Les risques d'OpenFlow et du SDN

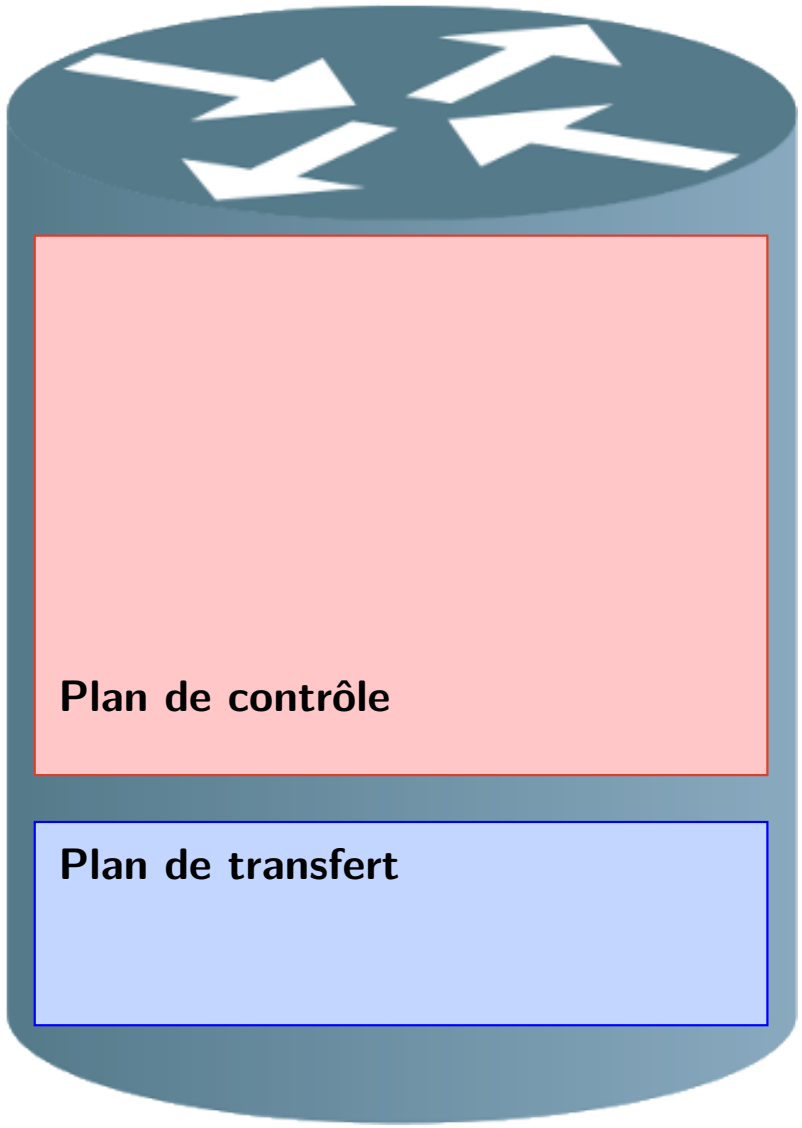
Maxence Tury
maxence.tury@ssi.gouv.fr

Agence nationale de la sécurité des systèmes d'information

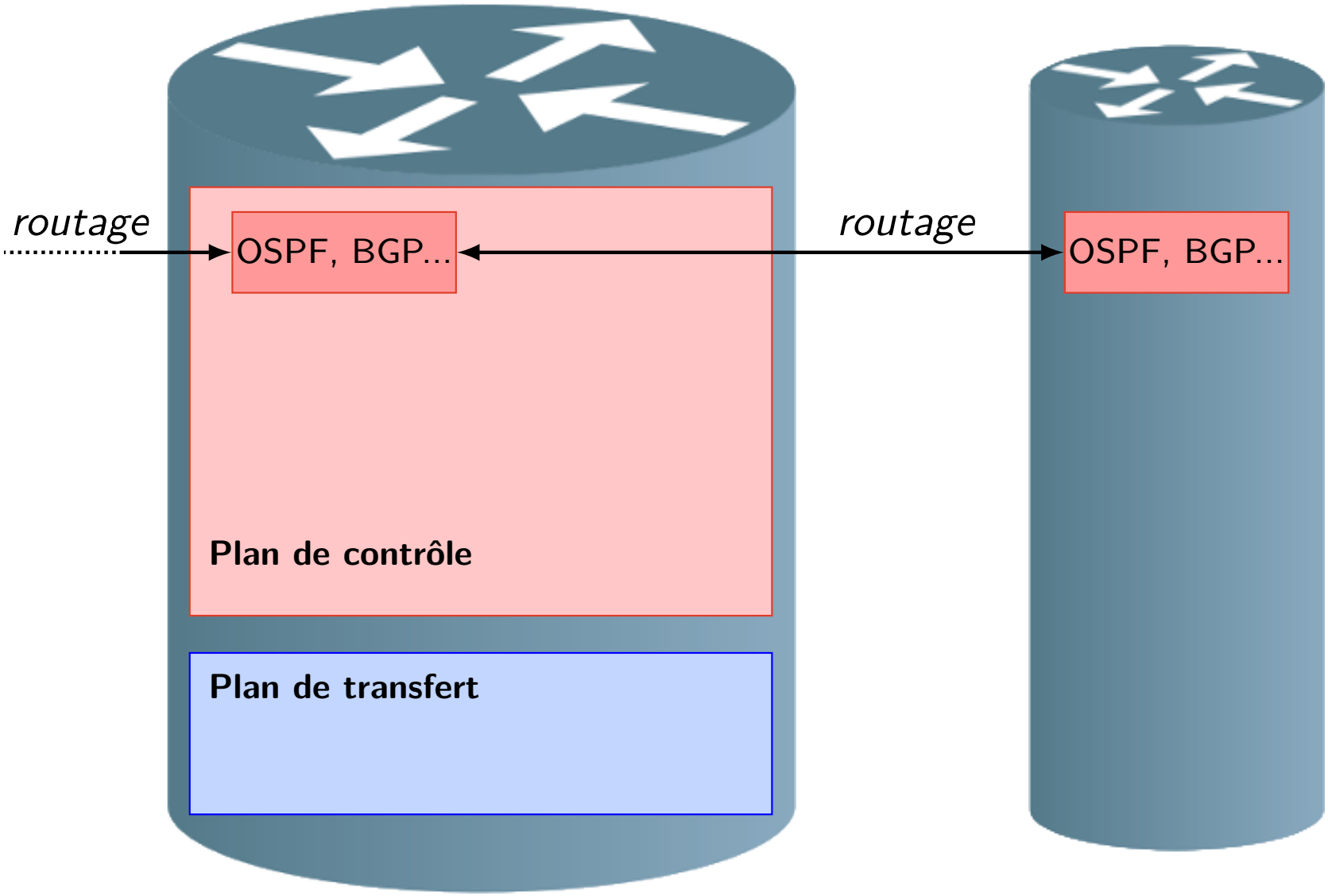
4 juin 2015

Le paradigme Software-Defined Networking

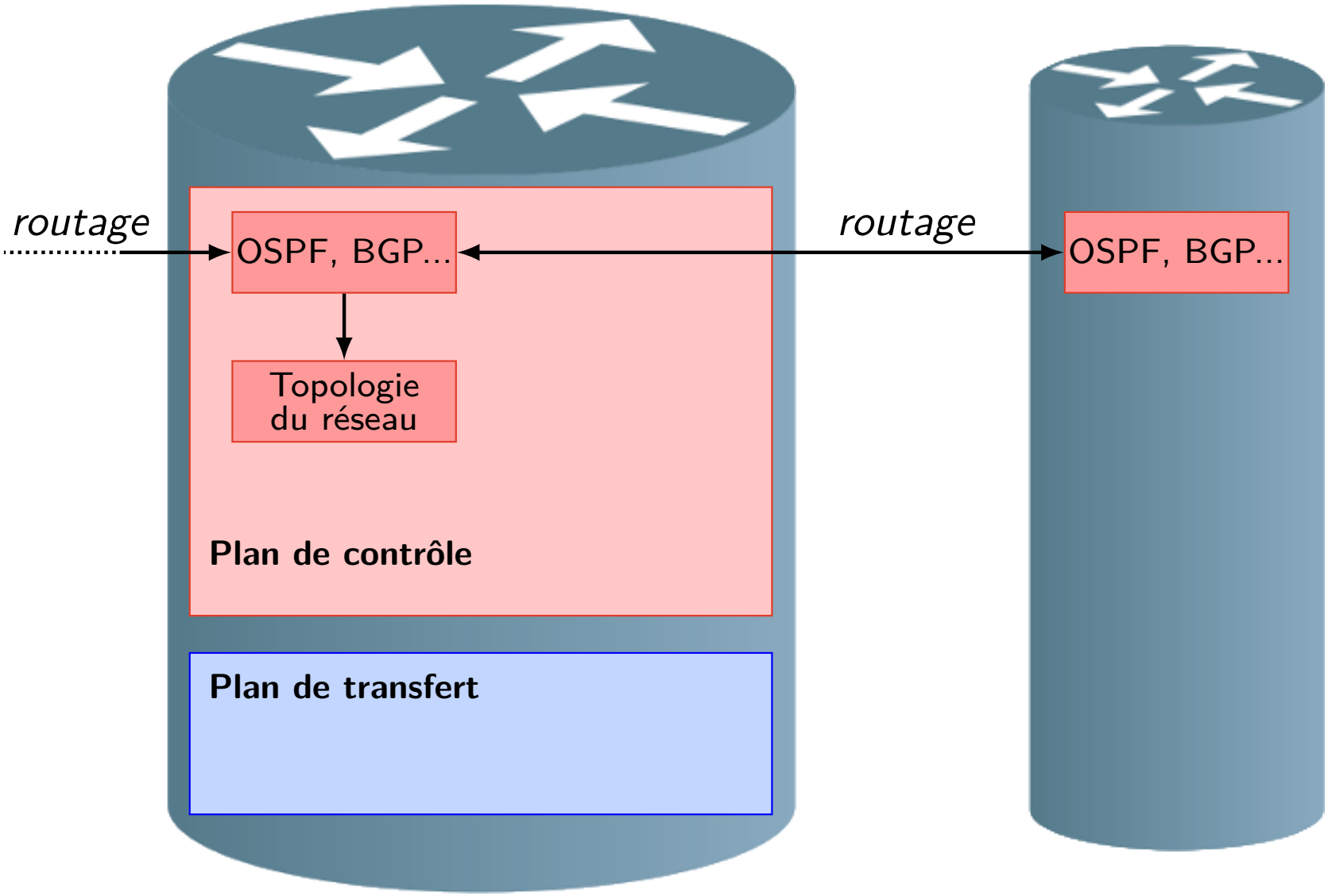
Routage traditionnel



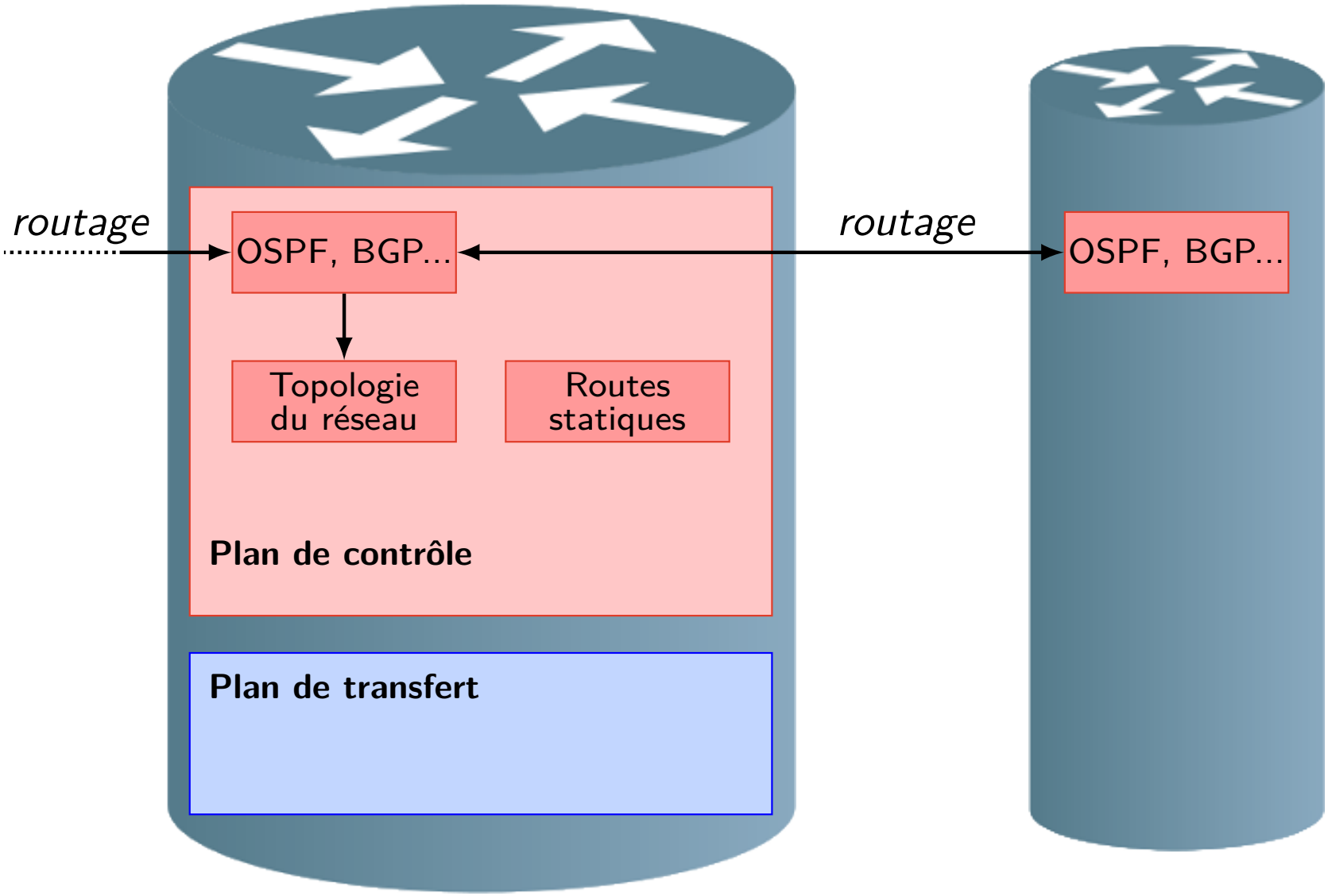
Routage traditionnel



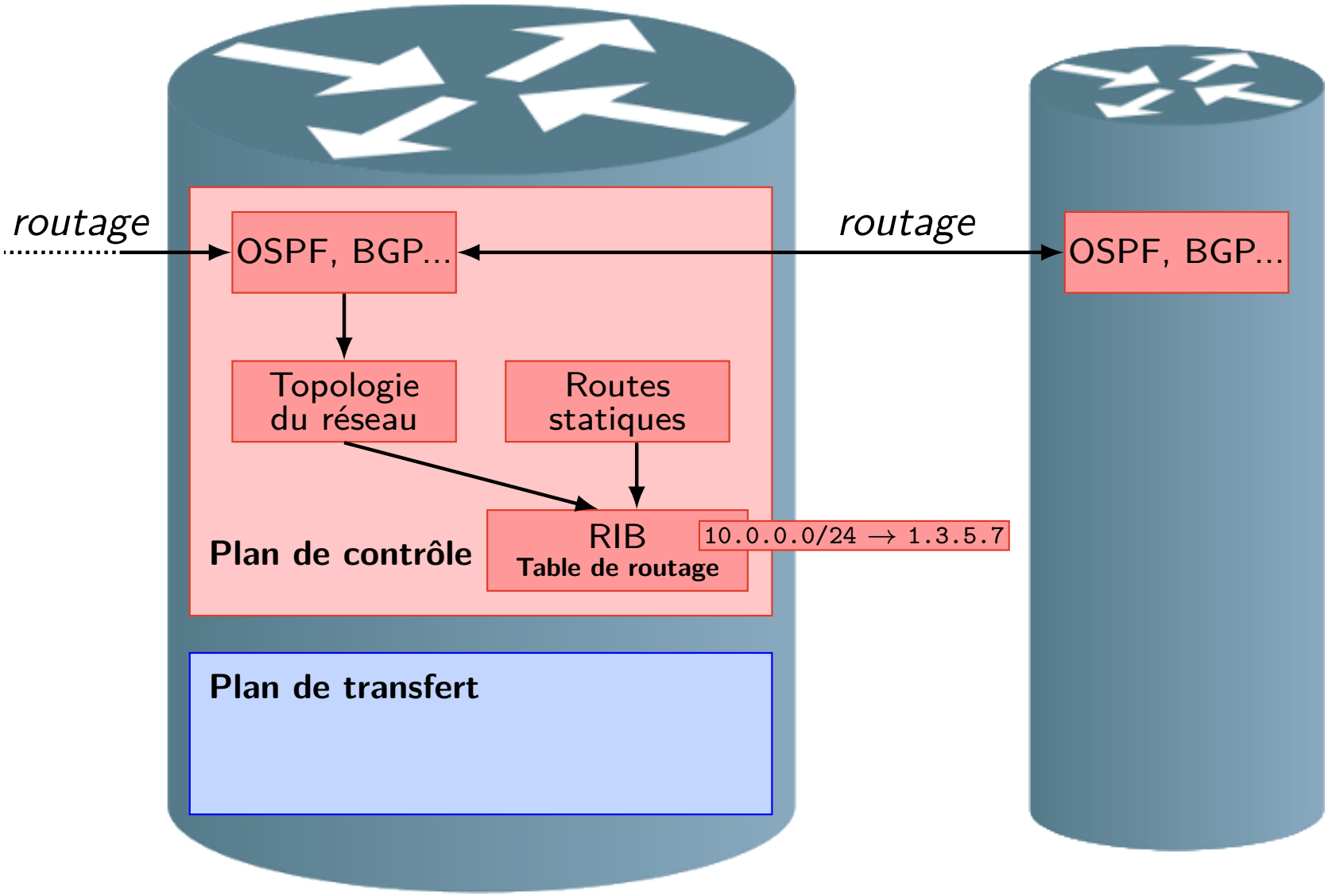
Routage traditionnel



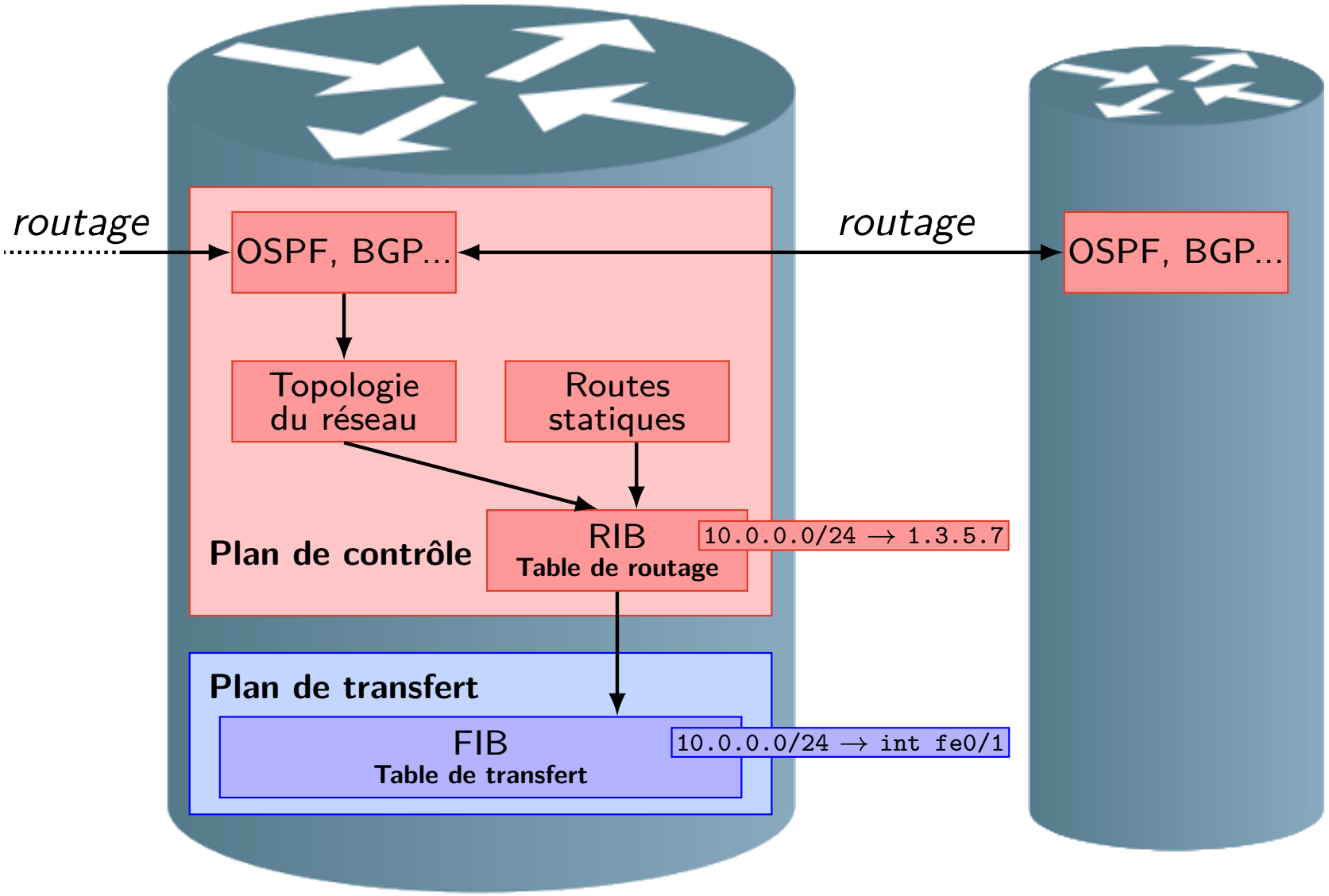
Routage traditionnel



