



Sideband OOB, l'ultimo dei Moi...Cavi

La trasferta che non abbiamo mai fatto, come valorizzare la tua rete OOB

Quando la rete cade, la console deve restare in piedi.

Christian Ciofi

Senior System & Network Administrator — Intendo S.r.l.

Elia Aielli

Network Engineer — Enegan S.p.a.



Eurekalabria · Idee in connessione



Chi siamo

Due percorsi diversi, lo stesso mestiere: far funzionare le reti degli altri



Christian Ciofi

Senior System & Network Administrator
Intendo S.r.l.

- Oltre 20 anni tra amministrazione di sistemi Linux e Windows, networking e virtualizzazione
- Progettazione di infrastrutture complesse, reti OT e POP di operatore
- MikroTik, Juniper, Proxmox, Docker — e molto codice scritto per togliersi problemi di dosso
- Sviluppo in proprio strumenti di gestione e automazione per il lavoro quotidiano



Elia Aielli

Network Engineer
Enegan S.p.A.

- Ingegnere delle telecomunicazioni (Università di Pisa), livornese
- Da oltre 10 anni progetta e gestisce infrastrutture di rete in ambito ISP
- Routing e security come terreno di casa
- Network automation: efficienza, affidabilità e scalabilità, semplificando il difficile



Il problema

L'apparato diventa irraggiungibile esattamente nel momento in cui serve raggiungerlo



Un commit di troppo

Cambio di configurazione su un router remoto: la sessione si chiude e non torna più. La rete di gestione passa da lì.



Upgrade andato storto

Firmware aggiornato, apparato che riparte in una configurazione che non ti aspettavi. Serve la seriale, subito.



VLAN o trunk sbagliati

Un errore su una porta e l'intero POP sparisce dal management. Il servizio dati magari continua a girare.



Link primario down

Il guasto è a monte: finché non arriva il tecnico dell'operatore, la sede è al buio dal punto di vista gestionale.

La risposta tradizionale

Salire in macchina · ore di viaggio, un cavo seriale e un portatile · il disservizio dura quanto la trasferta

Console server commerciale · ottimo prodotto, ma il costo per sede raramente si giustifica su decine di POP piccoli



L'idea

Portare la porta console dove serve, e renderla raggiungibile da un browser attraverso una via indipendente dalla rete che stiamo gestendo.

01

Hardware banale

Un single board computer con OpenWrt e un adattatore USB multiseriale. Poche decine di euro per sede, sostituibile ovunque.

02

Fuori banda davvero

Il client si collega da una connettività separata: DHCP di cortesia, VLAN dedicata o SIM LTE. Se cade la rete primaria, il tunnel continua a funzionare.

03

Zero configurazione manuale

Il server genera uno script per ogni client: si lancia sull'apparato e la sede è operativa. Nessuna chiave copiata a mano.

