

Sisteme de operare avansate

Connection Handoff Policies for TCP Offload Network Interfaces

Hyong-youb Kim and Scott Rixner

Rice University

Houston, TX 77005

{hykim, rixner}@rice.edu

Costul procesării TCP

- Procesarea TCP e dominată de costul accesării memoriei
 - atunci când se accesează date sau structuri de date asociate conexiunii (headere, structuri interne SO)
- Accesul la date
 - copierea pachetului în user-space; soluție: zero-copying I/O
 - Checksumming; soluție: hardware offloading

Accesul la structurile asociate conexiunii TCP

- Chiar dacă aceste structuri sunt relativ mici (de ordinul KB) la un număr mare de conexiuni vom ieși din cache
- Nu există o soluție care să remedieze această problemă
- Folosirea unui TCP offloader-ului va adresa această problemă prin plasarea structurilor într-o memorie rapidă

TCP offloading

- Folosirea plăcii de rețea pentru procesa direct TCP
- Poate reduce încărcarea procesorului
 - anumite pachete nu mai sunt procesate de host ci de NIC
- Poate reduce încărcarea magistralei
 - anumite pachete nu mai ajung în host (ACK, FIN, etc.)
- Poate crește throughput-ul

Full TCP offloading

- Toată procesare TCP se face de către NIC
- Abordarea nu e scalabilă
 - puterea de procesare (CPU, memorie) va limita performanța (i.e. performanța obținută nu se “adaugă” la performanța hostului)
 - e nevoie de un protocol de comunicare între host și NIC pentru a partaja porturile alocate, rutele și adresele IP

TCP offloading NIC

- Numărul de conexiuni pe care le poate suporta un astfel de device e relativ mic
- Resursele NIC-ului sunt limitate
 - procesor mai slab ca cel al sistemului
 - memorie de capacitate mai mica
- Viteza memoriei mai bună decât cea a sistemului

Concluzie

Placa de rețea poate fi folosită pentru a degreva procesorul de anumite operații TCP, dar acest lucru trebuie făcut cu măsură, în limita resurselor disponibile în cadrul NIC-ului.

Abordare

- Connection handoff
 - o parte din conexiunile active sunt pasate NIC-ului
 - sistemul de operare va trimite cererile de send / recv direct către NIC
 - NIC-ul va face procesarea TCP efectivă
 - eventual SO poate reclama o conexiune înapoi, spre a fi tratată în stiva software

Abordare [2]

- Spre deosebire de alte abordări SO este “în control”
 - decide ce conexiuni să fie offloaded
 - poate reduce numărul de conexiuni de pe NIC
 - SO ia în continuare deciziile de rutare, selectare adresă sursă, alocare de porturi și filtrare

Ce aduce nou paper-ul

- Studierea problematicii folosirii unui TCP offloading NIC
 - offloader-ul nu trebuie să influențeze negativ conexiunile procesate de host
 - SO un trebuie să supraîncarce NIC-ul
 - Pentru că NIC-ul nu poate procesa decât un număr limitat de conexiuni SO trebuie să aleagă pentru offload acele conexiuni care pot fi tratate eficient de NIC
- Politici de utilizare a NIC-ului într-un mod eficient

Sunt aceste politici necesare?

- Da. Simularea a patru scenarii de încărcare web arată că procesarea TCP făcută de NIC poate degrada performanța
 - cauza: crește latentă introdusă de NIC de la 10us la 1ms
 - în trei din cele patru scenarii performanța s-a degradat cu 14-30%

