SEBYONE

daas

Device as a Service

Presentazione della tecnologia: stato dell'arte, posizionamento, utilità pratica, settori di impiego e misure

Paolo Lo Giacco — p.logiacco@sebyone.it
Sebastiano Meduri — s.meduri@sebyone.it
Lorenzo Grillo — l.grillo@sebyone.it
Matteo Pagano — m.pagano@sebyone.it

3 novembre 2026 · libdaas 0.24.4 · daas-protocol 6.1.0

SEBYONE

Sebyone Srl · Via Pola 8, 87100 Cosenza · Piazza Capranica, Roma
sebyone.it · daasiot.com · info@sebyone.it · developers@sebyone.it
daas-api (libdaas) è distribuita con licenza MPL-2.0 · © 2019–2026 Sebyone Srl

Premessa

Questo documento riprende e aggiorna la presentazione ufficiale di daas al workshop IEEE, «Sebyone: Connecting the Future of IoT», che ha introdotto il paradigma Device as a Service Overlay Mesh e le sue quattro capacità fondamentali: rete overlay indipendente dai mezzi fisici, channelling dei protocolli, auto-sincronizzazione (dATS) e auto-provisioning. Rispetto a quella presentazione, qui si aggiungono lo stato dell'arte, il posizionamento, i risultati dei paper sul modello di identità e di routing e le misure sulla versione 0.24.4 della libreria.

Cos'è daas

daas (Device as a Service) è una tecnologia di rete che dà a ogni dispositivo un'identità propria, indipendente dall'indirizzo fisico, e lo fa dialogare con gli altri su qualunque mezzo di trasmissione, senza server, senza configurazione e senza infrastruttura di rete.

Il problema che risolve è semplice da enunciare. Oggi un dispositivo esiste in rete solo se qualcuno gli assegna un indirizzo, lo registra su un server, gli dice come raggiungere gli altri e gli fornisce un orologio comune. Ogni passaggio è un costo di installazione e un punto di guasto: se cade il server, il DHCP o il collegamento a Internet, i dispositivi smettono di parlarsi anche quando sono a un metro di distanza.

Con daas la rete nasce da sola. Un dispositivo acceso vicino a un altro lo scopre, si sincronizza con il suo tempo e riceve un'identità. Da quel momento può scambiare dati con chiunque appartenga alla rete, anche attraverso più salti, anche cambiando mezzo fisico (Ethernet, Wi-Fi, Bluetooth, LoRa, Zigbee, seriale). L'identità lo segue: se cambia indirizzo IP, canale radio o percorso, resta lo stesso nodo.

La libreria di riferimento si chiama daas-api (libdaas), è scritta in C++, è distribuita con licenza MPL-2.0 ed è disponibile su Linux, macOS, Windows, Android, iOS, Raspberry Pi, OpenWrt ed ESP32. daas-iot è il prodotto di Sebyone costruito sopra. Il sito ufficiale è daasiot.com.

