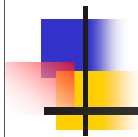


Adaptation des applications distribuées à la Qualité de Service fournie par le réseau de communication



Fabien Michaut, Francis Lepage
{fabien.michaut | francis.lepage}@cran.uhp-nancy.fr

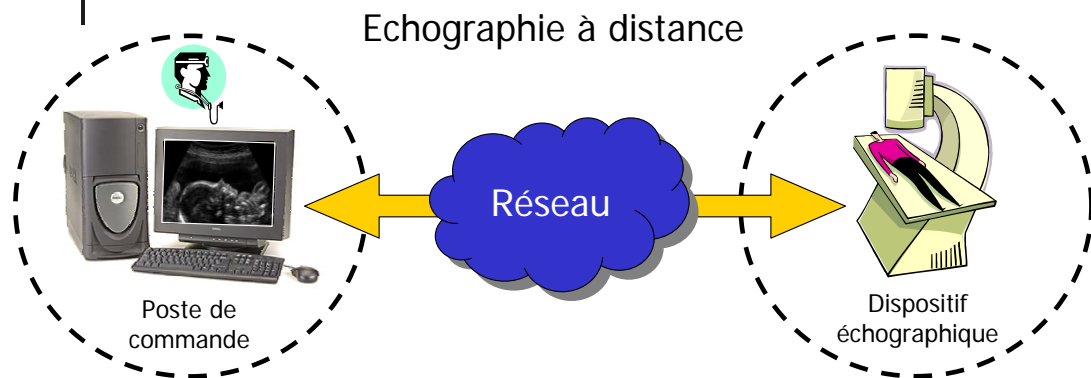


Centre de Recherche en Automatique de Nancy
CRAN - CNRS UMR 7039
Université Henri Poincaré - Nancy I





Exemple de télé-opération

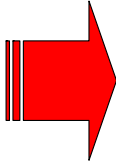


- Trafics "complexes": contrôle, vidéo
- Trafics contraints
- Fonctionnement de l'application dépendant de la Qualité de Service (QoS) fournie par le réseau de communication
- Prise en compte de la QoS du réseau indispensable



Approches possibles

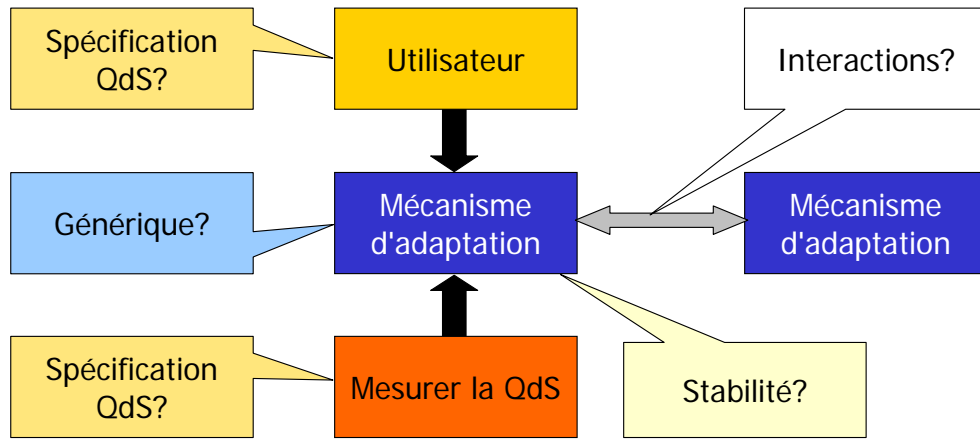
- Réserveation de ressources
 - [IntServ95], [Nahrstedt95], etc.
 - Garantie stricte de QoS
 - Performances optimales de l'application



- Maîtrise des ressources pas totale ou impossible
 - Réseaux hétérogènes (Ex: Internet)
 - Adaptation des applications aux ressources disponibles
 - "Multimedia applications should be adaptive" [Diot95]



Adapter les applications à la QoS disponible?



Besoin d'un cadre pour l'adaptation:
Architecture de QoS



Principes

- **Transparence et séparation [Campbell96]**
 - Mécanismes de gestion de QoS **transparents** pour l'application
 - Nombre limité de fonctionnalités "QoS" à implanter dans l'application
 - Complexité de la gestion de la QoS déléguée à l'architecture
 - **Séparation** des tâches de gestion de la QoS et de traitement des flux applicatifs
 - Ex. d'une application vidéo: Signalisation <-> Flux vidéo

- Facilite le travail du développeur d'applications
- Permet une vision / gestion globale de toutes les applications



Spécification/Traduction de QdS [Cheong99]

- Spécification de la QdS
 - Utilisateur, application, système d'exploitation, système de communication
- Exemple d'une application vidéo

Utilisateur	Qualité vidéo subjective
Application	Codec vidéo, taille de l'image, etc.
Système d'exploitation	Mémoire, etc.
Système de communication	Bande passante, etc.

- Traduction (translation) de la QdS
 - Conversion de la spécification de la QdS



Mécanismes d'adaptation existants

- Adaptation du débit de la source
 - Flux audio/vidéo
 - Taux de compression / choix du codec (Ex: [Fry99])
- Dimensionnement dynamique de mémoires-tampon
 - Compensation de gigue
 - Flux audio (Ex: [Benaissa02])
- Techniques plus génériques
 - Bibliothèques de composants adaptables [Chang99][Ott98]
 - Implique de prévoir tous les composants possibles
 - Modèle Prayer [Bharghavan97]
 - Exécution d'un code différent selon la QoS
 - L'application spécifie comment réagir lorsque la QoS varie
 - Un superviseur extérieur pilote l'adaptation
 - Transparence/Séparation



Modèles existants

		XRM [Lazar94]	OoS-A [Campbell98]	Omega [Nahrstedt96]	Heidelberg [Vogt98]	Prayer [Bharghavan97]	Adaptation du débit	Compensation de gigue
Cadre architectural	Spécification QoS	✓	✓	✓	✓	✗		
	Traduction QoS	✓	✓	✓	✓			
	Métrologie							
	Préférences Utilisateur		✓	✓				
Adaptation des applications	Générique					✓		
	Transparent					✓		
	Stabilité							
	Multi-application							
Réservation de ressources		✓	✓	✓	✓	✓		



Oui



Incomplet

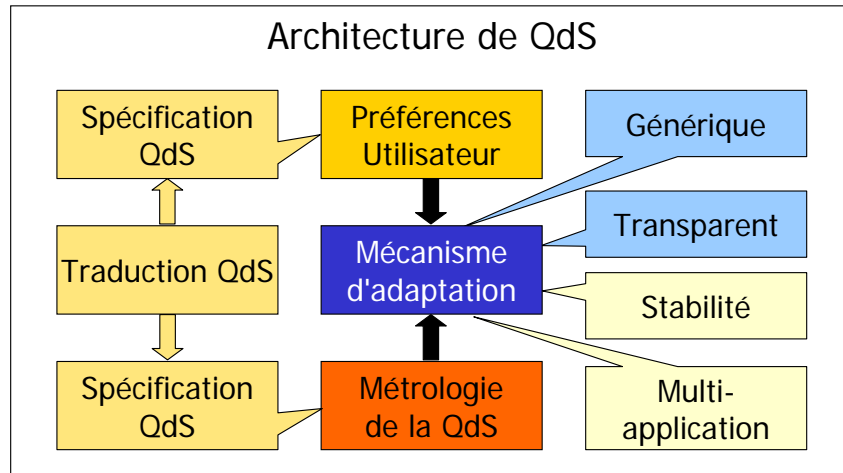


Non



Problématique

Adapter les applications en ligne à la QdS mesurée





Suite de l'exposé:

