



**Piotr LECH, Krzysztof OKARMA**

## **BADANIA TRANSFERU DANYCH W APLIKACJACH Z OBIEKTAMI ROZPROSZONYMI - JAVA RMI - W ŚRODOWISKU MASZYN WIRTUALNYCH**

### *Streszczenie*

*W artykule zbadano wpływ aplikacji rozproszonej wykonanej w technologii Java RMI na transfer danych w sieci komputerowej. Przeprowadzono badania określające transfer ładunku poprzez sieć dla technologii RMI bez wprowadzanych ograniczeń na przepustowość sieci w odniesieniu do popularnych protokołów HTTPS i TCP. Badania przeprowadzono w wirtualnym laboratorium zrealizowanym w oparciu o technologię maszyn wirtualnych.*

### **WSTĘP**

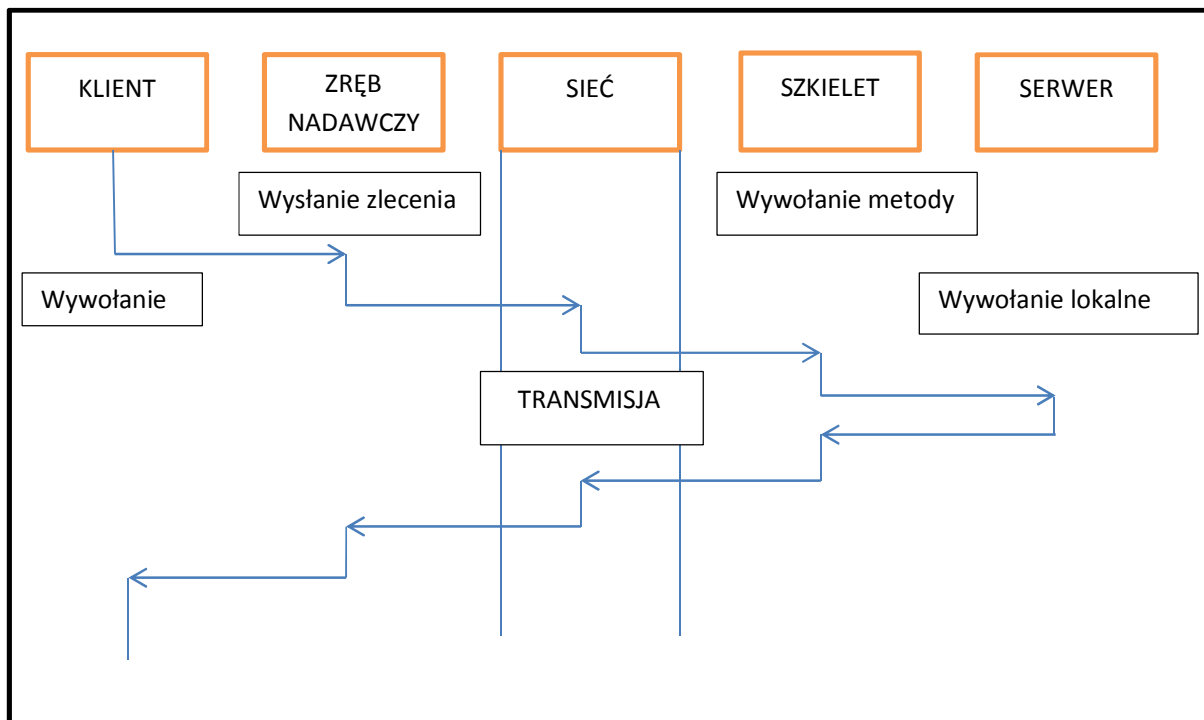
Gwałtowny rozwój infrastruktury sieci teleinformatycznych w szczególności sieci Internet wpływa na rozwój systemów rozproszonych i coraz popularniejszego „przetwarzania w chmurze” (ang. Cloud Computing). Oprócz rozproszonych aplikacji biznesowych, czy też hurtowni danych, realizowane są zadania związane z rejestracją danych pochodzących z urządzeń pomiarowych [1]. Opracowano wiele standardów i technologii wspierających budowę aplikacji rozproszonych, wśród których szczególnie wartą uwagi jest, związana z językiem programowania Java, technologia RMI (Remote Method Invocation). W przypadku sieci telekomunikacyjnych dedykowanych konkretnemu zadaniu z zarezerwowaną i określoną przepustowością sieci łatwo jest zadbać o budowę stabilnej aplikacji. Rozwój sprzętu komputerowego i technik wirtualizacji umożliwiającej jednoczesne uruchomienie wielu systemów operacyjnych ułatwia symulację zagadnień sieciowych w oderwaniu od rzeczywistej infrastruktury.

