# Étude de vulnérabilité des architectures réseaux à faible latence

## Cas d'usage: Architecture L4S

Marius Letourneau [1] Guillaume Doyen [2] Rémi Cogranne [1]

[1]UTT - Institut Charles Delaunay

[2]IMT Atlantique

11 mai 2021

# Mise en contexte

- De nouvelles architectures réseaux et de nouveaux protocoles
- De nouveaux services à faible latence




Services faible latence $\Longrightarrow$ **nouvelles surfaces d'attaque**

Notre scope :

- Attaques visant
  - la **disponibilité**
  - la réduction de la **QoS**
  - l'augmentation de la **latence** e2e
- Architecture L4S

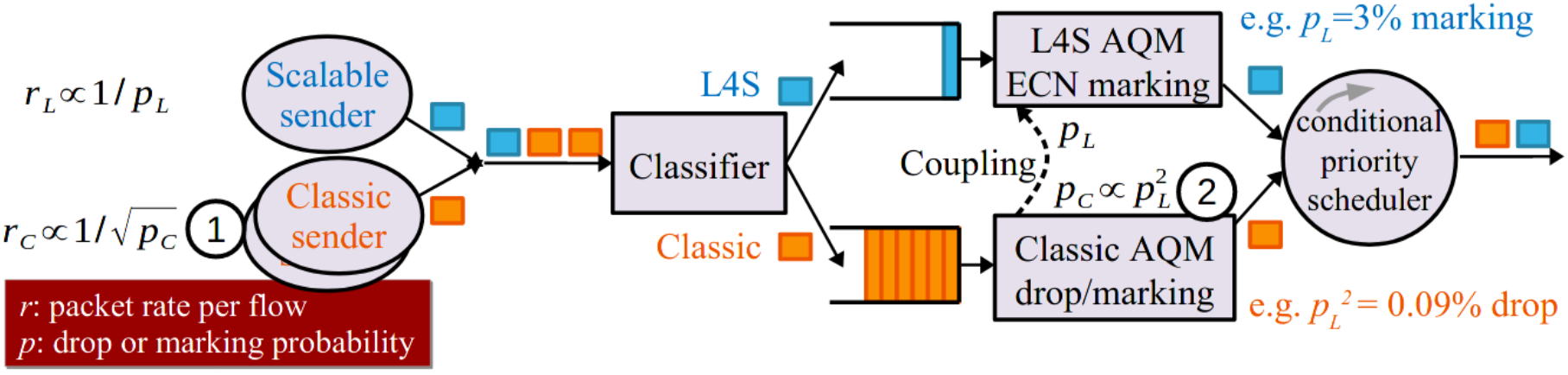
Défi à relever :

- Détection de ces attaques
  - Peu de traffic suffit à perturber
  - Difficulté de la détection
  - Performance de la détection ($\backsim$ms)

# Architecture L4S

Principes généraux :

- Coexistence assurée entre les flux Classiques ($C$) et L4S ($L$)
- Isolation de la latence entre les flux (C) et (L)




*Implementing the 'Prague Requirements'for Low Latency Low Loss Scalable Throughput (L4S)*
*Netdev 0x13, 2019*

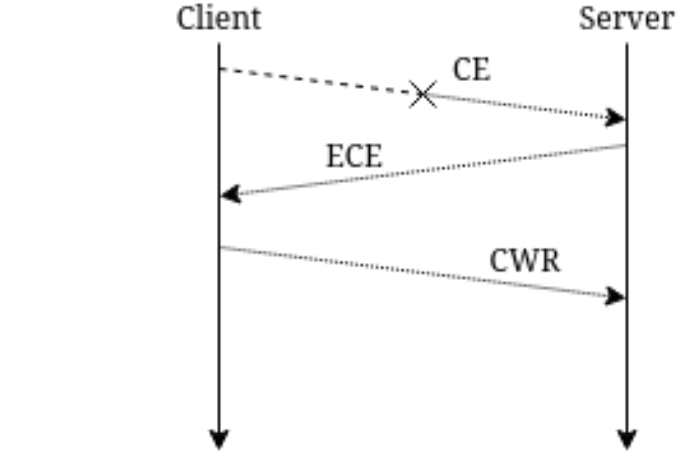
# Menaces sur la latence dans L4S

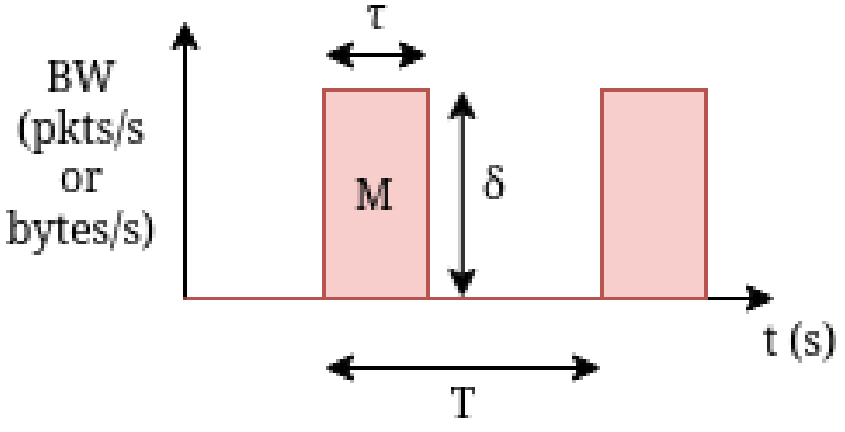
**Constat :**

Des *flux indésirables* peuvent mettre en défaut L4S

Par *flux undésirables* on entend :