Lo stato dell'arte: come si collegano i dispositivi oggi

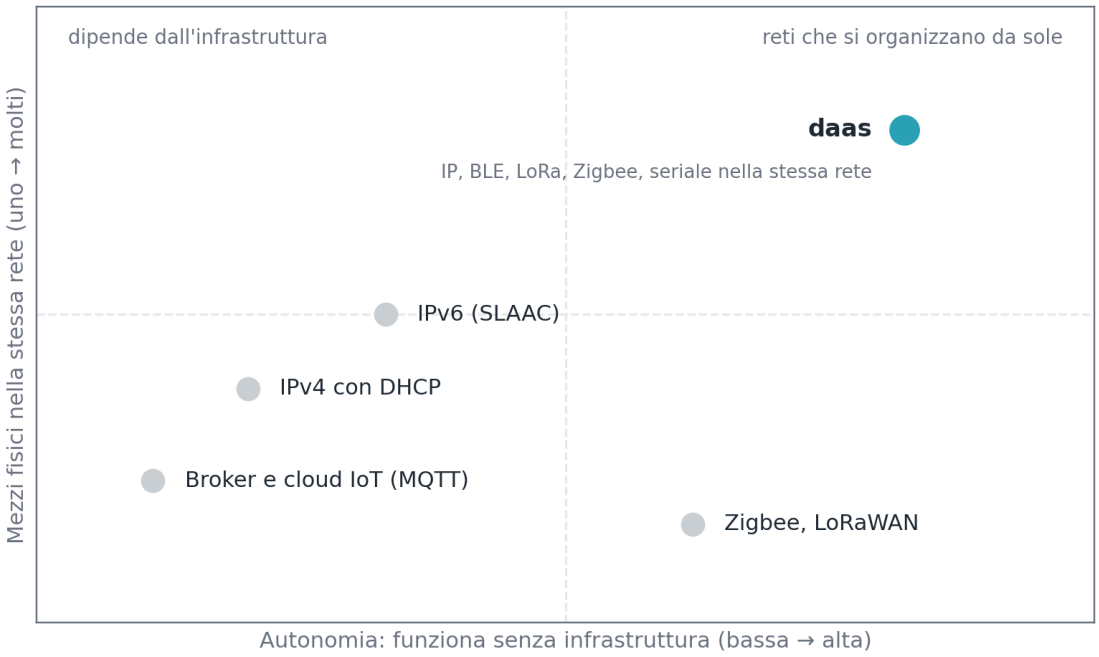
Tutte le soluzioni mature affidano l'identità dei dispositivi a qualcosa di esterno al dispositivo stesso: un registro, un server o un chip preassegnato. Nessuna porta nell'indirizzo il momento e il luogo in cui il nodo è entrato nella rete.

Approccio	Chi dà l'identità	Cosa serve per funzionare	Limite principale
IP classico (IPv4, DHCP)	un server DHCP o un amministratore	un piano di indirizzi e un server raggiungibile	l'indirizzo è un localizzatore, non un'identità: cambia se il nodo si sposta
IPv6 (SLAAC, RFC 7217)	il nodo stesso, da MAC o da hash	un router che annuncia il prefisso, più il Duplicate Address Detection	zero-touch solo con infrastruttura; l'indirizzo non dice né quando né dove il nodo è nato
Broker e cloud IoT (MQTT, piattaforme cloud)	la piattaforma, con credenziali per dispositivo	connettività Internet continua verso il broker	senza Internet i dispositivi vicini non si parlano; costi ricorrenti per nodo
Reti IoT a basso consumo (Zigbee, LoRaWAN)	il coordinatore o il network server, al join	un gateway o coordinatore per segmento	un solo mezzo fisico per rete; l'indirizzo è allocato da un'entità centrale
Identificativi a seme temporale (UUID v1/v7, Snowflake)	un generatore univoco più un contatore	un identificativo del generatore assegnato a priori	risolvono l'unicità, non il collegamento: non sono una rete

Il precedente più vicino a daas è IPv6: prefisso di rete più identificatore di nodo, auto-generazione per hash, rilevamento dei duplicati. Anche il concetto di rank, la distanza in salti dalla radice, esiste già in RPL (RFC 6550) per le reti a bassa potenza. daas riprende queste idee e le riunisce in un modello unico, inserendo nell'identità ciò che gli altri tengono fuori: il tempo di nascita e la posizione nella topologia.

Dove si posiziona daas

daas occupa lo spazio che gli altri lasciano scoperto: reti che devono esistere e organizzarsi da sole, senza infrastruttura, su mezzi fisici diversi. IPv6 ottimizza l'interoperabilità con il resto del mondo; daas ottimizza l'autonomia del segmento.



Mappa di posizionamento, qualitativa: più a destra meno infrastruttura serve, più in alto più mezzi fisici convivono nella stessa rete.

I due modelli sono complementari, non alternativi. daas può operare come livello di identità e auto-organizzazione sotto o accanto a IP: il segmento si forma, si sincronizza e assegna identità da solo; un gateway IP gli dà, quando serve, la raggiungibilità da Internet. Rispetto ai broker cloud, daas non richiede connettività continua né un canone per dispositivo; rispetto a Zigbee e LoRaWAN, non è legato a un solo mezzo fisico e non dipende da un coordinatore.

