

Domande su Mobile Edge Computing

- **Domanda:** Qual è la definizione e lo scopo principale del **Mobile Edge Computing (MEC)**, e come si differenzia dai modelli di cloud computing tradizionali o da altre architetture distribuite come il Grid Computing e lo Utility Computing? Quali sono le motivazioni chiave che hanno portato allo sviluppo del MEC?

Risposta: è un paradigma di calcolo distribuito che permette di offrire risorse computazionali e memoria in prossimità dell'utente. Senza permettere di dover connettersi con il Cloud. La differenza tra cloud computing e l'edge computing sta nel fatto che il cloud è posto in un data center a centinaia di chilometri di distanza dall'utente e questo dà un delay enorme rispetto a dei server MEC che hanno sicuramente memoria e risorse computazionali più limitati ma offrono servizi più vicini all'utente (offrendo un Quality of Experience migliore), permette una gestione più efficace contro la congestione, la sicurezza è migliore poiché i server MEC sono distribuiti e non permettono un attacco centralizzato.

- **Domanda:** Quali sono le **differenze e i vantaggi specifici del Mobile Edge Computing (MEC) rispetto al Mobile Cloud Computing (MCC)**? Inoltre, come si posiziona il MEC rispetto al **Fog Computing** e ai **Cloudlet** nella gerarchia dei paradigmi di calcolo distribuiti, considerando aspetti come latenza, risorse computazionali, sicurezza e applicazioni tipiche per ciascuno?

Risposta: Il Mobile Cloud Computing (MCC) è un tipo di paradigma di calcolo distribuito che è rappresentato come centrale, ha una latenza più grande del Mobile Edge Computing data dalle grandi distanze. Contiene, però, più risorse computazionali e più memoria, in ambito sicurezza, poiché centralizzato sarà meno sicuro e più soggetto a leaks. Le applicazioni per questo tipo di paradigma sono delay non sensitive e questo vuol dire che le applicazioni sul server sono app che non hanno bisogno di delay minimo.

Per quanto riguarda il MEC invece è più sicuro data dalla sua struttura più densamente distribuita, con meno capacità di calcolo e memoria ma in prossimità dell'utente e questo permette di avere applicazioni delay sensitive ovvero che danno importanza al ritardo dei dati (minimizzando). Il MEC è posto ai confini di reti e permette la gestione di dati senza passare dal backhaul, solitamente implementato nelle BS, il Fog Computing è formato da un insieme di server intermediari tra Edge e Cloud ma non ha un confine specifico. Il Fog Computing è dipendente dalla esistenza di un cloud e non può funzionare da solo, si ha una gerarchia di nodi (ognuno di loro cosciente degli altri), strutturato come una rete mesh e la comunicazione si ha o in verticale (comunica con i nodi di gerarchia più alta) o in orizzontale (comunica con altri peer).

Il Cloudlet è un altro paradigma di calcolo distribuito che rappresenta predecessore del MEC è un mini cloud che fa da intermediario tra datacenter e utente. Ha più prestazioni e

memoria rispetto ad un MEC rimuovendo però il ritardo dato dalla locazione del cloud, poichè posizionato in hotspot.

- **Domanda:** Descrivi in dettaglio l'**architettura gerarchica del Mobile Edge Computing (MEC)**

Quali sono i principali strati (User Layer, Edge Layer, Central Cloud Layer) e i loro componenti, le loro capacità e i tipi di comunicazione supportati tra di essi? Quali sfide emergono nella gestione delle risorse computazionali e di caching limitate a livello Edge?

