



Ministero dello Sviluppo Economico



INDUSTRIA 2015



I-mule



Progetto I-MULE

www.progetto-imule.it



DIPARTIMENTO DI INGEGNERIA
INFORMATICA AUTOMATICA E
GESTIONALE ANTONIO RUBERTI

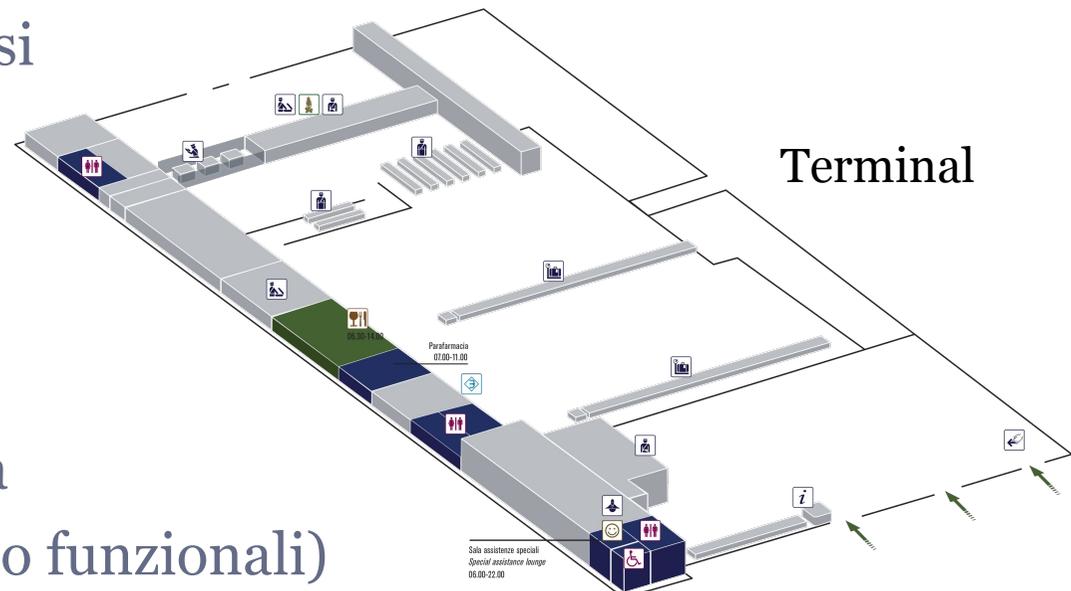
Motivazioni e finalità

- criticità dei sistemi di smistamento e instradamento dei bagagli mediante nastri trasportatori in ambito aeroportuale
 - costosi, ingombranti, con tempi di set-up elevati
 - poco flessibili, con basso parallelismo (fermo guasti)
- uso alternativo di una flotta di veicoli robotici tra check-in e baia di carico (interno aeroporto)
 - autonomia completa dei veicoli, con supervisione centrale
 - stessa gestione (controllo RX sicurezza, ecc.) per *singoli* bagagli
 - dimensione veicoli compatibili con i sistemi esistenti (interfacce)
- pensato come sistema complementare per gestire 'late arrival' ai check-in di grandi aeroporti o come metodo esclusivo di trasporto bagagli nei nuovi piccoli aeroporti

Area di lavoro con componenti presenti

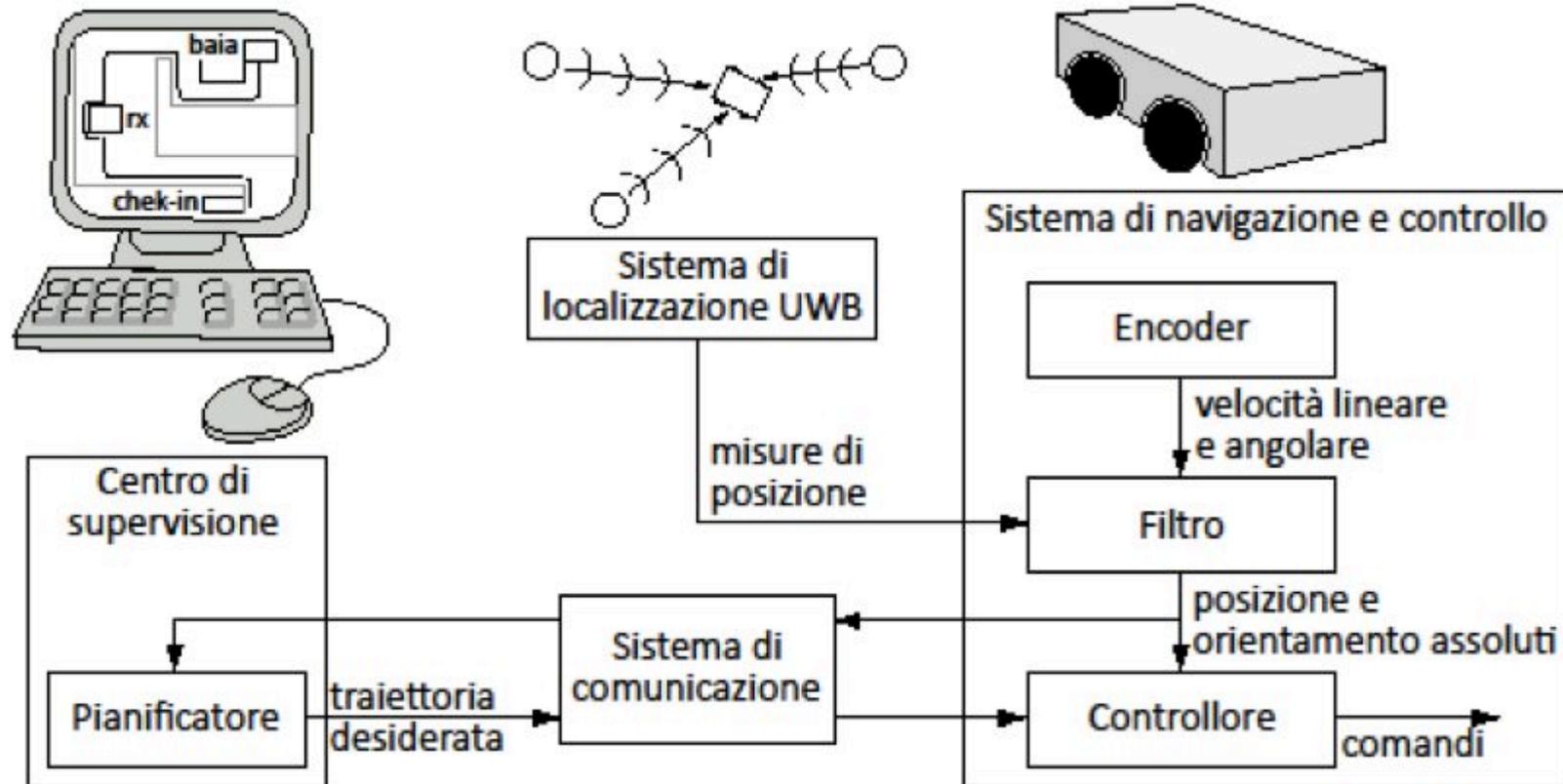
- layout di test

- geometria e dimensioni (inclusa presenza di piani inclinati)
- zone libere e ostacoli fissi
- # veicoli robotici
- # banchi check-in
- # baie di carico
- 1-2 stazioni di ricarica
- 1-2 apparati rx sicurezza
- #/capacità buffer (fisici o funzionali)
- # cammini e relative corsie (lanes, clearance)
- beacon per localizzazione globale/supervisione (copertura)
- telecamere sorveglianza