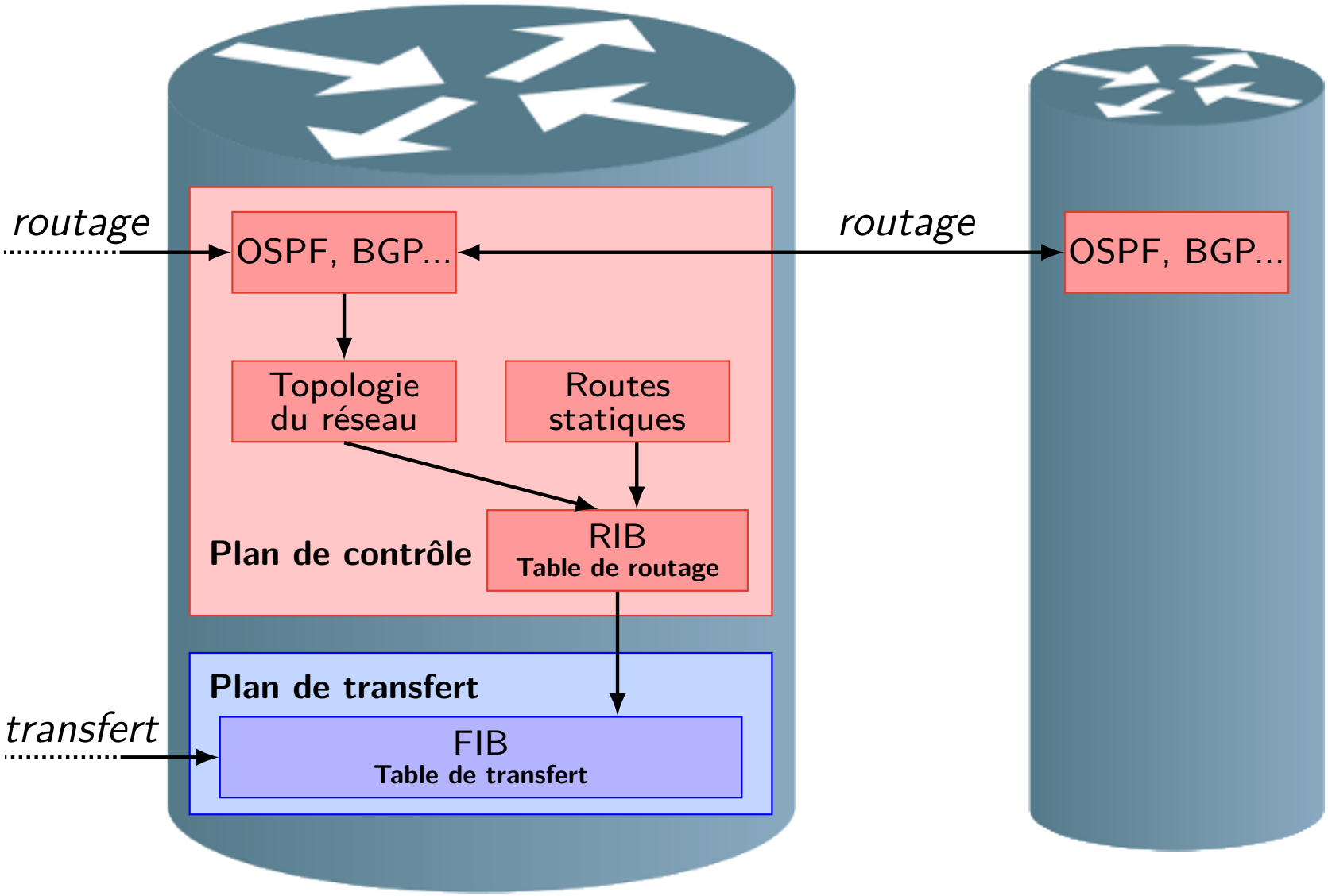
Routage traditionnel



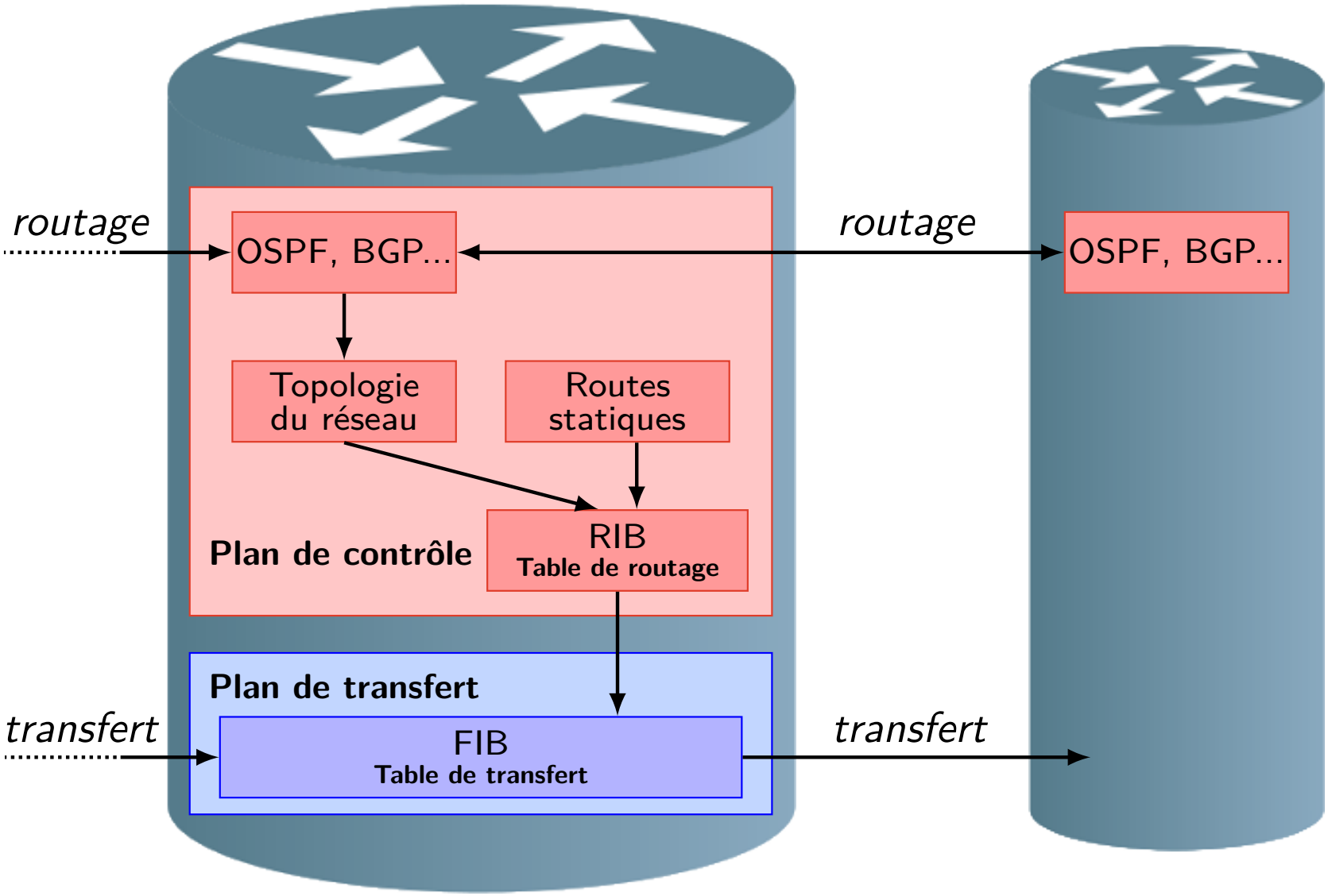
Routage traditionnel



Routage traditionnel



Routage traditionnel



Software-Defined Networking ?

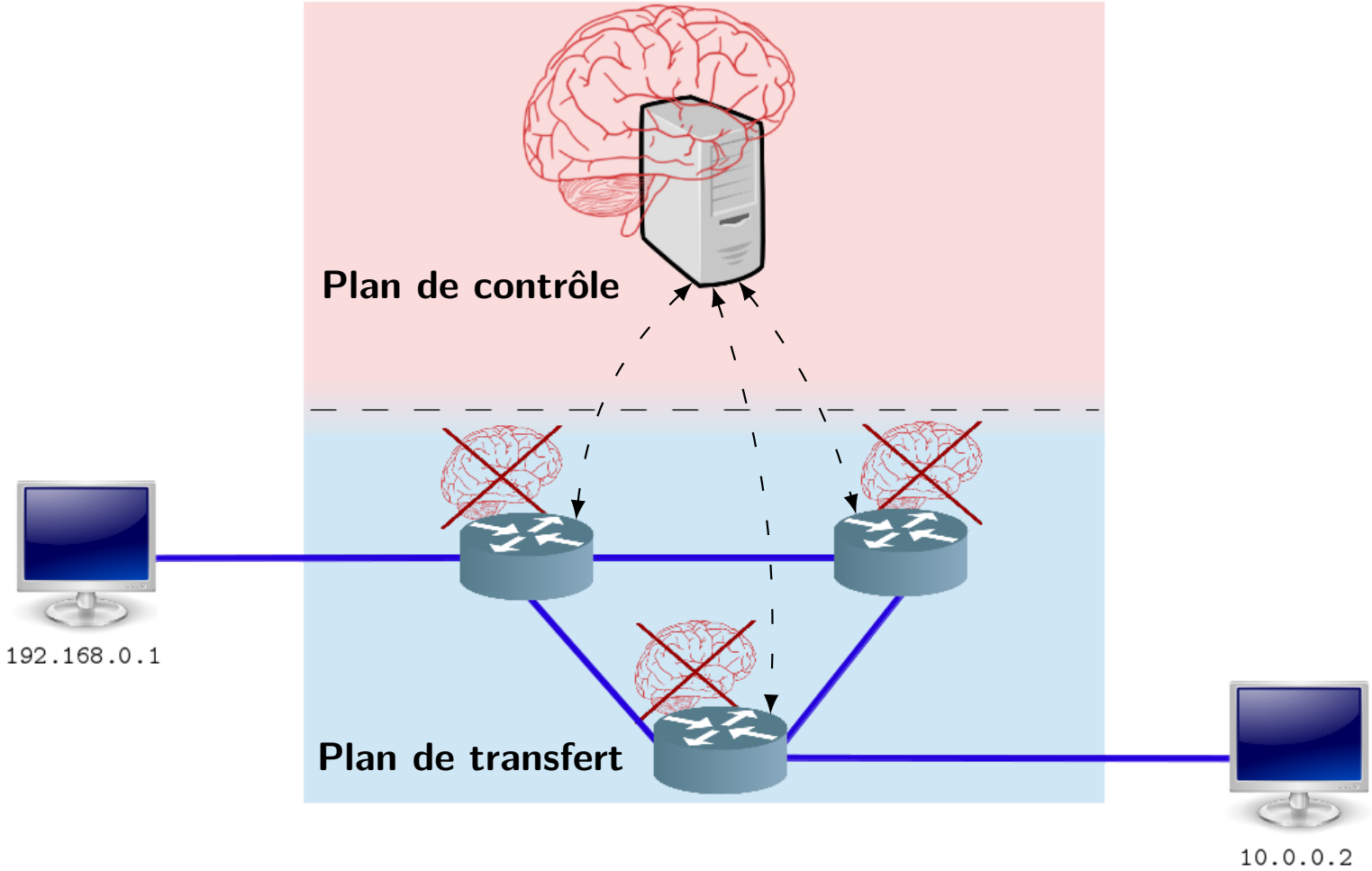
- Centralisation du plan de contrôle ? OpenFlow, Meru Center...
- Découplage entre les deux plans ? OpenFlow, Ryu...
- Programmation du réseau ? Cisco ACI, Juniper Contrail...
- White-box switching ? Switch Light OS, Cumulus Linux...

Software-Defined Networking ?

