

Lumen(스텔라루멘)

해당 번역본은 이용자가 디지털자산 관련 정보를 편리하게 확인할 수 있도록 단순 참고용으로 번역된 것이며, 투자 권유를 목적으로 하지 않습니다. 그 내용이 정확하지 않을 수 있으므로 정확한 정보 습득을 위해서는 원문을 참고하시거나 원문 작성 주체 측에 문의하시기 바랍니다. 두나무는 해당 번역본 내용의 진실성 및 정확성을 보장하지 않고, 제공된 정보에 의한 투자 결과에 대하여 어떠한 책임도 지지 않으며, 투자 손실은 투자자에게 귀속됩니다.

스텔라 합의 프로토콜(Stellar Consensus Protocol): 인터넷 단계 합의를 위한 연합 모델

DAVID MAZIÈRES, 스텔라 개발 재단

본 백서는 비잔틴 합의(Federated Byzantine Agreement, FBA)라 불리는 합의에 관한 새로운 모델을 소개합니다. FBA는 각 노드에서 개별 신뢰 결정을 내려 시스템 수준 쿼럼을 함께 결정하는 쿼럼 슬라이스(Quorum Slice)를 통해 견고성을 달성합니다. 슬라이스는 개별 네트워크의 대등 접속 및 전송 결정이 현재 인터넷을 통합하는 방식으로 시스템을 결합합니다.

또한 FBA를 위한 구성인 스텔라 합의 프로토콜(Stellar Consensus Protocol, SCP)도 제시합니다. 모든 비잔틴 합의 프로토콜과 마찬가지로 SCP는 공격자의 합리적인 행동을 가정하지 않습니다. 만장일치로 승인된 회원 목록을 전제로 하는 기존의 비잔틴 동의 모델과 달리 SCP는 유기적 네트워크 성장을 촉진하는 공개 회원제도를 적극 활용합니다. 탈중앙화된 작업증명(Proof of Work, PoW) 및 지분 증명(Proof of Stake, PoS) 방식에 비해 SCP는 컴퓨팅 및 재무적 요구 사항이 완만하여 진입 장벽을 낮추고 잠재적으로 새로운 참가자에게도 금융 시스템을 개방합니다.

CCS 개념: ●보안성 및 프라이버시 → 분산된 시스템 보안성, 보안 프로토콜

추가 키워드 및 구문: 비잔틴 오류 허용, 비동기식 시스템

1. 서론

현재 금융 인프라는 폐쇄형 시스템들이 혼재되어 있습니다. 이러한 시스템 간의 격차는 트랜잭션 비용이 높고 [Provost 2013] 금전의 이동이 정치 및 지리적 경계상에서 느리게 이동함을 뜻합니다 [Banning-Lover 2015; CGAP 2008]. 이러한 마찰로 인해 금융 서비스의 성장이 위축되어 수십억 명의 사람들이 재정적 지원을 받지 못하고 있습니다. [Demirguc-Kunt et al. 2015].

이러한 문제를 해결하려면 인터넷에서 본 것과 같은 유형의 유기적 성장과 혁신을 지원하면서도 금융 거래의 무결성을 보장하는 금융 인프라가 필요합니다. 그간 우리는 무결성을 보장하기 위해 높은 진입 장벽에 의존해 왔습니다. 우리는 기존 금융 기관을 신뢰하고 이들을 조절하는 데 최선을 다합니다. 그러나 이러한 배타성은 유기적 성장이라는 목표와 상충됩니다. 성장을 위해서는 약간의 재정 및 연산 리소스만 보유하는 새롭고 혁신적인 참가자가 필요합니다.

우리는 누구에게나 열려 있어 새로운 조직이 참여할 수 있고 서비스를 받지 못하는 커뮤니티에 재정적 접근을 확장할 수 있는 전 세계 금융 네트워크가 필요합니다. 이러한 네트워크의 과제는 참가자가 트랜잭션을 올바르게 기록하도록 돕는 것입니다. 진입 장벽이 낮은 경우 사용자는 공급자가 스스로를 감독하는 것을 신뢰하지 않을 것입니다. 제공업체는 전 세계를 무대로 함에 따라 단일 기관이 운영하는 네트워크의 경우 전적으로 신뢰하지 않을 것입니다. 한 가지 타당한 대안은 참가자들이 서로의 트랜잭션 유효성에 동의함으로써 무결성을 함께 보장하는 탈중앙화된 시스템입니다. 해당 합의(Agreement)의 성립 여부는 전적으로 전세계적 합의(Consensus)를 위한 메커니즘에 달려 있습니다.

본 백서는 전세계적 합의에 적합한 모델인 연합 비잔틴 합의(FBA)를 제시합니다. FBA에서 각 참가자는 자신이 신뢰하는 다른 참가자를 인지하고 있습니다. 트랜잭션을 완료한 것으로 간주하기에 앞서 대다수의 다른 사람들이 트랜잭션에 동의할 때까지 기다립니다. 또한 신뢰할 수 있다고 판단된 참가자 역시 본인이 신뢰하는 참가자가 동의할 때까지 트랜잭션에 동의하지 않습니다. 결국에는 공격자가 롤백하는 것이 불가능하게 될 만큼 충분한 수의 네트워크가 트랜잭션을 수락합니다. 참가자들은 그 후에만 트랜잭션이 완료된 것으로 간주합니다. FBA의 합의는 금융 네트워크의 무결성을 보장할 수 있습니다. 탈중앙화된 통제가 유기적 성장을 촉진할 것입니다.

본 백서는 또한 FBA의 기반이 되는 스텔라 합의 프로토콜(Stellar Consensus Protocol, SCP)을 제시합니다. 그러한 보장을 허용하는 모든 노드 실패 시나리오에서 합의를 보장한다는 점에서 SCP의 보안성이 비동기식 프로토콜(Asynchronous Protocol)에 최적임을 증명합니다.

또한 참가자의 실패가 신뢰 의존도를 충족시키는 것을 불가능하게 만드는 경우를 제외하고는 합의가 더 이상 불가능한 차단된 상태에서부터 SCP가 자유로움을 제시합니다. 우선 SCP는 안전을 입증할 수 있는 합의 메커니즘이며, 다음과 같은 4가지 핵심 요소를 보유하고 있습니다.

- 탈중앙화된 통제. 누구든지 참가할 수 있으며 합의에 필수적인 승인 권한을 남용하는 중앙 당국이 존재하지 않습니다.
- 낮은 지연율. 실제로 노드는 사람들이 웹 또는 결제 트랜잭션에서 기대하는 수준의 속도, 즉 최대 몇 초 이내로 합의에 도달할 수 있습니다.
- 유연한 신뢰. 사용자들은 본인이 생각하기에 적절한 당사자들의 조합을 신뢰할 자유가 있습니다. 예를 들면 소규모의 비영리기관이 훨씬 더 큰 규모의 기관을 정직하게 유지하는 데 핵심적인 역할을 맡을 수 있습니다.
- 비동기식 보안성. 안전은 상상할 수 없을 정도로 방대한 컴퓨팅 성능을 가진 공격자로부터 보호하기 위해 현실적으로 매개변수를 조정할 수 있는 디지털 서명 및 해시 패밀리를 기반으로 제공됩니다.

SCP는 조직이 중요한 기능을 정직하게 수행할 수 있도록 금융 시장 이외에도 여러 분야에 응용될 수 있습니다. 예를 들어 인증 당국(Certificate Authorities, CA)은 문자 그대로 웹에 대한 키를 보유합니다. CA가 현장에서 사용되는 잘못된 인증서에 서명한 사례들이 존재합니다 [Microsoft 2013; Langley 2015]. 몇몇 제안들에서는 인증서 투명성을 통해 이 문제를 다룹니다 [Kim et al. 2013; Laurie et al. 2013; Basin et al. 2014; Melara et al. 2014]. 인증서 투명성을 통해 사용자는 지정된 엔티티에 대해 발급된 인증서 내역을 검사하고 기존 키의 보증 없이 엔티티의 공개 키를 변경하려는 CA의 시도를 감지할 수 있습니다. SCP는 인증서 투명성의 핵심에서 지워지지 않는 인증서 기록을 강화할 수 있는 잠재력을 가지고 있습니다. 탈중앙화된 감사 그룹 간에 인증서 기록에 대한 글로벌 합의를 요구하는 경우 이전에 발급된 인증서를 철회하고 무효화하기가 더 어려워집니다.

다음 장은 합의에 대한 기존의 접근에 대해 논의합니다. 제3장에서는 연합 비잔틴 동의(FBA)를 정의하며 FBA 모델에 적용될 수 있는 안전 및 활성의 개념을 전개합니다. 제4장에서는 FBA 시스템의 실패에 대한 최적 복구에 대해 논의함으로써 SCP의 보안성 목표를 수립합니다. 제5장은 SCP 프로토콜의 핵심 구성 블록인 연합 투표를 전개합니다. 제6장은 SCP 자체를 제시하여 보안과 차단된 상태로부터의 자유를 제공합니다. 제7장은 SCP의 한계에 대해 논의합니다. 마지막으로 제8장은 결과를 요약합니다. 수학 표기법에 친숙하지 않은 독자들을 위해 부록 A는 백서 전반에 사용된 기호들을 정의합니다.

