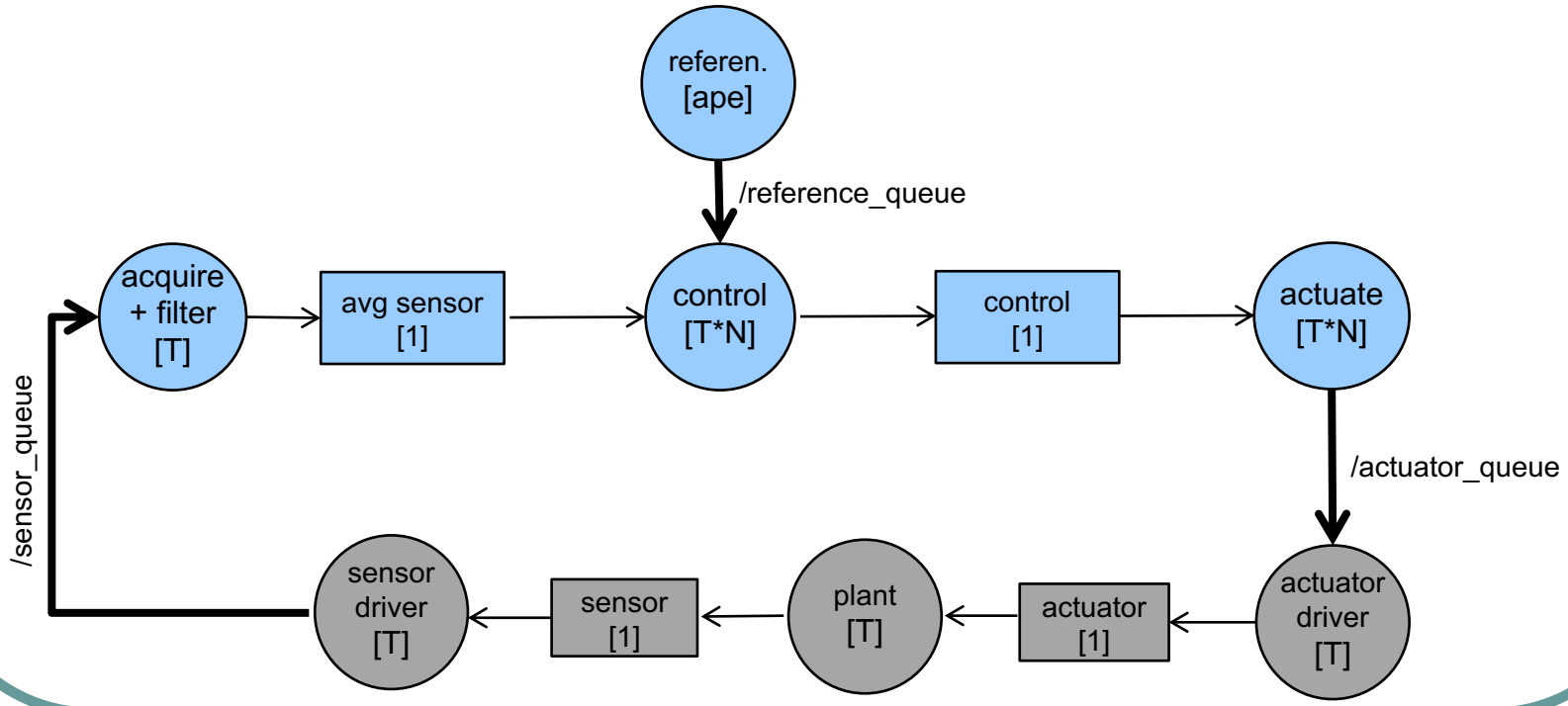
Schema di partenza: loop di controllo

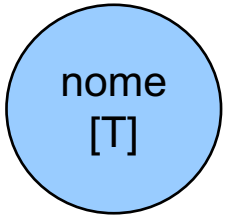
- Una tipica applicazione di controllo prevede più stadi di processing, realizzati da altrettanti task periodici:
  - Acquisizione dati dai sensori + filtraggio (ad es., rimozione rumore)
  - Elaborazione legge di controllo
  - Invio dati agli attuatori
- Ognuno di tali task può essere replicato per quante sono le grandezze da controllare nell'impianto
- Inoltre, task accessori (anche aperiodici) possono essere aggiunti per monitoraggio, diagnostica, …
- La sincronizzazione e comunicazione tra i task può essere realizzata con le primitive POSIX illustrate

