

# Vers une spécification de Mobile IP à l'aide de Mobile Z

**M. Maouche, M. Bettaz**  
**Philadelphia University – Jordan**

## Notere05

30 Aout - 1 Septembre

Gatineau (Québec)

# Plan

- Objectif
- Mobile Z: Motivation
- Travaux similaires
- Mobile Z: Principe & Ingrédients
- Mobile IP: Principe
- Spécification: Schémas Etat & Opération.
- Conclusion

# Objectif

- Montrer l' applicabilité de Mobile Z, 'extension' du langage formel Z, à spécifier des protocoles mobiles.
- Etudes de cas:
  - Réseau cellulaire type GSM.
  - Réseau IP Mobile.

# Mobile Z: Motivation

- Z est un langage de spécification formel
  - Basé Modèle
  - Simple à utiliser
  - Très utilisé en industrie
  - Techniques et Outils de preuves disponibles.
  - Mobilité: peu de tentatives menées.
- Offrir à la communauté Z la capacité de modéliser des systèmes et protocoles mobiles.

# Travaux similaires (1)

## ■ Mobile Unity:

### ■ Unity:

- ✓ Programme = ensemble de composants interconnectés.
- ✓ Composant = déclarations de variables + séquences d'affectations
- ✓ Execution non déterministe et équitable des instructions d'affectation.

### ■ Mobilité:

- ✓ composant, variable, affectation
- ✓ se base sur la notion de position
- ✓  $1 = \text{move}(1)$
- ✓ transient sharing

# Travaux similaires (2)

## ■ Mobile Object-Z

### ■ [K. Taguchi et al/ 2002]

- ✓ unité de mobilité : objet Z
- ✓ se base sur la notion de position
- ✓ opérations de mobilité : Go (adresse), Here (adresse)

### ■ [K. Taguchi et al/2004]

- ✓ unité de mobilité : objet Z
- ✓ Pas de notion de position
- ✓ Intégration du  $\Pi$ -calcul ( reconfiguration dynamique des communications).

# Mobile Z: Principe

## ■ Langage Z:

- Schémas d'état (variables, schemas d'etats, predicats)
- Schemas d'opérations (état avant, état après)
- Mécanismes de structuration (composition, importation,...).

## ■ Mobile Z:

- offre deux opérations génériques de mobilité.
- unité de mobilité: schéma d'état Z.
- se base sur la notion de position.
- migration implicite de l'unité de mobilité.

# Mobile Z: Ingrédients (1)

- Tout schéma d'état mobile comporte une variable *next\_location*.
  - Indique si un schéma d'état mobile est en cours de migration vers une position temporairement stable (position) ou s'est stabilisé (temporairement) dans une position fixe (valeur  $\perp$ ).
- Un schéma d'état modélisant l'état global et la configuration spatiale courante d'un système mobile.
  - Importe les schémas d'états des composants fixes et mobiles.
  - Comporte une ou plusieurs fonctions (au sens Z) qui dénote(n) la configuration spatiale courante.

# Mobile Z: Ingrédients (2)

## ■ Schéma d'opération (générique) *Sense*:

- La valeur de *next\_location* est mise à jour en cas de déplacement en cours du schéma d'état.
- Le corps de ce schéma d'opération dépend fortement de l'application mobile et du mécanisme concret de localisation utilisé.