Architettura di sistema

supervisione, pianificazione, comunicazione,
localizzazione, navigazione, controllo

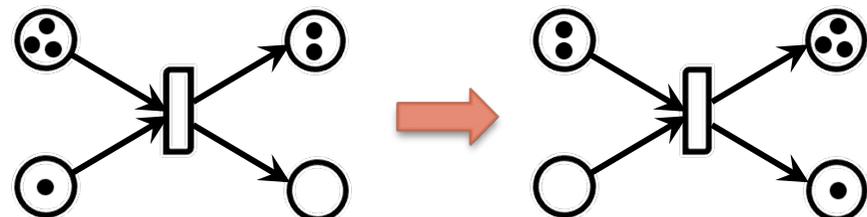
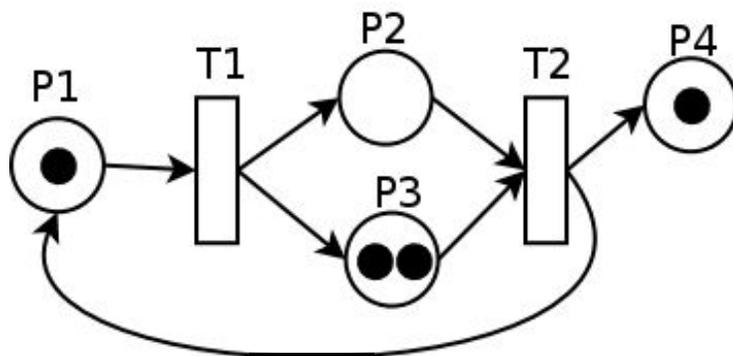


Modellazione

- **esempi di eventi da gestire**
 - richiesta da banco di check-in \Rightarrow assegnazione di un veicolo
 - veicolo in approccio al check-in \Rightarrow attiva controllo per allineamento veicolo al banco (tolleranza laterale ± 2 cm)
 - check-in completato \Rightarrow instradamento verso baia di carico ("corridoio")
 - bagaglio scaricato in baia \Rightarrow veicolo torna in zona check-in
 - veicolo libero e in attesa \Rightarrow pronto a soddisfare nuova richiesta
 - code (eventuali) nei buffer
 - battery-low \Rightarrow va alla stazione di ricarica
 - failure: ostacolo non gestibile, perdita bagaglio, blocco veicolo ecc. \Rightarrow strategie diverse
- **interfaccia grafica per visualizzazione stato del sistema**

Reti di Petri

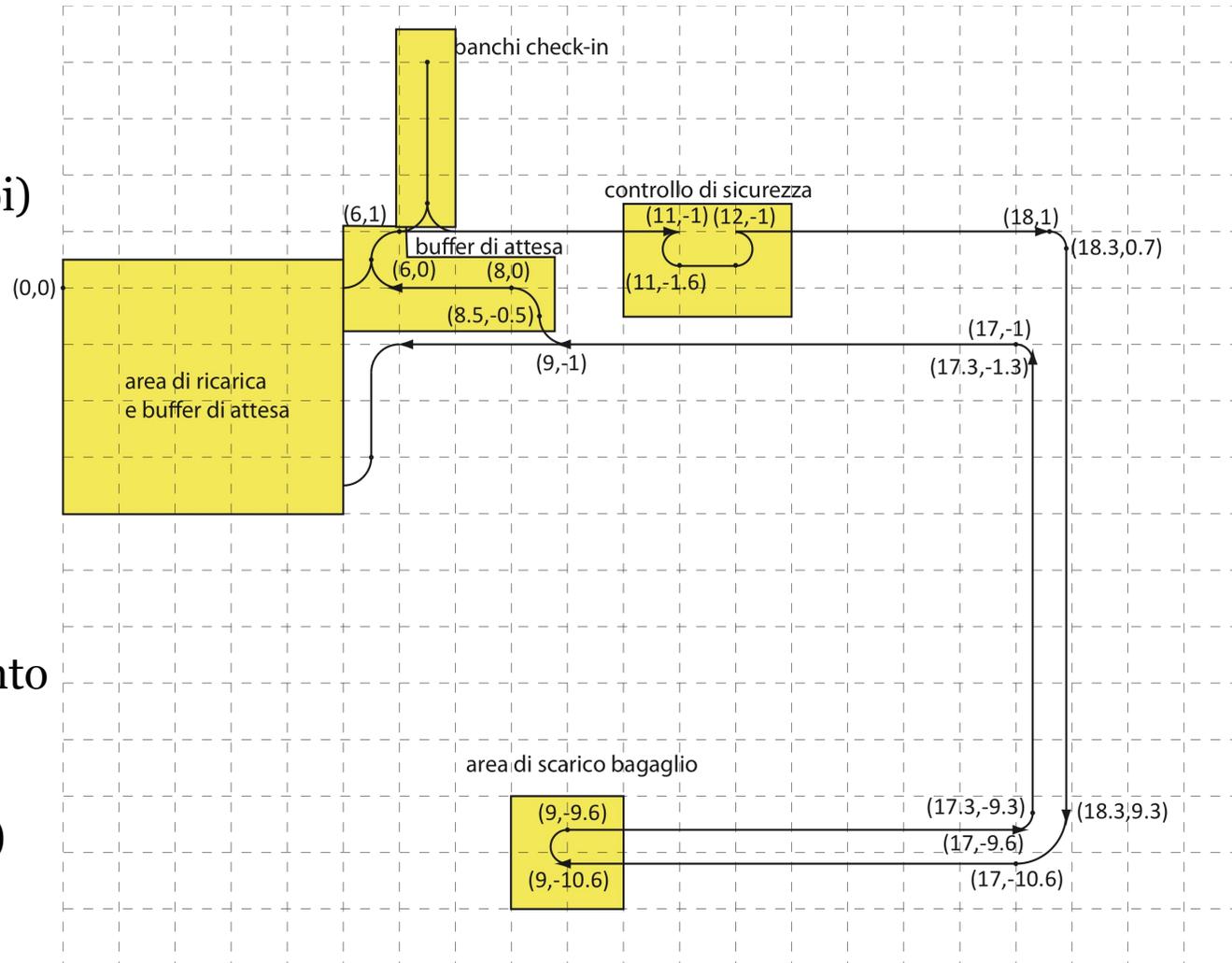
- modellazione e simulazione di sistemi a eventi discreti
 - con **Petri Nets** (PN: posti, transizioni, archi pesati, token, **inibizioni**, struttura modulare)
 - transizioni **temporizzate** (TimedPN, deterministiche/stocastiche)
modellano: tempi di trasferta, controllo rx o ricarica batterie veicolo, intervalli di arrivo clienti ai check-in, MTBF, ...
 - **usi**: throughput sistema (bagagli/h), analisi di code in buffer, deadlock, gestione concorrenze/conflicti, dimensionamento, ...



