilisateurs à l'évolution des télécommunications.

L'objectif d'Aristote se situe dans le domaine des techniques, moyens, outils et services de communication informatique, notamment :

- mettre en commun des efforts de prospective, d'étude et d'information que – font ses partenaires, et, s'il y a lieu – promouvoir l'élaboration et la mise en service de nouveaux produits, systèmes et services d'intérêt général au bénéfice de ses partenaires.

Cette activité se déroule dans le cadre des groupes de travail techniques d'Aristote.

- organiser ou encourager des actions avancées d'information ou de formation : séminaires d'intérêt général, séminaires de formation technique, journées d'études thématiques.

Les membres sont soit des entités de recherche ou d'enseignement publiques ou privées, soit des entités industrielles ou commerciales.

Ils sont répartis en trois catégories : titulaires, associés, correspondants. Ces trois catégories bénéficient des mêmes possibilités de travail dans les groupes techniques et des mêmes tarifs dans les séminaires et formations.

Voici les différents groupes de travail que présente Aristote : ATHENA⁵, Calcul Scientifique Distribué, C-SMIL (le club SMIL d'Aristote), C-UIML (le club UIML d'Aristote), GIHM, Multimédia, PIN, Wapiti.

¹ Institut National de la Recherche en Informatique et Automatique.

² Commissariat à l'Energie Atomique.

³ Electricité de France.

⁴ Centre National d'Etudes Spatiales.

⁵ Actions pour des Transmissions Harmonieuses et des Echanges de Natures Académiques.

B. Renater

1. Le Réseau National des Télécommunications pour la Technologie, l'Enseignement et la Recherche

a) Le réseau Renater et sa communauté d'utilisateurs

Les utilisateurs :

Aujourd'hui plus de 600 sites ayant une activité dans les domaines de la Recherche, la Technologie, l'Enseignement et la Culture sont raccordés au réseau Renater.

Ce réseau leur permet de communiquer entre eux, d'accéder aux centres de recherche publics et privés, aux établissements d'enseignement du monde entier et à l'internet.

Le réseau Renater :

Le réseau Renater est composé d'une infrastructure métropolitaine et de liaisons internationales à haut débit. Des points de présence de Renater sont également implantés dans les départements d'Outre Mer.

La partie métropolitaine du réseau :

Renater est basé sur une architecture distribuée : il comprend une épine dorsale nationale à haut débit multi-Gbit/s – Renater 3 – qui fédère des réseaux de collecte régionaux développés avec le soutien des collectivités territoriales dans le cadre de l'aménagement du territoire.

Les points de présence outre-mer :

Renater dispose également de points de présence dans les domaines et territoires d'outre-mer.

- Les liaisons vers les autres pays et vers l'Internet :

RENATER est interconnecté à 2.5 Gbit/s aux autres réseaux de recherche européens et américains via le réseau paneuropéen GEANT.

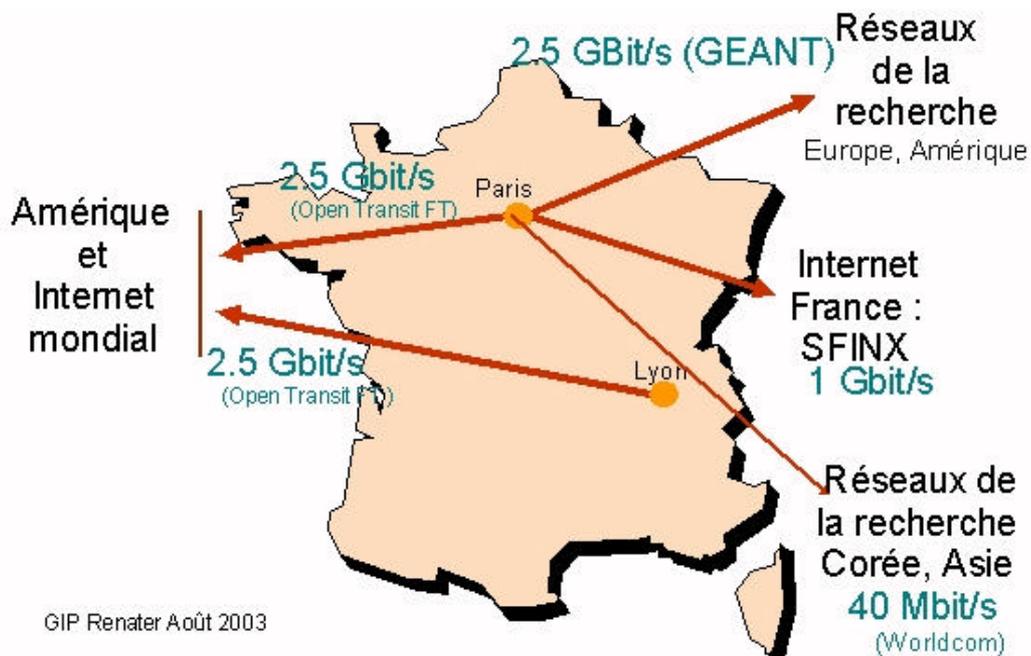


Figure 1 : Les liaisons du réseau Renater à l'international.

Une liaison de 20 Mbit/s aboutissant en Corée assure la communication avec les réseaux de la recherche de la zone Asie-Pacifique, notamment celui du Japon.

La communication avec l'Internet, en France, est réalisée par le nœud d'échange SFINX, auquel RENATER est relié à 1 Gbit/s. La communication avec l'Internet dans le reste du monde est assurée par le raccordement de RENATER à 5 Gbit/s à l'épine dorsale Internet mondiale OpenTransit de France Télécom.

b) Renater 3, épine dorsale du réseau Renater

Avec Renater 3, les débits des liaisons de l'épine dorsale sont presque partout de 2,5 Gbit/s ou davantage. Ces liaisons sont – presque partout – organisées en boucle : ceci augmente la disponibilité en cas d'incident – il y a toujours un chemin de secours disponible – et facilite la mise en place de liaisons de voisinage entre régions.

Ces débits atteignent même 80 Gbit/s en Ile de France.

LE RESEAU RENATER-3

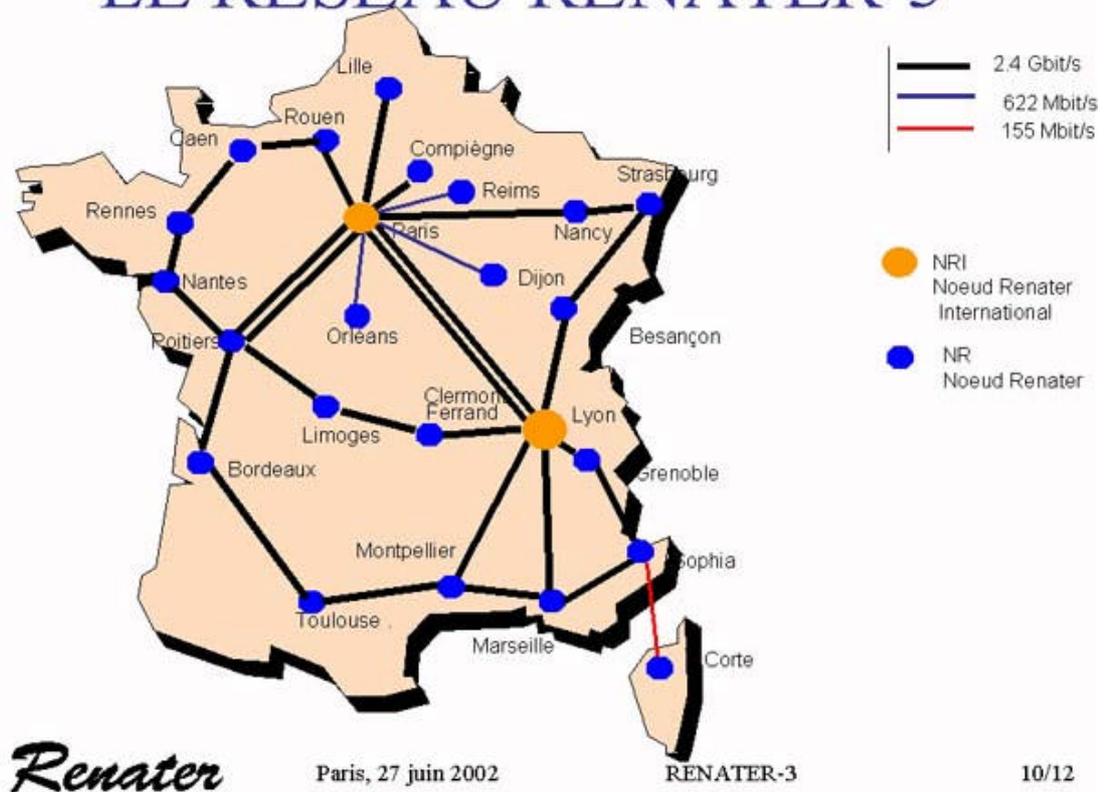


Figure 2 : La carte du réseau Renater 3.

c) Les services proposés par Renater

Au niveau des points d'accès régionaux à son épine dorsale, RENATER propose de nombreux services :

- Des services de réseau IP performants :

Un service de bout en bout avec qualité garantie et un guichet unique pour l'ensemble du territoire national, un service IPv4 conventionnel permettant d'accéder aux communautés de la recherche et de l'enseignement, nationales et internationales ainsi qu'à l'Internet.

- Des services de réseau avancés :

Un service IPv6, utilisé par les universités pour préparer la migration de IPv4 vers IPv6, ainsi que par de nombreux projets de recherche et développement : RNRT, RNTL, 6^e Programme Cadre...

Un service de diffusion IP (IP multicast), en IPv4 et en IPv6, utilisé pour la visioconférence et le téléenseignement interactif, un service d'appels pour la visioconférence en H.323, utilisé pour la visioconférence deux à deux et l'accès à des ponts de visioconférence pour les réunions de travail collaboratif.

Un accès à des contenus pédagogiques et culturels via une boucle à très haut débit reliant des sites à fort contenus culturels.

Ces services peuvent être utilisés par tous les organismes et sites de la communauté Renater si ceux-ci sont reliés aux points de présence, par des réseaux de collecte régionaux qui les relaient.

La sécurité est également un thème fort de RENATER: engagements et chartes de déontologies signés par les organismes utilisateurs, mesures techniques spécifiques dans le réseau, CERT-Renater y contribuent.

d) Renater 3, une technologie de pointe

- L'infrastructure : de la fibre optique à des dizaines de Gbit/s

Les technologies optiques, telles que le multiplexage de longueurs d'onde (DWDM), permettent d'utiliser la fibre optique à très haut débit : dans une même fibre, ce multiplexage permet de transporter plusieurs fois 2.5 Gbit/s, bientôt plusieurs fois 10 ou 40 Gbit/s. Ceci est utilisé en particulier sur l'épine dorsale nationale de RENATER 3 et dans la boucle optique qui constitue la partie francilienne à 80 Gbit/s de Renater.

- IPv6 : la nouvelle génération de l'Internet :

IPv6 est la nouvelle version du protocole IP de l'Internet. Elle va prendre la relève de la version 4, IPv4, au cours des prochaines années – et elle le fait déjà au Japon et en Asie-Pacifique.

En Europe, les réseaux de la recherche sont très actifs pour préparer et commencer cette migration, et proposer un service IPv6 pour les expérimentations des premiers utilisateurs. RENATER est au premier plan de cette évolution : dans RENATER 3, il y a un service IPv6 « natif », c'est à dire placé sur le même plan que le service IPv4 et tout aussi performant.

RENATER est même l'un des premiers au monde à proposer un service de diffusion (multicast) sur IPv6, clé du développement de la visioconférence et du téléenseignement sur cette nouvelle version du protocole IP !

- Former et accompagner les utilisateurs pour qu'ils maîtrisent ces technologies de pointe :

Les technologies des réseaux et de leurs utilisations avancées évoluent vite. En garder la maîtrise n'est pas chose simple, même avec la compétence que l'on trouve dans la communauté des utilisateurs de RENATER.

Pour les aider, RENATER organise des actions de formation ou d'accompagnement, et participe à d'autres actions animées par des experts de cette communauté : présentations techniques en vidéo : RENATER en vidéo, ainsi que des séminaires virtuels : les Causeries de RENATER, et aussi des cours en partenariat avec le CINES à Montpellier, et des conférences spécifiques : IPv6 en Octobre 2002, conférences JRES, IPv6 en Juin 2003, et même un groupe de travail spécifique pour les responsables réseaux d'organismes qui démarrent IPv6 : le GN6 ...

2. Le GIP Renater

Le GIP Renater réunit de grands organismes de recherche et d'enseignement, ainsi que le ministère en charge de l'éducation nationale, de la recherche et de la technologie, pour développer et faire fonctionner le réseau Renater.

Un GIP (Groupement d'Intérêt Public) est un organisme à but non lucratif réunissant des administrations de l'Etat et des organismes publics dans le cadre d'une activité bien définie, dans le cas du GIP Renater il s'agit de la gestion du réseau Renater.

En effet le GIP Renater est le maître d'ouvrage de la partie commune de Renater constituée de son épine dorsale Renater 3, des liaisons internationales, de ses actions pilotes et du SFINX (Service for French Internet eXchange). Il est le coordinateur technique et opérationnel global de l'ensemble du réseau Renater y compris de ses éléments régionaux. Il représente le réseau Renater auprès des institutions françaises et étrangères, et notamment auprès des autres réseaux de la recherche.

Le GIP Renater est financé : par les contributions de ses membres, par des subventions gouvernementales (et européennes pour certains projets spécifiques) et par les contrats passés par les organismes utilisateurs non membres (y compris les fournisseurs d'accès Internet qui utilisent le SFINX). Ses dépenses concernent : les coûts de Renater 3 et des liaisons internationales, les actions pilotes, le CERT, les services divers, les dépenses de fonctionnement, notamment les coûts liés au personnel.

C. LE GN6

De nombreux organismes de la communauté des utilisateurs de Renater désirent actuellement démarrer de petits réseaux IPv6 dans leurs organismes : cela leur permettra d'acquérir l'expertise nécessaire, et d'étudier concrètement divers aspects techniques et opérationnels de l'introduction de IPv6 : ils pourront ensuite planifier dans de bonnes conditions la migration globale de IPv4 vers IPv6 que l'on estime inéluctable dans les prochaines années.

Pour les aider et les accompagner tout au long de cette démarche, le GIP Renater a mis en place avec eux le Groupe des Néophytes IPv6, le GN6.

Le GN6 : un groupe de travail – à accès restreint - qui regroupe des acteurs « réseau » des organismes de la communauté Renater : organismes utilisateurs de Renater, réseaux de collecte de Renater, réseaux ou organismes étrangers qui ont une convention de coopération avec le GIP Renater.

Ses objectifs :

- permettre de s'initier à la mise en œuvre de IPv6 et de ses applications de base
- faciliter, par des échanges d'information et des retours d'expérience systématiques, le démarrage de réseaux pilotes IPv6 – qui devront être connectés au service IPv6 de Renater - dans les organismes
- faciliter, dans les mêmes conditions, la participation aux actions pilotes menées par le GIP Renater et/ou le G6, par exemple IPv6 multicast ou DNSsec
- faciliter l'accès aux connaissances des experts, notamment ceux du GIP Renater et de l'Association G6

Démarré mi-2002, le GN6 comporte déjà une vingtaine de personnes, et se réunit tous les deux mois. Certains de ses membres, éloignés de quelques milliers de kilomètres, participent régulièrement aux réunions en visioconférence.

D. Des actions de téléenseignement

Dans le cadre de ses activités ou bien encore dans le cadre de son partenariat avec le GIP Renater, l'association Aristote est à l'origine de plusieurs actions de téléenseignement au cours desquelles sont retransmis sur le réseau Renater des contenus à caractère pédagogique.

Voici une présentation des projets de téléenseignement dans lesquels est impliquée l'association Aristote.

1. Le projet ATHENA

Actions pour des Transmissions Harmonieuses et des Echanges de Natures Académiques

a) Présentation

ATHENA est un projet mis en œuvre par l'association Aristote en association avec différents partenaires pour l'étude et la validation de processus d'échanges académiques via des transmissions réseau de contenus scientifiques entre sites académiques francophones.

ATHENA a pour objectif de contribuer au développement d'échanges académiques à travers les NTIC (Nouvelles Technologies de l'Information et de la Communication) entre le réseau Renater et les milieux universitaires de ces pays.

A la différence d'autres expériences il ne s'agit pas de mettre à la disposition des partenaires des supports multimédia (cédéroms ou vidéocassettes par exemple), l'intérêt est de mettre en place un dispositif d'échanges en temps réel (interactif) à travers les outils modernes de communications et en particulier à l'aide de la visioconférence. D'où le besoin d'une analyse de la faisabilité technique avec l'étude des moyens de communications dont disposent les futurs partenaires : ordinateurs, équipements de visioconférence, connectivité Internet, réseaux analogiques, numériques, hertziens...

Les premiers tests sont validés par :

- des essais de visioconférence en point à point avec les partenaires
- la diffusion en direct d'évènements, l'ambition étant de faire du temps réel
- l'accent mis sur l'interactivité ne supposant pas l'ignorance d'autres dispositions notamment la mise, par chaque partenaire, à la disposition de l'ensemble du réseau constitué, des documents scientifiques à contenus consultables via Internet.

Ces échanges de contenu à caractères scientifiques avec les pays de l'Afrique francophone contribuent à une homogénéisation des connaissances dans les secteurs des télécommunications, réseaux...

Cette mise en partage des connaissances a pour objectif de donner aux étudiants africains le même savoir que celui dispensé aux étudiants européens.

La carence de personnel et de salles de cours dans beaucoup de pays africains pourrait être compensée par cet enseignement à distance.

Le projet ATHENA a été initié par l'apprenti du DESS ART au GIP Renater de la promotion 2000/2001, M. Harouna Siby, qui a prospecté et trouvé des contacts en Afrique et plus précisément au Sénégal, puis ce projet a vu arriver de nouveaux partenaires en Inde, en Tunisie et au Maroc avec le travail de M. Lionel David, apprenti du DESS ART au GIP Renater lors de l'année scolaire 2001/2002.

b) Les partenaires du projet ATHENA

- L'association Aristote et le GIP Renater que nous avons déjà présentés auparavant :



- L'université Paris 7 Denis Diderot :



C'est le DESS Applications des Réseaux et de la Télématique dépendant de l'UFR d'informatique de l'Université Paris 7 qui participe au projet ATHENA.



D.E.S.S.A.R.T
Université Paris 7 - Denis Diderot
Case 7068
2 place Jussieu
75251 Paris CEDEX 05

Ce DESS, qui utilise les services de VisioP7, le centre de visioconférence de l'Université Paris 7, se situe à la pointe de la technologie concernant les expérimentations de téléenseignement utilisant les technologies de la visioconférence.

Sa formation est ancrée sur la réalité du marché et la diversité des technologies, alliant Télématique, Télécommunications et Multimédia.

- L'université d'Evry :



L'université d'Evry est représentée dans le projet ATHENA par le DESS Ingénierie Documentaire et Multimédia.

Les étudiants sortant de cette formation sont des informaticiens capables de maîtriser à la fois les domaines techniques et organisationnels liés à la gestion, la création et l'utilisation de l'information documentaire et multimédia dans des environnements variés.

- L'ESMT à Dakar :



L'Ecole Supérieure Multinationale de Télécommunications (ESMT) située à Dakar est née en 1981 de l'initiative de sept pays de la sous-région de l'Afrique de l'Ouest (Bénin, Burkina Faso, Mauritanie, Niger, Sénégal, Togo), avec le concours de la coopération internationale dans le cadre d'un projet du Programme des Nations Unies pour le Développement (PNUD).

L'ESMT est maintenant une institution multinationale qui a pour vocation de former des ingénieurs en télécommunications spécialisés dans les domaines technique et commercial. Elle accueille en formation initiale ou en formation continue des stagiaires issus des pays membres et d'autres pays utilisateurs comme le Tchad, la Guinée Conakry, le Burundi, Djibouti, Madagascar, la Côte d'Ivoire, le Cameroun, le Congo...

L'ESMT offre également des possibilités de rencontres et d'échanges, en organisant des forums, des séminaires et des réunions à l'échelle africaine et internationale tels que le forum sur les radiocommunications mobiles regroupant opérateurs africains et consortiums internationaux (1994 et 1997) ou le séminaires interrégional sur les techniques et usages de la télématique (1996).

L'ESMT bénéficie d'aides de la part de ses partenaires internationaux sous diverses formes :

- renforcement des infrastructures techniques
- renforcement des ressources pédagogiques
- soutien d'assistance technique
- soutien financier

Le PNUD (Programme des Nations Unies pour le Développement) et l'UIT (Union Internationale des Télécommunications) sont à l'origine de l'ESMT et ont permis le développement de ses activités notamment par l'envoi d'experts pour la mise en place des formations (à commencer par celle des formateurs) et en permettant l'acquisition des équipements de laboratoire et des divers matériels logistiques.

La coopération avec la France s'est pratiquement mise en place dès la création de l'ESMT. Elle s'est traduite par la mise à disposition d'experts (France Télécom, etc), de CSN, d'aide au développement des ressources humaines de l'école et de dons en matériels.

L'appui de la Suisse à l'ESMT se manifeste de diverses manières : formation pédagogique de formateurs, octroi de bourses d'études à des élèves, dons de matériels didactiques, organisation en commun de séminaires

Dans le cadre de son programme d'appui institutionnel aux pays participant au projet PANAFTEL-ACDI, le Canada soutient diverses actions (installations d'énergie solaire, téléphonie rurale, transfert de compétences par des experts de CEGIBEL, etc.).

Fort de tous ses partenaires, l'ESMT se veut un pôle de développement des télécommunications des pays membres et utilisateurs, afin d'accroître la compétitivité des entreprises dans un environnement en profonde mutation.

- L'Université Cheikh Anta Diop (UCAD) à Dakar :



L'Université de Dakar a été créée le 14 février 1957 et elle a été inaugurée en décembre 1959.

Héritière de l'école africaine de Médecine créée en 1915, l'Université de Dakar a connu une longue évolution marquée en 1949 par la création d'un enseignement préparatoire aux études médicales et par l'ouverture au début des années cinquante d'écoles supérieures académiquement rattachées à l'Université de Bordeaux, puis érigées en 1957 en Facultés indépendantes pour former l'Université de Dakar.

Devenue le 30 mars 1987 l'Université Cheikh Anta Diop de Dakar, l'Université de Dakar est la plus ancienne et la plus importante structure d'enseignement supérieur existant actuellement au Sénégal. En dehors des services administratifs centraux du rectorat elle est composée de trente établissements d'enseignement supérieur et de recherche se répartissant comme suit : onze facultés, dix-neuf instituts d'université ainsi que l'Ecole Inter états des Sciences et Médecine Vétérinaires qui dépend scientifiquement de l'Université.

C'est l'Ecole supérieure Polytechnique rattachée à l'UCAD qui participe au projet ATHENA. L'Ecole Supérieure Polytechnique est répartie sur cinq départements : Génie Chimique, Génie Civil, Génie Electrique, Génie Mécanique et enfin le département Génie Informatique qui participe au projet ATHENA.