04

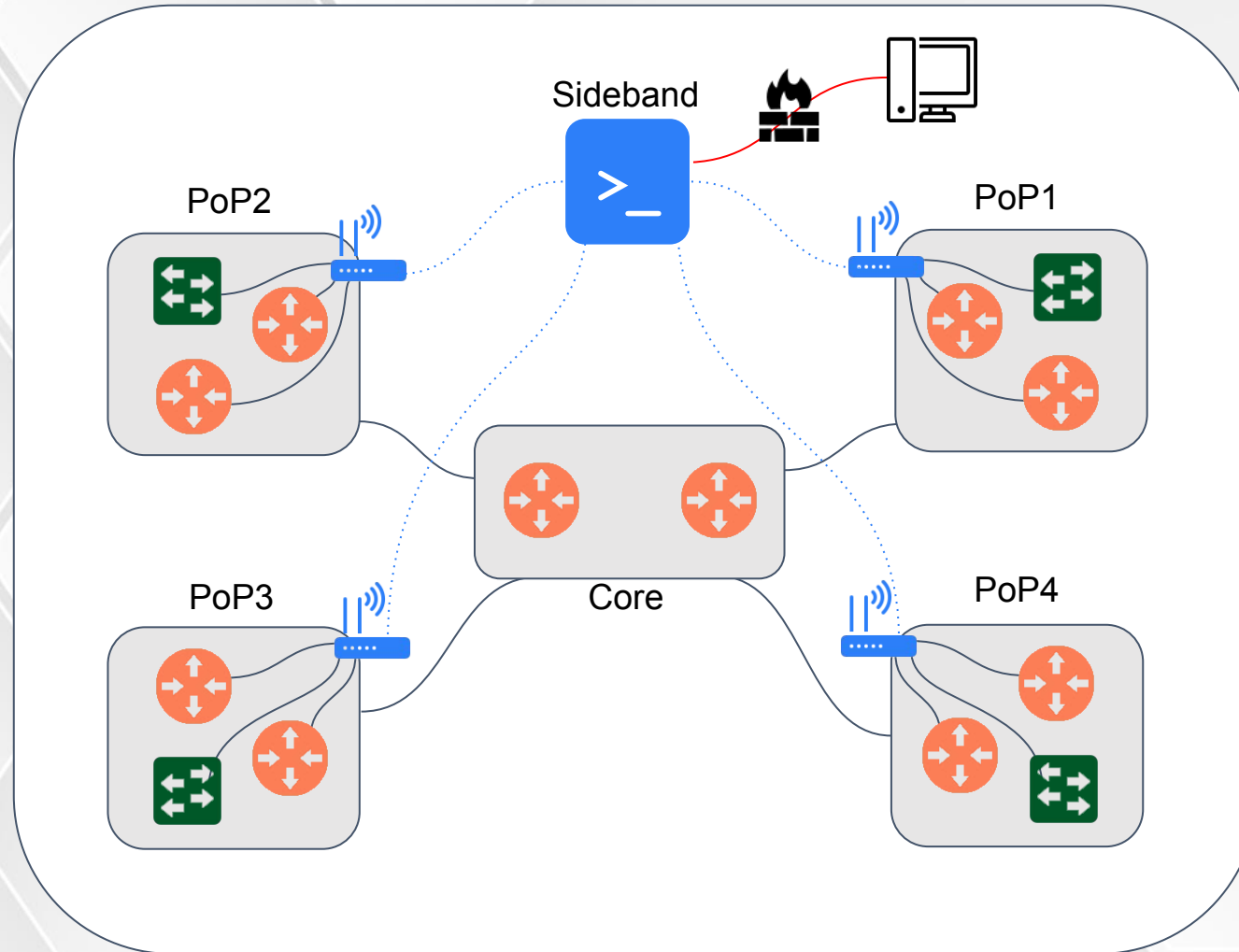
Una console, dal browser

Terminale nel pannello web, senza VPN sul portatile del tecnico e senza client da installare.



Architettura

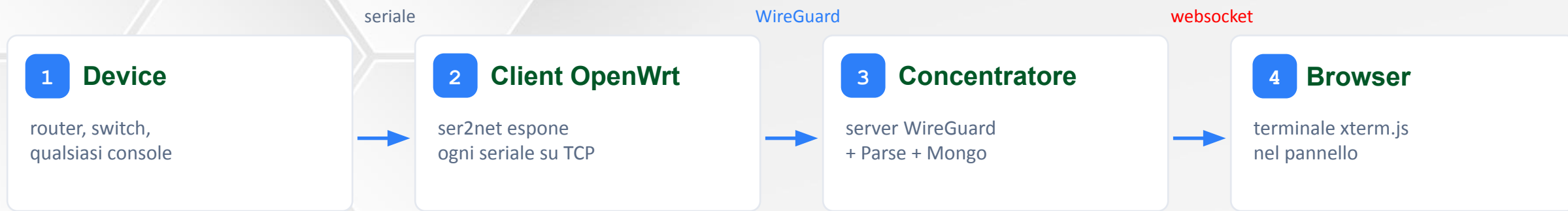
Dalla porta di console dell'apparato al terminale nel browser





Architettura

Dalla porta di console dell'apparato al terminale nel browser



Il punto chiave

Il tunnel viene sempre composto dal client verso il concentratore, con keepalive attivo. Le sedi possono stare dietro NAT, CGNAT o SIM LTE senza IP pubblico: non serve alcuna porta aperta in ingresso sul sito remoto.