### **1. BADANIE WIELKOŚCI PRZENOSZONEGO ŁADUNKU**

Sieci teleinformatyczne umożliwiają transfer danych z określoną przepustowością (ang. bitrate), co wprowadza ograniczenie dla aplikacji sieciowych związane z limitem ilości transmitowanych danych. Znaczny komfort podczas budowy aplikacji rozproszonych zapewnia dedykowana sieć z zagwarantowanym transferem danych, komplikacje występują natomiast w przypadku użycia sieci Internet - chwilowa przepustowość uwarunkowana jest wieloma czynnikami, a ponadto jest zmienna w czasie. Wielkość ładunku (ang. payload) przenoszonego przez sieć może mieć krytyczny wpływ na przetwarzanie danych i stabilność aplikacji. Proponowane badania pozwalają poznać ograniczenia wnoszone przez różne

technologie na ilość danych możliwych do transferu oraz możliwości alternatywnych metod transferu danych możliwych do zrealizowania w tych samych warunkach.

Do badań wybrano wspólną platformę programistyczną bazującą na języku programowania Java. Umożliwiło to uniknięcie dodatkowych rozbieżności w badaniach związanych z szybkością działania aplikacji wynikających z implementacji realizowanych wykorzystaniem różnych języków programowania. Działanie aplikacji testowej zostało porównane z klasycznymi aplikacjami w relacji klient-serwer w oparciu o implementacje protokołów HTTPS i TCP.



Rys. 1. Obszar badań dotyczący transmisji danych w sieci

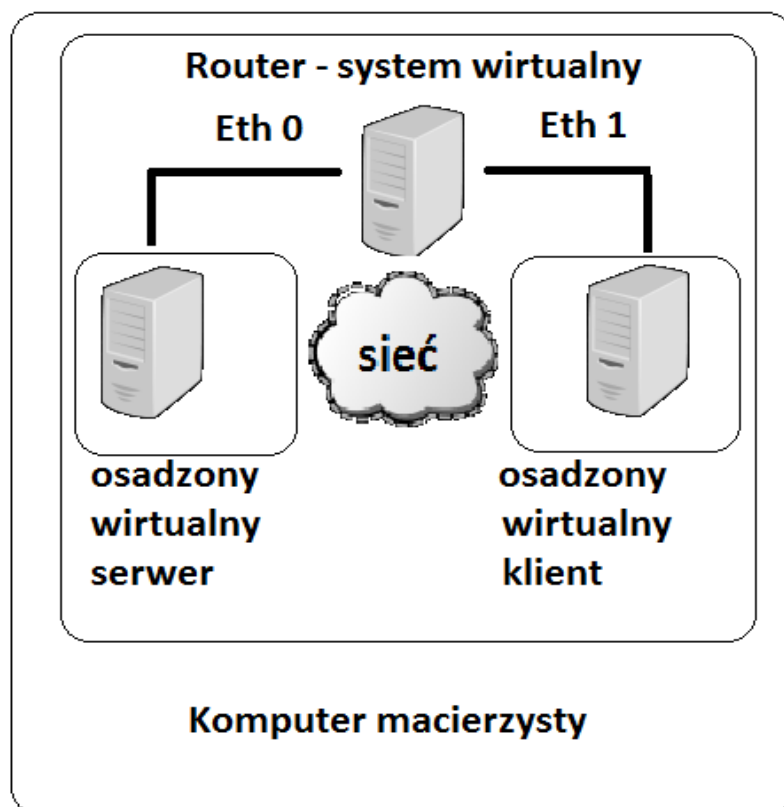
Źródło: opracowanie własne

## 1.1. Aplikacje testowe

Aplikacja testowa realizująca funkcje przetwarzania rozproszonego w oparciu o technologię Java RMI zrealizowana została na podstawie przykładu udostępnionego na stronie macierzystej opisującej technologię [2]. Program został zmodyfikowany do postaci umożliwiającej przesłanie komunikatu o określonej zdefiniowanej długości. W celu porównawczym stworzono aplikację klient-serwer napisaną w całości w języku Java umożliwiającą transmisję danych za pomocą protokołu HTTPS [3]. Ponadto stworzona została klasyczna aplikacja (również w języku Java) w oparciu o system gniazd i transmisję strumieniową z wykorzystaniem protokołu TCP. W każdym z tych programów ilość przesyłanych danych była parametryzowana.

## 1.2. Wirtualne laboratorium sieci komputerowych

W ramach badań wykorzystano wirtualną sieć Ethernet 100 BASE TX o maksymalnej prędkości transmisji 100 Mb/s. Połączenia sieciowe zrealizowano za pomocą zagnieżdżenia maszyn wirtualnych.



