A Bit About Myself.

- I enjoy building practical systems
- Backed by theories
- HotStuff protocol and its prototype
- Consensus protocol used by Libra
- libhotstuff surpasses the state-of-art PBFT implementation performance
- Snow protocol family and Avalanche
- This talk!
The Infrastructure for Blockchain 3.0
What is a “Blockchain Infrastructure”?

How is it like in the year of 2019?
A Giant Beast.

While \((x > 10)\) {
    destroy_the_world();
    create_a_new_world();
    x--;
}

Alice sends Bob: $10
Blockchain 3.0?

Blockchain 1.0
Blockchain 2.0
Blockchain 3.0
Blockchain 1.0: Nakamoto Era.

"Proof-of-Work" (PoW)

"Longest Chain" Rule

Block Growth

Bitcoin: A Peer-to-Peer Electronic Cash System
Satoshi Nakamoto
Blockchain 1.0: Nakamoto Era.

↑ Proof-of-work also serves as Sybil prevention ("Permissionless")
↑ Graceful safety degradation
↑ Loose membership knowledge

↓ Proof-of-work wastes a lot of energy!
  ○ As of 2018: one Austria, two Denmarks, or three Irelands
↓ Extremely low capacity (throughput as low as ~3 TPS)
↓ Extremely long confirmation time (> 1 hour)
↓ Poor efficiency in safety (compared to Blockchain 2.0)
Blockchain 2.0: PBFT — Resurrect the Pharaoh.

↑ Very fast when the network is small
↑ Deterministic safety (100% safe)
↑ A long line of research

↓ Cannot scale easily: the “leader/coordinator” dilemma
  ○ Randomized BFT approaches are even worse
↓ Quorum-based: 100% accurate knowledge of all participants
↓ Hard limit on the Byzantine adversaries
↓ Many are very complicated
Blockchain 2.0: PBFT – Resurrect the Pharaoh.

The Saddest Moment, Mickens 2013
Performance Bottlenecks: PBFT as an Example.
Blockchain 3.0?

Blockchain 1.0
Blockchain 2.0
Blockchain 3.0
Blockchain 3.0...

Consensus 1.0  Blockchain 1.0
Consensus 2.0  Blockchain 2.0
Consensus 3.0  Blockchain 3.0
"But we have sharding!"

Theorem 1: Sharding does NOT address the scalability issue of Consensus.

Corollary 1: Sharding comes at the cost of losing fault tolerance.
The “Sharding” Logic.

“This soup tastes sooo bad! :(

“But you can add some Gochujang so you don’t feel too bad.”
"This soup tastes sooo bad! :("
“But you can add some Gochujang so you don’t feel too bad.”
...
Why couldn’t I just have Budae Jjigae? And...you can also add Gochujang. :)

Sharding! Sharding? Sharding...
Then, Could We Do Better?

Possible?
<table>
<thead>
<tr>
<th>Year</th>
<th>Protocol/Innovation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1989</td>
<td>Quorum-Based; Paxos/Raft, PBFT</td>
</tr>
<tr>
<td></td>
<td>“2f + 1”</td>
</tr>
<tr>
<td></td>
<td>“3f + 1”</td>
</tr>
<tr>
<td>2008</td>
<td>Longest-Chain; Nakamoto</td>
</tr>
<tr>
<td></td>
<td>50% power</td>
</tr>
<tr>
<td>2018</td>
<td>Random Sampling; Snow/Avalanche</td>
</tr>
<tr>
<td></td>
<td>???</td>
</tr>
</tbody>
</table>
Blockchain 3.0: Snow/Avalanche.

- A paper presenting “Avalanche” was dropped on IPFS in May 2018
- Inspired by epidemic protocols and gossip networks
- New methodology that combines the best of 1.0 and 2.0

- Graceful safety degradation
- Loose membership knowledge
- Very fast regardless of the network size
- Energy efficient
- The operational logic just makes sense
Wait!

“But many other protocols also claim very high TPS!”

Theorem 2: Throughput only reflects how much load a system can buffer, while latency only reflects how fast a single user is served.

Corollary 2: We will need them both when evaluating a system!
The TPS vs. Latency Logic.

A bank branch could serve a single person very quickly.

But what if there are many people waiting in the line?

Throughput reflects the capacity.
The TPS vs. Latency Logic.

A truck full of hard drives can transport Petabytes ($2^{50}$ bytes) of data.

Oh, very high throughput!

Well, the latency...

Talking about throughput without latency is meaningless.
Ava Throughput.

![Throughput Chart]

- Throughput (tps): 6.9K, 7.0K, 6.7K, 7.1K, 6.8K, 7.2K, 6.9K, 7.4K, 6.5K, 6.9K
- Number of nodes: 125, 250, 500, 1000, 2000
Ava Latency.
Blockchain 3.0: Snow/Avalanche.

In an even more realistic setting:
- 2000 nodes in 20 cities across the globe
- All nodes directly participate in consensus
- Full signature verification

Our evaluation results:
- $\sim3400$ tps
- $\sim1.35\text{ sec}$

Fig. 19: Latency histogram/CDF for $n = 2000$ in 20 cities.
Blockchain 3.0: Snow/Avalanche.

That’s interesting.

Tell me how this protocol works and why it is so efficient!

Gossip network is not only a way to deliver data, but can also be used as a consensus!
The Snow Protocol Family: Let Rumors Spread.

- “Gossip”: N people are discussing whether some rumor is true
- Everyone independently choose k others at random — “sample” the crowd

For Alice:
- Asks those k people whether the rumor is true — a “query”
- Updates the confidence in her current bias (true/false) by checking the majority opinion of each “query”
Full Broadcast to Partial Sampling.

Sample all
Full Broadcast to Partial Sampling.

Sample all
Only Sample $k$ out of $N$
Alice asks 5 other people randomly

Alice: “Blue is the majority answer!”
Alice asks 5 other people randomly

Bob asks 5 other people randomly

Alice: “Blue is the majority answer!”

Bob: “Yellow is the majority answer!”
Alice asks 5 other people randomly

Bob asks 5 other people randomly

Alice: “Blue is the majority answer!”

Bob: “Yellow is the majority answer!”
Alice asks 5 other people randomly

Alice: “Blue is the majority answer!”

Bob asks 5 other people randomly

Bob: “Yellow is the majority answer!”
Protocol Demo

https://avalabs.org/snow-bft-demo
The Snow Protocol Family: Uniqueness

- The “magical” k: \( k = 10\sim20 \), for \( N = 1000, 10000, \ldots \).
- It could be viewed as a heavily relaxed quorum system.
- It doesn’t require any PoW.

- As fast as the network propagates.
- Super scalable with respect to the number of nodes.
- Loose membership.
- Any anti-Sybil mechanism could be applied.
The Infrastructure for Blockchain 3.0
Platform of Platforms

While \(x > 10\) {
    destroy_the_world();
    create_a_new_world();
    x--;
}

Alice sends Bob: $10
Platform of Platforms

While (x > 10) {
  destroy_the_world();
  create_a_new_world();
  x--;
}

Alice sends Bob: $10
Platform of Platforms

While (x > 10) {
   destroy_the_world();
   create_a_new_world();
   x--;
}

This reminds us of...
I told you, sharding is just another trick. ;)

“This soup tastes sooo bad! :("
“But you can add some Gochujang so you don’t feel too bad.”

We’re making our delicious Budae Jjigae, ...with Gochujang!
Public Testnet coming soon.
Our devs are working like a dog.
Stay tuned and contribute to our project in the future!

Kakao (AVA Korea, 아바랩스)