

***Masurarea capacitatii in  
retele de mare viteza.  
Integrarea in MonALISA***

**Student: Zota Stefan-Cosmin**

**Coordonator stintific:  
prof. dr. Ing. Valentin Cristea**

# *Scopul si importanta proiectului*

- Importanta masurarii capacitatii unei cai retea:
  - Detectarea rapida a schimbarilor din retea.
  - Nevoia de o unealta software mult mai convenabila decat un upgrade hardware costisitor.
  - Datele obtinute folosite de nivelul Transport sau alte framework middleware pentru optimizarea politicii de transfer end-to-end.
  - Rate de transfer Gigabyte necesare.
- Scopul proiectului:
  - Dezvoltarea unei unelte robuste pe host pentru masurarea capacitatii end-to-end.

# *Continut prezentare*

- Tehnici existente
  - “Single Packet”
  - “Packet Pair”
- Arhitectura MonALISA
- ADR – masurarea capacitatii end-to-end:
  - “Packet pair dispersion”
  - “Packet train dispersion”
  - “Average dispersion rate”
- ADR – structura generala
- Modulul *cap* MonALISA
- Rezultate
- Concluzii si proiecte viitoare

# *Tehnici existente*

- Tehnici "Single Packet"

- Estimare capacitate pentru fiecare link (forward time)
- Timpul de transmisie ( $t$ ) este determinat de marimea pachetului ( $P$ ), bandwidth legatura ( $b$ ) si o latentă fixată ( $l$ ):

$$t = \frac{P}{b} + l$$

- Tehnici "Packet Pair"

- Estimare capacitate pentru o cale
- Nu este nevoie de determinare a latentei
- Metode statistice pentru filtrarea rezultatelor

- Dezavantaje pentru ambele tehnici:

- Propagarea erorilor ("Single Packet")
- Puternic influențat de cross traffic ("Single Packet")
- Nevoia de confirmări ("Single Packet")
- Nedectare noduri invizibile
- Trafic mare generat ("Single Packet")
- Puternic influențate de queuing ("Packet Pair")

# *Arhitectura MonALISA*

- Aplicatie dinamica pentru o categorie larga de servicii:
  - Jini, UDDI – WSDL/SOAP
    - Descoperire automata a fermelor
    - Detectare si notificare la distanta
    - Inchiriere pentru o unitate inregistrata
- Configurare dinamica a listei de parametrii
- Unelte de monitorizare integrate (SNMP, LSF, Ganglia, Hawkeye)
- MonALISA permite:
  - Afisarea unor valori detaliate pentru fiecare nod
  - Extragerea datelor in timp real
  - Intocmirea unor grafice pentru informatia obtinuta
  - Inregistrare si notificarea clientului

# *ADR – "Packet Pair Dispersion"*

- Observatii preliminare:
  - Distributia Paxson multi-modala a bandwidth-ului
  - Scopul – gasirea CM (Capacity Mode)
  - Capacitatea = capacitatea minima pentru toate linkurile de pe cale ignorand rata de utilizare a linkurilor
  - Linkul cu rata de transmisie minima determina capacitatea ("*narrow link*")
- "Packet Pair Dispersion"
  - Mecanism intern de "self clocking"
  - "Packet Pair dispersion" = intervalul de primire intre pachete back-to-back
  - Calcularea bandwidthului la destinatie (H hops)  $b = L / \Delta H$  si intocmirea distributiei  $\beta$
  - Capacitatea nu este data de punctul de minim pentru  $\beta$
  - Cross-traffic persistent: SCDR (Sub-Capacity Dispersion range) si PNCM (Post-Narrow Capacity Modes)
  - CM si PNCM sunt distincte chiar cu o incarcare puternica

# *ADR – "Packet Train Dispersion"*

- Extindere a "Packet Pair Dispersion"
- Trimiti  $N > 2$  pachete back-to-back de marime  $L$  bandwidth
- Fara cross-traffic  $b = C$  (capacitatea)
- $\Delta(N)$  creste proportional cu  $N$  pana la un punct ADR (Average Dispersion Rate)
- ADR
  - Mod uni-modal de bandwidth
  - Independent de  $N$

$$b(N) = \frac{(N-1)L}{\Delta(N)}$$

# *Metodologia ADR – "Average dispersion rate"*

- ADR este:

$$R = \frac{(N - 1)L}{\Delta_H} = \frac{C_H}{\prod_{i=1}^H (1 + \frac{r_i}{C_{i-1}})}$$

- Etapa I – "packet pair probing"
  - Se trimite un numar mare de perechi de pachete pentru descoperirea distributiei
  - Daca distributia este uni-modala (cai putin incarcate), procesul se termina si capacitatea este modul unic m
- Etapa II – "packet train probing"
  - Se maresta N -> eliminare PNCM din distributie si  $\beta(N)$  devine uni-modala
  - Capacitatea este modul unic minim