**Rys. 2.** Wirtualne środowisko badawcze

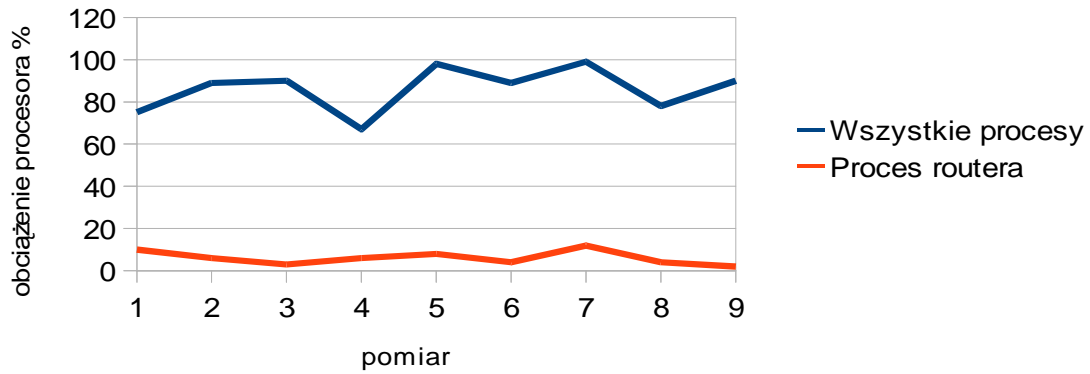
**Źródło:** opracowanie własne

W ramach badań ograniczono liczbę maszyn wirtualnych w sieci do routera i dwóch hostów, przy czym jeden realizował zadania serwera a drugi klienta. Schemat ideowy wirtualnego laboratorium wykorzystanego w badaniach eksperymentalnych przedstawiono na rysunku 2. Maszyny wirtualne skonfigurowane zostały w taki sposób, aby procesy wywoływane przez programy nie przeciążały wirtualizowanych procesorów. Obie maszyny (klient i serwer) pracowały pod kontrolą systemu operacyjnego Windows XP, natomiast router pracował pod kontrolą systemu operacyjnego Linux (dystrybucja Debian) z dwiema maszynami wirtualnymi zrealizowanymi za pomocą Oracle VM VirtualBox [4].

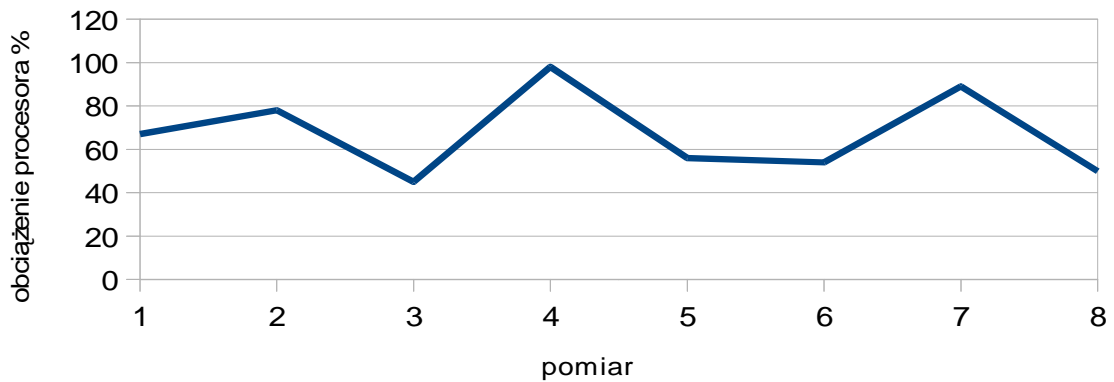
Jednostkę macierzystą stanowił komputer o następującej konfiguracji: procesor AMD Phenom II P840 1,9GHz, 4GB pamięci RAM z 64-bitowym systemem Windows 7. Wszelkie pomiary transferów danych odbywały się na maszynie routera o następującej konfiguracji: delegacja procesora procesor AMD Phenom II P840 z rezerwacją jednego rdzenia, 2 GB pamięci RAM, dwie karty sieciowe w standardzie Ethernet (AMD PCNET Family PCI Ethernet). Dla klienta przeznaczono zagnieżdżoną w systemie routera jednostkę wirtualną o parametrach: 512 MB pamięci RAM, karta sieciowa w standardzie Ethernet, delegacja procesora macierzystego. Serwer wykonano w oparciu o maszynę wirtualną o konfiguracji: 512 MB pamięci RAM, karta sieciowa w standardzie Ethernet, delegacja procesora macierzystego. Jak stwierdzono w wyniku wstępnych badań, konfiguracja ta charakteryzowała się najlepszą stabilnością.