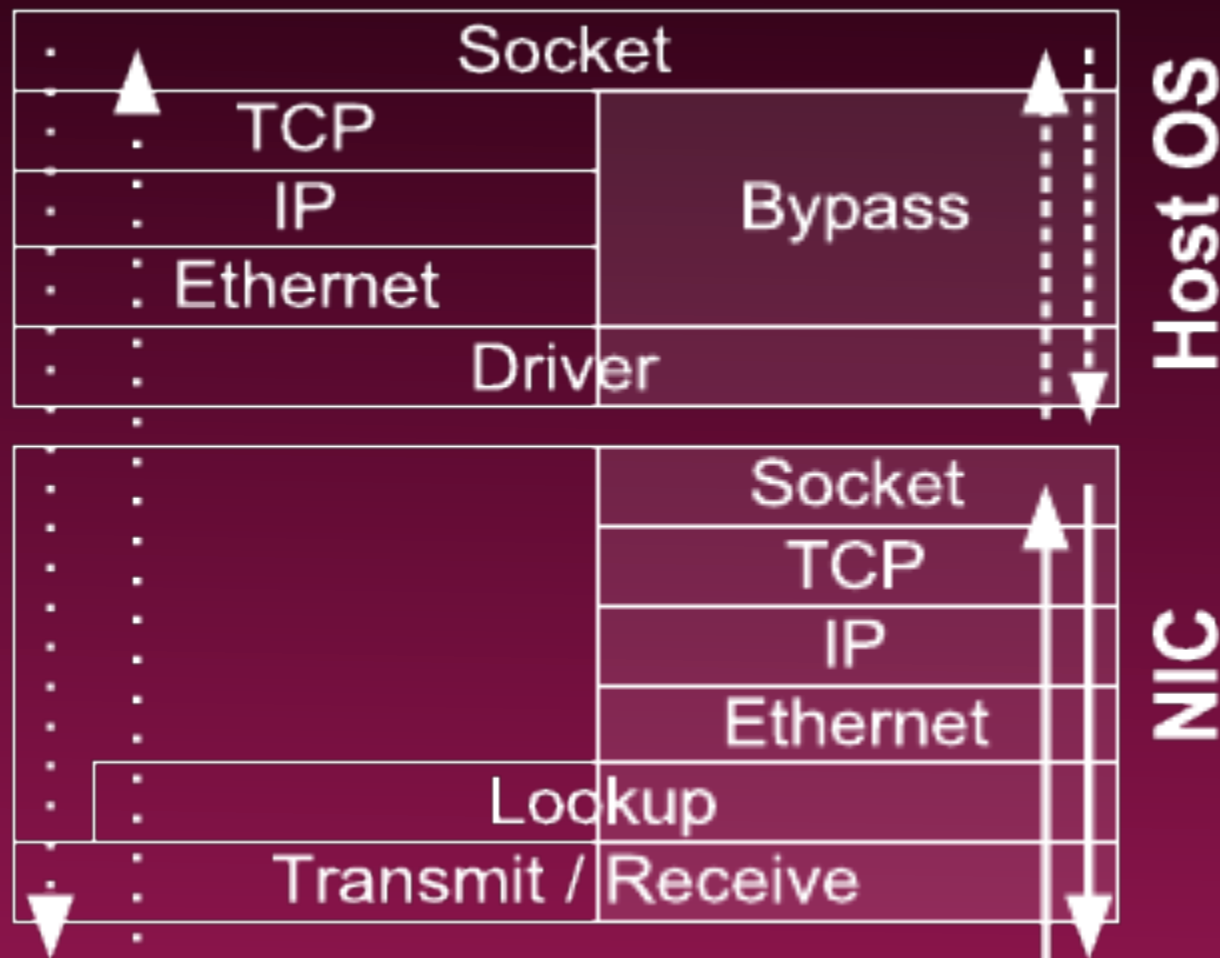
Prioritatea pachete host

- Dacă NIC-ul dă prioritate pachetelor ce trebuie trimise host-ului, se obține o creștere de performanță de 24% când NIC-ul un este suprasolicitat
- Dacă NIC-ul este suprasolicitat (prea multe conexiuni), se observă o scădere de performanță de 44%

Limitarea conexiunilor

- Dacă limităm și numărul de conexiuni obținem o creștere a performanței de până la 31%
- Care e politica de hand-off a conexiunilor?
 - FCFS: merge dacă numărul de conexiuni concurente suportate de NIC e mare
 - Altfel e preferat hand-off-ul pentru conexiunile cu durată mare de viață