- *Unresponsive flows* [DQCAQM]
- *Flux malformés* [Oljira20]
- *Misbehaving flows* [RFC7567]
  - Manipulation de protocole [Kothari11] :
    - Hacked ACK : opt-ack, lazy opt-ack, distributed opt-ack, ack-division
    - Hacked ECN : fausse notification, dissimulation, non réaction, mensonge
  - Malformation délibérée : attaque RoQ (et dérivées) [Guirguis04]



# État de la sécurité dans L4S

Quels mécanismes de protection existants contre les *flux indésirables* et contre-mesures anticipées ?

- *Unresponsive flows* :
  - Sacrifice de la performance (débit, délais ou drop)
- *Flux malformés* :
  - Scheduler adapté
  - Traffic shapping
  - Traffic policing pour gérer les cas pathologiques
- *Misbehaving flows*
  - Manipulation de protocole :
    - Hacked ACK : envoie d'ACK désordonnés, EFSM [Laraba20,Laraba21]
    - Hacked ECN : dissimulation : champ IP-ECN [RFC3168], EFSM [Laraba20,Laraba21]
  - Malformation délibérée : idem que pour flux malformés

# Comment attaquer la latence sur L4S ?

**Questions :**
*Quand* et *comment* attaquer la latence si l'objectif est d'être le plus **discret** possible tout en maximisant les **dégâts** ?

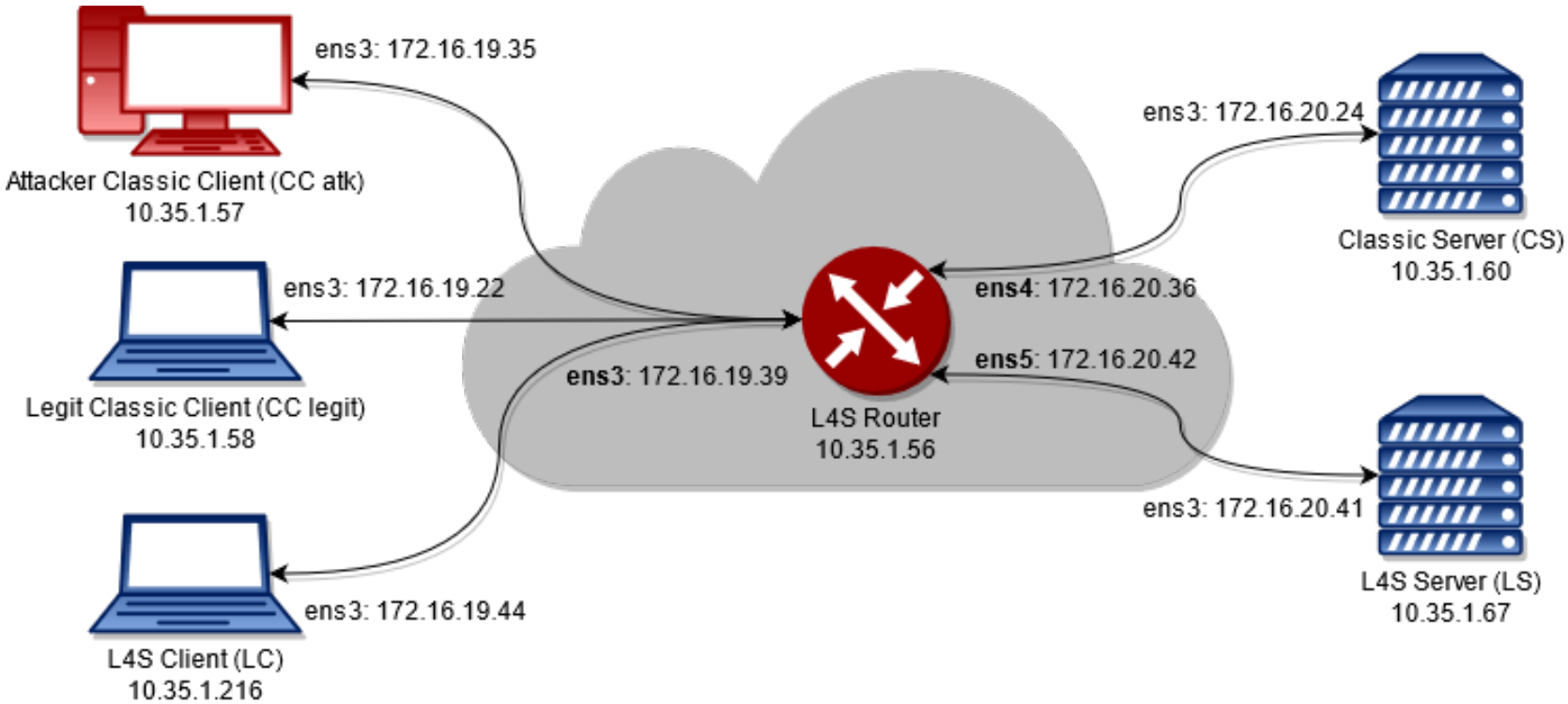
$\Longrightarrow$ Prendre avantage de la difficulté d'L4S à gérer les flux *bursty*[1] :