Wyniki obciążenia procesorów routera, klienta i serwera dla poszczególnych aplikacji przedstawione zostały na rysunku 3.

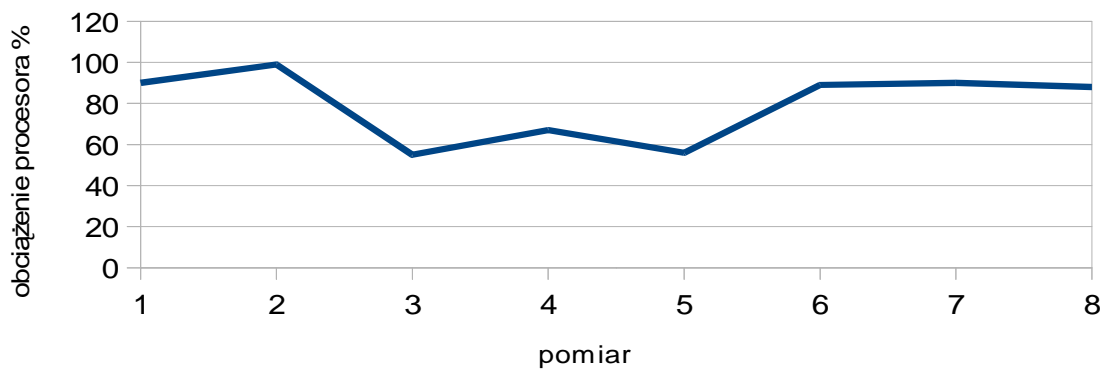
### Obciążenie wirtualnego routera



### Obciążenie wirtualnego klienta wszystkie procesy



### Obciążenie wirtualnego serwera wszystkie procesy



Rys. 3. Obciążenia wirtualnych procesorów przez aplikacje testowe

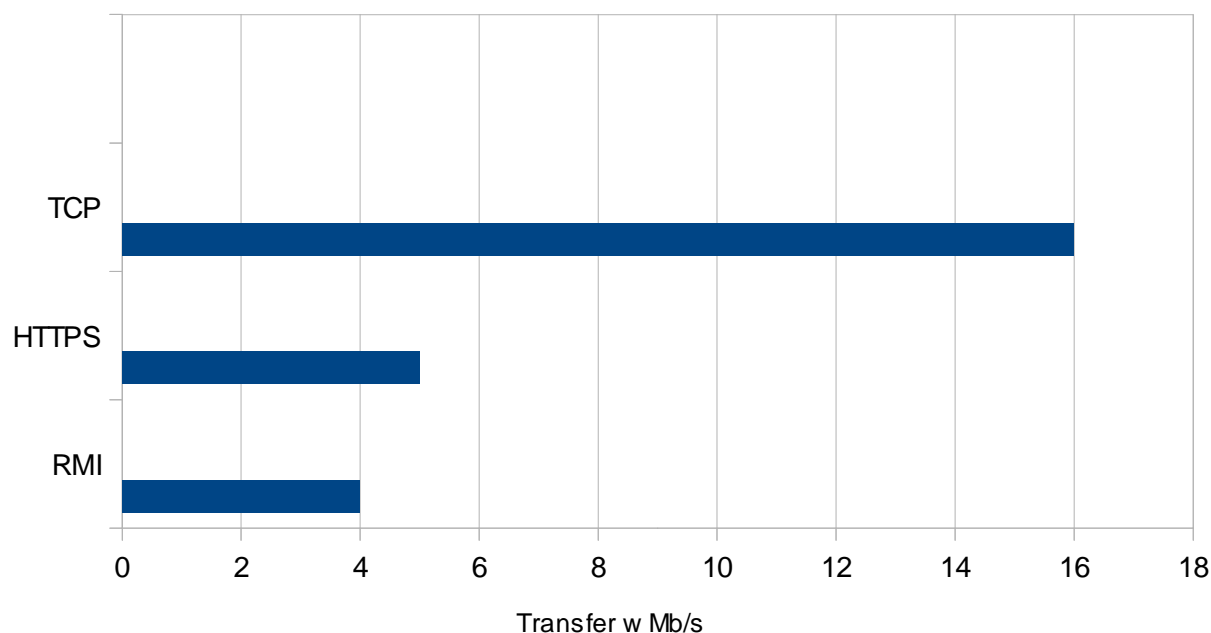
Źródło: opracowanie własne

### 1.3. Badania transferu

Badanie transferu danych przeprowadzono w nieobciążonej sieci. Każda z testowych aplikacji klient – serwer umożliwiała transfer paczek danych o wielkości 10, 20 oraz 30 kilobajtów realizowanych w pętli 100000 wywołań. Głównym zadaniem było określenie punktu nasycenia sieci, w którym dalszy ruch pakietów był niemożliwy i sygnalizowany przez aplikację w postaci wyjątku.