- "QoS-Adapt": Une architecture pour l'adaptation en ligne des applications
 - Mécanisme d'adaptation des applications
 - Service de métrologie de la QoS
 - Architecture "QoS-Adapt"

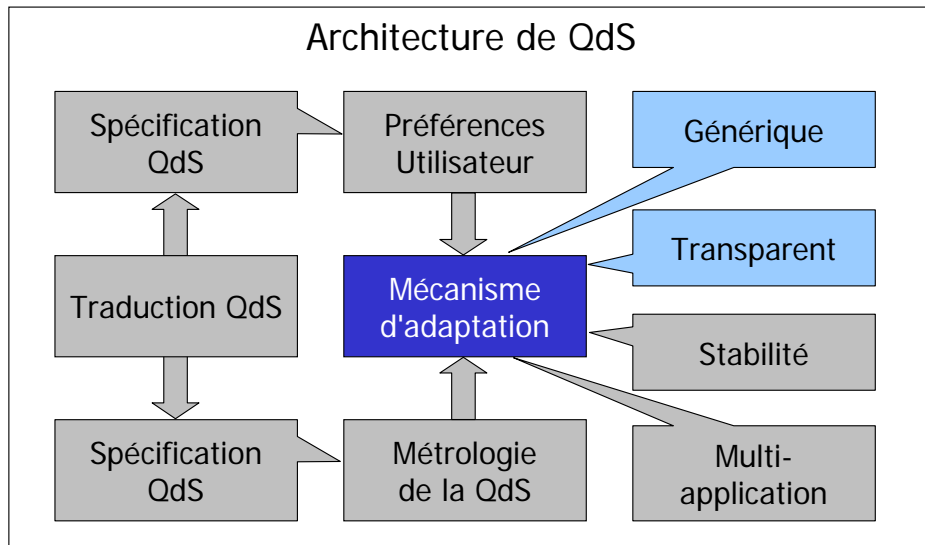
- Expérimentations
 - Télé-pilotage d'un robot mobile
 - Télé-asservissement en position d'un robot mobile

- Conclusions & Perspectives

"QoS-Adapt"

Une architecture pour l'adaptation en ligne des applications

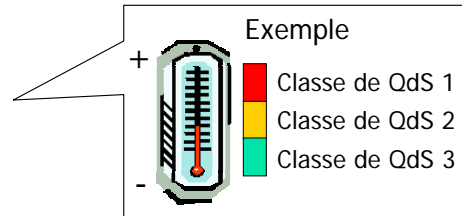
1. Mécanisme d'adaptation des applications
2. Service de métrologie de la QoS
3. Architecture "QoS-Adapt"





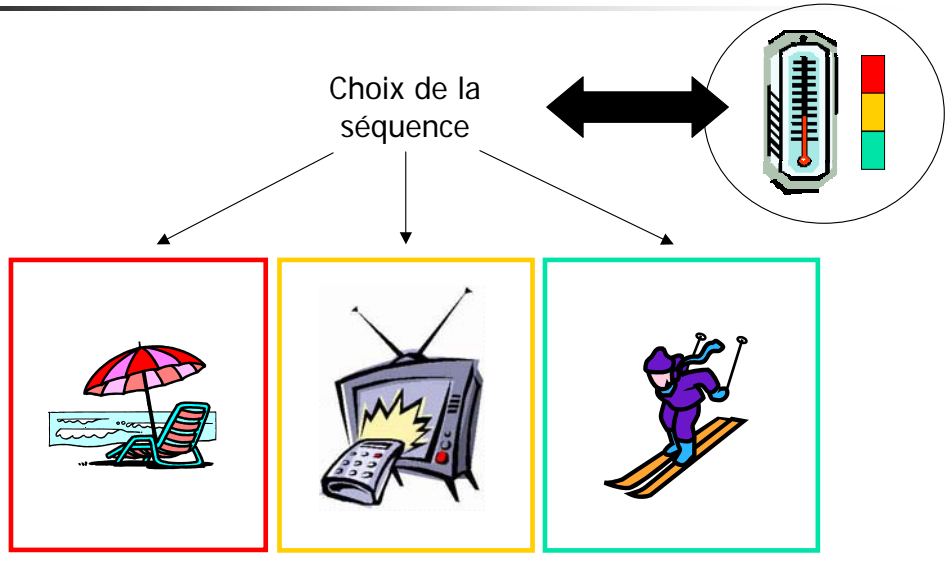
Mécanisme d'adaptation en ligne

- Basé sur des concepts issus du modèle Prayer [Bharghavan97]
- Classes de QoS
 - "Niveaux" de QoS
- Séquences d'exécution
 - Associées aux classes de QoS
 - Exécution d'un code différent selon la classe de QoS disponible



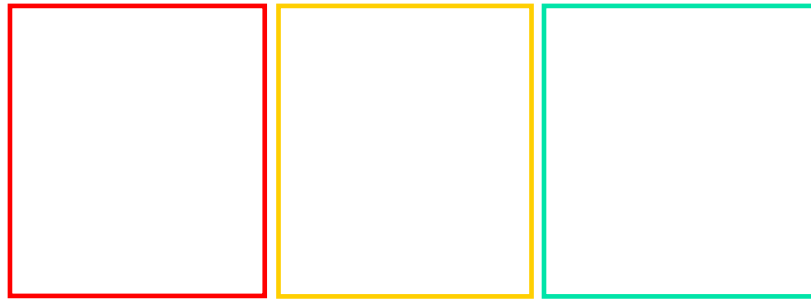
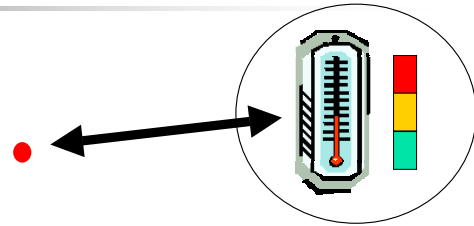


Mécanisme d'adaptation: exemple



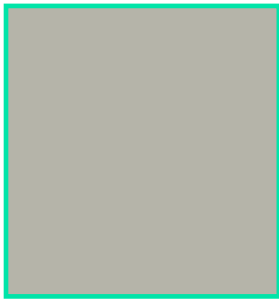
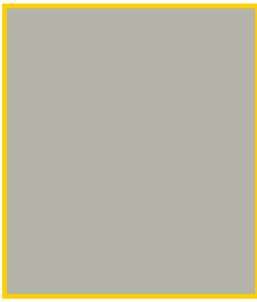
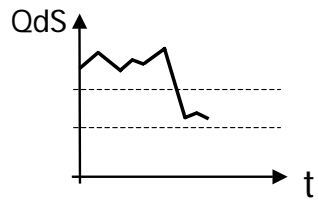
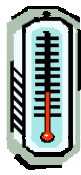