- Le CIRAD à Montpellier :



Organisme français spécialisé en recherche agronomique appliquée aux régions chaudes, le CIRAD a pour mission de contribuer au développement rural des pays tropicaux et subtropicaux par des recherches, des réalisations expérimentales, des actions de formation, en France et à l'Etranger. Ses activités recouvrent les domaines des sciences agronomiques, vétérinaires, forestières et agroalimentaires.

- Le Centre de Calcul El Khawarezmi (CCK) à Tunis :



Le CCK, établissement sous tutelle du Ministère de L'Enseignement Supérieur et dépendant de l'Université El Manar de Tunis a été créé en octobre 1976 à la suite de l'introduction de l'Informatique en tant que discipline autonome dans les milieux universitaires.

Le Centre de Calcul El Khawarezmi a été désigné au mois de juillet 1997 comme fournisseur de services Internet au profit des établissements d'enseignement supérieur et de recherche.

Sa mission principale est de mettre à la disposition des enseignants, des chercheurs, des doctorants et des étudiants exerçant au sein des institutions dépendant du Ministère de l'Enseignement Supérieur, les moyens et les services nécessaires pour favoriser le développement de la recherche scientifique par le biais de l'informatique. Dans le cadre de la réalisation du projet du Réseau National Universitaire (RNU) retenu dans le 9ème plan, le centre de calcul El Khawarizmi a été désigné, au mois de Juillet 1997 (circulaire 97/047 du 26 Juillet 1997) comme Fournisseur de Services Internet au profit des établissements de l'Enseignement Supérieur.

Dans le cadre du projet BIRUNI (projet d'informatisation des bibliothèques universitaires tunisiennes), le CCK permet l'interconnexion des bibliothèques

universitaires et assure l'hébergement du serveur central d'informations bibliographiques.

Le Centre participe au projet de l'utilisation des technologies avancées comme moyen de soutien à l'enseignement et à la pédagogie, notamment pour les applications que constituent le téléenseignement ou la visioconférence.

- Le réseau MARWAN au Maroc :



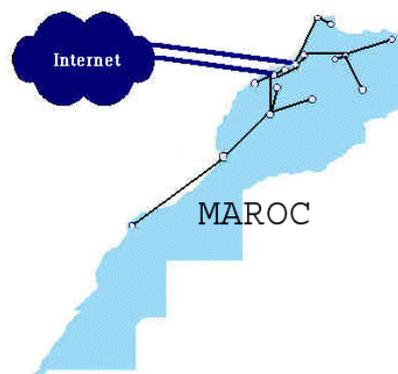
Acronyme de l'expression Maroc Wide Area Network (*réseau longue distance marocain*), le MARWAN est un réseau informatique national à but non lucratif dédié à l'éducation, à la culture et à la recherche.

Sa mise en place, décidée par le Premier Ministre marocain est effectuée conjointement par le Ministère de l'Enseignement Supérieur, de la Formation des Cadres et de la Recherche Scientifique ; le Ministère de l'Education Nationale ; Le Ministère du Développement Social, de la Solidarité, de l'Emploi et de la Formation Professionnelle et le Secrétariat d'Etat auprès du Premier Ministre chargé de la Poste et des Technologies de l'Information.

Ce réseau correspond à un choix stratégique qui consiste à fédérer l'infrastructure d'information et de communication des établissements et à connecter ces derniers aux réseaux internationaux équivalents.

Le MARWAN facilite les échanges d'information à l'échelle régionale, nationale et internationale.

Il permet la diffusion du savoir et offre également aux établissements scolaires, universitaires et de formation professionnelle ainsi qu'aux centres de recherche nationaux, la possibilité d'utiliser les technologies de l'information et de la communication.



Les Nouvelles Technologies de l'Information et de la Communication sont en train de provoquer une révolution, aussi importante que la révolution industrielle.

Le réseau MARWAN est l'outil adéquat qui va permettre au Maroc de développer ce secteur et en faire un puissant levier pour le développement et le décollage de la recherche scientifique et technique dans ce pays.

Le réseau MARWAN a pour objectifs :

- La généralisation de l'enseignement et l'amélioration de sa qualité par le développement des différents modes d'enseignement et de formation à distance (téléenseignement, télé médecine...)
- Le développement de la recherche scientifique et technique grâce à la multitude des formes de communications offertes par ce réseau et par la création de bases de données et de bases documentaires spécifiques
- L'amélioration du transfert de technologie et de la coopération internationale en connectant le réseau MARWAN aux autres réseaux internationaux similaires
- La généralisation et la vulgarisation de l'utilisation des technologies de l'information et de communication par la couverture de l'ensemble des établissements d'enseignement de formation et de recherche, de l'école à l'université, et sur la totalité du territoire marocain
- L'amélioration et la rationalisation de la gestion des ressources par le partage et l'échange de ces ressources, en réalisant d'importantes économies par le remplacement des modes classiques de communication, de transfert et de publication de l'information par les Nouvelles Technologies de l'Information et de la Communication
- La création d'emplois par la création de nouveaux métiers et nouveaux services autour de ces Nouvelles Technologies.

- L'université de Guadalajara au Mexique :

UNIVERSIDAD DE GUADALAJARA

L'université de Guadalajara a été fondée en 1792 sous le nom d'Université Royale de Guadalajara. C'est un organisme public décentralisé du gouvernement de l'état de Jalisco, qui jouit de son autonomie et d'un patrimoine personnel. Ses objectifs sont d'offrir des formations initiales et des formations continues à destination des techniciens, universitaires et étudiants, mais aussi plus généralement de diffuser la culture, la science et la technologie.

- L'Institut Indien de Technologie de Kanpur en Inde (IITK) :



L'Institut Indien de Technologie de Kanpur est engagé dans la recherche et la mise en œuvre des nouvelles technologies. Il forme des ingénieurs motivés et compétents. L'Institut célèbre la liberté de penser, cultive la vision et encourage la croissance mais inculque également des valeurs humaines et le souci de l'environnement de la société. L'Institut forme des bacheliers, des maîtres et des doctorants dans diverses branches technologiques et scientifiques. L'IITK a mis l'accent sur le recrutement d'un corps enseignant compétent venant de différents points géographiques nationaux et internationaux ainsi qu'une sélection très stricte des étudiants.

L'IITK est un établissement technologique d'importance nationale pour l'éducation mais aussi pour la recherche.

Concernant les nouvelles technologies de l'Internet, l'IITK profite de subventions du gouvernement indien, de projets, de personnes compétentes, étudiants et chercheurs.

Le projet ATHENA constitue donc pour l'association Aristote un moyen de nouer des partenariats et de partager des connaissances et des savoir-faires avec des partenaires académiques du monde entier.

2. Les cours DIM

a) Présentation

DIM (Diplômes d'Ingénierie Multimédia): c'est le nom d'un projet commun à plusieurs DESS – ou enseignements équivalents – de technologies d'informatique et de réseau.

DIM leur permet d'organiser régulièrement des cours communs, partagés en interactif grâce à la visioconférence et à des outils de travail coopératif sur Renater.

Ces enseignements sont dispensés à :

- L'Université d'Evry Val d'Essonne (UEVE)
- L'Institut National des télécommunications à Evry (INT)
- L'Université Paris VII sur le campus de Jussieu
- L'Université de Valenciennes et du Hainaut-Cambrésis (UVHC)
- L'Ecole Supérieure Multinationale de Télécommunications de Dakar (ESMT)

Le GIP Renater apporte sa contribution au support technique de DIM : réseau et service IP multicast, technologies de vidéo numérique sur IP.

DIM a démarré en Octobre 2001. Voici une vue de la salle du DESS ART à Paris VII, lors de la réception du premier cours DIM émis par le DESS IDM à Evry :



Figure 3 : à gauche se trouvent les transparents de l'orateur, à droite l'image de l'orateur. Les élèves peuvent poser des questions grâce aux micros qui se trouvent sur les tables.

Les cours sont dispensés à tour de rôle par un enseignant de chacune des cinq formations concernées.

Les cours DIM constituent une expérience grandeur nature de l'usage de la visioconférence dans le cadre du téléenseignement interactif, en effet en plus de voir et d'entendre l'orateur les étudiants peuvent poser des questions pendant le cours.

Jusqu'à présent les cours DIM étaient retransmis en IPv4 multicast mais à terme l'objectif est de les retransmettre en IPv6 multicast.

b) Perspectives d'améliorations

Les cours DIM se déroulent avec succès depuis maintenant deux années, cependant certaines innovations permettraient d'améliorer le confort et l'interactivité des sessions, par exemple la résolution maximale autorisée par le logiciel de vidéoconférence utilisé pour ces séances est actuellement de 352x288 ce qui implique une qualité de la vidéo de l'orateur moyenne et, plus contraignant encore, l'impossibilité de retransmettre dans de bonnes conditions les transparents des cours par le biais de ce logiciel car la lisibilité n'est alors pas suffisante.

Voici ce que donne un transparent transmis par le logiciel de vidéoconférence en 352x288:

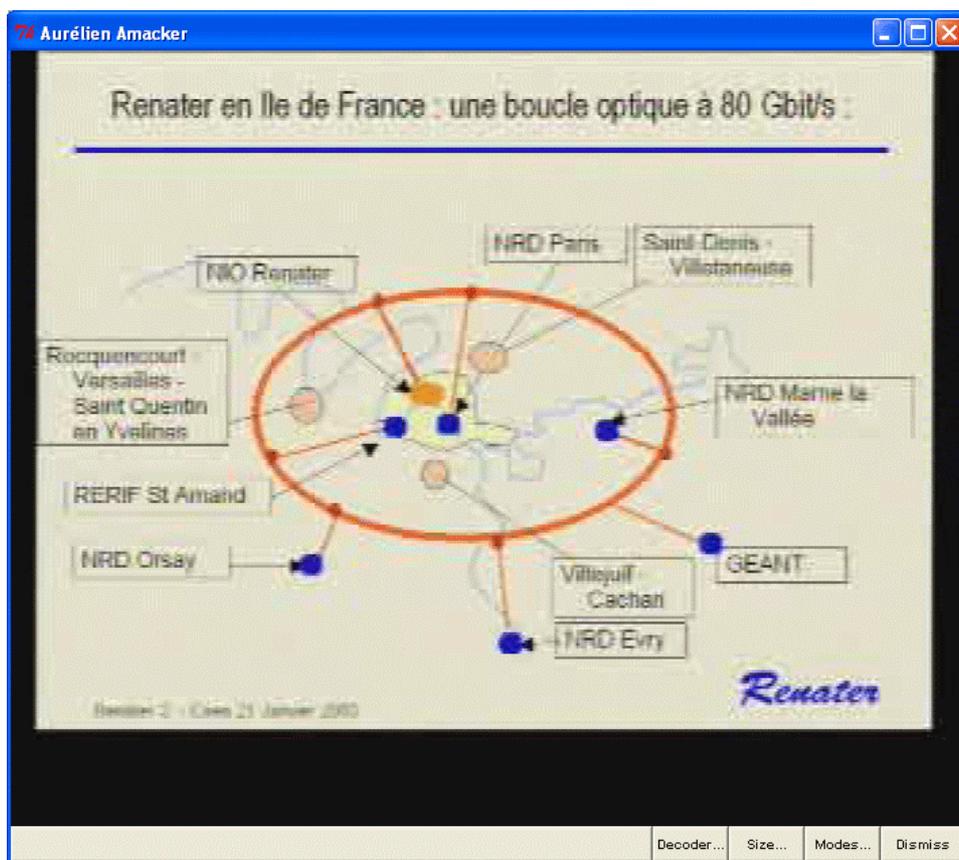


Figure 4 : Transparent transmis en 352x288.

On constate que la lisibilité n'est pas suffisante pour offrir aux récepteurs le confort nécessaire.

La solution actuellement utilisée pour retransmettre les transparents lors des cours DIM est d'utiliser le partage de fichiers de Netmeeting, cependant l'expérience des cours DIM a montré que cette solution manque de stabilité puisque régulièrement les participants à la session sont déconnectés et à ce moment-là il faut que tous les participants se déconnectent et se reconnectent ce qui entraîne une perte de temps.

Une version du logiciel de visioconférence utilisé lors des cours DIM autorisant une résolution de 704x576 serait une solution à ce problème puisqu'elle permettrait de retransmettre les transparents à l'aide de ce logiciel de visioconférence et donc de s'affranchir de l'usage de NetMeeting.

Enfin un problème pratique qui se pose également lors des cours DIM est le suivant : lorsqu'un élève veut poser une question il est obligé d'interrompre le professeur ou bien d'attendre que celui-ci demande à l'auditoire si quelqu'un désire poser une question, ce qu'il ne fait en général qu'à la fin du cours ou juste avant la pause.

Une solution à ce problème serait de développer un logiciel permettant à un élève de demander la parole.

Du côté de l'orateur ceci se manifesterait par un signal visuel sur un écran de contrôle.

Ceci permettrait donc de proposer un mode de fonctionnement similaire à celui qui est utilisé pour des cours en présentiel : l'élève lève sa main pour avertir le professeur qu'il désire poser une question et lorsqu'il le désire le professeur peut donner la parole à l'élève. Cette manière de procéder permettrait donc d'éviter aux élèves de couper la parole du professeur.

3. Les séminaires X-Aristote

a) Présentation

Dans le cadre de ses activités de diffusion de la connaissance auprès de ses membres, l'Association Aristote organise régulièrement des séminaires de haut niveau : les séminaires X-Aristote. Ils sont groupés en cycles de 6 séminaires d'une journée chacun par année scolaire.

Ces séminaires se déroulent dans un amphithéâtre de l'Ecole Polytechnique.

Ils sont principalement destinés aux utilisateurs évolués, aux administrateurs de réseaux et d'ordinateurs, et aux responsables et décideurs.

Chaque séminaire est consacré à un thème précis, choisi dans le domaine des technologies des réseaux et de leurs applications de base. Il en aborde divers aspects: principes et état de l'art, produits et réalisations industrielles et/ou opérationnelles; compte rendus d'expériences ou de stratégies d'utilisateurs. Il fait intervenir des orateurs de haut niveau : experts internationaux, spécialistes des organismes d'Aristote, représentants des industriels et des fournisseurs de produits.

Les séminaires X-Aristote font systématiquement l'objet d'une retransmission en IPv4 multicast et d'un passerillage en IPv6 multicast. A terme l'objectif est de les retransmettre en IPv6 multicast et de les passereller en IPv4 multicast afin d'accompagner le processus de transition du protocole IPv4 vers le protocole IPv6.

b) Perspectives d'améliorations

Etant donné que les personnes désirant assister à la retransmission en multicast des séminaires X-Aristote ne bénéficient pas tous de la même bande passante, la retransmission de la vidéo est effectuée avec un débit assez faible (128 kbit/s) afin de la rendre accessible au plus grand nombre, cependant ceci est frustrant pour toutes les personnes qui possèdent une bande passante importante car ils ne peuvent bénéficier d'une vidéo de bonne qualité.

L'emploi d'une passerelle permettant de réduire le débit d'une session vidéo afin de proposer une session haut débit et une session bas débit a été expérimenté mais cette solution ne s'est pas révélée satisfaisante par manque de stabilité de la part de ce logiciel.

La solution envisagée par la suite a donc été de disposer d'une version du logiciel de vidéoconférence permettant d'émettre un même signal vidéo à deux débits différents. Ainsi nous pourrions émettre de la vidéo à un débit relativement faible (moins de 128 kbit/s) pour les gens ne disposant pas d'une bande passante importante et de la vidéo de haute qualité (plus de 512 kbit/s) pour ceux disposant d'une large bande passante.

4. Les Causeries de Renater

a) Présentation

Les *Causeries de Renater* sont des conférences virtuelles d'une demi-journée à une journée disponibles sur Renater.

Elles se déroulent entièrement en télé présence, grâce aux possibilités de visioconférence permises par les performances de Renater.

Les orateurs et les auditeurs sont dans des salles de réunion équipées pour le multimédia, ou parfois simplement dans leurs bureaux : ces lieux sont réunis en une vaste salle de conférence virtuelle par le réseau RENATER et les autres réseaux de la recherche : certains sont en effet à l'étranger. Tous les participants peuvent voir et entendre l'orateur, interagir et poser des questions, puis participer au débat.

Au total, et suivant le sujet de la Causerie, il y a de 10 à 50 points d'écoute, pouvant représenter plus d'une centaine de personnes.

Chaque Causerie est dédiée à un thème précis. Il est traité par un ou plusieurs orateurs de haut niveau : membres de l'équipe du GIP Renater, responsables réseaux des Centres de Ressources Informatiques des universités, professeurs de DESS d'informatique, experts reconnus, utilisateurs compétents etc...

Certains des thèmes retenus jusqu'ici concernent le réseau Renater: déploiement, opération, technologies. D'autres abordent des applications avancées ou innovantes rendues possibles par Renater.

Les Causeries sont ouvertes à toute la communauté des utilisateurs de Renater.

Les organisateurs des Causeries sont actuellement : Guy Bisiaux (Université de Valenciennes) et Jacques Prévost (GIP Renater).

Note historique : les Causeries ont été connues autrefois sous les noms Causeries de l'UREC¹ puis Causeries du FMBone².

Les Causeries de Renater étant diffusées en IPv4 multicast, là encore l'objectif est à terme de les retransmettre en IPv6 multicast et de les passereller en IPv4 multicast.

b) Perspectives d'améliorations

De la même manière que pour les séminaires X-Aristote, actuellement le débit de la retransmission de la vidéo des Causeries de Renater est nivelé par le bas et là encore une version de Vic permettant d'émettre à deux débits donnerait la possibilité à l'auditoire de choisir la session vidéo la mieux adaptée à sa bande passante.

5. La Grille de Conférences

La Grille de Conférences est le plus récent des projets dans lequel est impliquée l'association Aristote, il s'agit de proposer une normalisation du mode opératoire des techniques utilisées pour la vidéoconférence et la retransmission en multicast d'évènements à caractère pédagogique.

¹ Unité REseaux du CNRS.

² *French Multicast backBONE*, épine dorsale multicast française.

La Grille de Conférences réunit sur Renater (et ailleurs) un ensemble de noeuds (points d'écoute et de participation) de visioconférence, équipés et opérés de manière cohérente et coordonnée.

La Grille permet de réaliser aisément des conférences virtuelles, avec des groupes d'auditeurs situés à chaque noeud.

La Grille sépare totalement l'organisation de la virtualité, qui est du domaine des animateurs des noeuds, et l'organisation du contenu, qui est du domaine de l'animateur de chaque conférence : celui-ci peut préparer sa conférence ou son séminaire pratiquement sans avoir à se soucier de la virtualisation – et en bénéficiant d'auditeurs à distance, capables de poser des questions et de participer aux débats, et aussi en programmant, s'il le désire, des intervenants à distance : il suffit qu'ils se rendent au noeud de la Grille le plus proche.

La Grille de Conférences, c'est :

- Une technologie de pointe pour la visioconférence dans chaque noeud
- Un mode opératoire commun bien défini
- Des interconnexions réseau de qualité entre les noeuds
- Un accueil et une mise en oeuvre dans chaque noeud

Actuellement on trouve des noeuds de la grille de conférence à Caen, Dakar, Paris, Montpellier et Toulouse.

La Grille s'efforce d'intégrer les technologies validées par les utilisations à grande échelle de ces derniers temps, notamment DIM, Causeries de Renater et séminaires X-Aristote, ainsi que les conclusions des retours d'expérience de ces utilisations pilotes et notamment leurs suggestions d'améliorations.

E. Objectifs de mon stage

Les objectifs de mon stage étaient donc d'une part de poursuivre le projet ATHENA en maintenant le contact avec les partenaires actuels, en développant de nouveaux contacts laissés par mon prédécesseur et d'autre part de développer de nouvelles fonctionnalités permettant d'améliorer le confort et l'interactivité des actions de téléenseignement mises en oeuvre par l'association Aristote, en effet ces nouvelles fonctionnalités devraient permettre de transmettre les transparents en multicast mais aussi de donner accès à des récepteurs situés sur des liaisons bas débit à de la vidéo en multicast.

Il est à souligner que mon travail s'est inscrit dans des projets à vocation technique (M6Bone¹) ou relationnelle (ATHENA) qui ont commencé bien avant le début de mon apprentissage et qui se termineront bien après, mon travail a donc consisté à apporter ma contribution personnelle à ces projets afin d'en assurer le développement.

C'est une procédure classique dans le monde de la recherche : ainsi, les outils de visioconférence sur IP multicast ont-ils déjà bénéficié de la contribution de nombreux développeurs, sans doute plus d'une centaine.

C'est aussi la méthode de développement classique des logiciels libres.

¹ *Multicast IPv6 backBONE*, épine dorsale multicast IPv6.

III. Les technologies employées

A. IPv6

La version actuelle du protocole IP, IPv4, a été initialement conçue pour un réseau comprenant une centaine d'ordinateurs. Avec le web, Internet a vu son utilisation augmenter de manière exponentielle, si bien que la saturation du réseau a été initialement prévue pour 1994. Des solutions comme le NAT (*Network Address Translation*, translation d'adresses réseau) et CIDR (*Classless InterDomain Routing*, routage inter domaine sans classes) ont permis de ralentir considérablement l'explosion du réseau et de gagner du temps afin de pouvoir mettre au point une nouvelle version du protocole IP.

Cette nouvelle version, appelée IPv6, a été conçue pour remédier aux limitations du protocole IPv4 mais aussi pour répondre à de nouveaux besoins au niveau des réseaux ou des applications.

Voici quelques unes des principales fonctionnalités du protocole IPv6 :

1. Un plus grand nombre d'adresses IP

Un des besoins les plus importants auquel répond IPv6 consiste en un nombre d'adresses quasiment illimité, en effet si les adresses IPv4 étaient codées sur 32 bits, offrant ainsi 2^{32} adresses différentes, les adresses IPv6 sont codées sur 128 bits, offrant ainsi 2^{128} adresses soit 2^{96} fois le nombre d'adresses IPv4. Ainsi le nombre d'adresses IPv6 disponibles a été estimé comme étant situé entre 1 564 et 3 911 873 538 269 506 102 adresses par mètre carré (océans compris) de la surface de la terre, on peut donc considérer ce nombre d'adresses comme étant virtuellement illimité.

La représentation textuelle d'une adresse IPv6 se fait en découpant le mot de 128 bits de l'adresse en 8 mots de 16 bits séparés par le caractère « : », chacun d'eux étant représenté en hexadécimal. Par exemple :

```
fedf:0000:0000:0000:0400:aab8:6543:220f
```

Dans chaque champ il n'est pas nécessaire d'écrire les « 0 » placés en tête et plusieurs champs nuls consécutifs peuvent être abrégés par le symbole « :: », cependant ce symbole ne peut apparaître qu'une seule fois dans une adresse pour éviter les ambiguïtés. De cette manière l'adresse précédente peut s'écrire :

```
fedf::400:aab8:6543:220f
```

Le protocole IPv6 distingue trois types d'adresses: multicast (multidiffusion), unicast et anycast (RFC 3513).

