

Logica di routing

Aggiornamento dei parametri generici

L'aggiornamento del valore di traffico e feromone avviene in maniera identica a prescindere dal tipo di routing utilizzato

Quando un pacchetto attraversa un nodo:

1. sceglie il nodo successivo attraverso il suo algoritmo di routing
2. aggiorna il valore di traffico nel canale corrispondente (si somma il peso del veicolo: truck 20, altri veicoli 1)
3. aggiorna il valore di feromone nel canale corrispondente (si somma 1 al passare di ogni veicolo)
4. viene calcolato il tempo di attraversamento del canale (funzione della distanza, della velocità del veicolo e del traffico attuale)
5. viene simulato l'attraversamento del canale con un'attesa pari al tempo di attraversamento tramite schedulazione ritardata
6. trascorso il tempo di attraversamento, viene inviato fuori istantaneamente al nodo prescelto
7. viene aggiornato il valore di traffico nel canale corrispondente (si sottrae il peso del veicolo)

In pseudocodice:

```
channelOut = pickChannelToNextNode(path)
//path è calcolato attraverso uno degli algoritmi di routing

updateTraffic(channelOut)
updatePheromone(channelOut)

waitTime = calculateTravelTime(speed, distance, getTraffic(channelOut))
waitFor(waitTime)

send(channelOut) //istantaneo

decayTraffic(channelOut)
```

- La decadenza del feromone avviene periodicamente ad intervalli regolari (utilizza il parametro `pheromoneDecayTime` misurato in secondi)
- Esso decade di una percentuale del valore corrente (utilizza il parametro `pheromoneDecayFactor` [0,1])
- La decadenza avviene nello stesso istante per tutti i nodi della rete.

every `pheromoneDecayTime`:

```
for (each node in grid){
    decayPheromone(pheromoneDecayFactor)
}
```

- Se il pacchetto è arrivato al nodo di destinazione, il routing provvede a inviarlo al modulo dell'applicazione.

Algoritmi di Routing

- Sia *calculateWeightedSingleShortestPathsTo*(destAddress) che *calculateUnweightedSingleShortestPathsTo*(destAddress) calcolano tutto il percorso dal nodo in cui vengono invocati alla destinazione usando il corrispettivo algoritmo di Dijkstra
- Dopo aver trovato il canale da attraversare per raggiungere il nodo successivo, i puntini fanno riferimento allo pseudocodice scritto sopra.

(I nomi sono provvisori)

Manhattan Routing (Dijkstra unweighted)

```
for each hop:

    if (myAddress != destAddress){
        path=calculateUnweightedSingleShortestPathsTo(destAddress)
        channelOut = pickChannelToNextNode(path)
        ..
    }
```

- Ogni hop è considerato con peso 1.

WeightedDijkstraTraffic

```
for each hop:

    if (myAddress != destAddress){
        for (each channel of each node in grid){
            setChannelWeight(actualTraffic)
        }
        path=calculateWeightedSingleShortestPathsTo(destAddress)
        channelOut = pickChannelToNextNode(path)
        ..
    }
```

Il peso viene aggiornato ad ogni hop poichè esso potrebbe essere diverso.

Ad ogni hop viene ricalcolato tutto il percorso fino a destinazione con i pesi aggiornati.

WeightedDijkstraPheromone

L'unica differenza con il routing precedente sono i valori dei pesi che, invece di essere aggiornati con i valori del traffico, vengono aggiornati con quelli del feromone.

```
for each hop:

    if (myAddress != destAddress){
        for (each channel of each node in grid){
            setChannelsWeight(actualPheromone)
        }
        path=calculateWeightedSingleShortestPathsTo(destAddress)
        channelOut = pickChannelToNextNode(path)
        ..
    }
```

Dijkstra pesato sul feromone è simile ad un approccio AAA: * in AAA il veicolo guarda il feromone a distanza di 1 hop e sceglie il percorso con il valore minore. * Nel nostro algoritmo il veicolo valuta i percorsi fino a destinazione e sceglie quello con il minore valore di feromone complessivo.

Il peso viene aggiornato ad ogni hop poichè esso potrebbe essere diverso.

Ad ogni hop viene ricalcolato tutto il percorso fino a destinazione con i pesi aggiornati.

WeightPheromonCivil_UnweightAmbulances

```
for each hop:

    if (myAddress != destAddress){
        if (vehicle == Ambulance){
            path=calculateUnweightedSingleShortestPathsTo(destAddress)
        }
        else{
            for (each channel of each node in grid){
                setChannelWeight(actualPheromone)
            }
            path=calculateWeightedSingleShortestPathsTo(destAddress)
        }
        channelOut = pickChannelToNextNode(path)
        ..
    }
```

La logica di routing cambia in base al veicolo che sta attraversando il nodo.

Le ambulanze utilizzano Dijkstra non pesato mentre tutti gli altri veicoli utilizzano il feromone.