Heterogeneous Broadcast

Murdoch J. Gabbay

Anoma Day

6 June 2025

What this talk is about

I'll report on some consensus research that Isaac Sheff and I have carried out over the last 12 months, to understand his ideas on *heterogeneous broadcast and consensus* algorithms.

Heterogeneous Bracha Broadcast (HBB) is a distributed *broadcast* protocol, which is a precursor to the *Heterogeneous Paxos* (HP) distributed *consensus* algorithm. Anoma plans to use these as broadcast and consensus algorithms respectively.

HBB and HP do not yet exist: part of our brief was to create them.

By the end of this talk, I hope you'll have a flavour of what a heterogeneous broadcast or consensus protocol looks like, and the issues involved in creating one.

Let's recall the standard (homogeneous) problem of distributed broadcast and consensus:

Distributed Broadcast and Consensus

Broadcast and consensus protocols help a group of participants decide/deliver/output/learn a value v (proposed by a participant), in the presence of some bounded amount of faulty behaviour. Relevant correctness properties are:

1. Agreement (safety guarantee):
   If correct (= live and follows the protocol) p delivers v and correct p' delivers v', then $v = v'$.
2. Totality for broadcast (liveness guarantee):
   If correct p delivers, then every correct p' eventually delivers.
3. Totality for consensus (liveness guarantee):
   Every correct p eventually delivers v.

Thus, subject to assumptions on the number of faulty participants:

1. For broadcast, correct participants must deliver and agree, if any one correct process delivers.
2. For consensus, correct participants must deliver and agree.

A problem

Before Anoma can use HBB and HP, someone has to invent them.

Isaac had proposed an HP algorithm, and I had developed an axiomatic method for analysing distributed protocols. We decided to apply the proposed axiomatic method to understand the proposed HP, and see what would happen.

During our analysis,

1. we broke the original HP protocol design (it was wrong) and
2. we proposed two replacement HBB protocols.

Let's consider some motivation for homogeneous/heterogeneous broadcast, from daily life:

An example from daily life (homogeneous version)

Consider my family: {Mum, Dad, Son}. Participants Mum and Dad communicate to deliver decisions $[v_1, v_2, \dots]$ to Son.

Our protocol guarantees the following properties:

- *Agreement*: if Mum and Dad announce values v_i and v_i' respectively for the ith decision, then these values are equal.
  E.g. I should not tell my son that he has to shower before bed, and my wife says he doesn't.
- *Totality*: if Mum announces some decision v_i then Dad will back her up, and vice versa.
  E.g. if my wife tells my son to shower, and he prevaricates, I will eventually confirm with 'do as your mother says'.

Heterogeneous version: Mum, Dad, Gran, Auntie, Son

In some situations, my son may expect heterogeneous values. This is a feature, not a bug.

Suppose my son is visiting his grandmother's house with his aunt. He knows that this pair of adults may occasionally produce values that are different from his parents'.

More specifically: Mum, Dad, Gran, and Auntie are likely to agree on most decisions, *but* if there are differences, then Mum and Dad will form one bloc and Gran and Auntie will form another.

Note that:

1. There may be differences in values decided, but this is not chaos.
2. Son has nontrivial, heterogeneous, hierarchical expectations about who will remain in agreement with whom.
3. These expectations exist *a priori*, before algorithms are run.

Heterogeneous Broadcast

Intuitively, in the heterogeneous case the universe might split into factions that agree within themselves but not necessarily between one another.

This may be a result of faulty behaviour, but it may also be desirable, e.g. if we are bridging between blockchains, or coordinating between groups of people or machines.

Participants don't necessarily know in advance who will turn out to be faulty or who might be communicating with whom.

We want to be able to make guarantees about behaviour, given suitable (externally but not internally known) assumptions on the state of the system.

Heterogeneous Broadcast

We annotate participants with a 'follows' relation $p \propto p'$. If we expect this to be symmetric, we may call this relation 'should agree' and write it $p \infty p'$.

In the previous example, my son expects Dad ∞ Mum and Gran ∞ Auntie with near certainty, and Mum ∞ Gran with lesser certainty.

Suitable generalisations of Agreement and Totality are:

1. Agreement: If correct p delivers v and correct p' delivers v' and $p' \propto p$, then $v = v'$.
2. Totality: If correct p delivers v, then every correct p' such that $p' \propto p$, eventually delivers v.

How to (dis)agree

The behaviour of $\propto$ and ∞ can be nontrivial.