In sintesi, daas è la scelta quando:

- i dispositivi devono parlarsi anche senza Internet o senza server;
- la rete cambia spesso (unità che entrano, escono, vengono sostituite);
- convivono mezzi fisici diversi (IP, Bluetooth, LoRa, Zigbee, seriale);
- il costo di installazione per unità deve tendere a zero.

L'aspetto più innovativo per la ricerca

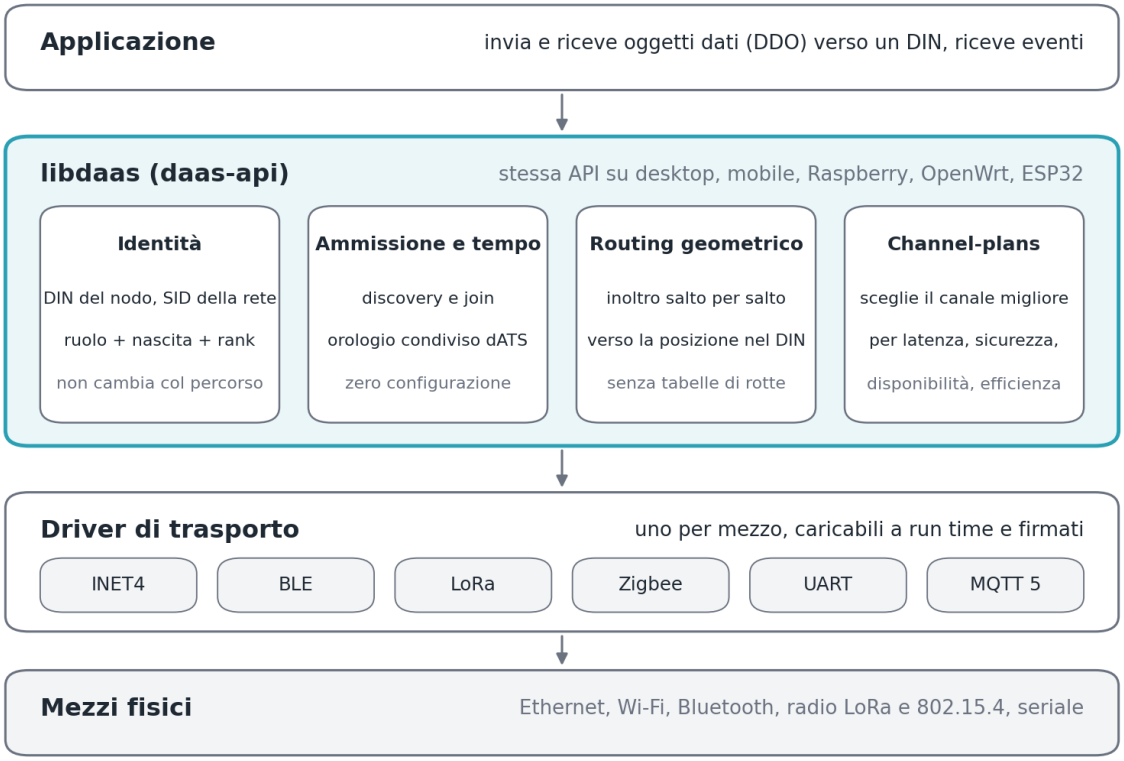
Dal punto di vista della ricerca sul networking, la novità di daas non è un protocollo più veloce né un nuovo mezzo fisico: è l'idea che l'identità di un nodo sia una coordinata. In tutte le architetture esistenti l'indirizzo è un localizzatore amministrato, assegnato da un registro o da un server, e non dice nulla su chi è il nodo, quando è entrato in rete e dove si trova nella topologia. In daas l'identità incorpora il tempo di nascita e la posizione nell'albero di ammissione, e da questo discendono tre proprietà che altrove richiedono tre sistemi separati: l'unicità senza registro, perché l'identità è univoca per costruzione; l'instradamento senza tabelle, perché la destinazione si calcola dall'identità stessa; la verificabilità senza crittografia, perché un'identità incoerente con il tempo e con la posizione è riconoscibile da chiunque. A questo si aggiunge la condizione fondante del modello: un nodo esiste solo quando ha un canale attivo e un tempo condiviso con un altro nodo. La rete non viene configurata; emerge.