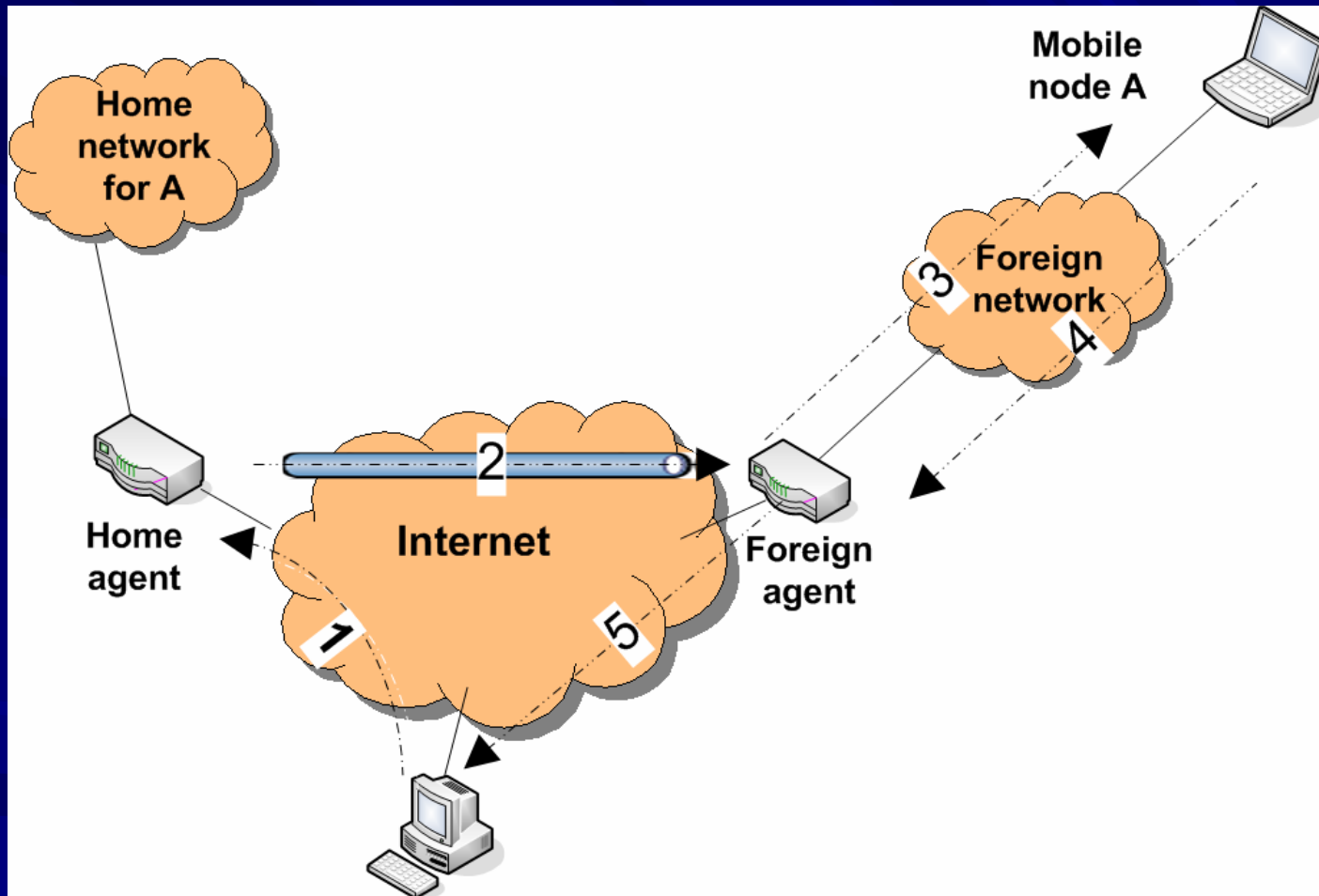
## ■ Schéma d'opération (générique) *Move*:

- Met à jour les fonctions définies dans le schéma d'état global.
- Remet à  $\perp$  la valeur de la variable *next\_location* du schéma d'état si celui-ci s'est stabilisé temporairement dans une position fixe.

# Mobile IP: Principe (1)

- Une station mobile IP peut se déconnecter d'un sous-réseau IP, migrer vers un autre sous réseau IP et se reconnecter sur ce dernier.
- Elle relève d'un sous réseau IP dit natal, une adresse IP natale lui est attribuée.
- Chaque sous réseau IP comporte deux routeurs spéciaux:
  - Routeur *Home*
  - Routeur *Foreign*

# Mobile IP: Principe (2)



# Mobile IP: Principe (3)

- Le protocole IP mobile comporte trois procédures:
  - Procédure de découverte
    - ✓ Détection d'une migration vers un autre sous réseau
    - ✓ Diffusion d'un paquet spécial par le routeur *Foreign*
  - Procédure d'enregistrement
    - ✓ Enregistrement d'une station mobile auprès de son routeur natal *Home* et du routeur *Foreign* du sous réseau auquel elle est actuellement connectée.
    - ✓ Requête/Réponse interaction
  - Procédure d'encapsulation

# Spécification: Paquets IP

*Msg*

*source : IP\_Address*

*dest : IP\_Address*

*type : Type*

*data : Data*

$\forall m : \text{Msg}$

$m.type = \text{Regular} \Rightarrow m.data \in \text{Data\_Regular}$

$m.type = \text{Encapsulated} \Rightarrow m.data \in \text{Data\_Encapsulated}$

$m.type = \text{Request} \Rightarrow m.data \in \text{Data\_Request}$

$m.type = \text{Reply} \Rightarrow m.data \in \text{Data\_Reply}$

$m.type = \text{Advertisement} \Rightarrow m.data \in \text{Data\_Advertisement}$

# Spécification: interface de communication

*Channel\_interface*

*in : seq Msg*

*out : seq Msg*

*gotflag : Bit*

*channel : Message*

*channel = ◦ ⇒ gotflag = 0*

*# out ≤ Max\_Size*

*# in ≤ Max\_Size*

# Spécification : Station Mobile

*Mobile\_Node*

*mid : Mid*

*next\_location : Number*

*home-\_address : IP\_Address*

*care-of\_address : IP\_Address*

*interface : Channel\_interface*

*confirmed : Bit*

$\forall m : \text{Mobile\_Node} \bullet \#(m.\text{interface.out}) = 1$

# Spécification : Sous-réseau IP

*Subnetwork*

*sid : Integer*

*h : Home\_Agent*

*f : Foreign\_Agent*

*interface : Channel\_interface*

*h.address.subnet = sid*

*f.address.subnet = sid*

# Spécification : Réseau mobile IP

*Network*

*Subnetwork*

*Mobile\_Node*

*located* : *Mobile\_Node*  $\mapsto$  *Subnetwork*

*moving* : *Mobile\_Node*  $\mapsto$  *Subnetwork*

$\forall m : \text{Mobile\_Node}$

$m \in \text{dom } \textit{located} \Rightarrow (m.\textit{next\_location} = \perp) \wedge$   
 $(m?\textit{confirmed} = 1)$