Schema di partenza: loop di controllo



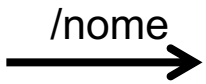
Simbologia dello schema

- I cerchi rappresentano i task
  - Il valore tra parentesi quadre rappresenta il periodo del task, o “ape” per i task aperiodici

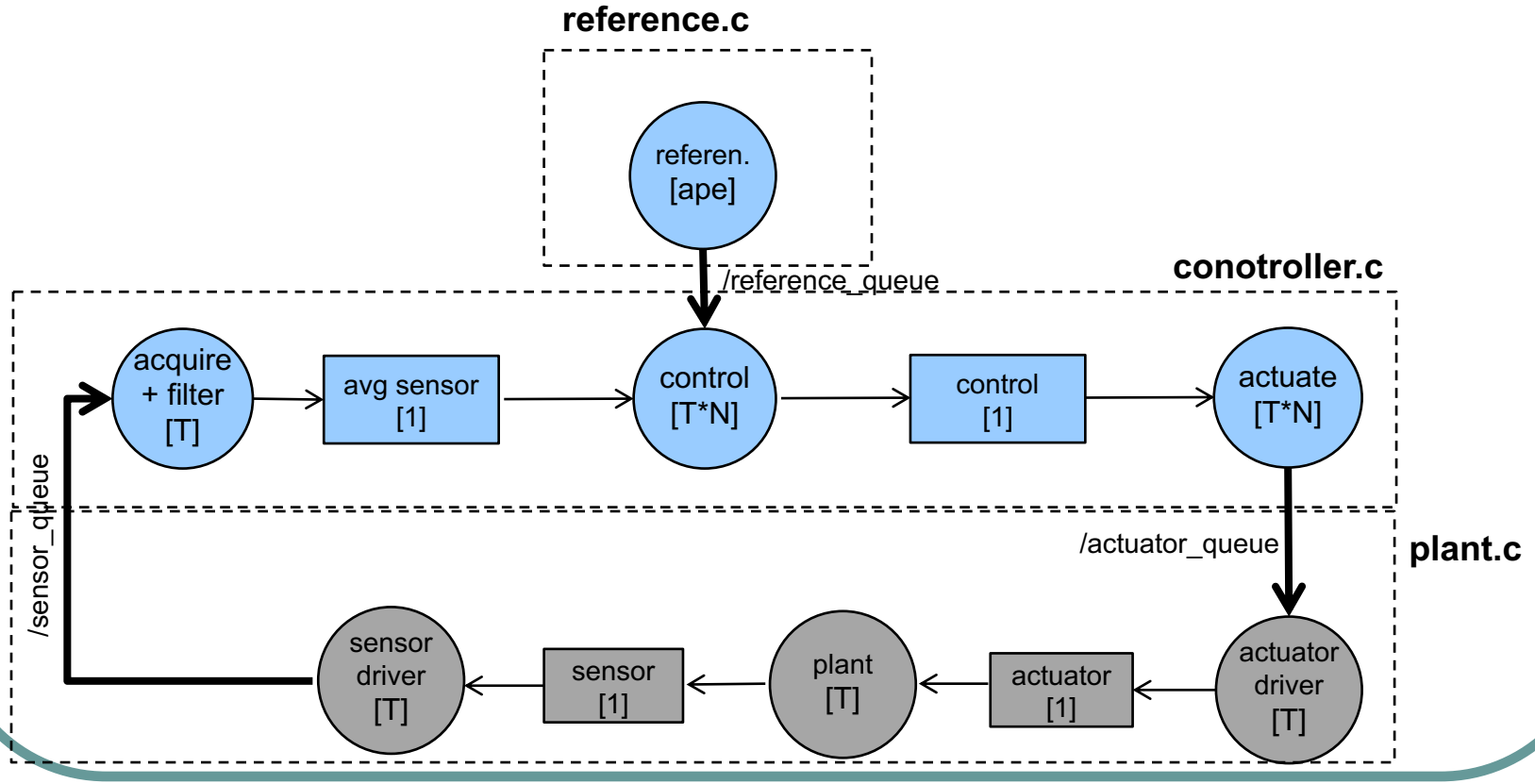


- I rettangoli rappresentano le risorse condivise