Perché questa ricerca conta, rispetto a chi lavora sugli stessi temi. I temi che daas unisce sono oggi studiati separatamente da comunità diverse e attive. L'IETF lavora da anni sull'auto-organizzazione delle reti a bassa potenza (RPL e 6LoWPAN nei gruppi ROLL e 6lo), sull'auto-generazione degli indirizzi IPv6 e sul loro legame crittografico (RFC 7217, CGA), e con il gruppo ANIMA su reti che si avviano e si gestiscono da sole. La ricerca accademica sul routing geometrico e greedy, dagli embedding iperbolici di Internet (Krioukov, Papadopoulos, Boguñá) ai lavori di Kleinberg sul routing senza tabelle, ha mostrato che una coordinata può sostituire le tabelle di instradamento, ma l'ha fatto assegnando coordinate a reti già esistenti, non facendole nascere con l'identità. Le reti peer-to-peer e il Web decentralizzato (Kademlia, libp2p, IPFS, gli identificatori decentralizzati DID del W3C) hanno reso l'identità indipendente dal server, ma la appoggiano a chiavi crittografiche e a una rete IP sottostante. Le alleanze industriali (Thread, Matter, LoRa Alliance) hanno reso semplice l'ingresso di un dispositivo in rete, ma dentro un solo mezzo fisico e con un coordinatore. daas si colloca nel punto in cui queste linee non si incontrano ancora: un'identità che nasce con la rete, porta con sé tempo e posizione, serve da coordinata per instradare ed è verificabile prima della crittografia, su qualunque mezzo fisico. È una direzione in cui la comunità sta convergendo da lati diversi, e per questo i risultati di daas, in particolare l'unicità per costruzione e il routing calcolato dall'identità, sono rilevanti oltre il progetto: possono essere confrontati, verificati e ripresi da questi gruppi, e viceversa i loro risultati maturi, la rigenerazione su collisione e il legame crittografico, indicano a daas i passi successivi.

Comitato tecnico-scientifico e rapporti con le università. La ricerca su daas è accompagnata da un comitato tecnico-scientifico che ne indirizza le linee di sviluppo, ne valuta i risultati e ne verifica la coerenza con lo stato dell'arte. Il progetto mantiene inoltre rapporti di collaborazione con due università, con cui condivide attività di ricerca, tesi e sperimentazioni sul modello di identità, sulla sincronizzazione temporale e sul routing geometrico: un confronto che garantisce il rigore scientifico dei risultati e apre la strada a pubblicazioni e validazioni indipendenti.

Come funziona, in parole semplici

Un nodo daas diventa tale nel momento in cui apre un canale con un altro nodo e si sincronizza con il suo tempo. Da lì in poi quattro meccanismi fanno tutto il resto.



Architettura a strati: l'applicazione non vede né indirizzi IP né mezzi fisici, parla a un DIN e la libreria fa il resto.

Identità (DIN e SID). Ogni nodo ha un DIN, un numero a 64 bit che lo identifica. Ogni rete ha un SID, che identifica il segmento. Il DIN è composto da una parte scelta dall'applicazione (il ruolo: il sensore 12, il braccio 3), da una parte temporale assegnata al momento dell'ingresso e da un rank, la distanza in salti dal nodo che ha fondato la rete. Il DIN non cambia se il nodo cambia indirizzo, mezzo o percorso.

Ammissione (discovery e join). Un nodo nuovo manda un segnale di ricerca. Un nodo della rete risponde, lo sincronizza e gli assegna DIN e posizione. L'operatore non interviene. La rete cresce come un albero: ogni nodo entra attraverso un altro, e la sua posizione nell'albero finisce dentro la sua identità.

Tempo condiviso (dATS). Tutti i nodi di una rete condividono lo stesso orologio, con un errore massimo configurabile (100 ms di default). Un nodo invia dati applicativi solo quando è sincronizzato: i dati scritti prima restano in coda e partono appena il nodo è pronto.

