



TC 2021: Mixed-Team Protocol

Rob Burgers (VDL RobotSports)

Jan Feitsma (ASML Falcons)

Introduction



- Original mission: Robots to beat humans in soccer by 2050
- Teams size increases from 5 to 11 robots
- Participation in MSL
- Specialization
- Practical experience and demonstration



https://youtu.be/s9gCgl_rrD8



Mixed-Team Protocol

Mixed Team Protocol



FALCONS
supported by ASML



Design overview



- upgrade of 2014 protocol
 - make it more scalable, extend with Role Allocation
- use RtDB+comm underwater, to create and communicate packets
 - MSL-standard technology
 - Includes refbox listener
- provide API + library and propose as new MSL-standard
 - for easy integration in team SW stack (C++):
 - set-, get- and tick- calls from teamplay and worldmodel components
 - (both Falcons and VDL RobotSports use C++ and RtDB)

Challenge: Role Allocation



- given a set of allowed roles, determine which robot takes which role

Role	Id	Min	Max	Semantics
undefined	0	0	0	Not yet decided. Should never happen after initialization.
goalkeeper	1	1	1	There can only be one, obviously.
attacker-main	2	1	1	Take kickoff, penalty, set-piece. The player with the best aim.
attacker-assist	3	0	1	Play forward, towards target goal. During a set-piece, reposition close to help attacker-main.
attacker-generic	4	0	n/a	Play offensively. (Scaling beyond n=5.)
defender-main	5	1	1	Play defensively.
defender-generic	6	0	n/a	Play defensively. (Scaling beyond n=5.)
disabled-out	7	0	n/a	Moving out of the field, either at substitution or end of half.
disabled-in	8	0	n/a	Moving into the field, either at substitution or start of half.

- design choices:
 - leader (s)election
 - leader calculates roles, taking preferences into account
 - optionally, preference can be set by each robot
 - example: goalkeeper (driven by robot hardware capability)
 - example: attacker-main (driven by proximity to ball during set-piece)
 - algorithm: standard bi-partite assignment problem (Hungarian method, Kuhn/Munkres)
 - alternative: linear programming

Challenge: WorldModel



- example: responsibility of filtering teammates from obstacles is left for client
- data is supposed to be interpreted as 'raw' vision data and treated accordingly by each team's WorldModel implementation
 - (this also was a design choice from 2014 baseline)
- alternative: let Mixed-Team Protocol take this responsibility?
⇒ *why not standardize WorldModel in MTP?*
 - benefit: prevent highly similar implementations in each team SW stack
 - might be useful to new teams especially
 - possible drawback: implementation not customizable, too simplistic
 - idea: plugin software architecture?

Challenge: include humans



- it is easy to 'spooF' a robot in MTP:
 - just send static packets
 - this is useful to play-test a single robot in a match-like situation without having to physically put extra robots on the field
 - also useful for inserting a **human** player!
- these packets can be generated:
 - either by an external system
 - or by a robot which has live human detection capability
 - (future challenge: detect shirt color, assign to correct mixed-team)

