Low Orbit Task Cannon

System komunikacji dla zarządzania rozproszonymi procesami.

Tomasz Jakubczyk, Eryk Ratyński, Andrzej Roguski, Kacper Stachyra

Treść zadania

W sieci jest zbiór zarządzanych węzłów, **serwer** zarządzający i stacja **konsoli** administratora. W węzłach pracują **agenty** zarządzające. Agent zarządzający może: załadować kod nowego procesu, usunąć kod procesu, uruchomić/zatrzymać/wznowić/zabić dany proces zgodnie z harmonogramem, wznowić proces nie raportujący swej żywotności, podać dane statystyczne serwerowi. **System umożliwia administratorowi zarządzanie rozproszonymi procesami.** System komunikacji powinien móc pracować w przestrzeni adresów IPv4 i IPv6. Ponadto należy zaprojektować moduł do Wireshark umożliwiający wyświetlanie i analizę zdefiniowanych komunikatów.

Co robimy?

- Serwer – zarządza agentami i komunikuje się z konsolą administratora
- Agent (klient) – wykonuje zlecone zadania
- Konsola administratora – pozwala wprowadzać zadania i monitorować działanie aplikacji

Ukryty dół góry lodowej

- Potrzebna synchronizacja czasu, żeby planowo wykonywać zadania – klient NTP
- Potrzebny dobry protokół (już mamy). Jedna implementacja obsługi protokołu.
- Modularny serwer: Kontroler, Serwer klienta, Serwer administratora, Model. Duże podobieństwa do wzorca MVC

Topologia
Low Orbit Task Cannon (LOTC)