  - Il valore tra parentesi quadre rappresenta la dimensione della risorsa
  - Le risorse si intendono protette da mutex

- Le frecce dirette tra task rappresentano messaggi con comunicazione tramite code



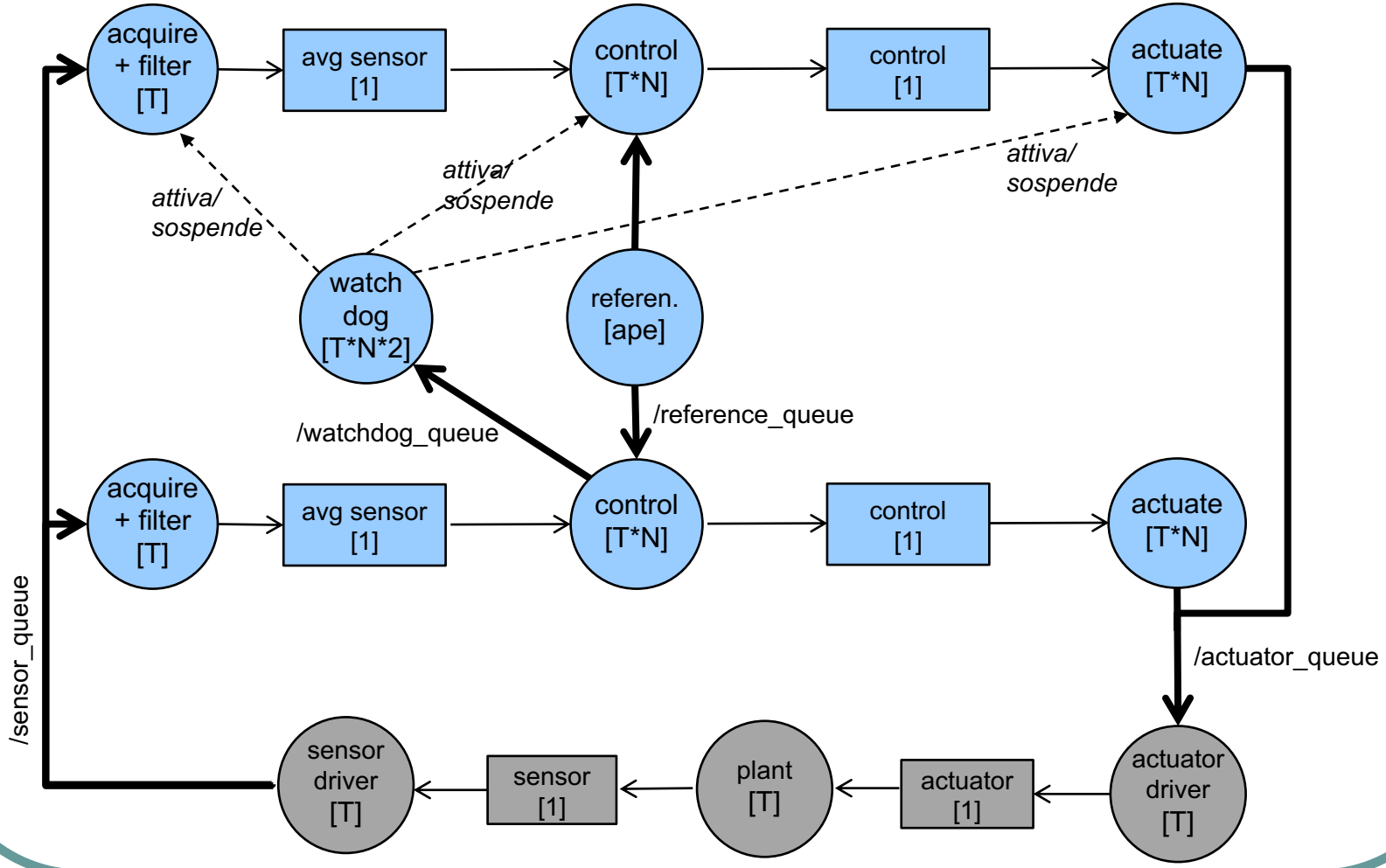
Schema di partenza: mapping sui file sorgente