Example: Suppose Jamie $\propto$ Isaac $\propto$ Chrisopher.

Now suppose I learn that hackers have used AI to steal Christopher's identity and clone his likeness.

Obviously, now I can't follow any information from Christopher. But the question is, do I know whether Isaac knows about the identity theft? If I believe that Isaac $\propto$ Christopher, then I also cannot follow any information from Isaac, because the hackers might be feeding him false information via their copy of Christopher.

So whom I $\propto$ or ∞ is determined not just about what I know or believe about Christopher: it's also about what I know or believe Isaac knows or believes about Christopher.

How to make this all formal?

So far, this has all been intuition and examples. We will make it formal, using mathematics.

We will need:

1. *Semitopology*, for quorums.
2. *History structures*, for assessing whom to trust.
3. *A modal logic*, for formally asserting the rules of the protocol.

I will define each in turn.

You might like to strap yourself in for this bit . . .

Background: semitopology

A semitopology $(\mathrm{P}, \mathrm{Open})$ ($\approx$ quorum system) is a set of

- participants P and
- open sets or quorums $\mathrm{Open} \subseteq \mathit{powerset}(\mathrm{P})$.

In the case that $\#\mathrm{P} = 3f + 1$, it is typical to take $O \in \mathrm{Open}$ when O is empty or $\#O \geq 2f + 1$,

Semitopology has a rich mathematical theory which is implicit in this talk. I will mostly elide it, but we will need one definition: if $p \in \mathrm{P}$ and $O \in \mathrm{Open}$ and $p \in O$ then call O a neighbourhood of p. Define

$$\mathit{nbhd}(p) = \{O \in \mathrm{Open} \mid p \in O\}.$$

I'll use neighbourhoods later.

Motivation for history structures

The notion of $p \propto p'$ is inherently *epistemic*. This is a fancy way of saying that the validity of $p \propto p'$ depends on who knows what.

1. 'It's raining' is not epistemic. Its validity just depends on whether it *is* raining.
2. 'I trust you' is epistemic. Its validity depends what I think I know about your behaviour *and* that of other participants, *and* what I think you know about.

$p \propto p'$ means, intuitively, that p trusts the communication channels between it and p' *and* p trusts whatever communication channels p' has.

All of this detail is packed into the word 'heterogeneous'.

History structure: our model of knowledge

1. Fix arbitrary disjoint sets of participants P and events Event.
2. Define PreHistory by

$$\text{PreHistory} = \mathit{fin}(\text{P} \times \text{Event} \times \text{PreHistory}).$$

   fin means finite powerset, so a prehistory is a finite set of participant-event-prehistory tuples.
3. Suppose $H, H' \in \text{PreHistory}$ and $p \in \text{P}$ and $E \in \text{Event}$. Define $H' \in H$ by

$$H' \in H \text{ when } \exists p' \in \text{P}, E' \in \text{Event}.(p, E, H') \in H.$$

4. Call $H \in \text{PreHistory}$ transitive when $\forall H'.H' \in H \Rightarrow H' \subseteq H$.
5. Let History be the the set of hereditarily transitive prehistories. (Pleasingly reminiscent of von Neumann ordinals.)

Thus, a history is a prehistory all of whose elements are histories. Or: *a history H is a set of (p', E', H') (participant-event-history) tuples.*

A modal logic over histories

Call a tuple $\mathcal{M} = (\mathcal{H}, \mathrm{P}, \mathrm{Open})$ a model, where $\mathcal{H}$ is a history and $(\mathrm{P}, \mathrm{Open})$ is a semitopology.

Logical time is a notion I use understand distributed algorithms. It represents a logical (not an algorithmic) stage in some algorithm.

- In Paxos, the logical time is the 'round' (a natural number).
- The relevant notion of logical time in HBB is History. This was not obvious.

(Comment on relative complexity. Comment on logical vs. algorithmic time here and here.)

A modal logic over histories

Define predicates by

$$\phi ::= \bot \mid \phi \Rightarrow \phi \mid \forall p.\phi \mid E \mid \mathsf{itp}(\phi) \mid \Diamond\phi \mid \boxdot_{l_1 \ldots l_n} \phi \mid \mathsf{seq} \mid p \propto p'.$$

Read

- itp as *in the past*;
- $\Diamond$ as *someone*;
- $\boxdot_{l_1 \ldots l_n}$ as *for some* $l_1 \ldots l_n$*-quorum intersection*; and
- seq as *sequential*.