2. 관련 문헌

그림 1은 기존의 합의 메커니즘과 SCP의 차이점에 대해 요약하고 있습니다. 가장 잘 알려진 탈중앙화된 합의 메커니즘은 비트코인(BTC)에 의해 진보를 이룬 작업증명입니다[Nakamoto 2008]. BTC는 합의에 대하여 이중 접근 방식을 취합니다. 먼저 바른 행동이 이루어지도록 합리적인 행위자들에게 인센티브를 제공합니다. 둘째로 충분한 수준의 시스템 컴퓨팅 파워를 확보하지 않은 악의적인 행위자들로부터 보호할 수 있도록 설계된 작업 증명 알고리즘을 통해 트랜잭션의 합의를 봅니다 [Dwork and Naor 1992]. BTC는 탈중앙화된 합의의 장점을 압도적으로 증명했습니다 [Bonneau et al. 2015].

그러나 작업 증명에는 제한 사항들이 있습니다. 먼저 자원을 낭비합니다. 2014년 한 예상에 따르면 BTC는 아일랜드 전체와 버금가는 전력을 소비합니다 [O'Dwyer and Malone 2014]. 둘째로 보안 트랜잭션 합의는 수 분 내지는 수십 분까지도 지연에 의해 악영향을 받습니다 [Karame et al. 2012]. 마지막으로 통상적인 디지털 프로토콜과 비교했을 때 작업 증명은 비동기적 보안을 제공하지 않습니다. 비합리적인 공격자(또는 합의를 방해하려는 외적 인센티브가 있는 공격자)가 존재하는 경우, 작은 계산상의 이점이 보안 가정을 무효화하여 소위 '51% 공격'으로 내역을 덮어쓰는 경우가 발생할 수 있습니다.

메커니즘	탈중앙화된 통제	낮은 지연율	유연한 신뢰	점근적(Asymptotic) 보안성
작업 증명(Proof of Work)	✓			
지분 증명(Proof of Stake)	✓	가능		가능
비잔틴 동의 (Byzantine Agreement)		✓	✓	✓
텐더민트(Tendermint)	✓	✓		✓
SCP(본 작업)	✓	✓	✓	✓

그림 1. 다양한 합의 메커니즘의 속성

더 심각한 현상으로는 최초 연산의 50% 미만을 통제하던 공격자들이 시스템을 조작하여 합류하는 이들에게 불균형한 보상을 제공함으로써 [Eyal and Sirer 2013]과반수의 통제 능력을 얻을 가능성이 존재합니다. 최대 연산 능력의 지원을 받고 있는 선두 디지털 화폐로, BTC는 51% 공격에 대한 보호 수단을 적극 활용합니다. 그러나 소규모에 해당하는 시스템들은 피해 대상이 되어 [crazyearner 2013; Bradbury 2013], BTC 블록체인에 구축되지 않은 모든 작업 증명 시스템에 문제를 제기합니다.

작업 증명의 대안은 지분 증명이며 [King and Nadal 2012], 여기서 합의는 담보를 제공한 당사자들에 따라 다릅니다. 작업 증명과 마찬가지로, 보상은 합리적인 참가자들이 프로토콜을 따르도록 독려합니다. 그리고 일부 설계 내용은 추가로 악의적인 행동에 패널티를 부과합니다 [Buterin 2014; Davarpanah et al. 2015]. 지분 증명은 이전에 담보를 제출했지만 나중에 이를 현금화하고 돈을 지출한 당사자가 그 이후에도 여전히 지분을 보유하고 있던 시점으로 돌아가 내역을 다시 쓸 수 있는 이른바 ‘무위험’ 공격의 가능성을 제공합니다. 이러한 공격을 완화하기 위해 시스템은 지분 증명과 작업 증명을 효과적으로 결합함으로써 지분에 비례하여 요청된 작업의 권한을 축소하거나, 다른(때로는 비공식적인) 합의 메커니즘이 되돌릴 수 없는 체크포인트를 충분히 설정할 수 있을 정도의 기간동안 담보를 반환하는 것을 유예합니다.

합의를 향한 또 다른 접근법은 비잔틴 합의이며 [Pease et al. 1980; Lamport et al. 1982], 그중 가장 잘 알려진 유형은 PBFT입니다 [Castro and Liskov 1999]. 비잔틴 합의는 참가자 중 일부의 (비합리적 및) 독단적 행동에도 불구하고 합의가 성립될 수 있도록 보장합니다. 이 접근은 두 가지 장점을 지니고 있습니다. 첫째, 합의가 빠르고 효율적으로 이루어질 수 있습니다. 둘째, 신뢰와 리소스 소유가 전적으로 분리되어 소형 비영리기관이 은행이나 CA를 비롯한 더욱 강력한 기관들이 정직성을 유지할 수 있도록 합니다. 그러나 문제를 복잡하게 하는 것은 모든 당사자가 정확한 참가자 목록에 동의해야 한다는 것입니다. 또한 여러 번 참여하여 시스템의 오류 허용 범위를 초과하는 것, 이른바 시빌(Sybil) 공격에 공격자가 가담하는 것을 방지해야 합니다 [Douceur 2002]. BFT-CUP [Alchieri et al. 2008]은 알려지지 않은 참가자를 수용하지만 여전히 시빌 증명 중앙화 승인 제어 메커니즘을 전제로 합니다.

일반적으로 비잔틴 동의 시스템의 멤버십은 중앙 당국 내지는 폐쇄된 협의를 통해 설정됩니다. 가입을 탈중앙화하기 위한 기존의 시도들은 혜택 중 일부를 포기하였습니다. 리플(XRP)이 채택했던 접근 방식은 참가자가 직접 편집할 수 있는 ‘입문자’ 회원 목록을 발행하여 사람들의 편집이 중요하지 않도록 만들거나 압도적인 참가자 비율을 통해 재생산되기를 기대하는 것입니다. 안타깝게도 일치하지 않는 목록들이 보안 보장성을 무효화하기 때문에 [Schwartz et al. 2014], 사용자들은 실제로 목록을 편집하는 것을 꺼리고 많은 권한이 결국 입문자 목록의 관리자에게 집중됩니다. 텐더민트가 취하는 또다른 접근법은 [Kwon 2014], 지분 증명을

토대로 멤버십을 수립하는 것입니다. 그러나 그러한 접근은 다시 한번 신뢰를 리소스 소유권과 연결시키게 됩니다. SCP는 신뢰할 수 있는 다른 참가자의 조합을 선택할 때 각 참가자에게 최대한의 자유를 주는 최초의 비잔틴 동의 프로토콜입니다.

3. 연합 비잔틴 합의 시스템 (FEDERATED BYZANTINE AGREEMENT SYSTEMS)

이 장은 연합 비잔틴 동의(FBA) 모델을 소개합니다. 비연합 비잔틴 동의와 마찬가지로, FBA는 트랜잭션 원장이나 인증서 트리와 같은 복제된 상태를 업데이트하는 문제를 다룹니다. 어떤 업데이트를 적용할지를 동의함으로써 노드는 모순적이거나 양립할 수 없는 상태를 피할 수 있습니다. 업데이트 간 종속성을 추론할 수 있는 고유한 슬롯으로 각 업데이트를 식별합니다. 예를 들어, 슬롯은 순차적으로 적용된 로그에서 연속적으로 번호가 매겨진 위치일 수 있습니다.

FBA 시스템은 노드가 슬롯 내용에 동의하도록 하는 합의 프로토콜을 실행합니다. 노드 v 는 i 가 의존하는 모든 슬롯에 업데이트를 안전하게 적용했을 때 슬롯 i 에 업데이트 x 를 안전하게 적용할 수 있으며 또한 올바르게 작동하는 모든 노드가 궁극적으로는 슬롯 i 에 대한 x 에 동의할 것으로 추정합니다. 이 때 우리는 v 가 슬롯 i 에 대해 x 를 표면화 했다고 말합니다. 외부에서는 비가역적 방식을 통해 표면화된 가치에 반응할 수 있으며 이 경우 노드는 추후에 의견을 변경할 수 없습니다.