Scambio dati e routing. L'applicazione invia oggetti dati (DDO) a un DIN; la libreria sceglie il canale migliore tra quelli disponibili (per latenza, sicurezza, disponibilità o efficienza, secondo i channel-plans) e, se il destinatario non è vicino, inoltra salto per salto verso la sua posizione nell'albero. Ogni nodo intermedio inoltra senza conoscere il destinatario: è il routing geometrico. Il trasporto è un dettaglio del driver: lo stesso nodo può avere canali IP, Bluetooth, LoRa e seriale verso vicini diversi.

A questi si aggiunge una difesa di ultima istanza: in caso di attacco, qualunque nodo può isolare l'intera rete raggiungibile (attack panic), e ogni nodo resta fermo finché un operatore non lo riavvia.

Spazio indirizzabile e routing sul piano complesso. Lo spazio degli indirizzi è ampio quanto basta per non essere mai un vincolo: ogni segmento può ospitare fino a 2^{64} nodi e una federazione di segmenti fino a 2^{128}, senza che nessuno debba tenere un registro dei singoli nodi, perché l'unicità è garantita dal modo in cui

l'identità viene costruita all'ingresso. Il modello di routing in evoluzione dà a questa identità anche una lettura geometrica: il DIN viene interpretato come un punto su un piano complesso, il piano di Gauss, normalizzato rispetto al nodo che ha fondato la rete. Il tempo di nascita e la posizione nell'albero diventano le due coordinate del punto, e la rete si legge come una mappa in cui ogni nodo sa, dalla sola identità del destinatario, in quale direzione e in quale settore inoltrare. Il percorso non va cercato né memorizzato: si calcola. Questo rende l'instradamento deterministico, leggero per i dispositivi più piccoli e indipendente dal numero di nodi, e lo stesso principio potrà essere esteso a reti federate. La formalizzazione matematica e le scelte implementative sono descritte nei paper dedicati.

Utilità pratica

Il vantaggio di fondo è uno: l'unicità e l'organizzazione della rete passano dal piano amministrativo al piano del protocollo. Ciò che oggi richiede un operatore, un foglio di indirizzi o un servizio centrale diventa una proprietà della rete stessa. Un'identità univoca per costruzione, datata alla nascita e posizionata nel grafo, è più di un indirizzo: è un'asserzione verificabile su chi è il nodo, quando è entrato e dove sta.

Problema oggi	Cosa cambia con daas
piano di indirizzamento e registrazione manuale di ogni unità	nessun piano né registro: il costo di provisioning per unità tende a zero
dipendenza da un registro o da un server raggiungibile online	funzionamento offline: l'allocazione è locale al segmento
riconciliazione tra identità di rete e asset logico (serie storiche, digital twin)	la coppia (parte utente, parte temporale) è già (asset, incarnazione): nessuna mappa esterna
clock non affidabili, id di evento non univoci	tempo condiviso e identità ordinate: eventi globalmente univoci e ordinati
indirizzi duplicati dopo la riunificazione di due partizioni	conflitto transitorio e auto-correttivo, rilevato per costruzione
spazio di indirizzamento esaurito o frammentato	2^{64} nodi per segmento, senza registro dei singoli nodi

Sul piano della sicurezza, l'identità daas offre tre controlli di plausibilità a costo nullo, prima e indipendentemente da qualunque strato crittografico: un DIN con parte temporale precedente alla rete o futura è forgiato; un nodo con rank incoerente con chi lo ha ammesso è anomalo; lo stesso DIN che compare da due indirizzi distinti segnala un dispositivo clonato. Questi controlli alzano il costo di un attacco e danno alla crittografia una base solida su cui ancorarsi, ma non sostituiscono l'autenticazione dell'ingresso in rete, oggi lasciata al livello applicativo.

Per chi sviluppa, il costo è contenuto: una libreria C++ con la stessa API su tutte le piattaforme, due modalità di esecuzione (con thread propri su desktop, server e smartphone; a ciclo cooperativo sui microcontrollori) e pochi byte di stato per nodo sulle piattaforme embedded.

