Università di Ferrara

Architettura di Reti

Lab 6

Carlo Giannelli
carlo.giannelli@unife.it
http://www.unife.it/scienze/informatica/insegnamenti/architettura-reti/
http://docente.unife.it/carlo.giannelli

Esercizio Lista Articoli

Si progetti un'applicazione Client/Server che, utilizzando le socket, implementi un sistema di peer-review per la gestione degli articoli per una conferenza scientifica. Più precisamente, l'applicazione deve permettere ai revisori, incaricati di selezionare gli articoli per la conferenza, di stampare la lista di articoli da giudicare a essi assegnati. Il Client deve offrire la seguente interfaccia:

lista_articoli server porta

dove "server" e "porta" sono rispettivamente il nome logico dell'host e il numero di porta su cui si trova il Server. Il Client si interfaccia con il revisore da cui riceve via terminale **username** (intero positivo), **indirizzo email** (stringa), **password** (intero). Si noti che lo **username è opzionale** (in tal caso l'utente specifica il valore -1) mentre **indirizzo email e password sono obbligatori**.

Esercizio Lista Articoli (cont.)

Per ogni richiesta, il Client deve comunicare le informazioni inserite dal revisore al Server. Il Server deve quindi verificare l'autorizzazione del revisore invocando un'apposita funzione, che si suppone essere già implementata e con il seguente prototipo:

int autorizza(char *id, int password);

Come primo argomento (id) bisogna passare **lo username (trasformato in stringa) se specificato dall'utente, altrimenti l'indirizzo email**. Se la funzione "autorizza" restituisce il valore 1, l'utente è autorizzato ad accedere al servizio. Altrimenti il Server dovrà rifiutarsi di eseguire il servizio e restituire al Client un opportuno messaggio di errore.

Il Server deve quindi selezionare le informazioni sugli **articoli da revisionare assegnati al revisore** e restituirle al Client, elencando gli articoli **in ordine di data di scadenza decrescente**.

Esercizio Lista Articoli (cont.)

Si supponga che sul Server le informazioni su tutti gli articoli da revisionare siano salvate in un file. Tale file, per ciascuno degli articoli registrati nel sistema, presenta una riga con: **data di scadenza** per la revisione in formato YYYYMMDD**, ID dell'articolo, indirizzo e-mail revisore, titolo dell'articolo,** ecc.

Una volta ricevute le informazioni dal Server, il Client le stampa a video e termina.

Versioni da implementare

Si richiede di inviare le informazioni da client a server tramite:

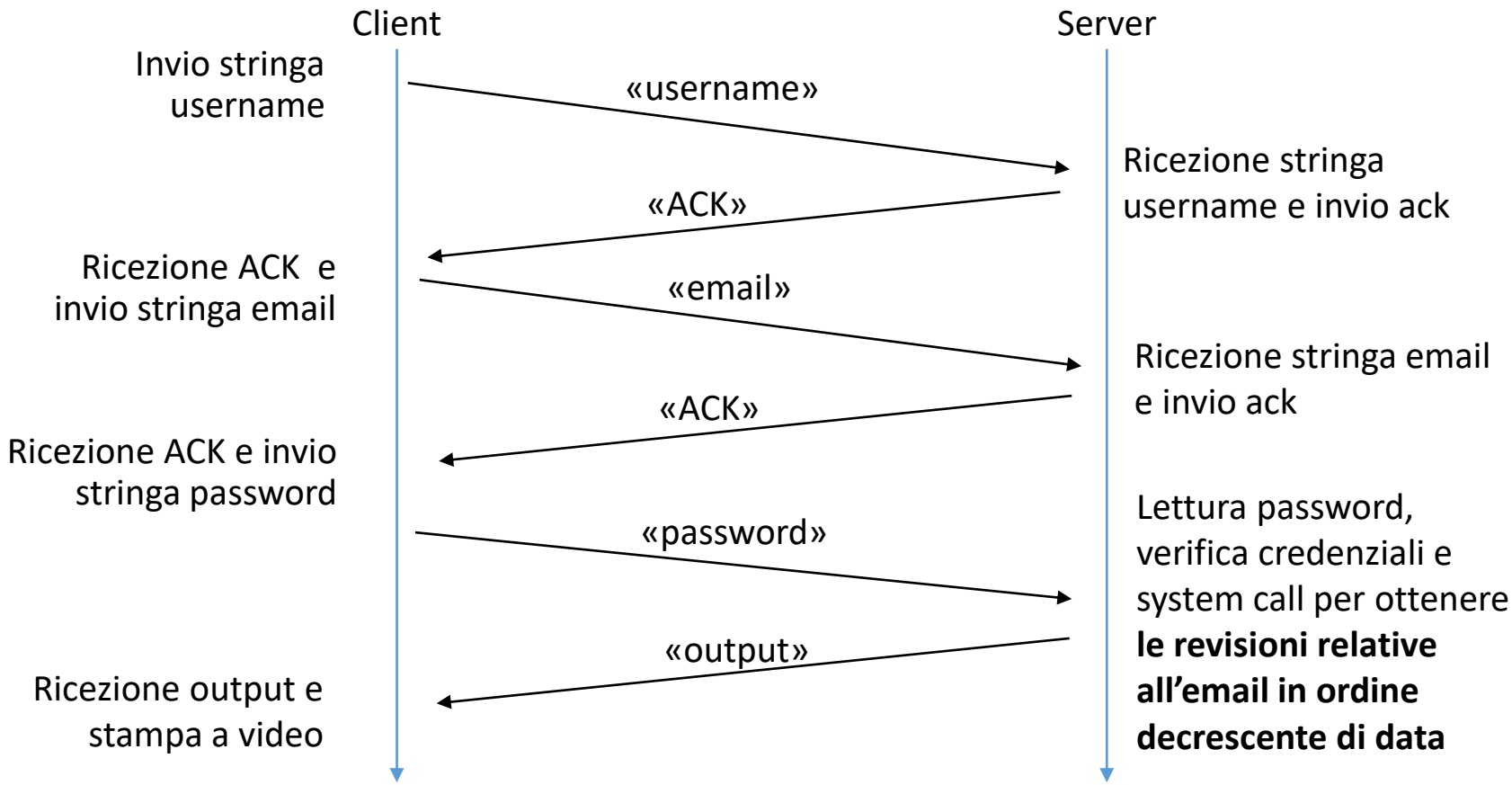
- protocollo testuale: username, email e password come stringhe
- protocollo binario: email come stringa e username/password come interi serializzando tramite
  - struct C
  - Protobuf, senza e con l'indicazione della dimensione dei dati trasmessi

