



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



Vers un Internet des Objets complètement programmable



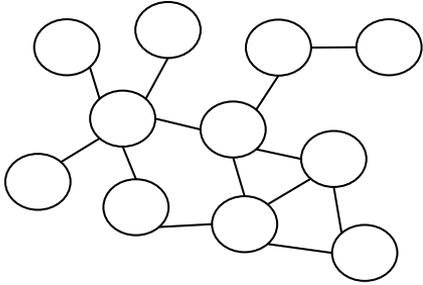
Amaury Bruniaux ^{1,2}, Julien Montavont ², Thomas Noel ²,
Georgios Z. Papadopoulos ¹, Nicolas Montavont ¹

Journées non thématiques
du GDR RSD, Lyon 2023

¹ IMT Atlantique, IRISA

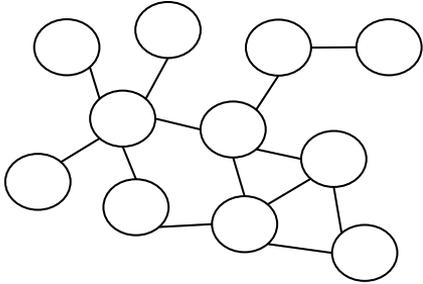
² ICube Laboratory (CNRS), University of Strasbourg

Contact: amaury.bruniaux@imt-atlantique.fr



- Les capteurs sont contraints en mémoire, puissance et énergie
- Les liens radios ne sont pas fiables
- Longue autonomie
- Potentielle difficulté d'accès après le déploiement

Réseaux radios maillés



- Les capteurs sont contraints en mémoire, puissance et énergie
- Les liens radios ne sont pas fiables
- Longue autonomie
- Potentielle difficulté d'accès après le déploiement

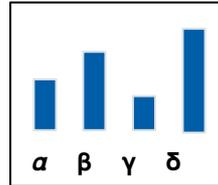
À chaque situation {protocoles, settings, environment} son compromis de performances.

Configuration

Environment 1

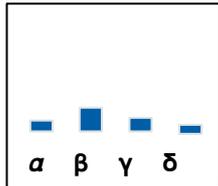
Protocol stack	Settings
Protocol A	7, c, 35
Protocol F	300, 5
Protocol K	18, yes

KPI

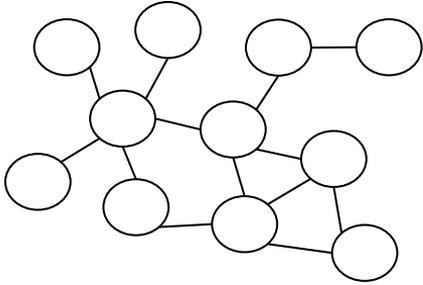


Environment 2

Protocol stack	Settings
Protocol A	7, c, 35
Protocol F	300, 5
Protocol K	18, yes



Réseaux radios maillés



- Les capteurs sont contraints en mémoire, puissance et énergie
- Les liens radios ne sont pas fiables
- Longue autonomie
- Potentielle difficulté d'accès après le déploiement

À chaque situation {protocols, settings, environment} son compromis de performances.

Configuration

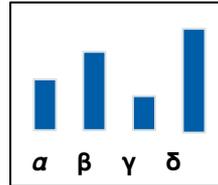
KPI

Configuration

KPI

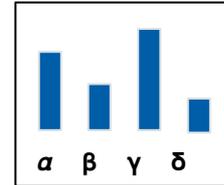
Environment 1

Protocol stack	Settings
Protocol A	7, c, 35
Protocol F	300, 5
Protocol K	18, yes



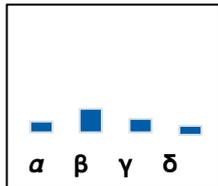
Environment 1

Protocol stack	Settings
Protocol A	4, b, 42
Protocol G	150, 8
Protocol K	18, no

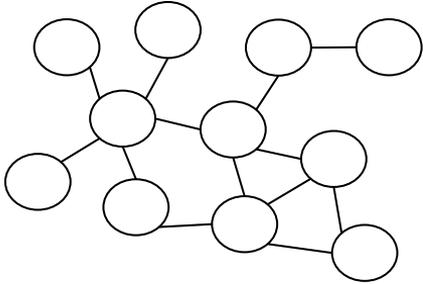


Environment 2

Protocol stack	Settings
Protocol A	7, c, 35
Protocol F	300, 5
Protocol K	18, yes



Réseaux radios maillés



- Les capteurs sont contraints en mémoire, puissance et énergie
- Les liens radios ne sont pas fiables
- Longue autonomie
- Potentielle difficulté d'accès après le déploiement