Traccia

- Si vuole tollerare un eventuale crash del controllore attraverso l'attivazione di una *replica*
- Si vuole inoltre dotare il controllore di un sistema di diagnostica asincrona gestito da un *Deferrable Server (DS)*

Schema da realizzare: la replica e il watchdog



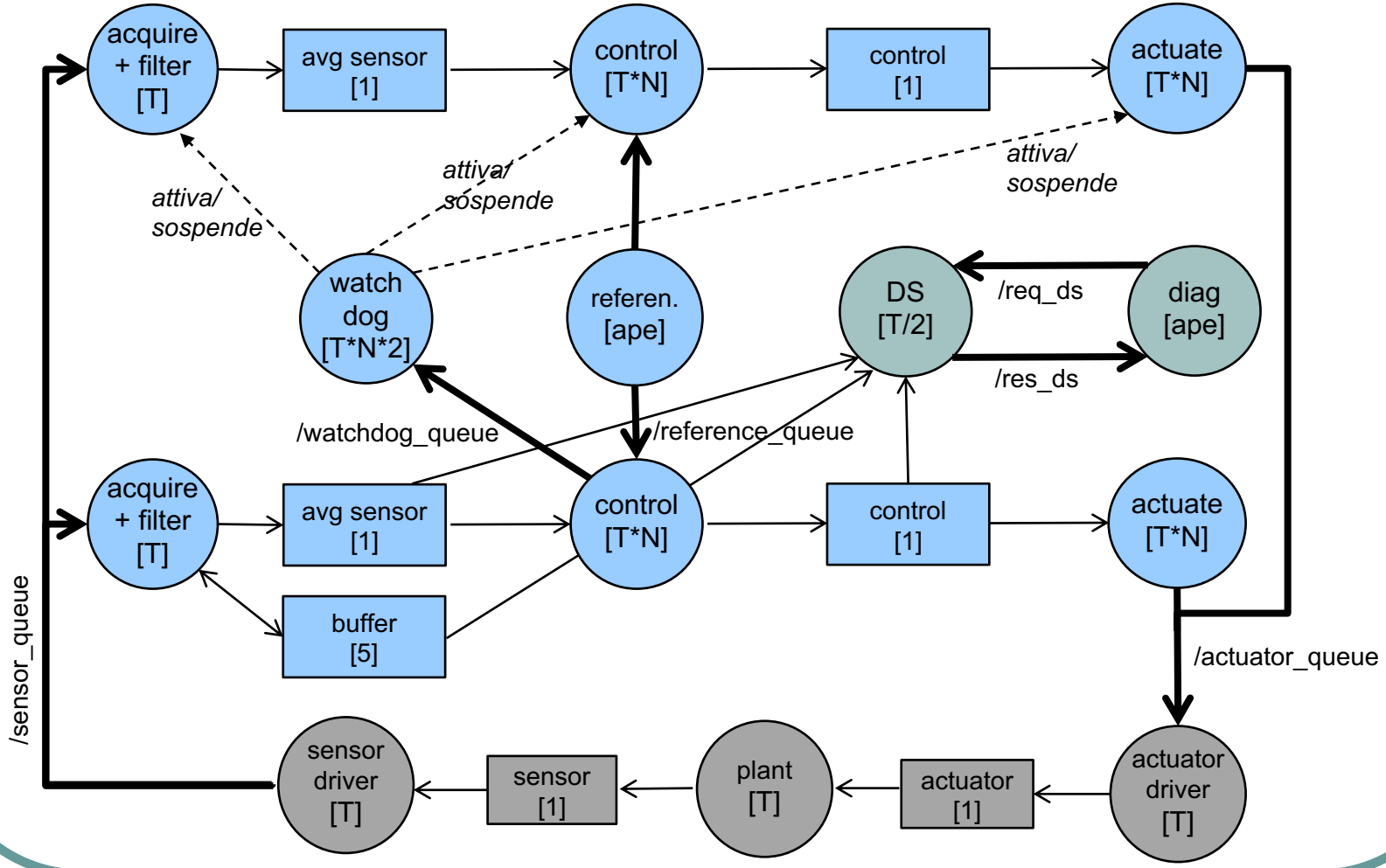
Traccia: replica e watchdog

- Implementare la replica e il watchdog in un file replica.c
- Il watchdog è un task periodico in attesa di un «segnale» di heartbeat da parte del controllore
  - Se non riceve il segnale entro 2 periodi del controllore, attiva le repliche
  - Appena torna a ricevere il segnale, spegne le repliche
- Il controllore in controller.c va modificato per mandare ogni due cicli un segnale sulla coda del watchdog
- Le repliche possono essere attivate o disattivate con diverse strategie
  - Il watchdog avvia i relativi task (pthread_create) e poi li ferma (pthread_kill)
  - Oppure il watchdog modifica lo stato di una variabile globale (*attivo*) che determina l'esecuzione o meno del ciclo dei rispettivi task

Note sul watchdog

- Si consiglia di far partire prima la replica, poi il plant e poi il controller
  - Altrimenti alla partenza il watchdog potrebbe trovare la coda già piena di heartbeat e rimanere sempre in ritardo
  - In alternativa, la periodicità del watchdog può essere realizzata tramite la receive
- Si consiglia di inviare attraverso l'heartbeat l'ultima reference impostata
  - In modo che la replica possa partire dall'ultima reference

Schema da realizzare: diagnostica attraverso DS



Traccia

- Il task diag, quando avviato, invia una richiesta di diagnostica al controller tramite la coda /req_ds