FBA가 직면한 문제는 바로 악의적인 당사자들이 여러번 가입할 수 있고 정직한 노드보다 많을 수 있다는 것입니다. 따라서 기존의 다수결 기반 쿼럼은 효과가 없습니다. 대신 FBA는 각 노드가 쿼럼 슬라이스 (Quorum Slice)라고 부르는 요소를 선택하도록 하여 탈중앙화된 방식으로 쿼럼을 결정합니다. 다음 절에서는 슬라이스를 기반으로 쿼럼을 정의합니다. 이후의 절에서는 몇 가지 예시와 토의를 제공합니다. 마지막으로 합의 프로토콜이 달성 목표로 삼아야 하는 안전 및 활성의 주요 속성을 정의합니다.

3.1. 쿼럼 슬라이스

합의 프로토콜에서 노드는 슬롯에 대한 스테이트먼트를 단언하는 메시지를 교환합니다. 이러한 주장들은 위조될 수 없다고 가정하며, 노드가 공개키로 명명되고 메시지에 디지털 서명하는 경우 보장될 수 있습니다. 한 노드가 충분한 수의 노드들이 어떠한 정보를 주장하고 있다는 것을 확인했을 경우, 작동중인 노드가 해당 정보와 절대 모순되지 않을 것이라 가정합니다. 이러한 충분 집합을 쿼럼 슬라이스, 또는 보다 간결하게 슬라이스라고 부릅니다. 노드 오류가 발생한 상황에서도 진행을 허용하기 위해 노드는 여러 슬라이스를 가질 수 있으며 각각의 슬라이스는 단독으로 충분히 스테이트먼트를 확신시킬 수 있습니다. 높은 수준에서 FBA 시스템은 각각 하나 이상의 슬라이스를 선택한 느슨한 노드 연합을 포함합니다. 더욱 공식적으로는

정의 (FBAS). 연합 비잔틴 동의 시스템(Federated Byzantine Agreement System, FBAS)은 $\langle V, Q \rangle$ 쌍으로, 일련의 노드 V 와 쿼럼 함수 $Q: V \rightarrow 2^{\{\emptyset\}}$ 를 포함하며 이 때 함수 Q 는 각 노드에 대해 하나 이상의 쿼럼 슬라이스를 명시하고 노드는 자신의 쿼럼 슬라이스들 모두에 포함됩니다. 즉 $\forall v \in V, \forall q \in Q(v), v \in q$ 입니다. (2^X 는 X 의 멱집합입니다.)

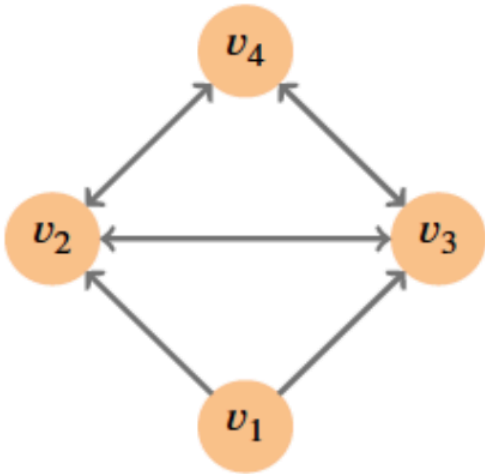
정의 (FBAS). 연합 비잔틴 합의 시스템(Federated Byzantine Agreement System, FBAS)은 $\langle V, Q \rangle$ 로 정의되며, 여기서 V 는 노드들의 집합이고, Q 는 각 노드에 대해 하나 이상의 쿼럼 슬라이스를 지정하는 함수이다. 즉, $Q: V \rightarrow 2^{\{V \setminus \{\emptyset\}\}}$ 이며, 모든 노드는 자신이 속한 모든 슬라이스에 반드시 포함된다 — 즉, $\forall v \in V, \forall q \in Q(v), v \in q$.

(참고: 2^X 는 X 의 멱집합(powerset)을 의미한다.)

정의 (쿼럼, quorum). FBAS $\langle V, Q \rangle$ 에서 노드의 부분집합 $U \subseteq V$ 가 쿼럼(quorum)이 되기 위한 조건은 다음과 같다: $U \neq \emptyset$ 이고, 모든 $v \in U$ 에 대해 $\exists q \in Q(v)$ 이며 $q \subseteq U$ 가 될 수 있도록 합니다. 즉, U 는 각 구성원이 자신의 슬라이스 중 하나를 포함하는 집합일 때 쿼럼이라 한다.

쿼럼은 합의를 이루기에 충분한 노드 집합입니다. 쿼럼 슬라이스는 쿼럼의 부분집합으로, 특정 합의 노드에게 합의를 확신시킵니다. 쿼럼 슬라이스는 쿼럼보다 작을 수 있습니다. 그림 2에 제시된 4노드 시스템을 생각해 보겠습니다. 각 노드에는 단일 슬라이스가 있고 화살표가 해당 슬라이스의 다른 멤버를 가리키고 있습니다. 노드 v_1 의 슬라이스 $\{v_1, v_2, v_3\}$ 은 v_1 에 스테이트먼트를 확신시키기에 충분합니다. 그러나 v_2 및 v_3 의 슬라이스는 v_4 를 포함하며, 이는 v_2 와 v_3 모두 v_4 의 합의 없이는 스테이트먼트를 단언할 수 없음을 의미합니다. 따라서 v_4 의 참여 없이는 합의가 불가능하며, v_1 을 포함하는 유일한 쿼럼은 모든 노드를 포함하는 집합 $\{v_1, v_2, v_3, v_4\}$ 입니다.

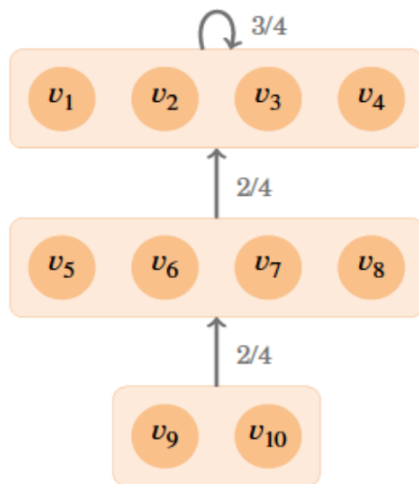
통상적으로 비연합 비잔틴 동의에서는 모든 노드가 동일한 슬라이스를 수락해야 합니다. 즉, $\forall v_1, v_2, Q(v_1) = Q(v_2)$ 을 의미합니다. 모든 멤버가 모든 슬라이스를 허용하기 때문에 통상적인 시스템은 슬라이스와 쿼럼을 구분하지 않습니다. 이 때 단점은 멤버십과 쿼럼이 반드시 미리 지정됨으로 인해 공개 멤버십과 탈중앙화된 통제가 불가능하다는 것입니다.



$$Q(v_1) = \{\{v_1, v_2, v_3\}\}$$

$$Q(v_2) = Q(v_3) = Q(v_4) = \{\{v_2, v_3, v_4\}\}$$

그림 2. v_1 의 쿼럼 슬라이스는 v_4 가 없는 쿼럼이 아닙니다.



최상위 계층: 슬라이스는 $\{v_1, v_2, v_3, v_4\}$ 중 3개이며 자기 자신을 포함합니다

중간 계층: 슬라이스는 자기 자신과 최상위 계층 노드 중 2개입니다

리프 계층: 슬라이스는 자기 자신과 중간 계층 노드 중 2개입니다

그림 3. 계층화된 쿼럼 구조 예시

PBFT [Castro and Liskov 1999]와 같은 기존 시스템에는 일반적으로 $3f + 1$ 개의 노드가 있으며 그 중 $2f + 1$ 이 쿼럼을 구성합니다. 여기서 f 는 시스템이 감당할 수 있는 최대 비잔틴 오류(임의로 작동하는 노드를 뜻함)를 나타냅니다.

본 백서에서 소개한 FBA는 더 넓은 범위의 설정을 수용하기 위해 비잔틴 동의를 일반화합니다. FBA의 핵심 혁신은 각 노드 v 로 하여금 고유한 쿼럼 슬라이스 세트 $Q(v)$ 를 선택할 수 있도록 하는 것입니다. 따라서 시스템 전체의 쿼럼은 각 노드의 개별 결정에서 발생합니다. 노드는 평판 또는 금융 협정을 비롯한 임의의 기준에 따라 슬라이스를 선택할 수 있습니다. 일부 설정에서는 어떠한 개별 노드도 시스템의 모든 노드에 대해 완전한 지식을 가질 수 없지만 합의는 여전히 가능해야 합니다.