Settori di impiego

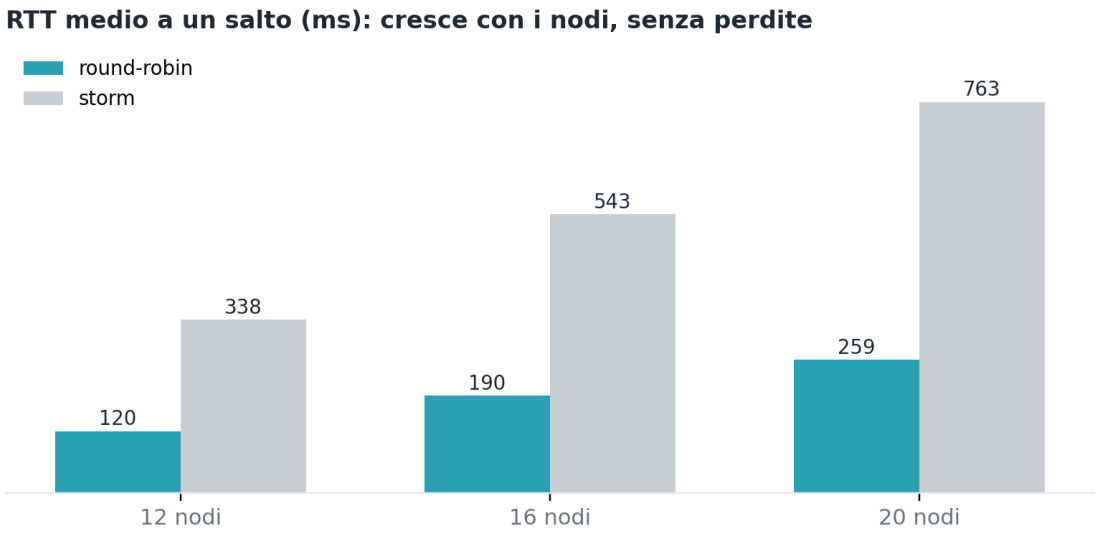
daas serve ovunque la configurazione preventiva non è praticabile: unità che entrano ed escono dal raggio radio, sostituzioni sul campo, siti senza infrastruttura.

Settore	Caso d'uso	Cosa porta daas
Robotica e sciami	flotte di robot, AGV, droni che si formano e si ricompongono sul campo	ingresso in formazione senza operatore; il rank nell'identità dice la distanza dal leader o dalla base e serve a scegliere i relè; un robot sostituito eredita il ruolo e riceve una nuova incarnazione, così i log restano consistenti
Flotte IoT industriali	sensori, contatori, telecamere installati a centinaia	nessun piano IP né registrazione per unità; mezzi fisici diversi nella stessa rete
Edge computing e siti remoti	impianti, cantieri, aree senza copertura stabile	la rete lavora offline; l'allocazione è locale al segmento e più segmenti si federano sotto un unico spazio di identità
Cybersicurezza e ambienti critici	reti dove un dispositivo estraneo è un rischio fisico oltre che informatico	identità verificabili per costruzione, rilevamento dei cloni, segmentazione nativa per cliente o sito, isolamento d'emergenza dell'intera rete
Serie storiche e digital twin	monitoraggio nel tempo di asset che cambiano hardware	l'identità distingue l'asset dalla sua incarnazione fisica: manutenzione predittiva e tracciabilità senza mappe esterne
Prossimità e indoor	localizzazione e tracking BLE, dispositivi personali	la stessa libreria gira su smartphone e microcontrollori e parla Bluetooth e IP nella stessa rete

Il tratto comune è lo stesso in tutti i casi: una rete che nasce da sola, riconosce i propri membri e continua a funzionare quando l'infrastruttura non c'è o non risponde.