À chaque situation {protocols, settings, environment} son compromis de performances.

Configuration

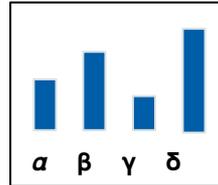
KPI

Configuration

KPI

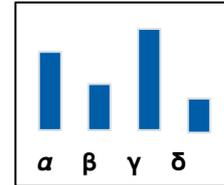
Environment 1

Protocol stack	Settings
Protocol A	7, c, 35
Protocol F	300, 5
Protocol K	18, yes



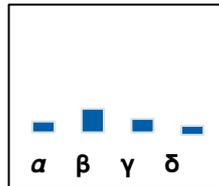
Environment 1

Protocol stack	Settings
Protocol A	4, b, 42
Protocol G	150, 8
Protocol K	18, no



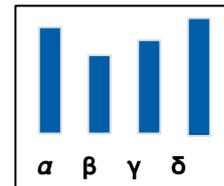
Environment 2

Protocol stack	Settings
Protocol A	7, c, 35
Protocol F	300, 5
Protocol K	18, yes



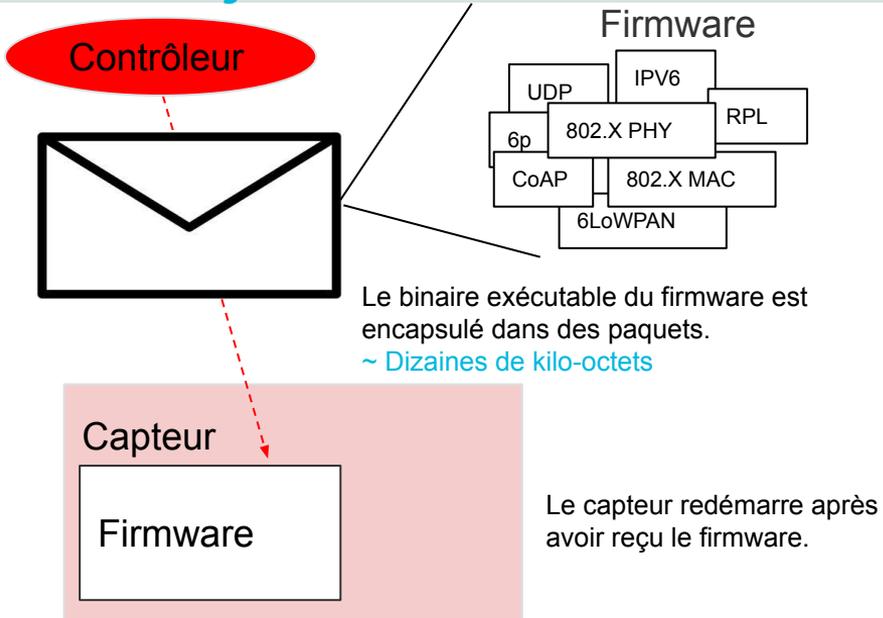
Environment 1

Protocol stack	Settings
Protocol B	24, ack
Protocol F	21, c, 4
Protocol L	34, 22



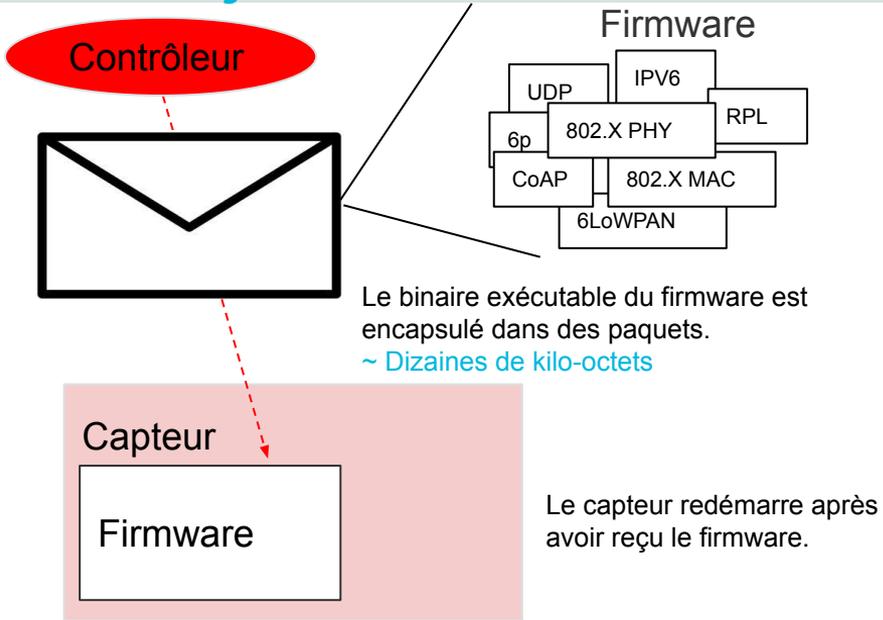
Programmabilité dans l'IoT

Mises à jour de firmware



Programmabilité dans l'IoT

Mises à jour de firmware

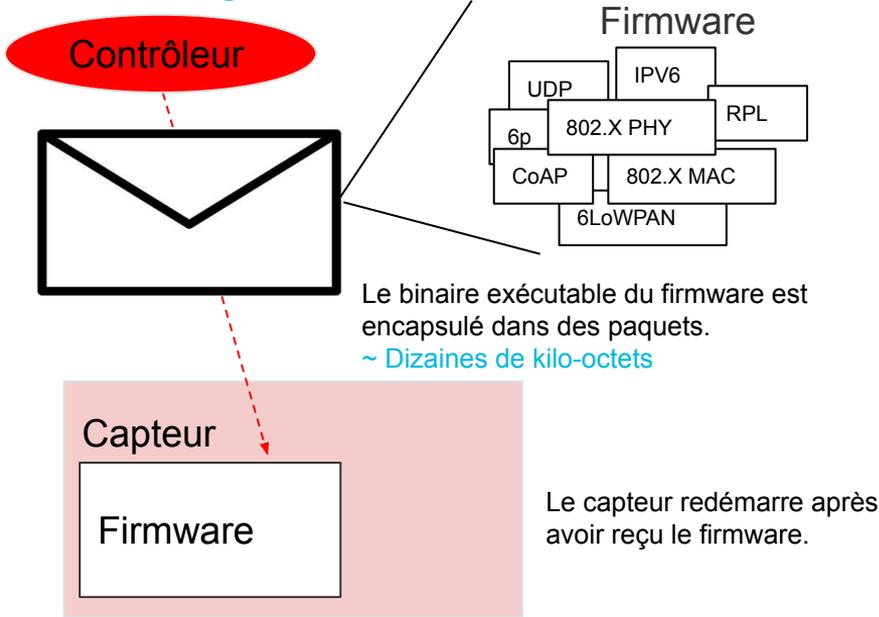


- Redémarrage => Perte de l'état
- Mise à jour volumineuse => Interruption de service conséquente