3.2. 예시 및 토의

그림 3은 노드마다 다른 슬라이스 세트를 보유한 계층화된 시스템의 예를 제시하는데, 이는 FBA에서만 가능합니다. $v_1, \dots, v_4$ 로 구성된 최상위 계층은 $f = 1$ 인 PBFT 시스템과 같은 구조를 지닙니다. 즉 다른 3개 노드에 접근이 가능하고 이들이 제대로 작동하는 한 비잔틴 오류를 1번 수용할 수 있습니다. 노드 $v_5, \dots, v_8$ 은 중간 계층을 구성하며 서로 의존하는 것이 아니라 최상위 계층에 의존합니다. 중간 계층 노드에 대한 슬라이스

이스를 형성하려면 2개의 최상위 계층 노드만 필요합니다. (최상위 계층은 최대 1번의 비잔틴 오류를 가정하므로 전체 시스템에 오류가 발생하지 않는 한 2개의 최상위 계층 노드가 모두 오류일 수는 없습니다.) 노드 v9 및 v10은 슬라이스가 2개의 중간 계층 노드를 포함하는 리프 계층에 있습니다. v9 및 v10은 {v5, v6} 및 {v7, v8} 등의 분리된 슬라이스를 선택할 수 있습니다. 그럼에도 불구하고 이들은 모두 간접적으로 최상위 계층에 의존합니다.

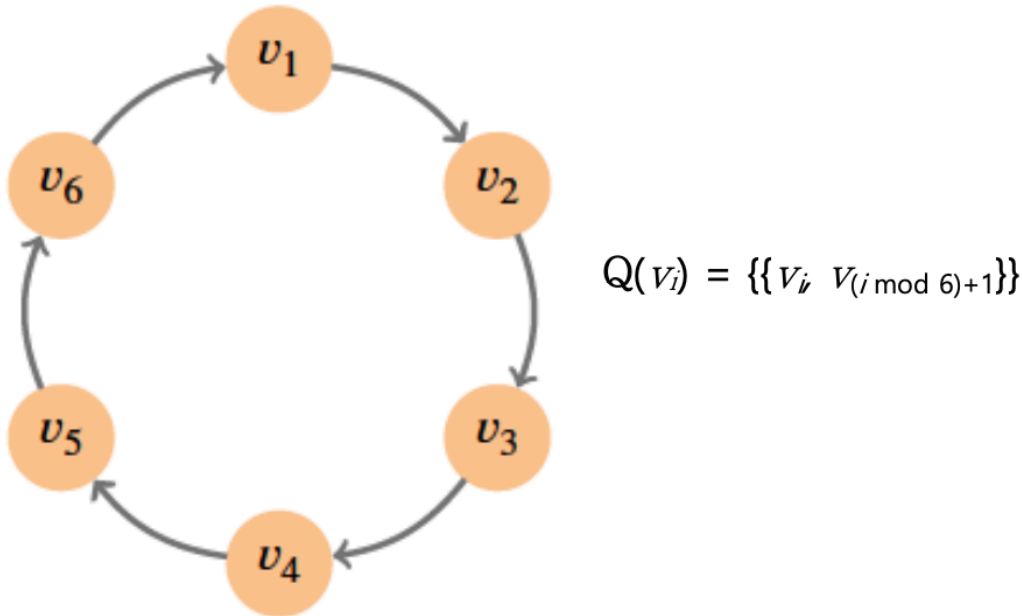


그림 4. 순환적 쿼럼 구조 예시

실사례에서 최상위 계층은 널리 알려지고 신뢰할 수 있는 금융 기관 4개 ~ 수십 개로 구성될 수 있습니다. 최상위 계층의 크기가 커짐에 따라 멤버십에 대한 정확한 합의가 없을 수 있지만 대부분의 당사자가 최상위 계층에 대한 개념 간에 일치하는 부분이 상당히 있을 것입니다. 또한 예를 들어 각 국가 또는 지리적 지역마다 1개씩 등, 다수의 중간 계층을 구상할 수 있습니다.

이 계층 구조는 도메인 간 네트워크 라우팅과 유사합니다. 오늘날 인터넷은 네트워크 쌍 간의 개별 피어링 및 트랜짓 관계에 의해 함께 유지됩니다. 중앙 당국은 이러한 조치를 지시하거나 중재하지 않습니다. 그러나 이러한 쌍성(pairwise) 관계는 사실상 1계층 ISP라는 개념을 만들기에 충분했습니다[Norton 2010]. 인터넷 접근성은 방화벽으로 인해 어려움을 겪지만 트랜짓 접근성은 거의 완전합니다. 예를 들어 방화벽이 The New York Times를 차단했으나 Google을 허용할 때 Google이 The New York Times에 연결할 수 있다면 The New York Times는 트랜짓을 거쳐 연결할 수 있습니다. 트랜짓 접근성은 웹 사이트에 대한 유용성 측면에서 제한적일 수 있으나, 합의에 있어서 매우 중요합니다. 이에 상응하는 예는 Google이 The New York Times가 수락하는 경우에만 스테이트먼트를 수락하는 것입니다.

쿼럼 슬라이스를 네트워크 접근성, 그리고 쿼럼을 트랜짓 접근성과 각각 유사하다고 간주해 보면 인터넷의 거의 완전한 트랜짓 접근성은 FBA 또한 전 세계적인 합의를 보장할 수 있음을 시사합니다. 여러 면에서 합의는 도메인 간 라우팅보다 쉬운 문제입니다. 트랜짓은 리소스를 소비하고 비용이 들지만 슬라이스의 포함은 디지털 서명을 확인하기만 하면 됩니다. 따라서 FBA 노드는 포괄성 측면에서 오류를 범함으로써 일반적으로 피어링 및 트랜짓 환경에서 볼 수 있는 것보다 더 큰 상호 의존성과 중복성을 가진 보수적인 슬라이스를 구성할 수 있습니다.

중앙화된 합의에서는 불가능한 또 다른 예는 그림 4에 표시된 것과 같은 순환적 종속성 구조입니다. 이러한 주기는 의도적으로 발생할 가능성은 낮으나 개별 노드가 자체 슬라이스를 선택하면 전체 시스템이 종속성 주기를 포함하게 될 수 있습니다. 더 중대한 요점은 바로 전통적인 비잔틴 동의와 비교할 때 FBA 프로토콜이 훨씬 더 다양한 쿼럼 구조를 처리해야 한다는 것입니다.

3.3 안전 및 활성

우리는 노드를 올바른 노드와 불량 노드로 분류합니다. 올바른 노드는 합리적인 쿼럼 슬라이스(섹션 4.1에서 추후 설명)를 선택하고 모든 요청에 최종적으로 응답하는 등, 프로토콜에 순응합니다. 불량 노드는 그렇지 않습니다. 불량 노드는 비잔틴 오류를 겪습니다. 즉 임의로 행동합니다.

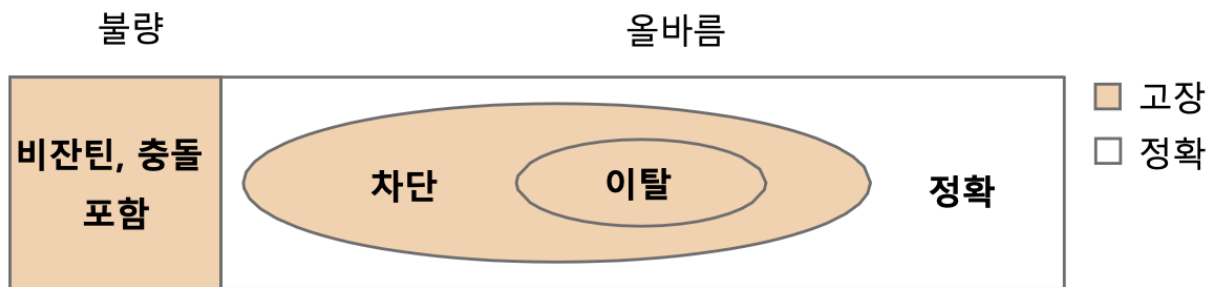


그림 5. 노드 오류의 벤 다이어그램

예를 들어, 불량 노드는 손상되었거나 소유자의 악의적인 소프트웨어 수정 또는 충돌을 겪었을 수 있습니다.

비잔틴 동의의 목표는 불량 노드가 존재하더라도 올바른 노드가 동일한 값을 표면화하도록 하는 것입니다. 이 목표에는 2가지 부분 목표가 있습니다. 첫째로는 노드가 동일한 슬롯에 대해 다른 값을 발산 및 표면화하는 것을 방지합니다. 둘째로 합의가 더 이상 불가능한 막다른 상태에서 차단되는 상황을 피하고 노드가 실제로 값을 표면화할 수 있도록 합니다. 이러한 속성에 대해 다음의 2가지 용어를 소개합니다.

정의 (안전, Safety). FBAS에 있는 노드 집합은 동일한 슬롯에 대해 어느 2개 집합도 다른 값을 표면화하지 않는 경우 안전을 확보합니다.

정의 (활성, Liveness). FBAS에 있는 노드는 고장(또는 불량) 노드의 참가 없이 새로운 값을 표면화할 수 있는 경우 활성을 확보합니다.

안전과 활성을 모두 확보한 올바른 노드를 정확한 노드라고 칭합니다. 정확하지 않은 노드들은 고장 노드입니다. 불량 노드는 모두 고장 났지만 올바른 노드도 불량 노드의 메시지를 무기한 기다리거나, 더 심각하게는 불량 노드의 잘못된 메시지에 의해 상태가 오염되어 고장 날 수 있습니다.

