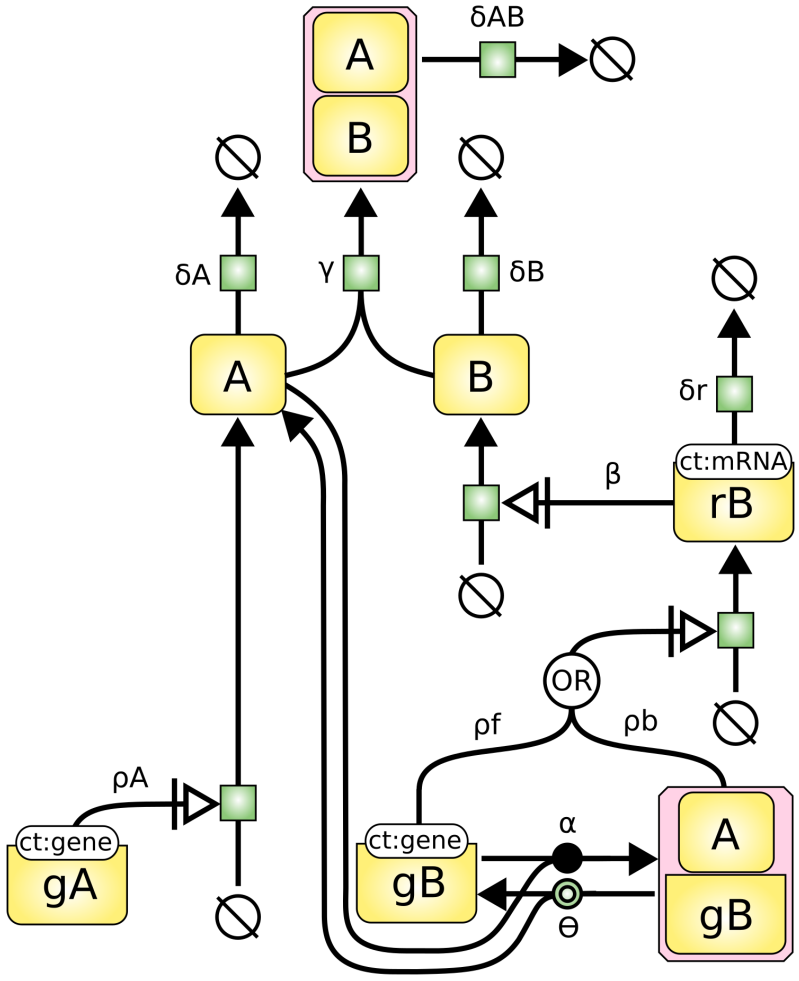
# Compositional Design and Tasking of Networks

John Baez, U. C. Riverside

NIST, 15 March 2018

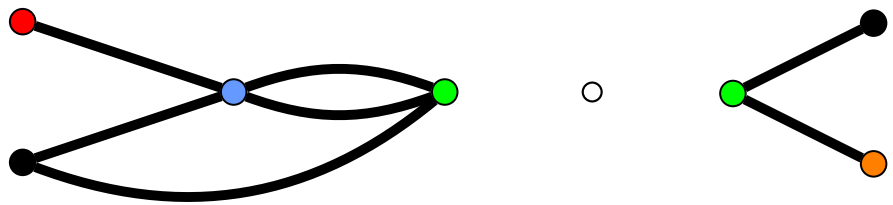
Complex systems are often studied and designed using *networks*:



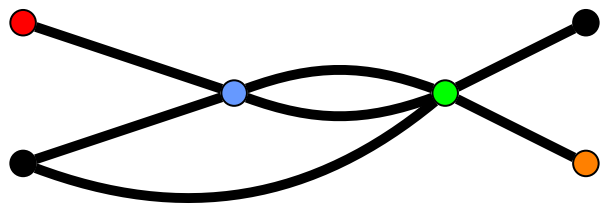
Categories and operads are uniquely suited to describing networks and ways of composing them.

Categories and operads are uniquely suited to describing networks and ways of composing them.

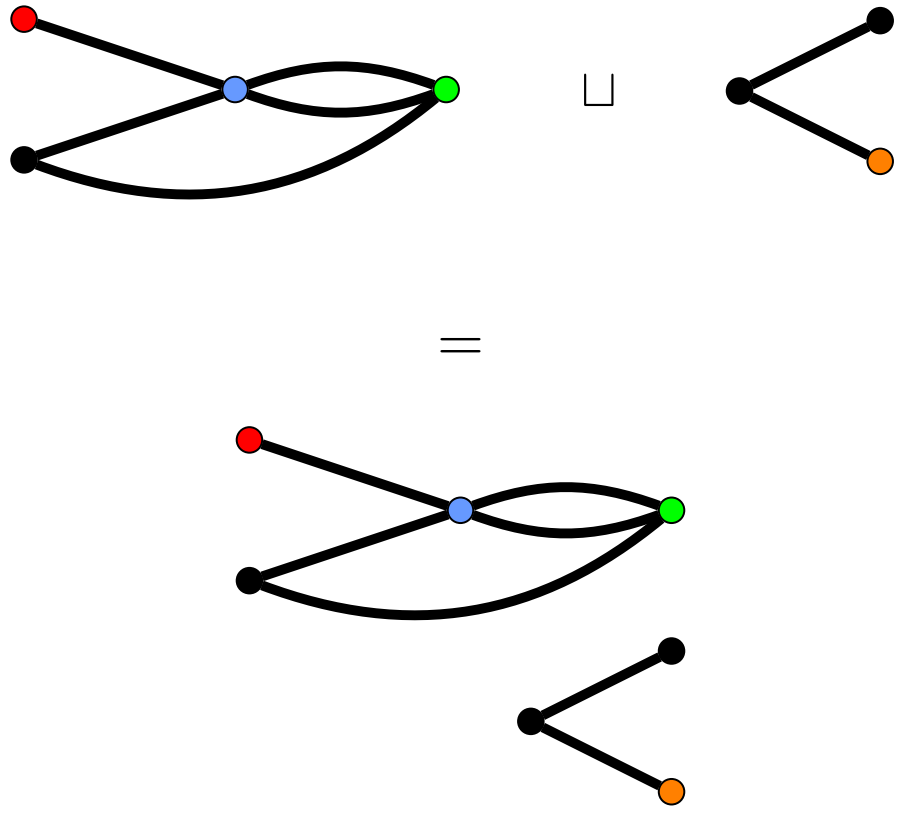
Networks can be connected ‘end to end’, in series:



=



Networks can also be set ‘side by side’, in parallel:



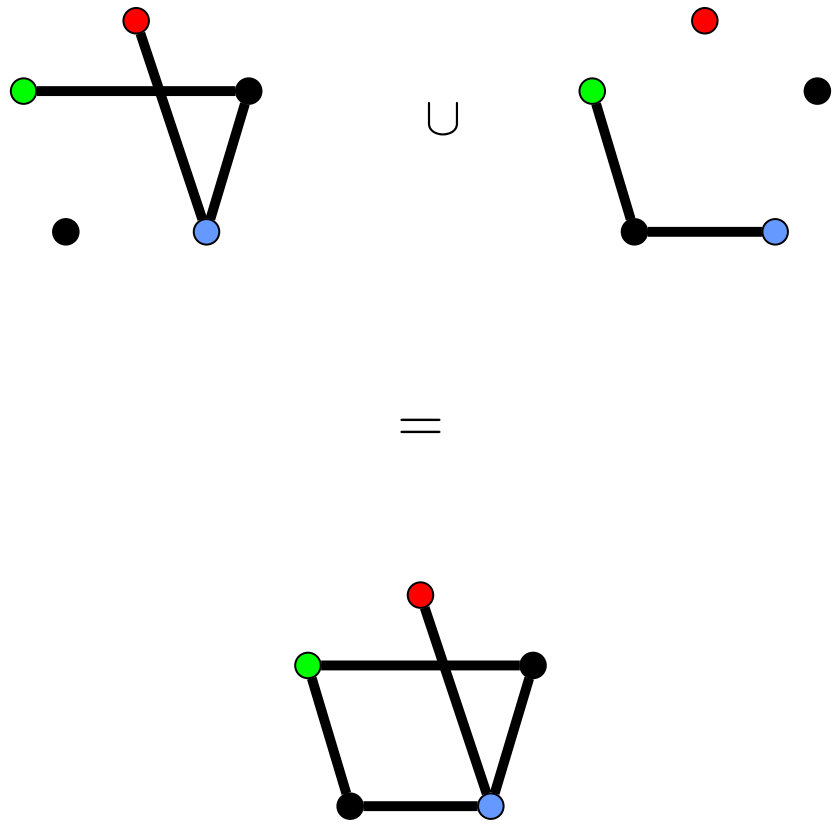
Mathematicians now understand these forms of composition very well using categories with extra structure: monoidal categories, dagger-compact categories, hypergraph categories... all these are algebras of various operads.

Mathematicians now understand these forms of composition very well using categories with extra structure: monoidal categories, dagger-compact categories, hypergraph categories... all these are algebras of various operads.

In the Complex Adaptive System Composition and Design Environment project, I'm working with Metron Scientific Solutions to build networks using another form of composition: *overlaying*.

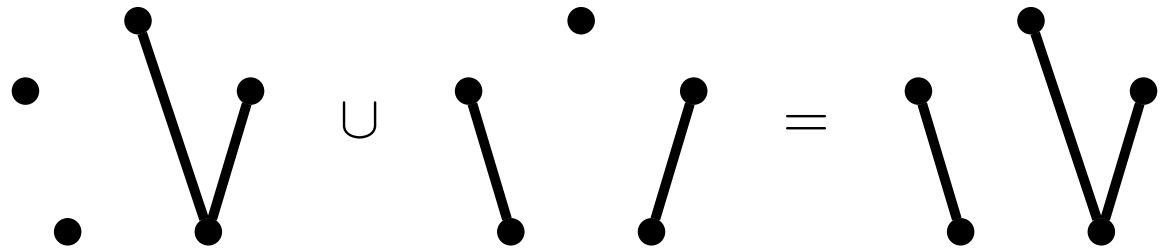
The hope is that new ideas from category theory can give new ways to design complex systems.

Sometimes we can ‘overlay’ two networks with the same vertices. For example,



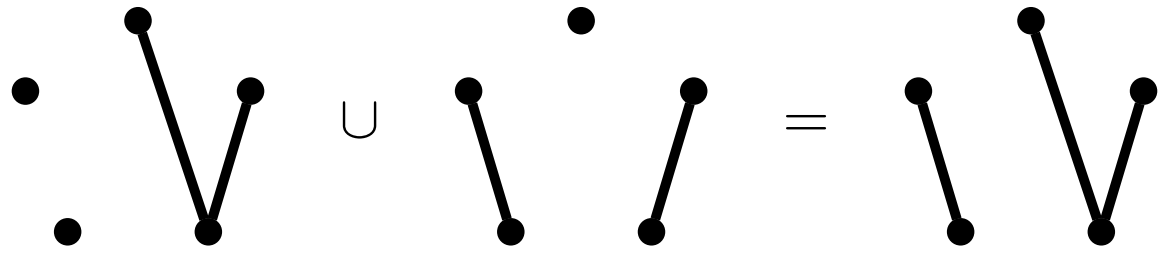
There are many kinds of networks, and many possible rules for overlaying them.

Simple graphs:

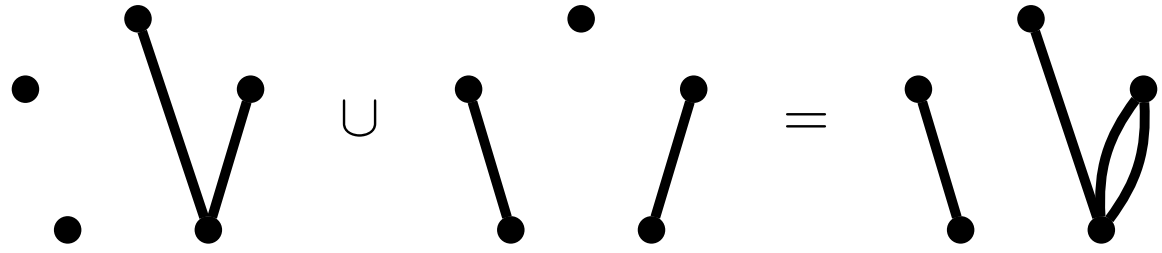


There are many kinds of networks, and many possible rules for overlaying them.

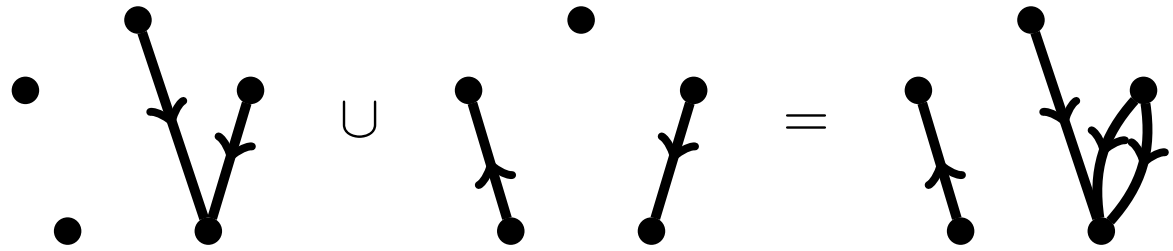
Simple graphs:



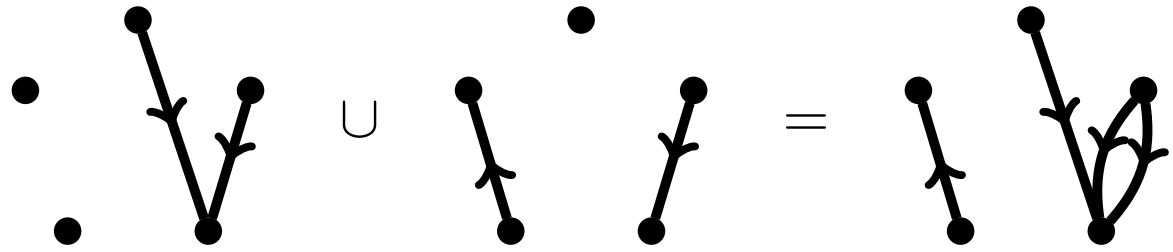
Multigraphs:



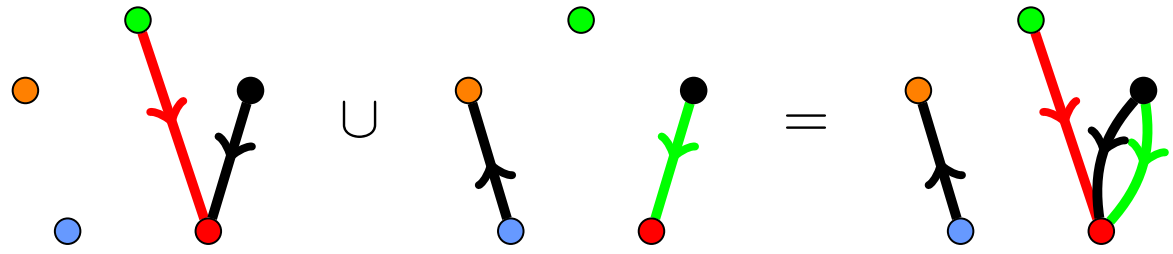
Directed multigraphs:



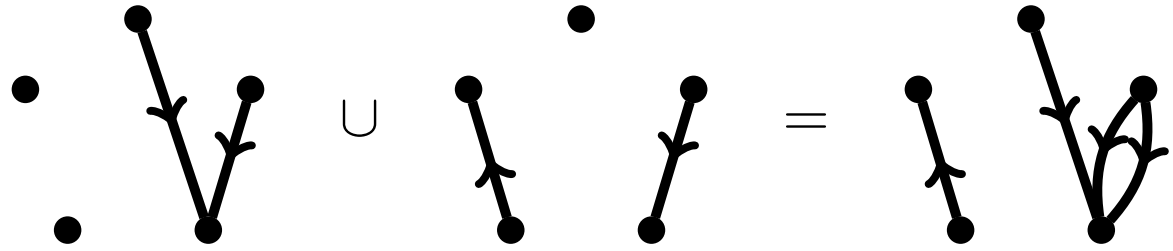
Directed multigraphs:



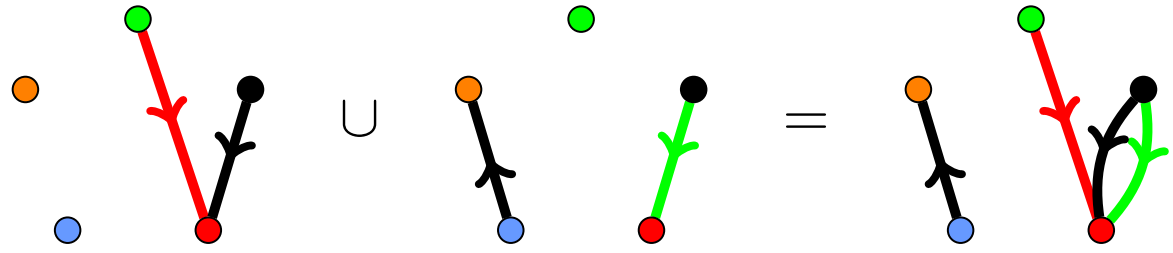
Directed multigraphs with colored edges and nodes:



Directed multigraphs:



Directed multigraphs with colored edges and nodes:



... and so on!

We should handle all these kinds of networks in a unified way.

Define a **network model** to be a symmetric lax monoidal functor

$$F : S(C) \to \mathrm{Mon}$$

where $S(C)$ is the free symmetric monoidal category on some set $C$ of **vertex colors**, and Mon is the category of monoids.

Define a **network model** to be a symmetric lax monoidal functor

$$F \colon S(C) \to \mathrm{Mon}$$

where $S(C)$ is the free symmetric monoidal category on some set $C$ of **vertex colors**, and Mon is the category of monoids.

For example: if $C = 1$, $S(C)$ is the groupoid of finite sets and bijections. For any finite set $X$, $F(X)$ is the set of networks having $X$ as their set of vertices. This is a monoid, so we can **overlay** networks:

$$\cup \colon F(X) \times F(X) \to F(X)$$

Since $F$ is lax monoidal we can also set networks 'side by side':

$$\sqcup \colon F(X) \times F(Y) \to F(X + Y)$$

For any network model $F$ there is an operad $O_F$ whose operations are ways to assemble networks of type $F$.

For any network model $F$ there is an operad $O_F$ whose operations are ways to assemble networks of type $F$. These operations include overlaying networks, setting networks side by side, and thus also attaching networks end to end.

For any network model $F$ there is an operad $O_F$ whose operations are ways to assemble networks of type $F$. These operations include overlaying networks, setting networks side by side, and thus also attaching networks end to end.

The operad $O_F$ will have many algebras. These describe different kinds of systems that can be modeled using networks of type $F$.

For any network model $F$ there is an operad $O_F$ whose operations are ways to assemble networks of type $F$. These operations include overlaying networks, setting networks side by side, and thus also attaching networks end to end.

The operad $O_F$ will have many algebras. These describe different kinds of systems that can be modeled using networks of type $F$. These algebras can involve:

For any network model $F$ there is an operad $O_F$ whose operations are ways to assemble networks of type $F$. These operations include overlaying networks, setting networks side by side, and thus also attaching networks end to end.

The operad $O_F$ will have many algebras. These describe different kinds of systems that can be modeled using networks of type $F$. These algebras can involve:

- vertex attributes (e.g. location, possibly time-dependent)

For any network model $F$ there is an operad $O_F$ whose operations are ways to assemble networks of type $F$. These operations include overlaying networks, setting networks side by side, and thus also attaching networks end to end.

The operad $O_F$ will have many algebras. These describe different kinds of systems that can be modeled using networks of type $F$. These algebras can involve:

- vertex attributes (e.g. location, possibly time-dependent)

- constraints on edges (e.g. range constraints)

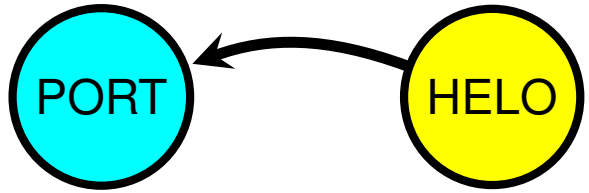
For any network model $F$ there is an operad $O_F$ whose operations are ways to assemble networks of type $F$. These operations include overlaying networks, setting networks side by side, and thus also attaching networks end to end.