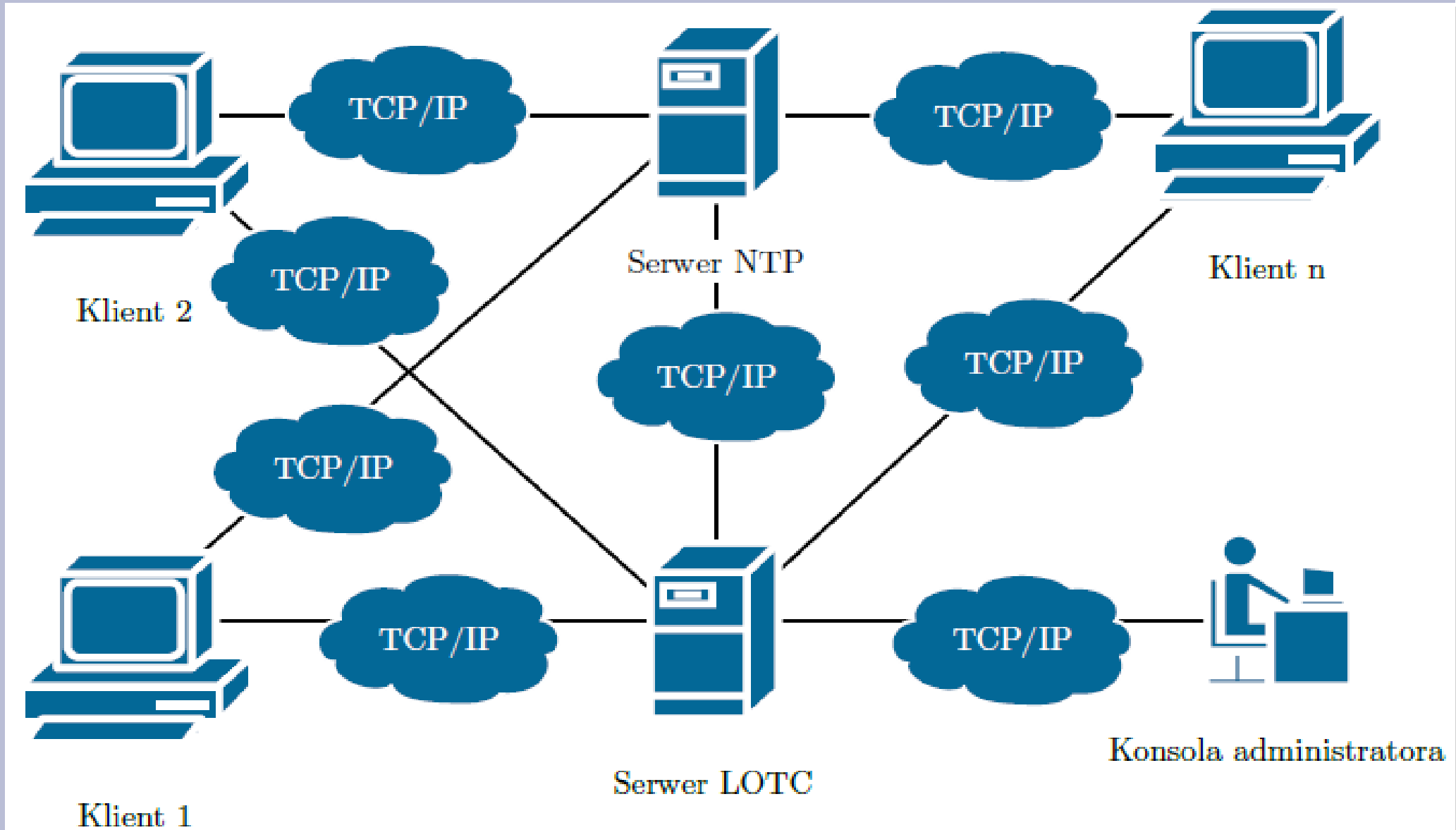
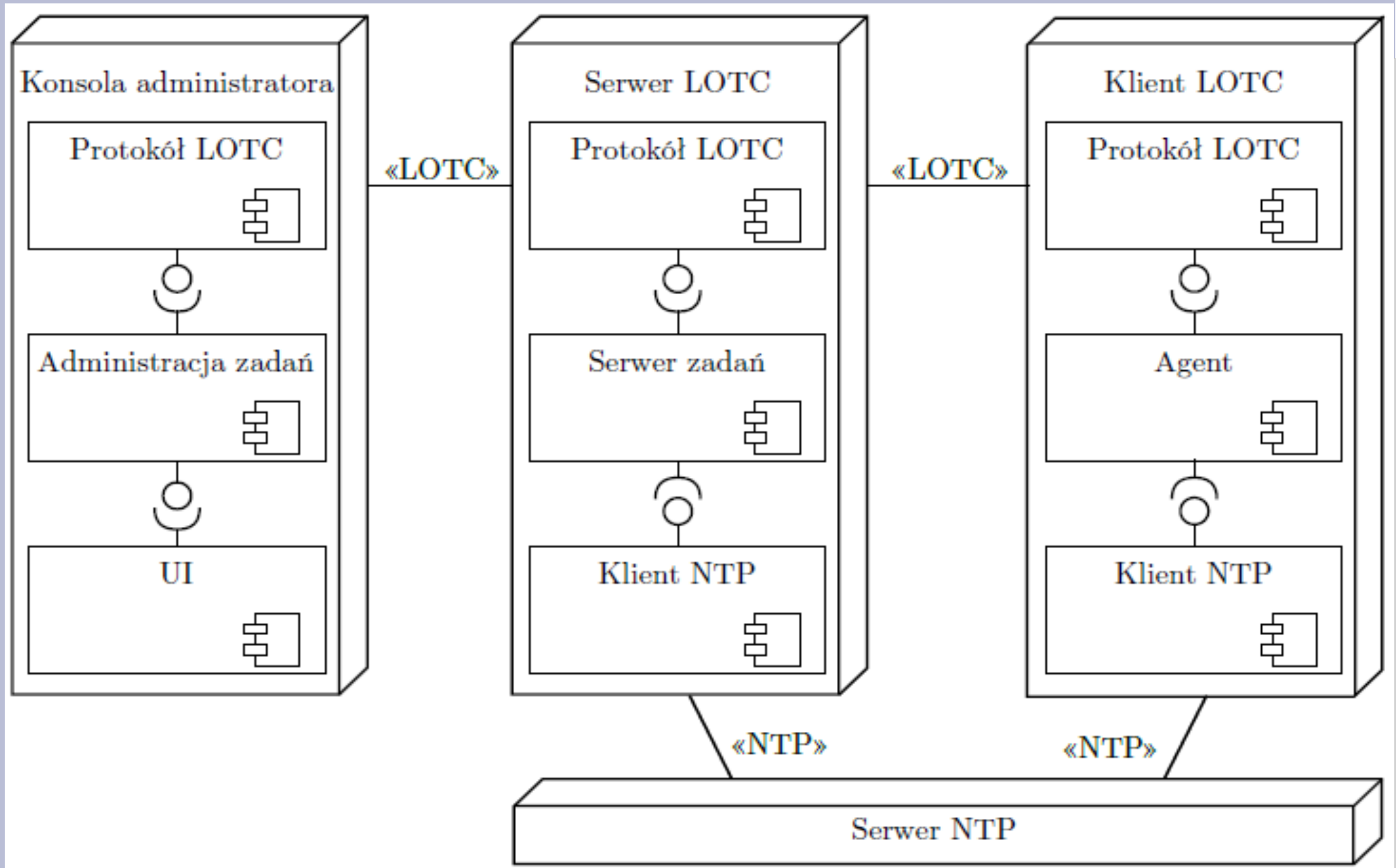


Diagram rozmieszczenia



Środowisko sprzętowo-programowe

- System operacyjny: GNU/Linux
- Język programowania: C++14 (system LOTC), Lua (Plugin Wireshark)
- Biblioteki: Boost >= 1.59 (z wyłączeniem nakładek na API gniazd BSD)
- Kompilator: GCC >= 5.2.0
- Debugger: GDB >= 7.10

Co już zrobiliśmy?

- Dokumentacja wstępna
- Opis protokołu do komunikacji sieciowej
- Szkielet kodu serwera
- Testowy Plugin Wireshark - jeszcze nie uwzględnia opisanego protokołu

Testowanie i prezentacja wyników

- Testy jednostkowe i wykrycie błędów przy kompilacji
- Testy integracyjne
- Skrypty symulujące przypadki użycia

- Demonstracja rezultatów – realizacja wybranych przypadków użycia przez wcześniej napisany skrypt testujący

I to by było na tyle

Repozytorium:
https://github.com/sigrond/procesy_rozproszone