État de l’art : Émulation de trafic vidéo pour NexSlice

Objectifs du projet

NexSlice est une plateforme basée sur OpenAirInterface et UERANSIM, elle permet d’expérimenter le network slicing sans matériel radio physique –lien(“https://github.com/AIDY-F2N/NexSlice”)[1]. Actuellement limitée aux tests iperf3 et ping, l’intégration de trafic vidéo applicatif devient nécessaire pour évaluer la QoS et la QoE dans des conditions réalistes, conformément aux objectifs du cours “Infrastructure intelligente logicielle des Réseaux mobiles”.

Cadre normatif et architecture 5G

Standards 3GPP et modèle QoS

Le **TS 23.501** (Technical Specification) définit l’architecture 5G avec les Network Slice Instances (NSI) composées de sous-réseaux RAN, Core et Transport isolés –lien(“https://www.etsi.org/deliver/etsi_ts/123500_123599/123503/18.07.00_60/ts_123503v180700p.pdf”)[2]. Comme étudié dans le cours (chapitre 2), l’architecture 5GC repose sur une Service-Based Architecture (SBA) où chaque fonction réseau (AMF, SMF, UPF, PCF, NSSF) communique via des interfaces HTTP/2 REST.

Le modèle QoS utilise les **5QI** (5G QoS Identifier) définissant les caractéristiques end-to-end. Pour le streaming vidéo, les 5QIs recommandés sont –lien(“https://itecspec.com/spec/3gpp-23-501-5-7-qos-model/”)[3] :

- **5QI 2** : streaming conversationnel, PDB 150ms, PER 10^-3
- **5QI 3** : gaming/VR temps réel, PDB 50ms, PER 10^-3
- **5QI 4** : streaming buffered, PDB 300ms, PER 10^-6
- **5QI 7** : streaming interactif non-GBR, PDB 100ms, PER 10^-3

Modèles de trafic vidéo standardisés

Le **TR 26.925** (Technical Report) spécifie les débits typiques : 720p HD (2-5 Mbps), 1080p Full HD (5-12 Mbps), 4K UHD (8-25 Mbps) –lien(“https://www.etsi.org/deliver/etsi_tr/126900_126999/126925/18.01.00_60/tr_126925v180100p.pdf”)[4]. Le **TR 26.926** définit les modèles statistiques du trafic incluant distribution log-normale des tailles de trames, périodicité selon le framerate (33.3ms pour 30fps), et jitter réseau –lien(“https://www.3gpp.org/dynareport/26926.htm”)[5].

Le **TS 28.554** spécifie les KPIs end-to-end essentiels pour évaluer le slicing : délai downlien NG-RAN, throughput par slice, PDU session success rate, et QoS flow retainability –lien(“https://www.etsi.org/deliver/etsi_ts/128500_128599/128554/18.05.00_60/ts_128554v180500p.pdf”)[6].

Architecture NexSlice et virtualisation

Composants OpenAirInterface

NexSlice déploie le cœur 5G OAI (Release 16/17) avec l’architecture NFV étudiée au chapitre 3 du cours. Les fonctions réseau virtualisées incluent –lien(“https://github.com/AIDY-F2N/NexSlice”)[1]–lien(“https://openairinterface.org/wp-content/uploads/2023/11/Adlen-Ksentini-EURECOM.pdf”)[7] :

- **AMF** : gestion accès et mobilité
- **SMF** : gestion sessions PDU avec support multi-slices
- **UPF** : plan utilisateur avec routage par slice
- **NSSF** : sélection du slice approprié selon S-NSSAI
- **PCF** : contrôle des politiques QoS

Chaque slice NexSlice utilise un couple SMF/UPF dédié, tandis que les fonctions de contrôle (AMF, AUSF, UDM) sont partagées, conformément aux principes NFV du cours (section 3.2).