그림 5는 일어날 수 있는 노드 고장 유형들을 보여줍니다. 좌측에는 비잔틴 오류가 있으며 즉 불량 노드를 뜻 합니다. 오른쪽에는 2종류의 올바르지만 고장 노드에 해당하는 노드가 있습니다. 활성이 부족한 노드는 차단 되었다고 하고, 안전하지 않은 노드는 이탈되었다고 합니다. 안전을 위반하는 공격은 활성만을 위반하는 공격 보다 훨씬 강력하므로 이탈된 노드를 차단된 노드의 부분집합으로 분류합니다.

활성에 대한 정의는 노드가 새로운 가치를 표면화한다는 것이 아니라 할 수 있다는 점에서 강도가 약합니다. 따라서 합의가 영원히 가능한 상태로 유지되는 영구적 선점 상태를 인정하지만 네트워크는 중요한 메시지를 잘못된 방식으로 지연시키거나 재정렬하여 이를 지속적으로 방해합니다. 노드 고장에서 살아남는 순수하게 비동기식이며 결정론적인 시스템에서는 영구적 선점이 불가피합니다[Fischer et al. 1985]. 다행히 선점은 일시적입니다. 시스템은 언제든지 복구할 수 있으므로 선점은 노드 고장을 의미하지 않습니다. 프로토콜은 임의성[Ben-Or 1983; Bracha and Toueg 1985] 또는 메시지 지연 시간에 대한 현실적인 가정 [Dwork et al. 1988]을 통해 문제를 완화할 수 있습니다. 지연 가정은 실행 시간을 제한하거나 보다 효율적인 무작위 알고리즘에서 자주 요구하는 신뢰할 수 있는 딜러를 피하려는 경우에 더 실용적입니다[?]. 물론 안전이 아닌 종료만 메시지 타이밍을 기반으로 해야 합니다.

4. 최적 탄력성(OPTIMAL RESILIENCE)

노드가 안전과 활성을 확보하는지의 여부는 선택한 쿼럼 슬라이스, 불량 노드의 종류, 그리고 물론 구체적인 합의 프로토콜 및 네트워크 동작을 비롯한 여러 요인에 따라 다릅니다. 비동기식 시스템에서 흔히 나타나는 바와 같이, 우리는 네트워크가 결국에는 올바른 노드 간에 메시지를 전달한다고 가정하지만 그렇지 않을 경우 메시지를 임의로 지연시키거나 재정렬할 수 있습니다.

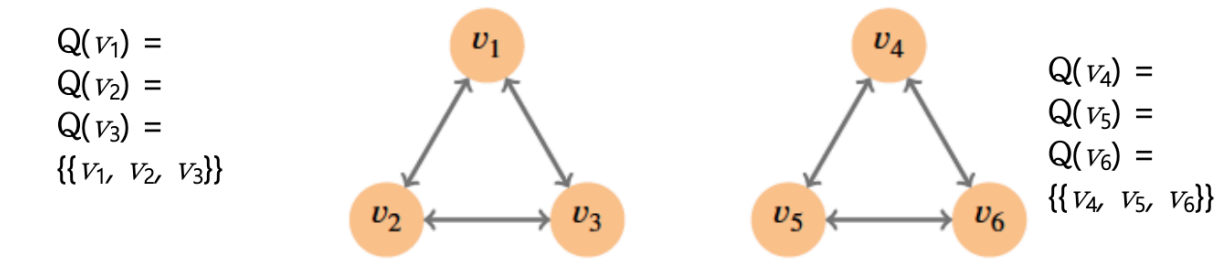


그림 6. 쿼럼 교차가 존재하지 않는 FBAS

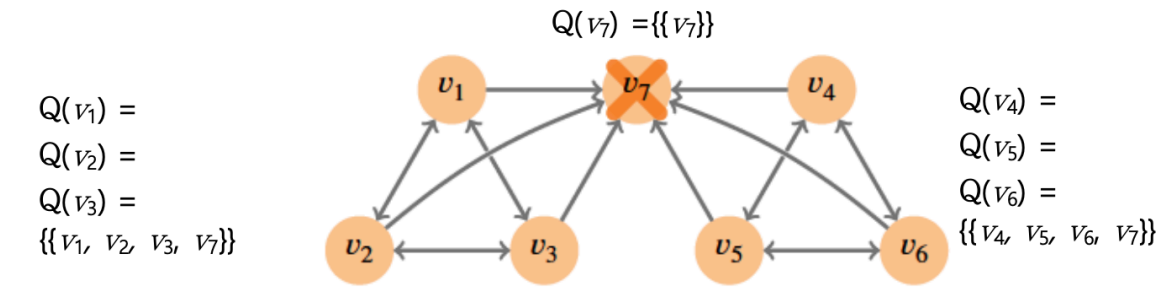


그림 7. 불량 노드 v7은 쿼럼 교차를 약화시킬 수 있습니다.

본 장은 다음의 질문에 대한 답변입니다. 특정 $\langle V, Q \rangle$ 및 V의 불량 특정 부분집합이 제시되었을 때 네트워크와 관계없이 연합 비잔틴 동의 프로토콜이 보장할 수 있는 최고의 안전과 활성은 무엇입니까? 먼저 안전을 보장하기 위해 반드시 필요한 특성인 쿼럼 교차에 대해 설명합니다. 다음으로 안전과 활성을 모두 보장할 수 있음에도 불구하고 고장 노드의 집합인 불필요한 집합이라는 개념을 소개합니다.

4.1 쿼럼 교차(Quorum Intersection)

프로토콜은 함수 Q 가 나타내는 쿼럼 슬라이스가 쿼럼 교차라고 하는 유효성 속성을 충족하는 경우에만 합의
를 보장할 수 있습니다.

정의(쿼럼 교차, Quorum Intersection). FBAS는 2개의 쿼럼이 노드를 공유하는 경우 쿼럼 교차를 확보
합니다. 즉, 모든 쿼럼 U_1 및 U_2 에 대해 $U_1 \cap U_2 \neq \emptyset$ 입니다.

그림 6은 Q 가 교차하지 않는 2개의 쿼럼 $\{v_1, v_2, v_3\}$ 및 $\{v_4, v_5, v_6\}$ 을 허용하며 쿼럼 교차가 존재하지
않는 시스템을 제시합니다. 분리된 쿼럼은 모순되는 스테이트먼트에 독립적으로 동의하여 시스템 전체의 합
의를 저해할 수 있습니다. 다수의 쿼럼이 존재할 때 그 중 교차하지 않는 2개가 존재한다면 쿼럼 교차에 실패
합니다. 예를 들어, 그림 6의 전체 노드 집합 $\{v_1, \dots, v_6\}$ 은 다른 2개와 교차하는 쿼럼이지만 그 2개는 서로
교차하지 않기 때문에 시스템에는 여전히 쿼럼 교차가 존재하지 않습니다.

어떠한 프로토콜도 쿼럼 교차가 없는 경우 안전을 보장할 수 없습니다. 이러한 구성은 메시지를 교환하지 않
는 2개의 서로 다른 FBAS 시스템으로 작동할 수 있기 때문입니다. 그러나 쿼럼 교차를 확보했다라도 불량
노드가 존재하는 경우 안전을 보장할 수 없습니다. 2개의 분리된 쿼럼이 있는 그림 6과 단일 노드 v_7 에서 2
개의 쿼럼이 교차하고 이 때 v_7 이 불량 노드인 그림 7을 비교해 보십시오. v_7 이 왼쪽 및 오른쪽 쿼럼에 대해
일치하지 않는 스테이트먼트를 작성하는 경우 효과는 분리된 쿼럼과 동일합니다.

실제로 불량 노드는 안전에 어떠한 기여도 하지 않기 때문에, 올바른 노드가 스스로 쿼럼 교차를 확보하지 않
는다면 어떤 프로토콜도 안전을 보장할 수 없습니다. 결국 안전을 위한 최악의 시나리오에서 불량 노드는 항
상 쿼럼을 완료하는 모든 가능한(모순적인) 스테이트먼트를 만들 수 있습니다. 불량 노드에서만 교차하는 2
개의 쿼럼은 불량 노드의 이중성으로 인해 다시 2개의 개별 FBAS 시스템처럼 작동할 수 있게 됩니다. 요약
하자면 $\text{FBAS} \langle V, Q \rangle$ 는 $\langle V, Q \rangle$ 가 V 및 Q 의 모든 슬라이스에서 B 의 노드를 삭제한 후 쿼럼 교차를 확보
한 경우에만 노드 $B \subseteq V$ 에 해당하는 노드 집합을 통해 비잔틴 오류를 버릴 수 있게 됩니다. 더욱 공식적으로
는

정의(삭제, Delete). $\langle V, Q \rangle$ 가 FBAS이고 $B \subseteq V$ 가 노드 집합이면 $\langle V, Q \rangle^B$ 로 작성된 $\langle V, Q \rangle$ 에서 B 를
삭제하는 것은 수정된 FBAS $\langle V \setminus B, Q^B \rangle$ 를 컴퓨팅하는 것을 의미하며 이 때 $Q^B(v) = \{ q \setminus B \mid q \in Q(v) \}$ 입니다.