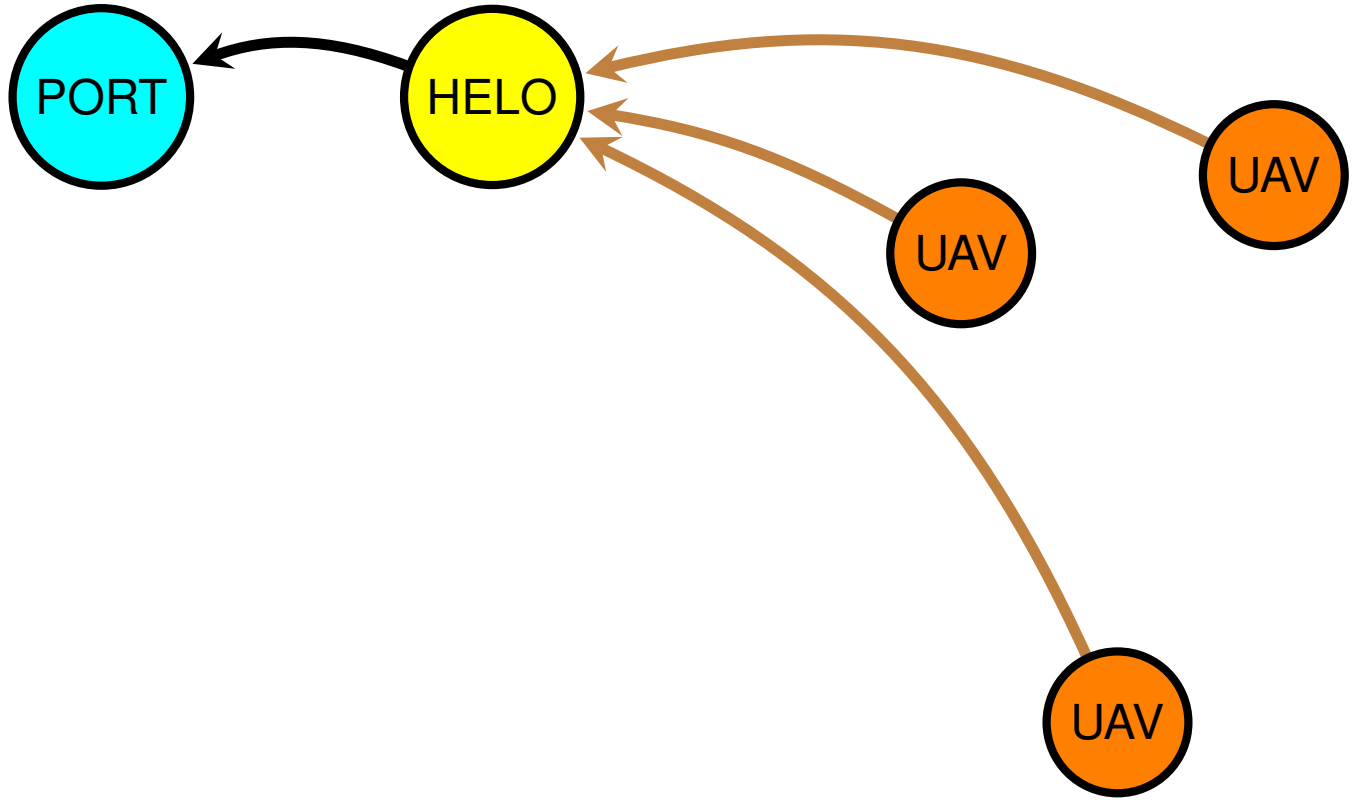
The operad $O_F$ will have many algebras. These describe different kinds of systems that can be modeled using networks of type $F$. These algebras can involve:

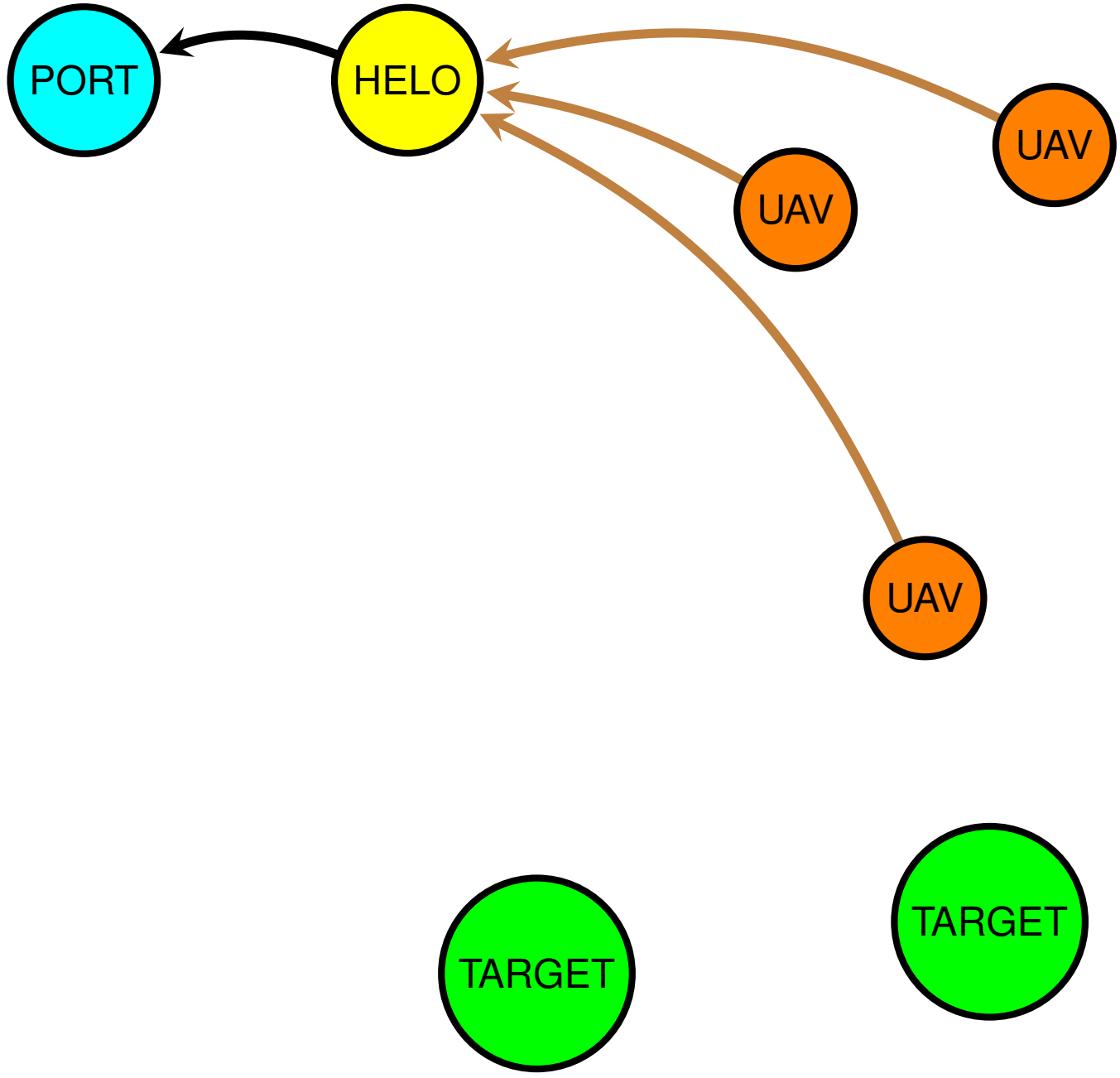
- vertex attributes (e.g. location, possibly time-dependent)

