*Paweł Pytel*

*Filip Philavong*

**Monitoring wizyjny – TCP**

Tematem projektu było stworzenie oprogramowania w architekturze klient serwer, służącego do przesyłania obrazu. Klient podłącza się do serwera i udostępnia swój obraz z kamery internetowej, natomiast serwer dla każdego z klientów tworzy nowy wątek, a w nim m.in. nowe okno, w którym wyświetla przesyłany obraz.

Program oparty został na protokole TCP. Serwer zaimplementowano w C/C++ natomiast aplikacja klient została napisania w języku Java.

Klient podłączając się do serwera, wysyła liczbę w postaci Integera, która mówi ile paczek nadeśle – jest to niezbędne, aby zaimplementować buforowanie, które i tak było konieczne – obraz nie zmieściłby się w pojedynczym pakiecie. Po wysłaniu informacji o liczbie paczek, klient wysyła paczki w odstępach czasowych (czekając na ACK od serwera), natomiast serwer kompletuje paczki, składając z nich pojedynczy obraz i wyświetla w stworzonym wcześniej oknie. Procedura powtarza się tak długo, aż klient się nie odłączy – wysyłając liczbę Integer równą 55, lub nastąpi błąd komunikacji – wtedy serwer sam odłączy użytkownika, który będzie musiał ponownie się podłączyć. Komunikacja pomiędzy serwerem a klientami odbywa się na porcie 6970.

Serwer i klient zostali zaimplementowani z użyciem biblioteki OPENCV. Umożliwia ona odbiór obrazu z kamerki internetowej oraz dekodowanie odebranych przez serwer danych, w postać obrazu. Serwer oparty został o serwer wielowątkowy z zajęć laboratoryjnych. Dla każdego nowego klienta tworzony jest osobny wątek i okno, w którym wyświetlany jest obraz. Klient pobiera obraz z kamerki internetowej w osobnym wątku niż wątek graficzny – po wciśnięciu przycisku połączenia i rozpoczęcia komunikacji, program nie „blokuje się” i można go nadal używać.

Do skompilowania programu klienta, można posłużyć się środowiskiem Eclipse, importując biblioteki, które znajdują się w pliku z rozszerzeniem .java, lub uruchamiając plik .jar.

Aby skompilować program serwera należy : użyć przełączanika –lpthread, aby móc używać wielowątkowości oraz podpiąć w linkerze odpowiednie biblioteki: najlepszym i najbezpieczniejszym jest podpięcie wszystkich plików z rozszerzeniem .so, które znajdują się w folderze lib w lokalizacji instalacji biblioteki OPEN CV.