$Q(v)$ 가 쿼럼 교차를 위반하지 않도록 하는 것은 각 노드 v 의 책임입니다. 이를 달성하는 하나의 방법은 큰 쿼
럼으로 이어지는 보수적인 슬라이스를 선택하는 것입니다. 물론 악의적인 v 는 의도적으로 $Q(v)$ 를 선택하여
쿼럼 교차를 위반할 수 있습니다. 그러나 악의적인 v 는 $Q(v)$ 값에 대해 거짓말을 하거나 $Q(v)$ 를 무시하여 임
의적인 주장을 할 수도 있습니다. 요약하자면 v 가 불량할 때는 $Q(v)$ 값이 의미가 없습니다. 그렇기 때문에 안
전을 위한 필수 속성인 불량 노드 삭제 후 올바른 노드의 쿼럼 교차는 불량 노드의 슬라이스에 영향을 받지 않
습니다.

그림 6이 쿼럼 교차가 있는 3개의 노드 FBAS v_1, v_2, v_3 에서 쿼럼 교차가 없는 6개의 노드 FBAS로 발전
했다고 가정합시다. v_4, v_5, v_6 합류 시 이들은 악의적으로 쿼럼 교차를 위반하는 슬라이스를 선택하며 어
떠한 프로토콜도 V 의 안전을 보장할 수 없습니다. 다행히 불량 노드를 삭제하여 $\langle V, Q \rangle \setminus \{v_4, v_5, v_6\}$ 를 얻
으면 쿼럼 교차가 복원되며 이는 최소 $\{v_1, v_2, v_3\}$ 에는 안전이 확보되었다는 의미입니다. 이 때 삭제는 최
적의 안전성을 설명하기 위한 개념적인 요소라는 점에 유의하십시오. 프로토콜은 v_4, v_5, v_6 의 불량 여부를
인지할 필요 없이 v_1, v_2, v_3 에 대한 안전성을 보장해야 합니다.

4.2 불필요한 집합(Dispensable sets, DSets)

불필요한 집합 내지는 DSet의 개념을 통해 노드 슬라이스 선택의 오류 허용을 포착합니다. 비공식적으로 DSet 외부 노드의 안전과 활성화는 DSet 내부 노드의 동작에 관계없이 보장될 수 있습니다. 다시 말해서 최적의 탄력성을 가진 FBAS에서 단일 DSet이 모든 불량 노드를 포함하는 경우 모든 고장 노드도 포함하며 반대로 DSet 외부의 모든 노드는 정확합니다. 예를 들어 노드가 $3f + 1$ 개이고 쿼럼 크기가 $2f + 1$ 인 중앙화된 PBFT 시스템에서 f 이하의 노드는 DSet을 구성합니다. PBFT는 실제로 f 번의 비잔틴 오류까지 견디므로 최적의 견고성을 지닙니다.

그림 3의 덜 일반적인 예를 살펴보면 1개의 최상위 계층 노드가 시스템의 나머지 부분에 영향을 주지 않고도 오류가 발생할 수 있으므로 $\{v_1\}$ 는 DSet입니다. 다른 노드가 정확성을 위해 v_9 에 의존하지 않기 때문에 $\{v_9\}$ 또한 DSet입니다. $\{v_6, \dots, v_{10}\}$ 은 DSet입니다. v_5 와 최상위 계층 모두 해당 5개 노드에 의존하지 않기 때문입니다. $\{v_5, v_6\}$ 은 v_9 및 v_{10} 에 대한 슬라이스이므로 DSet이 아니며 따라서 완전히 악의적인 경우 v_9 및 v_{10} 에게 거짓 정보를 전달하여 서로 또는 시스템의 나머지 부분과 일치하지 않는 주장을 설득시킬 수 있습니다.

오작동하는 DSet이 다른 노드의 정확성에 영향을 미치지 않도록 하려면 두 가지 속성을 유지해야 합니다. 안전 목적 상, DSet을 삭제해도 쿼럼 교차가 악화시킬 수 없습니다. 활성화 목적 상, DSet은 다른 노드가 기능하는 쿼럼을 거부하도록 할 수 없습니다. 이는 다음과 같은 정의로 이어집니다.

정의(DSet). $\langle V, Q \rangle$ 를 FBAS로, $B \subseteq V$ 를 노드의 집합으로 합니다. B 는 다음과 같은 경우에만 불필요한 집합 또는 DSet이라고 합니다.

- (1) (B 에도 불구하고 쿼럼 교차) $\langle V, Q \rangle^B$ 는 쿼럼 교차를 보유하며,
- (2) (B 에도 불구하고 가용성 확보) $V \setminus B$ 는 $\langle V, Q \rangle$ 의 쿼럼이거나 $B = V$ 입니다.

B 에도 불구하고 쿼럼 가용성은 요청 응답을 거부하고 다른 노드의 진행을 차단하는 B 의 노드로부터 보호합니다. B 에도 불구하고 쿼럼 교차는 반대, 즉 다른 노드가 동일한 슬롯에 대해 일관성 없는 값을 표현화할 수 있도록 모순적인 주장을 하는 B 의 노드로부터 보호합니다. 노드는 반드시 슬라이스 선택 시 존재하는 2가지 위협의 균형을 맞춰야 합니다.

올바름 /불량	다른 노드와 독립적인 노드의 로컬 속성 (슬라이스 선택의 유효성을 제외하고).
온전 / 오염	쿼럼 슬라이스와 특정 불량 노드 집합을 반영한 노드의 속성 오염 노드는 불량 노드이거나 지나치게 많은 수의 불량 노드에 직/간접적으로 의존합니다.
정확 /고장	쿼럼 슬라이스, 구체적인 프로토콜, 그리고 실제 네트워크 동작을 반영한 노드의 속성. 합의 프로토콜의 목표는 모든 온전한 노드의 정확성을 보장하는 것입니다.

그림 8. FBAS 노드의 핵심 속성

다른 모든 것들이 동일할 때, 슬라이스가 클수록 더욱 큰 쿼럼과 더욱 높은 중복을 초래하며, 즉 더욱 적은 수의 고장 노드로 구성된 집합 B가 삭제되었을 때도 쿼럼 교차를 약화시킵니다. 반면에 더 큰 슬라이스는 고장 노드가 포함될 가능성이 높아 쿼럼 가용성이 위태로워집니다. 모든 불량 노드를 포함하는 가장 작은 DSet은 올바른 노드도 포함할 수 있으며, 이는 불량 노드의 집합이 충분히 큰 경우 올바른 노드도 고장날 수 있다는 사실을 반영합니다. 예를 들어 그림 3을 보면 v5와 v6을 포함하는 가장 작은 DSet이 {v5, v6, v9, v10}인 것을 볼 수 있습니다. 모든 노드의 집합인 V는 항상 DSet이며 FBAS $\langle V, Q \rangle$ 는 V에도 불구하고 무의미하게 쿼럼 교차를 확보하며, 특수한 경우 V에도 불구하고 쿼럼 가용성을 확보합니다. 특수한 사례를 발생시킬 수 있는 동기는 충분히 많은 수의 불량 노드가 제공된 경우 V는 모든 불량 노드를 포함하는 가장 작은 DSet이 될 수 있다는 점으로, 이는 완전한 시스템 고장을 제외한 어떤 것도 프로토콜이 완벽하게 보장할 수 없음을 나타냅니다. FBAS의 DSet들은 쿼럼 함수 Q에 의해 우선순위로 결정됩니다. 어떤 노드가 불량하고 올바른지는 머신이 손상되는 등 런타임 동작에 따라 결정됩니다. 우리가 주목하는 DSet은 모든 불량 노드를 아우르는 것들로, 보장이 불가능한 노드들로부터 정확한 것으로 보장되어야 할 노드들을 구분할 수 있도록 돕습니다. 이를 위하여 다음과 같은 용어들을 소개합니다.

정의 (온전함). FBAS에 있는 노드 v 는 DSet B가 모든 불량 노드를 포함하며 $v \notin B$ 인 경우에만 온전합니다.

정의 (오염, Befouled). FBAS의 노드 v 는 온전하지 않은 경우에만 오염된 상태입니다.