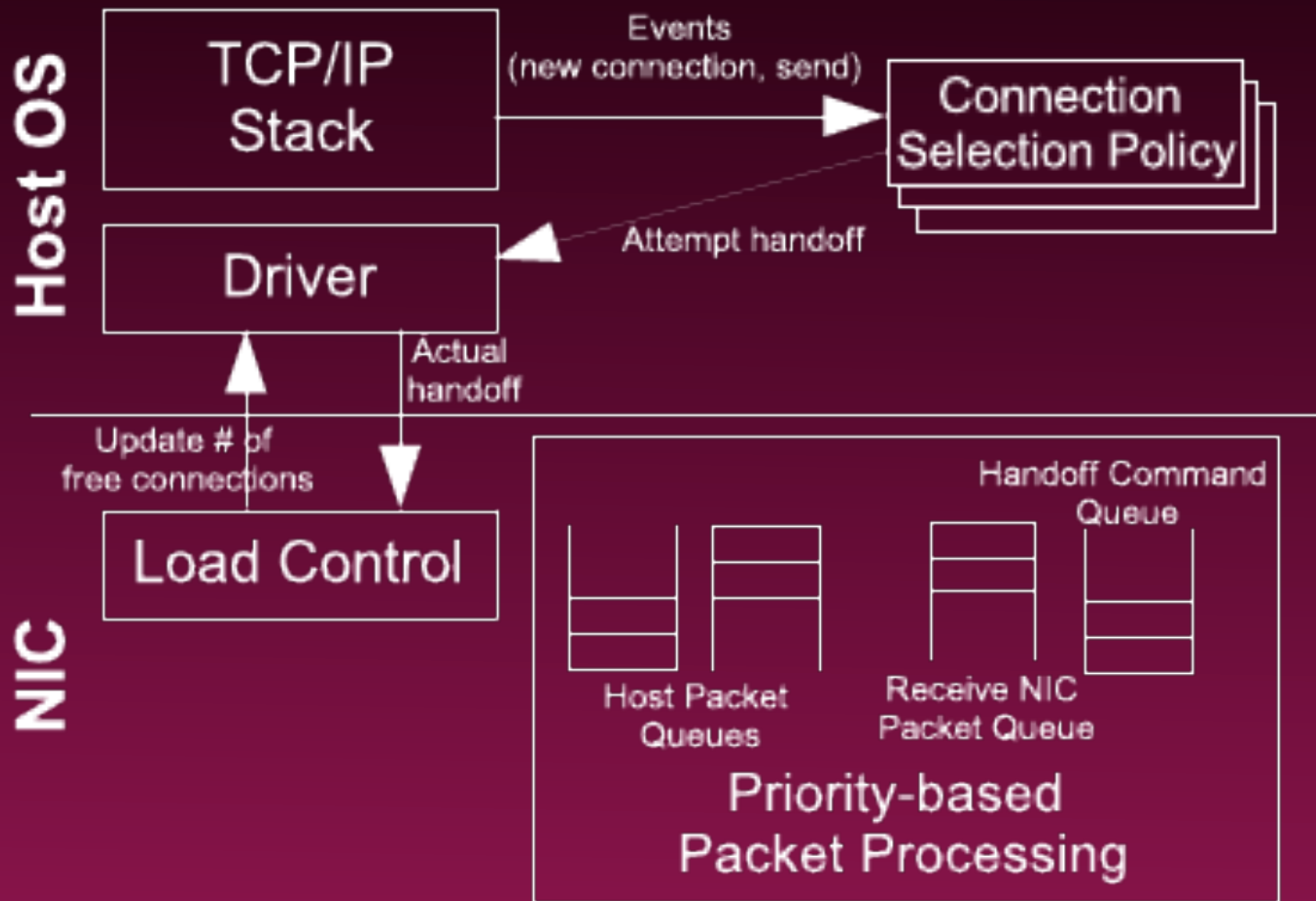
Connection handoff



Connection handoff

- Lookup layer
 - decide dacă pachetele trebuie sa ajungă la host sau să rămână în NIC
 - hash based
- Connection handoff API
 - protocolul prin care NIC-ul și driverul
 - mută o conexiune de la host la NIC și invers
 - send/receive payload (pentru eficiență datele sunt ținute în memoria host-ului)

Arhitectura sistemului



Procesare pachetelor

- Relativ la procesarea pachetelor NIC-ul are două roluri:
 - procesarea pachetelor pentru conexiunile offloaded
 - forwardarea pachetelor către host pentru conexiunile ce nu sunt offloaded
- Probleme: procesarea pachetelor în NIC poate întârzia transmiterea pachetelor către host și astfel afecta performanța

Exemplu de degradare a performanțelor

Offloaded Connections	Packet Priority	Host		NIC	Requests/s
		Packet Delay (usec)		Idle (%)	
		Send	Receive	Idle (%)	
0	<i>no handoff</i>	2	2	62	23031
256	FCFS	3	3	49	23935
1024	FCFS	679	301	3	16121
1024	Host first	10	6	5	26663

Exemplu de degradare a performanțelor

[2]