Simulation radio avec UERANSIM

UERANSIM simule UE et gNB 5G SA sans matériel radio, supportant 100+ UE simultanés avec une latence < 1ms –lien("https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/blob/master/docs/DEPLOY_SA5G_WITH_UERANSIM.md")[8]–lien("https://github.com/aligungr/UERANSIM")[9]. Cette approche correspond à la vision du cours (chapitre 3.4) d'une infrastructure flexible découplant logiciel et matériel.

Configuration type UERANSIM –lien("https://github.com/AIDY-F2N/NexSlice")[1] :

gNB configuré pour slice eMBB-Video

Implémentation streaming vidéo

Pour le streaming vidéo, l'approche documentée par EURECOM utilise VLC ou GStreamer avec transport RTP/UDP –lien("https://www.eurecom.edu/publication/7497/download/comsys-publi-7497.pdf")[10].

——————————————————————————

Protocoles streaming et métriques QoE

Protocoles adaptatifs

MPEG-DASH et **HLS** dominent le streaming adaptatif avec manifestes MPD/M3U8 et segments fMP4/MPEG-TS –lien("https://websites.fraunhofer.de/video-dev/automated-abr-testing-for-dash-and-hls-media-players/")[12]. Pour applications temps réel (< 100ms latency), **RTP/RTCP** reste privilégié avec transport UDP et feedback RTCP pour monitoring qualité –lien("https://www.eurecom.edu/publication/7497/download/comsys-publi-7497.pdf")[10]–lien("https://repositorio.upct.es/server/api/core/bitstreams/f8266800-d721-4619-8d2e-bce544e179e6/content")[13].

Évaluation QoE

ITU-T P.1203 fournit un modèle paramétrique calculant MOS (Mean Opinion Score) [1-5] depuis métadonnées streaming (bitrate, résolution, stalling, initial delay) avec corrélation 0.89 vs évaluations subjectives –lien("https://repositorio.upct.es/server/api/core/bitstreams/f8266800-d721-4619-8d2e-bce544e179e6/content")[13]–lien("https://github.com/itu-p1203/itu-p1203")[14]. **VMAF** (Netflix) agrège VIF/DLM/SSIM via SVM, échelle 0-100 corrélant mieux avec perception humaine que PSNR/SSIM –lien("http://www.rtcbits.com/2018/10/measuring-webrtc-video-quality-for.html")[15].

Le Machine Learning permet de prédire la QoE depuis les métriques réseau (RTT, throughput, jitter) avec accuracy 87-91% via Random Forest –lien("https://cora.ucc.ie/items/2468c5cc-c004-4cb9-80f6-35913869d99f")[16], applicable aux scenarios NexSlice pour validation automatisée.

Orchestration Kubernetes : Vision NexSlice cloud-native

Transition Docker Compose vers Kubernetes

NexSlice propose deux environnements (chapitre 10 du cours) –lien("https://github.com/AIDY-F2N/NexSlice")[1] :

1. **Docker Compose** : déploiement rapide pour apprentissage initial
2. **Kubernetes/K3s** : orchestration avancée pour projets et scénarios réalistes

Cette progression pédagogique reflète l'évolution industrielle vers les Cloud-Native Network Functions (CNF) étudiées au chapitre 4 du cours.

Avantages Kubernetes pour NexSlice

Kubernetes transforme NexSlice en plateforme d'expérimentation avancée, conforme aux principes du chapitre 4 (Cloud Native et Conteneurisation) –lien("https://ieeexplore.ieee.org/document/10014753")[17]–lien("https://www.eurecom.fr/publication/6417/download/comsys-publi-6417.pdf")[18] :

Orchestration native VNF/CNF :
- Auto-scaling horizontal basé CPU/mémoire (HPA)
- Self-healing avec restart automatique des pods
- Rolling updates sans interruption service
- Multi-cluster pour tests edge/core/cloud

Isolation et QoS par slice : Kubernetes offre trois mécanismes d'isolation réseau étudiés au cours (section 5.2.1) –lien("https://arxiv.org/html/2502.02842v1")[19] :
1. Resource reservation CPU/memory via `requests/limits`
2. Bandwidth limitation per-pod (Cilium CNI)
3. Priorisation du CPU par cgroups