Cosa è stato misurato

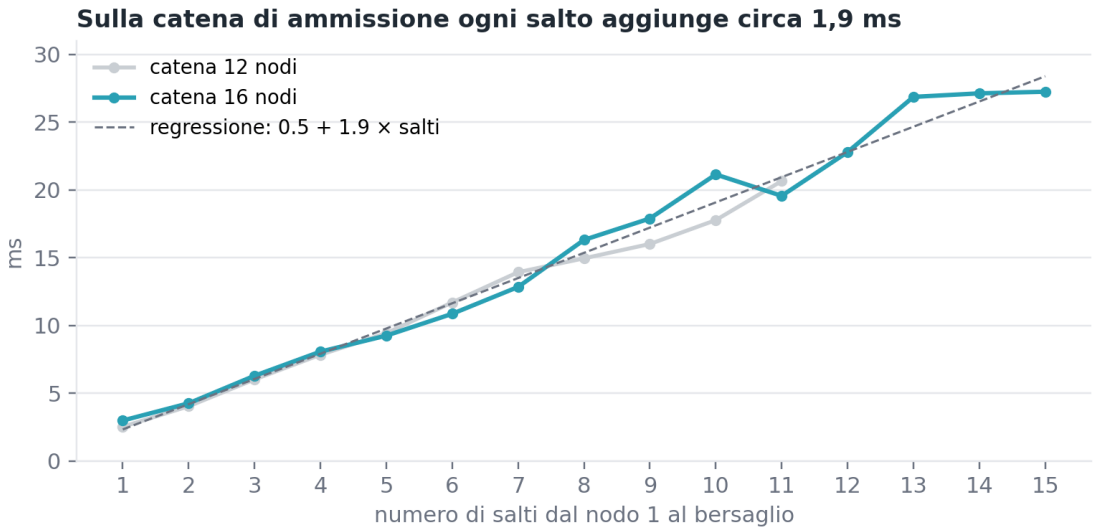
Con libdaas 0.24.4 (rapporto di test del 1° novembre 2026) una mesh completa è stabile a 12, 16 e 20 nodi: nessun crash, consegna delle sonde 100% in tutte e sei le prove, sia quando i nodi interrogano uno alla volta (round-robin) sia quando tutti interrogano tutti insieme (storm). Prima delle correzioni di questa versione la stessa prova a 12 nodi crashava e consegnava il 27–42%.



RTT medio a un salto, media dei per-nodo; 30 sonde per peer, un solo host. Fonte: rapporto RTT comparativo 12/16/20 nodi, 01/11/2026.

Il tempo di risposta a un salto cresce con il numero di nodi perché tutti i processi girano sullo stesso host: è un costo dell'ambiente di prova più che del protocollo. Sotto carico concorrente il tempo sale di 2,8–2,9 volte, ma senza perdite.

Su una catena reale di ammissione (ogni nodo entra attraverso il precedente), il routing geometrico consegna a ogni distanza fino a 15 salti e il tempo cresce in modo lineare, circa 1,9 ms per salto: 3 ms a 1 salto, 27 ms a 15 salti. L'unica sonda persa su 1500 è la prima verso il vicino diretto.



RTT medio per distanza in salti, 100 sonde per bersaglio, catene join+CX a 12 e 16 nodi. Fonte: stesso rapporto.

Limiti attuali e prossimi passi

I limiti noti sono di capacità e di maturità, non di modello.

Limite	Effetto	Prossimo passo
il driver IPv4 gestisce al massimo 20 connessioni per nodo	una mesh completa diretta arriva a 21 nodi; oltre, i peer in eccesso vanno raggiunti per instradamento	alzare il limite o instradare verso i peer oltre il ventesimo
il rank del DIN è a 4 bit	una catena di ammissione arriva a 16 nodi (15 salti)	topologie ad albero con profondità fino a 15; estensione del formato per linee più lunghe
l'ingresso in rete non è autenticato	chi partecipa alla negoziazione ottiene un'identità valida	legare l'identità a una chiave (come i CGA di IPv6) e autenticare il join
rilevamento dei duplicati non ancora implementato	un conflitto tra segmenti riunificati è transitorio ma non viene rilevato attivamente	verifica presso l'allocatore, come in 6LoWPAN
routing geometrico solo nelle build desktop	i nodi embedded interoperano ma non assegnano posizioni	portare la capacità CX su ESP32 e Raspberry Pi
misure su un solo host	il tempo a un salto cresce con il numero di processi sullo stesso host	ripetere le prove su host distinti per separare il costo del protocollo da quello dell'ambiente

La versione analizzata è la 0.24.4 con protocollo 6.1.0; la guida per gli sviluppatori e il rapporto di test sono i documenti di riferimento di questa presentazione.