Mécanisme d'adaptation

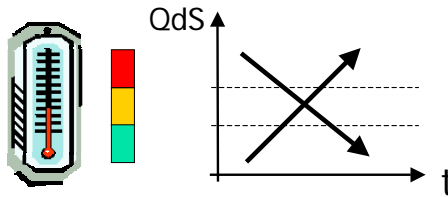




Adaptation en ligne: exemple



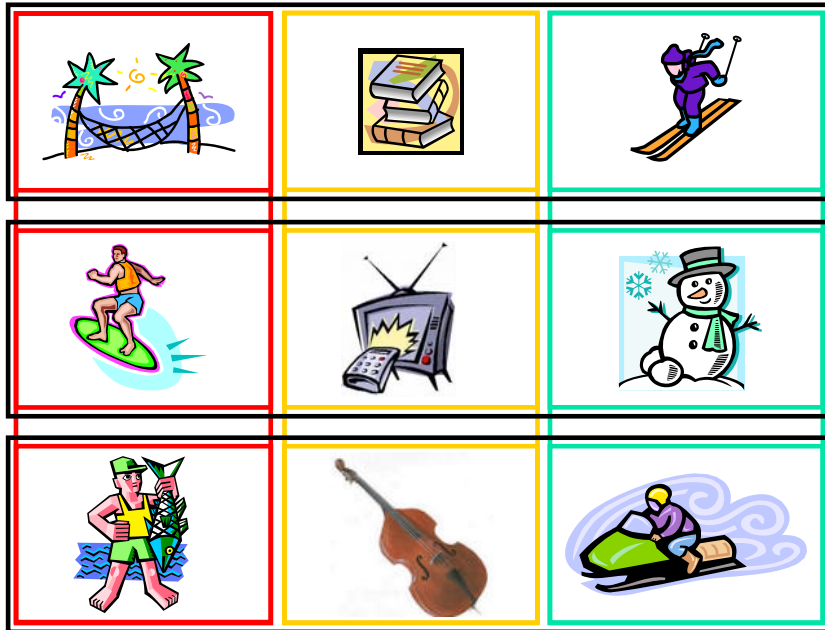
Actions d'adaptation

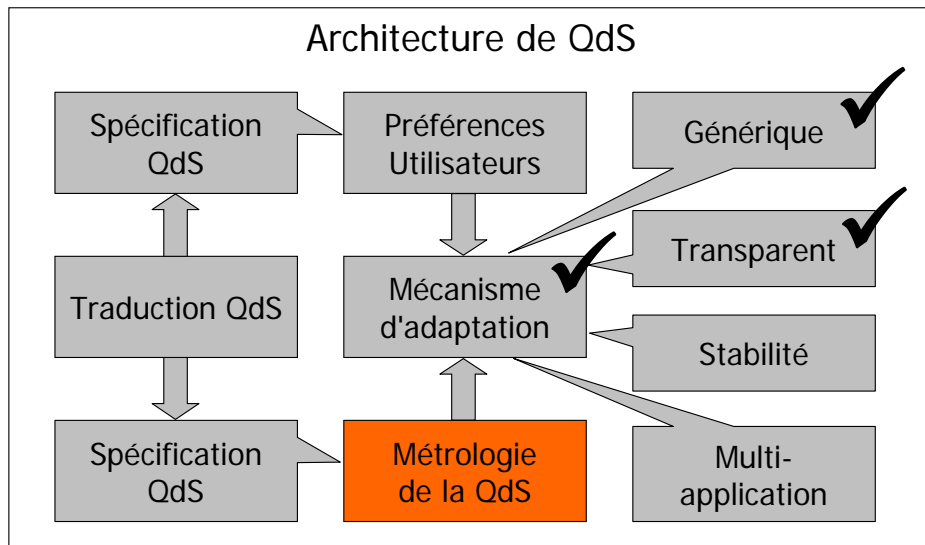


- Actions "Bas"
 - Initiées lorsque la classe de QoS courante plus disponible
 - BLOCK, BEST_EFFORT, ROLLBACK, ABORT, **SWITCH()**
- Actions "Haut"
 - Initiées lorsque une (des) classe(s) de QoS supérieure(s) disponible(s)
 - BEST_EFFORT, ROLLBACK, **SWITCH()**
- Un couple (Action Haut, Action Bas) associé à chaque séquence
- Application informée de l'initiation nécessaire d'une action d'adaptation



Blocs d'adaptation







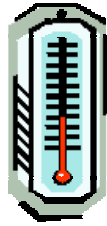
Métrologie de la QoS réseau

- Mesurer la QoS telle qu'elle est perçue par l'application
 - Mettre en œuvre une politique de métrologie

Fonctionnalités attendues
Paramètres différents
Plusieurs mesures simultanées
Mesures bidirectionnelles
Paramétrable



Métrologie de la QoS réseau



Mesurer la QoS telle qu'elle est ressentie par l'application



Mesures actives:
Effectuées en envoyant des
paquets-sonde

Systeme A



Systeme B



Paramètres retenus et outils de mesures

	Outils de mesure
Capacité (lien)	Pathchar [Jacobson97], Bing [Beysac95], Clink [Downey99], Pchar [Mah99], Nettimer [Lai01]
Bande passante disponible (lien/chemin)	Cprobe [Carter96], Pipechar [Jin01], Pathload [Jain02], Spruce [Strauss03], Topp [Melander03]
Bande passante minimale (chemin)	Bprobe [Carter96], Pathrate [Dovrolis01], Nettimer [Lai01], Sprobe [Saroiu01]
Délai unidirectionnel	-
Délai aller-retour	Ping [BSD4.3]
Variation de délai	Iperf [Tirumala03]
Pertes de paquets	Ping [BSD4.3], Sting [Savage99], Iperf [Tirumala03]
Route	Traceroute [Jacobson89]



Analyse

Fonctionnalités	
Paramètres différents	Non
Plusieurs mesures simultanées	Non
Mesures bidirectionnelles	Non
Paramétrable	Non

- Outils **inutilisables** pour mettre en œuvre une politique de métrologie



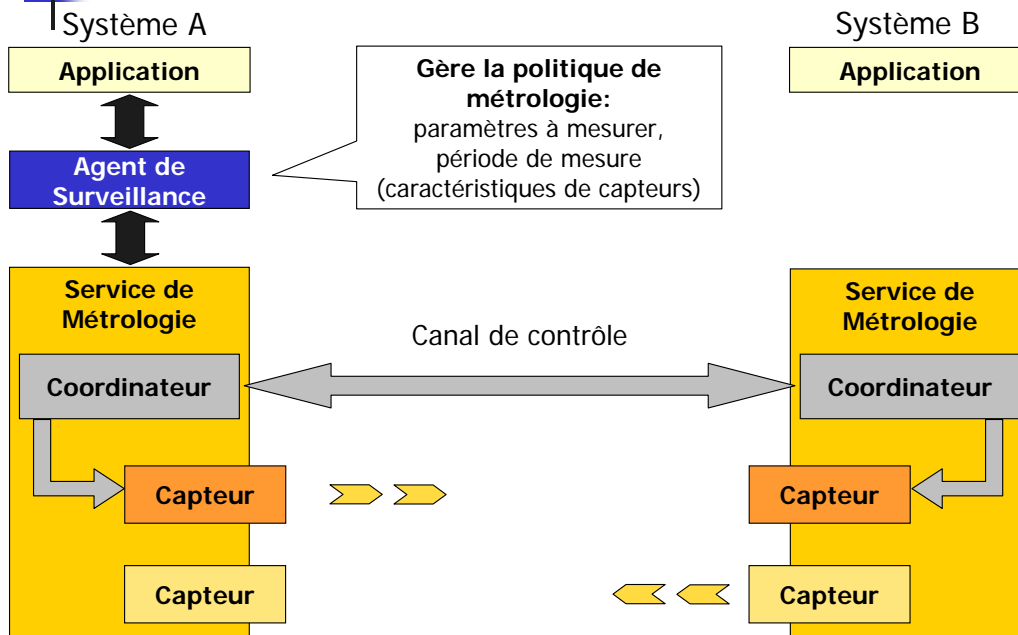
Service de métrologie

- Cadre générique pour l'implantation de techniques de mesure
 - Dont les techniques des outils précédents
- Mesures actives
 - Réalisées par des couples d'émetteur-récepteur de paquets-sonde: Capteurs
- Structure modulaire
 - Ajout de couples de capteurs