Une adresse unicast désigne une interface réseau, un paquet envoyé à une telle adresse sera donc remis à l'interface ainsi identifiée.

Une adresse multicast désigne un ensemble d'interfaces qui en général appartiennent à des nœuds différents pouvant être situés n'importe où dans l'Internet. Lorsqu'un paquet a pour destination une adresse de type multicast, il est acheminé par le réseau à toutes les interfaces membres de ce groupe. A noter qu'en IPv6 la notation d'adresse de broadcast a

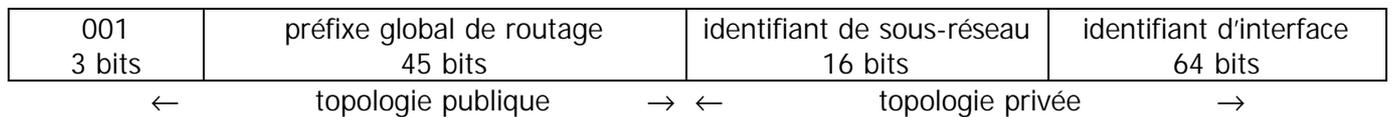
disparu pour être remplacée par une adresse multicast, ce qui permet d'économiser des ressources machines puisque de cette manière les ordinateurs peuvent éliminer des paquets qui ne leur sont pas destinés dès leur réception au lieu d'avoir à faire remonter l'information contenue dans le paquet jusqu'au niveau application.

Une adresse anycast désigne comme pour une adresse multicast un ensemble d'interfaces, la différence étant que lorsqu'un paquet est acheminé à une telle adresse il est acheminé à un seul élément du groupe et non à tous. C'est, par exemple, le plus proche au sens de la métrique des protocoles de routage.

2. Une meilleure hiérarchisation des adresses

Un des objectifs principaux d'IPv6 a été de mieux hiérarchiser les adresses afin de permettre de plus grands agrégats et donc de réduire la taille des tables de routage des routeurs situés sur les épines dorsales IP.

Voici le schéma d'une adresse IPv6 agrégée (RFC 3587) :



Il est à noter que la partie publique peut encore être amenée à changer, quant à elle la partie privée conservera cette structure.

Le préfixe global de routage permet aux opérateurs d'aggréger les tables de routage. Les opérateurs attribuent aux sites des réseaux dont le préfixe est de taille 48 bits, ce qui permet à ces sites de pouvoir adresser des sous-réseaux de préfixe de taille 64 bits.

Cette structure hiérarchique permet de pallier un des problèmes majeurs du protocole IPv4, à savoir l'explosion de la taille des tables de routage des routeurs des dorsales IP.

B. La multidiffusion (multicast) en IPv6

Une adresse multicast ne désigne pas une interface mais un ensemble d'interfaces.

Un couple (adresse multicast, numéro de port) désigne un *groupe multicast* ou bien encore *un groupe de multidiffusion*.

Lorsqu'une application s'abonne à un groupe de multidiffusion elle reçoit toutes les données émises par tous les membres du groupe et à l'inverse lorsqu'elle émet des données à destination de ce groupe tous les membres du groupe reçoivent ces données.

Le multicast est donc une technologie tout à fait adaptée à un contexte de diffusion de groupe comme c'est le cas pour la visioconférence.

En multicast, lorsqu'une source émet des données à destination d'un groupe les équipements réseau concernés répliquent les données de telle sorte que toutes les stations membres de ce groupe reçoivent une copie de ces données.

Prenons l'exemple suivant : une source émet de la vidéo et 4 ordinateurs (C1, C2, C3 et C4) veulent recevoir le flux vidéo. Sur le schéma de gauche nous pouvons observer la transmission des paquets en unicast et sur le schéma de droite nous pouvons observer la transmission des paquets en multicast.

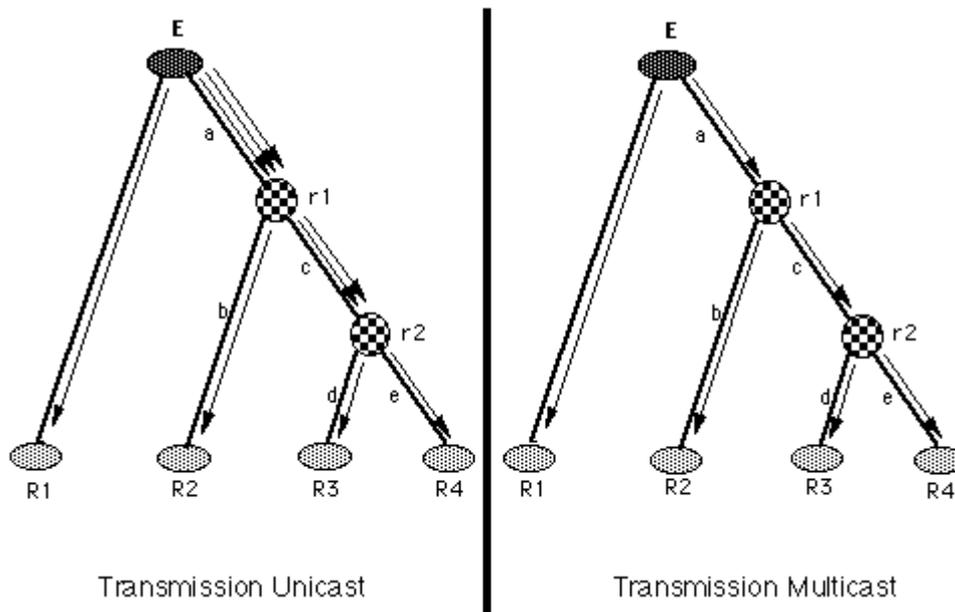


Figure 5 : Différences entre une transmission en unicast et une transmission en multicast.

L'utilisation du multicast permet donc de réaliser une économie de bande passante puisqu'un seul exemplaire des données est émis par la source.

Concernant les protocoles de routage utilisés par le multicast, on peut considérer deux niveaux distincts : les protocoles au niveau intra-domaine et les protocoles au niveau inter-domaine. Au niveau intra-domaine on a encore deux types de protocoles : les protocoles au niveau lien local et les protocoles au niveau inter-liens.

1. Protocoles de routage au niveau local

Au niveau local, le protocole de routage utilisé pour le multicast IPv6 est MLD (Multicast Listener Discovery, découverte de récepteurs multidiffusion) qui remplace le protocole IGMPv2 (*Internet Group Management Protocol version 2*, protocole de gestion de groupes version 2) d'IPv4. Ce protocole permet aux ordinateurs de rendre compte aux routeurs de sortie de leur réseau de leur appartenance à un groupe multicast. Les applications utilisent donc MLD pour s'abonner ou se désabonner à un groupe multicast. Les messages IGMPv3 sont véhiculés dans des paquets ICMPv6 (Internet Control Message Protocol, protocole de message de contrôle Internet pour IPv6).

Sur chaque réseau un routeur est élu par un mécanisme d'élection et ce sera lui qui traitera les appartenances des ordinateurs aux différents groupes multicast.

Voici la structure d'un message MLD :

type (8 bits)	code (8 bits)	checksum (16 bits)
délai maximum de réponse (16 bits)		réservé (16 bits)
adresse multicast IPv6 (128 bits)		

Le champ type permet de distinguer les 3 types de messages MLD:

Multicast Listener Report : (type = 131 en décimal) Ce message est envoyé au groupe de tous les routeurs multicast sur le lien (adresse multicast ff02::2). Quand une application désire s'abonner à un groupe elle va émettre un message *multicast listener report* avec un champ « adresse multicast IPv6 » contenant l'adresse multicast du groupe dont elle veut faire partie.

Le routeur qui reçoit ce message rajoute une entrée dans sa table MLD si elle n'existait pas déjà et commence à diffuser sur le réseau local les données à destination de ce groupe.

Multicast Listener Done : (type = 132 en décimal) Quand une application veut quitter un groupe multicast, elle envoie ce message au groupe des routeurs multicast (adresse ff02::2) en spécifiant dans le champ « adresse multicast IPv6 » l'adresse IPv6 du groupe qu'elle veut quitter.

Multicast Listener Query : (type = 130 en décimal) Il y a deux types de messages *Multicast Listener Query* qui diffèrent par le contenu de leur champ « adresse multicast IPv6 »

- General Query : ce type de message est utilisé par le routeur pour connaître tous les groupes auxquels sont abonnées les applications utilisant le multicast sur le réseau local. Dans ce cas le champ « adresse multicast IPv6 » possède la valeur 0. Ce message est envoyé au groupe multicast de toutes les stations sur le réseau local (ff02::1)

Lorsqu'elles reçoivent ce message MLD les stations déclenchent un compte à rebours avec une valeur aléatoire pour chacun des groupes multicast auxquels elles sont abonnées. Quand le compte à rebours correspondant à un groupe multicast est écoulé, la station envoie un message *Multicast Listener Report* (rapport de récepteur multicast) pour ce groupe si aucun

rapport n'a été envoyé pour ce groupe auparavant. Ainsi seule une réponse par groupe est donnée.

2. Protocoles de routage inter-liens

Les réseaux locaux qui supportent le multicast peuvent se grouper en domaines afin d'échanger entre eux du trafic multicast. En pratique on constate que sur une même zone géographique des réseaux locaux sont administrés de manière cohérente au sein d'un même domaine. Les protocoles de routage multicast inter-liens ont été développés pour répondre aux problèmes suivants :

- Comment atteindre les membres des différents groupes de diffusion répartis sur tout un domaine (arbres de diffusion) ?
- Comment économiser de la bande passante en n'acheminant les paquets multicast que là où il y a des abonnés ?
- Comment optimiser les échanges entre routeurs ? (vaut-il mieux annoncer les groupes que l'on veut recevoir où les groupes que l'on ne veut pas recevoir ?)

On peut distinguer deux familles de protocoles multicast qui résultent de deux approches différentes :

- Les protocoles de routage orientés « forte densité de récepteurs » (*dense mode*, mode dense) comme DVMRP (*Distance Vector Multicast Routing Protocol*, protocole de routage multicast à vecteur de distance) qui a aujourd'hui presque disparu et PIM-DM (*Protocol Independent Multicast – Dense Mode*, multicast indépendant du protocole – mode dense) qui est plus récent.

Ces protocoles partent du principe qu'il y a des membres de groupe multicast sur la plupart des réseaux et que l'absence de membres sur un réseau constitue l'exception. Dans un domaine PIM-DM, les routeurs vont inonder périodiquement tout le domaine en transmettant tous les flux multicast à leurs voisins. Les routeurs qui ne veulent pas recevoir le trafic multicast demandent à leurs voisins de cesser de leur envoyer ces flux (mécanisme de *pruning* ou d'élagage). C'est de cette manière qu'est créé l'arbre de diffusion.

Le problème est qu'un routeur qui ne désire pas recevoir de trafic multicast va passer son temps à demander qu'on cesse de lui envoyer ces flux multicast, c'est pourquoi cette approche a disparu au profit du mode clairsemé (*sparse mode*).

- Les protocoles de routage orientés « faible densité de récepteurs » (*sparse mode*, mode clairsemé) comme PIM-SM (*Protocol Independent Multicast-Sparse Mode*, multicast indépendant du protocole, mode clairsemé).

Ces protocoles supposent au contraire des précédents que les membres de groupes multicast sont très dispersés et peu nombreux par rapport au nombre de réseaux desservis.

Un ou plusieurs points de rendez-vous sont configurés dans le domaine PIM. Ces points de rendez-vous connaissent l'ensemble des sources et des récepteurs des différents groupes du domaine, ainsi ils peuvent permettre aux sources et aux abonnés de se rencontrer sans inonder le réseau.

Pour des applications de visioconférence l'expérience montre que les récepteurs sont peu nombreux et souvent très espacés, c'est pourquoi le protocole PIM SM est le plus adapté et c'est ce protocole qui est utilisé dans le M6Bone, le réseau expérimental IPv6 multicast.

Dans un contexte multipoints il est nécessaire de disposer d'une entité qui recense les différents participants à la session et qui centralise les informations relatives à la session. Le principe du multicast est donc de déplacer du niveau applicatif au niveau réseau l'entité centralisant les informations relatives à une session.

Ainsi les concepteurs d'applications multipoints peuvent se concentrer sur les caractéristiques intrinsèques des logiciels sans avoir à développer à chaque fois les mêmes mécanismes de gestion de session.

C. Le M6Bone

1. Présentation

Le M6Bone (*Multicast IPv6 Backbone*, épine dorsale multicast IPv6) est un réseau multicast IPv6 expérimental.

Son déploiement a débuté en juin 2001 avec le concours de l'association Aristote, du G6 (groupe d'experts français d'IPv6) et du GIP Renater.

L'objet de ce déploiement est d'offrir aux sites intéressés une connectivité multicast IPv6 afin de tester et de développer les applications et les équipements relatifs à ce protocole.

Enfin le but final de ce projet est de déployer un service avancé du protocole IPv6 afin de participer à la normalisation de ce protocole.

2. Topologie du M6Bone

A l'heure actuelle plus de 25 sites sont reliés au M6Bone, parmi lesquels des partenaires du projet ATHENA comme l'ESMT Dakar, l'université de Guadalajara ou bien encore le DESS Applications des Réseaux et de la Télématique à l'université Paris 7.

Etant donné que peu d'équipements implémentent aujourd'hui le multicast en IPv6, les différents sites composant le M6Bone sont reliés entre eux par des tunnels IPv6 dans IPv6 ou bien IPv6 dans IPv4, c'est-à-dire que chaque paquet multicast IPv6 est encapsulé dans un paquet IPv4 unicast ou bien dans un paquet IPv6 unicast, le choix du tunnel dépendant de la connectivité disponible. Ce mécanisme permet donc de faire transiter du trafic multicast en IPv6 par des équipements réseau qui ne supportent pas ce protocole.

Le cœur du M6Bone est hébergé au GIP Renater, il s'agit d'une distribution FreeBSD installée sur un pentium III qui joue le rôle de RP (*Rendez-vous Point*, point de rendez-vous).

Voici les cartes du M6Bone au niveau national, européen et mondial :

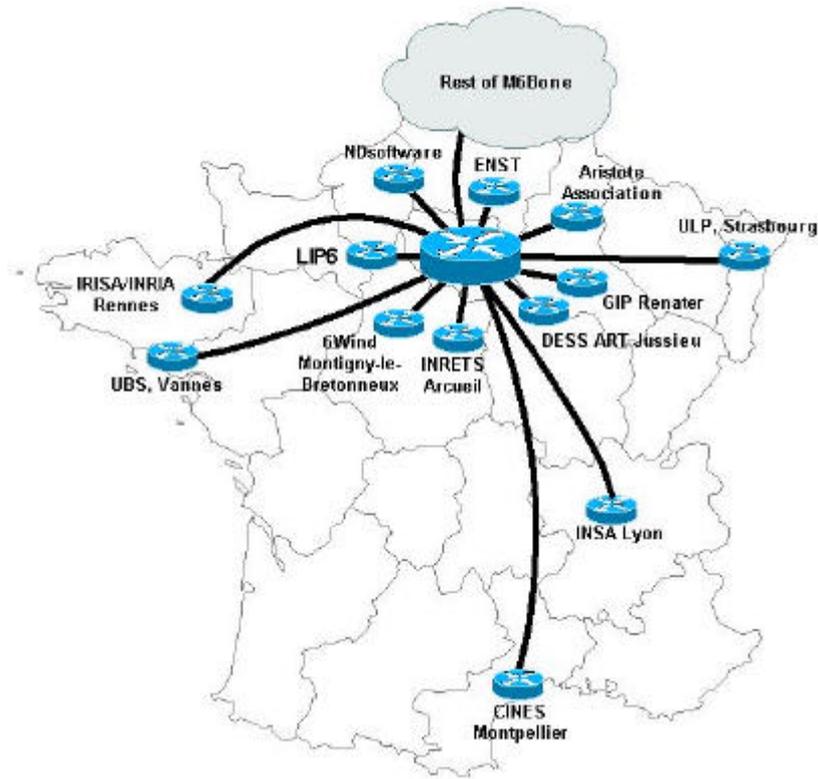


Figure 6 : Carte du M6Bone à l'échelle nationale.

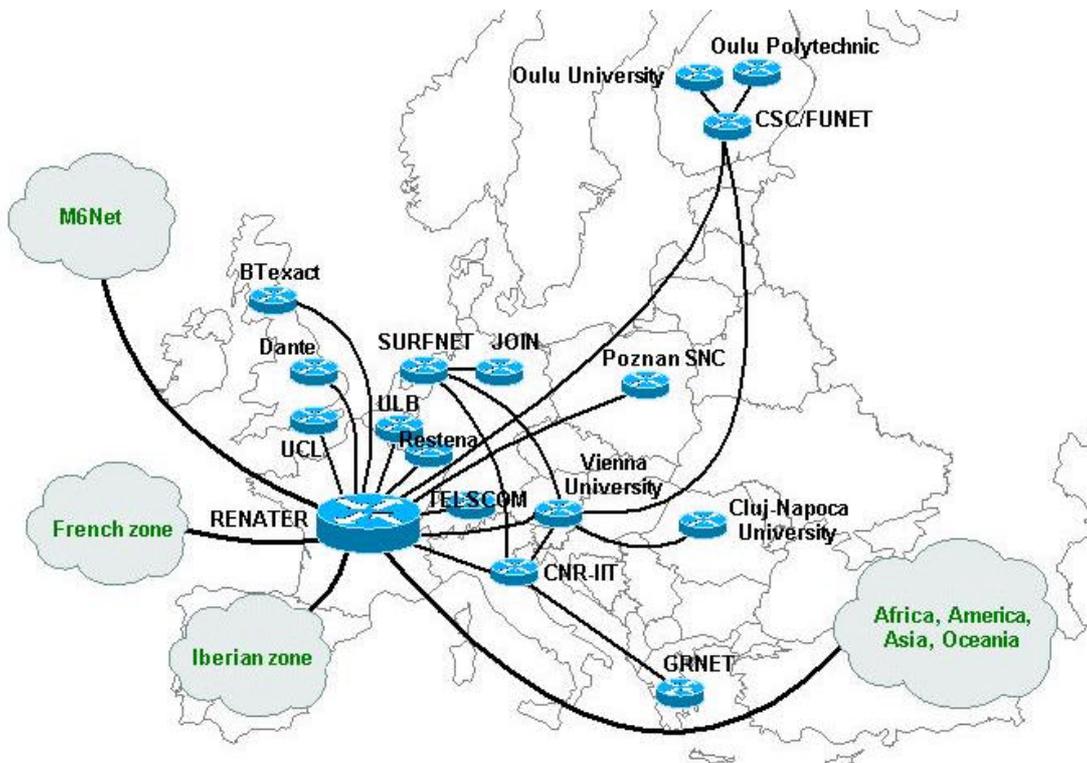


Figure 7 : Carte du M6Bone à l'échelle européenne.

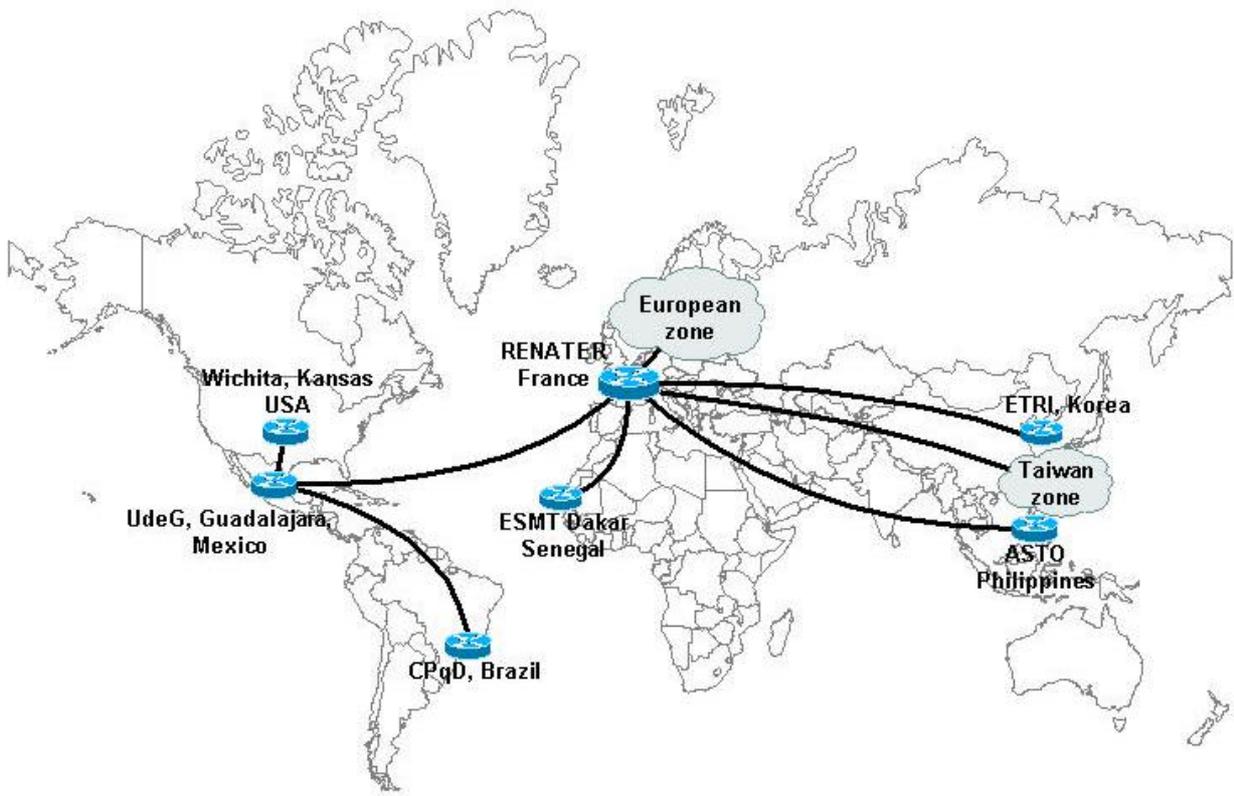


Figure 8 : Carte du M6Bone à l'échelle mondiale.

D. L'interface de programmation d'applications en multicast IPv6 en C/C++ sous Unix/Linux

Afin de comprendre le code source de certaines applications utilisant le multicast IPv6 j'ai été amené à me documenter au sujet des fonctions réseaux utilisées par celles-ci.

Cette section a donc pour objet de présenter l'interface de programmation d'applications en multicast IPv6 pour les environnements Unix/Linux.

1. L'interface de programmation d'applications en IPv6

Les constantes suivantes ont été définies :

Famille de protocoles (*Protocol Family*) : PF_INET6

Famille d'adresses (*Address Family*) : AF_INET6

Voici les structures de données propres au protocole IPv6:

Structure de la socket: sockaddr_in6

```
struct sockaddr_in6 {
    u_short sin6_family;      /* AF_INET6 */
    u_short sin6_port;       /* Transport layer port # */
    u_long sin6_flowinfo;    /* IPv6 flow label */
    struct in_addr6 sin6_addr; /* IPv6 address */
};
```

Structure de l'adresse: in6_addr

```
struct in6_addr {
    u_char s6_addr[16];      /* IPv6 address */
};
```

2. L'interface de programmation d'applications en multicast IPv6

Pour créer et initialiser des sockets on procède de manière similaire qu'en unicast, c'est-à-dire à l'aide de l'appel système `socket()`.