- FCFS – pachetele sunt servite în ordinea în care vin, indiferent dacă trebuie tratate în NIC sau în host
- Host first – prioritate pentru pachetele host
 - două cozi: una cu pachetele hostului (send & receive)
 - una cu pachetele NIC-ului
 - se procesează întâi toate pachetele din coada host

Load control

- Controlul dinamic al numărului de conexiuni offloaded
- Previne supraîncărcarea NIC-ului
- numărul de pachete din coada de recv (pentru pachetele NIC-ului) este folosit pentru a aproxima încărcarea NIC-ului

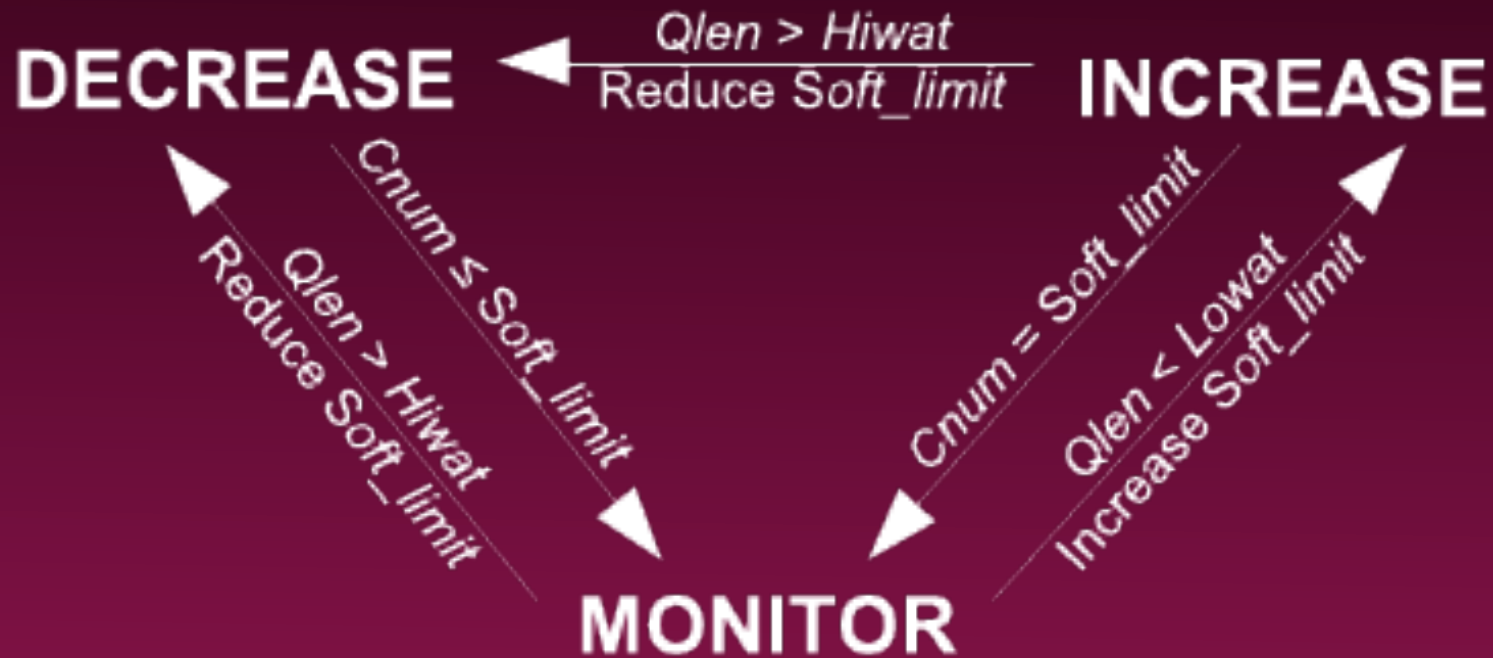
Parametrii de control

- **hard limit = limitează numărul maxim de conexiuni**
 - este setat la inițializarea firmware-ului
 - dependent de câtă memorie avem pe NIC
- **soft limit = numărul de conexiuni maxime curent**
 - modificat în funcție de încărcarea curentă
- **Qlen = numărul de pachete ale NIC-ului din coada de receive**