Attenzione! Nel caso lo username specificato sia -1, il messaggio Protobuf non deve contenere lo username.

Lista articoli stringhe + ack

articoli.txt

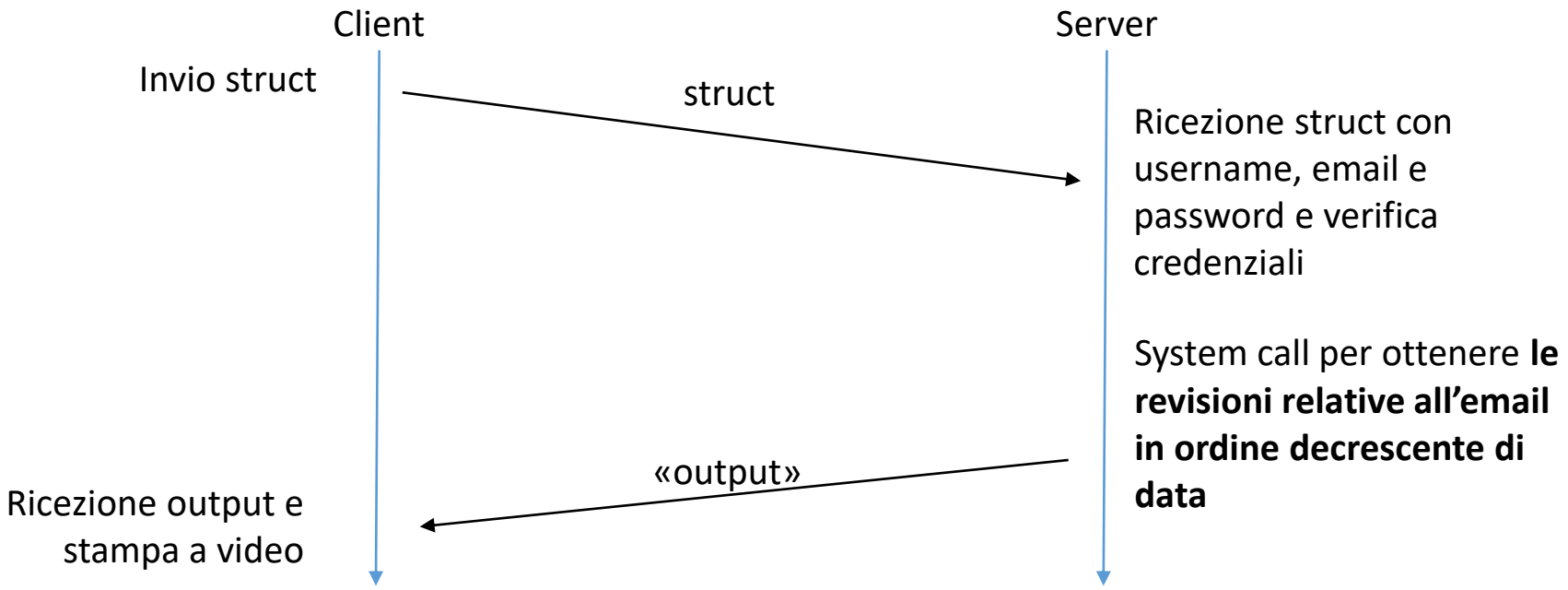
```
20170101 8 luca@luca.it
20171231 5 luca@luca.it
20173012 7 lg@lg.it
20143012 4 pippo@pippo.it
```