“firing” di una transizione

Layout

5 zone di lavoro
caratteristiche
(con vincoli e tempi)



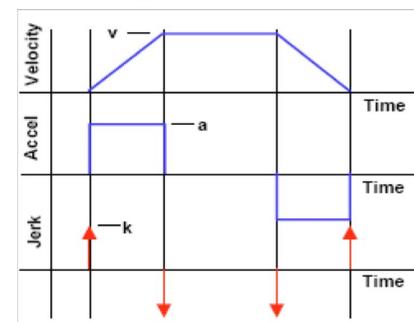
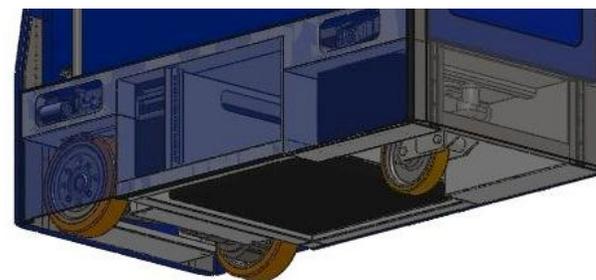
mappa di riferimento
per lunghezze
(tempi di trasferta
dei veicoli robotici)

Veicolo robotico

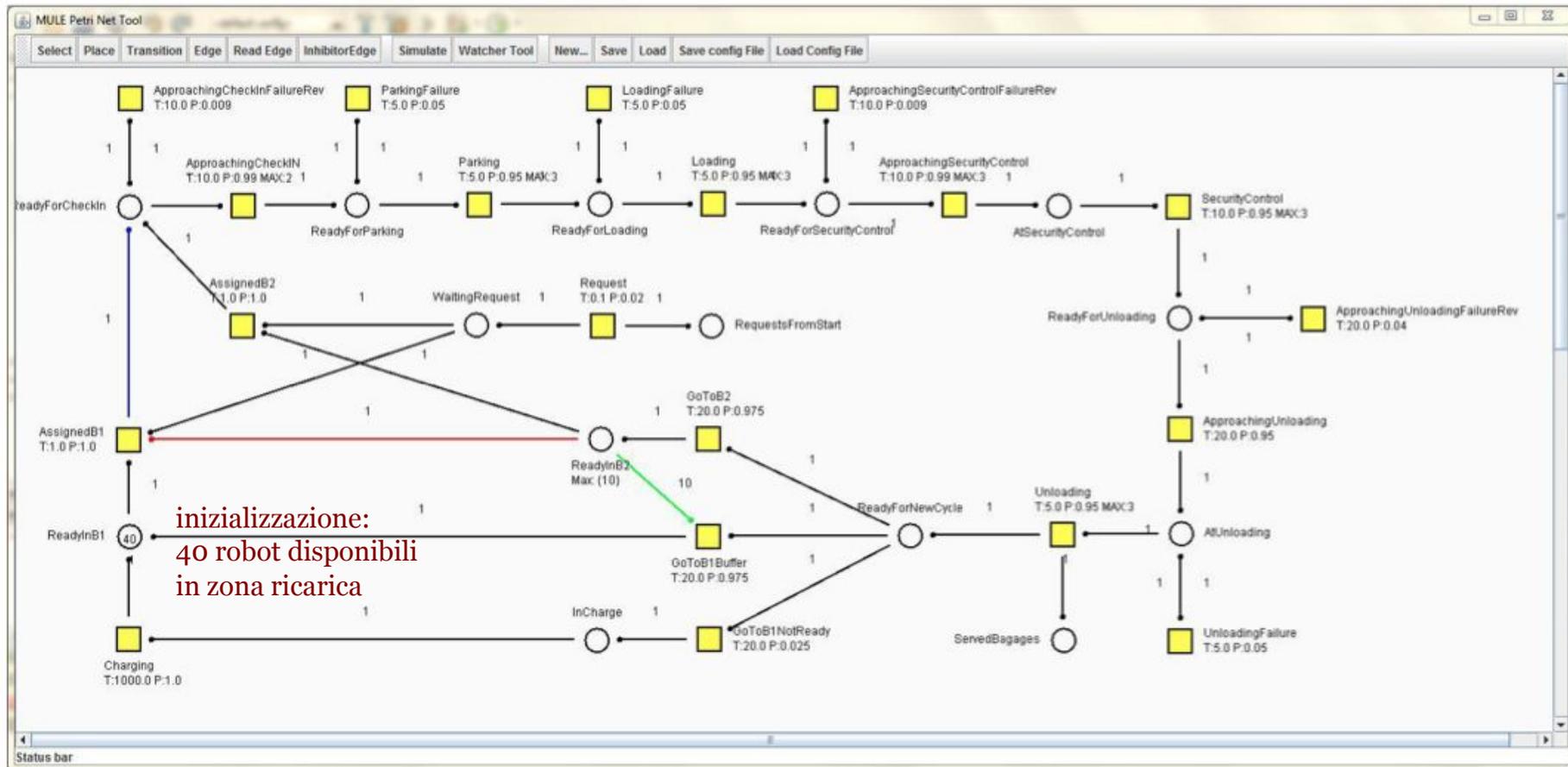
- **specifiche di moto**
 - velocità crociera = 2 m/s
(= 7.2 km/h)
 - accelerazione max = 0.5-1 m/s²
(1.6 m/s² max)
 - tempo di autonomia operativa
 - cinematica tipo **uniciclo**
 - capacità di rotazione sul posto
 - curve a raggio arbitrario percorribili a velocità costante
 - profili orari a velocità trapezoidale per cammini rettilinei rest-to-rest
 - manovre di docking/parcheggio



due ruote azionate
(servo-controllate in
modo differenziale)
+ castor passivo



Il sistema I-Mule



- Tool di modellizzazione e simulazione con TPN (+ estensioni) sviluppato in Java (platform independent), con GUI interattiva in C++

