Outline

1. What is tf2
   Implementation
   Early Adopter of tf2

2. Looking Forward
   Outstanding Work
What is tf2

A refinement of tf

- Cleaner dependencies
  - Templated API
  - Separation of ROS and non ROS components
- First class support for python at the ROS level
- Remote querying mechanism not requiring full tf stream
- Support for static transforms
Implementation

- Templated TransformData method
- Templated conversion methods
- Internalized transform math
- Cleanly separated ROS and non-ROS components
- Action based remote query API
- Separate /tf_static topic
- Removed support for tf namespacing (tf_prefix)
Data Conversions

If all messages define toMsg and fromMsg static methods.

```cpp
#include "tf2_bullet/tf2_bullet.h"
#include "tf2_geometry_msgs/tf2_geometry_msgs.h"
#include "tf2_kdl/tf2_kdl.h"

tf2::Stamped<btVector3> b(btVector3(1,2,3),
    ros::Time(), "my_frame");

google_msgs::Vector3Stamped m;
tf2::convert(b, m);

tf2::Stamped<KDL::Vector> k;
tf2::convert(b, k);
```
Transform Any Datatype

If a template specialization for `doTransform<T>` exists. You can simply use the following:

```cpp
T& transform(T& in,
    string& target_frame, Time& target_time,
    string& fixed_frame, Duration timeout=0.0)
```

Combine templated API and conversion:

```cpp
B& transform(const A& in, B& out,
    string& target_frame, Duration timeout=0.0)
```
Static Transforms

- Latched topic /tf\_static
- Published once
- Assumed to be not changing therefore no time history stored.
Remote Query

Provide an action based remote query capability, giving client/server model. This has benefits including:

- Allow one process with a longer history
- Background monitoring scripts do not require a `/tf` subscription
- Enable querying of tf data from offboard the robot over low bandwidth links (aka wireless)
pr2_plugs

A test deployment to verify things

- Verified correctness
- Verified robustness
Robot Web Tools

Using transforms on the web

Major considerations

• Bandwidth
• Low level API access
Outline

1. What is tf2
   - Implementation
   - Early Adopter of tf2

2. Looking Forward
   - Outstanding Work
Dealing with partitioned networks

Major considerations

- Bandwidth between robots
- Disambiguating similar robots
Proxying tf data

What parameters are important?

- What are the frames of interest?
- How are frames on one robot connected to the other robots?
- What update rate is required?
- How to balance bandwidth vs update rate?
- What latency is tolerable?
Multi Robot

[Images of multi-robot systems]
Multi Robot

Robot 1
Topics: /tf & /scan
Coordinate Frame: base_link

Robot 2
Topics: /tf & /scan
Coordinate Frame: base_link

How do you merge them?

- Forward /tf from one robot to another, rewrite frame_ids (base_link becomes robot1_base_link)
- Forward any data and rewrite specified fields which contains frame_ids (requires semantic interpretation)
Questions?

For full documentation see:

http://www.ros.org/wiki/tf2