- Créer des flux malformés (désactiver TCP Pacing et `fq_codel` etc...)
- Exploiter la réactivité des autres à la congestion (i.e. créer des flux *unresponsive ECN-capable*)
- Créer des bursts volontaire de paquets (attaque RoQ / Low-Rate DoS ...)
- Effets bonus : bandwidth stealing !

1. la question de quantifier l'impact reste ouverte [draft-ietf-tsvwg-ecn-l4s-id]

# Travaux en cours et à venir

- Évaluation empirique et mesure d'impact des attaques sur notre banc de test
- Élaboration d'un détecteur **performant**
  - Basé sur une approche statistique
- Intégration du détecteur dans un micro-service orchestrable
  - Environnement NFV (OpenNetVM)
  - Monitoring P4
  - Scrubbing facilities

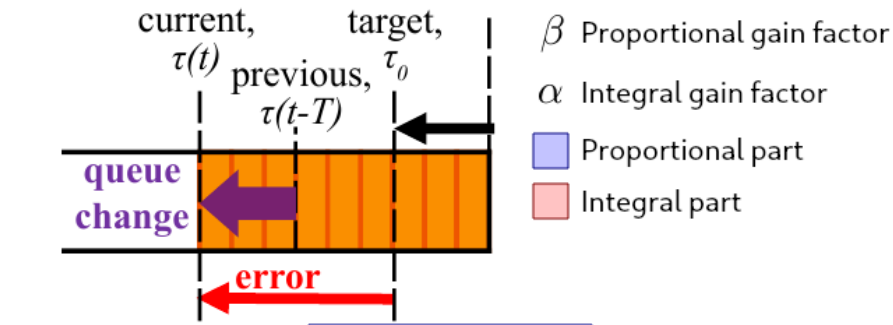


# Références


- [Guirguis04] : *Bandwidth Stealing via link-targeted RoQ attacks*, IASTED CCN 2004
- [RFC7567] : *IETF Recommendations Regarding Active Queue Management*, IETF, 2015
- [DeSchepper16] : *PI² : a Linearized AQM for both Classic and Scalable TCP*, CoNEXT 2016
- [Prague] : *Prague Congestion Control*, IETF 2021
- [DQCAQM] : Section 2.5. Normative Requirements for a DualQ Coupled AQM ; Section 4. Security Considerations ; Annexe A.1. Pass #1 : Core Concepts et Pseudo-codes
- [draft-ietf-tsvwg-ecn-l4s-id-14] : Section 6. L4S Experiments
- [Oljira20] : *Validating the Sharing Behavior and Latency Characteristics of the L4S Architecture* - SIGCOMM, 2020
- [Laraba20] : *Defeating Protocol Abuse with P4 : Application to Explicit Congestion Notification* - IFIP, 2020
- [Laraba21] : *Mitigating TCP Protocol Misuse With Programmable Data Planes* - TNSM, 2021
- [Kothari11] : *Finding protocol manipulation attacks* - SIGCOMM, 2011
- [L4S-ARCH] : *Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service : Architecture* - IETF TSVWG, 2020

# Génération de la probabilité de signalement

- Régulateur proportionnel intégral : système de contrôle permettant d'améliorer la qualité d'un asservissement en boucle fermée
- Un **signal de commande** est délivré à partir de la différence entre la **consigne** et la **mesure**
- Pour le réseau, la probabilité $p$ de signalement de congestion est le signal de commande (basé sur le taux de remplissage de la file d'attente d'un équipement)
- La consigne s'exprime en nombre de paquet ou en délais, les mesures périodiques rendent compte de l'**erreur** et de sa **variation**




*PI² : A Linearized AQM for both Classic and Scalable TCP, CoNEXT'16, USA, 2016*

- D'après les modèles de fonction de réponses [2] à la probabilité de congestion :
  - $cwnd = \frac{K}{p^B}$ la fonction de réponse du CCA utilisé
  - Reno : $B = \frac{1}{2}$
    CUBIC : $B = 0.7$
    DCTCP : $B = 1$

$\Longrightarrow$ $p$ est donc le liant entre le queuing delay et les CCA !

2. Response function : [DeSchepper16] [Prague]