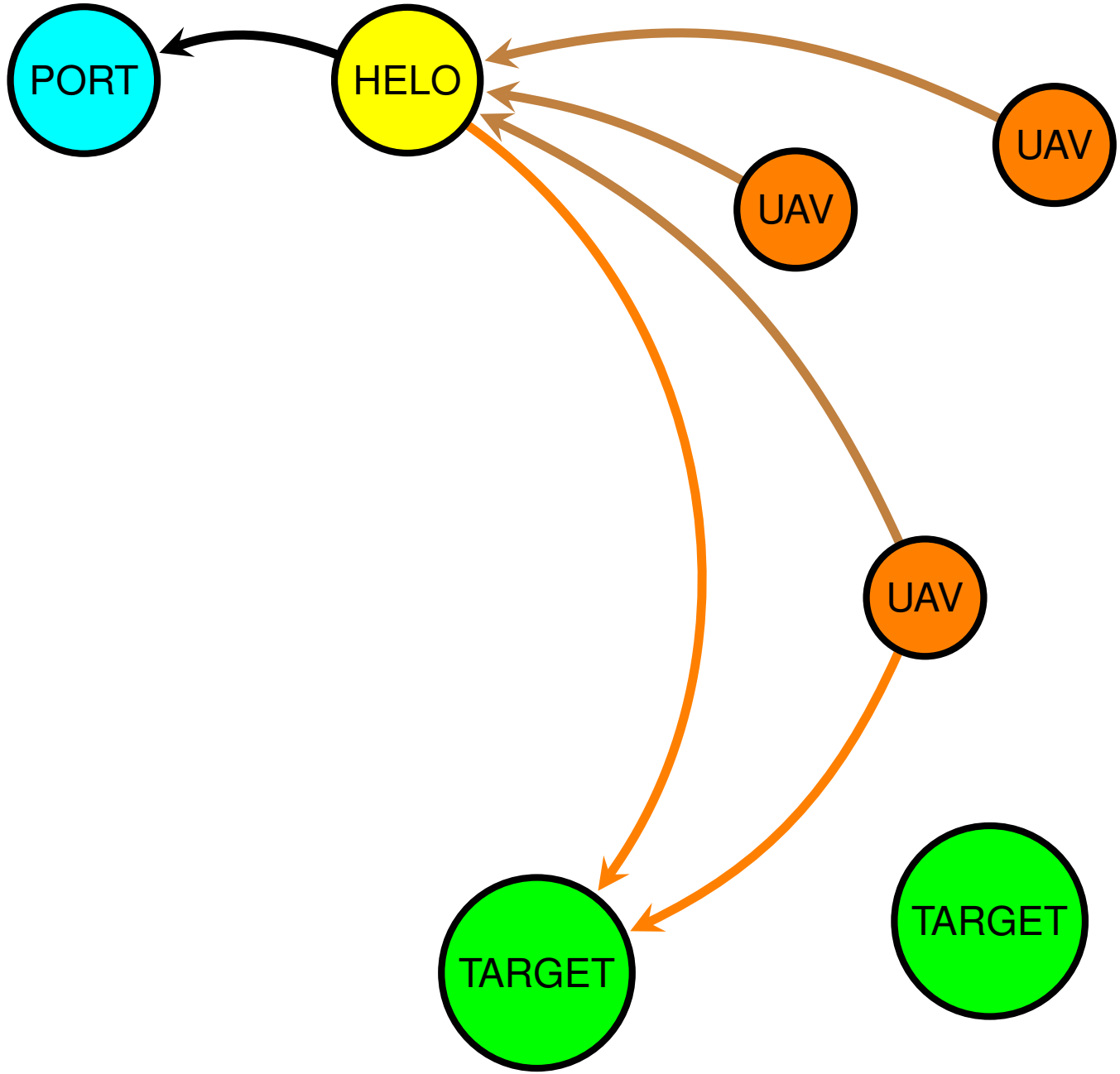
- constraints on edges (e.g. range constraints)

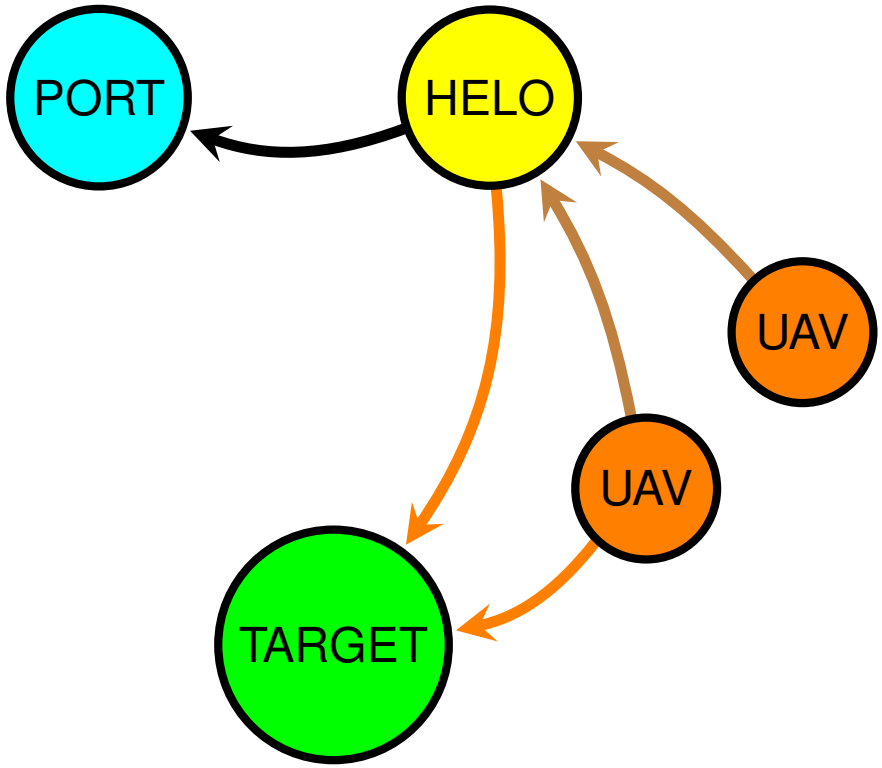
and much more. *Maps between algebras* let us design and task systems incrementally, adding a bit of detail at a time.