// Déclaration de la socket

```
struct sockaddr_in6 s;
```

// Création de la socket

```

s = socket(PF_INET6, SOCK_DGRAM, 0);
/*
 * Initialisation de la socket
 */

// Réserve d'espace en mémoire pour la socket

memset(&sock_w, 0, sizeof(sock_w));

// La famille de protocole de la socket est initialisée avec la constante représentant la famille
du protocole IPv6

s.sin6_family = AF_INET6;

// On initialise la valeur du port que va utiliser la socket avec la valeur désirée, ici "2222"

s.sin6_port = htons(2222);

// On indique à la socket l'adresse IP du groupe multicast qu'elle va utiliser

inet_pton(AF_INET6, "ff0e:2222", s.sin6_addr.s6_addr);

// A présent on peut spécifier certains paramètres à la socket à l'aide de l'appel système
setsockopt()

Par défaut le Hop Limit des paquets qui seront émis par la socket est à un, on peut modifier
cette valeur par un appel à la fonction setsockopt() :

unsigned int hlim = 255; /* valeurs possibles: 0 à 255 */

setsockopt( s, IPPROTO_IPV6, IPV6_MULTICAST_HOPS, &hlim, sizeof(hlim) );

```

Réception d'un paquet multicast

Avant qu'une machine puisse recevoir des paquets multicast, elle doit être membre du groupe. L'application peut demander à la machine sur laquelle elle s'exécute de rejoindre le groupe multicast en utilisant l'option socket `IP6_ADD_MEMBERSHIP` :

```

if (setsockopt(sdr, IPPROTO_IPV6, IP6_ADD_MEMBERSHIP, (void *) &imr,
              sizeof(struct ip_mreq)) < 0) {
    perror("setsockopt - IP6_ADD_MEMBERSHIP");
    exit(1);
}

```

où `imr` est une structure de type `ip6_mreq`, définie dans le fichier `/usr/include/netinet/in.h`, qui est décrite comme suit :

```

struct ip6_mreq {
    struct in6_addr imr_multiaddr; /* groupe multicast auquel s'abonner */
    struct in6_addr imr_interface;
}

```

```

/*

```

```
* allocation de la structure imr
*/
```

```
struct ip6_mreq imr;
```

```
/*
 * initialisation de la structure imr
 */
```

```
imr.imr_multiaddr.s6_addr = inet6_addr(GROUP);
```

```
imr.imr_interface.s6_addr = htonl(INADDR_ANY);
```

En mode réception il est aussi nécessaire d'effectuer une opération *bind()* entre la socket et son descripteur :

```
/*
 * opération bind
 */
```

```
if (bind(sdr, (struct sockaddr *)&sock_r, sizeof(sock_r)) < 0) {
    perror("bind");
    exit(1);
}
```

La réception proprement dite des données se réalisera par un appel à la fonction *recvfrom()* à travers l'interface socket.

```
/*
 * réception des datagrammes
 */
```

```
while (1) {
    cnt = recvfrom(sdr, buf, sizeof(buf), 0, (struct sockaddr *)&sock_r, &len_r);

    if (cnt < 0) {
        perror("recvfrom");
        exit(1);
    }
    else if (cnt == 0) { /* fin de transmission */
        break;
    }
    printf("%s\n", buf); /* affichage du message */
}
```

Emission de paquets multicast

```
for(;;)  
{
```

```
// La fonction sendto permet de procéder à l'envoi d'un datagramme dont les données sont  
celles contenues dans le tableau de caractères « buf »
```

```
cnt = sendto(sdw, buf, strlen(buf), 0, (struct sockaddr *) &sock_w, len_w);
```

```
if (cnt < 0) {  
    perror("sendto");  
    exit(1);  
}  
}
```

Ainsi on peut constater que l'interface de programmation d'applications en multicast IPv6 diffère peu de celle utilisée pour le multicast IPv4 car si les constantes et les structures de données changent les fonctions restent les mêmes.

E. Les protocoles de transport de trafic temps réel, RTP et RTCP

Les applications en temps réel n'ont pas les mêmes besoins que les applications de transfert de données. En effet, lors de la diffusion d'une vidéo sur IP, si certains paquets se perdent il est inutile de les émettre à nouveau et la caractéristique "temps réel" de ce type d'application est due au fait qu'il est ici plus important de recevoir les données au bon moment que de recevoir toutes les données.

Un protocole de transport a donc été défini pour cette tâche : RTP (*Real Time Protocol*, protocole temps réel). Celui-ci fait partie des protocoles qu'on appelle de "nouvelle génération" car adapté tout spécialement à un type d'application : la diffusion d'informations multimédia interactives sur l'internet.

Ainsi RTP propose des services particuliers comme le transport d'information compressée sous différents formats, le numérotage d'une séquence de paquets, l'estampillage et une surveillance de bonne réception de données multimédia.

Quant à RTCP, il s'agit d'un protocole destiné à assurer le contrôle du trafic. Il permet à l'émetteur de récolter des informations au sujet de la qualité de la transmission.

Chacun de ces deux protocoles utilise un port séparé d'une paire de ports. RTP utilise le port pair et RTCP le port impair immédiatement supérieur.

1. RTP

RTP est un protocole de transport de données multimédia ou temps réel. Il ne fournit aucune garantie de qualité de service et se place généralement au dessus d'UDP¹ pour fonctionner.

Ainsi RTP ne garantit pas une diffusion ordonnée des données mais aide l'application à reconstruire une séquence de données multimédia grâce à ses fonctions de numérotage et d'estampillage.

Il est important de savoir que RTP est destiné à transporter un seul type d'information à la fois, par exemple lors de la diffusion d'une vidéo, la vidéo et le son utiliseront deux flux de données RTP différents et le multiplexage d'informations diverses devra plutôt être assuré par un protocole sous-jacent (le port destination dans UDP par exemple). Voici quelques définitions qui permettront de mieux comprendre ce protocole.

- **Données RTP** : Il s'agit des informations transportées par le protocole, par exemple du son ou de la vidéo encodée de différentes manières.
- **Paquet RTP** : Un paquet RTP contient une entête fixe RTP, une liste pouvant être vide des sources participant au paquet et les données RTP. Typiquement un paquet du protocole sous-jacent (UDP) contient un seul paquet RTP.
- **Port** : L'abstraction permettant à un protocole de transport de différencier des destinations multiples au sein d'un même ordinateur.
- **SSRC** : "Synchronization source", il s'agit de l'identifiant unique d'un flux de données RTP afin d'être indépendant de l'adresse réseau du protocole sous-jacent.
- **CSRC** : "Contributing source", certaines applications qui sont appelées des mixers combinent plusieurs flux de données RTP en un seul, pour savoir d'où proviennent

¹ *User Datagram Protocol*, protocole de datagrammes en mode utilisateur.

les différents flux, une liste de sources ayant contribué au flux est insérée dans l'en-tête du paquet RTP.

Le format d'un paquet RTP est le suivant :

V=2	P	CC	M	PT	Numéro de séquence
Estampille					
Identificateur SSRC					
Identificateurs CSRC					
...					
Données					
...					

Chaque ligne fait 32 bits, les 12 premiers octets sont toujours présents dans un paquet RTP alors que la liste des CSRC est seulement présente si elle a été insérée par un mixer. La signification des différents champs est la suivante :

- **Version (V) : 2 bits**
Identifie la version de RTP, la dernière étant la numéro 2. L'utilitaire vat (ancêtre de Vic) utilisait la version 0 de RTP et le premier *draft* (brouillon) sur RTP définissait la version 1 de RTP.
- **padding (P) : 1 bit**
Si ce bit est posé alors le paquet RTP contient un ou plusieurs octets de remplissage à la fin du paquet qui ne font pas partie des données, le dernier octet du *padding* (bourrage) indique la taille de celui-ci en octet.
- **extension (X) : 1 bit**
Si ce bit est posé alors l'entête fixe RTP est suivie d'une entête d'extension qui peut être définie par l'utilisateur.
- **CSRC Count (CC) : 4 bits**
Indique le nombre d'identifiants CSRC qui suivent l'entête fixe RTP.
- **marker (M) : 1 bit**
L'interprétation de ce bit est laissée à l'application, par exemple elle peut servir à définir le moment où l'on change d'image lors de la diffusion d'une vidéo
- **payload type (PT) : 7 bits**
Indique le format des données RTP et donc leur interprétation par l'application.
- **Numéro de séquence : 16 bits**
Ce numéro est incrémenté de un à chaque paquets de données RTP envoyé, il peut être utilisé pour détecter des pertes ou reconstruire une séquence de paquets.
- **Estampille : 32 bits**
L'estampille contenue dans le paquet RTP indique l'instant auquel le premier octet de données a été encapsulé dans le paquet RTP. Elle doit être dérivée d'une horloge et doit augmenter de façon monotone et linéaire pour permettre une synchronisation.
- **SSRC : 32 bits**
Identifie de façon unique la source sur laquelle il faut se synchroniser.
- **liste de CSRC : de 0 à 15 éléments de 32 bits chacun**
le nombre de CSRC est donné par le champs CC. les CSRC sont insérés par les mixers.

Le format des paquets RTP doit rester simple et il est déconseillé d'essayer de faire du multiplexage sur les différents formats de données pour les raisons suivantes :

1. Si deux flux de données partagent le même paquet RTP et donc le même identificateur SSRC et qu'il faille changer le type d'encodage de l'un des flux de données il ne sera pas possible d'identifier quel flux de données a changé son encodage
2. Par définition un identificateur SSRC est relatif à un seul espace-temps et une seule numérotation des paquets. Si on multiplexe plusieurs types de données il faudrait disposer de plusieurs temporisations et plusieurs numérotations pour pouvoir identifier le flux de données ayant subi des pertes.
3. Comme on va le voir plus loin, les rapports de réception RTCP sont relatif qu'a un seul timing et une seule numérotation par SSRC et ne précisent par le type de flux de données.
4. Un mixer ne sait mixer qu'un seul type de données.
5. Un paquet RTP contenant plusieurs types de données rendrait difficile : l'utilisation possible de différentes ressources réseau en envoyant par exemple les paquets contenant de la vidéo sur un lien différent de ceux contenant du son, le choix de la réception d'un sous ensemble des données multimédia (seulement le son par exemple)

Ainsi l'utilisation d'une SSRC unique par média permet de remédier aux trois premiers points. On voit que la partie qui transporte les données du protocole RTP est assez simple, en effet toute sa complexité se situe dans le contrôle "temps réel" de ses données.

2. RTCP

RTCP est un protocole qui permet d'obtenir des informations diverses sur une session RTP en cours : nombre de participants, qualité de réception, départ ou arrivée d'un nouveau participant. C'est donc un protocole de Feedback qui se doit d'être extensible par rapport au nombre participants à une session, qu'ils soient dix ou quelques millions.

Ainsi RTCP transmet périodiquement des paquets de contrôle à tous les participants d'une session.

RTCP a quatre rôles :

1. La fonction première de RTCP est de fournir un retour sur la qualité de la distribution des données. Ce retour peut être utile pour l'adaptation dynamique de l'encodage lors d'une transmission et permet par exemple de déterminer si une mauvaise réception est globale ou locale à une session.
2. Le protocole RTCP permet de transporter un identificateur qu'on nomme CNAME pour "canonical name". En effet SSRC peut changer lors d'une session par exemple si l'on se rend compte d'une collision de SSRC alors que CNAME ne change pas.

3. Les deux premières fonctions nécessitent que tous les participants à une session RTP doivent envoyer des paquets RTCP ainsi la bande passante utilisée par les messages RTCP doit être extensible avec le nombre de participants à une session. Si chaque participant a une session envoie des paquets RTCP aux autres, il est possible de connaître le nombre de participants et ensuite de calculer le débit auquel il faut envoyer des messages de contrôle.

4. La dernière fonction de RTCP est optionnelle et permet de fournir des informations de session minimales comme par exemple l'identification des participants à une session. C'est particulièrement utile pour des applications où il n'y a que très peu de contrôle des participants à une session.

Les différents types de paquets RTCP sont les suivants :

- **SR** : *Sender Report* indique les statistiques de réception et d'émission des participants à une session qui envoient des données.
- **RR**: *Receiver Report* indique les statistiques de réception des participants à une session qui n'envoient pas des données et dont les statistiques n'excèdent pas 31 sources.
- **SDES** : les éléments décrivant une source
- **BYE** : indique la fin d'une participation.
- **APP** : fonctions spécifiques à l'application.

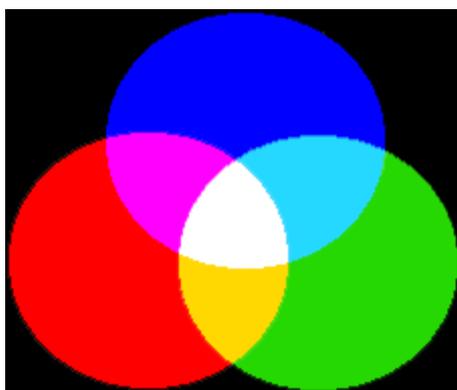
De la même manière que pour le protocole RTP chaque paquet RTCP contient une partie fixe et une partie variable, plusieurs paquets RTCP peuvent être concaténés dans un paquet de la couche inférieure (UDP par exemple) mais le premier paquet RTCP contenu dans un paquet de la couche inférieure doit toujours contenir des informations relatives aux statistiques de transmission ou de réception (paquets SR et RR) même lorsque des données ne sont pas reçues ou transmises.

F. La vidéo numérique

1. Codage de la couleur

Au début de la chaîne de codage un capteur convertit les signaux lumineux en signaux électriques, on obtient alors un premier signal codé en RVB (Rouge, Vert, Bleu).

Cette méthode se base sur le fait que l'on peut recréer une impression physiologique satisfaisante pour l'œil humain, pour n'importe quelle couleur, en superposant du rouge, du vert et du bleu d'intensité adéquate.



Par la suite, afin de réduire la quantité d'information nécessaire pour coder le signal vidéo, on utilise une technique de matricage qui consiste à décomposer le signal vidéo en un signal de luminance¹, Y, et deux signaux de chrominance², un rouge et un bleu, Cr et Cb.

Les composantes de chrominance sont obtenues par différence du rouge avec la luminance ($Cr = R - Y$) et par différence du bleu avec la luminance ($Cb = B - Y$). Le signal de chrominance vert, Cv, est obtenu de la façon suivante : $Cv = Y - (Cb + Cr)$.

Après le matricage une autre opération consiste à n'utiliser qu'une composante de chrominance, C, et une composante de luminance, Y. On obtient le signal de chrominance C par codage des deux composantes différentielles Cr et Cb.

Ce signal est appelé S-Vidéo et c'est lui qui sera utilisé ensuite par les différents standards d'image: PAL (*Phase Alternation Line*, ligne alternée en phase), NTSC (*National Television System Committee*, Comité national du système télévisuel) et SECAM (Système sEquentiel Couleur A Mémoire).

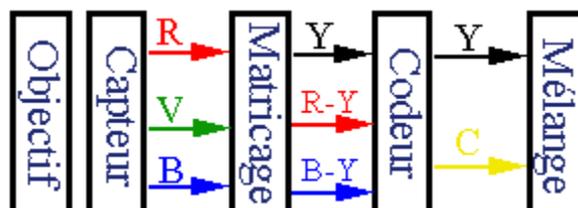


Figure 9 : Les différentes étapes pour coder un signal vidéo.

¹ Intensité lumineuse d'un signal vidéo.

² C'est la couleur d'un signal vidéo, composée de la teinte et de la saturation.

2. Les formats d'images vidéo

Avec l'avènement de la télévision deux standards de formats d'images vidéo sont apparus : le NTSC pour les Etats-Unis et une grande partie de l'Asie et le PAL pour l'Europe ou bien un dérivé du PAL comme le SECAM en France.

Ces deux familles regroupent chacune une diversité de variantes qui ont en commun la fréquence de rafraîchissement et le nombre de lignes qui composent l'image.

Pour des raisons techniques il a été choisi comme base de temps la fréquence du réseau électrique du pays concerné. Ainsi aux Etats-Unis, le secteur étant en 110V à 60Hz, la télévision utilise un rafraîchissement de 60Hz. En France nous sommes en 220V à 50Hz, le SECAM utilise donc un rafraîchissement de 50Hz.

Le standard NTSC propose une image de 480 lignes tandis que le standard PAL propose une image de 576 lignes.

Par la suite, avec l'avènement de la micro-informatique est apparu le format VGA (*Video Graphics Array*, tableau graphique vidéo) proposant une définition de 320x200 en 256 couleurs ou une définition de 640x400 en 16 couleurs, puis le SVGA (*Super VGA*, super VGA) proposant 4 définitions supplémentaires : 800x600, 1024x768, 1280x1024, 1600x1200 ainsi qu'une palette de couleurs allant jusqu'à 16 millions de couleurs.

Enfin avec le développement des applications de vidéoconférence est apparu le besoin de trouver un format commun à toutes ces applications et c'est l'UIT-T¹ qui a défini, en 1990, deux formats couvrant les besoins exprimés dans les domaines de la visiophonie et de la visioconférence : les formats CIF (*Common Intermediate Format*, Format Commun Intermédiaire) et QCIF (*Quarter CIF*, quart de CIF).

Le format d'image CIF est caractérisé par 352x288 points pour la luminance et par 176x144 points pour la chrominance. Le balayage est séquentiel et la fréquence d'image est de 29,97 Hz.

Le format d'image QCIF est caractérisé par 176x144 points pour la luminance et par 88x72 points pour la chrominance, avec une fréquence d'image sous-multiple de celle du format CIF.

Par la suite le format 4CIF est apparu, il correspond au double du format CIF et il est caractérisé par une taille d'image de 704x576 points pour la luminance et par 352x288 points pour la chrominance.

¹ Union Internationale des Télécommunications – Secteur de normalisation des Télécommunications

3. Les formats de compression de signaux vidéo

Afin de rendre compte d'un mouvement fluide il est nécessaire de transmettre au moins 25 images par seconde. Supposons que chaque image ait une dimension de 800x600 et que chaque pixel soit codé sur 8 bits, il faut alors un débit de :

$$25 \times 800 \times 600 \times 8 = 96 \text{ Mbit/s}$$

ce qui correspond au débit de la télévision à l'origine.

Pour la transmission par l'Internet du signal vidéo numérique il est donc nécessaire d'utiliser des techniques de compression afin de réduire le débit nécessaire.

Le principe utilisé est le suivant : la source procède à un encodage du signal vidéo, puis au niveau du récepteur un décodage du signal vidéo est réalisé.

Différents formats de compression ont donc été normalisés, parmi lesquels le fameux MPEG (*Motion Picture Experts Group*, groupe d'experts en images animées), néanmoins l'UIT-T a aussi émis des normes de codage vidéo pour des applications de visiophonie et de vidéoconférence comme la norme H.261 qui s'applique à des services audiovisuels utilisant des liaisons de $p \times 64$ Kbit/s (la norme H.261 est donc tout particulièrement adaptée pour des services audiovisuels utilisant des liaisons RNIS¹) ou la norme H.263 qui vise à transférer de la vidéo sur des liaisons bas débit.

Bien que les deux normes de codage vidéo diffèrent, leur principe reste le même. Chaque image est découpée en blocs, puis on applique une transformée en cosinus discrète à chacun des blocs pris séparément. La sortie de chaque transformée est alors une matrice de coefficients transformés.

Ces normes de codage permettent donc de transmettre des images animées sur des liaisons à faible débit (moins de 128 Kbit/s).

¹ Réseau Numérique à Intégration de Services

G. Les outils du M6Bone

1. Vic (Video Conference)

a) Présentation

Le logiciel Vic est un logiciel de visioconférence permettant d'émettre et de recevoir de la vidéo en unicast ou en multicast en utilisant différentes normes de codage vidéo comme les normes H261, H263 ou H263+. Ce logiciel est disponible pour Windows 98, 2000 et XP ainsi que pour les systèmes d'exploitation Unix/Linux.

b) Historique

Avec l'apparition du multicast dans les réseaux académiques au début des années 90, des chercheurs ont commencé à développer des outils exploitant les fonctionnalités offertes par ce protocole.

Ainsi des membres de l'INRIA (Institut National de Recherche en Informatique et Automatique) ont-ils développé IVS (*INRIA Videoconferencing System*, système de vidéoconférence de l'INRIA) un logiciel de vidéoconférence dont la première version a été diffusée en 1992.

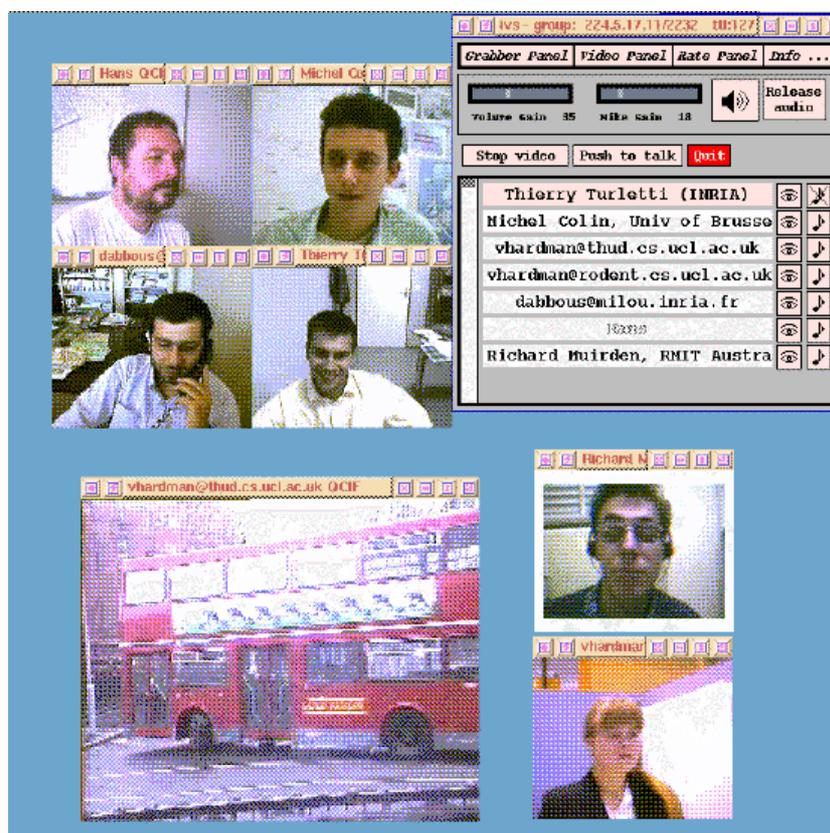


Figure 10 : Aperçu de l'interface du logiciel IVS.