Perché WireGuard

- Kernel space, leggero anche su hardware economico
- Configurazione autosufficiente: una coppia di chiavi e un endpoint
- Nessun controller esterno da cui dipendere
- Riaggancio immediato al cambio di WAN



Il concentratore

Quattro container, un solo host: lo stack che regge tutto

parse

logica

Parse Server e cloud code: modello dati, generazione degli script di provisioning, API del pannello. Serve anche il frontend statico.

mongo

dati

Persistenza di client, porte seriali, peer, sessioni e utenti.

backend

rete

Interfaccia WireGuard del concentratore, API interna per la gestione dei peer e bridge websocket verso le seriali.

dashboard

admin

Parse Dashboard per manutenzione e ispezione diretta dei dati.

Scelte di campo

Parse non è privilegiato

Non tocca la rete: chiede al backend, che è l'unico con NET_ADMIN.

Nessun reverse proxy interno

Un solo upstream verso Nginx Proxy Manager, che fa da terminazione TLS.

IPAM integrato

Ogni nuovo client riceve indice e indirizzo nella overlay senza intervento umano.

Chiavi generate lato server

La coppia WireGuard nasce nel pannello e finisce nello script: nessun copia-incolla.



Il pannello di gestione

Tutto quello che serve in esercizio, niente di più

> **Inventario dei client**

Nome, hardware, indirizzo nella rete overlay, numero di porte, stato.

> **Stato delle sessioni VPN**

Ultimo handshake, endpoint da cui arriva il client, traffico. Aggiornamento continuo.

> **Mappa delle porte seriali**

Etichetta, device, porta TCP e baudrate modificabili al volo, porte aggiunte o rimosse.

> **Console nel browser**

Terminale sulla singola porta, senza strumenti installati sulla macchina del tecnico.

> **Script di provisioning**

Generato e scaricato dalla scheda del client, già compilato con chiavi e indirizzi.

> **Utenti e credenziali**

Più operatori, password gestite dal pannello, sessioni tracciate.

Apertura di una console: il pannello crea una sessione e un ticket monouso valido 30 secondi → il browser apre il websocket con quel ticket → il bridge lo consuma e collega il socket alla porta ser2net del client. Il ticket muore all'uso.



Il provisioning del client

La parte che decide se il progetto è usabile o resta un giocattolo da laboratorio

Creo il client nel pannello → scarico lo script → lo lancio sull'apparato → la sede è operativa

1 Il pannello genera

Alla creazione del client vengono generate la coppia di chiavi WireGuard, l'indirizzo nella rete overlay e la mappa delle porte seriali. Il peer viene registrato sul concentratore in quello stesso momento.

2 Lo script è personale

Non è un template da riempire a mano: contiene già chiave privata, indirizzo, chiave pubblica del server, endpoint e la configurazione ser2net di quel client. Si scarica dalla scheda e si esegue.

3 Una sola esecuzione

Idempotente sui punti critici, verboso su cosa sta facendo. Chi lo lancia in sede può essere un tecnico che non ha mai visto il progetto.



Cosa fa lo script, passo per passo

Sette fasi, dall'orologio di sistema al servizio attivo

1

Verifica dell'orologio

Se la data non è valida, sincronizza via NTP: senza un clock corretto la verifica dei certificati fallisce e il gestore pacchetti non parte.

2

Installazione dei pacchetti

wireguard-tools, ser2net, mwan3 e i driver USB-seriale (FTDI, CH341, PL2303, CP210x), più l'estensione LuCI per WireGuard.

3

Configurazione del tunnel

Interfaccia e peer scritti via uci: chiave privata del client, chiave pubblica del server, endpoint, keepalive, rotte verso la overlay.

4

Firewall

Zona dedicata all'interfaccia del tunnel. La gestione entra solo da lì, il forwarding resta chiuso.