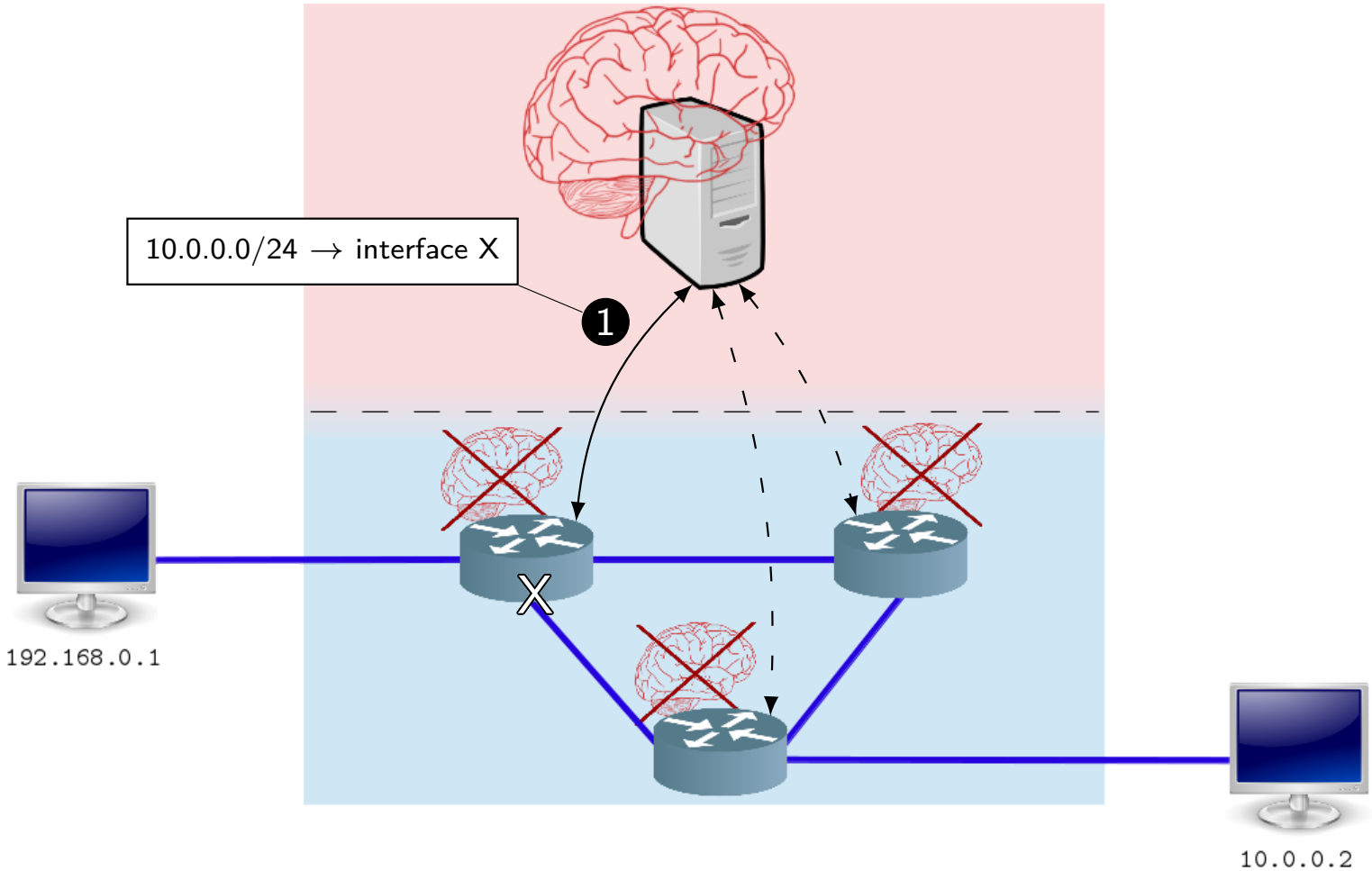
- Centralisation du plan de contrôle ? OpenFlow, Meru Center...
- Découplage entre les deux plans ? OpenFlow, Ryu...
- Programmation du réseau ? Cisco ACI, Juniper Contrail...
- White-box switching ? Switch Light OS, Cumulus Linux...

Pas de définition précise ni universelle !