Demonstration #2



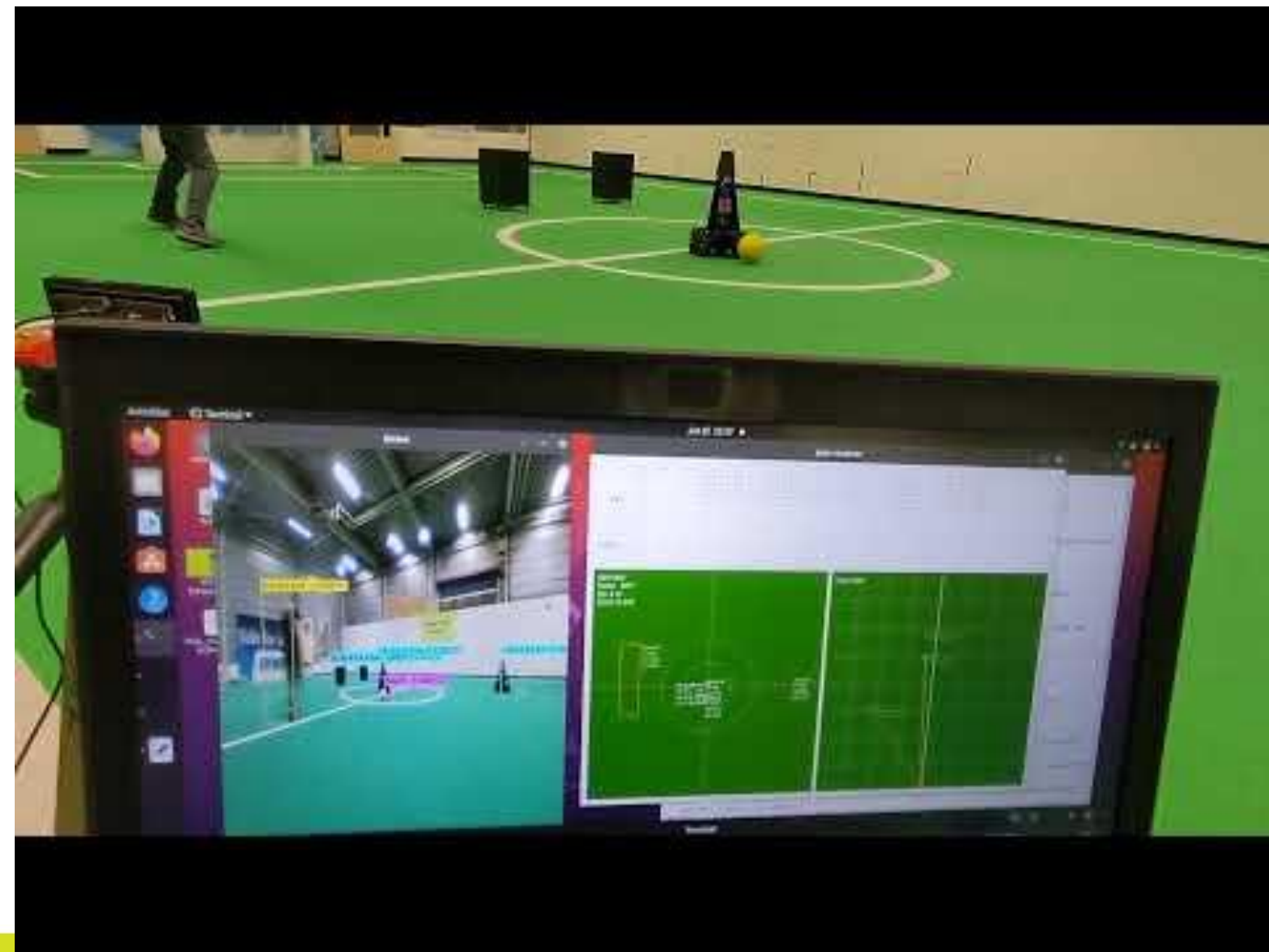
robot passing the ball back and forth with human player

setup:

- 2 robots on field (attacker-main and defender-main)
- spoof goalkeeper statically so robots do not get that role
- external system: static camera mount + Genius Widecam
- human MTP client
 - run on base station laptop, claim attacker-assist role
 - connect with camera
 - use Machine Learning YOLO system to produce relative coordinates
 - human and ball
 - (4Hz with some latency -- good enough)
 - calculate field coordinates using static camera location
 - (accuracy of coordinates: <0.5m with little calibration effort -- good enough)



FALCONS
supported by ASML



<https://youtu.be/vcli9-ByNmY>



RtDB

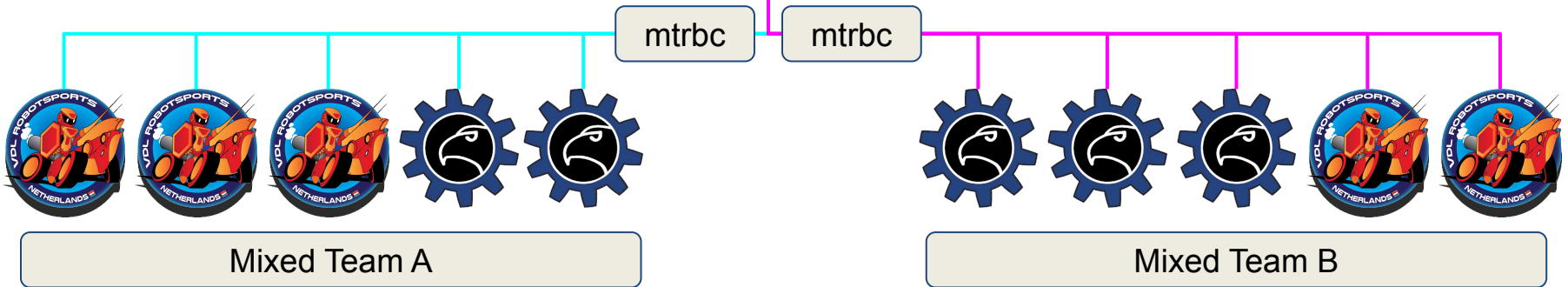
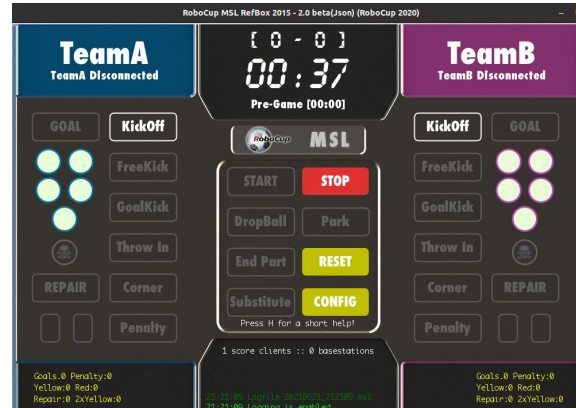
Mixed Team with RTDB