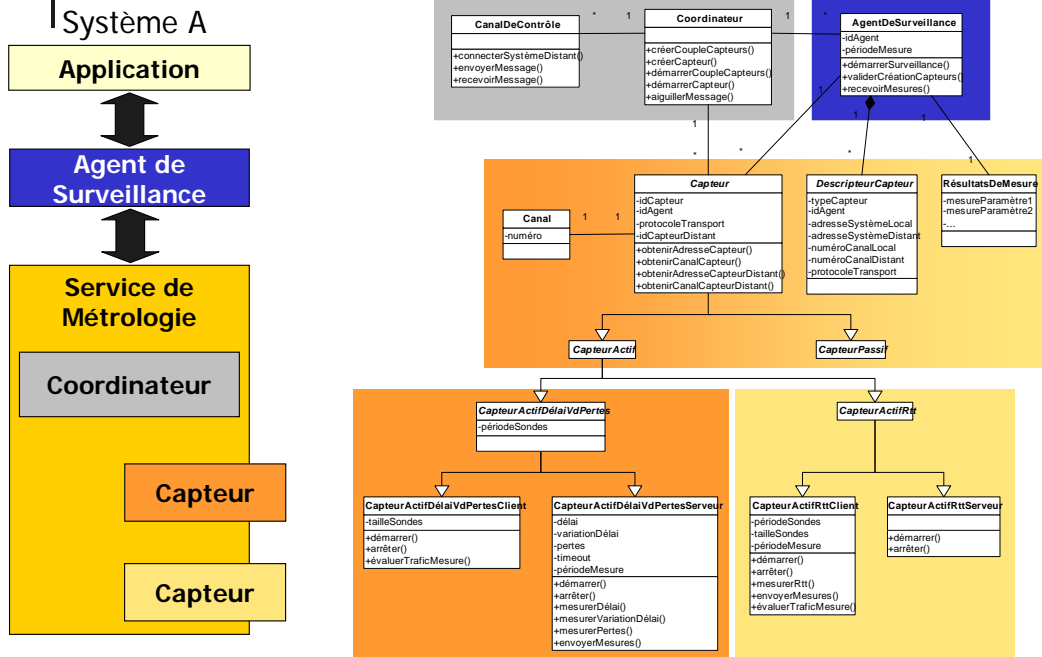


Service de métrologie: principe



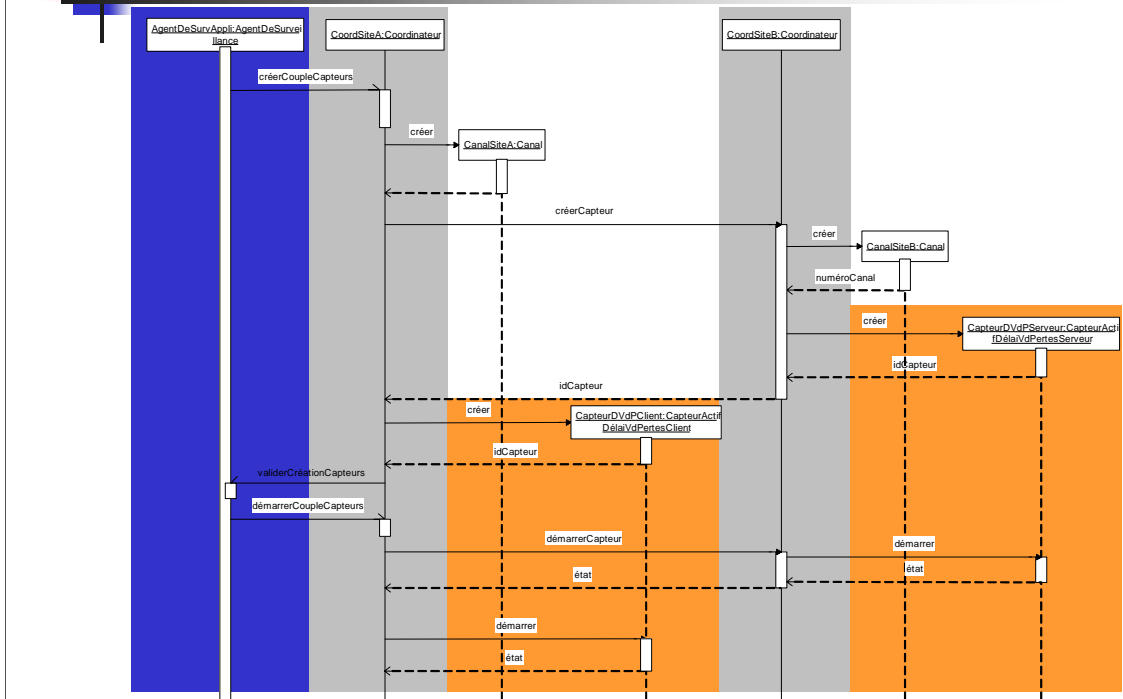


Modélisation UML - Diagramme de Classes





Modélisation UML – Diagramme de Séquence





Fonctionnalités du service de métrologie

Fonctionnalités	
Paramètres différents	✓
Plusieurs mesures simultanées	✓
Mesures bidirectionnelles	✓
Paramétrable	✓



Implantation: QoSmet

- Compatible One-Way Active Measurement Protocol [IETF]

One-way delay (RFC2679)

- Type-P-One-way-Delay
- Type-P-One-way-Delay-Poisson-Stream
- Type-P-One-way-Delay-Percentile
- Type-P-One-way-Delay-Median
- Type-P-One-way-Delay-Minimum

One-way delay variation (RFC3393)

- Type-P-One-way-ipdv
- Type-P-One-way-ipdv-Poisson-Stream
- Type-P-One-way-ipdv-jitter
- Type-P-One-way-peak-to-peak-ipdv

Packet loss (RFC2680)

- Type-P-One-way-Packet-Loss
- Type-P-One-way-Packet-Loss-Poisson-Stream
- Type-P-One-way-Packet-Loss-Average
- Type-P-One-way-Packet-Loss

Packet loss pattern sample metrics (RFC3357)

- Type-P-One-way-Loss-Distance-Stream
- Type-P-One-way-Loss-Period-Stream
- Type-P-One-way-Loss-Noticeable-Rate

Packet reordering (draft)