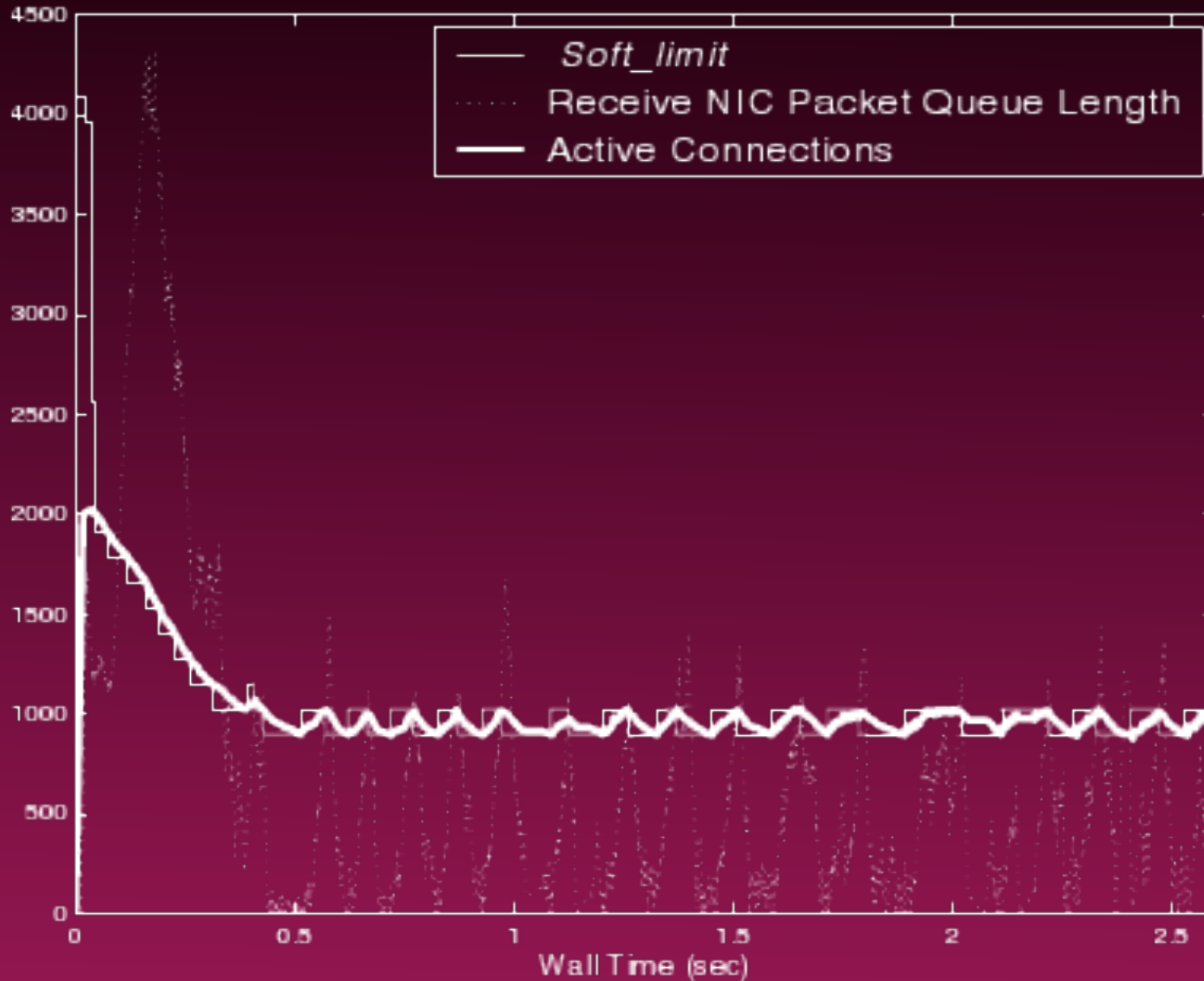
Parametrii de control [2]

- hiwat – high watermark pentru coada de pachete
 - când $qlen > \text{high watermark}$ trebuie să reducem numărul de conexiuni
- lowat – low watermark pentru coada de pachete
 - când $qlen \leq \text{low watermark}$ putem să creștem numărul de conexiuni