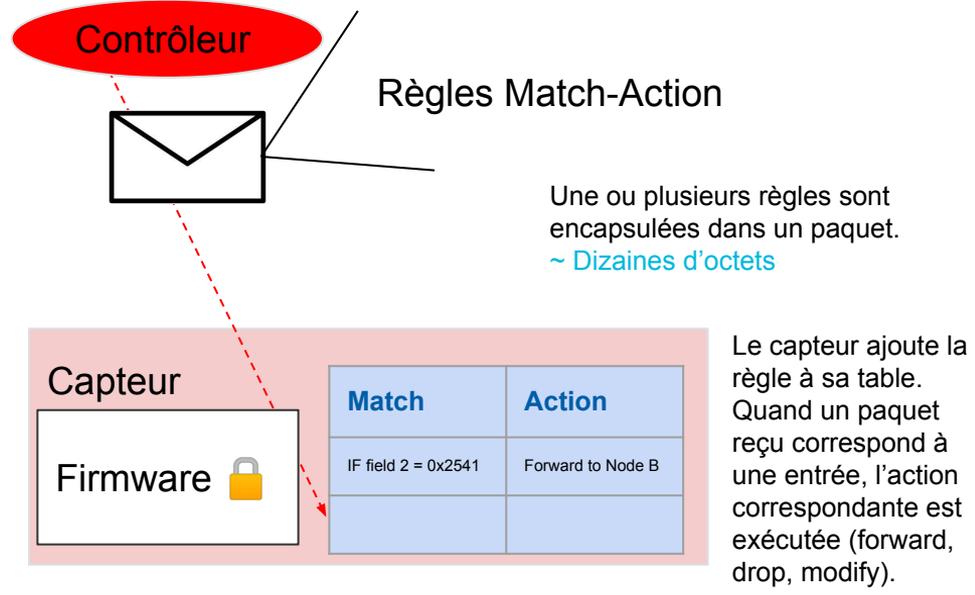
- Toutes les parties du comportement de l'appareil peuvent être programmées

Programmabilité dans l'IoT

Mises à jour de firmware



Software Defined Networks

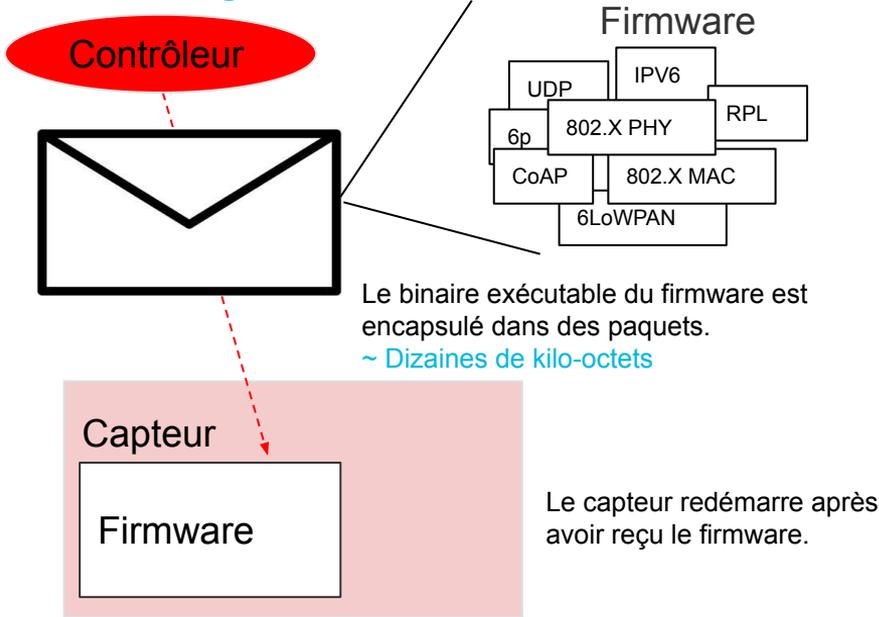


- Redémarrage => Perte de l'état
- Mise à jour volumineuse => Interruption de service conséquente