Inspiré par le logiciel IVS, Vic a été développé dans les années 90 par le groupe de recherche Multimédia et Réseaux (*Network and Multimedia Research Group*) de UCL (*University College of London*).

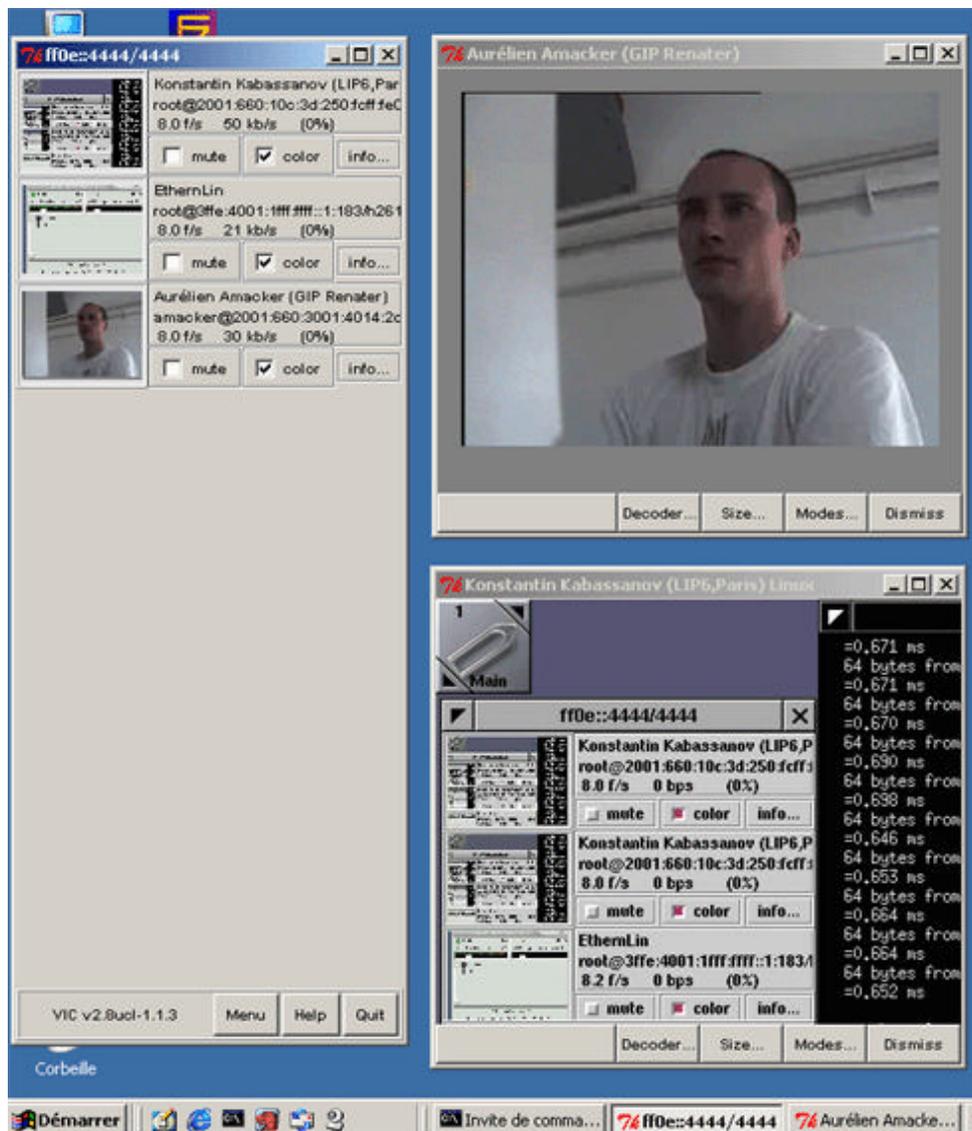


Figure 11: Interface du logiciel Vic.

Sur la figure 11 on remarque sur la gauche l'interface principale du logiciel Vic dans laquelle on peut voir tous les membres de la session qui émettent. En bas de l'interface principale se trouve le bouton « Menu » qui permet notamment de spécifier les paramètres d'émission de la vidéo. En cliquant sur l'image d'un des participants on peut agrandir sa fenêtre comme on peut le voir sur la droite de l'image. La fenêtre en bas à droite présente une fonctionnalité intéressante de Vic pour procéder à des tests puisque celui-ci permet de transmettre sous Linux une partie de l'écran, ici il s'agit d'une machine Linux de Konstantin Kabassanov au LIP6.

La version de Vic développée par UCL est une des principales versions de ce logiciel et c'est cette version qui est essentiellement utilisée dans la communauté académique en France. D'autres versions ont été développées comme par exemple la version *Access Grid* de Vic

(grille de conférences utilisée par des universitaires aux Etats-Unis) afin de proposer des fonctionnalités particulières.

Pour des raisons budgétaires UCL a dû se désengager du développement de Vic, c'est pourquoi ce logiciel n'évolue presque plus alors même qu'il reste des modifications à apporter et des bogues à corriger, cependant comme ils nous l'ont affirmé lors d'une conférence téléphonique ils sont tout à fait prêts à incorporer dans leur version des modifications issues de la communauté académique.

Dans les années 90 Steve McCanne, un étudiant de UCL ayant contribué au développement de Vic, est parti préparer sa thèse à l'université de Berkeley en Californie et a initié le projet OpenMash, ce projet visant à développer des outils de visioconférence utilisant le multicast comme ceux ayant été développés par le groupe de recherche Multimédia et Réseaux de UCL.

Le projet OpenMash a donc développé sa propre version de Vic en reprenant une grande partie du code de la version initiale. Ces deux versions du logiciel Vic constituent les deux principales versions de ce logiciel et elles sont compatibles entre elles, c'est-à-dire qu'on peut organiser une vidéoconférence impliquant des participants utilisant la version UCL de Vic et des participants utilisant la version OpenMash de Vic.

Le projet OpenMash propose un site Internet disponible à l'adresse www.openmash.org ainsi que des listes de diffusion permettant à la communauté des utilisateurs et des développeurs de leurs logiciels de partager leurs connaissances et de contribuer à leur développement.

Il y a deux ans, M. Konstantin Kabassanov, un chercheur du LIP6 (Laboratoire d'Informatique de l'université Paris 6), a porté la version UCL de Vic pour IPv6. Cette version est disponible sur son site personnel à l'adresse www.kabassanov.com.

En ce qui concerne la version OpenMash de Vic, une adaptation de ce logiciel pour IPv6 a été réalisée il y a deux ans par M. David Pate, un universitaire français de l'université de Strasbourg mais ses modifications n'ont pas été intégrées à la distribution officielle du Vic d'OpenMash, la contribution au développement d'IPv6 n'étant pas considérée comme une priorité par la plupart des membres de la communauté académique aux Etats-Unis.

c) Utilisation

Pour lancer Vic en multicast la syntaxe est (en ligne de commande) :

`vic -t 64 adresse/port`

Où « adresse » est l'adresse multicast du groupe sur lequel on veut émettre et « port » le numéro de port du groupe sur lequel on veut émettre.

Le paramètre « -t » est optionnel, il permet de spécifier le *hop limit* (nombre maximum de nœuds du réseau que peuvent traverser les données) qui est à 16 par défaut. Limiter la valeur du *hop limit* permet de conscrire la portée d'une émission à une certaine zone géographique.

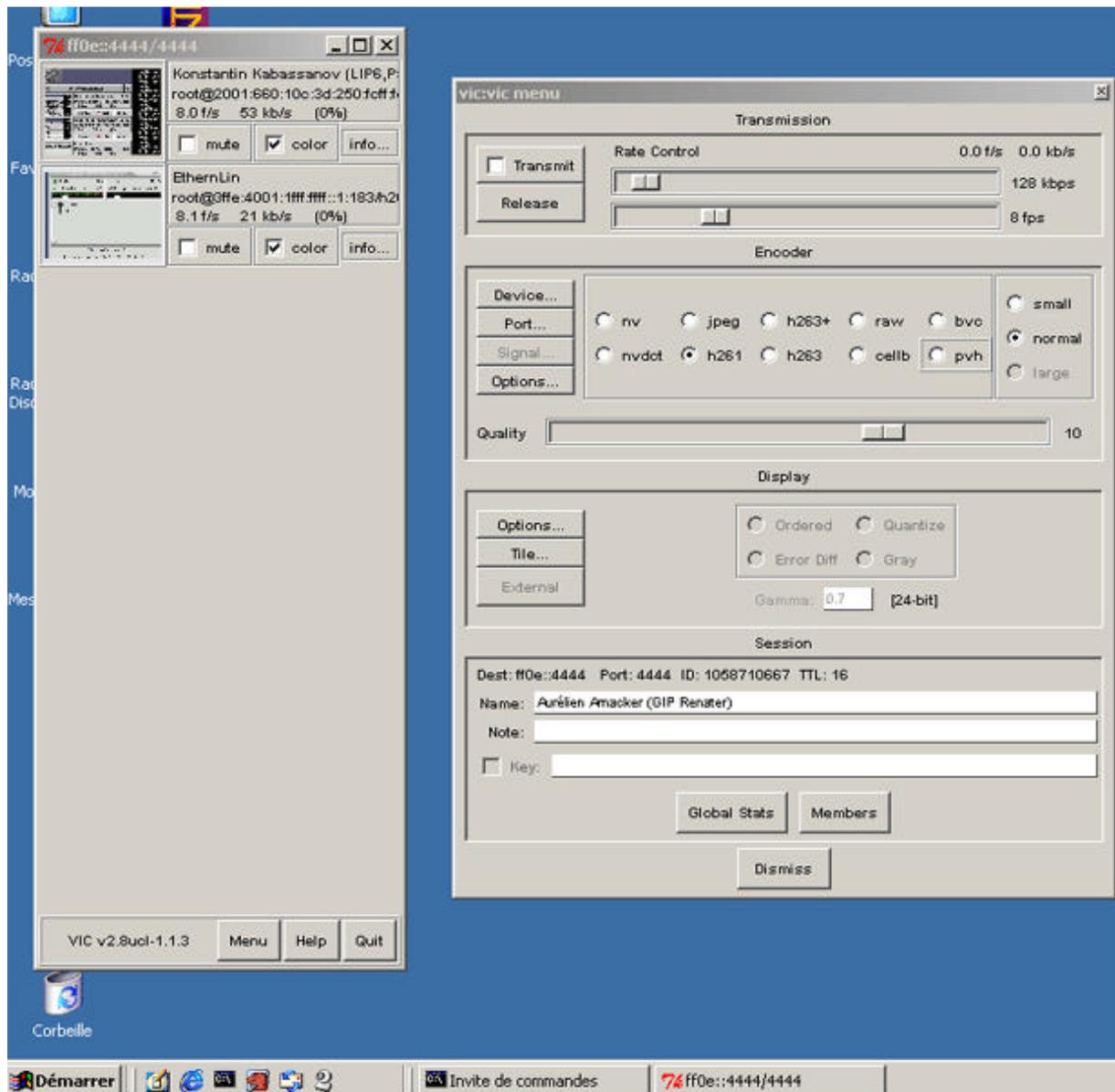


Figure 12 : Menu de configuration de la transmission.

Sur la figure 12 on peut voir sur la droite le menu permettant de renseigner les paramètres d'émission de la vidéo. Les deux barres de défilement en haut permettent de spécifier le débit utilisé en Kbit/s et le nombre d'images par seconde que l'on veut émettre.

En dessous, dans la partie « Encoder », on spécifie le périphérique que l'on veut utiliser ainsi que le format d'encodage et la taille des images qu'on va transmettre.

Pour utiliser Vic en unicast il faut taper la commande suivante :

`vic adresse/port`

Où « adresse » désigne l'adresse IP de la personne avec laquelle on veut faire une visioconférence et « port » un numéro de port.

On remarque que contrairement à l'utilisation de Vic en multicast les deux participants à la visioconférence lancent Vic avec chacun une adresse différente (celle de leur correspondant) en paramètre.

Le numéro de port utilisé doit par contre être le même pour les deux personnes.

Dans le cas de l'utilisation de Vic en unicast seules deux personnes peuvent participer à la visioconférence, le multicast constituant l'entité logicielle permettant de répertorier l'ensemble des participants d'une même session.

L'utilisation de Vic en unicast est surtout utilisée à des fins de tests, ou bien encore pour organiser une visioconférence avec une personne qui ne dispose pas du multicast sur son réseau.

d) Fonctionnement interne du logiciel

Vic est un logiciel écrit en Tcl/Tk¹ pour ce qui concerne la partie interface utilisateur et en C++ pour ce qui concerne la partie traitement de l'information. Pour des questions de portabilité la version de Tcl/Tk utilisée par Vic est une version fournie par UCL. Vic utilise également la librairie *common* développée par UCL et qui fournit notamment des fonctions réseau de bas niveau.

(1) Interactions entre la partie Tcl/Tk et la partie C++

L'interaction entre l'interface utilisateur écrite en Tcl/TK et la partie C++ se fait de la façon suivante : dans certains scripts Tcl, des objets C++ sont créés et sont représentés par des variables. Par la suite les scripts Tcl peuvent envoyer des commandes aux objets C++ afin d'influer sur leur comportement.

Ces objets C++ qui peuvent être manipulés dans des scripts Tcl sont tous des objets dont la classe hérite de la classe TclObject. Toutes les classes héritées de la classe TclObject héritent de la fonction « command » qu'ils doivent redéfinir, et c'est cette fonction qui permet d'envoyer des commandes aux objets C++ depuis des scripts Tcl.

Par exemple dans le script `tcl/ui-ctrlmenu.tcl` qui est appelé lorsque dans l'interface de Vic on clique sur le bouton « Menu » afin de renseigner les paramètres d'une émission vidéo on a la ligne suivante :

```
set V(grabber) [$videoDevice open $ff]
```

Cette ligne se décompose en deux parties : tout d'abord la partie entre crochets est interprétée par l'interpréteur Tcl. `$videoDevice` est une variable Tcl qui représente un objet C++ dont la classe hérite de la classe `InputDevice` (définie dans le fichier `video/device-input.cpp`) selon le périphérique vidéo qu'on aura choisi et le système d'exploitation utilisé. Par exemple dans le cas de l'utilisation d'une caméra et d'une carte d'acquisition avec un système d'exploitation Linux, l'objet C++ représenté par la variable `$videoDevice` sera de type `v4lDevice` (`v4l` est l'acronyme de `video 4 linux`, l'API² de capture d'images sous Linux).

¹ Tcl est un langage de scripts tandis que Tk (Tool Kit, *boîte à outils*) est une librairie graphique. Dans des programmes Tcl/Tk, les scripts Tcl appellent des *widgets Tk* (des composants graphiques) ce qui permet de créer rapidement des applications disposant d'interface graphique.

² *Application Programming Interface*, Interface de programmation d'applications

L'instruction « `$videoDevice open $ff` » signifie que la méthode « `command` » de l'objet `$videoDevice` est appelée avec en paramètres les arguments « `open` » et « `$ff` ». Si on regarde le code de la fonction « `command` » de la classe `v4lDevice` on peut voir que si le premier argument est « `open` », cela signifie qu'une ouverture sur le périphérique vidéo est réalisée avec une résolution égale à `$ff` (`ff = frame format`, format d'image), de plus la valeur de retour de la fonction « `command` » est dans ce cas un objet de type `v4lGrabber`. Le résultat de l'évaluation de la partie entre crochets est donc un objet C++ de type `v4lGrabber` et, dans notre exemple, la première partie de l'instruction permet donc d'affecter à la variable `$V(Grabber)` un objet C++ de type `v4lGrabber`.

De cette manière les scripts Tcl qui constituent l'interface graphique de Vic peuvent commander les structures de données qui procèdent au traitement de l'information nécessaire pour effectuer les captures d'images, l'encodage et le décodage de la vidéo ainsi que le découpage de l'information, l'encapsulation dans des paquets IP et l'envoi de ces paquets sur le réseau.

(2) Code source de Vic

Le code source du logiciel Vic se décompose en quatre répertoires.

Le premier s'appelle « `common` », il contient les fonctions de la librairie *common*.

Le deuxième répertoire s'appelle « `tcl-8.0` », il contient les sources de l'interpréteur Tcl¹ fourni avec Vic et le troisième répertoire s'appelle « `tk-8.0` » et contient les sources de la librairie graphique Tk. L'interpréteur Tcl ainsi que la librairie graphique Tk sont intégrés à la version UCL du logiciel Vic en raison du fait qu'au cours du développement de Vic des problèmes se sont posés car les utilisateurs avaient des versions différentes de Tcl et de Tk ce qui a conduit le laboratoire de UCL à intégrer ces deux éléments logiciels à leur distribution de Vic.

Enfin le dernier répertoire s'appelle « `vic` » et contient les sources des fichiers Tcl et C++ qui composent le coeur du logiciel Vic.

Lorsqu'on veut apporter des modifications au logiciel Vic, la plupart du temps il suffit de modifier les fichiers sources contenus dans le répertoire Vic, c'est pourquoi je vais décrire brièvement les principaux éléments de ce répertoire.

Le fichier `main.cpp` constitue le fichier principal du programme vic car c'est celui qui contient la fonction `main()` qui est la première fonction appelée lors de l'exécution de Vic.

Le répertoire `codec/` contient les encodeurs et les décodeurs vidéo, les principaux formats de compression utilisés par Vic étant les formats `h261`, `h263` et `h263+`.

Le répertoire `tcl/` contient tous les scripts tcl qui sont appelés par le programme Vic.

Le répertoire `net/` contient les fonctions réseaux de bas niveau permettant d'envoyer et de recevoir des paquets IPv4 ou IPv6 en multicast ou en unicast.

Le répertoire `rtp/` contient les fichiers source nécessaires quant à l'utilisation du protocole de transport RTP, ainsi que les définitions de différentes structures de données utilisées lors de la transmission de la vidéo.

Enfin le répertoire `video/` contient les définitions de structures de données permettant de réaliser les captures d'images sur le périphérique vidéo.

¹ Un interpréteur est un logiciel qui lit un script et qui exécute les instructions de ce script, ainsi tout script nécessite un interpréteur pour s'exécuter.

2. Rat (*Robust Audio Tool*, outil audio robuste)

Rat est au son ce que Vic est à la vidéo, c'est à dire qu'il permet d'envoyer et de recevoir du son en multicast ou en unicast.

Pour répondre aux contraintes temps réel Rat utilise comme Vic le protocole RTP au-dessus d'UDP.

Rat a été développé par le groupe de recherche Multimédia et Réseaux de UCL tout comme Vic, mais à la différence de Vic, UCL continue le développement et le support de ce logiciel. Deux versions de Rat sont actuellement disponibles sur le site Internet de l'UCL, la première, la version 3, est la version stable. Elle propose une qualité de son moyenne avec un échantillonnage à 8 KHz et n'est pas compatible avec IPv6 tandis que la seconde version, la version 4, est la version de test et propose un son de haute qualité avec un échantillonnage allant jusqu'à 48 KHz et une compatibilité avec IPv6.

Toutes ces versions sont disponibles pour Windows et Unix/Linux.

Voici un aperçu de l'interface de ce logiciel :

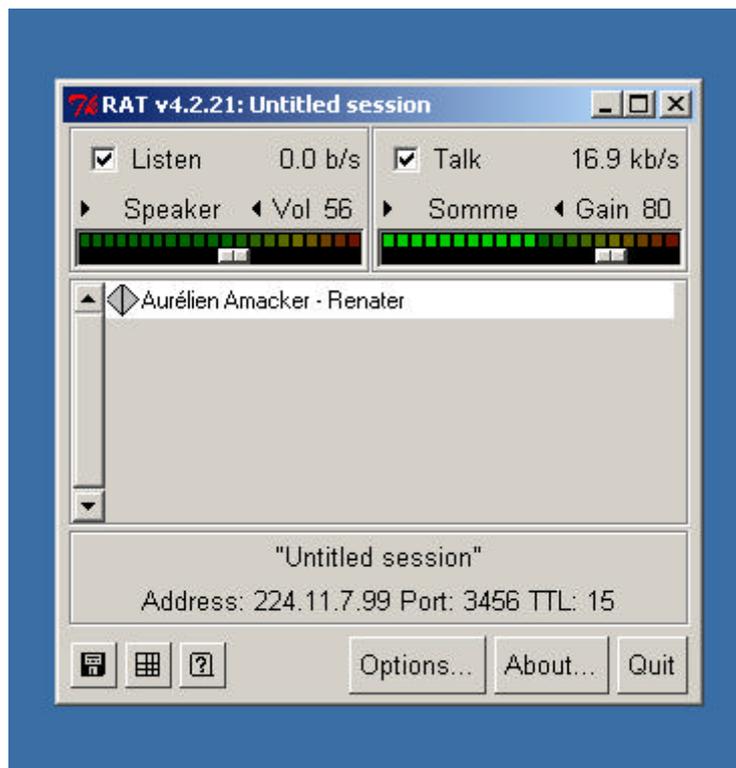


Figure 13 : Interface graphique du logiciel Rat.

Rat s'utilise de manière similaire à Vic :

En multicast :

[rat adresse/port](#)

où « adresse » désigne l'adresse du groupe multicast à destination duquel on désire émettre et « port » le port du groupe multicast.

En unicast :

[rat adresse_correspondant/port](#)

où « adresse_correspondant » désigne l'adresse de la personne avec laquelle on désire faire une vidéoconférence et « port » un port de la machine.

IV. Mises en oeuvre

A. Poursuite du projet ATHENA

1. Support technique auprès des partenaires du projet ATHENA

Au début de mon apprentissage, mon prédécesseur au GIP Renater en charge du projet ATHENA, M. Lionel David, m'avait mis en relation avec M. Netzahualcoyotl Ornelas García, un responsable réseau à l'Université de Guadalajara au Mexique.

Etant en charge de la connectivité IPv6 de son département à l'Université de Guadalajara, il était désireux de connecter son réseau au M6Bone afin de pouvoir bénéficier du multicast IPv6, en effet si son réseau disposait d'IPv6 unicast grâce à un tunnel établi avec le CERN en Suisse, il ne disposait ni du multicast IPv4 ni du multicast IPv6.

J'ai donc commencé par l'orienter vers le site www.m6bone.net qui est maintenu par l'équipe IPv6 du GIP Renater et sur lequel se trouvent les informations techniques nécessaires pour se connecter.

Par la suite et en communiquant par messagerie instantanée j'ai pu l'aider à établir un tunnel entre son routeur multicast IPv6 et le routeur du GIP Renater qui constitue le coeur du M6Bone.

Tout au long de l'année les membres de l'Université de Guadalajara se sont montrés très actifs dans le cadre du déploiement du multicast IPv6 puisque par la suite à leur tour ils ont aidé d'autres universités au Mexique à se raccorder au M6Bone.

A l'occasion de la conférence IPv6 à Caen nous avons eu le plaisir de recevoir M. Netzahualcoyotl Ornelas García qui a effectué une présentation à laquelle ont assisté à distance des membres de l'équipe IPv6 de l'Université de Guadalajara grâce à la retransmission de la conférence en multicast IPv6.