오염 노드 v 는 v 자신은 올바르다 해도 진행을 방해하거나 상태를 오염시키기에 충분한 수의 고장 노드로 둘러싸여 있습니다. 어떠한 FBAS도 오염 노드의 정확성을 보장할 수 없습니다. 그러나 최적의 FBAS는 모든 온전한 노드가 정확하게 유지되고 있음을 보장합니다. 그림 8은 노드의 핵심 속성들을 요약합니다. 다음의 정리들은 오염 노드의 집합이 항상 FBAS에서 쿼럼 교차를 가지는 DSet임을 제시함으로써 분석을 용이하게 합니다.

정리 1. U를 FBAS $\langle V, Q \rangle$ 의 쿼럼, $B \subseteq V$ 를 노드의 집합으로 가정하고, $U' = U \setminus B$ 라 할 때, 만약 $U' \neq \emptyset$ 이라면 U' 는 $\langle V, Q \rangle^B$ 의 쿼럼입니다.

증명. U가 쿼럼이므로, 모든 노드 $v \in U$ 는 $q \subseteq U$ 로 하는 $q \in Q(v)$ 를 가집니다. $U \subseteq U$ 이므로, 그에 따라 모든 $v \in U'$ 가 $\langle V \setminus B \rangle$ 인 $q \in Q(v)$ 를 가집니다. 삭제 표기법으로 재작성하면, $\forall v \in U', \exists q \in Q^B(v)$ 를 얻게 되며 이 때 $q \subseteq U'$ 가 되는 데, $U' \subseteq V \setminus B$ 이므로 U' 은 $\langle V, Q \rangle^B$ 의 쿼럼임을 의미합니다.

정리 2. 만약 B_1 과 B_2 가 쿼럼 교차가 적용된 FBAS $\langle V, Q \rangle$ 의 DSet인 경우, $B = B_1 \cap B_2$ 또한 DSet입니다.

증명. $U_1 = V \setminus B_1$ 이며 $U_2 = V \setminus B_2$ 라 하고 합니다. 만약 $U_1 = \emptyset$ 이라면, $B_1 = V$ 이며 B_2 (DSet)이므로 증명 완료입니다. 이와 유사하게 만약 $U_2 = \emptyset$ 이라면 $B_2 = V$ 이므로 증명 완료됩니다. 그 외에, 쿼럼 가용성에 따라 DSet B_1 과 B_2 에도 불구하고 U_1 과 U_2 는 $\langle V, Q \rangle$ 의 쿼럼이라는 점에 주의하십시오. 이는 두 쿼럼의 합집합 또한 쿼럼이라는 정의를 따릅니다. 따라서 $V \setminus B = U_1 \cup U_2$ 는 쿼럼이며 B에도 불구하고 쿼럼 가용성이 존재합니다.

이제 B에도 불구하고 쿼럼 교차를 증명해야 합니다. U_1 와 U_2 를 $\langle V, Q \rangle$ 의 어느 두 쿼럼이라고 가정합니다. $U = U_1 \cup U_2 = V \setminus B_1$ 이라 가정합니다. $\langle V, Q \rangle$ 의 쿼럼 교차에 의해, $U_1 \cap U_2 \neq \emptyset$ 입니다. 그러나 정리 1에 따라, $U' = U \setminus B_1$ 는 반드시 $\langle V, Q \rangle^B$ 에 포함된 쿼럼이어야 합니다. 이제 U_1, B_1 와 U_2, B_2 가 둘 다 공집합이 될 수는 없음을 고려합니다. 그렇지 않을 경우 $U_1 \cup B_1 = U_2 \cup B_2$ 가 공집합이 될 것이기 때문입니다. 따라서 정리 1에 따라, $U_1 \setminus B_1$ 가 $\langle V, Q \rangle^{B_1} = \langle V, Q \rangle^{B_1}$ 에 포함된 쿼럼이거나, 또는 $U_2 \setminus B_2$ 가 $\langle V, Q \rangle^{B_2} = \langle V, Q \rangle^{B_2}$ 에 포함된 쿼럼이거나, 둘 다 해당할 수 있습니다. 전자의 경우 만약 $U_1 \setminus B_1$ 이 $\langle V, Q \rangle^{B_1}$ 에 포함된 쿼럼이라면, $\langle V, Q \rangle^{B_1}$ 의 쿼럼 교차에 따라, $(U_1 \setminus B_1) \cap U_2 \neq \emptyset$ 이 될 것입니다. $(U_1 \setminus B_1) \cup (U_2 \setminus B_2)$ 는 $\langle V, Q \rangle^B$ 이므로, $U_1 \neq \emptyset$ 를 따르고, U_1, B_2 를 $\langle V, Q \rangle^{B_2}$ 의 쿼럼에 해당되게 합니다. 유사한 논거에 따라, U_1, U_2 는 반드시 $\langle V, Q \rangle^B$ 에 포함된 쿼럼이어야 합니다. 그러나 B에도 불구하고 쿼럼 교차는 $(U_1 \setminus B_2) \cap (U_2 \setminus B_2) \neq \emptyset$ 이라는 것을 알리며, 이는 $U_1 \cap U_2 \neq \emptyset$ 인 경우에만 가능합니다.

정리 3. 쿼럼 교차를 가지는 FBAS에서, 오염 노드 집합은 DSet입니다.

증명. $B_{\min}$ 을 전체 불량 노드를 포함하는 모든 DSet의 교차라고 합시다. 이는 $v \in B_{\min}$ 인 경우에만 노드가 온전하다는 온전함의 정의를 따릅니다. 따라서 $B_{\min}$ 은 정확하게 오염 노드 집합에 해당합니다. 정리 2에 따라, DSet은 교집합에서 닫힌 상태이므로 $B_{\min}$ 은 또한 DSet에 해당합니다.

5. 연합 투표

본 장은 FBAS 노드가 스테이트먼트에 동의하기 위해 사용할 수 있는 연합 투표 기법을 설명합니다. 높은 수준에서, 어느 스테이트먼트 a에 대해 합의하는 절차는 두 세트의 메시지를 노드가 교환하는 과정을 필요로 합니다. 먼저 노드는 a에 대해 투표합니다. 만약 해당 투표가 성공한 다음엔, 노드는 a를 확정하고 첫 번째 투표가 성공하였다는 사실에 입각하여 두 번째 투표를 유효하게 진행합니다. 각 노드의 관점에서, 2회의 메시지는 스테이트먼트 a에 대한 합의를 미상, 승인, 확정 3단계로 나눕니다. (이 패턴은 3단계 실행에서 비롯된 것입니다 [Skeen and Stonebraker 1983].) 최초에 a의 상태는 노드 v에게 완전히 미상으로 존재합니다. a는 참, 거짓, 심지어 영구적 불확정 상태에 갇힐 수도 있습니다. 만약 첫 번째 투표가 성공하는 경우, v는 a를 수락할 수 있습니다. 2개의 온전한 노드가 모순된 스테이트먼트를 수락하는 경우는 존재하지 않으므로, v가 온전하며 a를 수락하는 경우 a는 거짓이 될 수 없습니다. 그러나 2가지 이유로 인해 v가 a를 수락하는 것만으로는 v가 a에 작용할 수 없습니다. 먼저 v가 a를 수락했다는 사실은 모든 온전한 노드에서 가능한 건 아니며, a는 다른 노드의 경우였다면 갇힐 수도 있습니다. 둘째로 만약 v가 오염되었다면 a를 수락하는 것은 아무 의미도 없습니다. a는 온전한 노드의 경우였다면 거짓일 수도 있습니다. 그러나 v는 알 수 없지만 v가 오염되었다 하더라도 시스템에는 여전히 올바른 노드의 쿼럼 교차가 존재할 수 있으며, 이 경우 최적의 안전을 위해 v는 a에 대해 더욱 큰 수준의 확신이 필요합니다. 두 번째 투표를 진행하는 것은 2가지 문제를 모두 제기합니다. 만약 두 번째 투표가 성공하는 경우, v는 확정된 단계로 이동하여 최종적으로 a를 사실로 간주하고 작용합니다. 이후 몇 개 절에서는 연합 투표 절차에 대해 자세히 다룹니다. 투표는 스테이트먼트가 고착되었을 가능성을 배제하지 않으므로, 제5, 6장에서는 이들을 다루는 방법을 논의합니다. 제6장은 연합 투표를 온전한 노드에 대해 슬롯이 갇힐 가능성을 피하는 합의 프로토콜로 전환할 것입니다.

5.1 개방된 멤버십으로 투표 진행

비잔틴 합의 시스템의 정확한 노드는 다른 정확한 노드들이 a 에 모순되는 스테이트먼트에 절대 합의하지 않을 것이라는 사실을 아는 경우에만 스테이트먼트 a 를 처리합니다. 대부분의 프로토콜은 이러한 목적을 위해 투표를 이용합니다. 올바른 노드는 스테이트먼트 a 가 유효한 경우에만 투표합니다. 또한 올바른 노드도 투표를 절대 변경하지 않습니다. 따라서 중앙화된 비잔틴 동의에서는 올바른 노드 대다수를 포함하는 퀴럼이 투표한 경우 a 를 수락하는 것이 안전합니다. 스테이트먼트가 필요한 투표 수를 얻은 경우 이를 비준(ratified)되었다고 합니다.