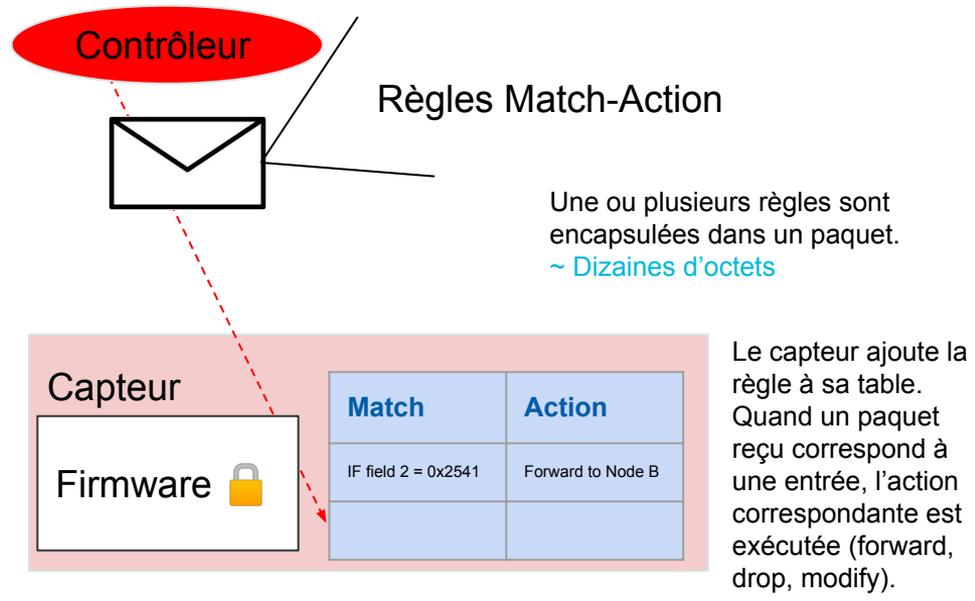
- Toutes les parties du comportement de l'appareil peuvent être programmées

Programmabilité dans l'IoT

Mises à jour de firmware



Software Defined Networks



- Redémarrage => Perte de l'état
- Mise à jour volumineuse => Interruption de service conséquente

- Toutes les parties du comportement de l'appareil peuvent être programmées

- Programmation à chaud

- Limité à la couche réseau ou l'allocation de ressources
- Limited statefulness

- Toutes les parties du comportement de l'appareil peuvent être programmées
- Programmation à chaud

Propriétés désirables pour la programmabilité complète

- Toutes les parties du comportement de l'appareil peuvent être programmées
- Le paquet de mise à jour doit être compact
- Les informations en mémoire sont utilisables à tout niveau du programme
- Programmation à chaud
- Programmation incrémentale
- Permettre la programmation de tout comportement supporté par le matériel

Controler



Mise à jour de
bytecode

La pile protocolaire est aplatie
et traduite en un bytecode.

Trois types de mises à jour:

- monolithique ~ Centaines d'octets
- paramétrique ~ Quelques octets
- incrémentale ~ Dizaines d'octets

À l'exécution, le firmware lit les
instructions décrivant notamment
les protocoles de la pile
protocolaire et les exécute

Node

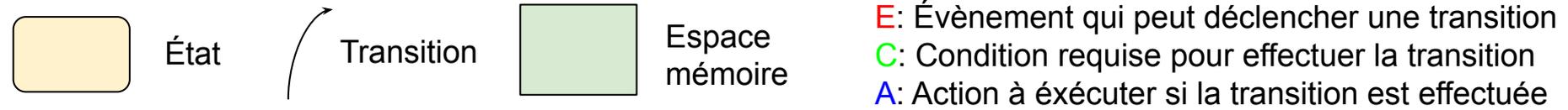
Firmware incluant
l'Interprète du
bytecode 

Bytecode

Propriétés désirables pour la programmabilité complète

- Toutes les parties du comportement de l'appareil peuvent être programmées
- Le paquet de mise à jour doit être compact
- Les informations en mémoire sont utilisables à tout niveau du programme
- Programmation à chaud
- Programmation incrémentale
- Permettre la programmation de tout comportement supporté par le matériel

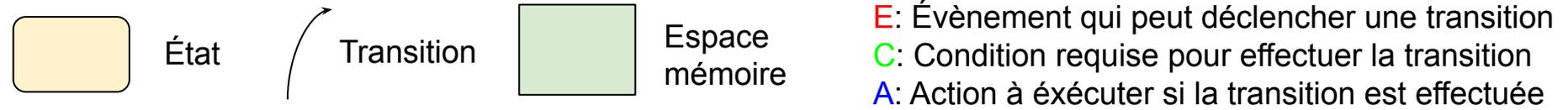
Extended Finite State Machine



Une XFSM est une FSM combinée à de la mémoire, ce qui la rend Turing-complète. Tous les évènements, conditions et actions sont basés sur des abstractions qui généralisent les capacités d'un appareil communiquant.