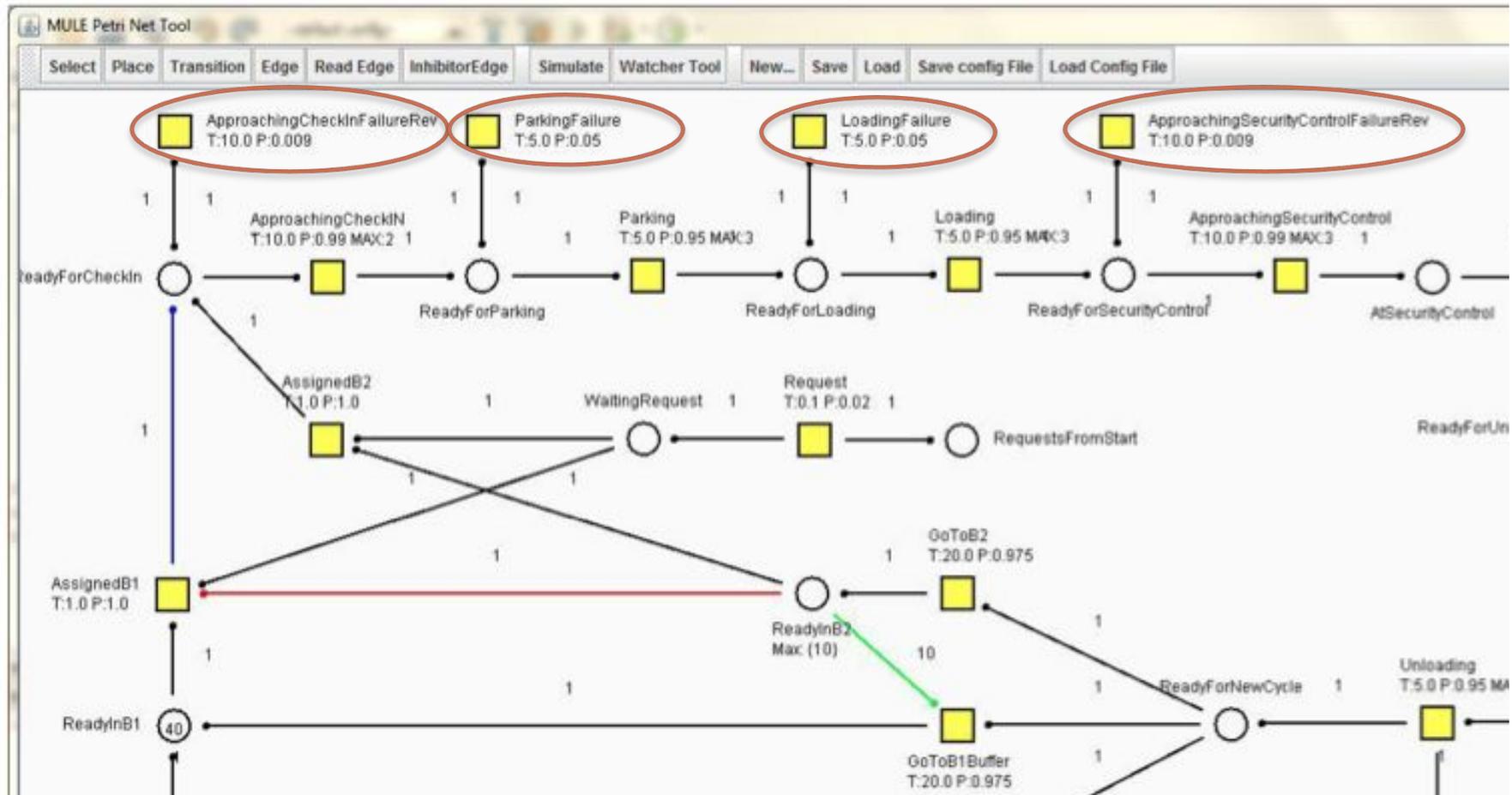
Caratteristiche del tool

- **estensioni delle Petri Nets incluse**
 - archi inibitori
 - disabilitano l'occorrenza di un evento
 - check (in sola lettura) dei token presenti in un posto
 - per procedere con test logici: if, until, ...
 - transizioni temporizzate
 - durata (nominale) degli eventi
 - tempi non deterministici
 - eventi che accadono con diverse distribuzioni di probabilità
 - posti con massima capacità (buffer limitati)
 - transizioni con massima attivazione
 - massimo numero di eventi concorrenti

Alcune scelte operative

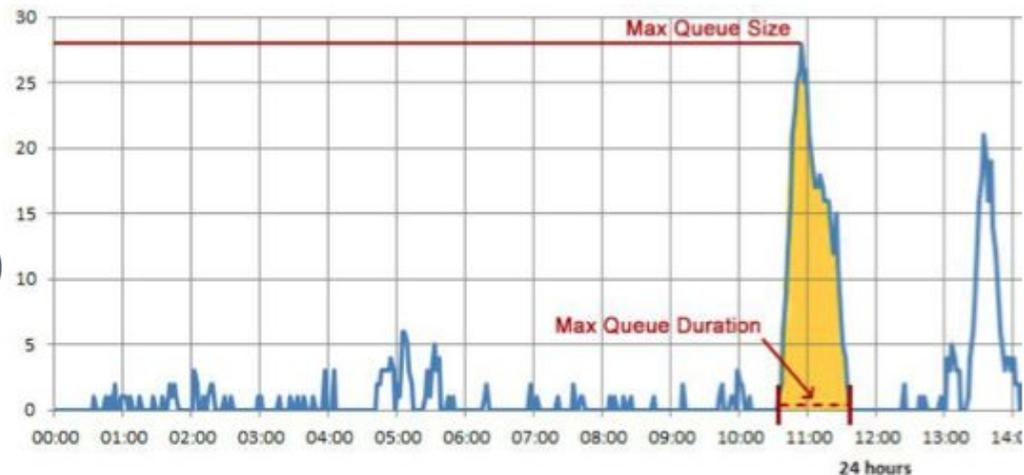
- **capacità**
 - sul #robot: buffer \Rightarrow check-in area ≤ 2 ; check-in \Rightarrow security check ≤ 3
- **tempi**
 - buffer \Rightarrow check-in: 10 sec; parking: 5 sec; loading: 5 sec; security \Rightarrow bay area: 20 sec (in base a velocità robot = 2 m/s); charging: 1000 sec; ...
- **probabilità**
 - uniformi (o esponenziali): buffer \Rightarrow check-in area: 0.95; parking: 0.95; check-in \Rightarrow security: 1; ...
 - il complemento a 1 sono errori recuperabili (“nuovo tentativo”)
- **arrivi clienti**
 - intervalli $t = -\log(1-r)/\lambda$ (tassi di arrivo λ , r uniforme in $[0,1]$)
- **failures (vari malfunzionamenti)**
 - hanno una probabilità di insorgenza, richiedono un tempo di recovery