Risposta: L'architettura MEC è rappresentata da tre diversi layer ognuno rappresentate funzioni e risorse differenti. Il primo strato è lo user layer che raggruppa tutti i dispositivi degli EU che possono commissionare task ai server edge in modo totale ,parziale e locale. Abbiamo visto tre modi di comunicazione che sono:

- **Heterogeneous Network:** è formata da diverse categorie di base station , esse hanno equipaggiato delle funzioni AI. È una rete densa di piccole celle che permettono di ridurre i requisiti di data rate, con benefici di consumo di potenza, copertura della rete e un'efficienza spettrale.
- **Vehicular Network:** serve nelle smart city per avvertire i veicoli di informazioni riguardo la strada si utilizza la comunicazione VX2(Vehicular to Everything). IEEE 802.11p,LTE. La sfida di questa rete è movimento ad alta velocità e topologia della rete dinamica.
- **Mobile2Mobile (M2M/D2D) Device2Device:** i task vengono inoltrati ai server edge poichè i dispositivi sono vincolati e non permettono elaborazione in locale per poca computazione e poco storage.

Il livello edge comprende i server edge e le tecnologie per la connettività alla rete (BS, RSU, WAP), con risorse relative a calcolo (cicli di CPU) , storage (caching) e connettività (bandwidth).

Il cloud layer ha maggiori risorse di calcolo e memoria a discapito di una maggiore latenza e si occupa di elaborare task complessi non sensibili al delay e gestione dei server MEC.

- **Domanda:** Spiega il concetto di "**Computation Offloading**" nel Mobile Edge Computing.
 - Quali sono i **benefici** (es. prolungamento della durata della batteria dei dispositivi mobili, riduzione della latenza di elaborazione dei task) e le **sfide** associate a questa pratica?
 - Descrivi le diverse **tipologie di offloading** (esecuzione locale, offloading completo, offloading parziale) e i **modelli computazionali** associati a ciascuna.
 - Quali **fattori** (es. requisiti dell'applicazione, qualità del link di comunicazione, capacità delle risorse del server Edge, consumo energetico del dispositivo) influenzano la **decisione di offloading** e come viene gestito il **compromesso tra consumo energetico e ritardo di esecuzione**?

Risposta: Il computational offloading consiste, da parte dell'EU, di commissionare task ad un server MEC. L'offloading permette ai dispositivi mobili di risparmiare batteria, e talvolta tempo, affidando l'elaborazione a server che hanno più capacità di calcolo del EU stesso e nessuna limitazione di batteria, per ottenere questo però è necessario valutare in modo efficace se e a quale server inoltrare la richiesta, questa scelta può essere fatta seguendo criteri e requisiti differenti.

Ci sono tre possibilità per EU di svolgere il task:

- **esecuzione locale:** il dispositivo stesso esegue l'elaborazione.
- **commissione parziale** possibile per quel tipo di task che possono essere suddivisi così da affidare parte del calcolo ad un server,
- **commissione totale:** quando si fa svolgere tutto il task ad un server esterno.

I principali fattori per la decisione sull'offloading sono: batteria del dispositivo, requisiti applicazione, risorse disponibili sul server e qualità link connessione. Per gestire il tradeoff tra consumo energetico e latenza si valuta quanto del task conviene e se conviene commissionare ad un server, inoltre per migliorare la latenza di trasmissione è possibile effettuare prima una compressione dei dati di input.

Per analizzare questi svolgimenti di task si utilizza → Modello di Computazione.

In questo modello si hanno:

- cicli di CPU (f_m)
- il massimo numero di cicli che riflette la capacità di calcolo del dispositivo mobile (f_{max})
- Descrittore del Task da calcolare $D(d,c,T)$ che corrispondono:
 - $d \Rightarrow$ data size del task
 - $c \Rightarrow$ numero di cicli per calcolare un bit (1500 cicli / 32 bit → tot cicli per un solo bit)
 - $T \Rightarrow$ massima latenza per il task

Guardiamo per Local: $T^L = \frac{d \cdot c}{f_m}$ (che se si ha più cicli di CPU → meno tempo delay).