5

Naming stabile delle seriali

Regola hotplug che crea collegamenti per posizione fisica sul bus USB: la porta 2 resta la porta 2 anche dopo un riavvio o una disconnessione.

6

Configurazione di ser2net

Una porta TCP per ogni seriale, in ascolto solo sull'indirizzo del tunnel, con parametri e baudrate presi dal pannello.

7

Servizi e riaggancio

Abilitazione al boot e un trigger che riavvia ser2net quando l'interfaccia del tunnel sale, per non perdere il bind.

Perché tre di queste fasi esistono

Le fasi 1, 5 e 7 non erano nel progetto iniziale: le abbiamo aggiunte dopo aver sbattuto la testa in laboratorio e sul campo. La slide successiva racconta come.



Lo script, in concreto

Estratti dallo script generato per un client

Tunnel — configurazione via uci

```
# interfaccia: chiave e indirizzo del client,
# già compilati dal pannello
uci set network.wg0='interface'
uci set network.wg0.proto='wireguard'
uci set network.wg0.private_key='<chiave client>'
uci add_list network.wg0.addresses='10.99.0.7/32'

# peer: si compone sempre verso il concentratore
uci set network.peer='wireguard_wg0'
uci set network.peer.public_key='<chiave server>'
uci set network.peer.endpoint_host='vpn.sideband.it'
uci set network.peer.endpoint_port='51820'
uci set network.peer.persistent_keepalive='25'
uci add_list network.peer.allowed_ips='10.99.0.0/24'
uci commit network
```

Seriali — una porta TCP per console

```
connection: &sb0
  accepter: tcp,10.99.0.7,5000
  connector: serialdev,/dev/ttyUSB0,
             115200n8l,local
options:
  kickolduser: true
  max-connections: 1
```

Due dettagli che contano

- L'ascolto è vincolato all'indirizzo del tunnel: dalla WAN quelle porte non esistono.
- Una sola sessione per porta, con subentro: chi arriva dopo prende la console.



La connettività out of band

Il client deve poter uscire da qualunque strada disponibile

1 Ethernet in DHCP

La via più semplice: una presa di cortesia in sede, spesso su una rete di un altro operatore.

2 VLAN dedicata

Dove esiste già una rete di management separata, il client ci si appoggia con tag dedicato.

3 Modem LTE

La strada più indipendente: SIM dati, quasi sempre dietro CGNAT. Per il tunnel non è un problema.

mwan3 gestisce le uscite e il failover · il tunnel non entra nel bilanciamento · la rotta verso l'endpoint segue la WAN attiva

Hardware di riferimento

Raspberry Pi — una ethernet, wifi, USB. Il taglio minimo per una sede piccola.

Banana Pi R4 — quattro ethernet, due SFP, USB e modulo LTE. Per i POP dove serve di più.

Una scelta deliberata

Due sole piattaforme, entrambe con OpenWrt. Un parco hardware omogeneo costa poco in acquisto e ancora meno in manutenzione: un solo script, un solo insieme di problemi da conoscere.



Management out of band

Non solo console, anche ssh quotidiano

- La raggiungibilità IP è fornita dalla connettività OOB e dai tunnel wireguard
- Basta creare sull'OOB un bridge dedicato al management e scegliere uno dei seguenti modi per raggiungere la management dei PoP

1 Stretched L2

Un unica VLAN di management
Condivisa tra regioni geografiche diverse
Dominio di broadcast condiviso
Shared fate su tutta la rete

2 Routed L3

Una VLAN per PoP
Il router OOB annuncia la propria rete via IGP
Dominio di broadcast rilegato al PoP

A Pseudowire

Pseudowire punto punto tra sideband ed i PoP
Flood and learn
Spanning-tree geografico necessario

B EVPN/VXLAN

L2 over L3 tramite piano dati VXLAN
Mac address learning su control plane
Riduzione BUM
Minori rischi di broadcast storm

C IGP

Soluzione più semplice
Comunque richiesta per A e B
Permette introduzione di firewall L3
Firewall su PoP singolo o centralizzato

In tutte queste soluzioni c'è la possibilità di introdurre un Bastion / Jump Box per un layer aggiuntivo di sicurezza e possibile audit



Esempi di cosa puo' andare storto...