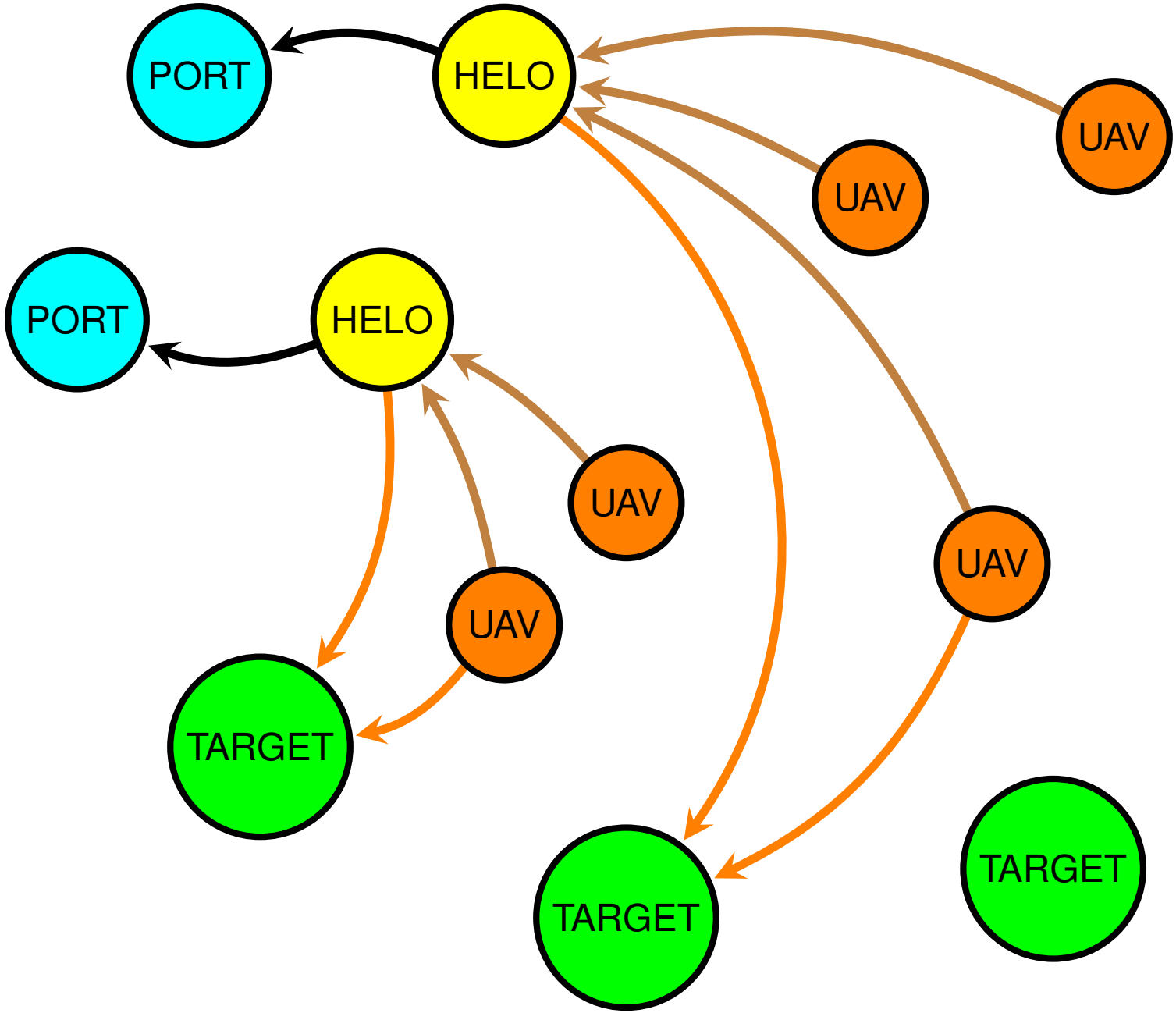

PORT
HELO

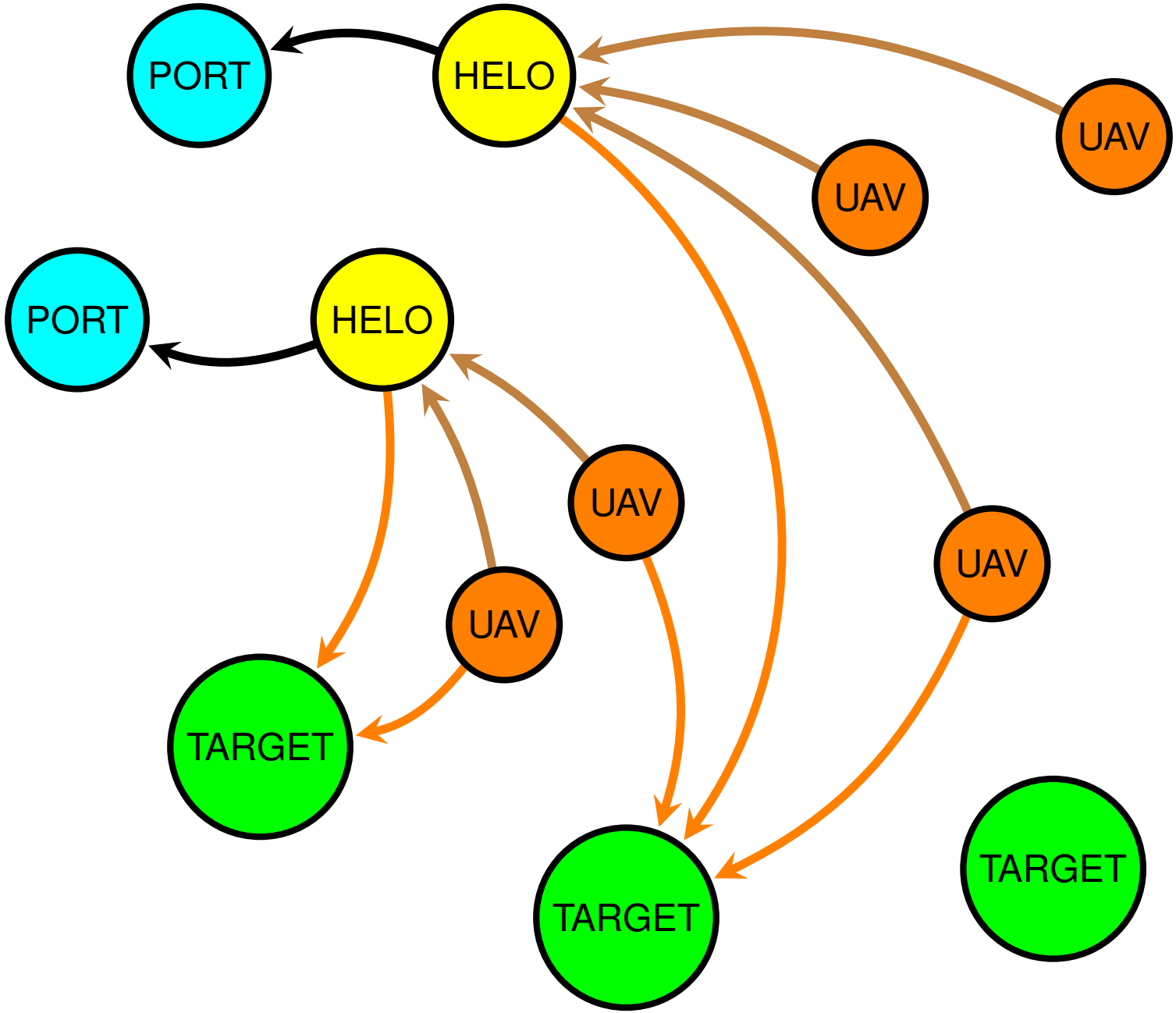

PORT
HELO
UAV
UAV
UAV

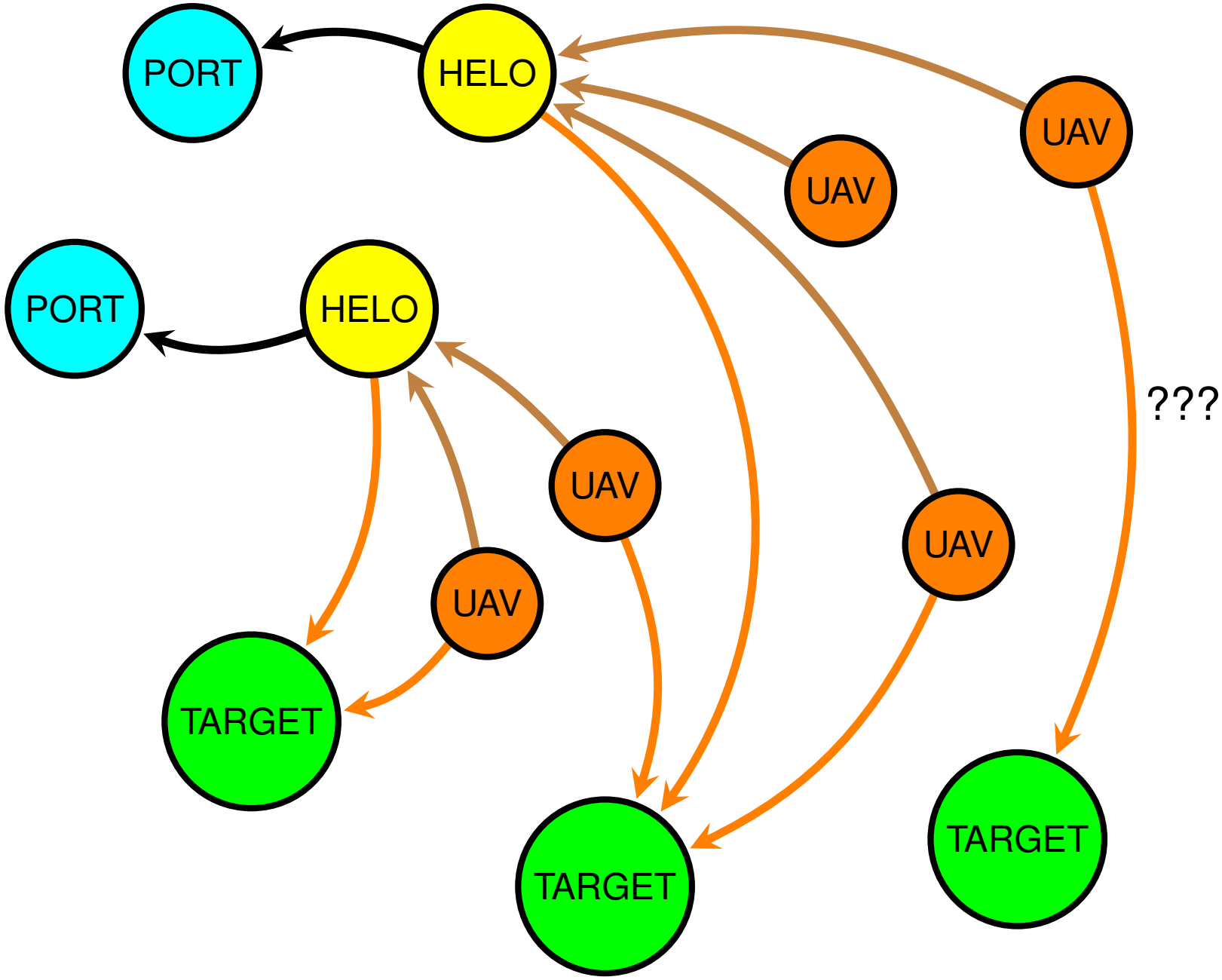

PORT
HELO
UAV
UAV
UAV
TARGET
TARGET

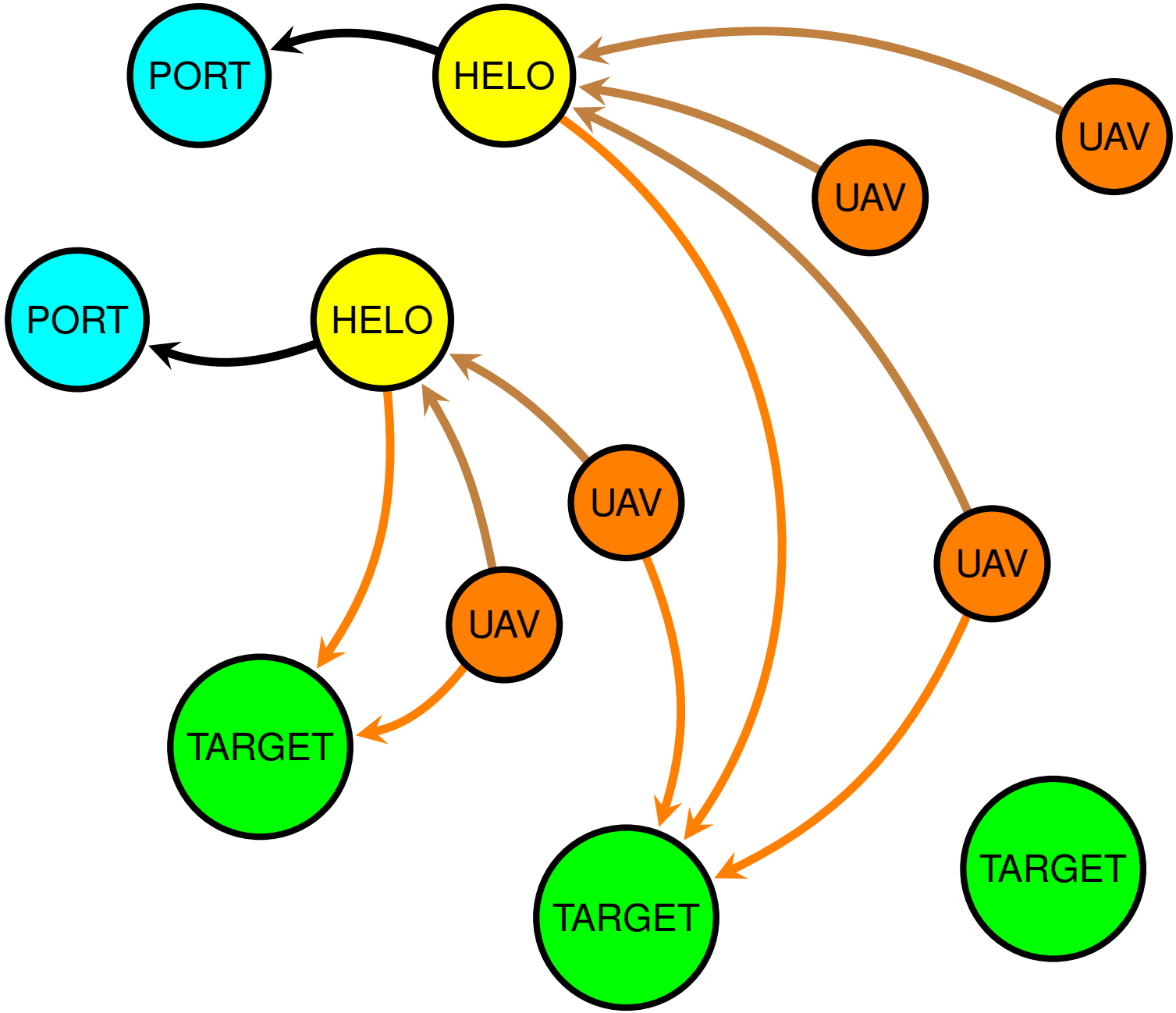

PORT
HELO
UAV
UAV
UAV
TARGET
TARGET

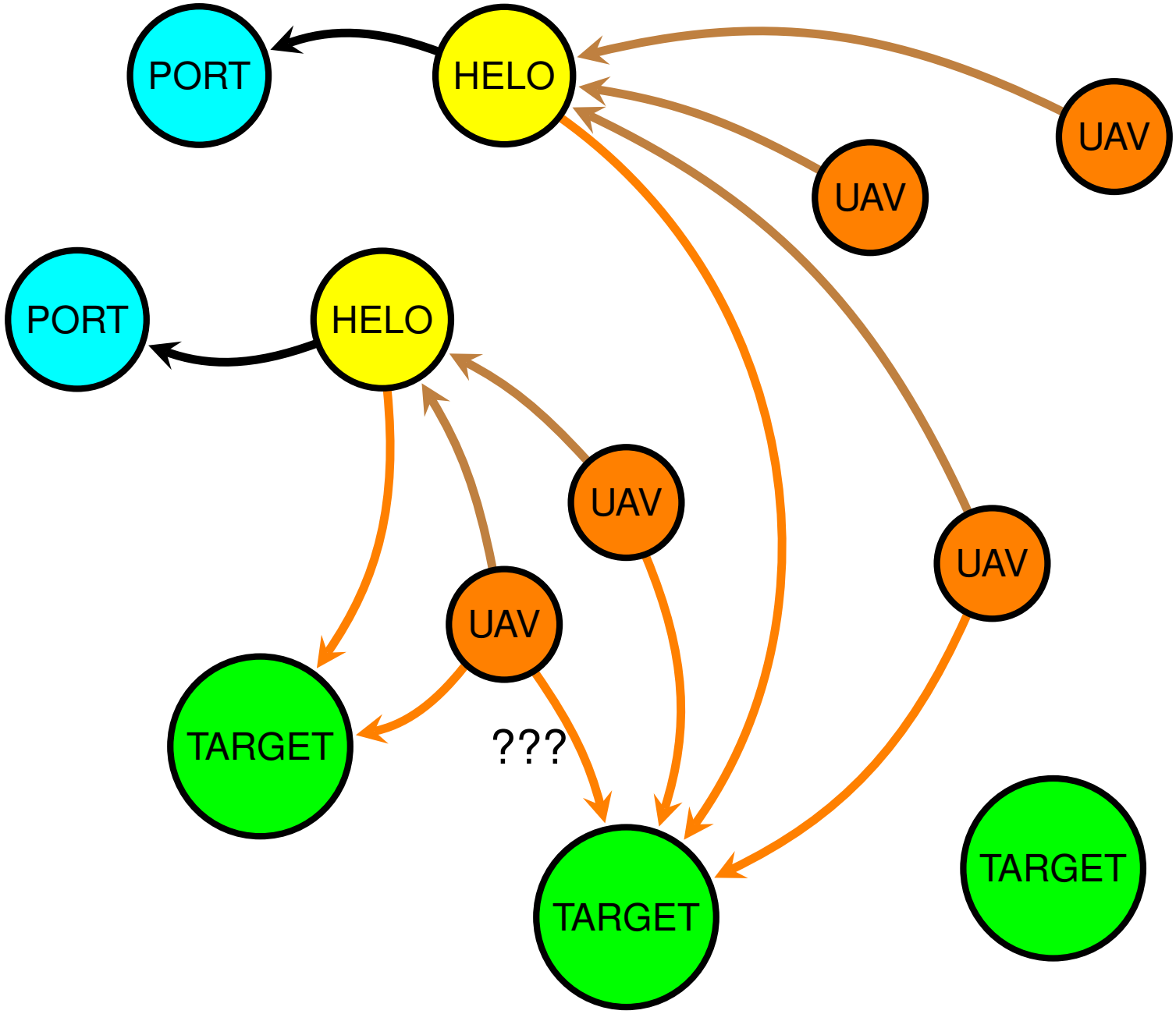

