```
task $\mathcal{P}_i$.Join($\mathcal{R}$)
    wait for $[\mathcal{S}_j] \leftarrow \mathcal{R}$
    for wait for $\mathcal{P}_k \in [\mathcal{T}_j] \leftarrow \mathcal{S}_j$ do
        $[\texttt{hello}] \rightarrow \mathcal{P}_k$
        $\mathrm{RTT}_k$ = timeit wait for $[\texttt{hello}] \leftarrow \mathcal{P}_k$ with timeou
    $N(\mathcal{P}_i) = \{\mathcal{P}_k \mid \mathrm{RTT}_k < \mathrm{timeout}\}$
    $N^*(\mathcal{P}_i) = \{\textbf{first } K\ \mathcal{P}_k \in N(\mathcal{P}_i) \mid \mathrm{RTT}_k < \mathrm{RTT}_{k+1}\}$
task $\mathcal{P}_o$.Leave($\mathcal{S}_j$)
    $[\texttt{goodbye}] \rightarrow \mathcal{S}_j$
    for $\mathcal{P}_k \in N(\mathcal{P}_o$ do $[\texttt{goodbye}] \rightarrow \mathcal{P}_k$
    relay pending chunks
    wait for $[\texttt{goodbye}] \leftarrow \mathcal{S}_j$ with timeout
    while timeout do
        $[\texttt{goodbye}] \rightarrow \mathcal{S}_j$
        wait for $[\texttt{goodbye}] \leftarrow \mathcal{S}_j$ with timeout
task $\mathcal{P}_k$.EchoHello()
    while True do
        wait for $[\texttt{hello}] \leftarrow \mathcal{P}_i$
        $[\texttt{hello}] \rightarrow \mathcal{P}_i$
        $N(\mathcal{P}_k) = (N\mathcal{P}_k) \cup \mathcal{P}_i$
task $\mathcal{P}_k$.ProcessGoodbye()
    while True do
        wait for $[\texttt{goodbye}] \leftarrow \mathcal{P}_o$
        $N(\mathcal{P}_k) = N(\mathcal{P}_k) \setminus \mathcal{P}_o$
task $\mathcal{P}_k$.ShortestPathRouting()
    for all $\mathcal{P}_i \in N(\mathcal{P}_k)$ do
        $D[\mathcal{P}_i] = \infty$
    $D[\mathcal{P}_k] = 0$
    for all $\mathcal{P}_i \in N^*(\mathcal{P}_k)$ do
        $G[\mathcal{P}_i] = \mathcal{P}_i$
    for all $\mathcal{P}_i \in N^*(\mathcal{P}_k)$ do
        $[G[\cdot]] \rightarrow \mathcal{P}_i$
    while True do
        $\mathrm{found\_shorter\_route} = \mathrm{false}$
        $[D^*_{\mathcal{P}_n}[\cdot], G^*_{\mathcal{P}_n}[\cdot]] \leftarrow$ from any $\mathcal{P}_n \in N^*(\mathcal{P}_k)$
        for all $D^*_{\mathcal{P}_n}[\mathcal{P}_i] \in D^*_{\mathcal{P}_k}[\cdot]$ do
            $\mathrm{alternative\_distance\_}\mathcal{P}_i = D^*_{\mathcal{P}_n}[\mathcal{P}_i] + D[\mathcal{P}_n]$
            if $\mathrm{alternative\_distance\_}\mathcal{P}_i < D[\mathcal{P}_j]$ then
                $\mathrm{found\_shorter\_route} = \mathrm{true}$
                $D[\mathcal{P}_i] = \mathrm{alternative\_distance}$
                $G[\mathcal{P}_i] = \mathcal{P}_n$
        if found_shorter_route then
            for all $\mathcal{P}_i \in N^*(\mathcal{P}_k)$ do
```

```
                $[D[\cdot], G[\cdot]] \rightarrow \mathcal{P}_i$
task $\mathcal{P}_k$.ChunkFlooding()
    while True do
        wait for $[x, \texttt{chunk}] \leftarrow \mathcal{S}_j$ or $[x, \mathcal{P}_i, \texttt{chunk}] \leftarrow \mathcal{P}$
        if sender $= \mathcal{S}_j$ then
            for all $\mathcal{P}_n \in N^*(\mathcal{P}_k)$ do
                $[x, \mathcal{P}_k, \texttt{chunk}] \rightarrow \mathcal{P}_n$
        else
            for all $\mathcal{P}_n \in N^*(\mathcal{P}_k) \setminus \mathcal{P}_m$ do
                if $\mathcal{P}_k \in G^*_{\mathcal{P}_n}[\mathcal{P}_i]$ then
                    $[x, \mathcal{P}_i, \texttt{chunk}] \rightarrow \mathcal{P}_n$
task $\mathcal{P}_k$.FreeRidingControl()
    while True do
        for all $\mathcal{P}_o \in N(\mathcal{P}_k)$ do
            & wait for $[x, \mathcal{P}_i, \texttt{chunk}] \leftarrow \mathcal{P}_o$
              with timeout of $L$ rounds
              on timeout $N(\mathcal{P}_k) = N(\mathcal{P}_k) \setminus \mathcal{P}_o$
```