Consumo di Energia per ciclo di CPU: $E^L = \zeta \cdot f_m^2$

Consumo di Energia per tutto il task = $\zeta \cdot f_m^2 \cdot c \cdot d$

La decisione per Offloading è se T^L è maggiore della massima latenza e se l'energia è maggiore della capacità della batteria.

Full Offloading: stessi calcoli ma si ha anche da considerare nel tempo il tempo di trasmissione (d/r^s) e nel consumo di energia $p \cdot d/r^s$. (Single user)(Multi User per ogni i-esimo dispositivo).

Partial Offloading: $t^P = (1 - \lambda) \cdot \frac{c \cdot d}{f_m} + (\lambda \frac{c \cdot d}{F_e})$

Total Task Execution Time: $t^P + \lambda \cdot d/r$

- **Domanda:** Come il Mobile Edge Computing (MEC) può migliorare significativamente le prestazioni della rete e l'esperienza utente, specialmente per le **applicazioni sensibili alla latenza**? Fai riferimento alla **formula di Mathis** e agli **esempi di osservazioni sulla qualità della rete** tra abitazioni per illustrare l'impatto della prossimità fisica sulla latenza logica e sul throughput.

Risposta:

Il Mobile Edge Computing (MEC) può migliorare significativamente le prestazioni della rete e l'esperienza utente, specialmente per le applicazioni sensibili alla latenza, agendo sui fattori chiave che limitano il throughput e la reattività nelle reti attuali. Questo concetto è ben illustrato dalla **Formula di Mathis** e confermato da osservazioni reali sulla qualità della rete. La **Formula di Mathis** è un'equazione fondamentale che descrive il tasso di trasferimento dei dati (throughput, **R**) in una rete IP in relazione alla latenza e alla perdita di pacchetti. La formula è:

$$R \leq MSS / (RTT \cdot \sqrt{Ploss})$$

Dove:

- **R** è il tasso di trasferimento dei dati (throughput).
- **MSS** (Maximum Segment Size) è la dimensione massima del segmento.
- **RTT** (Round Trip Time) è la latenza di rete end-to-end, ovvero il tempo impiegato da un pacchetto per andare dal mittente al destinatario e tornare indietro.
- **Ploss** è la probabilità di perdita dei pacchetti.

Questa formula è **ampiamente accettata e validata** da misurazioni reali e viene utilizzata per prevedere la velocità dei dati delle implementazioni TCP. Una delle implicazioni più cruciali è che **aumentare la larghezza di banda e implementare funzionalità di Quality of Service (QoS) non è sufficiente** a migliorare l'esperienza utente (QoE), in quanto esiste una limitazione intrinseca nelle reti attuali.

Per calcolare la **massima perdita di pacchetti tollerabile** per un dato tasso di dati (**R**) e un RTT specifico, la formula può essere invertita come segue:

$$Ploss \leq (MSS / (R \cdot RTT))^2$$

Il Ruolo del Mobile Edge Computing (MEC) nell'ottimizzazione della Formula di Mathis

Il Mobile Edge Computing (MEC) è un **abilitatore potenzialmente dirompente** che può migliorare significativamente la qualità della rete agendo su due fattori chiave della formula di Mathis: la latenza (RTT) e il tasso di errore dei pacchetti (Ploss).

1. Riduzione della Latenza (RTT):

- I sistemi MEC posizionano le risorse di calcolo e storage **molto più vicino agli utenti finali**, spesso a decine o centinaia di metri, ad esempio presso le stazioni base o i gateway wireless.

- Questo si traduce in **distanze di propagazione molto brevi** (inferiori a un chilometro) e trasmissioni attraverso reti mobili edge o connessioni dispositivo-dispositivo, risultando in **latenze tipicamente inferiori a decine di millisecondi**.
- Al contrario, i sistemi Mobile Cloud Computing (MCC) tradizionali si affidano a data center centralizzati a centinaia o migliaia di chilometri di distanza, con latenze che possono superare le centinaia di millisecondi a causa del passaggio attraverso più reti.
- Per le **applicazioni sensibili alla latenza** (come la guida autonoma, i giochi online, la realtà aumentata e virtuale), l'offloading computazionale verso i server MEC **fornisce una soluzione praticabile** per rispettare rigorosi vincoli di ritardo, cosa che l'offloading verso server cloud remoti spesso non può garantire.