PORT
HELO
UAV
UAV
TARGET

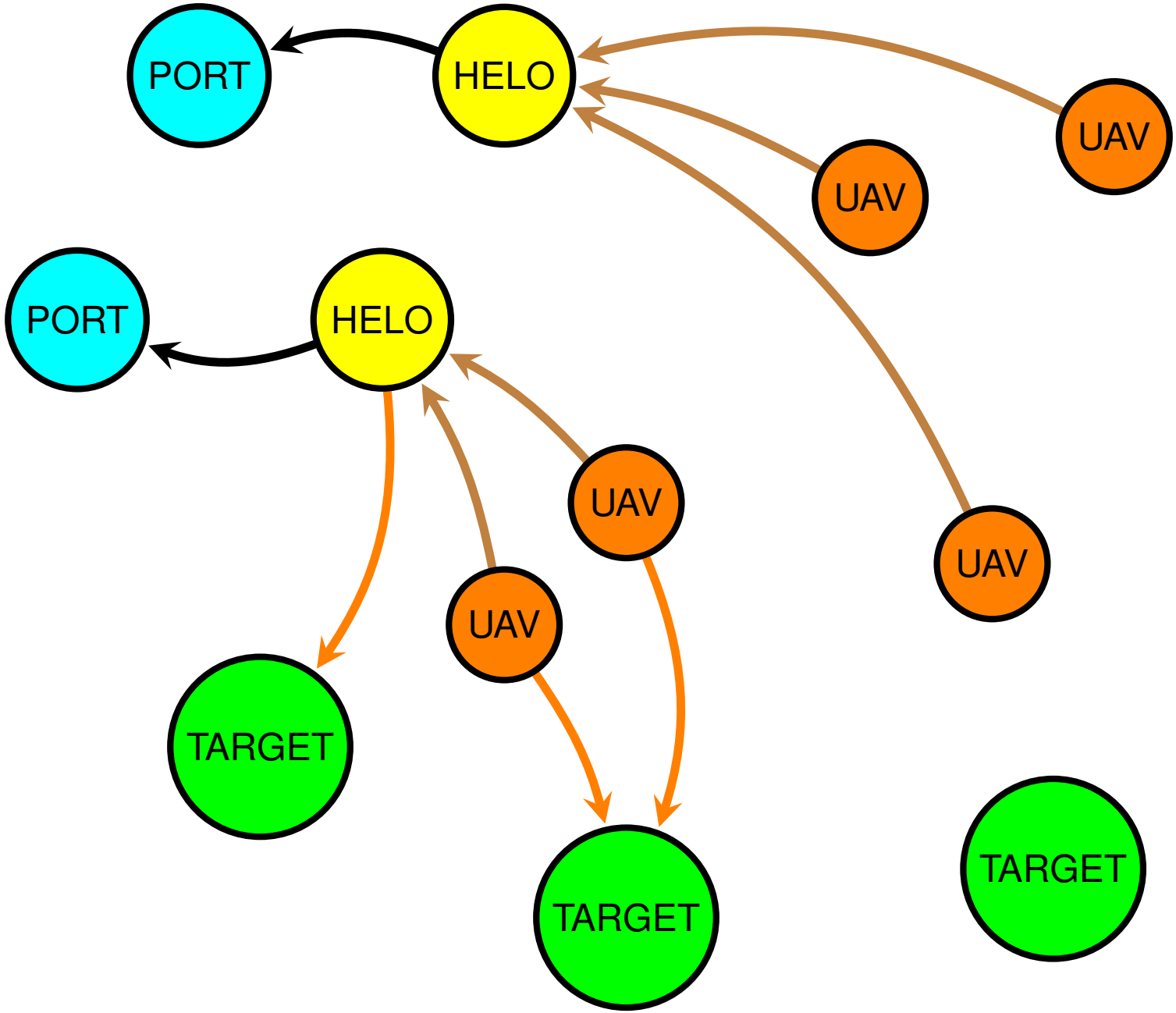

PORT
HELO
UAV
UAV
PORT
HELO
UAV
UAV
UAV
TARGET
TARGET
TARGET


PORT
HELO
UAV
UAV
PORT
HELO
UAV
UAV
UAV
TARGET
TARGET
TARGET


PORT
HELO
UAV
UAV
PORT
HELO
???
UAV
UAV
UAV
TARGET
TARGET
TARGET


PORT
HELO
UAV
UAV
PORT
HELO
UAV
UAV
UAV
TARGET
TARGET
TARGET


PORT
HELO
UAV
UAV
PORT
HELO
UAV
UAV
UAV
TARGET
???
TARGET
TARGET


PORT
HELO
UAV
UAV
PORT
HELO
UAV
UAV
UAV
TARGET
TARGET
TARGET

For more see:

- John Baez, John Foley, Joseph Moeller and Blake Pollard, Network models: arXiv:1711.00037.

- John Baez, Complex adaptive system design, *Azimuth*.