Lista articoli struct

articoli.txt

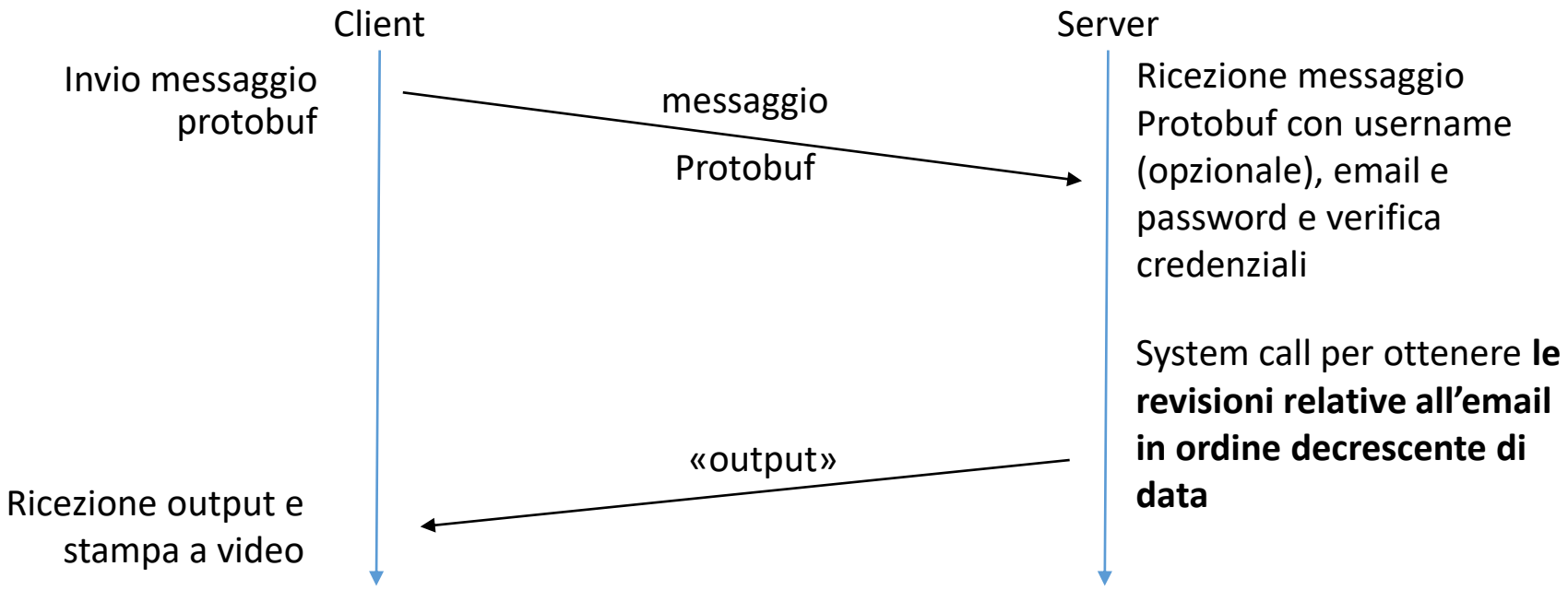
```
20170101 8 luca@luca.it
20171231 5 luca@luca.it
20173012 7 lg@lg.it
20143012 4 pippo@pippo.it
```



Lista articoli Protobuf

articoli.txt

```
20170101 8 luca@luca.it
20171231 5 luca@luca.it
20173012 7 lg@lg.it
20143012 4 pippo@pippo.it
```



Lista articoli dim + Protobuf

articoli.txt

```
20170101 8 luca@luca.it
20171231 5 luca@luca.it
20173012 7 lg@lg.it
20143012 4 pippo@pippo.it
```