2. Maintien du contact avec les partenaires du projet ATHENA

En septembre 2002 nous avons reçu au GIP Renater Redouane Merrouch qui est l'administrateur du MARWAN, ce qui nous a permis de faire le point sur la situation des réseaux de la recherche au Maroc.

Au cours de cette réunion M. Merrouch nous a fait part de son désir de sensibiliser les membres du gouvernement au besoin de développer les réseaux de la recherche au Maroc.

Les capacités actuelles en terme de débit du réseau MARWAN ne nous permettent pas encore de réaliser des expériences concrètes avec les membres de la communauté académique marocaine, néanmoins l'objectif est de garder le contact en prévision du futur.

M. Merrouch nous a également présenté le projet EUMEDCONNECT auquel participe le MARWAN et qui vise à développer les réseaux de la recherche dans les pays du pourtour de la Méditerranée.

Concernant le Mexique, nous avons reçu au GIP Renater au début de l'année deux ingénieurs de l'Université de Guadalajara, Arturo Gómez García et León Felipe Rodríguez

Jacinto, qui se rendaient à une conférence en Espagne et qui ont fait un détour par Paris pour nous rendre visite. Ceci nous a permis de les recevoir et de renforcer notre contact avec cette université dont le personnel est manifestement très motivé et compétent.

Deux fois par an une conférence est organisée sur le thème d'IPv6 par l'équipe IPv6 du GIP Renater, et à chacune de ces occasions j'ai été chargé de prendre contact avec les partenaires du projet ATHENA afin de les inviter, mais aussi de les aider à trouver un hébergement, ou bien encore de leur proposer un financement de leur voyage par l'association Aristote, en effet il est très important de maintenir les relations avec les partenaires du projet ATHENA et les conférences IPv6 constituent une très bonne occasion de se réunir régulièrement.

B. Mise en oeuvre de la passerelle IPv4 multicast – IPv6 multicast

Le jour même de mon arrivée au GIP Renater j'ai été amené à installer sur une machine dédiée (un FreeBSD) la passerelle IPv4 multicast – IPv6 multicast développée par Luc Beurton à l'université Bretagne Sud.

Cette passerelle s'inscrit parmi les mécanismes de transition de IPv4 vers IPv6 puisqu'elle permet au monde multicast IPv4 et au monde multicast IPv6 d'interopérer.

Son mécanisme est le suivant : on lance la passerelle avec en paramètres les adresses d'un groupe multicast IPv4 et d'un groupe multicast IPv6.

La passerelle s'abonne alors au groupe multicast IPv4 et réémet toutes les données qu'elle reçoit en provenance de ce groupe à destination de l'adresse du groupe IPv6 multicast. Inversement, toutes les données qu'elle reçoit du groupe multicast IPv6 sont réémises à destination du groupe multicast IPv4.

Ceci permet donc aux membres du groupe multicast IPv4 de recevoir les données provenant du groupe multicast IPv6 et de pouvoir émettre à destination de ce groupe et donc inversement de permettre aux membres du groupe multicast IPv6 de recevoir les données provenant du groupe multicast IPv4 et de pouvoir émettre à destination de ce groupe.

Par exemple si on veut passereller le groupe multicast IPv4 « 224.11.7.99/3456 » avec le groupe multicast « ff0e::2222/3456 » on va lancer la commande suivante :

```
mforward -4 224.11.7.99/3456 -6 ff0e::2222 -p 3456
```

De cette manière tous les membres du groupe multicast 224.11.7.99/3456 verront les membres du groupe multicast ff0e::2222/3456 et inversement.

Le port utilisé doit être le même pour le groupe multicast IPv4 et pour le groupe multicast IPv6. Ceci n'est pas une contrainte puisque par exemple si on diffuse de la vidéo sur un groupe multicast IPv4 et qu'on désire passereller cette vidéo vers un groupe multicast IPv6 , il suffit de choisir un groupe multicast IPv6 dont le numéro de port soit le même que celui du groupe multicast IPv4

Voici un schéma qui présente le principe de fonctionnement de la passerelle multicast IPv4 – IPv6 :

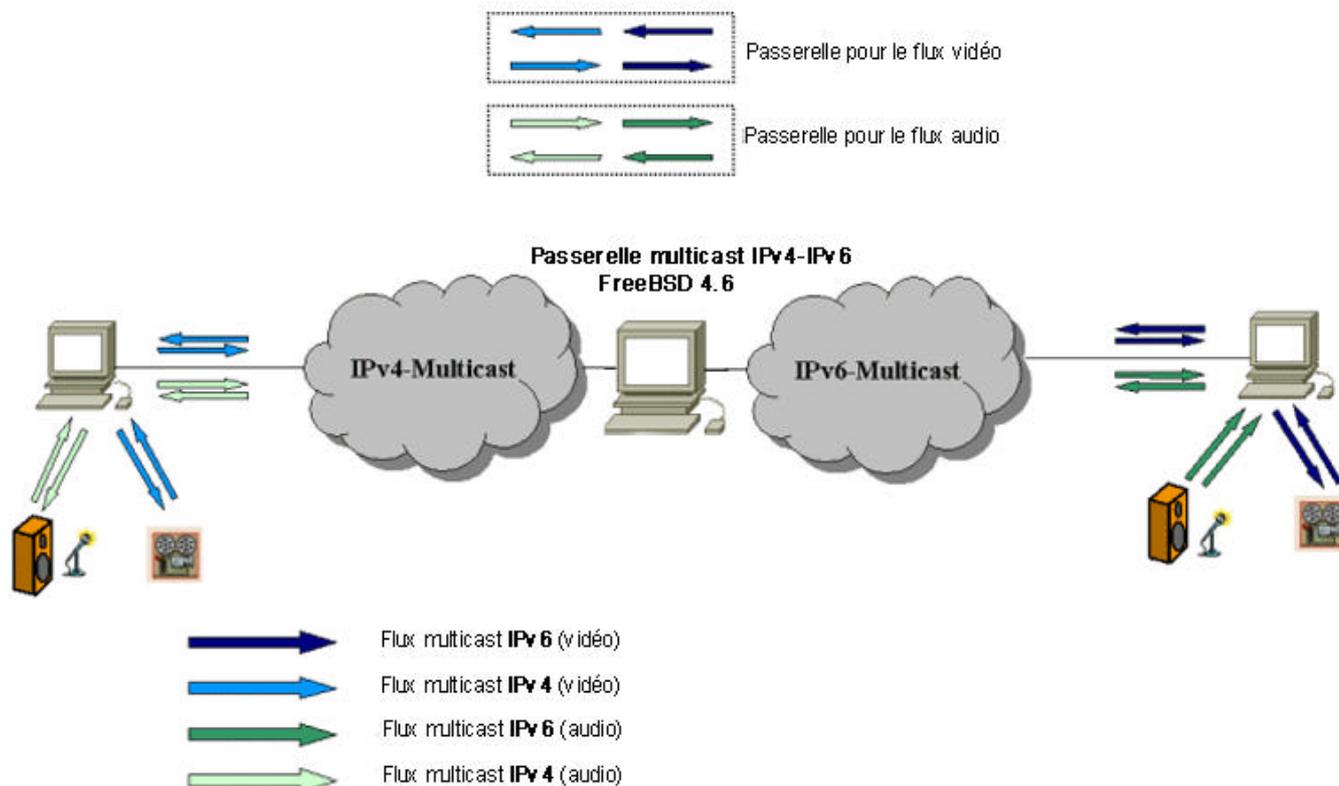


Figure 14 : Principe de fonctionnement de la passerelle multicast IPv4/IPv6.

Bien qu'au niveau réseau il y ait un groupe multicast IPv4 et un groupe multicast IPv6, au niveau utilisateur il n'y a donc qu'une seule communauté logique.

Cette passerelle s'étant révélée satisfaisante dans un environnement de production puisqu'elle avait servi en juin 2002 à passereller en multicast IPv6 la retransmission d'un séminaire X-Aristote diffusé en multicast IPv4, il avait été décidé de l'installer sur une machine du GIP Renater afin de pouvoir offrir ce service à l'ensemble de la communauté académique.

Après le départ en décembre 2003 d'un stagiaire de l'association Aristote responsable de son fonctionnement j'ai été chargé de la maintenance et de l'utilisation de cette passerelle et à chaque fois que nous avons participé à la retransmission d'un événement en multicast IPv4 nous l'avons passerellé en multicast IPv6.

Ceci nous a permis de nous positionner en tant qu'acteurs de la transition du protocole IPv4 vers IPv6, mais aussi de permettre à la communauté des utilisateurs du multicast IPv6 de profiter des contenus retransmis en multicast IPv4 comme cela a été le cas pour l'université de Guadalajara qui n'a accès qu'au multicast IPv6.

Toujours dans la perspective de contribuer au processus de transition du protocole IPv4 vers le protocole IPv6 il a été décidé à l'occasion de la conférence IPv6 qui s'est tenue à Caen du 11 au 13 juin 2003 d'assurer sa retransmission en multicast IPv6 et de passereller cette diffusion en multicast IPv4.

C. Développement des fonctionnalités des outils de visioconférence pour le téléenseignement interactif

1. Contexte

Avec l'expérience acquise lors de la retransmission en multicast de différents évènements à caractère pédagogique il s'est avéré nécessaire de développer de nouvelles fonctionnalités pour le logiciel Vic, ces fonctionnalités permettant d'améliorer le confort et l'interactivité des actions de téléenseignement dans lesquelles est impliquée l'association Aristote.

2. Vic haute résolution (4CIF)

a) Expression des besoins

Un des principaux problèmes qui s'est posé jusqu'à présent lors de la retransmission de présentations ou de conférences en multicast concerne la retransmission des transparents puisqu'il n'y a que peu de solutions et qu'aucune de ces solutions ne s'avère être pleinement satisfaisante.

La première solution consiste à brancher la sortie vidéo de l'ordinateur de l'orateur sur un rétroprojecteur, puis de projeter l'image sur un écran et enfin de filmer cet écran à l'aide d'une caméra branchée sur un ordinateur afin de retransmettre l'image de l'écran par Vic.

Cette opération permet donc de faire une conversion implicite de format depuis les transparents de l'orateur qui sont en plein écran, et donc adoptent la résolution de l'écran de l'orateur (généralement 1024x768), jusqu'à la résolution maximale imposée par Vic qui est de 352x288.

Comme on peut s'en douter, cette solution est fastidieuse à mettre en œuvre et très coûteuse en matériel, d'autant plus que le résultat à l'arrivée n'est pas complètement satisfaisant puisque les transparents proposent une lisibilité très moyenne. Le principal avantage de cette solution est que les auditeurs n'ont pas à se préoccuper de la synchronisation des transparents avec ceux de l'orateur puisqu'ils reçoivent directement sur leur écran les transparents de l'orateur.

Sur la capture d'écran suivante on peut voir le résultat d'un transparent filmé sur un écran puis retransmis ensuite au format CIF:

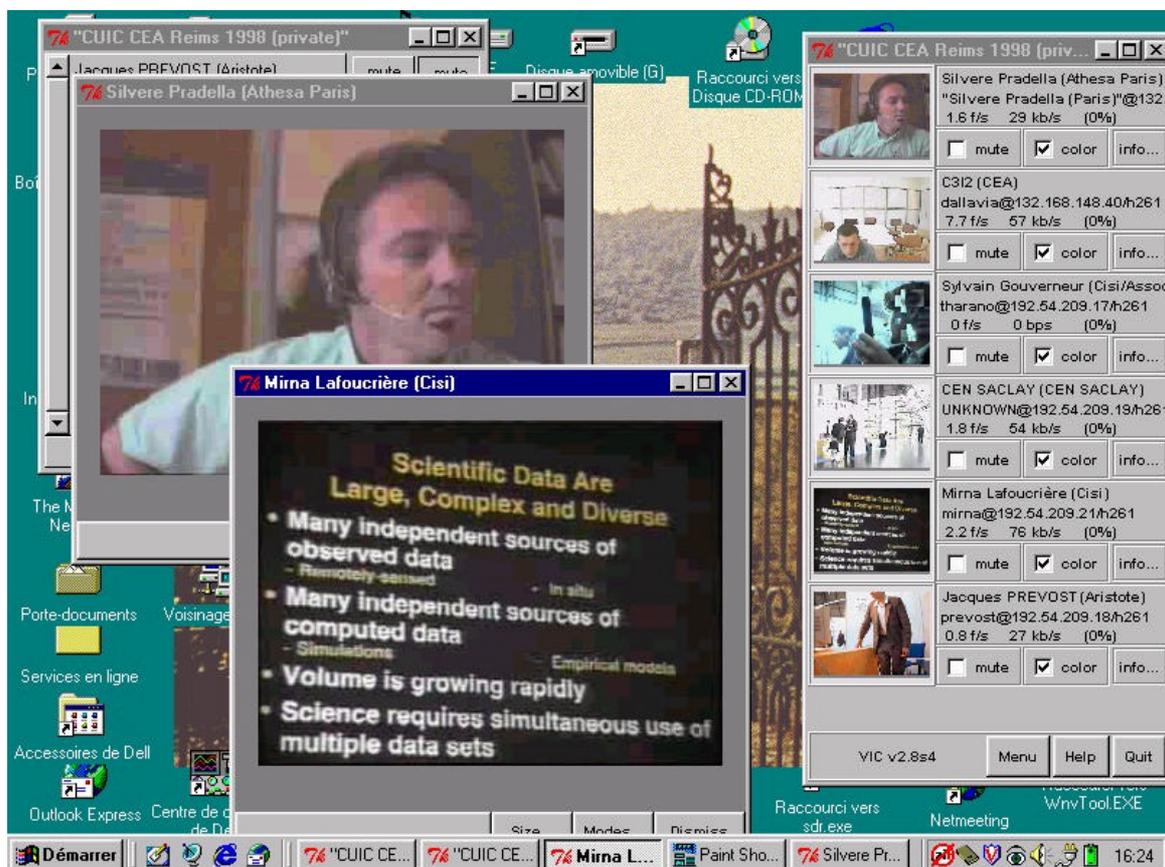


Figure 15 : Transmission d'un transparent filmé sur un écran puis transmis au format CIF.

Une autre solution consiste tout simplement à déposer les transparents sur un site Internet, au format HTML ou PowerPoint, de cette manière les auditeurs disposent de transparents de très bonne qualité, mais à ce moment-là le problème est que les auditeurs doivent faire eux-mêmes la synchronisation de leurs transparents avec ceux de l'orateur ce qui est très inconfortable et peu pratique.

Ainsi lors de la retransmission d'un événement en multicast lors de laquelle la vidéo de l'orateur était retransmise en multicast et les transparents disponibles sur le site de Renater, plusieurs auditeurs m'ont demandé s'il était possible de tourner la caméra vers l'écran projetant les transparents plutôt que de filmer l'orateur.

Ceci illustre bien le besoin de pouvoir retransmettre les transparents sous la forme d'un flux vidéo et ceci n'est possible qu'en utilisant la sortie vidéo de l'ordinateur de l'orateur afin de conserver la synchronisation des transparents transmis aux récepteurs avec ceux de l'orateur.

Une autre solution consiste à utiliser Netmeeting, mais cette solution n'est envisageable que pour des visioconférences impliquant un nombre très limité de participants (pas plus d'une demi-douzaine) du fait des problèmes de stabilité de Netmeeting. C'est cette solution qui est utilisée pour les cours DIM, mais l'expérience a montré que cette solution est loin d'être satisfaisante.

Une solution simple à mettre en œuvre et qui permettrait à l'auditoire de ne pas avoir à gérer la synchronisation des transparents avec ceux de l'orateur, mais aussi d'offrir aux auditeurs des transparents de très bonne qualité serait de disposer d'une version de Vic permettant d'émettre avec une résolution 4CIF. Il suffirait alors de brancher la sortie vidéo de l'ordinateur de l'orateur sur l'entrée de la carte d'acquisition d'un ordinateur afin d'émettre avec Vic les transparents en multicast. Pour ceci on peut utiliser un boîtier permettant de convertir la sortie vidéo de l'ordinateur de l'orateur en PAL, ceci afin de pouvoir par la suite brancher ce signal en entrée d'une carte d'acquisition.



Figure 16 : Transmission des transparents en multicast.

De cette manière les utilisateurs pourraient donc recevoir les transparents de l'orateur directement sur leur ordinateur en utilisant Vic.

Une autre application de cette nouvelle fonctionnalité de Vic consisterait à pouvoir disposer d'un vidéo projecteur virtuel pour des salles de réception distantes, en effet en ayant un ordinateur réceptionnant les transparents, il serait alors possible dans une salle de réception distante de projeter les transparents sur un écran afin de donner la possibilité à une assemblée de pouvoir suivre l'événement à distance. Ceci permettrait aux salles distantes de disposer d'un vidéoprojecteur virtuel.

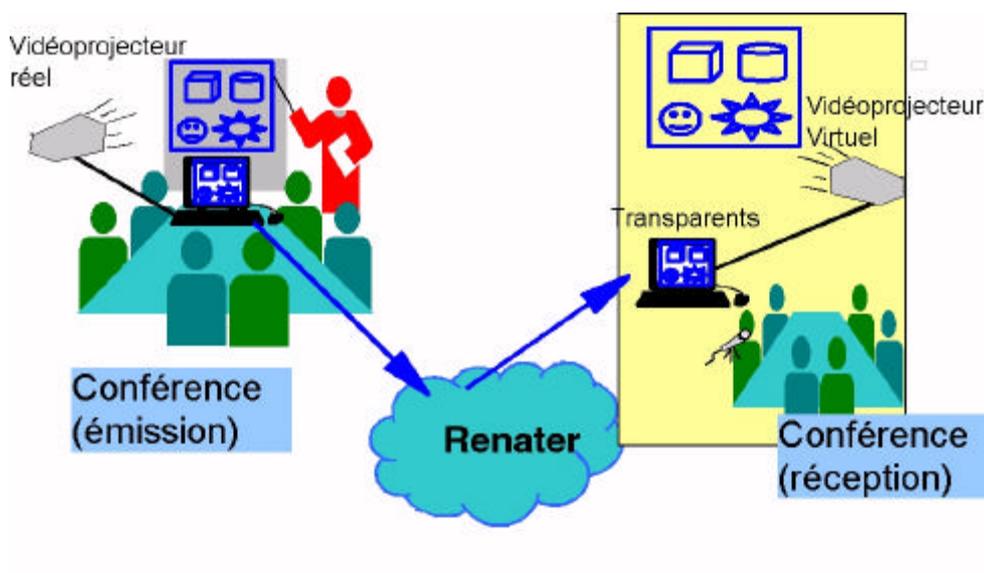


Figure 17 : Principe du vidéo projecteur virtuel.

Comme je l'ai expliqué précédemment, la version actuelle de Vic permet d'émettre avec un format d'image maximal de 352x288 (format CIF), or Vic permet d'émettre de la vidéo codée en H263 ou en H263+ et ces normes précisent qu'on peut émettre de la vidéo jusqu'en

16CIF (résolution de 1408x1152), il devait donc être possible de modifier Vic de manière à lui permettre d'émettre de la vidéo en 4CIF en utilisant le codec H263 ou le codec H263+.

b) Réalisation

Ayant une formation universitaire en Informatique j'ai procédé à des modifications du logiciel Vic en travaillant sous Linux puisque je suis très familier avec cet environnement de travail ainsi qu'avec son API système.

J'avais à ma disposition une carte d'acquisition Osprey 200 et une caméra Sony ce qui m'a permis de travailler dans de bonnes conditions puisque les cartes d'acquisition Osprey sont très bien supportées sous Linux (pour charger le module bttv permettant d'utiliser la carte d'acquisition Osprey il faut taper en tant que root la commande « modprobe bttv »).

Pour commencer il m'a fallu récupérer le code source de la version UCL du logiciel Vic, pour cela j'ai utilisé un client CVS (*Concurrent Version System*, système de gestion de versions concurrentes) pour me connecter au serveur CVS de UCL.

Tout d'abord il faut renseigner la variable d'environnement CVSROOT avec l'adresse du serveur CVS de UCL :

```
#export CVSROOT=:pserver:cvsanon@scary.cs.ucl.ac.uk:/cs/research/mice/starship/src/local/CVS_repository
```

Ensuite on se connecte au serveur CVS avec la commande suivante :

```
#cvs login
```

Un mot de passe est demandé, celui-ci est indiqué sur le site de UCL, il s'agit de « FoolsGold ».

Ensuite il faut récupérer successivement les quatre répertoires qui composent le code source de Vic :

```
#cvs checkout common
```

```
#cvs checkout tcl-8.0
```

```
#cvs checkout tk-8.0
```

```
#cvs checkout vic
```

A présent nous disposons du code source de Vic, il va falloir maintenant procéder à la compilation.

Pour cela il faut compiler successivement la librairie *common*, l'interpréteur Tcl, la librairie graphique Tk et le cœur du logiciel Vic en tapant successivement les commandes suivantes :

```
./configure  
make
```

Si on désire compiler Vic avec le support IPv6, dans le répertoire vic/ il faut taper la commande suivante:

`./configure --enable-ipv6`

au lieu de la commande :

`./configure.`

Lorsqu'on veut émettre de la vidéo avec Vic on choisit d'abord le codec vidéo qu'on va utiliser et le format de la vidéo. Dans la version actuelle de Vic si on choisit le codec H263 ou le codec H263+ on peut choisir le format 4CIF, cependant si on choisit le format 4CIF, lorsqu'on clique sur le bouton « transmit » afin d'émettre, Vic termine immédiatement son exécution, ce bogue étant par ailleurs bien connu des utilisateurs de Vic.

Il m'a donc fallu commencer par chercher la raison pour laquelle Vic terminait son exécution dès qu'on essayait d'émettre de la vidéo au format 4CIF.

L'utilisation d'un *debugger*¹ m'a permis de suivre pas à pas l'exécution du programme et ainsi de déterminer la raison du problème :

Dans le fichier `codec/encoder-h263v2.cpp`, dans la fonction `H263plusEncoder::size(int w,int h)`, un test était réalisé pour déterminer le format de la vidéo à encoder, et si l'image avait une taille supérieure à la taille CIF, le programme réalisait un appel à la fonction `exit()` ce qui avait pour conséquence de le faire terminer son exécution.

Le format 4CIF n'était donc pas supporté par la version officielle de Vic.

Voici la partie du code qui pose problème, les commentaires sont en bleu.

Il s'agit de la fonction « size » prenant en paramètres deux entiers `w` et `h`. Cette fonction est une méthode de la classe `H263plusEncoder`.

//les deux paramètres w et h désignent respectivement la largeur (width) et la hauteur //(height) des images de la vidéo à encoder

```
void
H263plusEncoder::size(int w, int h)
{
    Module::size(w, h);

    sed_pels = w;
    sed_lines = h;
    base_sed_pels = w;
    base_sed_lines = h;
```

//le format de la vidéo est comparé avec les formats standard

//comparaison avec le format CIF

```
    if (w == CIF_WIDTH && h == CIF_HEIGHT) {

        state->pic->source_format = 3;
```

¹ Logiciel permettant de suivre l'exécution d'un programme afin de détecter les erreurs de conception.

```
//comparaison avec le format QCIF (Quarter CIF, quart de CIF)
```

```
    } else if (w == QCIF_WIDTH && h == QCIF_HEIGHT) {  
        state->pic->source_format = 2;
```

```
//si le format est différent de CIF et QCIF, alors terminer l'exécution du programme
```

```
    } else {  
        exit(1);  
    }  
}
```

J'ai donc effectué des modifications au logiciel Vic de manière à lui permettre d'émettre de la vidéo au format 4CIF.