Alcune failures



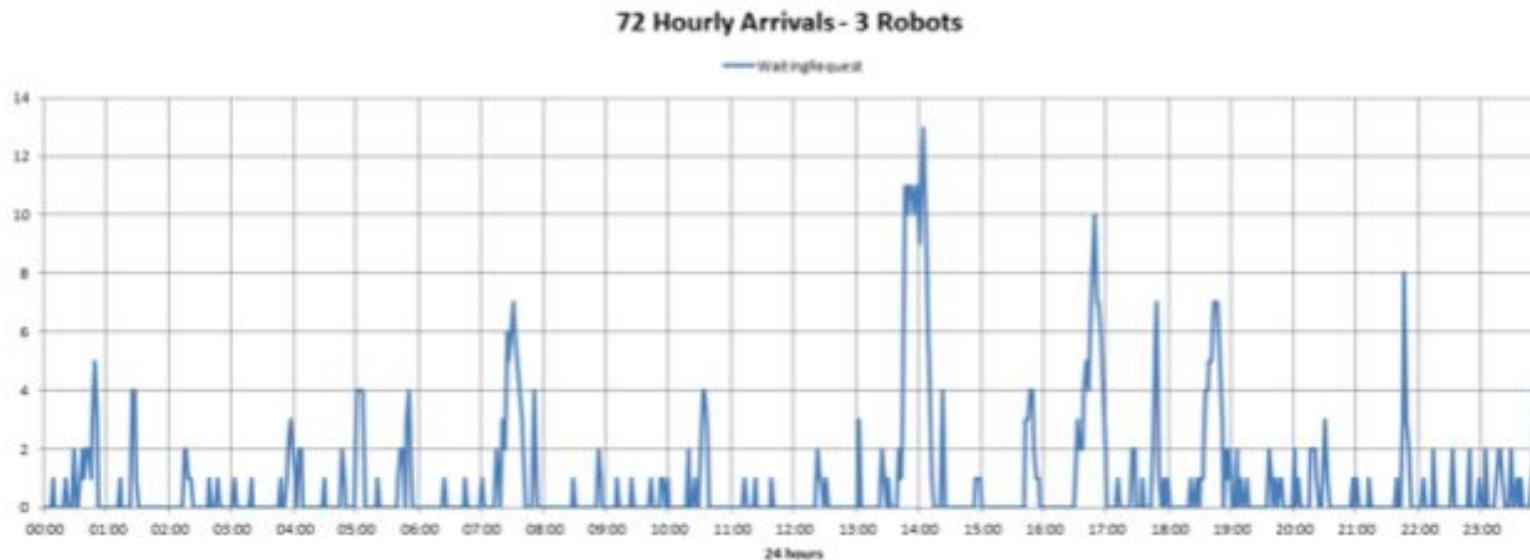
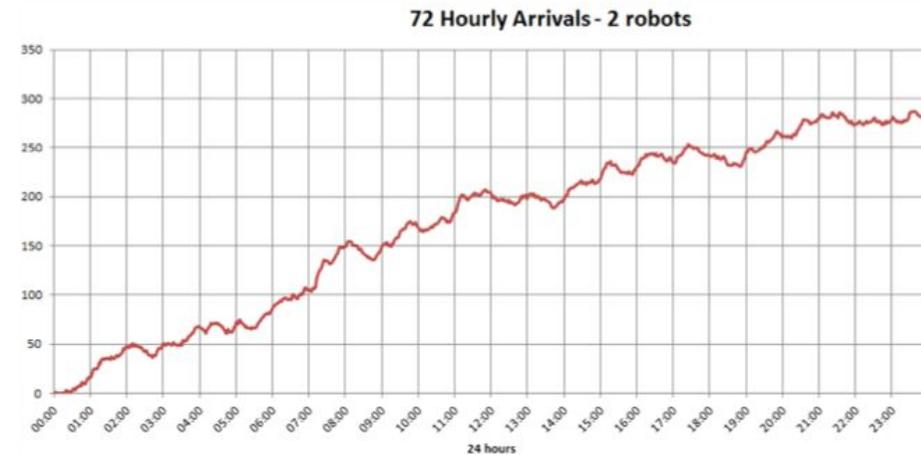
Test effettuati

- tassi di arrivo clienti
 - $\lambda=72/h, 180/h, 720/h$
- numero robot e banchi check-in
 - #robot = 1, 2, 3, ..., 7+; #banchi check-in: 2, anche 3 per 720/h
- 24h di simulazione
 - con sampling 2 ÷ 5 min (+ zoom ogni 10 sec per 2.7h = 100K campioni)
- indici di prestazione (media e varianza su esperimenti ripetuti)
 - throughput (#bagagli/h)
 - #clienti medio in attesa
 - max #clienti in coda
 - max durata coda (clearing)



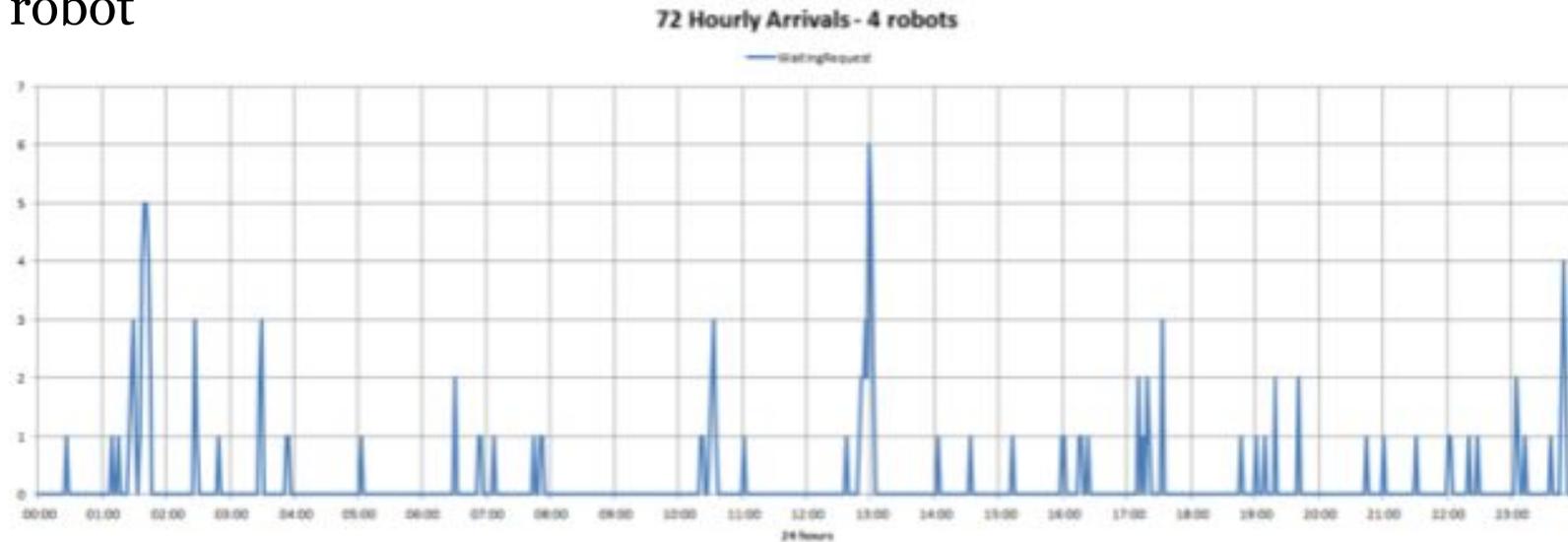
Risultati – 72/h

- 1 robot: 34.8 clienti/h serviti
- 2 robot: 68.4 clienti/h serviti
- 3+ robot: coda **limitata**
 - max coda in 24h = 14 clienti
 - clearing in 52 minuti