Na rysunku 4 przedstawiającym wyniki badań można zaobserwować zbliżone graniczne wartości transferu dla technologii Java RMI oraz aplikacji z protokołem HTTPS [5]. Porównanie protokołu TCP, należącego do warstwy transportowej modelu ISO/OSI, z protokołami wyższych warstw stanowi dobrą ilustrację dystansu dzielącego aplikacje realizowane w oparciu o system gniazd w stosunku do bardziej zaawansowanych technologii. Technologia RMI jest co prawda stosunkowo łatwa do wdrożenia, jednak nie jest ona optymalna dla intensywnej wymiany pakietów w sieci. Aplikacje z protokołami HTTPS i TCP funkcjonują w takich warunkach znacznie lepiej, jednak jest to okupione większym stopniem trudności wdrożenia protokołów.

Warto zauważyć, iż zjawisko nasycenia transferu jest jedną z przyczyn pojawiania się błędów transmisji, jednak prawidłowo zaprojektowana aplikacja jest w stanie ten problem zmarginalizować.



**Rys. 4.** Uzyskana w wyniku przeprowadzonych badań eksperymentalnych przepustowość dla technologii RMI, HTTPS i TCP

**Źródło:** opracowanie własne

### 1.4. Stabilność aplikacji

Badanie stabilności aplikacji zostało przeprowadzone poprzez skokowe ograniczenie liczby transmitowanych pakietów na routerze przy jednoczesnym pomiarze przepustowości sieci. Badanie przeprowadzono z wykorzystaniem aplikacji klient-serwer z rozproszonymi obiektami Java RMI. Wykazały one małą odporność na ograniczenie transferu danych

w sieci, gdyż wystarczyło ograniczyć przepustowość sieci o 10% w stosunku do zmierzonej uprzednio wartości nasycenia 4MB/s, aby pojawiały się wyjątki sygnalizujące błędy. Wyjątki generowane przez oprogramowanie Java RMI rzadko odpowiadają sytuacji związanej ze zjawiskami związanymi z siecią, co jest związane z ukryciem aspektu sieci. W tych samych warunkach aplikacje z segmentem TCP lub protokołem HTTPS raportują błędy w sposób poprawny.

## PODSUMOWANIE

Technologia Java RMI stanowi dobre narzędzie ukrywające przed programistą aspekt działania sieci, pozwalając skupić się na wdrażaniu algorytmów przetwarzania rozproszonego. Przeprowadzone badania pozwalają na określenie granic transferu danych w sieci, dla których ten problem może zostać pominięty. Warto zauważyć, iż w celu zminimalizowania kosztu badań możliwe jest ich efektywne przeprowadzenie w wirtualnym środowisku laboratorium sieci komputerowych.

# DATA TRANSFER RESEARCH IN THE APPLICATIONS OF DISTRIBUTED OBJECTS - JAVA RMI - ON THE VIRTUAL MACHINES

### *Abstract*

*The article examines the impact of the distributed application developed using Java RMI technology on the data transfer over a computer network. The objective of the conducted tests was determining the load transfer through the network for the RMI technology without restrictions related to the capacity of the network in comparison to the popular protocols such as HTTPS and TCP. The experiments were conducted in a virtual laboratory implemented using virtual machines technology.*

## BIBLIOGRAFIA

1. Nithiyananthan K., Ramachandran V., Dha-Modharan D.: *RMI Based MultiArea Power System Load Flow Monitoring*, IASTED International Conference, Power and Energy Systems in Rhodes, 2001, pp. 249-252.
2. <http://docs.oracle.com/javase/6/docs/technotes/guides/rmi/hello/hello-world.html>
3. <http://www.herongyang.com/JDK/HTTPS-Server-Test-Program-HttpsHello.html>
4. <http://www.virtualbox.org/manual/ch06.html>
5. Juric M.B., Rozman I., Brumen B., Colnaric M., Hericko M.: *Comparison of performance of Web services, WS-Security, RMI, and RMI-SSL*, The Journal of Systems and Software, 2006, vol. 79, pp. 689–700.

### *Autorzy:*

**dr inż. Piotr LECH** – Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

**dr inż. Krzysztof OKARMA** – Zachodniopomorski Uniwersytet Technologiczny w Szczecinie