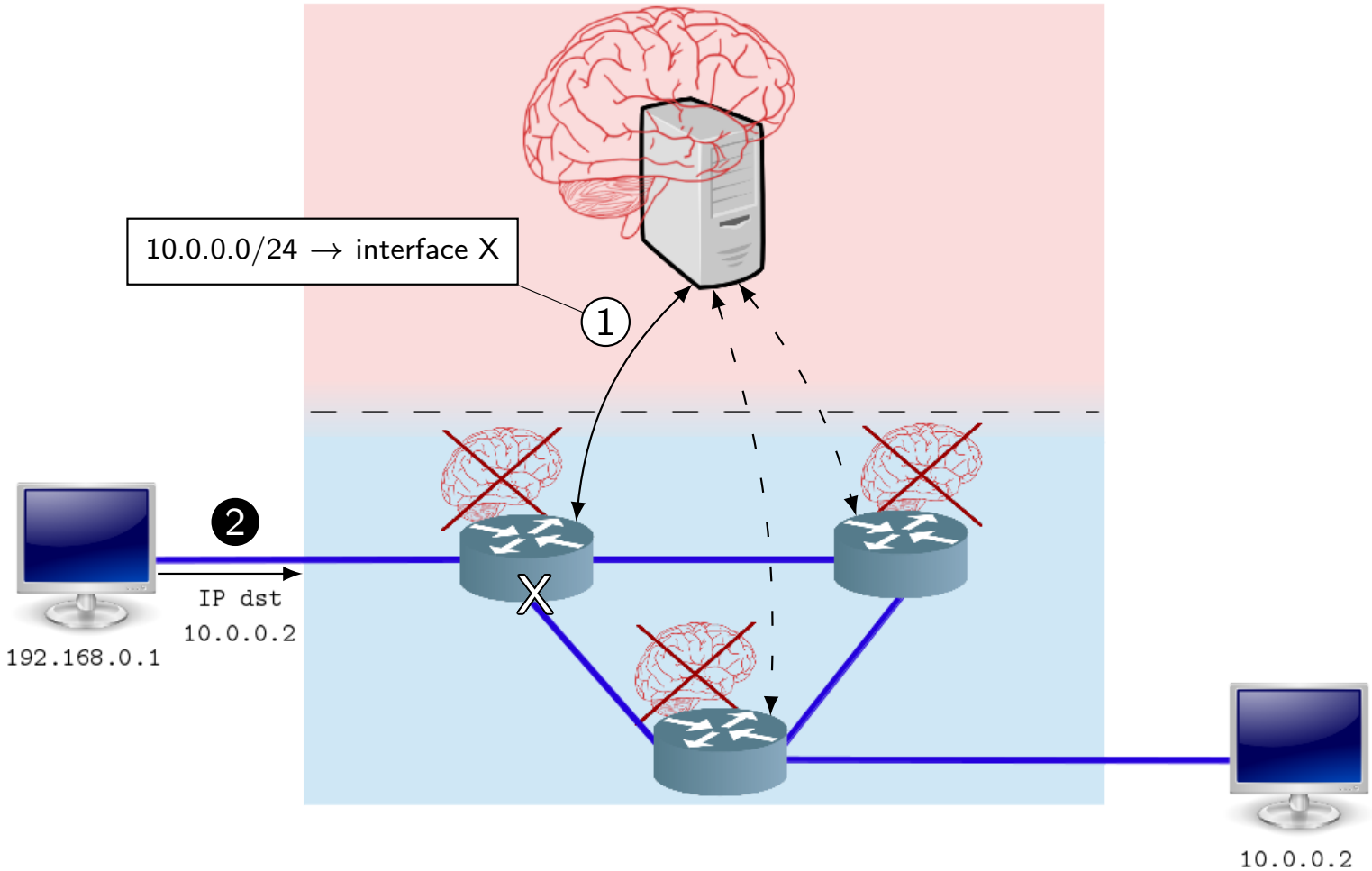
SDN à la OpenFlow



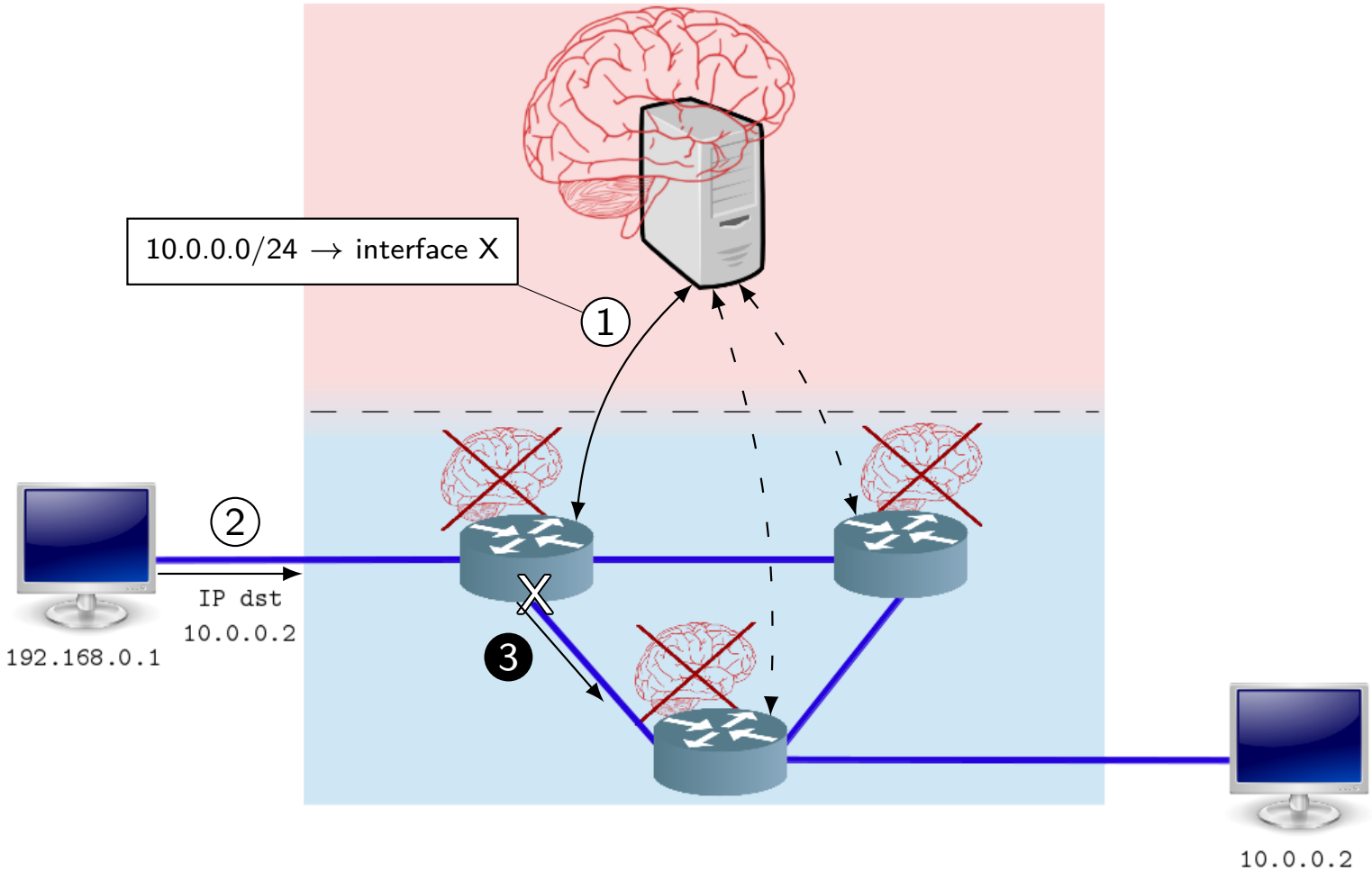
SDN à la OpenFlow



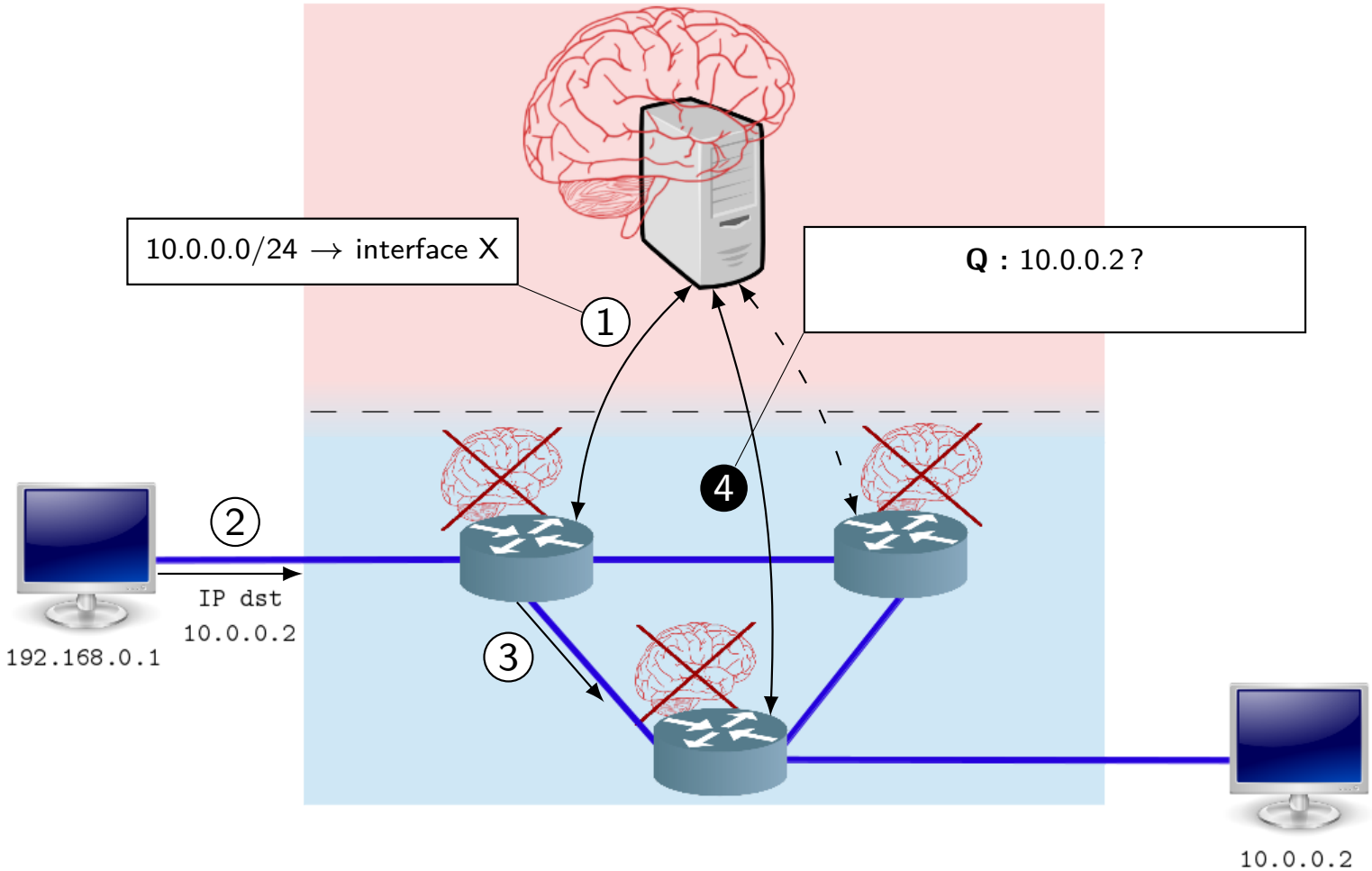
SDN à la OpenFlow



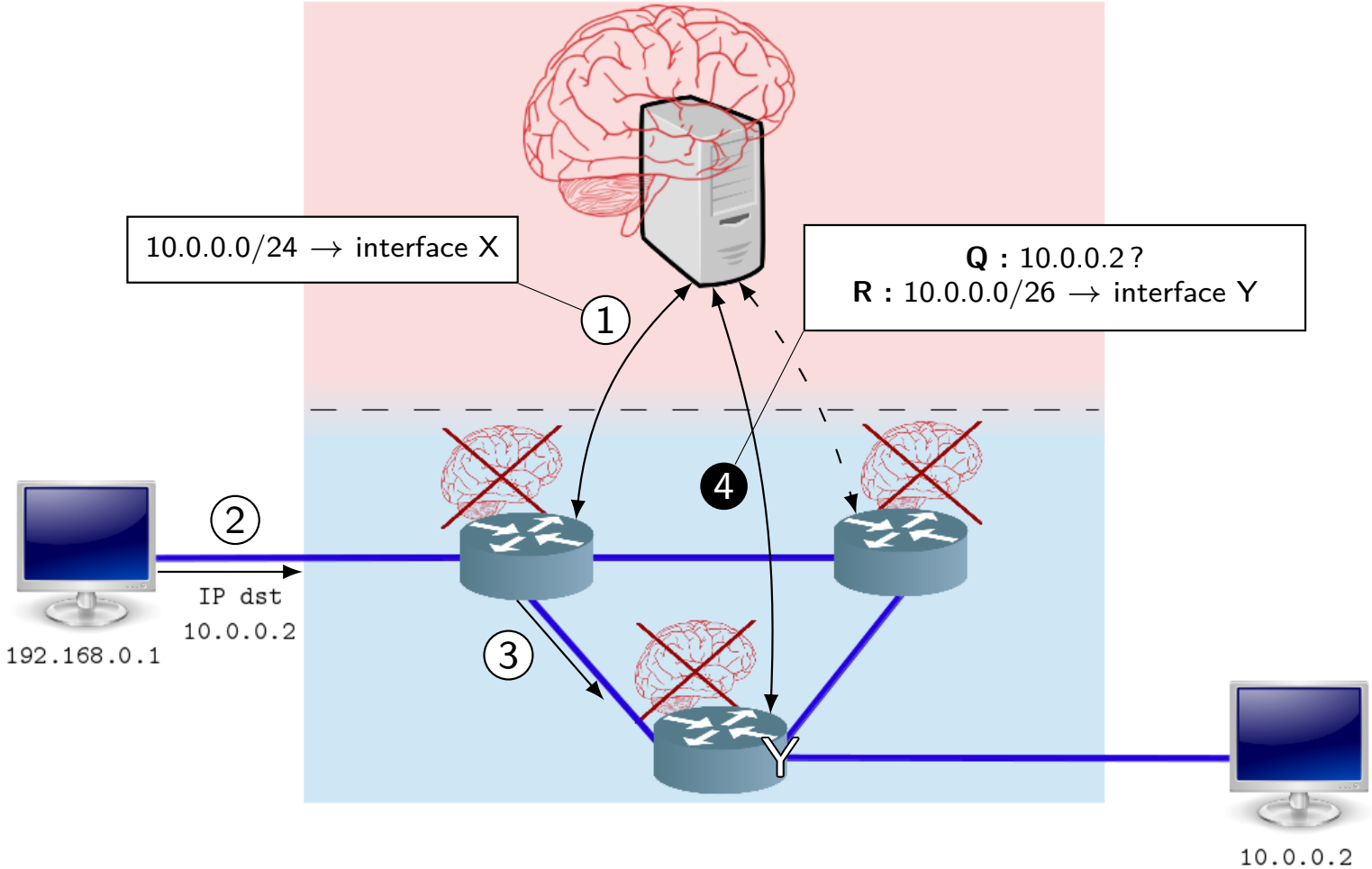
SDN à la OpenFlow



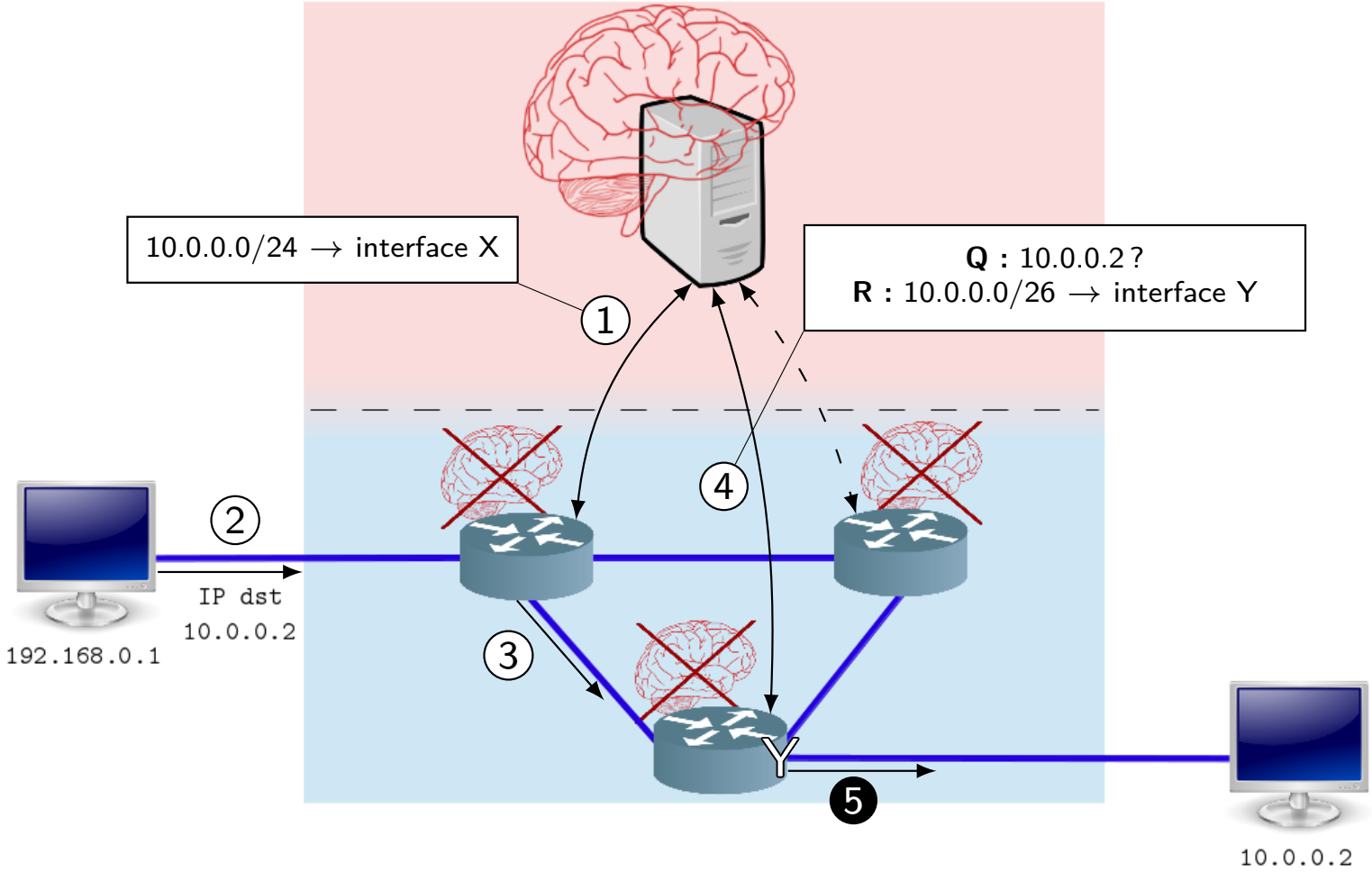
SDN à la OpenFlow



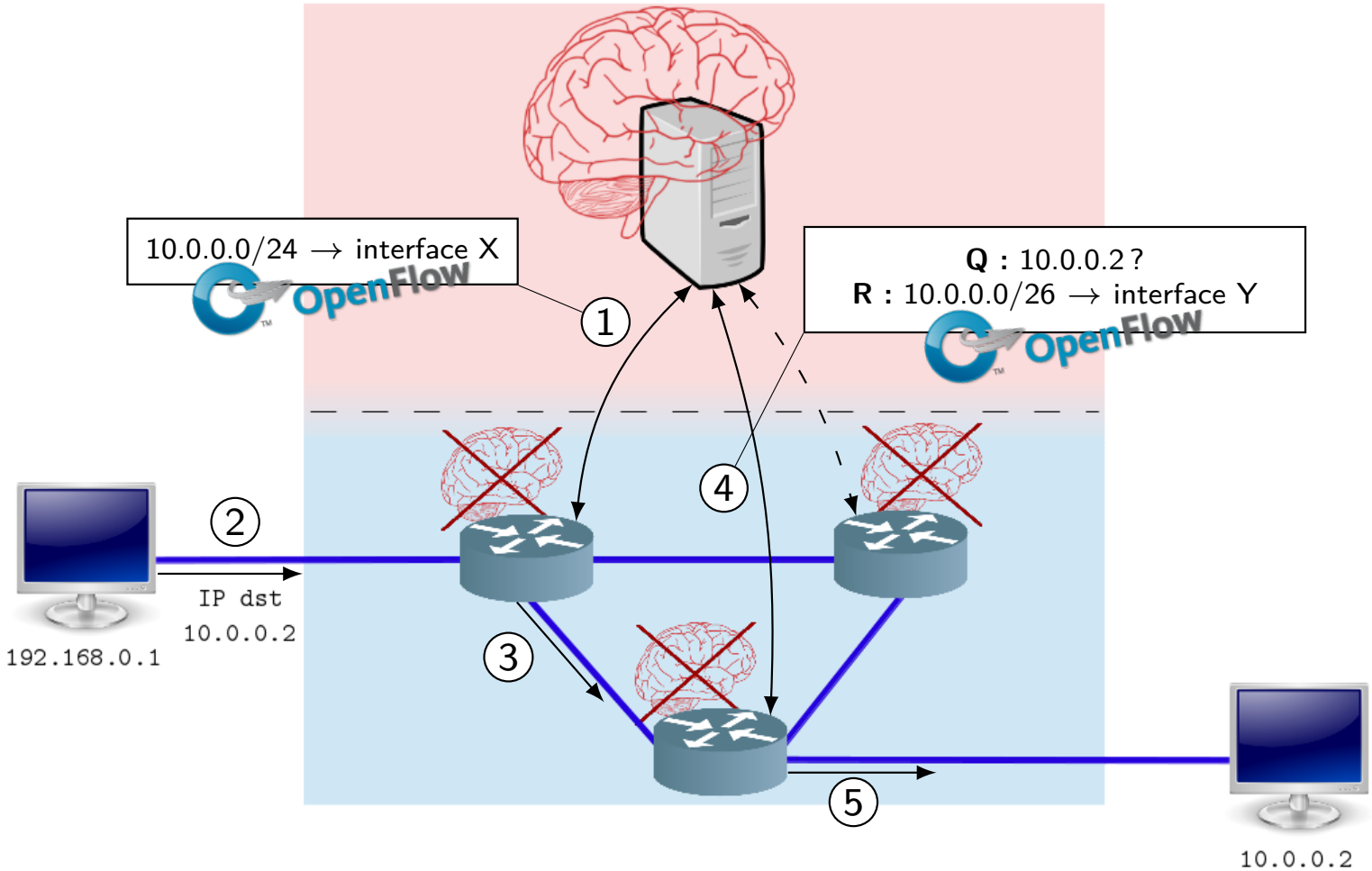
SDN à la OpenFlow



SDN à la OpenFlow



SDN à la OpenFlow



Constructeurs, développeurs, utilisateurs

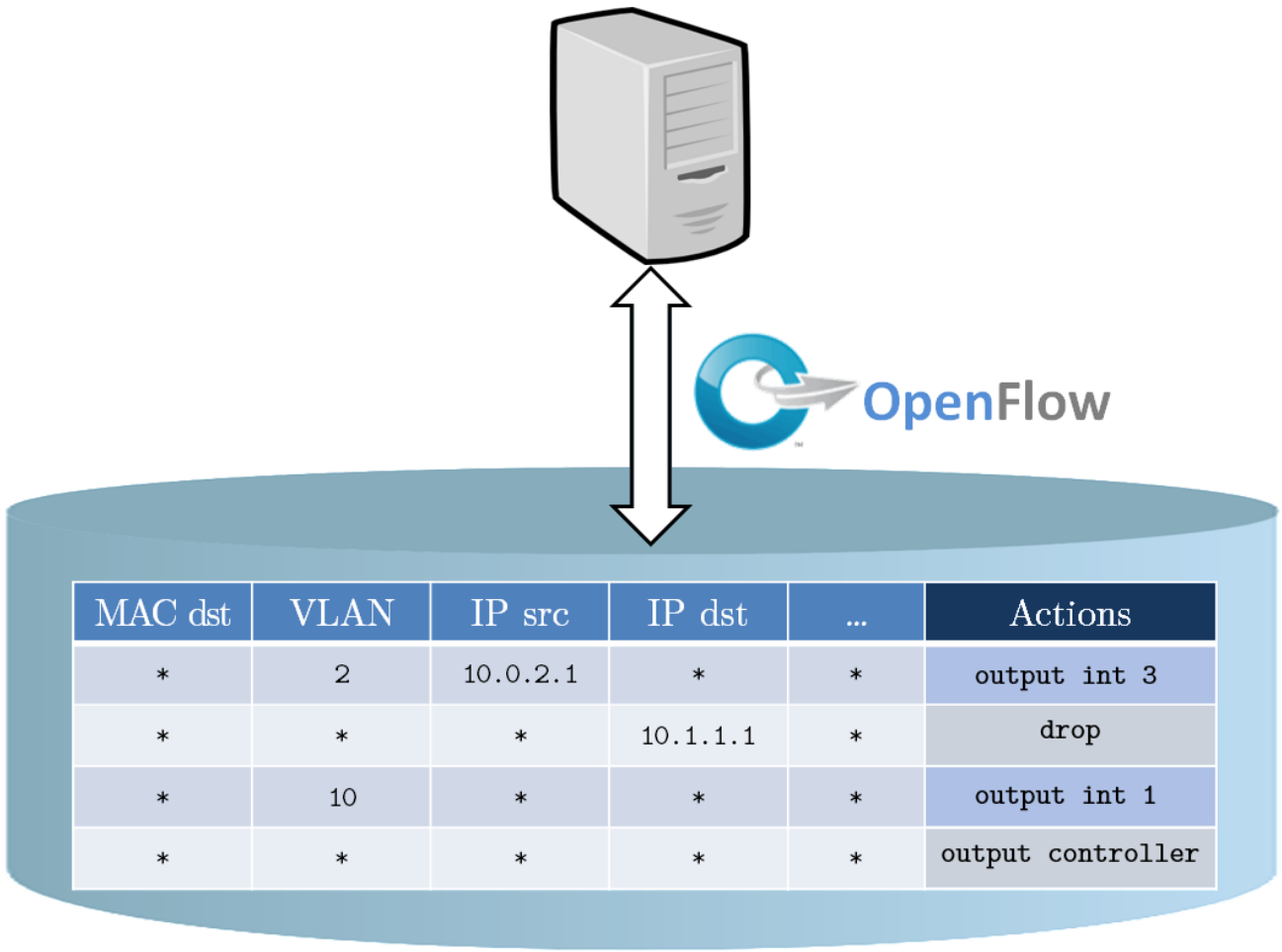
Contexte très concurrentiel et dynamique

Présentation du standard OpenFlow

OpenFlow - Aperçu

- Version 1.0 fin 2009, version 1.5 fin 2014
- Consortium de développement : Open Networking Foundation
- Port TCP 6653 alloué par l'IANA (6633 obsolète)

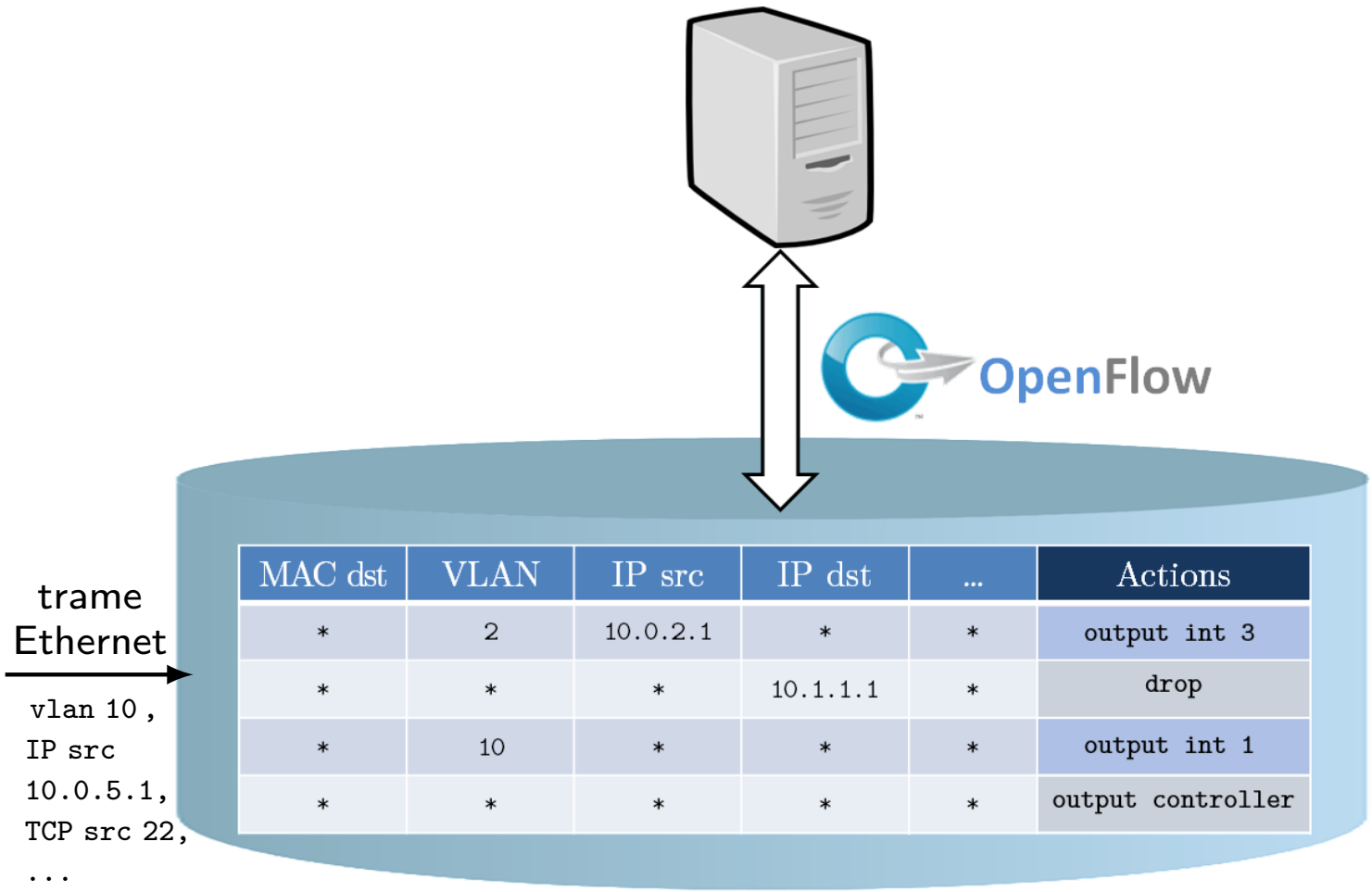
OpenFlow 1.0



MAC dst	VLAN	IP src	IP dst	...	Actions
*	2	10.0.2.1	*	*	output int 3
*	*	*	10.1.1.1	*	drop
*	10	*	*	*	output int 1
*	*	*	*	*	output controller