Risultati – 72/h

- 4 robot

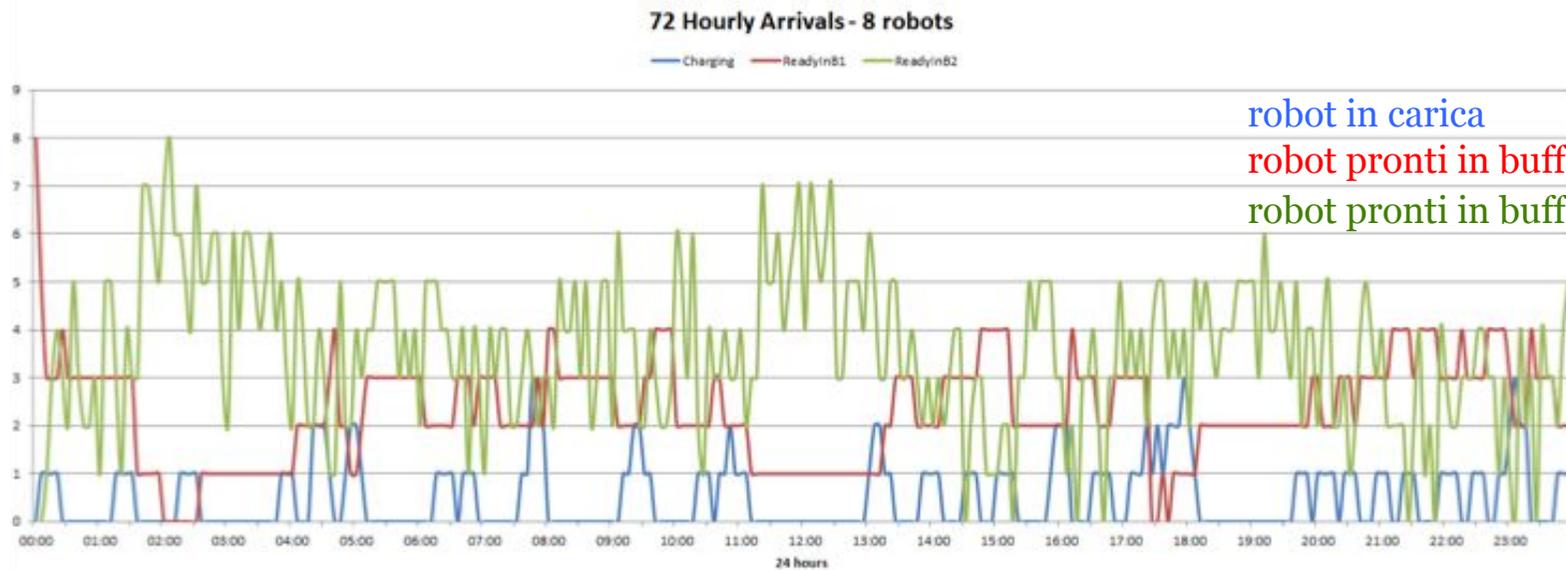


- 5 & 6 robot



Risultati – 72/h

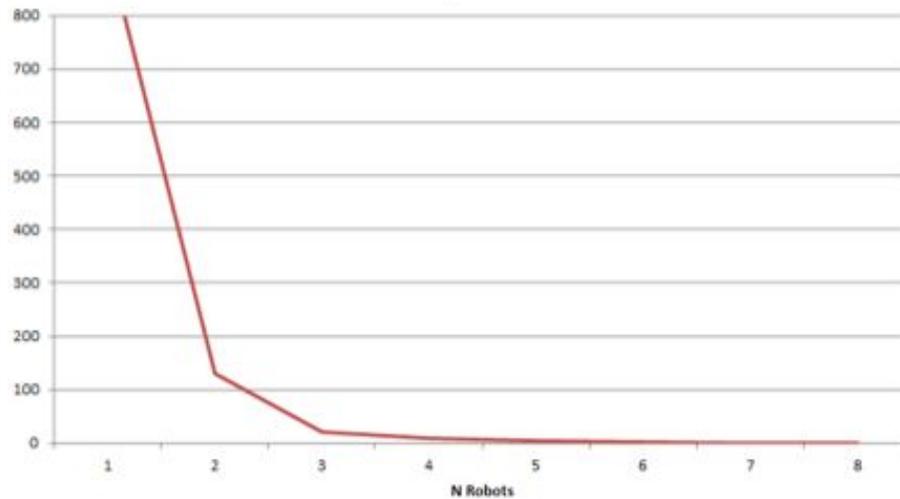
- 7 (o +) robot: caso ideale



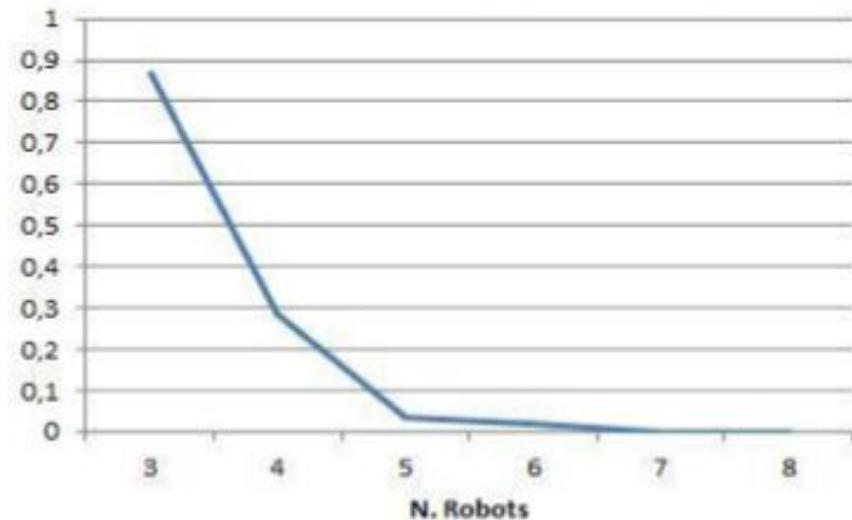
Riassunto – 72/h

- al variare del numero di robot

max lunghezza coda attesa

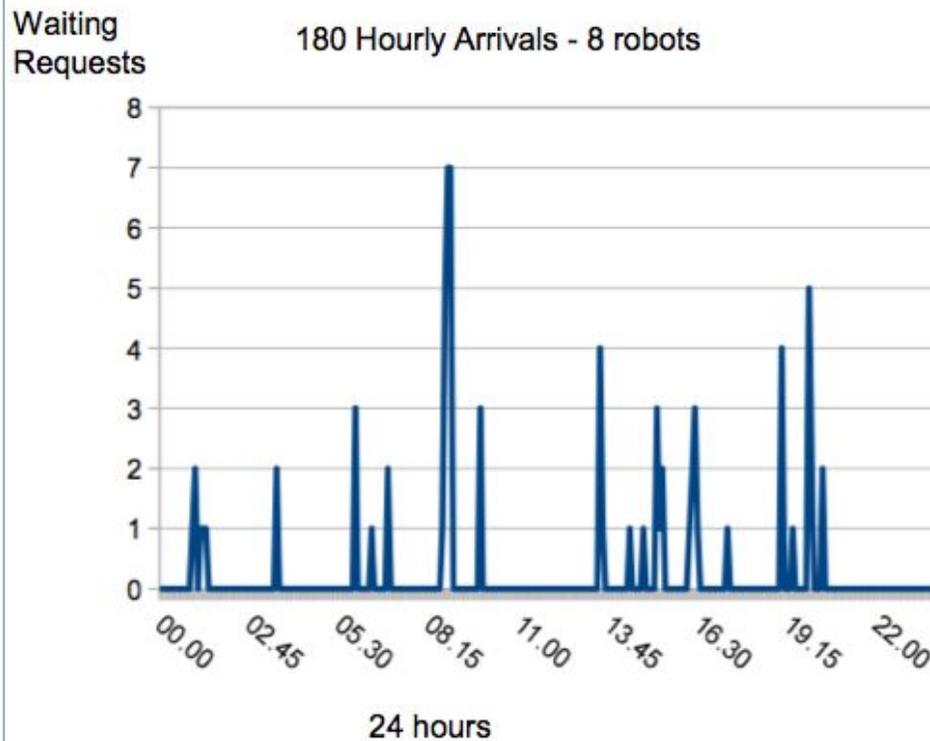


media #clienti in attesa/unità tempo



Risultati – 180/h

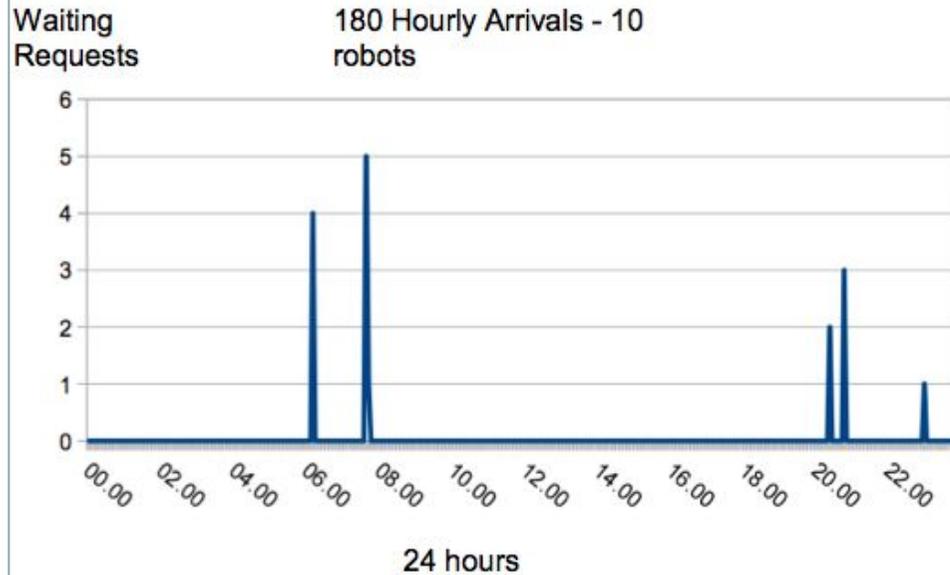
- risultati adeguati con 8 robot, ideali con 10, da 12+ molti robot inutilizzati



8 robots			
	Max Q. Duration	Max Q. Size	Average Q. Size
1	10	7	0,047
2	15	1	0,02
3	20	10	0,08
4	10	3	0,01
5	15	2	0,06
6	15	7	0,12
7	20	6	0,01
8	20	6	0,07
9	10	6	0,08
10	15	9	0,16
Average			
	15	5,7	0,07
Variance			
	16,66	8,46	0,0001

Risultati – 180/h

- risultati ideali con 10, da 12+ molti robot rimangono inutilizzati



10 robots			
	Max Q. Duration	Max Q. Size	Average Q. Size
1	10	7	0,14
2	5	1	0,02
3	10	3	0,08
4	5	3	0,01
5	10	2	0,06
6	10	7	0,12
7	5	1	0,01
8	5	2	0,07
9	15	3	0,08
10	10	9	0,16
Average	8,50	3,80	0,08
Variance	11,38	7,96	0,0001

Risultati – 720/h

	Max Queue Duration	
Robots	2 check-in	3 check-in
25	1252,5	102
30	881,5	32,5
35	409	22,5
40	269,5	12,5

	Max Queue Size	
Robots	2 check-in	3 check-in
25	326,3	64,7
30	168,6	36,9
35	94,7	19,4
40	73,7	15,3

	Average Queue Size	
Robots	2 check-in	3 check-in
25	149,97	10,20
30	64,19	3,11
35	25,71	1,11
40	13,62	0,52