2. Miglioramento del Tasso di Perdita dei Pacchetti (Ploss):

- L'adozione di piattaforme e meccanismi di QoS, come l'elaborazione del flusso di traffico all'edge, mira a migliorare la perdita di pacchetti.
- La possibilità di accedere a **informazioni locali sul canale radio** è fondamentale per ottimizzare il flusso di traffico nel link di accesso, che è più suscettibile agli errori.

Esempi di Osservazioni sulla Qualità della Rete tra Abitazioni

Un esercizio basato su **osservazioni reali** della qualità della rete tra tre abitazioni situate nello stesso quartiere (un'area di 1 km²) dimostra chiaramente l'impatto della prossimità fisica sulla latenza logica e sul throughput.

- **Scenario delle Abitazioni:**
 - Le case A e B erano connesse allo stesso ISP (ISP1).
 - La casa C era connessa a un ISP diverso (ISP2).
 - Nonostante la vicinanza fisica, le misurazioni hanno mostrato che la latenza end-to-end e la larghezza di banda potevano essere scarse, specialmente quando erano coinvolti ISP diversi.
- **Osservazioni sulla Latenza e Throughput:**
 - Questo indica che la **prossimità fisica (geografica) non si traduce necessariamente in prossimità logica (latenza end-to-end)**.
 - Il percorso tra le case A e B (stesso ISP) forniva un throughput e una latenza migliori rispetto ai percorsi che coinvolgevano ISP diversi.
 - **Home A & B (stesso ISP):** Throughput di 2.3 Mbps e latenza mediana di 26.4 ms.
 - **Home B & C / Home C & A (ISP diversi):** Throughput di 0.9 Mbps e latenza mediana di 44.9 ms.

- **Implicazioni per il MEC**

- Queste osservazioni evidenziano il **potenziale guadagno in termini di prestazioni** offerto dal MEC.
- Un'applicazione MEC in esecuzione su un host MEC posizionato presso l'ISP2 (quindi più vicino all'utente della casa C) può **migliorare significativamente le prestazioni della rete** per quell'utente, rispetto a una comunicazione tradizionale con un'applicazione remota sul cloud dell'ISP1.

Impatto sulle Applicazioni Sensibili alla Latenza (Esempio Reale)

Un'ulteriore analisi basata sulla **funzione di distribuzione cumulativa (CDF)** ha confrontato i tempi di risposta per carichi di lavoro (fluid graphics, face recognition, augmented reality) tra un client utente (UE) alla casa C e un server applicativo situato o nella casa C (vicino, tramite MEC) o nella casa A (distante, tramite cloud).

- **Risultati della CDF:** In tutti i casi, un'applicazione MEC vicina ha prodotto **tempi di risposta significativamente inferiori**.
- **Esempio di Riconoscimento Facciale:**
 - Tempo di risposta mediano con **app MEC vicina: 104 ms**.
 - Tempo di risposta mediano con **app server cloud distante: 882 ms**.
 - Questa enorme differenza di prestazioni è dovuta **esclusivamente alle diverse condizioni di rete**, sottolineando l'importanza del Mobile Edge Computing per le applicazioni che richiedono bassa latenza. Infatti, i tipici ritardi di rete di trasporto e backhauling possono essere misurati in decine o centinaia di millisecondi, rendendo l'offloading su server remoti spesso impraticabile per vincoli di ritardo stringenti.

In sintesi, il MEC offre un vantaggio cruciale per le applicazioni sensibili alla latenza, garantendo tempi di risposta rapidi e un'esperienza utente migliorata grazie alla riduzione della distanza di calcolo e trasmissione, come previsto dalla Formula di Mathis e dimostrato da dati reali.