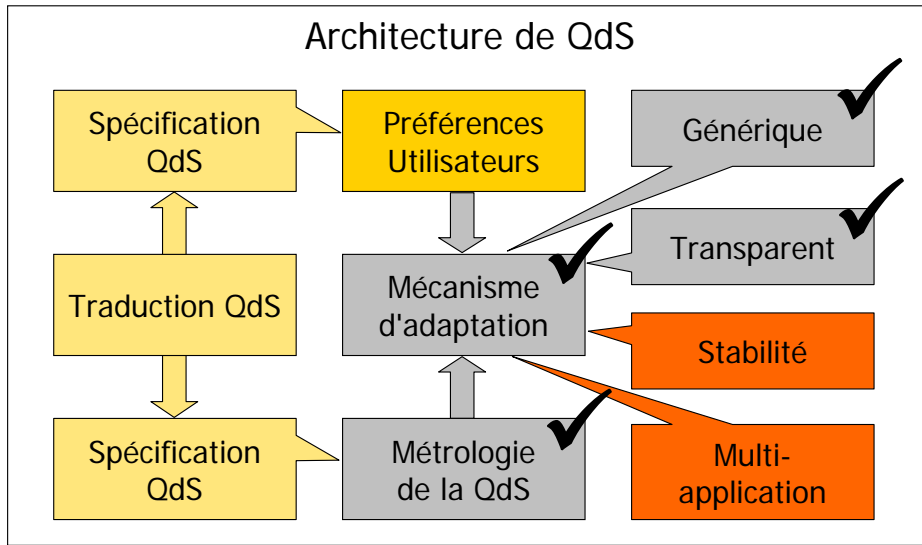
- Type-P-Reordered-Ratio-Stream

En cours d'implantation

- Bande passante disponible selon l'algorithme "Spruce" [Strauss03]

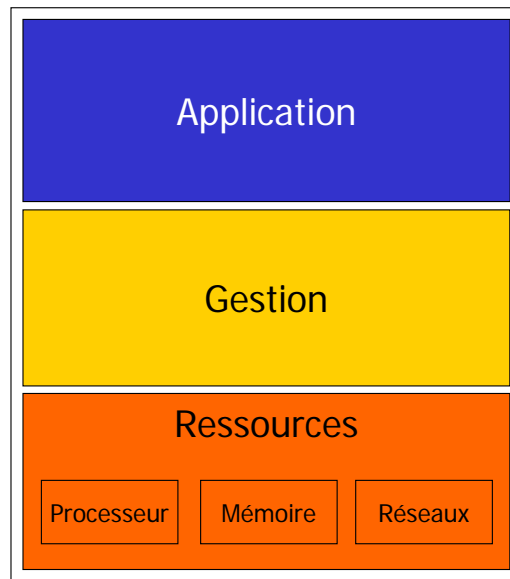
Paramètres mesurés selon les recommandations de l'IETF (métriques conformes aux RFCs)

Architecture de QoS



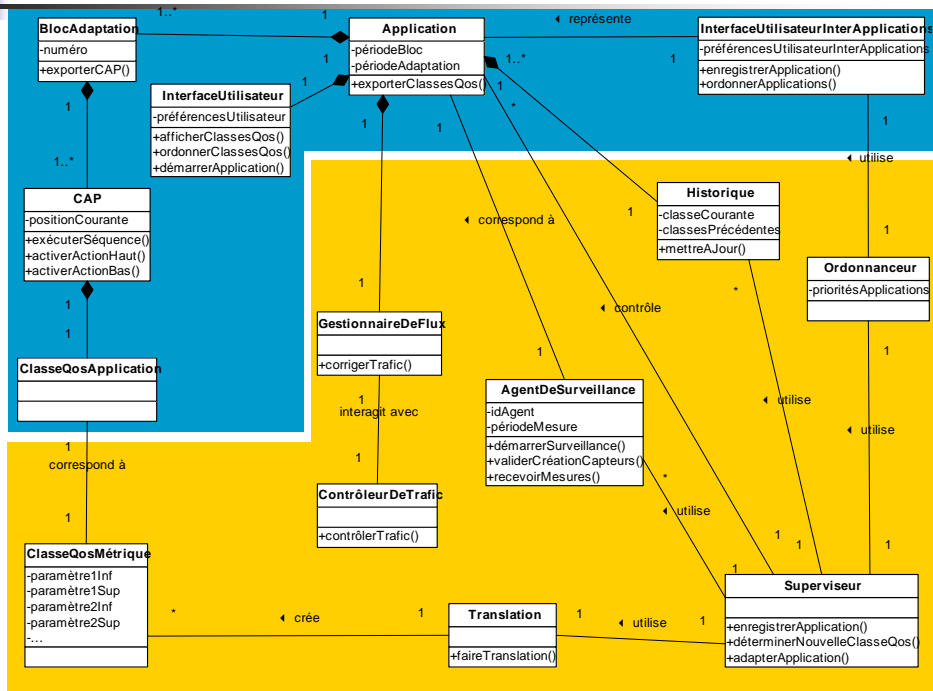


Architecture "QoS-Adapt"





Modélisation UML - Diagramme de Classes

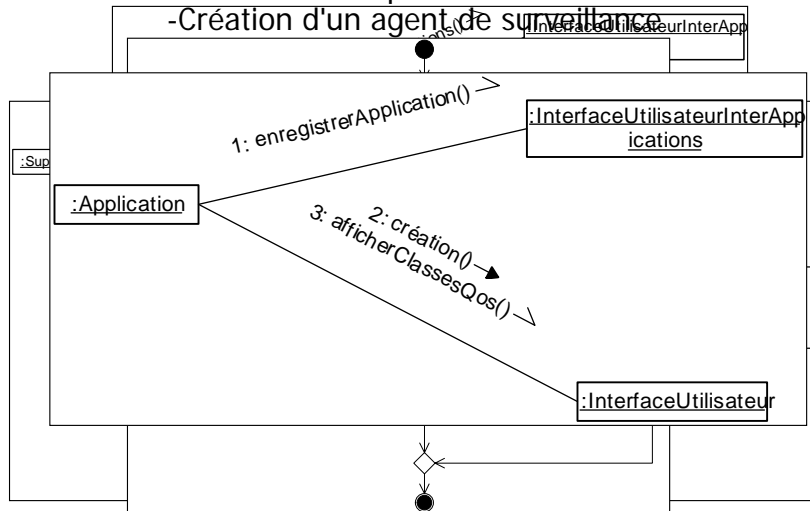




Fonctionnement

Pendant l'exécution de l'application

- Enregistrement des particularités de la plateforme sur laquelle l'application est exécutée
- Choix de la séquence d'actions à effectuer
- Traduction des interfaces utilisateur en "prototypes"
- Création d'un agent de surveillance





Calcul de la classe de QdS la plus adaptée

Principe

1. Parmi toutes les classes du bloc en cours, déterminer les classes disponibles selon la QdS courante
2. Trier l'ensemble obtenu selon l'ordre indiqué par l'utilisateur

Soit A l'ensemble des couples (classes de QdS de l'application, préférence de l'utilisateur):

$$A = \{(a_i, p_i)\} \text{ avec } p_i \in \mathbb{N}, p_i = \{1, n\}, i = 1 \text{ à } n$$