Concernant le format d'encodage vidéo utilisé, mon choix s'est porté sur le codec H263+ car il présente de meilleures performances.

Je vais à présent exposer les modifications que j'ai apportées au logiciel Vic afin de lui permettre d'émettre et de recevoir de la vidéo codée en H263+ au format 4CIF.

Ne disposant pas d'un logiciel permettant d'émettre en H263+ au format 4CIF, ma démarche a été de commencer par essayer de rendre Vic capable d'émettre de la vidéo au format 4CIF.

Au niveau de l'encodage, dans le fichier codec/encoder-h263v2.cpp il m'a fallu faire en sorte que le programme prenne en compte le format 4CIF, tout d'abord en déclarant les constantes définissant la largeur et la hauteur du format 4CIF de la manière suivante :

```
#define CIF4_WIDTH 704  
#define CIF4_HEIGHT 576
```

Ensuite dans la fonction `H263plusEncoder::size(int w,int h)`, j'ai modifié le corps de la fonction de manière à prendre en compte le format 4CIF.

```
void  
H263plusEncoder::size(int w, int h)  
{  
    /*fprintf(stderr,"send-h263: new frame size: %dx%d\n",w,h);*/  
    Module::size(w, h);  
  
    sed_pels = w;  
    sed_lines = h;  
    base_sed_pels = w;  
    base_sed_lines = h;
```

```
//On compare la taille de l'image qu'on veut encoder avec les tailles d'images supportées
```

```
if (w == CIF_WIDTH && h == CIF_HEIGHT) {  
    state->pic->source_format = 3;
```

```

    } else if (w == QCIF_WIDTH && h == QCIF_HEIGHT) {
        state->pic->source_format = 2;
//prise en charge du format 4CIF
        } else if (w == CIF4_WIDTH && h == CIF4_HEIGHT) {
            state->pic->source_format = 4;
        } else {
//Si le format est différent du QCIF, du CIF et du 4CIF alors afficher un message d'erreur
            fprintf(stderr,"send-h263: unsupported format\n");
//et terminer l'exécution du programme
            exit(1);
        }
    }
}

```

Ceci permet donc d'émettre de la vidéo codée en H263+ au format 4CIF avec Vic.

Au niveau de la réception il a fallu modifier Vic afin de lui permettre de recevoir de la vidéo au format 4CIF codée en H263+ car la version originale de Vic terminait son exécution dès lors qu'elle recevait de la vidéo au format 4CIF.

L'utilisation d'un debugger m'a permis de déterminer que les problèmes consistaient essentiellement en des tampons (variables permettant de stocker des données) dont la taille était insuffisante, ce qui était la cause d'erreurs de segmentation¹.

Ces erreurs de programmation provenaient du fichier `codec/decoder-h263v2.cpp` qui implémente le décodeur H263+.

Voici la manière de procéder du décodeur H263+ de Vic.

Lorsqu'un paquet est reçu, la zone du paquet contenant les données propres à la partie de l'image transmise est copiée à la suite des données déjà contenues dans la variable *bitstream*, un tableau de caractères.

Lorsqu'on a reçu tous les paquets contenant chacun une partie de la même image on copie le contenu de la variable *bitstream* (qui contient donc à ce moment-là une image entière) dans la variable *xxx_frame*, puis l'image nouvellement reçue est utilisée pour rafraîchir la fenêtre.

Un des problèmes provenait du fait que la variable *bitstream* était un tableau de caractères, ce qui entraînait que la taille de cette variable était fixée une fois pour toute, en effet en C++ la taille des tableaux de caractères doit être fixée une fois pour toute au moment de

¹ Une erreur de segmentation intervient lorsqu'un processus essaye d'écrire sur une zone mémoire pour laquelle il n'a pas les droits en écriture.

leur déclaration au début de la fonction. J'ai donc changé le type de cette variable en un pointeur¹ sur caractères (type char*) de la façon suivante :

```
char *bitstream;
```

Ceci permet de déterminer dynamiquement la taille de cette variable en fonction des besoins et du contexte.

Au départ j'ai fixé la taille de cette variable à zéro de la manière suivante :

```
//Si bsize vaut zéro c'est donc qu'on a pas encore écrit de données dans la variable bitstream
//on alloue donc la taille par défaut au tampon bitstream
```

```
if( bsize == 0 ){
```

```
//BITSSIZE est une constante valant 65536*4, c'est la taille initiale du tampon bitstream
```

```
    bitstream = (char *) malloc(BITSSIZE);
```

```
    if( ! bitstream ){
```

```
        perror("malloc failed");
```

```
    exit(1);
```

```
    }
```

```
    bsize=BITSSIZE;
```

```
    }
```

La variable bsize est utilisée afin de garder en mémoire la taille courante de la variable bitstream. En effet en C++ c'est au développeur de gérer l'allocation de la mémoire aux variables. Ainsi si on essaye d'écrire une chaîne de 1000 caractères (un caractère est codé sur un octet) dans une variable de taille 500 octets ceci va entraîner une erreur de segmentation. En mémorisant la taille de la variable bitstream on va pouvoir vérifier que la taille de ce tampon est suffisante avant de rajouter des données.

```
//Si on va dépasser la taille du tampon en écrivant les données du dernier paquet reçu
//à la fin du tampon alors on double la taille du tampon
```

```
if(b_off+(cc-bs) > bsize){
```

```
    while(b_off+(cc-bs) > bsize){
```

```
        bsize *=2;
```

```
    }
```

```
    fprintf(stderr,"realloc bitstream %d\n",bsize);
```

¹ Un pointeur est une variable spéciale dont la valeur est l'adresse d'une autre variable.

```
//On procede a la réallocation de mémoire

bitstream = (char *) realloc(bitstream,bsize);

if( ! bitstream ){

    perror("realloc failed");

    exit(1);

    }

}
```

De cette manière l'écriture des données dans le tampon n'entraîne plus de dépassement de tampon, ce qui permet à Vic de recevoir des données codées en H263+ au format 4CIF sans causer d'erreurs d'exécution.

Cependant cette version de Vic ne fonctionne que sous Linux.

J'ai donc modifié la version Windows de Vic afin de lui permettre de recevoir de la vidéo codée en H263+ au format 4CIF, cependant cette version n'est pas capable d'émettre de la vidéo au format 4CIF, en effet je me suis trouvé confronté à des difficultés concernant la taille des tampons à utiliser et je n'ai pas eu le temps nécessaire pour finaliser la version Windows.

Porter sous Windows la version de Vic permettant d'émettre en 4CIF sera donc un des objectifs de l'apprenti du DESS Applications des Réseaux et de la Télématique au GIP Renater lors de l'année 2003/2004.

c) Tests

Une fois l'implémentation réalisée il a fallu procéder à des tests afin de valider les concepts exposés précédemment : transmission des transparents en 4CIF et vidéo projecteur virtuel.

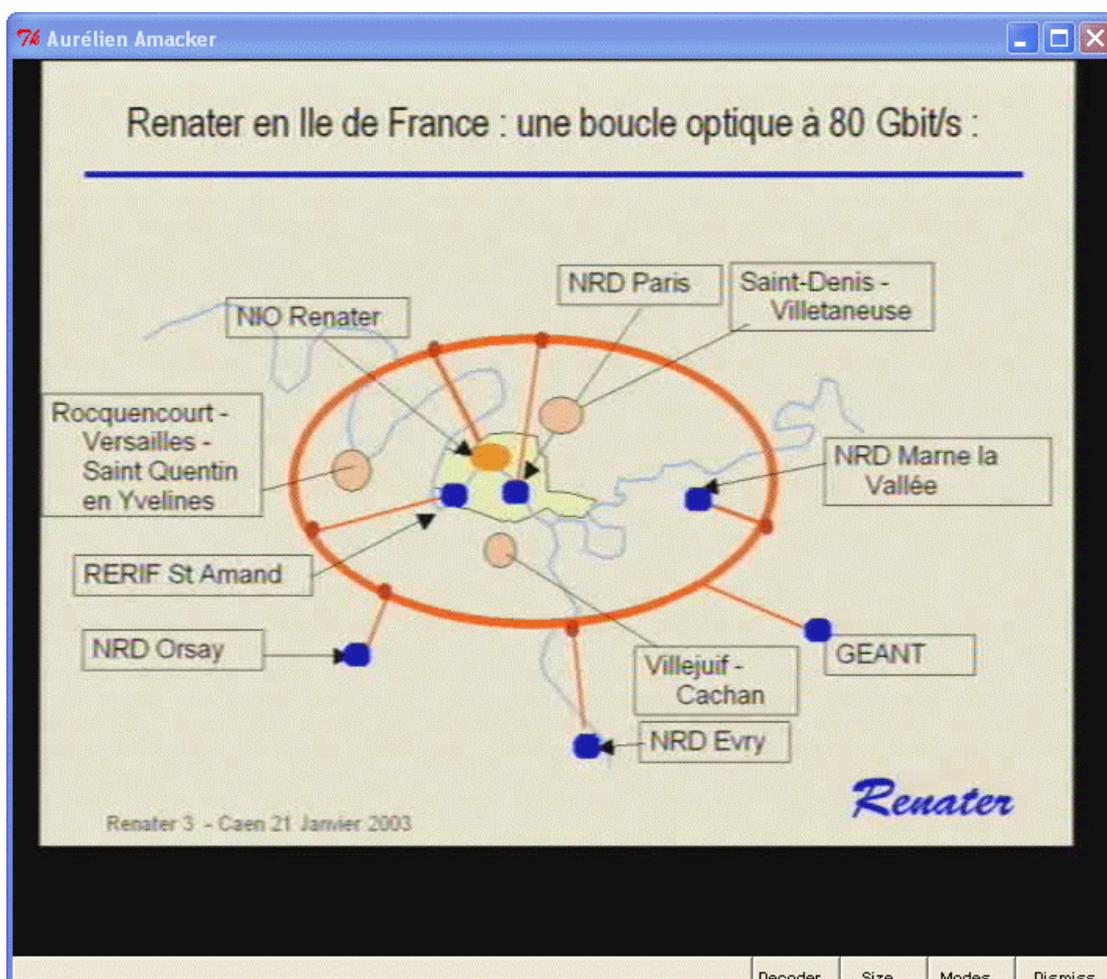
Nous avons donc procédé à des tests avec la participation de Guy Bisiaux (membre de la cellule technique du CRU, Comité Réseau des Universités) à l'université de Valenciennes.

Pour le premier test nous disposions d'un ordinateur portable sur lequel se trouvaient des transparents, cet ordinateur simulant le portable d'un orateur lors d'une présentation. Nous avons branché sur la sortie vidéo de cet ordinateur un boîtier permettant de convertir le signal VGA issu de l'ordinateur portable en PAL et ensuite nous avons branché la sortie du boîtier en entrée d'une carte d'acquisition d'un PC sous Linux qui émettait les transparents en 4CIF. De son côté Guy Bisiaux recevait les transparents à Valenciennes avec un PC équipé d'un Windows XP.

Voici le montage utilisé pour réaliser ce test :



Voici un transparent au format 4CIF reçu par Guy Bisiaux à Valenciennes sur son PC équipé d'un Windows XP :



On constate que la fenêtre de la vidéo en CIF fait la même taille que celle de la vidéo en 4CIF. Ceci s'explique par le fait que lorsqu'on reçoit de la vidéo en CIF avec Vic il est possible d'agrandir la fenêtre au format 4CIF, mais à ce moment-là on perd en résolution, c'est pour cette raison que la qualité d'un transparent émis au format CIF puis agrandi au format 4CIF est si mauvaise.

Cette expérience nous a donc permis de valider la transmission des transparents au format 4CIF à l'aide de Vic.

Ensuite nous avons fait un premier test en conditions réelles à l'occasion de la Causerie de Renater du 9 septembre 2003 au cours de laquelle nous avons émis les transparents en multicast au format 4CIF.

L'expérience s'est révélée concluante puisque le Vic sous Linux qui émettait à Renater ainsi que le Vic sous Windows XP qui recevait à Valenciennes ont très bien fonctionné pendant toute la durée de la présentation, c'est-à-dire pendant près de deux heures.

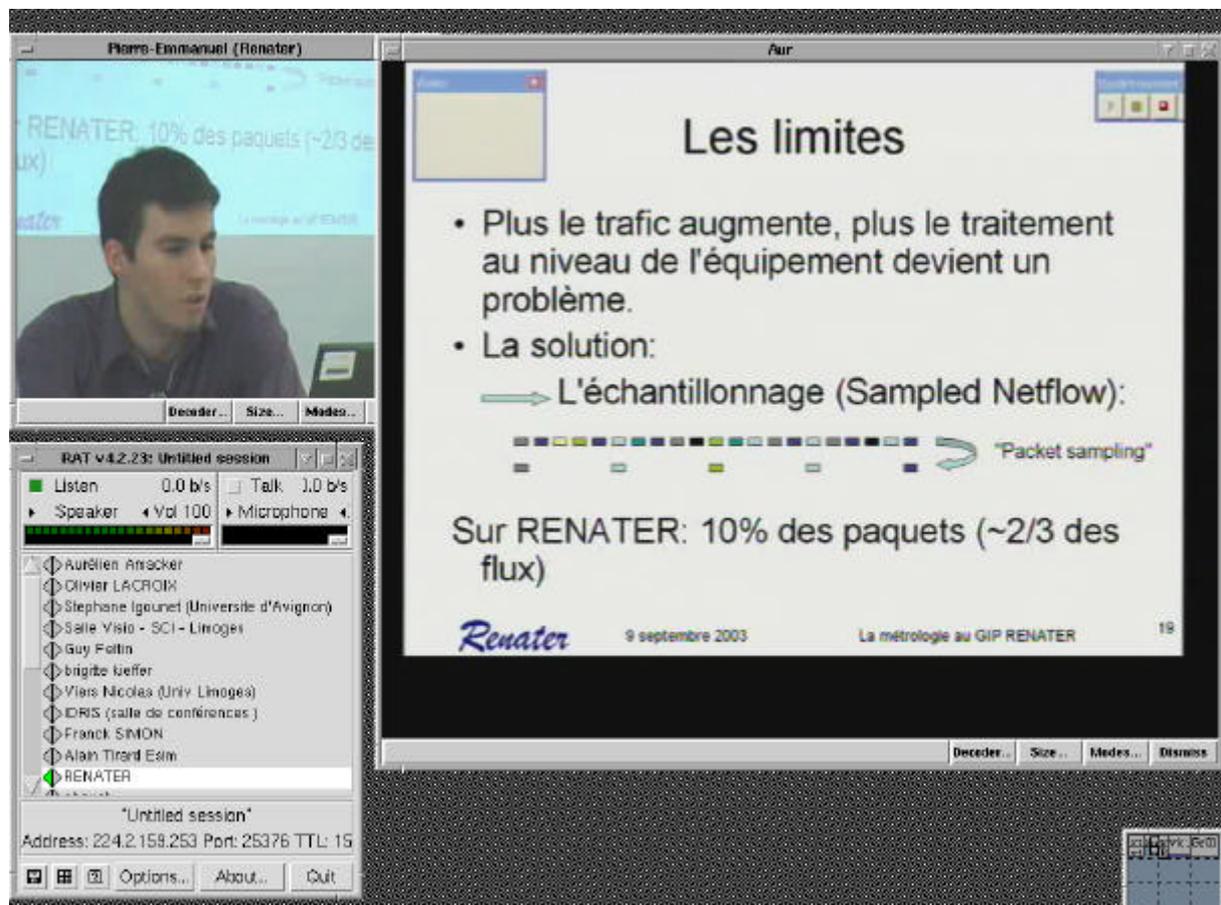


Figure 18 : capture d'écran du PC sous Linux émettant les transparents lors de la Causerie de Renater du 9 septembre 2003 sur le thème de la métrologie.

Cette solution sera donc utilisée dès l'année scolaire 2003/2004 lors des retransmissions des Causeries de Renater et des séminaires X-Aristote.

D'autre part nous avons validé le concept de vidéo projecteur virtuel puisque nous avons fait l'expérience de rétroprojeter sur un écran des transparents reçus en multicast au format 4CIF et nous avons obtenu un résultat très satisfaisant.

Ceci permettrait à des groupes de personnes d'assister à distance à des événements retransmis en multicast dans les mêmes conditions qu'en présentiel puisqu'il suffirait de vidéo projeter les transparents et la vidéo de l'orateur sur un écran.

3. Vic à deux vitesses

a) Expression des besoins

Lors de la retransmission d'événements en multicast comme des conférences IPv6 ou des Causeries de Renater, le débit utilisé pour émettre la vidéo est assez faible (aux alentours de 128 Kbit/s), en effet certains sites ne disposent que d'une faible bande passante comme c'est le cas par exemple pour certains partenaires du projet ATHENA, or l'objectif est de favoriser l'accès au plus grand nombre aux ressources pédagogiques transmises en multicast.

A l'inverse, sur le réseau Renater beaucoup d'établissements disposent de débits très importants, or jusqu'à présent lors de la retransmission de séminaires ou de conférences en multicast ils doivent se contenter d'une vidéo de faible qualité.

C'est pourquoi Lionel David, mon prédécesseur au GIP Renater, avait travaillé l'année dernière sur des solutions permettant de rendre accessibles des retransmissions d'événements en multicast pour des gens disposant d'une connexion bas débit sans pour autant pénaliser les gens disposant d'une connexion à haut débit.

Ceci l'a conduit à mettre en place un serveur UTG (*UCL Transcoding Gateway*, passerelle de transcodage de UCL) qui est à la fois une passerelle multicast – unicast et une passerelle haut débit – bas débit.

Son fonctionnement est le suivant : les gens disposant d'un accès bas débit à Internet et désirant suivre la retransmission d'un événement en multicast installent le client UTG sur leur ordinateur et le configurent en indiquant l'adresse du serveur UTG et le groupe multicast sur lequel est diffusée la vidéo de l'événement.

Le client UTG va alors émettre une requête au serveur UTG afin de lui demander de s'abonner au groupe multicast concerné. Le serveur reçoit donc le flux vidéo, le décode et le réencode en appliquant une technique de dégradation du signal, puis il le réemet en unicast à destination des clients.

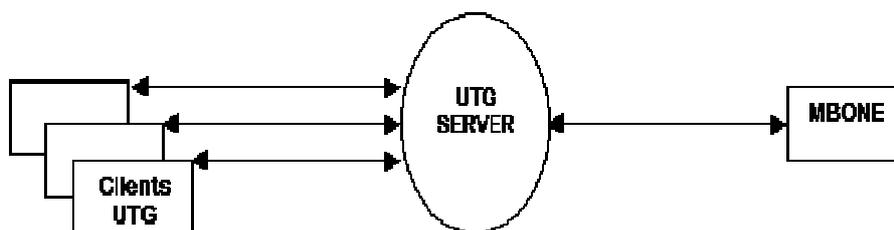


Figure 19 : principe de fonctionnement du serveur UTG

Ceci permet donc d'offrir aux récepteurs situés sur des liaisons bas débit la possibilité de suivre l'événement tout en offrant aux récepteurs situés sur des liaisons haut débit la possibilité de disposer d'une vidéo de bonne qualité.

Cependant, après avoir mis en place un serveur UTG à Renater, Lionel David avait estimé que cette solution n'était pas pleinement satisfaisante. J'ai moi-même eu l'occasion de tester ce service et je suis arrivé à la même conclusion.

Tout d'abord le serveur est très instable et demande à être relancé régulièrement, ce qui est très certainement dû au fait que UTG a été développé au milieu des années 1990 et n'a pas été maintenu par UCL, et de plus cette solution est contraignante puisqu'elle nécessite de la part des récepteurs l'installation d'un logiciel spécifique (le client UTG).

C'est pour toutes ces raisons que UTG ne représentait pas une solution satisfaisante par rapport aux besoins exprimés par la communauté académique.

Il est alors apparu qu'une solution idéale et pratique à mettre en œuvre consisterait à disposer d'une version du logiciel Vic permettant d'émettre à deux débits différents sur deux groupes multicast différents.



Figure 20 : schéma de principe du Vic à deux vitesses.

Ainsi l'on pourrait lors de la retransmission d'un événement diffuser deux flux vidéo, l'un de bonne qualité à destination des récepteurs disposant d'une importante bande passante et un autre de qualité moindre destiné aux récepteurs disposant d'une faible bande passante.

Ceci permettrait à chacun de pouvoir choisir la session qui lui convient le mieux en fonction de la bande passante dont il dispose, et ce sans avoir à installer de logiciel spécifique.

Enfin cette solution présente l'avantage de répondre au problème posé sans avoir à mettre en place un logiciel tiers comme une passerelle car cela alourdit le processus de mise en place d'une retransmission en multicast.

b) Réalisation

Lorsqu'on veut émettre de la vidéo en utilisant Vic (en fait lorsqu'on clique sur le bouton « Transmit » et qu'un appel à la fonction « transmit » définie dans le fichier tcl/ui-ctrlmenu.tcl est réalisé), un objet hérité de la classe Grabber est créé (*to grab* = capturer en anglais) et cet objet va capturer des images avec une fréquence fonction du nombre d'images par seconde que l'utilisateur a indiqué dans l'interface utilisateur.

Ces objets héritant de la classe Grabber sont spécialisés en fonction du système d'exploitation et du périphérique utilisé. Par exemple sous Windows l'objet hérité de la classe

Grabber sera un objet de type `VfwGrabber` (`Vfw` pour Video For Windows, l'API de capture d'images sous Windows) défini dans le fichier `video/grabber-win32.cpp` tandis que sous Linux en utilisant une carte d'acquisition et une caméra on utilisera un objet grabber de type `V4IGrabber` (`V4I` pour Video4Linux, l'API de capture d'images sous Linux) défini dans le fichier `video/grabber-video4linux.cpp`.

Sous Linux l'objet `V4IGrabber` ouvre le périphérique vidéo en lecture/écriture et effectue un appel à l'appel système `mmap()`.

`mmap()` est une fonction système permettant de réaliser une projection d'un fichier ou d'un périphérique sur une zone en mémoire, de cette manière toute écriture sur cette zone mémoire implique une écriture sur le périphérique ou sur le fichier et réciproquement toute écriture sur le fichier ou sur le périphérique implique une écriture sur la zone mémoire.

Ceci permet de procéder à des lectures et à des écritures sur le périphérique ou sur le fichier d'une manière beaucoup plus efficace car les temps d'accès à la mémoire vive sont bien inférieurs aux temps d'accès à un périphérique ou à un fichier (un accès à un fichier correspond à un accès au disque dur).