- Il DS, quando riceve una richiesta sulla coda /req_ds
  - Preleva i valori delle variabili:
    - Avg_sensor
    - control
    - buffer (i 5 elementi contenuti nel buffer del task acquire_filter)
    - la reference
  - Li invia al task diag sulla coda /res_ds
- Quando il task diag riceve la risposta, stampa lo stato del controllore a video e termina
- Per semplicità si suppone che la richiesta venga evasa entro la capacità del DS

Note

- Tutti i task periodici devono essere schedulati con Rate Monotonic
- E’ opportuno dunque che eseguano tutti sulla stessa CPU (impostando l’affinity o utilizzando il comando taskset)

Traccia: Calcolo WCET (opzionale)

- Si effettui il calcolo del WCET dei 3 task del file controller.c
- I valori di WCET (calcolati come massimi dei tempi di esecuzione dei rispettivi task) vanno memorizzati in tre variabili globali
- Tali valori vengono letti e restituiti dal DS nel caso di richiesta di diagnostica, unitamente al fattore di utilizzazione cumulativo calcolato per i tre task

Traccia: Uso di priorità nei messaggi per il DS (opzionale)

- Si implementi un task separato di diagnostica, diagWCET, che chiede solo il dato sul WCET
- diagWCET invia la richiesta ad una priorità più alta di diag
- Il tipo di richiesta viene gestita attraverso un valore inserito nel messaggio
- Il DS, a seconda del valore ricevuto, preleva tutto, oppure solo il dato sul WCET.

Consegna

- Entro le 20:00 del 10 maggio 2022
- E’ possibile lavorare in gruppi di massimo 4 persone
- Ogni studente deve consegnare individualmente l’elaborato su Teams, aggiungendo, se ha lavorato in gruppo, un file di testo con i nominativi e matricole dei componenti del gruppo