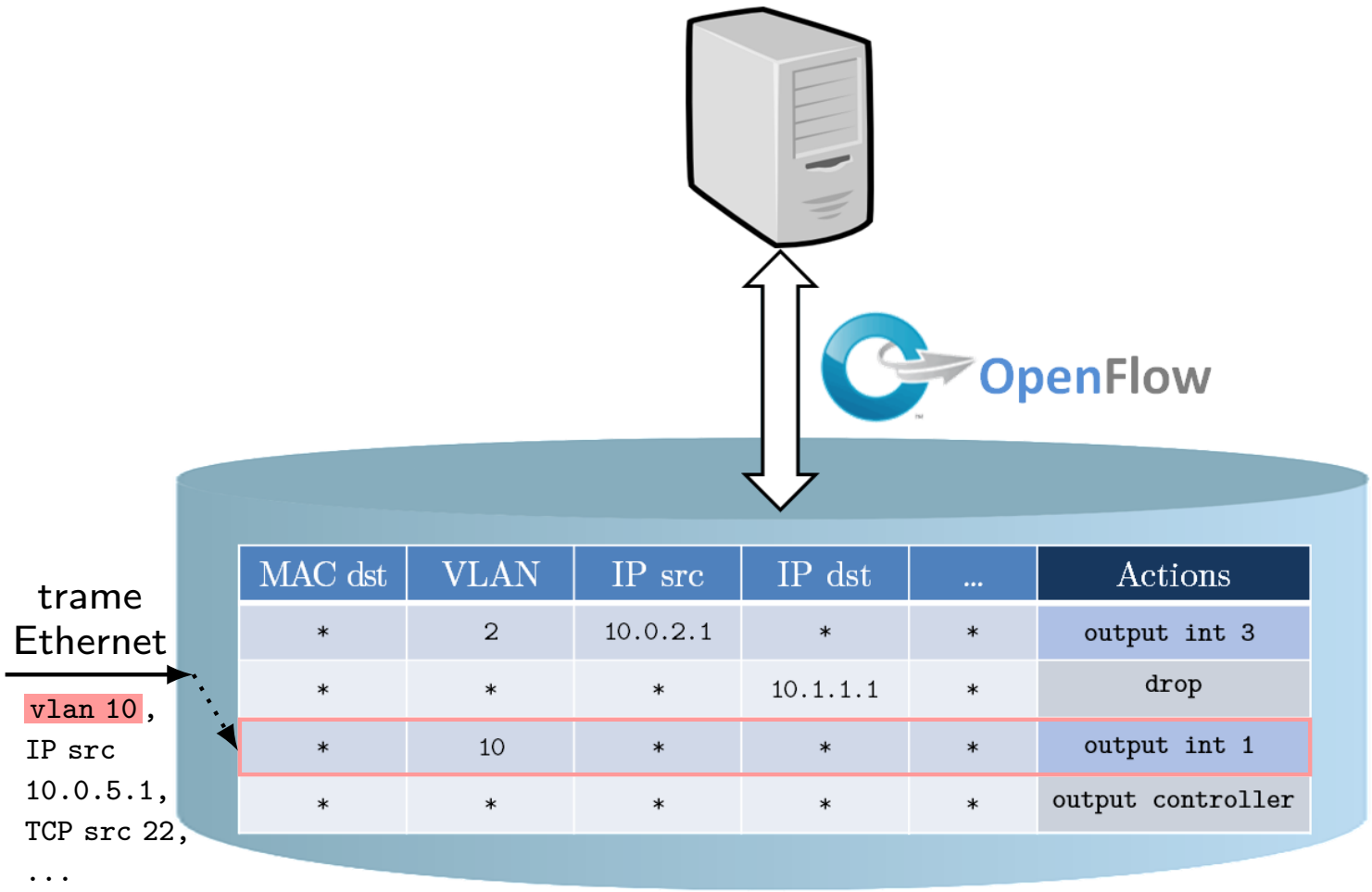
Table de flux écrite sur le switch

OpenFlow 1.0



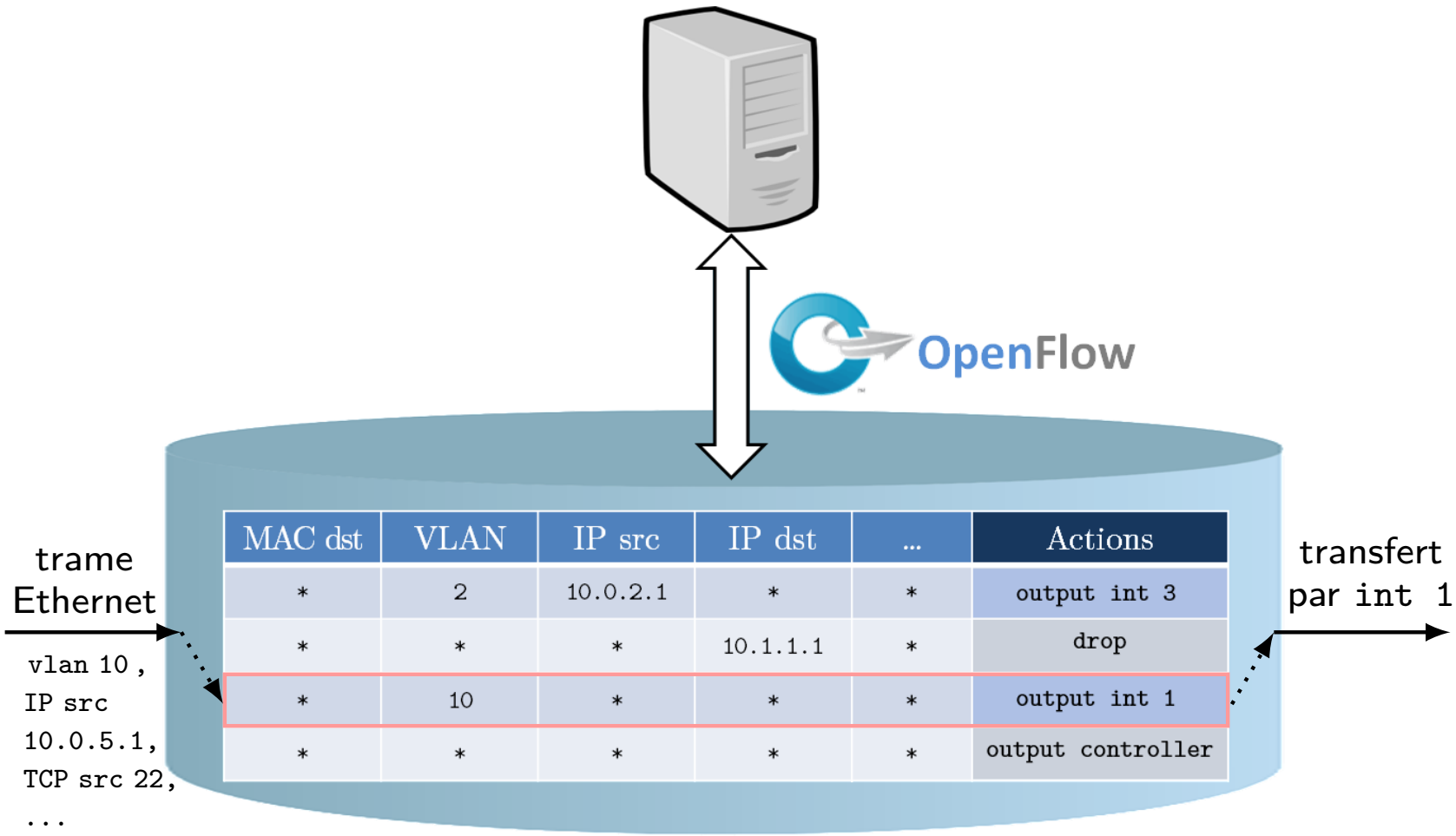
MAC dst	VLAN	IP src	IP dst	...	Actions
*	2	10.0.2.1	*	*	output int 3
*	*	*	10.1.1.1	*	drop
*	10	*	*	*	output int 1
*	*	*	*	*	output controller

OpenFlow 1.0

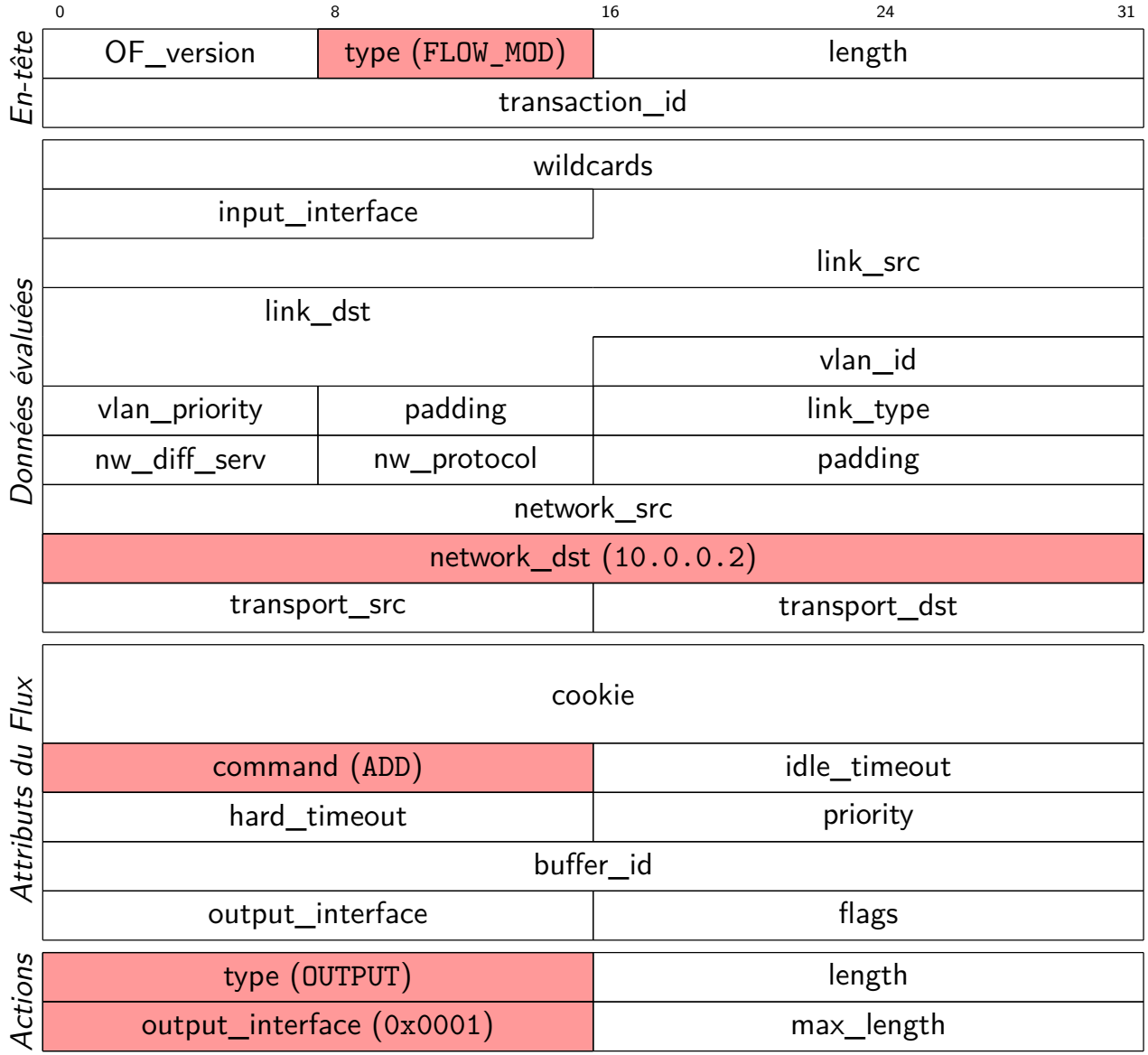


MAC dst	VLAN	IP src	IP dst	...	Actions
*	2	10.0.2.1	*	*	output int 3
*	*	*	10.1.1.1	*	drop
*	10	*	*	*	output int 1
*	*	*	*	*	output controller

OpenFlow 1.0


OpenFlow
trame Ethernet
vlan 10 , IP src 10.0.5.1, TCP src 22, ...
transfert par int 1
MAC dst | VLAN | IP src | IP dst | ... | Actions
* | 2 | 10.0.2.1 | * | * | output int 3
* | * | * | 10.1.1.1 | * | drop
* | 10 | * | * | * | output int 1
* | * | * | * | * | output controller

Écriture d'une règle de FIB avec OpenFlow 1.0



OpenFlow 1.0+ - Évolutions

- Plusieurs tables de flux

OpenFlow 1.0+ - Évolutions

- Plusieurs tables de flux
- Comportement en cas de perte du contrôleur

OpenFlow 1.0+ - Évolutions

- Plusieurs tables de flux
- Comportement en cas de perte du contrôleur
- Système maître/esclave pour plusieurs contrôleurs

OpenFlow 1.0+ - Évolutions

- Plusieurs tables de flux
- Comportement en cas de perte du contrôleur
- Système maître/esclave pour plusieurs contrôleurs
- Prise en charge de champs IPv6
- 41 champs filtrables (11 à l'origine)
- Gestion de groupes d'interfaces
- *Traps* destinés aux contrôleurs
- Connexions TCP auxiliaires
- Et autres redéfinitions pas toujours rétrocompatibles...

Plate-forme de test

Implémentations utilisées

HP J9264A (ProCurve 6600)

- Firmware K.15.13.0005, janvier 2014
- Firmware K.15.14.0007, juillet 2014 (OpenFlow 1.3, TLS)

Implémentations utilisées

HP J9264A (ProCurve 6600)

- Firmware K.15.13.0005, janvier 2014
- Firmware K.15.14.0007, juillet 2014 (OpenFlow 1.3, TLS)

Contrôleur Floodlight v0.90

- Développement libre, soutenu par Big Switch Networks
- Administration par l'intermédiaire de requêtes HTTP(S)
- API restreinte et peu souple...

Modules Scapy pour le contrôleur de test

```
###[ OFPT_FLOW_MOD ]###
  version   = OpenFlow 1.0
  type      = OFPT_FLOW_MOD
  len       = 80
  xid       = 0
  \match     \
   |###[ OFP_MATCH ]###
   |  wildcards1= DL_VLAN_PCP+NW_TOS
   |  nw_dst_mask= 0L
   |  nw_src_mask= 63L
   |  wildcards2= IN_PORT+DL_VLAN+DL_SRC+DL_DST+NW_PROTO+TP_SRC+TP_DST
   |  in_port   = 0
   |  dl_src    = 00:00:00:00:00:00
   |  dl_dst    = 00:00:00:00:00:00
   |  dl_vlan   = 0
   |  dl_vlan_pcp= 0
   |  pad1      = 0x0
   |  dl_type   = 2048
   |  nw_tos    = 0
   |  nw_proto  = 0
   |  pad2      = 0x0
   |  nw_src    = 0.0.0.0
   |  nw_dst    = 10.0.0.2
   |  tp_src    = 0
   |  tp_dst    = 0
  cookie    = 0
  cmd       = OFPFC_ADD
  idle_timeout= 0
  hard_timeout= 0
  priority  = 0
  buffer_id = NO_BUFFER
  out_port  = NONE
  flags     =
  \actions   \
   |###[ OFPAT_OUTPUT ]###
   |  type      = OFPAT_OUTPUT
   |  len       = 8
   |  port      = 1
   |  max_len   = NO_BUFFER
```

Modules Scapy pour le contrôleur de test

```
###[ OFPT_FLOW_MOD ]###
  version   = OpenFlow 1.0
  type      = OFPT_FLOW_MOD
  len       = 80
  xid       = 0
  \match     \
   |###[ OFP_MATCH ]###
   |  wildcards1= DL_VLAN_PCP+NW_TOS
   |  nw_dst_mask= 0L
   |  nw_src_mask= 63L
   |  wildcards2= IN_PORT+DL_VLAN+DL_SRC+DL_DST+NW_PROTO+TP_SRC+TP_DST
   |  in_port   = 0
   |  dl_src    = 00:00:00:00:00:00
   |  dl_dst    = 00:00:00:00:00:00
   |  dl_vlan   = 0
   |  dl_vlan_pcp= 0
   |  pad1      = 0x0
   |  dl_type   = 2048
   |  nw_tos    = 0
   |  nw_proto  = 0
   |  pad2      = 0x0
   |  nw_src    = 0.0.0.0
   |  nw_dst    = 10.0.0.2
   |  tp_src    = 0
   |  tp_dst    = 0
  cookie    = 0
  cmd       = OFPFC_ADD
  idle_timeout= 0
  hard_timeout= 0
  priority  = 0
  buffer_id = NO_BUFFER
  out_port  = NONE
  flags     =
  \actions   \
   |###[ OFPAT_OUTPUT ]###
   |  type      = OFPAT_OUTPUT
   |  len       = 8
   |  port      = 1
   |  max_len   = NO_BUFFER
```

bb.secdev.org/scapy/src

Étude de sécurité

TLS, un chantier en cours

- TLS obligatoire entre switch et contrôleur avec OpenFlow 1.0
- ...mais absent des premières implémentations !
- Protection rendue optionnelle à partir d'OpenFlow 1.1
- Tout de même, début des implémentations de TLS en 2014
- Et le canal administrateur–contrôleur ?
- Et le cloisonnement des réseaux ?