A modal logic over histories

We interpret validity of predicates with respect to a model $\mathcal{M}$, a participant $p \in \mathrm{P}$, and a logical time $H \in \mathcal{H}$, thus the validity judgement has the form

$$p, H \vDash_{\mathcal{M}} \phi.$$

We now define it:

A modal logic over histories (easy bits)

Recall that $\mathcal{H}$ is a set of (p, E, H)-tuples, and ϕ is (in)valid in $\mathcal{H}$ at some p and H:

$$\begin{array}{lcl}
p, H \models_{\mathcal{M}} \bot & \Leftrightarrow & \bot \\
p, H \models_{\mathcal{M}} \phi \Rightarrow \phi' & \Leftrightarrow & p, H \models \phi \Rightarrow p, H \models_{\mathcal{M}} \phi' \\
p, H \models_{\mathcal{M}} \forall p'.\phi & \Leftrightarrow & \forall p'' \in \mathrm{P}. p, H \models \phi[p' := p''] \\
p, H \models_{\mathcal{M}} E & \Leftrightarrow & (p, E, H) \in \mathcal{H} \\
p, H \models_{\mathcal{M}} \mathrm{itp}(\phi) & \Leftrightarrow & \exists H' \in H.\ p, H' \models_{\mathcal{M}} \phi \\
p, H \models_{\mathcal{M}} \Diamond \phi & \Leftrightarrow & \exists p' \in \mathrm{P}. p', H \models_{\mathcal{M}} \phi
\end{array}$$

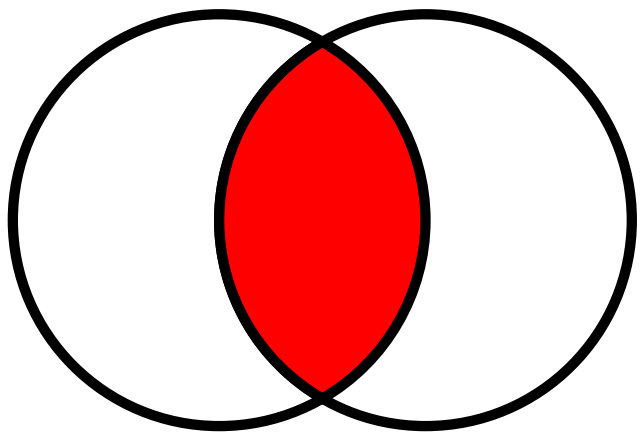
We can express other logical connectives in the usual way, e.g.

$$\neg\phi = (\phi \Rightarrow \bot) \quad \phi \vee \phi' = (\neg\phi \Rightarrow \phi') \quad \exists p'.\phi = (\neg \forall p'.\neg\phi).$$

A modal logic over histories: quorum intersections

$$p, H \models_{\mathcal{M}} \boxdot_{I_1 \ldots I_n} \phi \;\Leftrightarrow$$
$$\exists O_1 \in nbhd(I_1) \ldots O_n \in nbhd(I_n). \forall p' \in O_1 \cap \ldots \cap O_n. (p', H \models_{\mathcal{M}} \phi)$$

This expresses that there exists *some* quorum intersection between quorums of $I_1, \ldots, I_n$, *all* of whose participants has ϕ. E.g for I_1 and I_2:



(Image credit: Wikipedia.)

A useful special case is $\boxdot_I \phi$, which means that there exists an I-quorum of participants *all* of whom have ϕ.

A modal logic over histories: quorum intersections

Also write $\diamondsuit_{l_1 \ldots l_n} \phi$ for $\neg \boxdot_{l_1 \ldots l_n} \neg \phi$ (the *de Morgan dual*). Then:

$$p, H \models_{\mathcal{M}} \diamondsuit_{l_1 \ldots l_n} \phi \;\Leftrightarrow$$
$$\forall O_1 \in nbhd(l_1) \ldots O_n \in nbhd(l_n). \exists p' \in O_1 \cap \ldots \cap O_n. (p', H \models_{\mathcal{M}} \phi)$$

This expresses that *all* $l_1 \ldots l_n$-quorum intersections contain *some* participant with ϕ.

A modal logic over histories: sequentiality

$$p, H \models_{\mathcal{M}} \text{seq} \;\Leftrightarrow\; \forall E, E' \in \text{Event}.\; p, H \models_{\mathcal{M}} \text{itp}(E) \wedge \text{itp}(E') \Rightarrow p, H \models_{\mathcal{M}} \text{itp}(E \wedge \text{itp}(E')) \vee \text{itp}(E' \wedge \text{itp}(E))$$

This says that events at p in H occur *in sequence*.