Sviluppi in corso

Due evoluzioni del modello sono in fase di studio e vengono qui solo annunciate; i dettagli tecnici saranno oggetto di documenti dedicati.

Sicurezza per nodi fidefacenti. L'ingresso in rete sarà protetto da un modello distribuito di fiducia: un nodo nuovo viene ammesso perché altri nodi già membri, i fidefacenti, ne garantiscono l'identità. La fiducia non risiede in un server né in un'autorità centrale, ma si propaga lungo l'albero di ammissione, con lo stesso principio che oggi governa l'assegnazione delle identità. L'obiettivo è chiudere il limite indicato sopra, l'ingresso non autenticato, senza introdurre infrastruttura.

Filtro sul tempo condiviso (dATS). La sincronizzazione temporale tra i nodi sarà affinata con un filtro che stima e corregge nel tempo lo scostamento di ogni orologio locale rispetto al riferimento di rete. Il risultato atteso è un tempo di rete più stabile e un errore massimo più basso, anche su mezzi fisici lenti o irregolari come LoRa e Bluetooth, senza aumentare il traffico di sincronizzazione.

Conclusioni

daas fa nascere la rete da sola: identità, tempo e percorsi non si configurano, si formano con la rete. Non sostituisce IP, i broker cloud o le reti a basso consumo; copre ciò che questi non coprono: dispositivi che devono parlarsi anche senza infrastruttura, su mezzi fisici diversi, con un costo di installazione per unità che tende a zero.

Le misure confermano che il modello funziona nella pratica: una mesh completa stabile da 12 a 20 nodi senza perdite, e un routing multi-hop che consegna a ogni distanza fino al limite di profondità del protocollo con un costo lineare di circa 1,9 ms per salto. I limiti noti sono di capacità e di maturità, non di modello, e i prossimi passi, la sicurezza per nodi fidefacenti e il filtro sul tempo condiviso, li affrontano restando fedeli allo stesso principio: nessuna infrastruttura, nessuna autorità centrale.

Lo scenario a cui daas guarda è quello di una rete diffusa: migliaia di dispositivi in una città, in un distretto industriale o lungo una filiera, che si riconoscono e si parlano tra loro senza che nessuno li abbia registrati, anche quando il collegamento a Internet manca o cade. Sensori ambientali, contatori, telecamere, veicoli, robot e dispositivi personali formano un'unica rete che cresce man mano che si accendono, si riorganizza quando qualcuno si spegne e continua a funzionare per segmenti quando resta isolata. In questo scenario i dati non devono più salire a un centro per essere utili: restano vicini a chi li produce e a chi li usa.

È qui che daas incontra i nuovi paradigmi dell'intelligenza artificiale distribuita. Modelli eseguiti sul bordo della rete, agenti che collaborano tra dispositivi vicini, apprendimento federato tra nodi che non condividono i dati grezzi: tutto questo richiede esattamente ciò che daas fornisce di base, cioè un modo per sapere chi è un nodo, quando è nato, dove sta e che ora è, senza un server che lo certifichi. Un'identità verificabile per costruzione permette a un agente di fidarsi di ciò che riceve; un tempo condiviso permette di ordinare osservazioni e decisioni prese da nodi diversi; un routing calcolato dall'identità permette di raggiungere il nodo giusto senza una mappa centrale. In prospettiva, daas può essere il tessuto di comunicazione su cui un'intelligenza distribuita si forma, si coordina e si adatta, con lo stesso principio con cui si forma la rete: senza infrastruttura e senza autorità centrale.

Contatti

Sebyone Srl · Via Pola, Cosenza · Piazza Capranica, Roma
sebyone.it · daasiot.com · info@sebyone.it · developers@sebyone.it
Paolo Lo Giacco (p.logiacco@sebyone.it) · Sebastiano Meduri (s.meduri@sebyone.it) · Lorenzo Grillo (l.grillo@sebyone.it) · Matteo Pagano (m.pagano@sebyone.it)

daas-api (libdaas) è distribuita con licenza MPL-2.0 · © 2019–2026 Sebyone Srl. Documento redatto il 3 novembre 2026 su libdaas 0.24.4, daas-protocol 6.1.0.