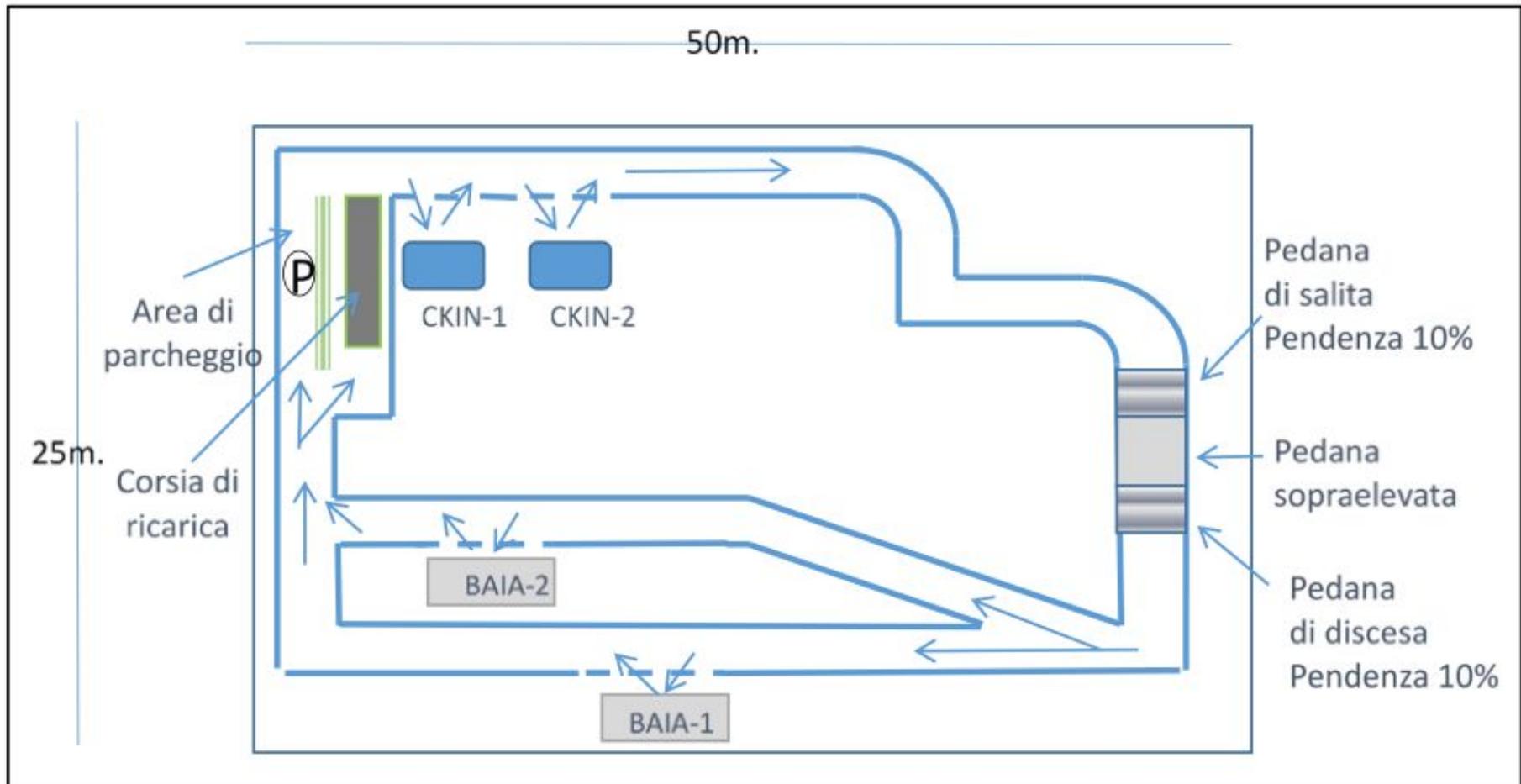
	SB/RFS	
Robots	2 check-in	3 check-in
25	0,9822	0,9987
30	0,9922	0,9992
35	0,9961	0,9992
40	0,9964	0,9992

SB/RFS = ServedBaggages/RequestsFromStart

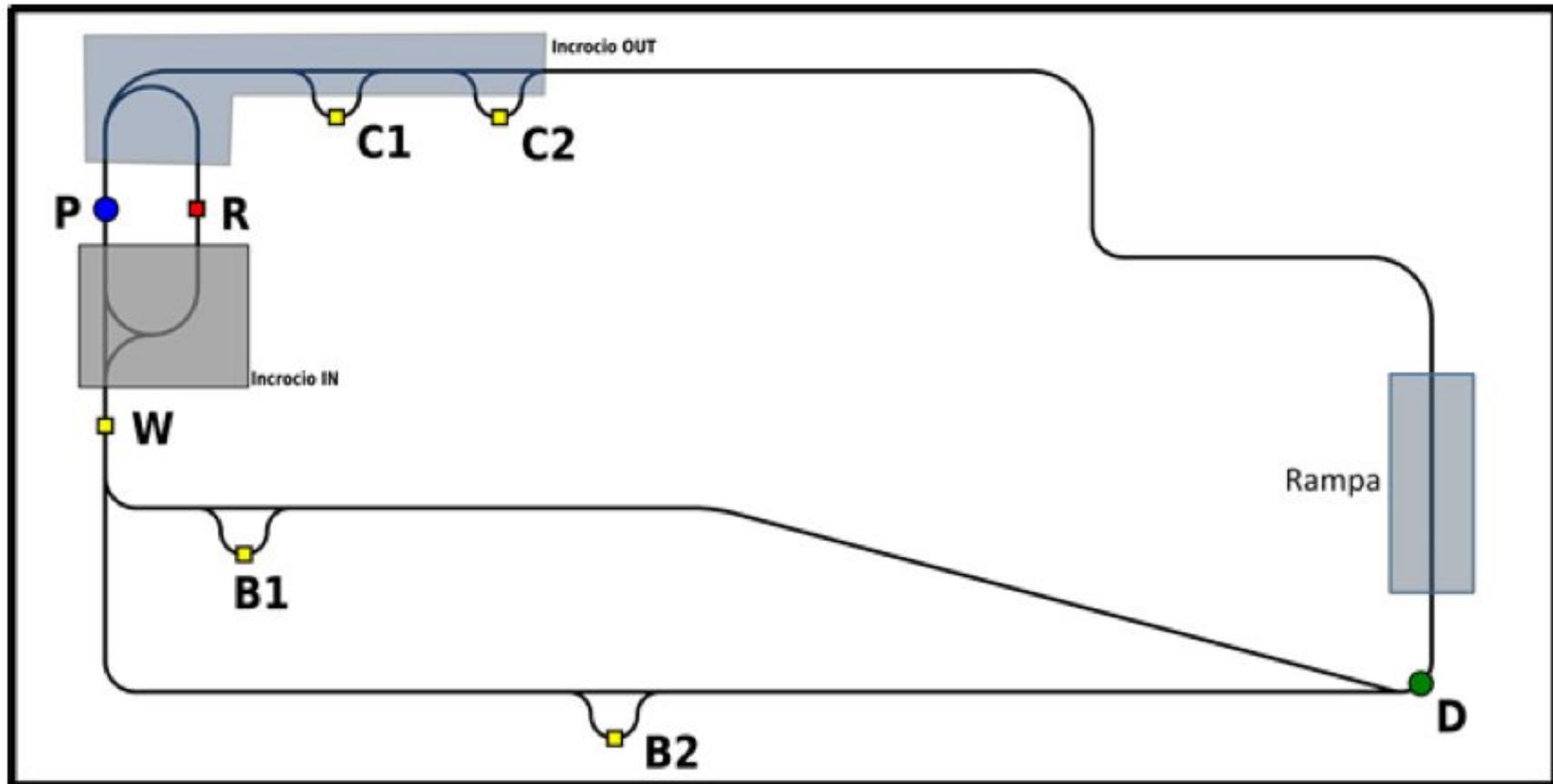
Sviluppi

- studio delle proprietà strutturali
- inclusione di altri aspetti e miglioramenti possibili
 - nuovo layout + introduzione di corsie
 - raffinamento processo di ricarica
 - eventuali logiche alternative di gestione
 - verifica RX direttamente al check-in
- interfaccia con visualizzazione 2D/3D del sistema
- controllo del moto e navigazione
 - uso di sistema di visione esterno nelle zone di manovra
 - localizzazione dei robot con EKF (beacon fissi + odometria)

Nuovo layout



Nuovo layout semplificato



Simulazione 3D della navigazione