I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano and F. Gringoli, "Wireless MAC processors: Programming MAC protocols on commodity Hardware," 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 2012, pp. 1269-1277, doi: 10.1109/INFOCOM.2012.6195488.

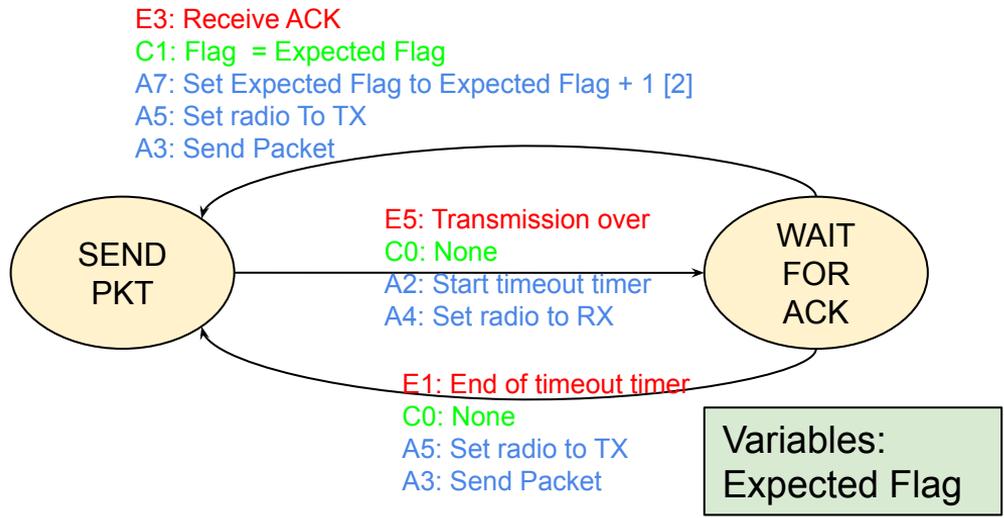
Extended Finite State Machine



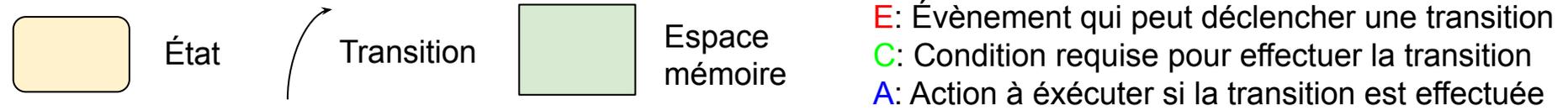
Une XFSM est une FSM combinée à de la mémoire, ce qui la rend Turing-complète. Tous les évènements, conditions et actions sont basés sur des abstractions qui généralisent les capacités d'un appareil communiquant.

I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano and F. Gringoli, "Wireless MAC processors: Programming MAC protocols on commodity Hardware," 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 2012, pp. 1269-1277, doi: 10.1109/INFOCOM.2012.6195488.

Events	Actions
VIRTUAL	START_TIMER
TIMER_EXPIRES	STOP_TIMER
CCA_BUSY	REFRESH_TIMER
CCA_FREE	CCA
COLLISION	SET_VAR
RX_COMPLETE	ADD_ROW
TX_COMPLETE	COPY_PACKET
MEASURE_COMPLETE	DEQUEUE
TX_MODE_SWITCH	MEASURE
RX_MODE_SWITCH	SET_FIELD
RADIO_SLEEP_SWITCH	INCREASE_VAR
Conditions	RADIO_TX_MODE
NO_CONDITION	RADIO_RX_MODE
GREATER	RADIO_SLEEP_MODE
EQUALS	EMPTY_RX_BUFFER
QUEUE_LONGER	SEND_PKT
QUEUE_SHORTER	PARSE_PKT
RADIO_MODE_IS	COPY_TEMPLATE