DoS - Sans accès au canal OpenFlow

Buffer du switch en attente du contrôleur

- Les trames inconnues attendent une réponse du contrôleur
- Un attaquant génère alors de telles trames et sature le buffer

Calcul de nouveaux flux par le contrôleur

- Des trames inconnues sollicitent à nouveau le contrôleur
- Le contrôleur est mobilisé pour calculer les actions à appliquer

DoS - Avec accès au canal OpenFlow

Maintien du contrôleur dans un rôle d'esclave

- Un faux contrôleur requiert le rôle unique de maître
- Les autres contrôleurs deviennent esclaves *read-only*

Taille limitée de la table de flux du switch

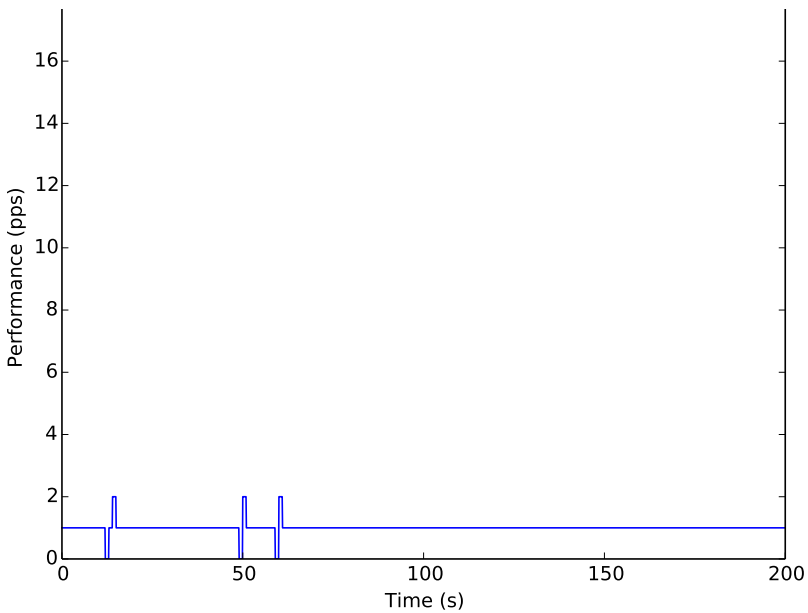
- OpenFlow 1.4+ signale ou filtre les flux en excès
- L'implémentation 1.0 tient mal la charge

DoS - Surcharge d'une table HP OpenFlow 1.0

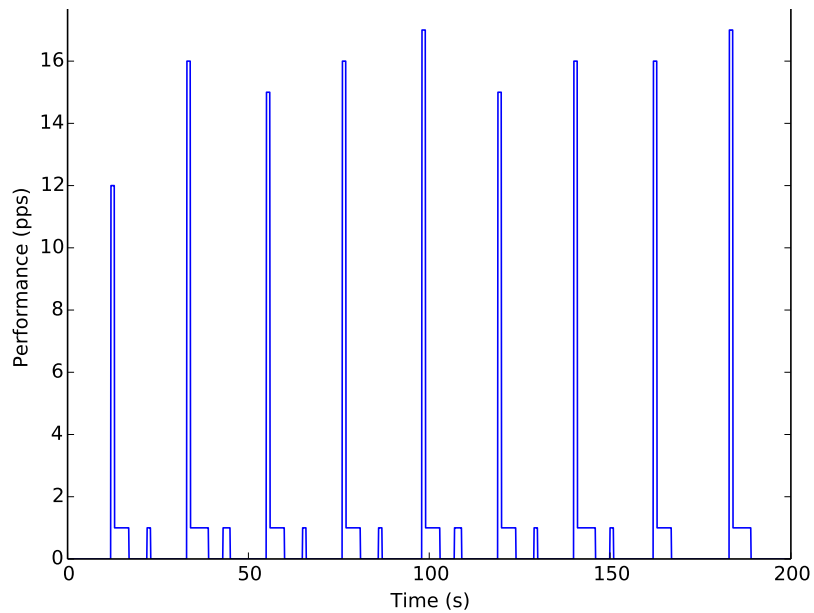
- Code erreur `ALL_TABLES_FULL` au-delà de 2^{16} flux $\rightarrow$ OK
- Si les `flow_mod` arrivent trop vite $\rightarrow$ Redémarrage du switch !
- Si la table est saturée $\rightarrow$ Perte partielle de disponibilité !

DoS - Surcharge d'une table HP OpenFlow 1.0

- Code erreur `ALL_TABLES_FULL` au-delà de 2^{16} flux $\rightarrow$ OK
- Si les `flow_mod` arrivent trop vite $\rightarrow$ Redémarrage du switch !
- Si la table est saturée $\rightarrow$ Perte partielle de disponibilité !



Avec 2^{14} flux



Avec 2^{16} flux

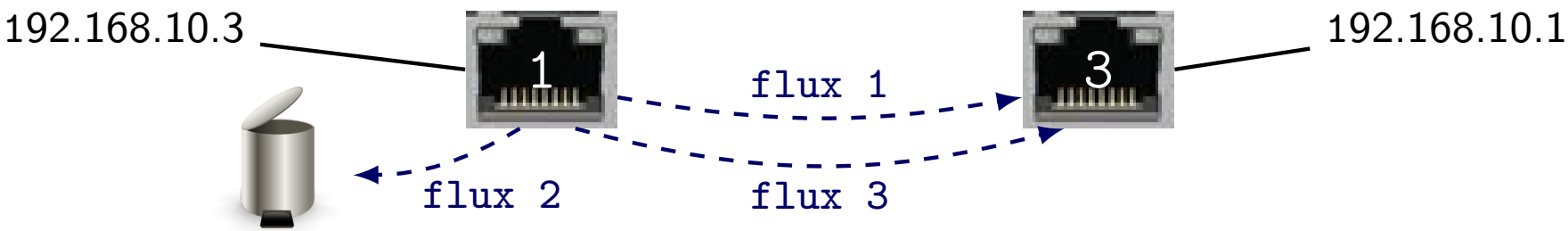
Autres soucis d'implémentation

Concurrence de flux sur le switch HP

Une trame peut correspondre à plusieurs flux...

	Interface	EtherType	IP Src	Autres	Priorité	Action
flux 1	1	*	*	*	DEFAULT	output int 3
flux 2	1	0x0800	*	*	DEFAULT	drop
flux 3	1	0x0800	192.168.10.3	*	DEFAULT	output int 3

→ Si la priorité est la même, c'est au switch de décider quel flux appliquer

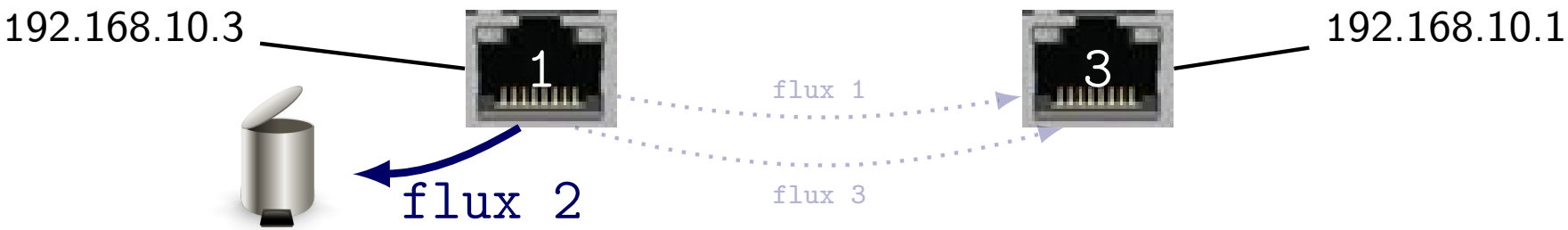


Concurrence de flux sur le switch HP

Une trame peut correspondre à plusieurs flux...

	Interface	EtherType	IP Src	Autres	Priorité	Action
flux 1	1	*	*	*	DEFAULT	output int 3
flux 2	1	0x0800	*	*	DEFAULT	drop
flux 3	1	0x0800	192.168.10.3	*	DEFAULT	output int 3

→ Si la priorité est la même, c'est au switch de décider quel flux appliquer

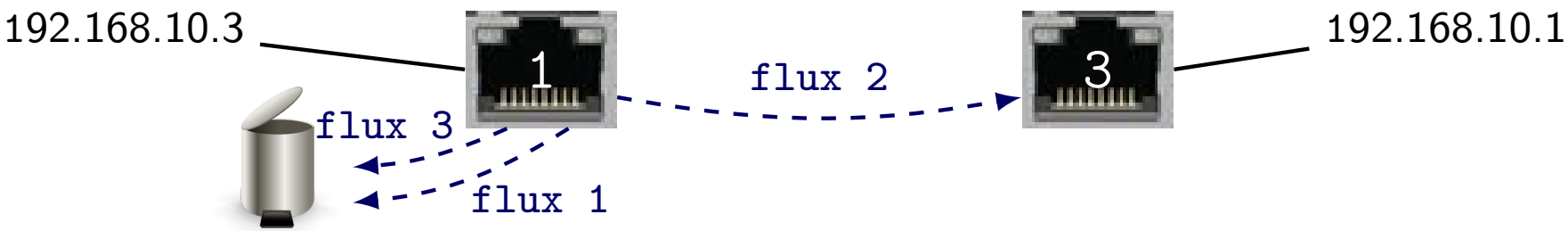


Concurrence de flux sur le switch HP

Une trame peut correspondre à plusieurs flux...

	Interface	EtherType	IP Src	Autres	Priorité	Action
flux 1	1	*	*	*	DEFAULT	drop
flux 2	1	0x0800	*	*	DEFAULT	output int 3
flux 3	1	0x0800	192.168.10.3	*	DEFAULT	drop

→ Si la priorité est la même, c'est au switch de décider quel flux appliquer



Concurrence de flux sur le switch HP

Une trame peut correspondre à plusieurs flux...

	Interface	EtherType	IP Src	Autres	Priorité	Action
flux 1	1	*	*	*	DEFAULT	drop
flux 2	1	0x0800	*	*	DEFAULT	output int 3
flux 3	1	0x0800	192.168.10.3	*	DEFAULT	drop

→ Si la priorité est la même, c'est au switch de décider quel flux appliquer

Concurrence de flux sur le switch HP

	Interface	EtherType	IP Src	Autres	Priorité
flux 1	1	*	*	*	DEFAULT
flux 2	1	0x0800	*	*	DEFAULT
flux 3	1	0x0800	192.168.10.3	*	DEFAULT

	K.15.13.0005	
flux 1	output	drop
flux 2	drop	output
flux 3	output	drop

Concurrence de flux sur le switch HP

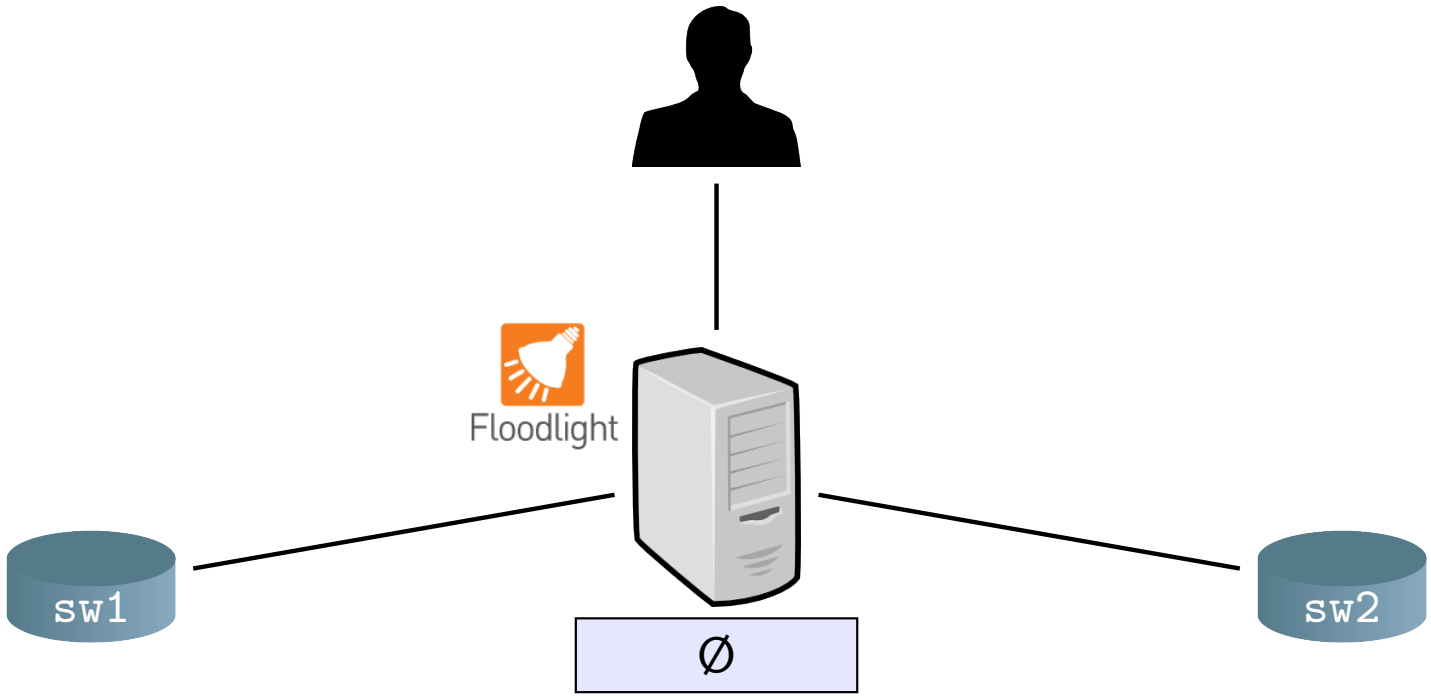
	Interface	EtherType	IP Src	Autres	Priorité
flux 1	1	*	*	*	DEFAULT
flux 2	1	0x0800	*	*	DEFAULT
flux 3	1	0x0800	192.168.10.3	*	DEFAULT

	K.15.13.0005		K.15.14.0007			
flux 1	output	drop	output	drop	output	drop
flux 2	drop	output	drop	output	output	drop
flux 3	output	drop	output	drop	output	drop

→ Possibilité d'utiliser le drapeau `CHECK_OVERLAP` pour chaque `FLOW_MOD`

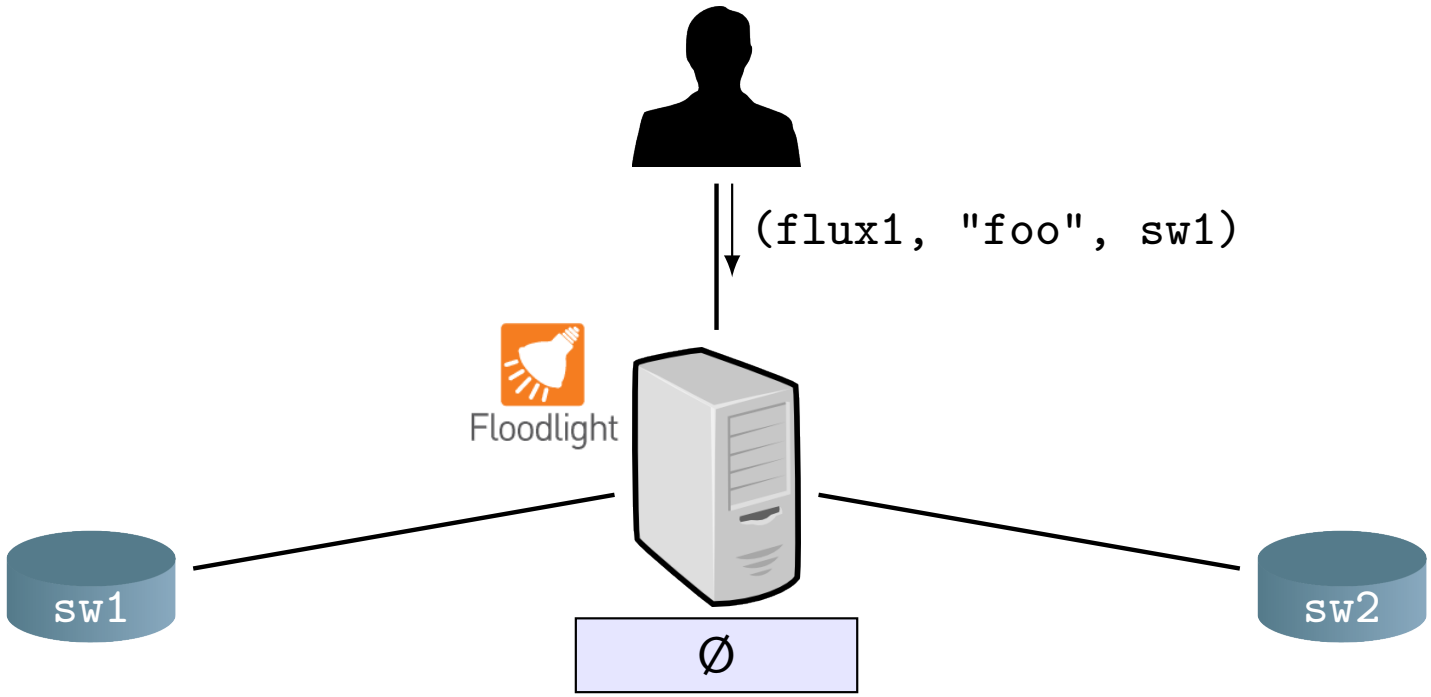
→ La logique d'élection reste à définir par le constructeur