# *ADR – structura generala*

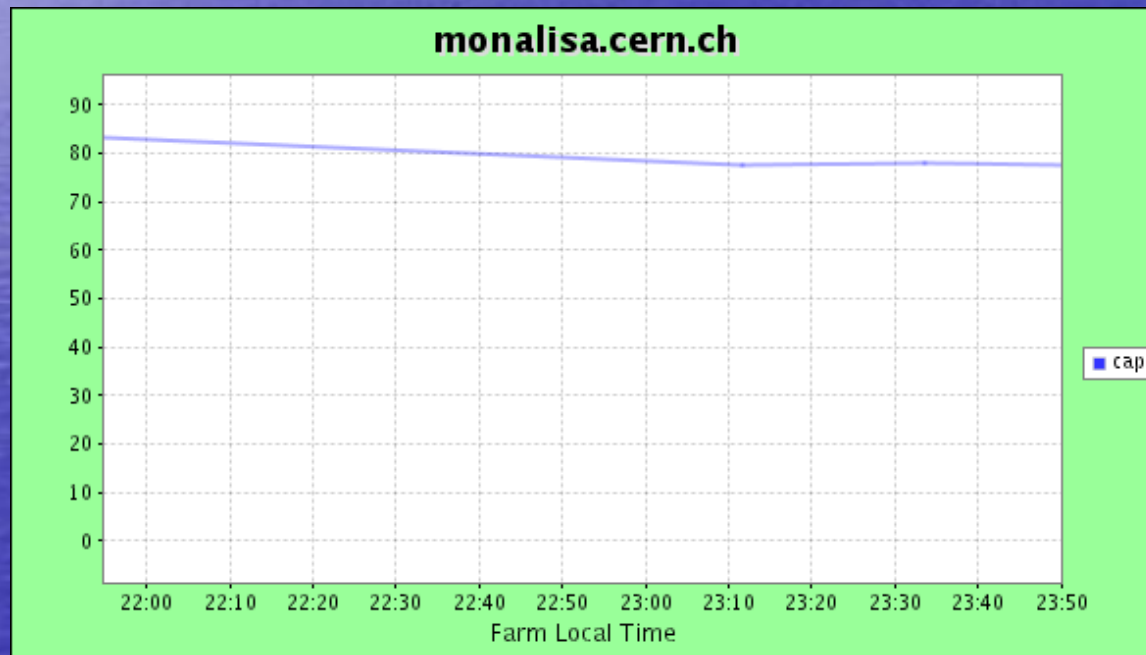
- Patru nivele ale implementarii:
  - Server, client – vizibil utilizatorului, executie algoritm in receiver. Comunicatie Pathrate sender si receiver prin pachete UDP si prin conexiunea de control TCP
  - Management thread-uri – un service thread pool, worker thread si o strategie de load balancing
  - Comunicare nivel inalt – cod Java. ConnectionHandler si ConnectionMethods receiver
  - Comunicare nivel redus – cod C si librariile: libudprcv, libudpsnd, libtime
- Clase externe
  - Clase care contin constante necesare in program (Common, Globals etc.)
  - Clase care contin rezolvarea matematica a unor aproximari statistice (UtilMethods, UtilMethodsRcv)

# *Modulul cap MonALISA*

- Integrare in MonALISA (integrarea in client)
- Package Lia.Monitor.Farm.Pathrate
- Clasa monPathrate din lia.Monitor.modules – configuratie, setari conexiuni si extragere rezultate. Instanta pentru receiver
- Clasa AppPathrate din lia.App.Pathrate – logica server ferma. Start pentru server, instanta pentru sender
- Fisier de configuratie incarcat de la URL

# *Rezultate obtinute*

- Trafic generat 1.56GB, bandwidth ocupat 4MBps, timp executie mediu 400s



Rezultate bune si pentru legaturi Gigabyte si 10MB

# *Concluzii si proiecte viitoare*

- *Robustete* – adaptabila la multe medii
- *Transfer pachete UDP in C* – comunicatia optima in C
- *Structura ierarhica* – usor de extins
- *Independent de platforma* – Java si librariile cod C
- *Interfata prietenoasa* – usor de utilizat si de interpretat
- Usor de configurat (constante), loguri
  
- Unealta utila pentru estimarea capacitatii
  
- Aplicatia poate fi dezvoltata pentru masurarea capacitatii pentru fiecare legatura de pe cale (eventual algoritmul tailgating)

# *Bibliografie*

- [1] O'Reilly – Java NIO
- [2] C. Dovrolis, P. Ramanathanm si D. Moore – Packet Dispersion Techniques and a Capacity Estimation Methodology
- [3] C. Dovrolis, P. Ramanathanm si D. Moore – What do packet dispersion techniques measure
- [4] J. C. Bolot – Characterizing End-to-End Packet Delay and Loss in the Internet
- [5] V. Jacobson – Congestion Avoidance and Control
- [6] M. Jain si C. Dovrolis – End-to-End Available Bandwidth: Measurement methodology, Dynamics and Relation with TCP Throughput
- [7] A. Pasztor si D. Veitch – The Packet Size Dependence of Packet Pair Like Methods
- [8] V. Paxson – Measurement and Analysis of End-to-End Internet Dynamics

Intrebari?