Control load state machine



Exemplu



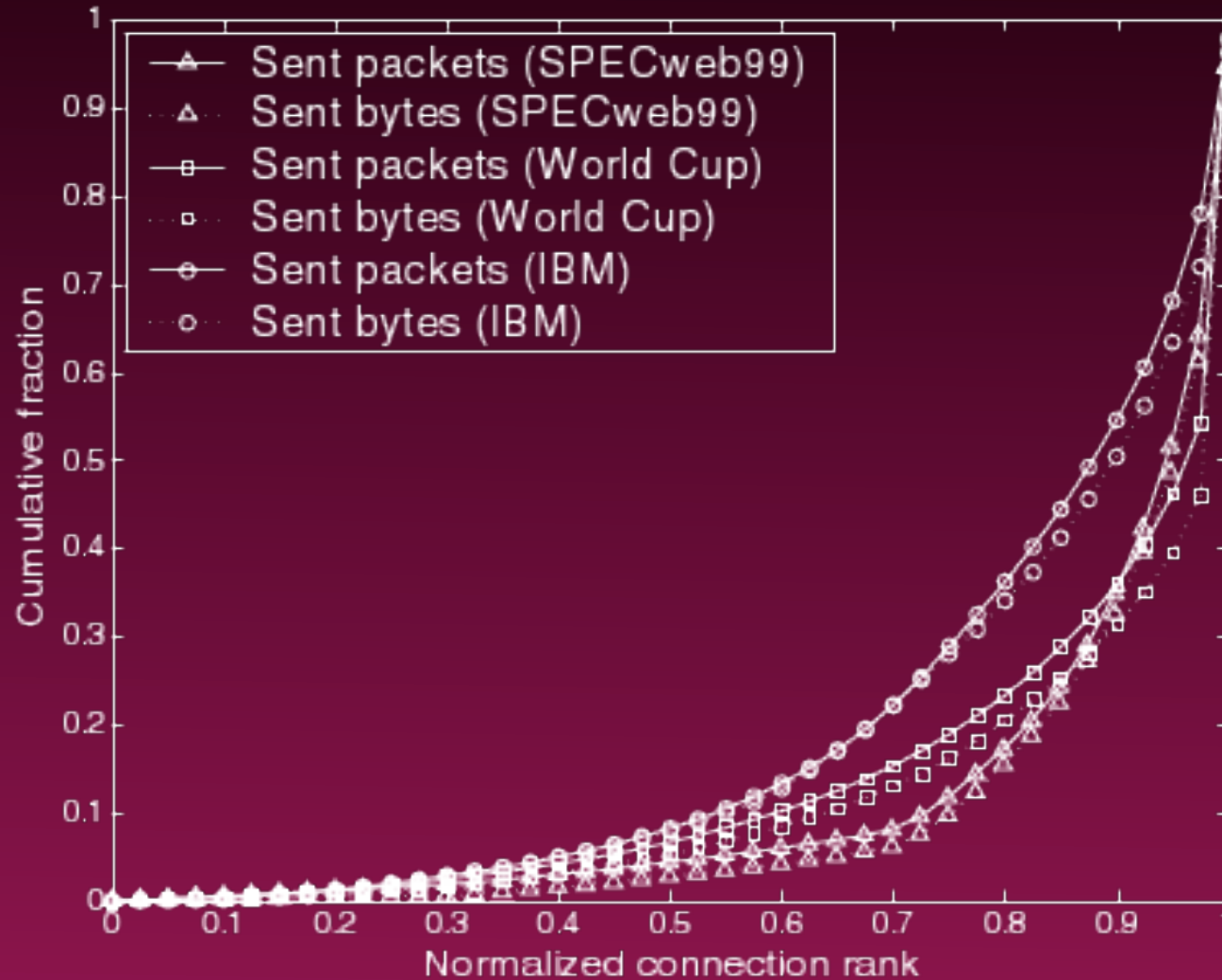
Selectarea conexiunilor

- Decide ce conexiune să fie offloaded atunci când încărcarea este suficient de mică
- SO încearcă sa facă handoff
 - la stabilirea conexiunii sau la send
 - dacă handoff-ul eșuează nu se mai încearcă pentru acea conexiune mai târziu

Selectarea conexiunilor [2]

- Costul de hand off trebuie să fie balansat de beneficiile offloading-ului
- Caracteristici ale conexiunilor ce influențează acest raport
 - durata de viață (măsurată în numărul de pachete trimise per conexiune)
 - rata de pachete
- SO trebuie să estimeze durata de viață
 - se folosește de o distribuție ce descrie comportarea traficului web

Distribuția duratei de viață



Discuție

- 50% din conexiuni trimit 10% din trafic
- celelalte 50% trimit 90% din trafic
- Ne putem folosi de această observație pentru a determina conexiunile cu o durată de viață mare pe baza unui valori limită (threshold)