Conflit de flux dans Floodlight v0.90



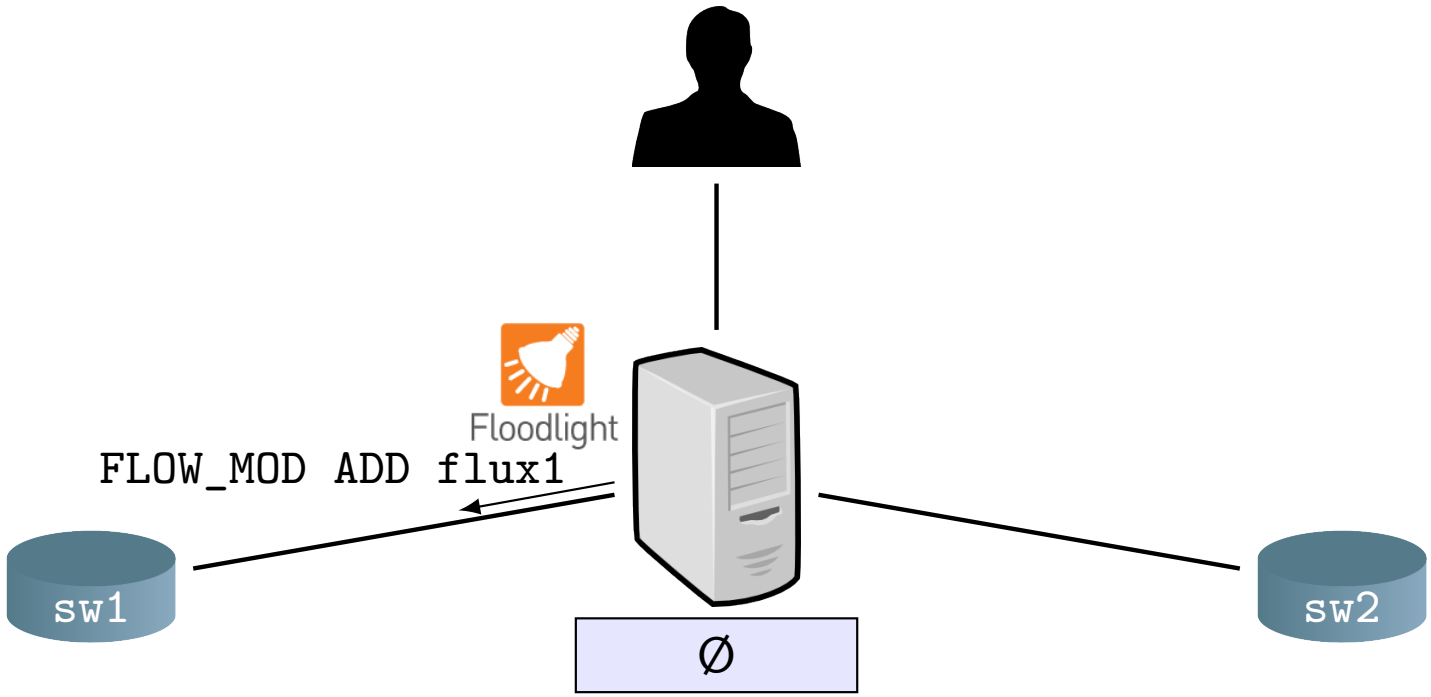
- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



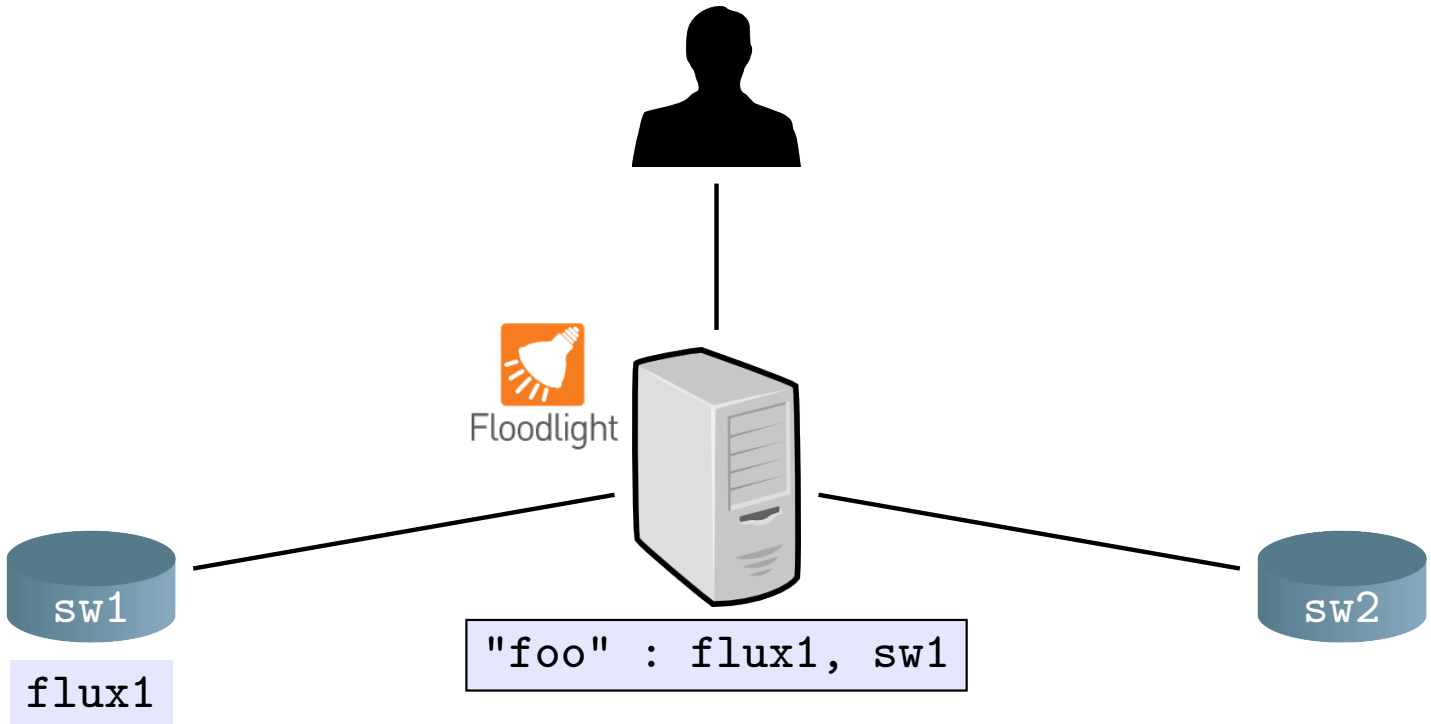
- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



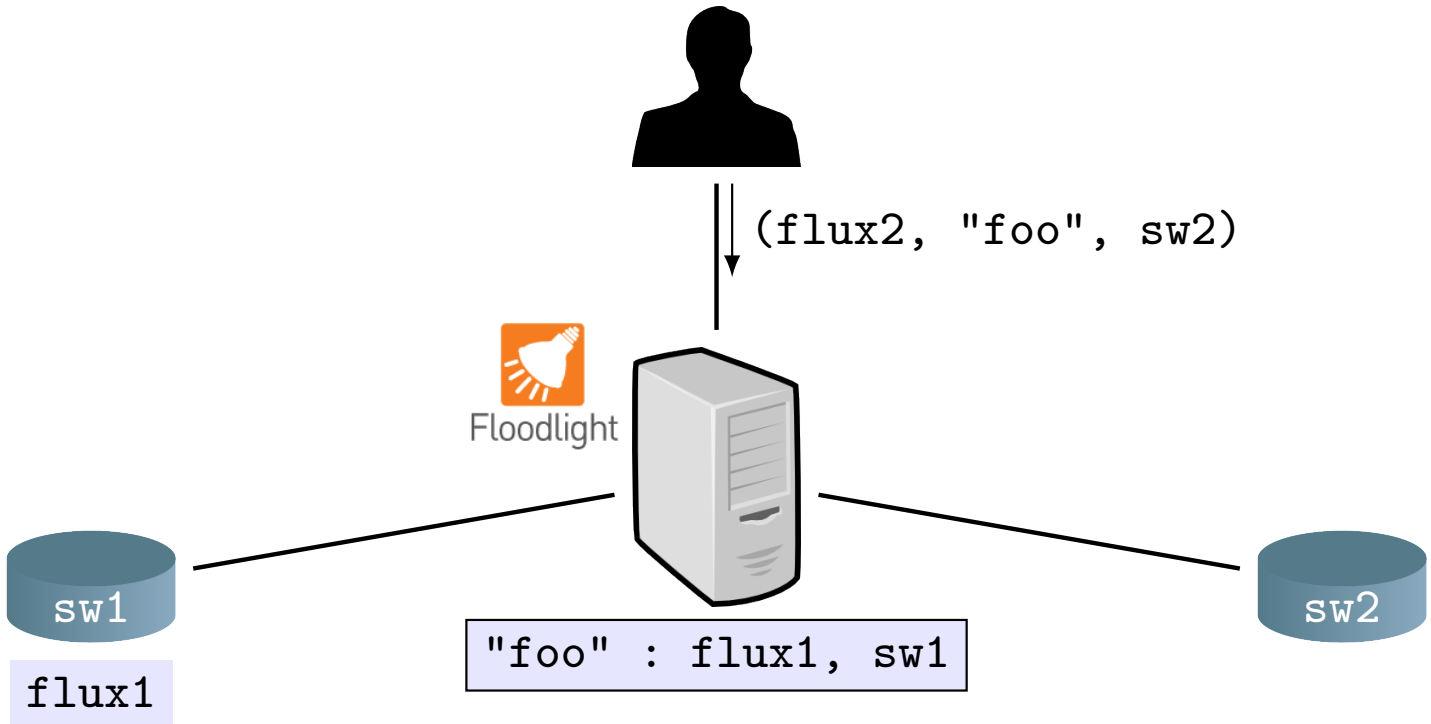
- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



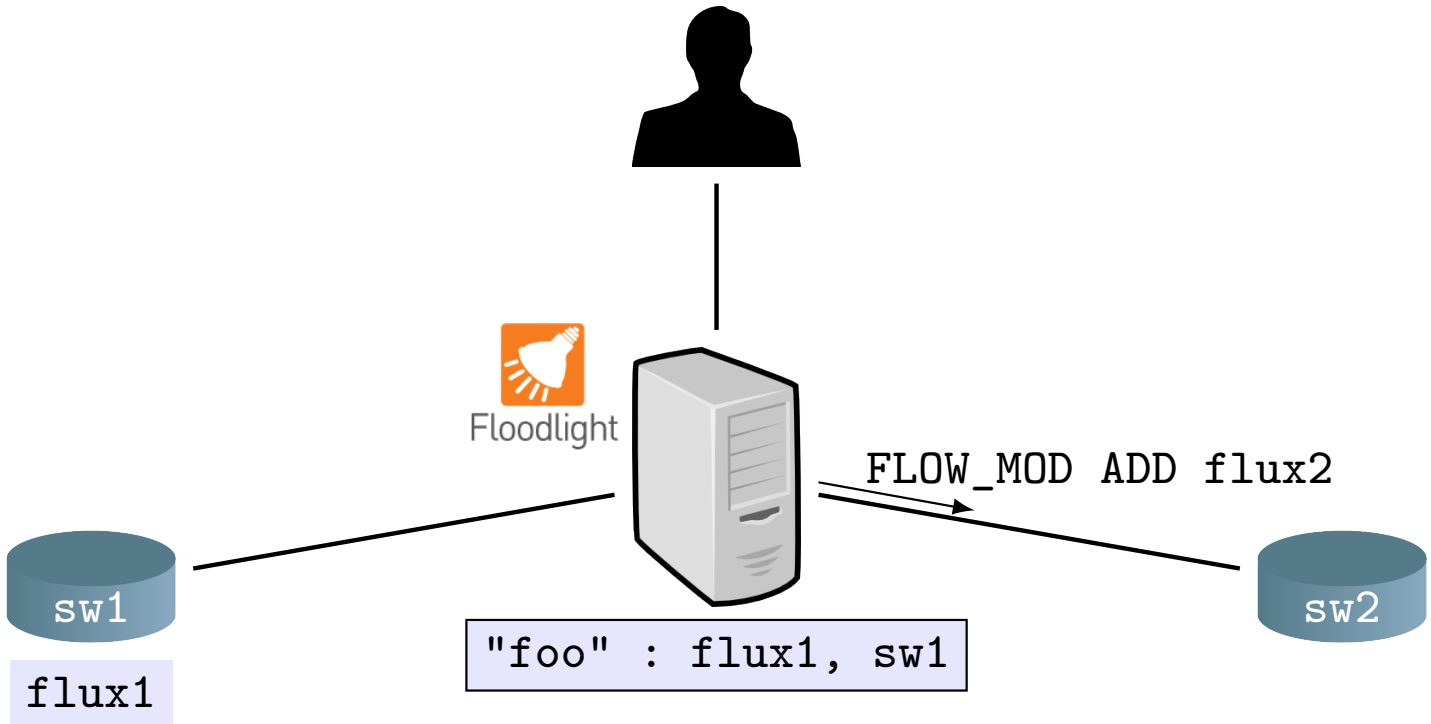
- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



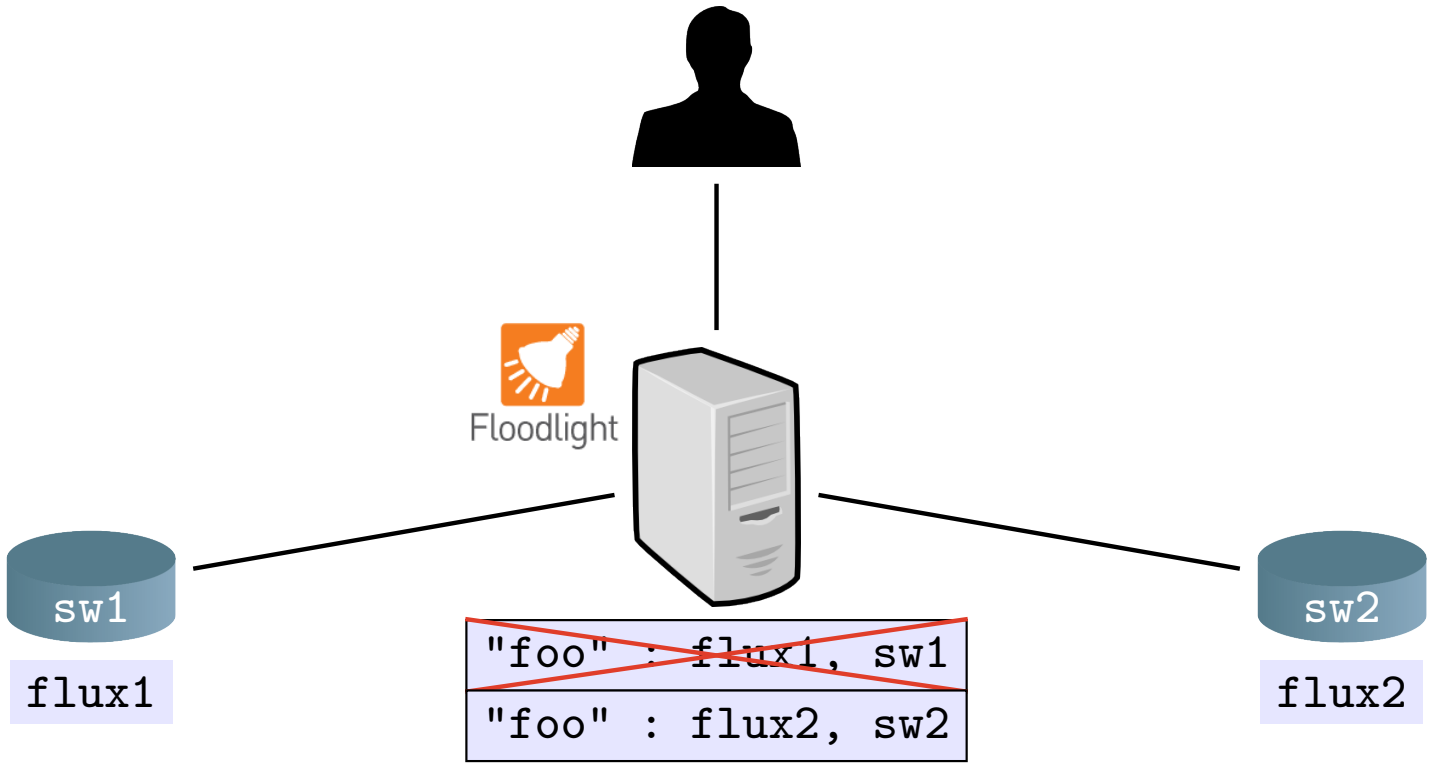
- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