Configuration type pour UPF slice Premium :

resources: requests: cpu: "1000m" – 1 vCPU garanti memory: "2Gi" – 2GB RAM réservé limits: cpu: "2000m" – Max 2 vCPU memory: "4Gi" – Max 4GB

Kube5G et automatisation

Kube5G développé par EURECOM constitue un framework opérateur Kubernetes pour 5G –lien("https://www.eurecom.fr/publication/6417/download/comsys-publi-6417.pdf")[18], aligné avec les concepts d'orchestration du chapitre 6 du cours :

Architecture technique :
- Custom Resource Definitions (CRD) pour slices réseau
- Operator pattern avec reconciliation loops (MAPE-K)
- Helm charts pour packaging NF

Performances mesurées –lien("https://www.eurecom.fr/publication/6417/download/comsys-publi-6417.pdf")[18] :
- Provisioning : 1min12s vs 56s docker-compose (overhead 29%)
- Reconfiguration : 2x plus rapide via redéploiement sélectif
- Scaling horizontal UPF : 1 à 5 replicas en < 30s

Automatisation closed-loop pour NexSlice

Le cours (section 6.5) décrit trois approches complémentaires applicables à NexSlice :

1. Policy-based automation : règles explicites via OPA/Rego

package policies.upf scale_out { input.metrics.cpu95p > 0.8 input.labels.slice == "eMBB-Video" input.metrics.current_replicas < 5 }

2. Closed-loop automation : cycle MAPE-K (Monitor-Analyze-Plan-Execute)
- Monitor : Prometheus collecte métriques UPF (CPU, PDR hits, latency N3)
- Analyze : détection seuil cpu>80% durant 5min
- Plan : ajouter 1 réplique UPF, rééquilibrer sessions via SMF
- Execute : `kubectl scale` + update SMF N4

3. AI-driven automation : prédiction charge via LSTM, admission control proactif

Exemple HPA personnalisé pour NexSlice :

apiVersion: autoscaling/v2 kind: HorizontalPodAutoscaler metadata: {name: upf-hpa} spec: scaleTargetRef: {kind: Deployment, name: upf} minReplicas: 1 maxReplicas: 8 metrics:

type: Pods pods: metric: {name: upf_cpu_usage} target: {type: AverageValue, averageValue: “700m”}

Migration NexSlice vers Kubernetes
Procédure technique alignée avec le cours (section 6.3.1) –lien(“https://www.ieeelcn.org/prior/LCN46/lcn46demos/Demo_11_1570761822.pdf”)[20] :

1. Conversion docker-compose ⟶ K8s avec Kompose :

   kompose convert -f docker-compose-basic-nrf.yaml -o k8s/

2. Ajustements manuels

   - PersistentVolumeClaims pour MySQL OAI
   - NetworkPolicies pour isolation slices
   - ConfigMaps pour configurations AMF/SMF

3. Déploiement K3s

   kubectl create namespace nexslice kubectl apply -f k8s/ -n nexslice

4. Monitoring Prometheus/Grafana

   helm install prometheus prometheus-community/kube-prometheus-stack -n nexslice

Sources :

Documentation académique française
Télécom SudParis : Le cours “Infrastructure intelligente logicielle des Réseaux mobiles” (coord. Badii Jouaber) couvre NFV/SDN (chapitre 3), Cloud Native (chapitre 4), Network Slicing (chapitre 5), et Orchestration (chapitres 6-7), constituant la base théorique de NexSlice.

EURECOM : Centre de recherche de référence en France pour 5G, développeur principal d’OpenAirInterface –lien(“https://www.eurecom.edu/publication/7497/download/comsys-publi-7497.pdf”)[10]–lien(“https://www.eurecom.fr/publication/6417/download/comsys-publi-6417.pdf”)[18]. Contributions majeures : Kube5G, intégration streaming vidéo OAI, études performance slicing.