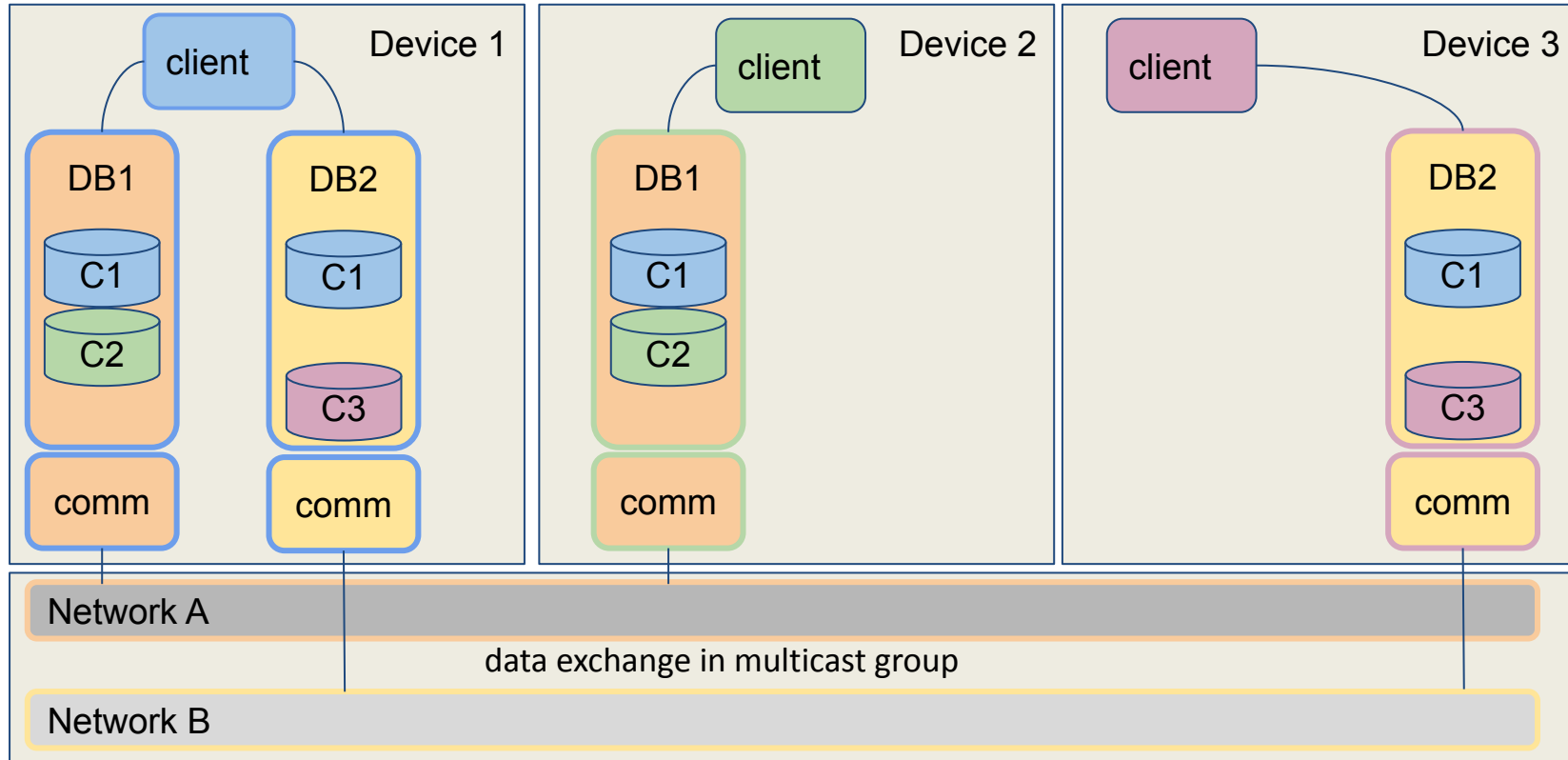
What does it mean to compose a mixed team with RTDB?

- Ease for robots to participate in a mixed team
- Ease for robots to switch between teams
- Unique identification of robots
- Separation of teams on the field

RTDB Network Architecture



RtDB Network detailed





Conclusion

More info, next steps



- see ***MixedTeamProtocol*** repository in MSL github project:
<https://github.com/RoboCup-MSL/MixedTeamProtocol>
- contributions / feedback is welcome!
- future work:
 - worldModel data sharing / fusing?
 - support teams integrating the API + library + tools?
 - standardize ***intentions***?
(example use case: prevent robots from driving away, or in the way when a pass/shot is about to occur)
- ⇒ let's discuss upcoming MSL workshop

Questions?



FALCONS
supported by **ASML**