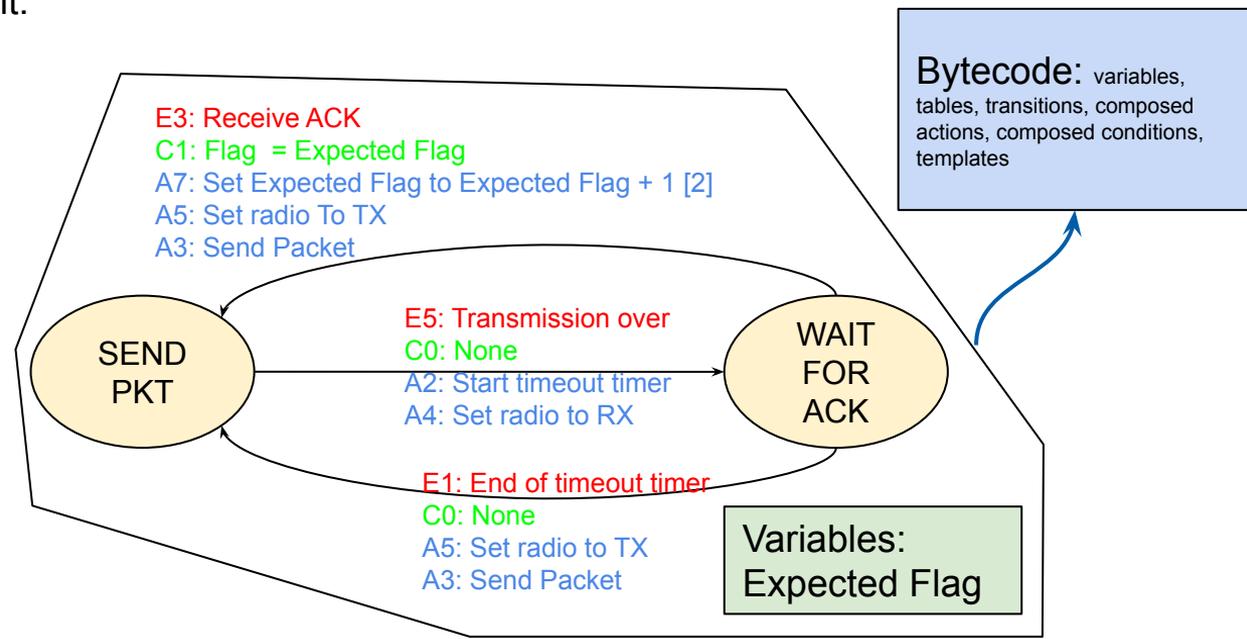


Extended Finite State Machine



Une XFSM est une FSM combinée à de la mémoire, ce qui la rend Turing-complète. Tous les évènements, conditions et actions sont basés sur des abstractions qui généralisent les capacités d'un appareil communiquant.

Events	Actions
VIRTUAL	START_TIMER
TIMER_EXPIRES	STOP_TIMER
CCA_BUSY	REFRESH_TIMER
CCA_FREE	CCA
COLLISION	SET_VAR
RX_COMPLETE	ADD_ROW
TX_COMPLETE	COPY_PACKET
MEASURE_COMPLETE	DEQUEUE
TX_MODE_SWITCH	MEASURE
RX_MODE_SWITCH	SET_FIELD
RADIO_SLEEP_SWITCH	INCREASE_VAR
Conditions	RADIO_TX_MODE
NO_CONDITION	RADIO_RX_MODE
GREATER	RADIO_SLEEP_MODE
EQUALS	EMPTY_RX_BUFFER
QUEUE_LONGER	SEND_PKT
QUEUE_SHORTER	PARSE_PKT
RADIO_MODE_IS	COPY_TEMPLATE



- Les besoins et conditions changent
=> la pile protocolaire doit s'adapter
- La taille du paquet de mise à jour est un point clé
- L'abstraction des capacités physiques permet la programmabilité complète

Prochaines étapes / Perspectives

- Bootstrapping et robustesse de la mise à jour
- Intégration de la couche physique (Software-Defined Radio)
- Réseaux auto-programmables