

ANALYSE COMPARATIVE DES PERFORMANCES DE TRANSMISSION

1. LENZ - Performances de transmission

Architecture de communication :

- **XpressNet** : technologie RS 485 qui par ses qualités offre une liaison haute gamme puisqu'il autorise davantage de possibilités non seulement en distance mais aussi en capacité à intégrer un nombre important de périphériques [Coffret centrale Z21 avec routeur / télécommande | Roco 10834 | Jura Modélisme](#)
- **RailCom** : signal de retour permet notamment de signaler quel train roule actuellement sur telle section (canton), mais permet aussi de donner la vitesse réelle d'une locomotive [Digital Command Control — Wikipédia](#)

Temps de transmission estimé :

- **Logiciel → Centrale** : 10-50ms (RS485/USB)
- **Centrale → Rails DCC** : 5-15ms (traitement interne)
- **Rails → Décodeur** : **Immédiat** (signal continu)
- **Total** : 15-65ms

Dégradation avec locomotives multiples :

- **Peu de dégradation** : RS485 haute qualité
- **Limite** : jusqu'à 31 appareils sur XpressNet [Lenz 20200 Centrale Digital plus LZV200 incl. amplificateur de puissance - Lokschuppen Berlinski](#)
- **Répétition DCC** : Commandes répétées automatiquement

2. Z21 - Performances de transmission

Architecture de communication :

- **WiFi** : une quinzaine d'appareils mobiles pouvant exploiter en même temps la Z21 [Centrale DCC Z21 — équipeTonTrain.com](#)
- **Ethernet** : Le port LAN est utilisé pour réaliser ces opérations au niveau du PC [Maurienne-trains](#)
- **Bus CAN** : Le bus CAN est le plus rapide et le plus fiable des bus DCC [Centrale DCC Z21 — équipeTonTrain.com](#)

Temps de transmission estimé :

- **Smartphone → WiFi → Z21** : 20-100ms (variable WiFi)
- **PC → Ethernet → Z21** : 5-20ms (plus stable)
- **Z21 → Rails DCC** : 5-10ms
- **Total WiFi** : 25-110ms
- **Total Ethernet** : 10-30ms

Dégradation avec utilisateurs multiples :

- **WiFi** : Dégradation notable avec 10+ appareils
- **Latence variable** : 50-200ms selon encombrement WiFi
- **Ethernet stable** : Pas de dégradation significative

3. LOCODUINO BOX - Performances de transmission

Architecture de communication :

- **WiFi ESP32** : Communication directe
- **Bus CAN** : Entre modules satellites
- **CommandStation-EX** : solutions DCC faciles à utiliser, abordables
[DCC-EX Team and Products EX-CommandStation \(DCC-EX Team and Products/EX-CommandStation\) Command Station - DCCWiki](#)

Temps de transmission estimé :

- **App → WiFi → ESP32** : 15-80ms (ESP32 optimisé)
- **ESP32 → Traitement** : 2-5ms (microcontrôleur rapide)
- **ESP32 → Rails DCC** : 2-5ms
- **Total** : 19-90ms

Dégradation avec charge :

- **ESP32 efficace** : Meilleure gestion que routeurs classiques
- **Bus CAN** : Très fiable pour satellites
- **Limite** : Dépend de la programmation

PROBLÈMES DE TRANSMISSION DCC SUR FEEDER

1. Nature du signal DCC

Principe physique :

- **Signal carré** : $\pm 15V$ commutant à 8-10 kHz
- **Données + Puissance** : Le signal codé envoyé sur la voie donne des ordres aux équipements tout en fournissant la puissance [Digital Command Control — Wikipédia](#)
- **Répétition** : La norme DCC recommande donc de répéter les commandes autant que possible, en laissant toutefois un petit délai de 5 milli-secondes entre commandes destinées à un même décodeur [Commande d'un réseau en DCC avec un module ARDUINO.](#)

2. Problèmes de transmission par les rails

Mauvais contacts :

- On sait tous que les mauvais contacts sont légion et une partie des commandes sont perdues [Commande d'un réseau en DCC avec un module ARDUINO.](#)
- **Conséquence** : Perte de paquets DCC
- **Solution** : Répétition automatique

Longueur des feeders :

- **Résistance électrique** : Chute de tension
- **Inductance** : Déformation du signal carré
- **Capacité parasites** : Atténuation haute fréquence

3. Impact sur les locomotives multiples

Partage de bande passante DCC :

- **1 locomotive** : Commande toutes les 20-50ms
- **10 locomotives** : Commande toutes les 200-500ms par loco
- **20+ locomotives** : Réponse dégradée, saccades possibles

Solution par secteurs :

- **Boosters multiples** : Division du réseau
- **Feeders courts** : <100m maximum recommandé
- **Qualité contacts** : Maintenance critique

COMPARAISON PERFORMANCES RÉELLES

SCÉNARIO 1 : 5 locomotives, pilotage smartphone

Système	Temps total	Stabilité	Remarques
Lenz	40-80ms	★★★★★	Le plus stable
Z21	50-120ms	★★★★	Variable WiFi
LocoduinoBox	30-100ms	★★★★	Optimisable

SCÉNARIO 2 : 15 locomotives, logiciel PC

Système	Temps total	Stabilité	Remarques
Lenz	60-120ms	★★★★★	Excellent
Z21	40-80ms	★★★★★	Via Ethernet
LocoduinoBox	50-150ms	★★★	Dépend config

SCÉNARIO 3 : 6 utilisateurs simultanés

Système	Temps total	Stabilité	Remarques
Lenz	80-200ms	★★★★	Limite matérielle
Z21	100-300ms	★★★	Congestion WiFi
LocoduinoBox	60-250ms	★★★	Très variable

FACTEURS DE DÉGRADATION

1. Réseau électrique (tous systèmes)

- **Longueur feeders** : +10-50ms selon distance
- **Qualité contacts** : Perte paquets 5-20%
- **Nombre boosters** : Amélioration significative

2. Réseau informatique

- **WiFi surchargé** : x2 à x5 latence
- **Distance WiFi** : +20-100ms
- **Qualité routeur** : Facteur critique

3. Charge système

- **10+ locomotives** : +50% latence
- **Multiples utilisateurs** : +100-300% latence
- **Logiciel complexe** : +20-50ms traitement

RECOMMANDATIONS PAR USAGE

Performance critique (temps réel) :

1. **LENZ** avec connexion filaire RS485
2. **Z21** avec Ethernet (pas WiFi)
3. **LocoduinoBox** optimisé

Usage familial/ludique :

1. **Z21** (ergonomie compense latence)

2. **LocoduinoBox** (excellent rapport qualité/prix)
3. **LENZ** (plus technique)

Réseau complexe (20+ trains) :

1. **LENZ** (architecture robuste)
2. **Z21** avec boosters multiples
3. **LocoduinoBox** (si expertise technique)

La **latence réelle** dépend énormément de l'**installation physique** (qualité feeders, contacts) autant que du système choisi !