Evaluare

- SO: FreeBSD 4.7
- Hardware: simulat cu Simics
 - Simics = un simulator de sistem complet
 - procesor = x86
 - a fost extins...
 - ... cu un emulator de memorie
 - cache, memory controller, simulator DRAM
 - ... și un NIC

Simulare NIC

- Procesor MIPS (400, 600, 800 MIPS)
- 32MB memorie
- Controller PCI
 - DMA
- Ethernet controller
 - MAC, CRC
- Timer

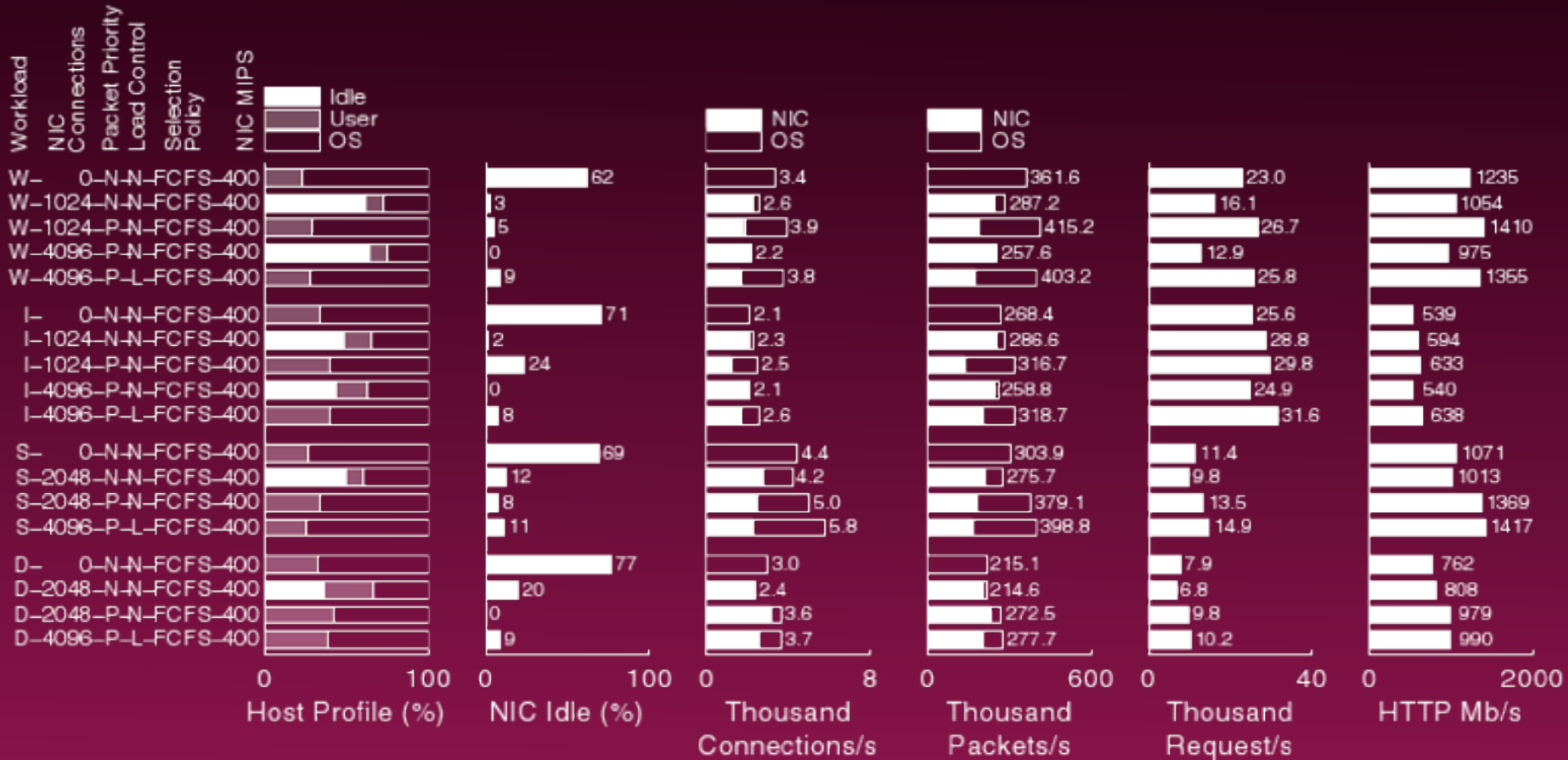
Detalii

	Configuration
CPU	Functional, single-issue, 2GHz x86 processor Instantaneous instruction fetch
L1 cache	64KB data cache Line size: 64B, associativity: 2-way Hit latency: 1 cycle
L2 cache	1MB data cache Line size: 64B, associativity: 16-way Hit latency: 15 cycles Prefetch: next-line on a miss
DRAM	DDR333 SDRAM of size 2GB Access latency: 195 cycles
NIC	Functional, single-issue processor Varied instruction rates for experiments Varied maximum number of connections 10Gb/s wire