Soit B l'ensemble des classes de QdS du bloc,

$$B \subset A$$

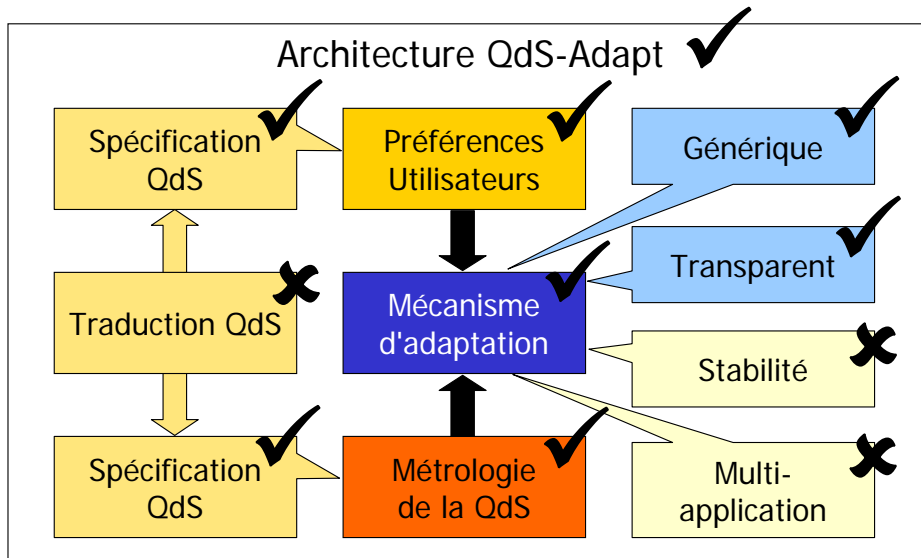
Soit D l'ensemble des classes de QdS de l'application disponibles en fonction de la QdS courante,

$$D \subset A$$

L'ensemble des classes de QdS disponibles pour le bloc est

$$E = B \cap D = \{(e_j, q_j)\} \text{ avec } j = 1 \text{ à } m, m \leq n$$

La classe de QdS la plus adaptée est e_j tel que $q_j = \min_{k=1 \dots m} q_k$

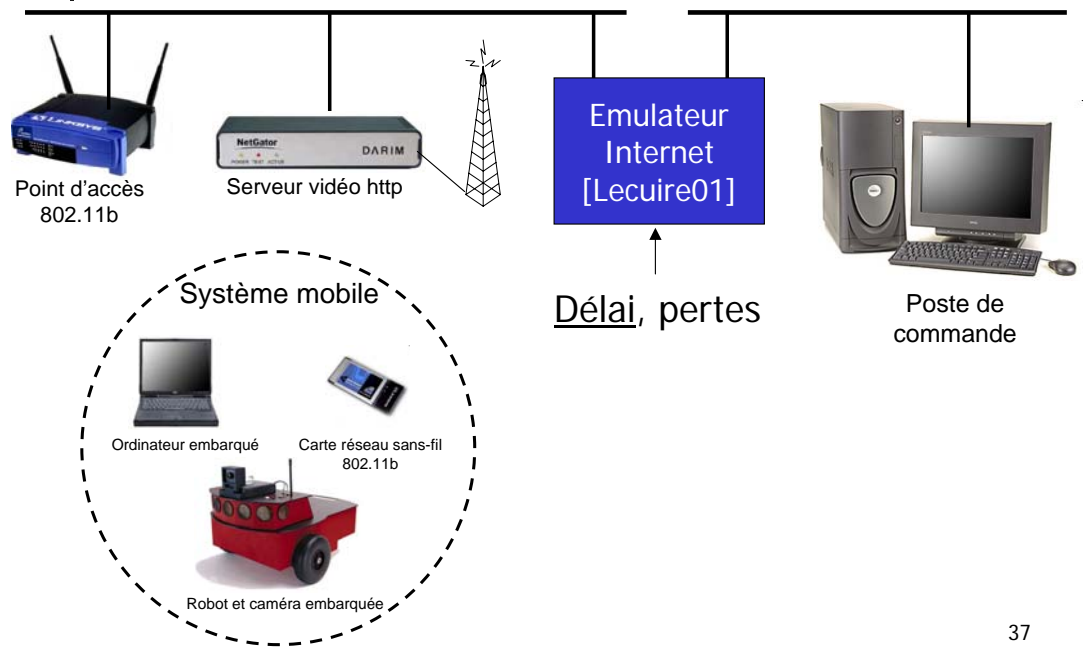


Expérimentations



Télé-pilotage d'un robot mobile
Télé-asservissement en position d'un robot mobile

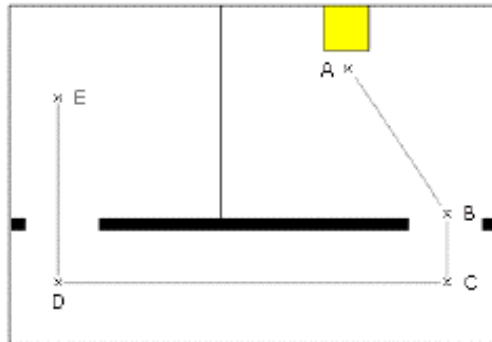
Plate-forme d'expérimentation





Application de télé-pilotage

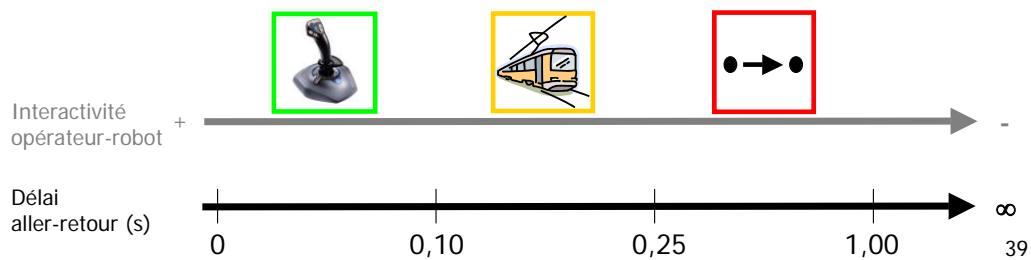
- Application démonstrative
- Suivre le trajet de A à E





Stratégie d'adaptation / Classes de QoS

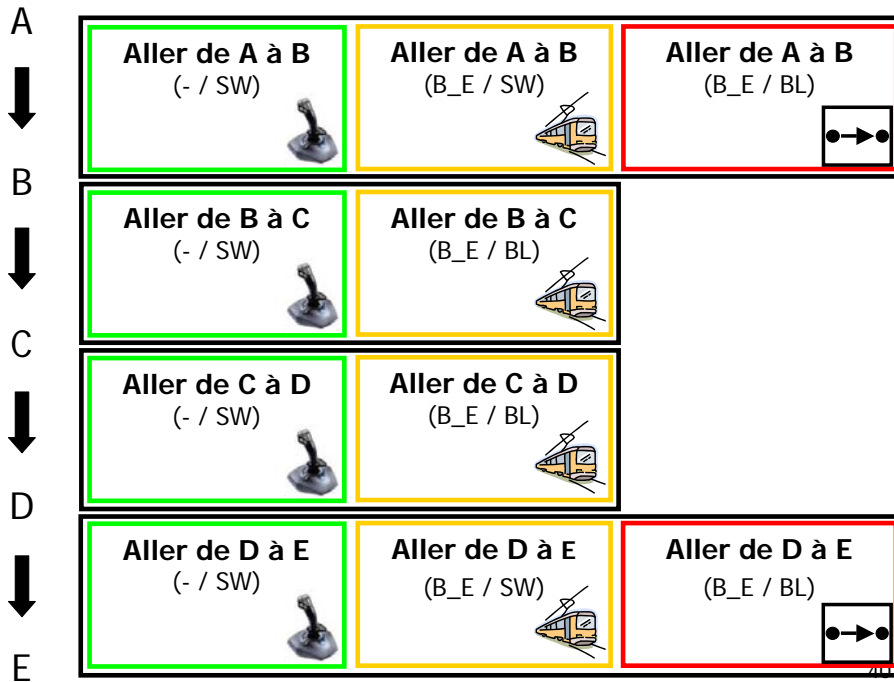
- Degré d'autonomie du robot variable selon l'importance des perturbations [Lin96]
- Modes de pilotage
 - Mode "joystick"
 - Maîtrise complète du robot par l'opérateur (pilotage interactif)
 - Mode "rail"
 - Déplacement le long d'un rail virtuel
 - Maîtrise des degrés de liberté du robot partagée entre le robot et l'opérateur
 - Mode "point à point"
 - Maîtrise de la trajectoire par le robot (interactivité quasi-nulle)