I problemi veri non erano quelli che ci aspettavamo



Un orologio sbagliato blocca tutto

Single board computer senza RTC: al primo avvio la data è fuori scala e la verifica dei certificati fallisce. Il gestore pacchetti restituiva un errore TLS generico, il problema era la data.

Sincronizzazione NTP prima di qualsiasi installazione



Il bind arriva prima dell'interfaccia

Il servizio si lega all'indirizzo del tunnel: se parte prima che l'interfaccia sia salita, fallisce in silenzio e sembra un problema di rete.

Trigger sulla salita dell'interfaccia



Il servizio non ripartiva al riavvio

La configurazione era corretta, ma scritta in un percorso diverso da quello che lo script di avvio legge. A mano funzionava, al boot no.

Percorso allineato al servizio di sistema



Le seriali si scambiano di posto

I nomi dei device dipendono dall'ordine di enumerazione. Gli adattatori economici spesso non hanno un seriale univoco su cui agganciarsi.

Identificazione per posizione fisica sul bus



Sicurezza

Uno strumento che apre console su tutti gli apparati va trattato per quello che è

Superficie ridotta al minimo

Le porte seriali ascoltano solo sull'indirizzo della overlay. Dalla rete pubblica del sito quelle porte non sono raggiungibili.

Accesso alla console a tempo

Ogni apertura richiede un ticket monouso con validità di pochi secondi, consumato all'uso.

Separazione dei privilegi

L'applicazione non ha capacità di rete: le richiede a un componente dedicato con un'API interna autenticata.

Nessuna porta in ingresso

Le sedi non espongono nulla: il tunnel parte sempre da loro verso il concentratore.

Tracciabilità

Ogni sessione registra chi ha aperto quale console e quando.

Il punto critico resta il centro

Chi entra nel concentratore entra ovunque. Va protetto come un bastion: accesso stretto, aggiornamenti, sorveglianza.



Sicurezza

Implementazioni possibili

01 HARDENING

nftables

Zone-based firewall
performance

iptables

Compatibilità legacy
know-how

firewall

hardware dedicato
firewall aziendale/uffici

02 ACCESS

VPN remote access

Wireguard
Openvpn

VPN orchestrata

Tailscale
Zerotier

web page

Accesso più semplice
necessita di più hardening

03 CONTROL

MFA

Yubikey
open authenticator

identity

permessi per utente
permessi per gruppo

reverse proxy

layer di sicurezza

WAF

protezione L5+



Dove vogliamo arrivare

Quello che manca, in ordine di utilità reale

prossimo passo

Applicazione a caldo

Modificare porte e baudrate dal pannello e vederle attive sul client senza rilanciare lo script.

prossimo passo

Registrazione delle sessioni

Log completo di quello che passa sulla console, ricercabile: utilissimo per capire cosa è successo davvero. Supporto via Bastion

prossimo passo

Monitoring raggiungibilità

Avviso quando una sede perde il tunnel, prima che serva la console.
Path OOB per l'observability

in valutazione

Trasporti VPN alternativi

Innestare ZeroTier o Tailscale dove il concentratore non ha endpoint pubblico.

in valutazione

Accesso per ruoli

Operatori con visibilità limitata ai propri apparati, per aprirlo a più squadre o a clienti.

idea

Automazione sugli apparati

Playbook ripetibili sulla seriale: gestione password, cfg iniziale, raccolta stato.
Rollback condiviso in caso di problemi

E il rilascio? Ci stiamo pensando. Il progetto nasce da un'esigenza interna e, se lo apriamo, vogliamo aprirlo fatto bene: documentato e con un percorso di installazione che regga fuori dal nostro laboratorio.



Sideband

Grazie

Domande, critiche e idee sono la parte più utile della giornata.

Christian Ciofi

Intendo S.r.l.
christian.ciofi@intendo.it

Elia Aielli

Enegan S.p.A.
e.aielli@enegan.it