

Internet of Threads

Renzo Davoli*

29 Giugno 2011

Cosa collega la rete? Sicuramente la ricerca in questi ultimi anni ha messo in discussione quali siano i *nodi* di una rete. La rete Internet e i protocolli TCP/IP sono nati attorno all'idea che le entità da collegare fossero i computer, e che le entità indirizzabili fossero le singole interfacce di rete.

Con l'evoluzione delle reti e, soprattutto, dei servizi di Internet è venuta meno la centralità del singolo computer ed è apparso sempre più innaturale l'indirizzamento delle interfacce come nodi della comunicazione.

- Nei sistemi ad alta affidabilità [4] vengono assegnati alle interfacce numerosi indirizzi IP suddividendo i servizi fra di essi, per poter far migrare i servizi in caso di malfunzionamento di un nodo riassegnando gli indirizzi ad altri elaboratori.
- I sistemi di virtualizzazione quali macchine virtuali [8] o container [6] creano interfacce virtuali alle quali assegnare indirizzi.
- Nei sistemi operativi multiuser risulta molto complesso assegnare indirizzi di rete, QoS differenti a specifici utenti e servizi perché occorre applicare filtri [5] al traffico smistato dall'unico stack del sistema. Spesso la scelta più lineare è quella di attivare macchine virtuali o container che generino interfacce virtuali.
- L'utente rimane vincolato dalle scelte di networking dell'amministratore di sistema: senza riconfigurare il sistema appare arduo poter, per esempio, eseguire più browser che funzionino con indirizzi IP diversi di altrettante VPN distinte. Anche in questo caso l'utente può attivare macchine virtuali (QEMU/KVM/User-Mode Linux/VirtualBox) e usare reti virtuali (VDE).

Questo articolo presenta una proposta alternativa: l'introduzione della possibilità di avere come nodi di rete singoli processi o gruppi di processi. Il nome scelto per questo concetto è *Internet of Threads*¹ che indica bene il cambiamento di prospettiva come ulteriore evoluzione dei concetti di *Internet of Things* [3] e di *Internet of Services* [7].

*Dipartimento di Scienze dell'Informazione, Università di Bologna.

¹ringrazio Pietro Galliani della University of Amsterdam per avermi suggerito un termine così calzante.

La proposta è già corredata di un insieme di strumenti software che costituiscono un proof-of-concept operativo di questo cambiamento di prospettiva sulla rete.

Il laboratorio internazionale di ricerca sulla virtualità *VirtualSquare* [1, 2] ha realizzato molteplici strumenti:

- VDE: la rete Ethernet virtuale e distribuita. Consente di connettere in una rete locale macchine virtuali (e altre entità virtuali) degli utenti, in esecuzione sullo stesso elaboratore o su sistemi fisici diversi anche se geograficamente distribuiti.
- LWIPv6: uno stack ibrido LWIPv6, IPv6 retrocompatibile con IPv4, interamente implementato come una libreria modulare. Ogni applicazione che utilizza LWIPv6 accede a una rete TCP-IP con propri indirizzi e proprio routing. LWIPv6 supporta autoconfigurazione, DHCP (server e client), NAT e slirp (anche IPv6).
- *View-OS/umnet/umnetlwip6*: View-OS è un progetto di macchine virtuali parziali che consente di virtualizzare alcune funzionalità del sistema host. (per esempio, *umfuse* consente mount virtuali di parti del file system, *umdev* crea device virtuali). *Umnet* è il modulo di virtualizzazione della rete, e il sottomodulo *umnetlwip6* consente di usare la libreria *lwip6* per usare reti virtuali. Con View-OS è possibile usare le applicazioni esistenti in contesti di rete virtuale, per esempio essa consente di attivare un browser collegato ad una VPN mentre il resto del sistema usa la rete locale.

Questo cambiamento di prospettiva consente una estrema flessibilità nella migrazione dei servizi. Una rete ethernet virtuale forma un cloud naturale per le applicazioni direttamente connesse come *nodi* di rete: una volta collegati alla rete virtuale l'applicazione che risponde ad un determinato indirizzo può essere ovunque nella rete virtuale (e a maggior ragione in quella fisica sottostante).

Con l'*Internet of Threads*, si perde l'idea dello stack unico per sistema (o per container) e lo stack diventa invece una libreria di implementazione di protocolli ai vari livelli del modello. Considerazioni di efficienza e di affidabilità del sistema possono guidare la scelta verso librerie del kernel o come librerie utente, entrambe le scelte sono possibili. Il cambiamento fondamentale sta nel fatto che ogni utente può attivare molteplici stack di rete per le proprie applicazioni (processi o thread) e decidere a quali reti collegare ognuna di esse, quindi quali indirizzi e quale routing associare ad ogni applicazione.

In quest'ottica deve essere anche possibile creare applicazioni che possano operare da gateway fra stack diversi. Questo ha messo in luce una limitazione strutturale dell'interfaccia di programmazione *Berkeley Sockets*, che implicitamente assume che ci sia un solo stack di rete nel sistema. E' stato necessario estendere la API per consentire di poter utilizzare più stack contemporaneamente. Questa estensione tra l'altro consente:

- di poter operare contemporaneamente con stack con caratteristiche diverse: e.g. politiche di ritrasmissione, ampiezza delle finestre TCP etc;

- la compresenza di stack di produzione e sperimentali. Oggi è comune avere una sola implementazione dello stack anche quando vengono definite diverse reti virtuali, come per esempio tramite i containers [6]. Con una *Internet of Threads* è possibile far convivere al contempo uno stack affidabile per la produzione e il controllo degli esperimenti mentre si opera anche con uno stack sperimentale per valutarne per esempio correttezza, performance ed affidabilità. Senza *Internet of Threads* l'uso di uno stack sperimentale potrebbe rendere inaccessibile un nodo remoto in caso di errore.

Gli esempi trattati mostrano che il concetto di *Internet of Threads* apre nuove possibilità per creare nuove applicazioni o per semplificare l'implementazione di servizi esistenti. La nuova definizione di nodo di rete che ricomprende anche singoli programmi (processi o thread) consente di ridisegnare, ampliandolo, il concetto stesso di networking a ogni livello: al livello del supporto da parte di sistemi operativi, dell'interfaccia di programmazione, di routing, sicurezza, QoS, etc.

Riferimenti bibliografici

- [1] R. Davoli and M. Goldweber, editors. *Virtual Square: Users, Programmers & Developers Guide*. Lulu books, 2011.
- [2] R. Davoli, M. Goldweber, et al. Virtualsquare international lab on virtuality wiki. <http://wiki.virtualsquare.org/>.
- [3] N. Gershenfeld, R. Krikorian, and D. Cohen. The internet of things the principles that run the internet are now creating a new kind of network of everyday devices, an "internet-0.". *Scientific American*, 291(4):76–81, 2004.
- [4] E. Marcus and H. Stern. *Blueprints for High Availability: Designing Resilient Distributed Systems*. John Wiley & Sons, 2003.
- [5] P. McHardy et al. Netfilter: firewalling, nat and packet mangling for linux. <http://www.netfilter.org/>.
- [6] P. B. Menage. Adding generic process containers to the linux kernel. In *Proc. of the Ottawa Linux Symposium*, 2007.
- [7] C. Schroth and T. Janner. Web 2.0 and soa: Converging concepts enabling the internet of services. *IT Professional*, 9(3):36–41, 2007.
- [8] J. E. Smith and R. Nair. The architecture of virtual machines. *IEEE Computer*, 38(5):32–38, May 2005.