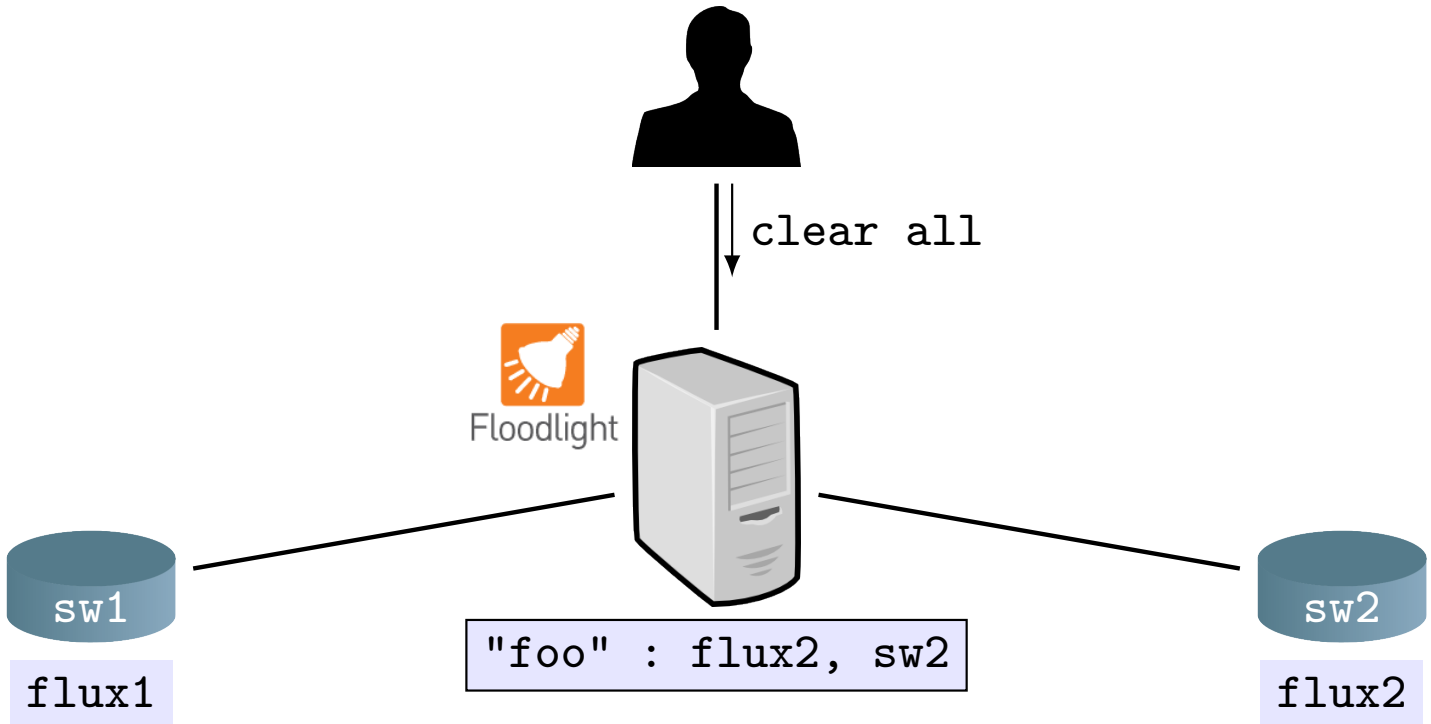
- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



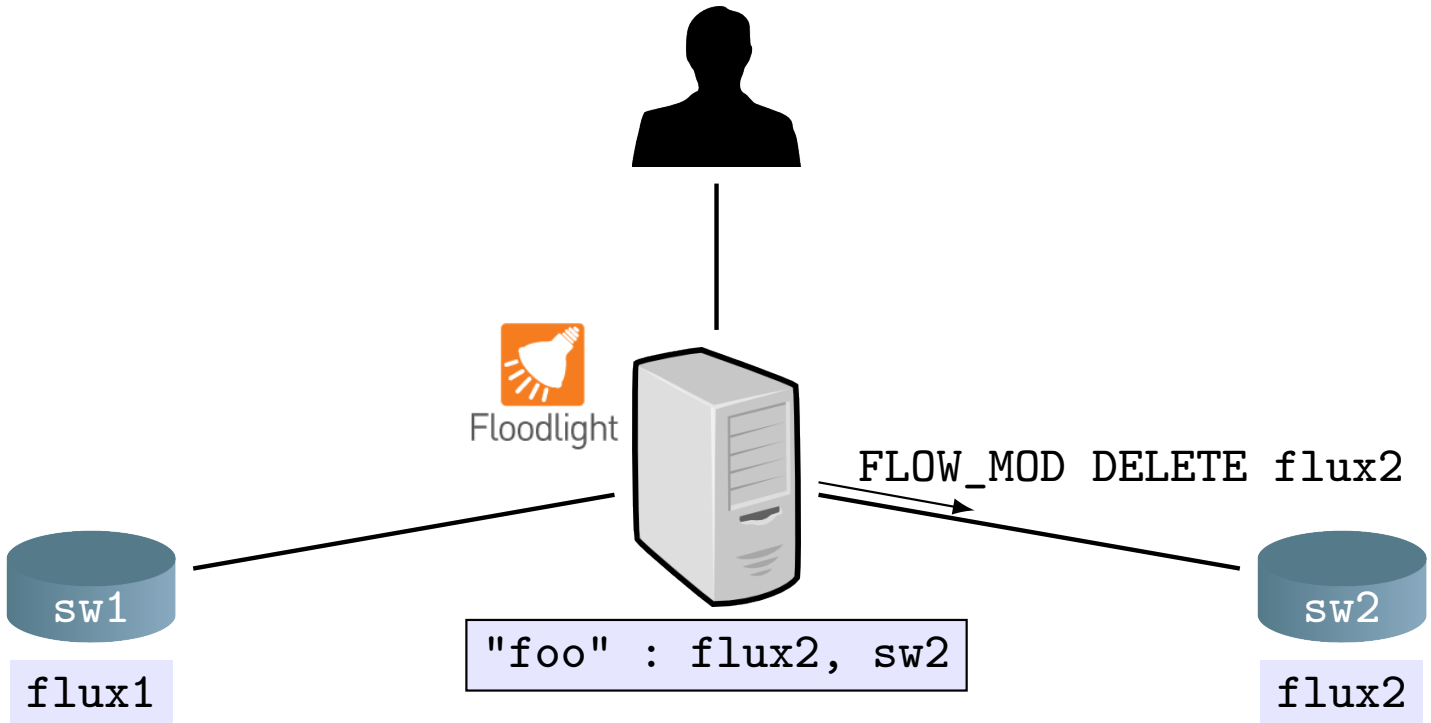
- **Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base**
- **En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch**

Conflit de flux dans Floodlight v0.90



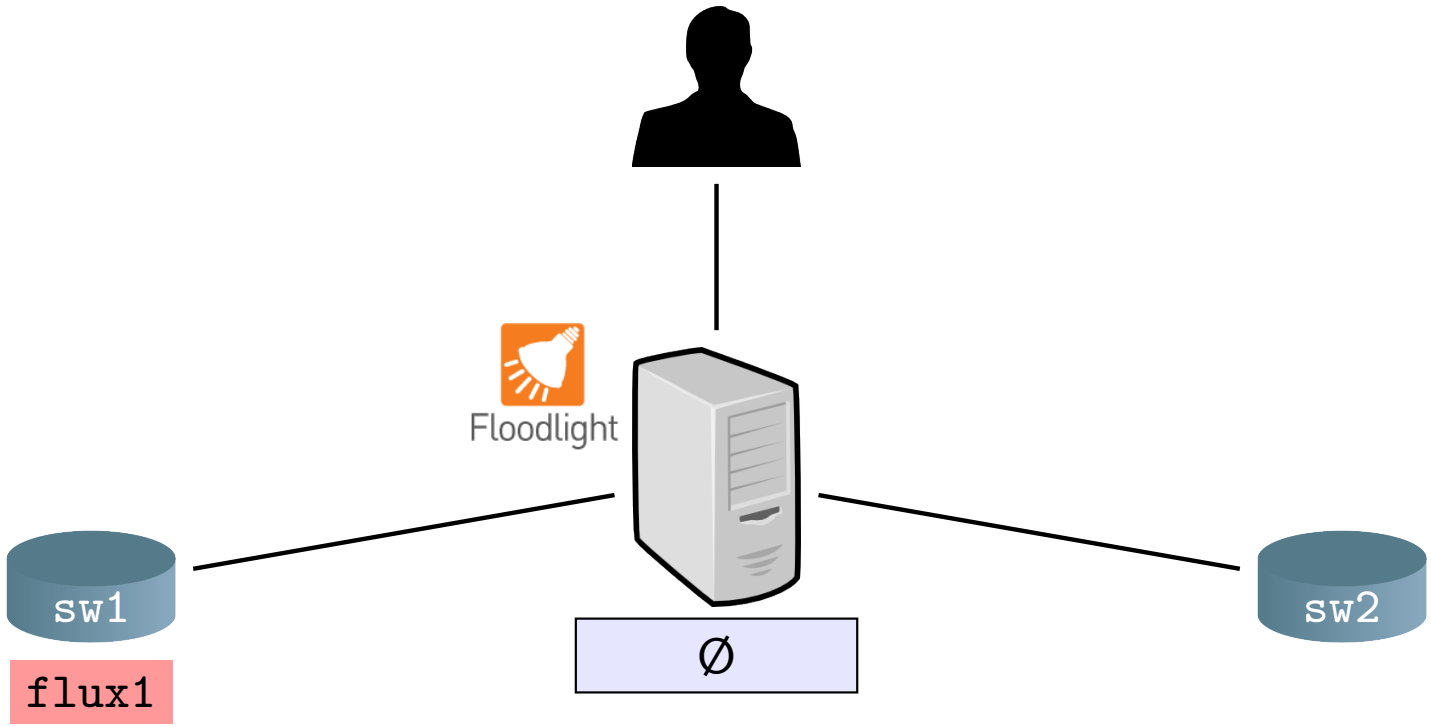
- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



- Le contrôleur Floodlight se sert des noms des flux comme clés primaires de sa base
- En cas de conflit il écrase le flux le plus ancien, mais oublie de vérifier si le nouveau est relatif au même switch

Conflit de flux dans Floodlight v0.90



→ Création de flux oubliés par le contrôleur

→ Impossibilité de supprimer ces flux actifs avec l'API standard

→ Bug remonté, patché dans Floodlight v1.0

Conclusion

Conclusion

- Protocole OpenFlow **et** implémentations à surveiller
- Déploiements à envisager au cas par cas
- Respect des bonnes pratiques toujours d'actualité

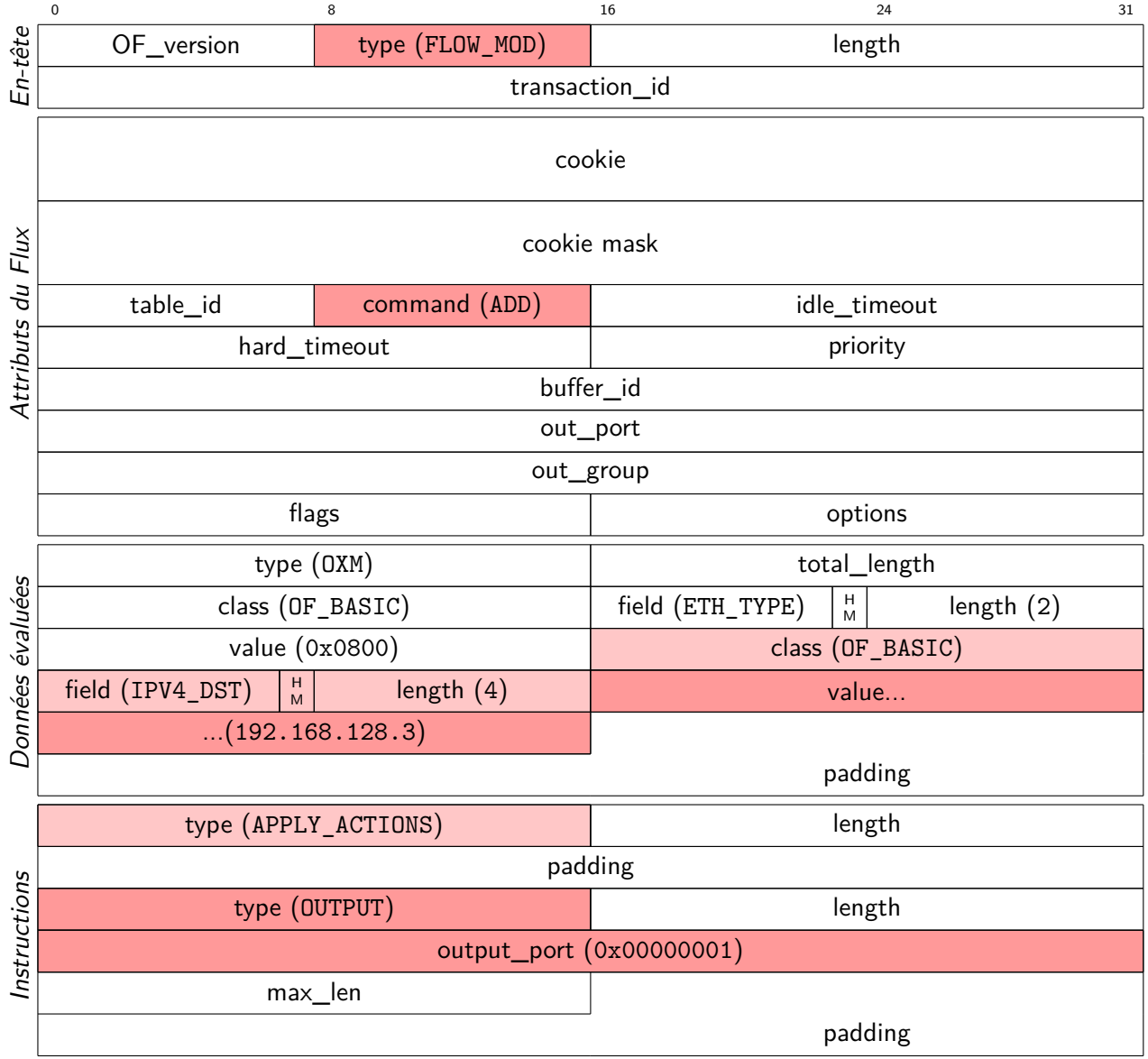
Questions ?

http://www.ssi.gouv.fr
maxence.tury@ssi.gouv.fr

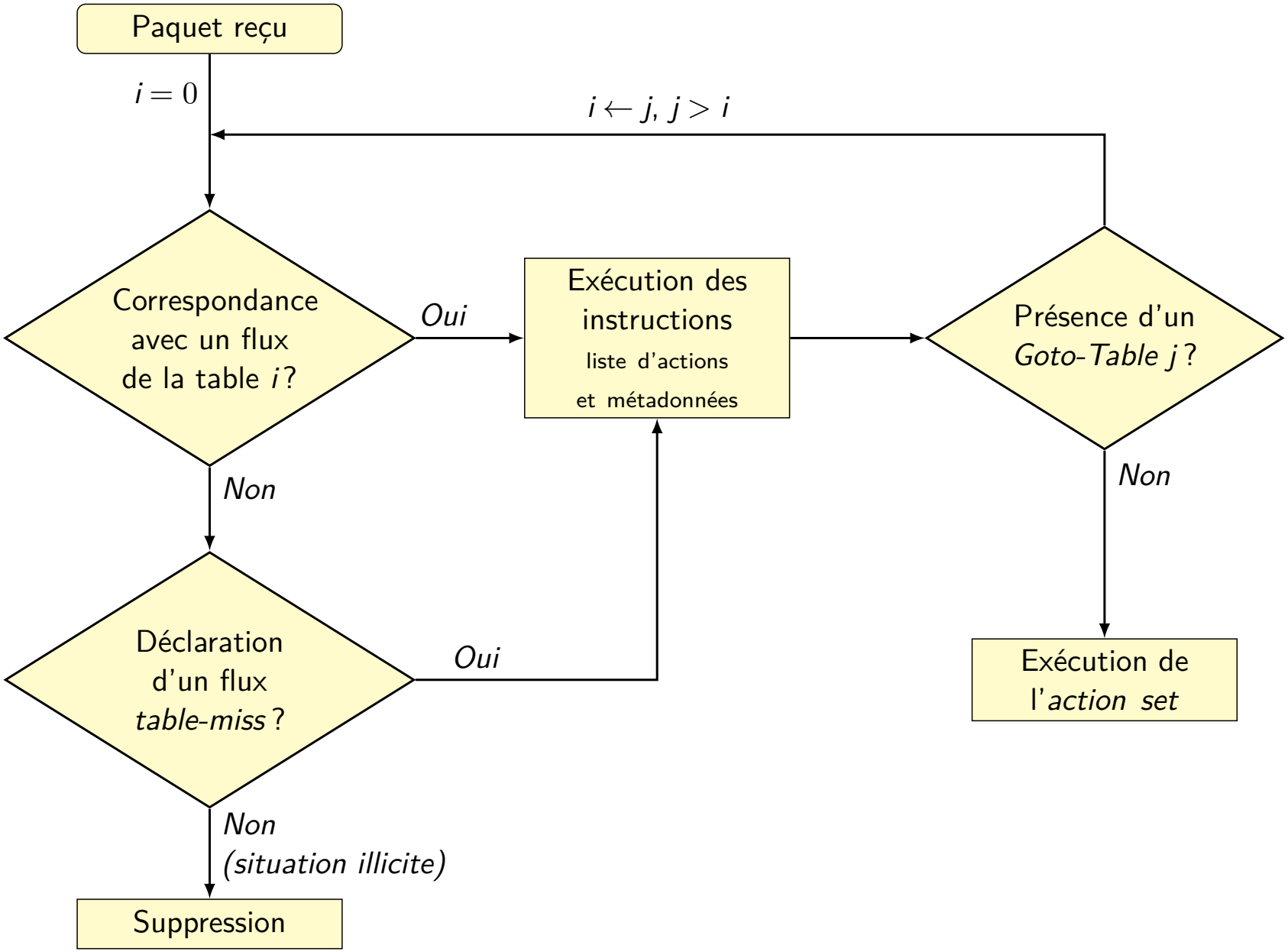
https://github.com/floodlight
Contact HP : mauricio.sanchez@hp.com

Annexes

Écriture d'une règle de routage avec OpenFlow 1.4



Traitement d'une trame avec OpenFlow 1.4



Temps de réponse d'un switch surchargé en flux