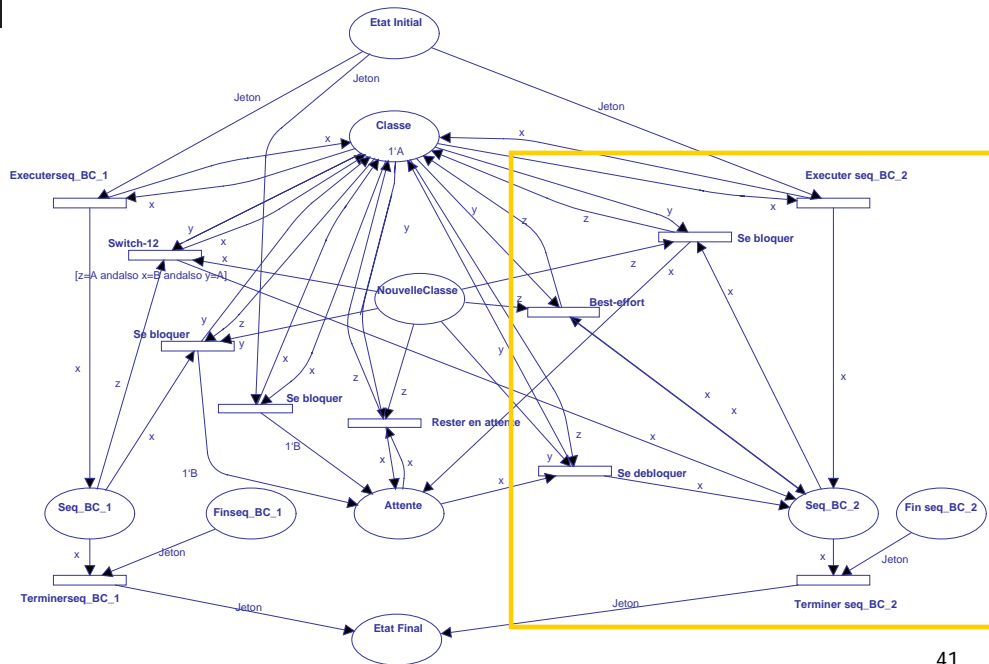
Blocs – Séquences – Actions d'adaptation

Trajet





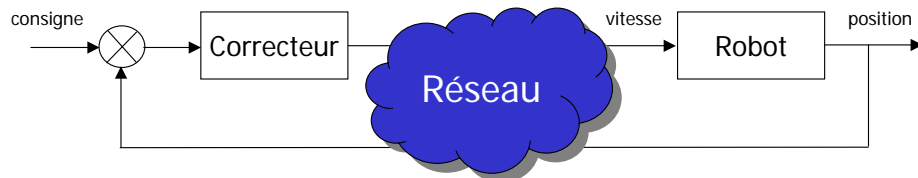
Exemple: Bloc B-C





Application de télé-asservissement

- Asservir la position du robot
- Commande en vitesse



■ Hypothèses

- Délais identiques à l'aller et au retour
- Pas de perte de paquets
- Correcteur proportionnel

$$D_A = D_R = \frac{D_{AR}}{2}$$



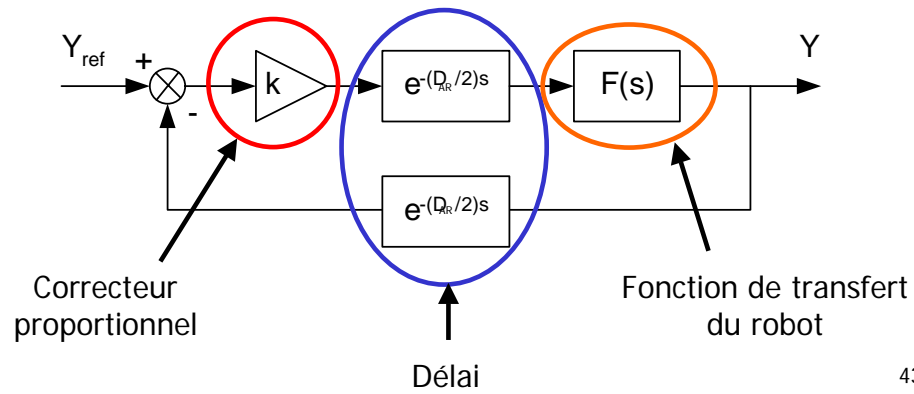
Boucle de commande

- Identification du robot
 - CONTSID [Garnier02]

$$F(s) = \frac{16,6117.k.e^{-(70.10^{-3})s}}{s^3 + 4,0450s^2 + 16,5611s}$$

- Fonction de transfert en boucle fermée

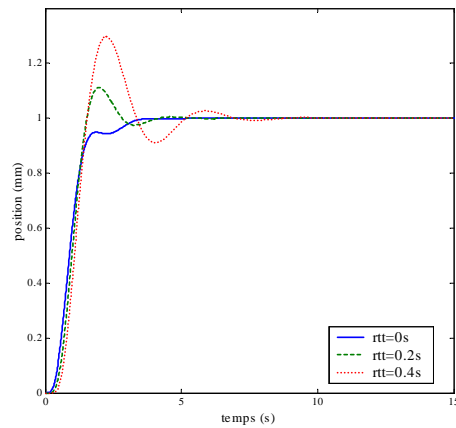
$$FTBF(s) = \frac{16,6117.k.e^{-\left(70.10^{-3} + \frac{D_{AR}}{2}\right)s}}{s^3 + 4,0450s^2 + 16,5611s + 16,6117.k.e^{-(70.10^{-3} + D_{AR})s}}$$





Etude en simulation: Impact du réseau

- Introduction de délai dans la boucle
- Provoque des dépassements

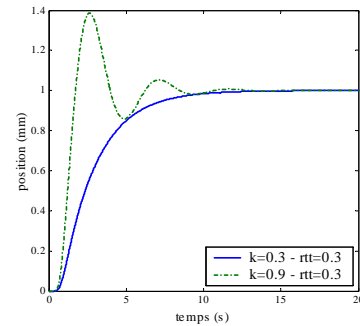


Simulation Matlab



Comment supprimer le dépassement?

- Calcul d'un gain pour délai_{AR} max tel que dépassement nul
- Pas satisfaisant
- Système inutilement lent lorsque le retard est très inférieur au retard max



Gain (k)	0,9	0,7	0,5	0,3
Délai _{AR} max (s)	0,1	0,3	0,5	1,1
t _m (s)	1,38	2,37	3,59	6,53



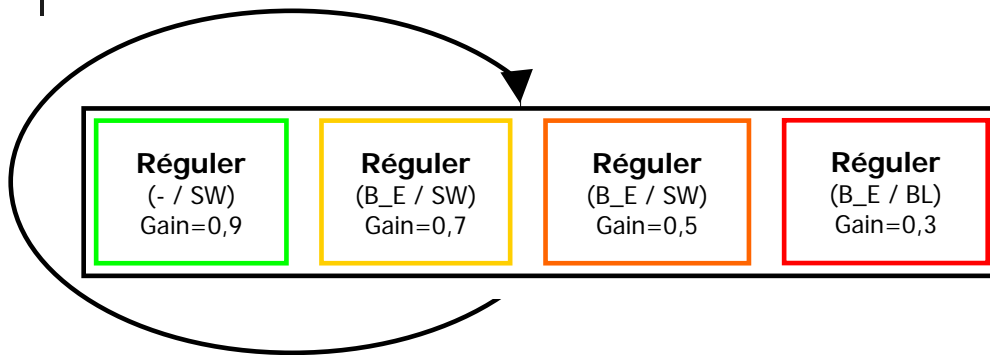
Stratégie d'adaptation / Classes de QdS

- Choix du gain du correcteur en fonction du délai_{AR}
 - Dépassement au détriment de la rapidité du système