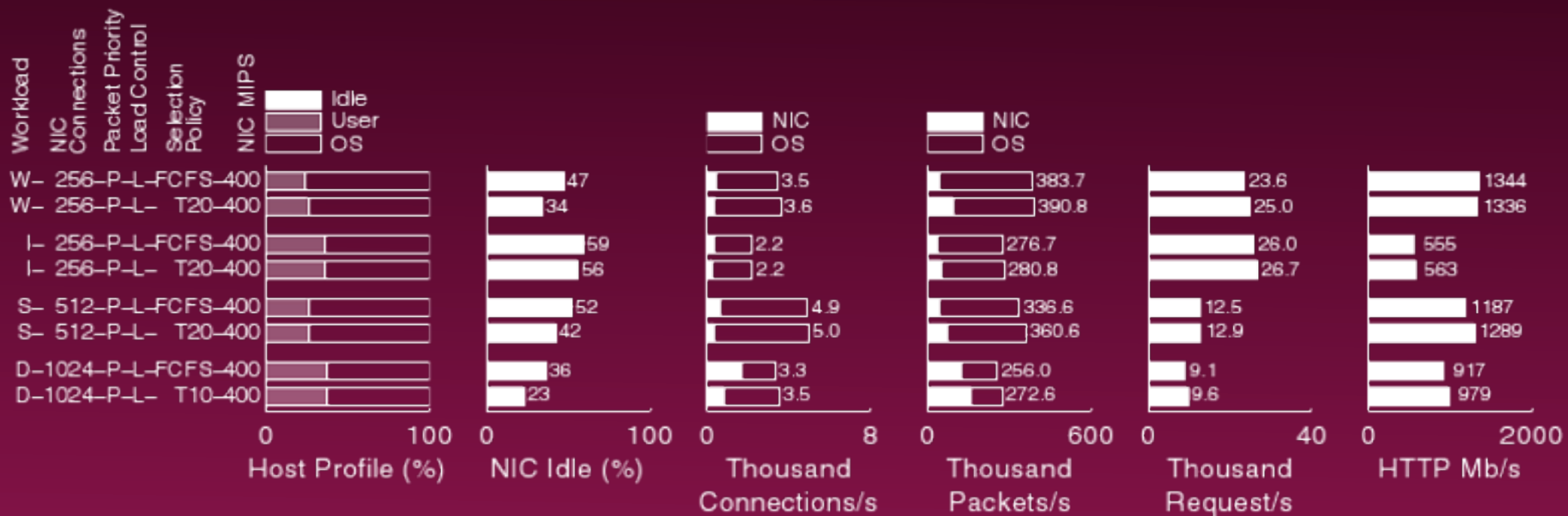
Rezultate [1]

Workload	<p>Web server workload</p> <p>W: World Cup, 2048 clients</p> <p>I: IBM, 2048 clients</p> <p>D: SPECweb99, 2048 clients</p> <p>S: SPECweb99-static, 4096 clients</p>
NIC Connections	<p>Maximum number of connections on the NIC</p> <p>0 means handoff is not used.</p>
Packet Priority	<p><i>host first</i> priority-based packet processing on the NIC</p> <p>P: Used</p> <p>N: Not used</p>
Load Control	<p>Load control mechanism on the NIC</p> <p>L: Used</p> <p>N: Not used</p>
Selection Policy	<p>Connection selection policy used by the operating system</p> <p>FCFS: First-come, first-served</p> <p>T_n: Threshold with value <i>n</i></p>
NIC MIPS	<p>Instruction rate of the NIC in million instructions per second</p>

Rezultate [2]



Rezultate [3]



Rezultate [4]