We assume of the real world that each event is cryptographically signed by whichever participant performed it, so p can't just report an E' for some other p' if that didn't actually happen — but p can *omit* historial events when it reports its events to *other* participants.

That's the episemic part: reporting events while omitting context, i.e. out of sequence. Nonsequentiality indicates epistemological hacking!

A modal logic over histories

Define

$$\text{safe} = \forall p'.\diamondsuit_{pp'} \text{seq}.$$

Thus,

$$p, H \models_{\mathcal{M}} \text{safe} \;\Leftrightarrow\; p, H \models_{\mathcal{M}} \forall p'.\diamondsuit_{pp'} \text{seq}.$$

p is safe when it has sequential quorum intersections with every participant.

If $p, H \models_{\mathcal{M}} \phi$ does not depend on the choice of participant p, then omit it and just write $H \models_{\mathcal{M}} \phi$. Define

$$H \models_{\mathcal{M}} p_1 \propto p_2 \;\Leftrightarrow\; p_2, H \models_{\mathcal{M}} \text{safe}.$$

So p_1 follows p_2 at logical time H when p_2 is safe at that time (and if it's not, it doesn't).

Now for the protocol

Finally we can express an HBB protocol. To be fair, it's pretty short:

(**Echo0**?)	$\mathsf{echo}(0, v) \Rightarrow \mathsf{itp} \Diamond \mathsf{propose}(v)$
(**Echo1**?)	$\mathsf{echo}(1, l, v) \Rightarrow \boxdot_l \mathsf{itp}\, \mathsf{echo}(0, v)$
(**EchoDup**)	$\mathsf{echo}(0, v) \Rightarrow \mathsf{itp}(\mathsf{echo}(0, v')) \Rightarrow v{=}v'$
(**Vote**?)	$\mathsf{vote}(l, l', v) \Rightarrow l \propto l' \wedge \mathsf{itp} \Diamond \mathsf{echo}(1, l', v)$
(**Deliver**?)	$\mathsf{deliver}(l, l', v) \Rightarrow \boxdot_l \mathsf{itp}\, \mathsf{vote}(l, l', v)$

Agreement property

Lemma: $H \models_{\mathcal{M}} \text{itp}\Diamond\text{deliver}(I_2, I_2', v_2) \wedge \text{itp}\Diamond\text{deliver}(I_1, I_1', v_1)$ and $H \models_{\mathcal{M}} \diamondsuit_{I_1 I_2} \text{seq}$ implies $v_1 = v_2$.

Proof: Suppose $H \models_{\mathcal{M}} \text{itp}\Diamond\text{deliver}(I_2, I_2', v_2) \wedge \text{itp}\Diamond\text{deliver}(I_1, I_1', v_1)$. By (**Deliver**?) we have

- an I_2-quorum of past $\text{vote}(I_2, I_2', v_2)$, and
- an I_1-quorum of past $\text{vote}(I_1, I_1', v_1)$.

By assumption, these quorums intersect at a sequential participant q. Assume wlog $H' \in H$ and

$$q, H' \models_{\mathcal{M}} \text{vote}(I_2, I_2', v_2) \wedge \text{itp}(\text{vote}(I_1, I_1', v_1)).$$

By (**Vote**?) I_2' is safe at logical time H', so it has sequential quorum intersections with I_1'. Following (**Echo1**?), there exist an I_2'-quorum of $\text{echo}(0, v_2)$ and an I_1'-quorum of $\text{echo}(0, v_1)$ and so a sequential participant in their intersection that has echoed both, in sequence! By (**EchoDup**) $v_1 = v_2$, and we're done.

What have we learned?

1. This stuff is hard. There are a lot of definitions, and finding them was not obvious. It's taken months to get here.
2. On the other hand, the axioms expressing the protocol are fairly straightforward. The end result is pretty good.
3. There's a large design space of possible protocols. We've only found two that work — I present the simpler one! I haven't found any systematic way to explore the design space.
4. You get used to this world of histories and sequential and safe participants.
   It's a lot to understand in this talk, but as a conceptual structure it seems a solid foundation for future work.
5. The logical approach was (for me) *absolutely necessary* to understand the protocol. I find it highly effective.

Future work

1. Write this up as a conference or (perhaps more likely) journal paper.
2. Implement it.
3. Build a modal-checker for history structures might speed up exploration of the state space.
4. Improve the protocols. They work, but they're either a bit weak or a bit slow. It would be nice, at least, to have a fast path.