- **Domanda:** Qual è il ruolo dell'**ETSI MEC (Multi-access Edge Computing)** come standard internazionale leader nel campo dell'edge computing?
 - Come si relaziona con il **3GPP** e quali sono le principali fasi di standardizzazione del MEC (es. Phase 1: Mobile Edge Computing, Phase 2: Multi-access Edge Computing)?
 - Quali sono i principali **scenari di servizio** (es. Intelligent Video Acceleration, Video Stream Analysis, Augmented Reality, Connected Vehicles) e le **API standardizzate** dall'ETSI ISG MEC (es. RNI API, Location API, UE Identity API, V2X API) per facilitare l'integrazione e lo

sviluppo di applicazioni e per migliorare la QoE?

Risposta: L'ETSI MEC ha definito la sua standardizzazione attraverso fasi, ognuna con un focus specifico:

- Phase 1: Mobile Edge Computing:
 - Focus: Questa fase si è concentrata esclusivamente sulle reti cellulari.
 - Benefici: L'obiettivo era ridurre i colli di bottiglia del traffico e migliorare le prestazioni della rete.
 - Sviluppi chiave: È stata introdotta l'API RNI (Radio Network Information). Sono stati pubblicati documenti fondamentali su terminologia, requisiti tecnici, framework e architettura di riferimento, oltre a principi generali per le API di servizio MEC e la gestione MEC.
- Phase 2: Multi-access Edge Computing:
 - Focus: In questa fase, il termine è stato rinominato "Multi-access Edge Computing" per estendere il suo ambito a tecnologie di accesso eterogenee, inclusi LTE, 5G, Wi-Fi e accesso fisso.
 - Coinvolgimento: Ha incluso attori del settore IT e fornitori di servizi cloud.
 - Sviluppi chiave: Sono state introdotte nuove API come WLAN API, Fixed API e V2X API, oltre a continuare e aggiornare le specifiche della Fase 1 (versioni 2.x.x dei documenti). È stata studiata la mobilità end-to-end (MEC Application Mobility Service API) e il supporto MEC per i casi d'uso V2X12.
- **Domanda:** Come è formata l'architettura ETSI MEC server?

Risposta: L'ETSI MEC è lo standard utilizzato nei server MEC per l'Edge Computing ed è armonizzato con il 3GPP per l'implementazione del 5G. La sua architettura è definita da un framework specifico con due parti importanti dove la prima Host Level Management ha le seguenti componenti:

- MEC Host: che contiene un MEC platform e un'infrastruttura di virtualizzazione. Fornisce calcolo di risorse, storage e rete. Si occupa di instradare traffico da e verso le applicazioni. È agnostico rispetto all'accesso nel senso che funziona con diverse tecnologie di rete.
- MEC Platform: ospita servizi MEC e applicazioni MEC, espone i servizi MEC tramite API. Permette di scoprire, consumare, offrire i servizi tramite Application Enablement Framework. Può succedere che dei servizi MEC abbiano bisogno di altri servizi MEC.
- MEC App: sono le applicazioni del software che vengono applicate sulle infrastrutture virtualizzate, sono progettate per essere sensibili alla latenza e computazionalmente intense. Questi servizi offerti possono essere consumati da altri servizi MEC.
(Tipologie di App sono Client - Device)
- VIM: Virtualized Infrastructure Management simile a quello delle NFVI, alloca per ogni servizio MEC un VM o container.
Dopo di che si ha il System Level Management dove ha presente il Multiaccess Edge

Orchestrator che ha una visione complessiva del sistema MEC, gestisce i life cycle dei servizi MEC e selezione i MEC host in base ai requisiti del servizio. Basato su frameworks open source. È responsabile della gestione delle risorse, selezionando gli host MEC appropriati in base a vincoli come latenza, risorse e servizi disponibili.