Ensuite en utilisant l'API `video4Linux`, l'objet `V4IGrabber` capture des images (ces images sont donc stockées dans la projection en mémoire créé par `mmap()`) et les transmet à la classe `TransmitterModule` qui se charge ensuite d'encoder ces images selon le format de compression voulu, de découper ces données en paquets puis de les envoyer sur le réseau.

J'ai donc choisi de procéder de la manière suivante : deux objets `V4IGrabber` sont créés, le premier sera celui de la session haut débit tandis que le deuxième sera celui de la session bas débit.

Le premier objet `V4IGrabber` va donc réaliser l'appel à `mmap()` et effectuer les captures successives tandis que le deuxième objet `V4IGrabber` ne va pas faire de captures, mais il lui sera permis de lire la projection en mémoire créée par l'appel à `mmap` effectué par le premier objet `V4IGrabber` (la zone mémoire renvoyée par l'appel à `mmap()` est déclarée comme *static* (mot-clé du langage C++) et est donc accessible par tous les objets de la classe `V4IGrabber`), de cette manière au lieu de faire des captures il va se contenter de lire le contenu de la zone en mémoire et va pouvoir transmettre les images à un objet de la classe `TransmitterModule` pour que celui-ci effectue l'encodage de ces images et leur envoi sur le réseau.

Le contenu de la projection du périphérique en mémoire étant régulièrement modifié par les captures d'images du premier objet `V4IGrabber`, à chaque fois que le deuxième objet lit le contenu de cette zone en mémoire il trouve la dernière image issue de la capture réalisée par le premier objet de la même manière que s'il venait lui-même de réaliser la capture.

Il est important que le deuxième objet `V4IGrabber` se contente de lire le contenu de la projection en mémoire sans réaliser de captures lui-même car dans un premier temps j'avais testé une version selon laquelle les deux objets effectuaient des captures, mais les captures réalisées par le deuxième objet gênaient les captures d'image réalisées par le premier ce qui était la cause d'un effet visuel saccadé gênant dans la vidéo capturée par le premier objet `V4IGrabber`.

Ceci permet donc à la version de Vic ainsi modifiée d'émettre sous Linux la vidéo issue d'une même source (caméra, webcam,...) à deux débits différents sur deux groupes multicast différents.

Voici une capture d'écran de l'ordinateur de M. Netzahualcoyotl Ornelas Garcia recevant les deux flux vidéo émis à deux débits différents :

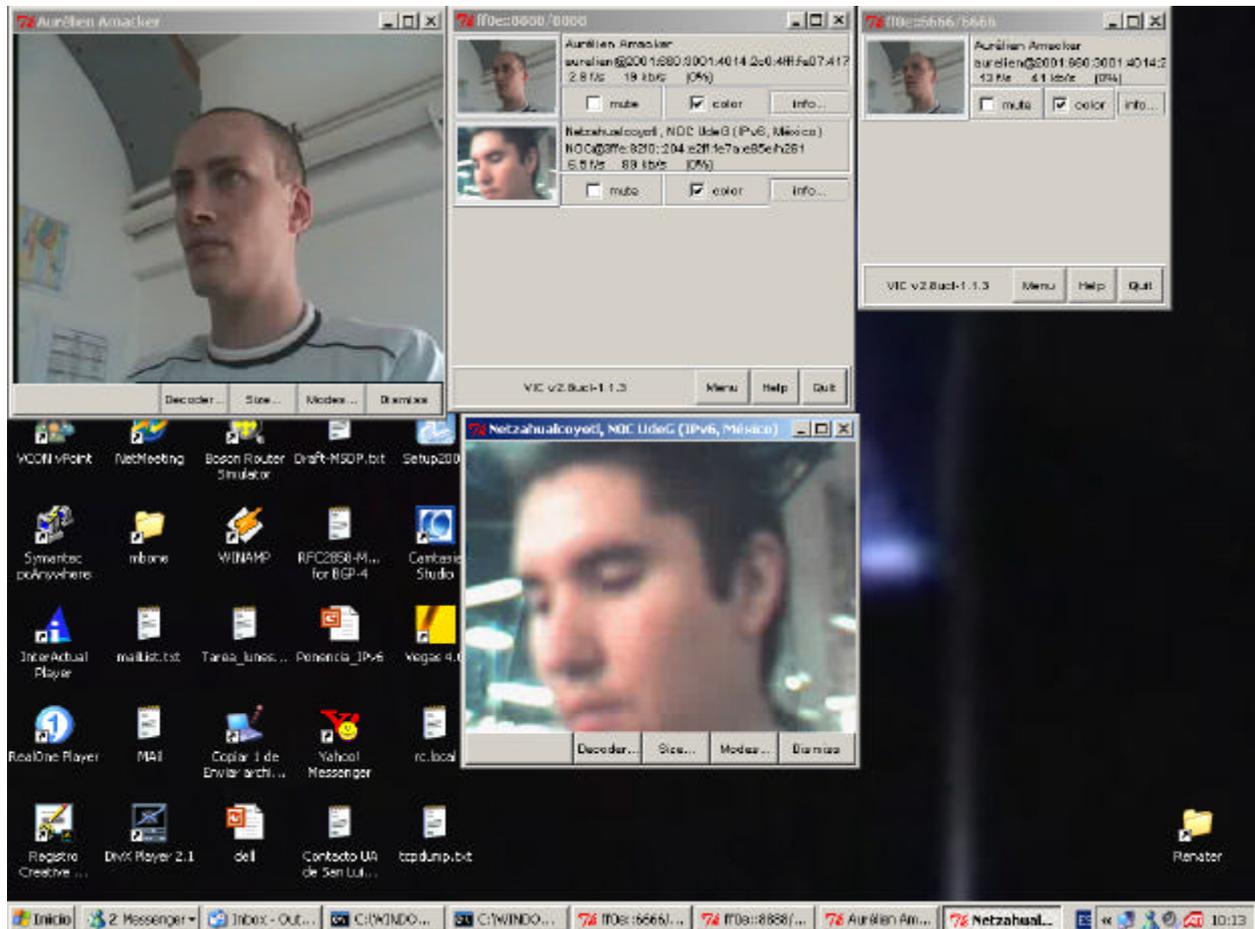


Figure 21 : Vic à deux vitesses.

Sur la figure 21 on peut voir un même signal vidéo transmis depuis Renater en multicast à deux débits différents. Sur le premier groupe multicast (ff0e ::8888/8888) le débit utilisé est de 19 Kbit/s avec une fréquence d'images de 2,8 images par seconde tandis que sur le deuxième groupe multicast (ff0e ::6666/6666) le débit utilisé est de 41 Kbit/s avec une fréquence d'images de 13 images par secondes.

Pour lancer le Vic à deux vitesses il faut taper la commande :

`vic -2 adresse_multicast1/numero_port1 adresse_multicast2/port2`

Par la suite j'ai procédé à une modification de l'interface graphique de Vic afin de permettre à l'utilisateur de renseigner les paramètres de chacune des deux sessions, c'est-à-dire le nombre d'images par seconde ainsi que le débit utilisé.

Voilà une partie du panneau de configuration de la transmission du Vic à deux vitesses lancé avec la commande suivante :

```
vic -2 224.11.7.99/3456 224.11.8.99/5678
```

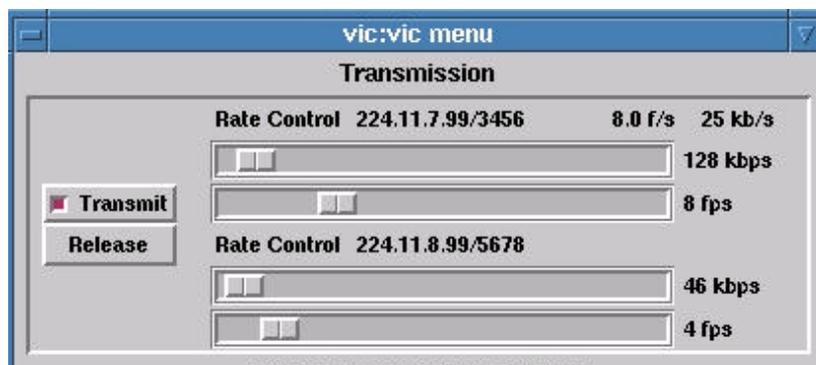


Figure 22 : Partie du panneau de configuration permettant de renseigner les paramètres propres à chacune des deux sessions

Après avoir réalisé une version du Vic à deux vitesses sous Linux j'ai travaillé sur le développement d'un Vic à deux vitesses sous Windows.

J'ai donc essayé de procéder de la même manière que sous Linux, tout en adaptant le code aux spécificités de la programmation sous Windows, mais je ne suis pas arrivé à un résultat satisfaisant, en effet l'API Video For Windows ne permet pas de réaliser deux captures différentes sur un périphérique d'acquisition vidéo (tout comme Linux), j'ai donc essayé de procéder de la même manière que sous Linux avec deux objets grabber dont le premier réalise les captures sur le périphérique et le second se contente de lire les images capturées par le premier, mais le fait que le deuxième objet grabber ne capture pas les images impliquait qu'il commençait à lire les images avant même que le premier ait pu capturer une seule image, ce qui du fait des limitations de l'API Video For Windows, empêchait le deuxième de réaliser les captures d'image.

J'ai donc essayé de synchroniser les deux objets en forçant le deuxième à attendre que le premier objet hérité de la classe Grabber ait fini de capturer la première image avant de lire le résultat des captures mais je n'y suis pas parvenu. Porter cette version de Vic à deux vitesses sous Windows sera donc un des objectifs de l'apprenti du DESS ART au GIP Renater de l'année 2003/2004.

Cependant il est à noter que du côté réception n'importe quelle version de Vic (Vic UCL ou Vic OpenMash) permet de recevoir des données émises avec le Vic à deux vitesses.

D. Transfert de connaissances

Etant donné que d'une part il reste encore des travaux de développement à réaliser dans le cadre des activités de téléenseignement dans lesquelles est impliquée l'association Aristote et que d'autre part il n'existe pratiquement pas de documentation relative au développement du logiciel Vic, j'ai réalisé une documentation traitant du fonctionnement interne de Vic, de la façon de le compiler, de l'interface entre Tcl/Tk et C++.

Cette documentation est disponible sur le site Internet de Renater, dans la partie « Grilles de Conférences », elle devrait permettre à d'autres développeurs désirant contribuer au développement du logiciel Vic de comprendre plus rapidement le fonctionnement interne de ce logiciel et ainsi d'être plus efficaces.

D'autre part, l'apprenti du DESS ART qui me succède au GIP Renater, M. Christophe Herviaux, doit porter le Vic à deux vitesses ainsi que le Vic 4CIF (en émission) sous Windows lors de l'année 2003/2004, j'ai donc fait en sorte de lui apprendre tout ce dont il aura besoin afin de reprendre mon travail dans de bonnes conditions.

Après l'avoir familiarisé avec les logiciels utilisés dans le cadre des retransmissions en multicast, je lui ai montré comment récupérer le code source de Vic, comment le recompiler sous Linux et sous Windows, enfin je lui ai expliqué les parties du code qu'il sera amené à modifier ainsi que les modifications que j'ai apportées au logiciel Vic lors de mon apprentissage.

Ce transfert de connaissances devrait lui permettre d'aborder cette année de travail au GIP Renater avec tous les moyens nécessaires à la réussite de sa mission.

V. Conclusion

Le travail effectué cette année a permis de confirmer l'expansion du projet ATHENA avec le nouveau partenariat très prometteur avec l'université de Guadalajara au Mexique mais il a surtout permis de développer les fonctionnalités des outils utilisés pour les retransmissions en multicast de contenus à caractère pédagogique et par là même d'améliorer le confort et l'interactivité des actions de téléenseignement dans lesquelles est impliquée l'association Aristote.

Ainsi lors de la retransmission en multicast IPv4 ou en multicast IPv6 de conférences ou de séminaires il sera possible de transmettre la vidéo de l'orateur à deux débits différents afin de proposer aux récepteurs une vidéo en adéquation avec leur bande passante.

Il sera également possible désormais de retransmettre les transparents en multicast grâce au Vic 4CIF ce qui permettra de s'affranchir de l'usage de Netmeeting et de ses inconvénients dans le cadre des cours DIM mais aussi de valoriser le multicast en proposant ainsi une utilisation innovante des fonctionnalités offertes par ce protocole.

Cependant, afin d'assurer une plus large diffusion du Vic 4CIF et du Vic à deux vitesses, il est important de porter ces logiciels sous Windows et c'est donc le travail que devra effectuer mon successeur au GIP Renater lors de l'année 2003/2004.

L'expérience a démontré que le développement de l'utilisation des nouvelles technologies de l'information et de la communication dans le cadre du téléenseignement nécessite non seulement des avancées techniques dans le domaine des applications et des réseaux mais aussi de nouer des partenariats afin de tester et de valider ces innovations.

Ceci explique pourquoi l'association Aristote est impliquée à la fois dans des projets à composante relationnelle comme le projet ATHENA et dans des projets à composante technique comme la Grille de conférences.

VI. Annexes

A. Bibliographie – Références

"An experimental IPv6/IPv4 multicast gateway" (une passerelle expérimentale multicast IPv4/IPv6), Luc Beurton.

Norme H.261 de l'UIT-T « Utilisation des lignes pour la transmission des signaux autres que téléphoniques – Codec vidéo pour services audiovisuels à p x 64 kbit/s »

Norme H.263 de l'UIT-T « Codage vidéo pour communications à bas débit »

RFC 1889 « *RTP: A Transport Protocol for Real-Time Applications* » (RTP : un protocole de transport pour les applications temps réel).

RFC 2236 « *Internet Group Management Protocol, Version 2* » (protocole de gestion des groupes Internet, version 2).

RFC 2362 « *Protocol Independent Multicast-Sparse Mode (PIM-SM)* » (multicast indépendant du protocole, mode clairsemé).

RFC 2463 « *Internet Control Message Protocol (ICMPv6) for IPv6* » (protocole des messages de contrôle d'Internet pour IPv6).

RFC 2710 « *Multicast Listener Discovery (MLD) for IPv6* » (Protocole de découverte des récepteurs multicast pour IPv6).

RFC 3493 « *Basic Socket Interface Extensions for IPv6* » (Extensions pour IPv6 de l'interface basique des connecteurs logiciels).

RFC 3513 « *IP Version 6 Addressing Architecture* » (Architecture d'adressage du protocole IPv6).

RFC 3587 « *IPv6 Global Unicast Address Format* » (Format des adresses unicast IPv6 globales).

B. Références sur Internet

<http://aristote.asso.fr> : le site de l'association Aristote.

<http://www.renater.fr> : le site du GIP Renater.

<http://www.renater.fr/GC/> : le site du projet Grille de Conférences.

<http://www-mice.cs.ucl.ac.uk/multimedia/software/> : le site du laboratoire Multimédia et Réseau de UCL.

<http://www.openmash.org> : le site du projet OpenMash à l'université de Berkeley en Californie.

<http://www.kabassanov.com> : le site personnel de Konstantin Kabassanov sur lequel on peut trouver des versions de Vic pour Windows compatibles avec IPv6.

C. Glossaire

API	: <i>Application Programming Interface</i> (Interface de programmation d'applications) : ensemble de fonctions utilisées par un programme pour demander et exploiter les services d'un autre programme ou d'une bibliothèque de fonctions.
ATHENA	: Actions pour des Transmissions Harmonieuses et des Echanges de Natures Académiques, projet initié par l'association Aristote et visant à développer les échanges de contenu scientifique à travers les réseaux de la recherche.
Bit	: <i>Binary Digit</i> (nombre binaire) : unité binaire de quantité d'information qui peut prendre deux valeurs distinctes : 0 ou 1.
Bogue	: Anomalie dans la conception d'un programme entravant le déroulement correct de son exécution.
CEA	: Commissariat à l'Energie Atomique.
Chrominance	: C'est la couleur d'un signal vidéo, composée de la teinte et de la saturation.
CIDR	: <i>Classless Inter-Domain Routing</i> (routage inter-domaine sans classes).
CIF	: <i>Common Intermediate Format</i> (format commun intermédiaire) : format d'image normalisé par l'UIT-T spécifiant une taille d'image de 352x288.
Classe	: Abstraction d'un type de donnée caractérisée par des propriétés (attributs et méthodes) communes à des objets et permettant de créer des objets ayant ces propriétés.
CNES	: Centre National d'Etudes Spatiales.
Codec	: CODEur/DECODEur : logiciel permettant de procéder à l'encodage et au décodage d'un signal vidéo numérique.
Compiler	: Action de rassembler tous les fichiers composant le code source d'un programme afin de les traduire en langage machine en vue de constituer un logiciel exécutable.
Constante	: Symbole désignant un emplacement mémoire contenant une valeur fixe durant toute l'exécution d'un programme.
CVS	: <i>Concurrent Version System</i> (système de gestion de versions concurrentes) : logiciel client/serveur permettant de conserver un historique des modifications apportées au code source d'un logiciel.

C++	: Langage orienté objet développé par Bjarne Stroustrup dans les laboratoires d'AT&T au début des années 1980.
<i>Debugger</i>	: Logiciel permettant de suivre pas à pas l'exécution d'un programme afin de détecter les bogues.
DESS	: Diplôme d'Etudes Supérieures Spécialisées.
DIM	: Diplôme Informatique et Multimédia : projet commun à plusieurs DESS en France et au Sénégal visant à expérimenter l'utilisation de la visioconférence en multicast dans le cadre du téléenseignement interactif.
Erreur de segmentation	: Une erreur de segmentation intervient lorsqu'un processus essaye d'écrire sur une zone mémoire pour laquelle il n'a pas les droits en écriture.
Fonction	: Ensemble d'instructions effectuant un calcul à partir de valeurs fournies en entrée.
GN6	: Groupe des Néophytes IPv6.
H261	: Norme de l'UIT-T spécifiant un codec vidéo adapté pour des liaisons RNIS.
H263	: Norme de l'UIT-T spécifiant un codage vidéo adapté pour des liaisons bas débit.
<i>Hop Limit</i>	: Limite du nombre de sauts : durée de vie d'un paquet IPv6. A chaque traversée d'un routeur ce compteur est décrémenté, lorsqu'il atteint zéro le paquet est détruit. Equivalent du TTL (<i>Time To Leave</i> , temps restant à vivre) en IPv4.
HTML	: <i>Hyper Text Markup Language</i> (langage de balises hypertexte) : langage permettant la création de pages Internet intégrant des liens « hypertexte ».
ICMPv6	: <i>Internet Control Message Protocol (ICMPv6) for IPv6</i> » (protocole des messages de contrôle d'Internet pour IPv6).
IGMPv2	: <i>Internet Group Management Protocol version 2</i> (protocole de gestion de groupes Internet, version 2).
INRIA	: Institut National de la Recherche en Informatique et Automatique
IP	: <i>Internet Protocol</i> (protocole Internet) : protocole de la couche réseau du modèle ISO.
IPv4	: <i>Internet Protocol Version 4</i> (version 4 du protocole Internet).

IPv6	: <i>Internet Protocol Version 6</i> (version 6 du protocole Internet).
ISO	: <i>International Organization for Standardization</i> (organisation internationale de standardisation) : organisme de standardisation à l'échelle mondiale.
IVS	: <i>INRIA Videoconferencing System</i> (système de vidéoconférence de l'INRIA).
Langage de scripts	: Type de langage informatique dont le code source n'est pas compilé mais interprété par un logiciel (un interpréteur) qui procède à l'exécution des instructions contenues dans le code source.
LIP6	: Laboratoire d'Informatique de l'université Paris 6.
Luminance	: Intensité lumineuse d'un signal vidéo.
M6Bone	: <i>Multicast IPv6 backBONE</i> (épine dorsale multicast IPv6) : réseau multicast IPv6 expérimental composé de tunnels IPv6 dans IPv6 ou IPv6 dans IPv4.
MLD	: « <i>Multicast Listener Discovery</i> » (Protocole de découverte des récepteurs multicast).
<i>Multicast</i>	: Multidiffusion : fonctionnalité du protocole IP dédiée à la transmission de données multimédia.
NAT	: <i>Network Address Translation</i> (translation d'adresses réseau).
NTSC	: <i>National Television System Committee</i> , Comité national du système télévisuel : standard télévisuel aux Etats-Unis et au Japon.
Objet	: Instance d'une classe.
PAL	: <i>Phase Alternation Line</i> , ligne alternée en phase : standard télévisuel en Europe.
Paramètre	: Valeur fournie en entrée d'une fonction ou d'un programme.
PIM	: <i>Protocol-Independent multicast</i> (multicast indépendant du protocole) : protocole de routage multicast indépendant du protocole de routage unicast sous-jacent.
Pointeur	: Variable dont la valeur est l'adresse d'une autre variable.
PowerPoint	: Logiciel d'édition et de présentation de transparents.

Rat	: <i>Robust Audio Tool</i> (outil audio fiable) : logiciel d'audioconférence en IPv4 ou en IPv6 multicast ou unicast.
RNIS	: Réseau Numérique à Intégration de Services.
RTP	: <i>Real Time Protocol</i> (protocole temps réel) : protocole se situant entre les couches réseau et transport du modèle ISO destiné à véhiculer du trafic avec des exigences de temps réel.
RTCP	: <i>Real Time Control Protocol</i> (protocole de contrôle temps réel) : protocole destiné à recueillir des informations concernant des données transmises à l'aide du protocole RTP.
SECAM	: Système sEquentiel Couleur A Mémoire : standard télévisuel en France.
Script	: Fichier source d'un langage de script.
Socket	: Connecteur logiciel: mécanisme logiciel de communication entre processus informatiques, souvent utilisé entre une application et un réseau.
Static	: Mot-clé du langage C++, un attribut d'une classe déclaré comme <i>static</i> est accessible par toutes les instances de cette classe.
Tcl	: Langage de scripts disponible sous Unix/Linux et Windows.
Téléenseignement	: Enseignement à distance.
Tk	: Librairie graphique permettant de faire appel à des <i>widgets</i> (composants graphiques) dans des scripts.
Tunnel IP	: Technique consistant à transmettre un datagramme IP contenu dans un autre datagramme IP.
UCL	: <i>University College of London</i> .
UDP	: <i>User Datagram Protocol</i> (protocole de datagrammes utilisateurs) : protocole de transport non fiable, sans connexion, utilisé principalement par des applications privilégiant la rapidité à la fiabilité de la transmission.
UIT-T	: Union Internationale des Télécommunications – Secteur de normalisation des Télécommunications.
UTG	: <i>UCL Transcoding Gateway</i> (passerelle de transcodage de UCL) : logiciel de type client/serveur permettant de diminuer le débit d'une source multicast audio ou vidéo et de l'émettre en unicast.

- Variable : Symbole désignant un emplacement mémoire contenant une valeur pouvant varier au cours de l'exécution d'un programme.
- V4L : *Video For Linux* (vidéo pour Linux) : API de capture d'images sous Linux.
- VFW : *Video For Windows* (vidéo pour Windows) : API de capture d'images sous Windows.
- Vic : *VideoConferencing Tool* (outil de vidéoconférence) : logiciel de vidéoconférence en IPv4 ou en IPv6 multicast ou unicast