Classe de QdS				
Gain (k)	0,9	0,7	0,5	0,3
Délai _{AR} max (s)	0,1	0,3	0,5	1,1



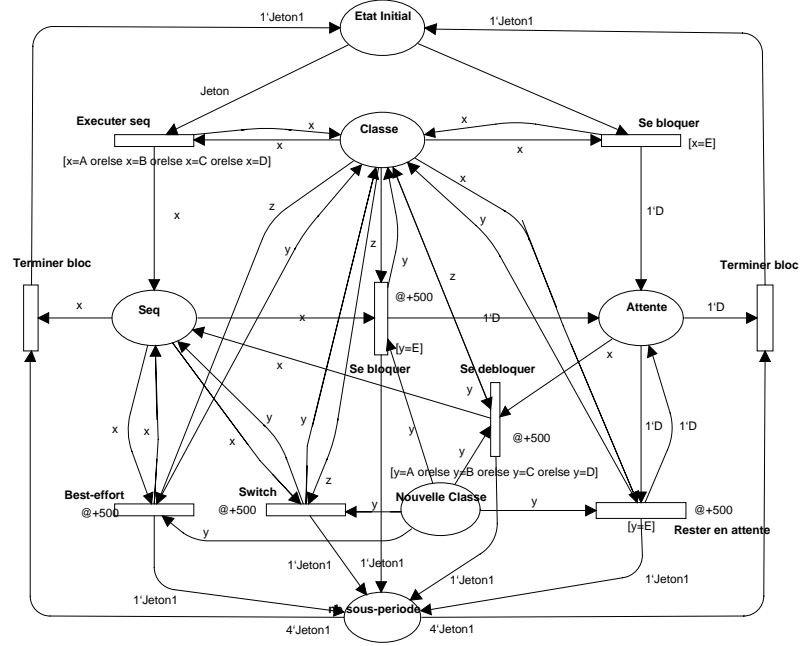
Bloc – Séquences – Actions d'adaptation



- Périodes
 - Choisis en fonction de la dynamique du système
 - Période du bloc: 2s
 - Période du calcul de la classe de QdS la plus adaptée: 0,5s



Modélisation

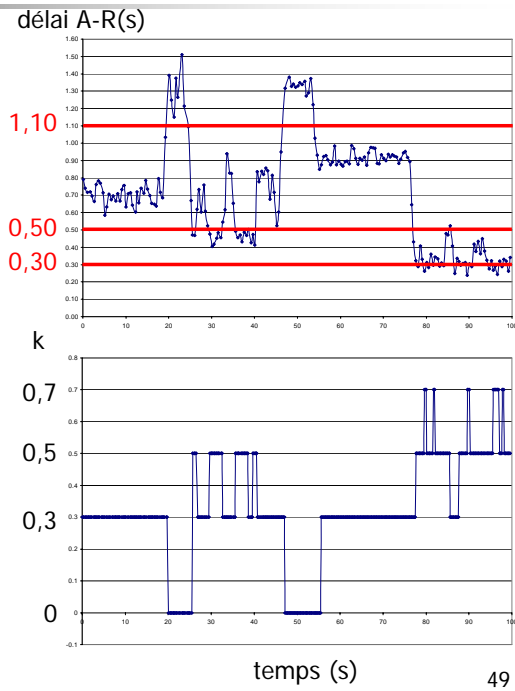




Résultats Expérimentaux

Mesure du
délai aller-retour
(secondes)

Gain

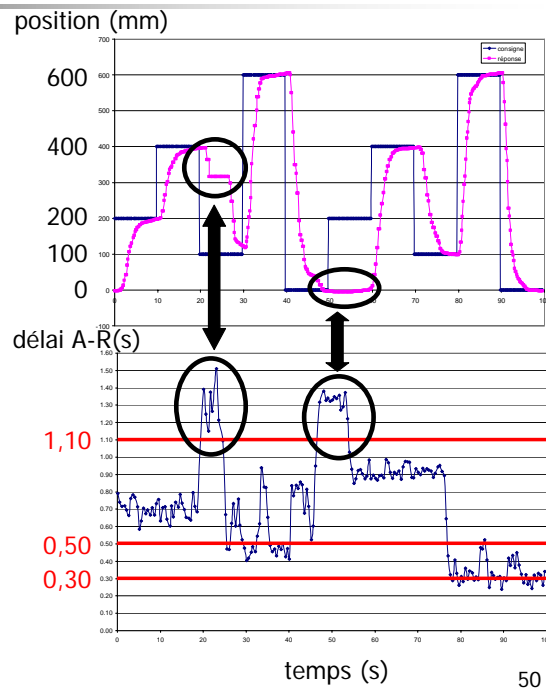




Résultats Expérimentaux (2)

Consigne / Réponse
(mm)

Mesure du
délai aller-retour
(secondes)





Conclusions

- Adaptation en cours d'exécution de l'application en fonction de la QoS mesurée possible
- Contributions
 - Architecture de QoS pour l'adaptation
 - Service de métrologie de la QoS
 - Plate-forme d'expérimentation opérationnelle pour l'accueil et le test de
 - Nouvelles techniques de métrologie
 - Algorithmes de commande évolués pour les systèmes à retard (Equipe Projet Multi-Laboratoires "Retards de Transmission en Télé-opération", CRAN-LAIL-LIRMM)



Conclusions

		XRM [Lazar94]	QoS-A [Campbell98]	Omega [Nahrstedt96]	Heidelberg [Vogt98]	Prayer [Bharghavan97]	QoS-Adapt [Michaut03]
Cadre architectural	Spécification QoS	✓	✓	✓	✓	✗	✓
	Traduction QoS	✓	✓	✓	✓		✗
	Métrologie						✓
	Préférences Utilisateur		✓	✓			✓
Adaptation des applications	Générique					✓	✓
	Transparent					✓	✓
	Stabilité						✗
	Multi-application						✗
Réservation de ressources		✓	✓	✓	✓	✓	intégrable

Oui
 Incomplet
 Non



Conclusions

	Réservation de ressources	Adaptation de l'application
+	<ul style="list-style-type: none">- Garanties strictes de QoS- Performances optimales de l'application	<ul style="list-style-type: none">- Fonctionnement correct de l'application même lorsque la maîtrise des ressources incomplète ou impossible- Orienter les dégradations de la QoS
-	<ul style="list-style-type: none">- Non utilisable pour les réseaux hétérogènes (Internet)- Réservation impossible: Mauvais fonctionnement (voire fonctionnement impossible) de l'application	<ul style="list-style-type: none">- Performances inférieures de l'application- Tolérance aux dégradations limitée (Ex: Action ABORT)



Perspectives

- Classification des applications
 - Modèles de spécification de la QdS
 - Fonctions de traduction de la QdS
 - Politiques de métrologie adaptées
 - Paramètres pertinents, période, etc.
- Métrologie
 - Impact du flux de mesure sur le réseau (finesse des capteurs)
 - Quantifier les incertitudes de mesures
 - Prédiction des mesures
- Stabilité
- Priorités inter-applications