Standardisation européenne
ETSI (European Telecommunications Standards Institute) : Organisme européen définissant :
- NFV MANO (Management and Orchestration) –lien(“https://www.etsi.org”)[21]
- MEC (Multi-access Edge Computing)
- Standards ZSM (Zero-touch Service Management)

Ces standards sont intégrés aux spécifications 3GPP utilisées par NexSlice.

Implémentation recommandée pour NexSlice

Architecture proposée
Composants NexSlice enrichis :
1. **OAI 5G Core** (Kubernetes) : 2-3 slices (eMBB-Video SST=1/SD=0x000001, Premium-4K SST=1/SD=0x000002)
2. **UERANSIM** : 5-10 UE + gNB multi-slice
3. **Serveur streaming** : GStreamer sur pod Core Network
4. **Clients UE** : VLC/GStreamer dans namespaces UERANSIM
5. **Monitoring** : Prometheus + Grafana + cAdvisor (déjà intégrés NexSlice)

Configuration slicing vidéo
Provisioning base de données OAI (`oai_db.sql`) :

– Slice eMBB-Video HD INSERT INTO NetworkSlice VALUES (1, 'eMBB-Video', 1, '0x000001');

– Slice Premium 4K INSERT INTO NetworkSlice VALUES (2, 'Premium-4K', 1, '0x000002');

– UE assignment slice eMBB-Video INSERT INTO AuthenticationSubscription VALUES ('001010000000001', '8baf473f2f8fd09487cccbd7097c6862', 'opc', '8e27b6af0e692e750f32667a3b14605d', 1, '0x000001');

Scénarios de test
Test 1 : Isolation slicing (conforme exercice chapitre 5)
- UE1 (slice eMBB-Video) : streaming 720p@30fps, 3 Mbps
- UE2 (slice Premium-4K) : streaming 1080p@60fps, 8 Mbps
- Background traffic : iperf3 UDP 50 Mbps slice Best-Effort
- Validation : pas d'impact cross-slice sur qualité vidéo

Test 2 : Congestion et ABR
- Limitation bandwidth UPF : 10 Mbps total
- 3 UE streaming 1080p simultanés (demande 24 Mbps)
- Observation : rebuffering, adaptation bitrate, MOS dégradation

Test 3 : Closed-loop automation
- Charge progressive slice eMBB-Video
- Déclenchement auto-scaling UPF à cpu>80%
- Mesure : temps réponse, impact QoE durant scaling

Métriques KPI/QoE
KPIs réseau (via logs OAI + Prometheus NexSlice) :
- PDU Session Establishment Success Rate par slice
- E2E latency (ping UE-serveur)
- Throughput UL/DL par slice (iperf3)
- Packet loss rate (analyse RTCP)

Métriques vidéo (instrumentation GStreamer) :
- Framerate effectif vs configuré
- Bitrate moyen vs pics
- Stalling events et durée
- Jitter buffer occupancy

QoE estimation (post-processing) :
- VMAF score : comparaison source vs reçue
- ITU-T P.1203 Mode 1 : depuis logs stalling/bitrate
- MOS subjectif : tests utilisateurs (questionnaire ITU-T P.910)

Conclusion
L'intégration de trafic vidéo dans NexSlice nécessite la combinaison de standards 3GPP (QoS 5QI, KPIs TS 28.554), solutions open-source OAI/UERANSIM, protocoles streaming RTP/GStreamer, et métriques QoE standardisées (P.1203, VMAF). La migration vers Kubernetes, conforme aux enseignements du cours (chapitres 4 et 6), transforme NexSlice en plateforme CI/CD pour validation slicing 5G avec auto-scaling, isolation réseau, et automatisation closed-loop. Cette évolution positionne NexSlice comme référence pédagogique française pour expérimentation réseaux 5G cloud-native, alignée avec les pratiques industrielles et la vision 6G du cours (chapitres 7-9).