$m \in \text{dom } \textit{moving} \Rightarrow (m.\textit{next\_location} \neq \perp)$

$\forall m : \text{Mobile\_Node}, \forall s : \text{Subnetwork}$

$((m \mapsto s) \in \textit{located}) \vee (m \mapsto s) \in \textit{located} \Rightarrow$   
 $m.\textit{interface.channel} = s.\textit{interface.channel}$

# Spécification : Sense

*Sense*

$\Delta$  *Mobile\_Node*

$\Xi$  *Message*

*m?* : *Mobile\_Node*

*msg?* : *Msg*

*msg?.type = Advertisement*

*msg?.data.foreign\_flag = 1*

*msg?.source  $\neq$  care\_of\_address*

*(m?.next\_location)' = msg?.source.subnet*

# Spécification : Move

*Move*

$\Delta$  *Network*

$\Xi$  *Subnetwork*

$\Delta$  *Mobile\_Node*

$m? : \textit{Mobile\_Node}$

$\exists s : \textit{Subnetwork} \bullet m?.\textit{next\_location} = s.\textit{sid}$

$m?.\textit{next\_location} \neq \perp$

$m?.\textit{confirmetd} = 1$

$(m?.\textit{care\_of\_address} = s.\textit{sid}) \Rightarrow$

$(\textit{located}' = \textit{located} \cup \{m? \mapsto s.\textit{sid}\}) \wedge (\textit{moving}' = \textit{moving} \setminus \{m? \mapsto s.\textit{sid}\})$

$\wedge (m?.\textit{next\_location} = \perp)$

$m?.\textit{care\_of\_address} \neq s.\textit{sid}) \Rightarrow$

$\textit{located}' = \textit{located} \setminus \{m? \mapsto s.\textit{sid}\}) \wedge (\textit{moving}' = \textit{moving} \cup \{m? \mapsto s.\textit{sid}\})$

# Spécification : Autres opérations

*Process\_Registration\_Reply*

$\Delta$  *Network*

$\exists$  *Message*

*m?* : *Mobile\_Node*

*msg?* : *Msg*

$msg?.type = Reply$

$msg?.data.answer = 1 \wedge msg?.data.id = m?.out.data.id$

$\Rightarrow (m?.confirmed' = 1) \wedge (m?.care\_of\_address)' = msg?.source$

$msg?.data.answer = 0 \wedge msg?.data.id = m?.out.data.id$

$\Rightarrow m?.care\_of\_address' = \square$

# Spécification : Autres opérations

*Get\_From\_Channel*

$\Delta$  *Channel\_interface*

*c?* : *Channel\_interface*

$c?.channel \neq \circ$

$c?.in' = (c?.in).(c?.channel)$

$(c?.channel).dest \neq All \Rightarrow c?.channel' = \circ$

$(c?.channel).dest = All \wedge c?.gotflag = 0 \Rightarrow$

$(c?.channel' = c?.channel) \wedge c?.gotflag' = 1$

# Spécification : Autres opérations

*Receive\_From\_Network*

$\Delta$  *Channel\_interface*

*c?* : *Channel\_interface*

*msg!* : *Msg*

*c?.in*  $\neq \diamond$

*msg!* = *head* (*c?.in*)

*c?.in'* = *tail* (*c?.in*)

# Spécification : Autres opérations

*Put\_In\_Channel*

$\Delta$  *Channel\_interface*

*c?* : *Channel\_interface*

*c?.channel* =  $\circ$

*c?.out*  $\neq \langle \rangle$

*c?.channel'* = *head* (*c?.out*)  $\vee$  ( *c?.channel'* =  $\circ$   $\wedge$  *c?.gotflag'* = 0)

*c?.out'* = *tail* (*c?.out*)

# Spécification : Autres opérations

*Send\_To\_Network*

$\Delta$  *Channel\_interface*

*c?* : *Channel\_interface*

*msg?* : *Msg*

# *c?.out* < *Max\_Size*

*c?.out'* = (*c?.out*). *msg?*

# Vérification de Propriétés

- Techniques et Outils de preuve usuels de Z, associé à une logique (temporelle ou autre).
- Propriétés liées a la mobilité.
- Exemple de Propriété :  
‘Toute station mobile loin de son sous-réseau natal et n’ayant pas encore reçu un paquet *Advertisement* devra, dans un état prochain, soit demeurer à la même position, soit migrer vers une autre position soit recevoir un paquet *Advertisement*.’

# Conclusion

- Applicabilité de Mobile Z à modéliser des systèmes et protocoles mobiles, en particulier le protocole IP mobile.
- Vérification de propriétés IP mobile en cours.
- Travaux futurs: Formalisation, à l'aide de la théorie des catégories, du 'mapping' entre Mobile Z et Graph Transformation.