연합 환경에서는 개방된 멤버십을 수용하도록 투표를 수정해야 합니다. 한 가지 차이점은 퀴럼이 더 이상 올바른 노드 대다수에 대응하지 않는다는 것입니다. 그러나 이 다수성 요건은 주로 올바른 노드의 퀴럼 교차를 보장하는 역할을 맡으며, 제4.1장에서 이미 FBA에 적용하였습니다. 개방된 멤버십의 또 다른 의미는 노드가 반드시 투표 절차의 일부로서 퀴럼을 구성하는 요소가 무엇인지 밝혀내야 한다는 것입니다. 퀴럼 발견을 구현하기 위해 프로토콜은 v 로부터 전달되는 모든 메시지에서 $Q(v)$ 를 명시해야 합니다.

정의 (투표, Vote). 노드 v 는 다음과 같은 경우에만 (추상적인) 스테이트먼트 a 에 투표합니다.

- (1) v 가 a 를 유효한 것으로 주장하며 v 가 수락한 모든 스테이트먼트에서 일관적이며,
- (2) v 가 절대로 a 에 반대하여 투표, 즉 a 에 모순되는 스테이트먼트에 투표한 적이 없음을 주장하며, 향후에도 v 는 a 에 반대하여 투표하지 않을 것으로 약속합니다.

정의 (비준, Ratify). U_a 는 U_a 의 모든 멤버가 a 에 투표한 경우에만 스테이트먼트 a 를 비준합니다. 노드 v 는 v 가 a 를 비준하는 퀴럼 U_a 의 멤버인 경우에만 a 를 비준합니다.

정리/4. 퀴럼 교차가 적용되어 있으며 불량 노드를 포함하지 않는 FBAS에서는 2개의 상호 모순적인 스테이트먼트 a 및 $\bar{a}$ 는 동시에 비준될 수 없습니다.

증명. 모순성에 따름. 퀴럼 U_1 가 a 를 비준하고 퀴럼 U_2 가 $\bar{a}$ 를 비준한다고 가정합니다. 퀴럼 교차에 따라 $\exists v \in U_1 \cap U_2$ 입니다. 해당 v 는 불법적으로 a 및 $\bar{a}$ 모두에 투표한 것이 분명하며, 불량 노드가 없다는 가정을 위반합니다.

정리/5. $\langle V, Q \rangle$ 를 B 에도 불구하고 퀴럼 교차가 적용된 FBAS로 하고, B 는 모든 불량 노드를 포함하는 것으로 가정합니다. v_1 과 v_2 를 B 에 없는 노드 2개로 하고, a 와 $\bar{a}$ 는 모순되는 스테이트먼트로 합니다. 만약 v_1 이 a 를 비준한다면 v_2 는 $\bar{a}$ 를 비준할 수 없습니다.

증명. 모순성에 따름. v_1 이 a 를 비준하고 v_2 는 $\bar{a}$ 를 비준한다고 합시다. 정의에 따라 a 를 비준하는 v_1 을 포함하는 퀴럼 U_1 과 $\bar{a}$ 를 비준하는 v_2 를 포함하는 퀴럼 U_2 가 존재해야 합니다. 정리 1에 따라, $U_1 \setminus B \neq \emptyset$ 이며 $U_2 \setminus B \neq \emptyset$ 이므로 둘은 모두 $\langle V, Q \rangle B$ 의 퀴럼이어야 하며, 이는 곧 $\langle V, Q \rangle B$ 에서 각각 a 와 $\bar{a}$ 를 비준했음을 의미합니다. 그러나 $\langle V, Q \rangle B$ 에는 퀴럼 교차가 적용되어 있고 불량 노드가 없으므로, 정리 4에 따라 a 와 $\bar{a}$ 는 동시에 비준될 수 없습니다.

정리/6. 퀴럼 교차를 가지는 FBAS의 온전한 노드 2개는 모순된 스테이트먼트를 비준할 수 없습니다.

증명. B 를 오염 노드의 집합이라고 합시다. 정리 3에 따라, B 는 DSet입니다. DSet의 정의에 따라, $\langle V, Q \rangle$ 는 B 에도 불구하고 퀴럼 교차가 허용됩니다. 정리 5에 따라, B 에 없는 노드 2개는 모순된 스테이트먼트를 비준할 수 없습니다.

5.2 차단 집합

중앙화된 합의에서, 활성화는 시스템의 이분법적 속성(전부 아니면 전무)입니다. 만장일치로 올바른 쿼럼이 존재하거나, 또는 불량 노드로 인해 시스템의 나머지 부분이 새 스테이트먼트를 수락하지 못할 수 있습니다. 이와 대조적으로 FBA에서는 활성화가 노드마다 다를 수 있습니다. 예를 들어, 그림 9의 계층 쿼럼 예시에서 만약 중간 계층 노드 v_6, v_7, v_8 충돌이 발생하면 리프 계층은 차단되지만 최상위 계층과 노드 v_5 는 계속해서 활성을 유지합니다.



그림 9. v_4 는 $\bar{a}$ 에 투표하였으며, 이는 비준된 스테이트먼트 a 에 모순됩니다.

FBA 프로토콜은 $Q(v)$ 가 올바른 노드만을 포함하는 쿼럼 슬라이스를 최소 1개 이상 포함하는 경우에만 노드 v 의 활성을 보장할 수 있습니다. 고장 노드로 구성된 집합 $B \subseteq V$ 가 v 에 있는 슬라이스들의 멤버를 최소 각각 1개씩 포함하는 경우 이 속성을 위반할 수 있습니다. 이를 v -차단 B 집합이라고 부르는데, v 에 의한 진행을 차단할 수 있기 때문입니다.

정의(v -차단, v -blocking). $v \in V$ 를 FBAS $\langle V, Q \rangle$ 의 노드라고 합시다. $B \subseteq V$ 인 집합은 B 가 v 의 모든 슬라이스와 중복되는 경우, 즉 $\forall q \in Q(v), q \cap B \neq \emptyset$ 인 경우에만 v -차단에 해당합니다.

정리 7. $B \subseteq V$ 를 FBAS $\langle V, Q \rangle$ 에 포함된 노드의 집합이라고 가정합니다. $\langle V, Q \rangle$ 는 B 가 어떠한 $v \in V \setminus B$ 에 대해 v -차단에 해당하지 않는 경우에만 B 에도 불구하고 쿼럼 가용성이 적용됩니다.

증명. " $\forall v \in V \setminus B, B$ 는 v -차단이 아님"은 " $\forall v \in V \setminus B, \exists q \in Q(v)$ 이며 이 때 q 가 $q \in V \setminus B$ "의 경우와 동일합니다. 쿼럼의 정의에 따라 후자는 $V \setminus B$ 가 쿼럼이거나 $B = V$ 인 경우에만 B 에도 불구하고 쿼럼 가용성이 존재함의 정확한 정의가 적용됩니다.

필연적인 결과로, 오염 노드의 DSet은 온전한 v 의 경우 절대 v -차단에 해당하지 않습니다.

5.3. 수락 스테이트먼트

온전한 노드 v 가 어떠한 스테이트먼트를 비준하였음을 알게 되는 경우, 정리 6에 따라 다른 온전한 노드들이 모순되는 스테이트먼트를 비준하지 않을 것이라는 점을 v 에게 알리게 됩니다. 이는 v 로 하여금 a 를 수락할 충분조건에 해당하나, 필요조건으로는 지정할 수 없습니다. 어느 스테이트먼트를 비준하는 경우 이에 대한 투표가 필요하며, 일부 노드는 이미 모순된 스테이트먼트에 대해 투표했을 수 있습니다. 예를 들어 그림 9에서 v_4 는 다른 3개 노드들이 모순되는 스테이트먼트 a 를 비준했음을 인지하기 전 $\bar{a}$ 에 투표합니다. v_4 는 이제 a 에 투표할 수 없으나, 다른 노드들과 일관성을 유지하도록 a 를 수락하는 쪽이 선호됩니다.

중요한 것은 만약 노드 v 가 온전한 경우, 그 어떤 v -차단 집합인 B 도 오염 노드로만 구성될 수 없다는 점입니다. 이제 B 가 v -차단 집합이며 B 의 모든 멤버가 스테이트먼트 a 의 수락을 주장한다고 가정합니다. 만약 v 가 온전한 경우, B 의 멤버 중 최소 1개는 반드시 온전해야 합니다. 이 온전한 멤버는 a 의 수락에 대해 거짓을 말하지 않습니다. 따라서 a 는 참이며 v 가 수락할 수 있습니다. 물론 v 가 오염되었다면 a 는 참이 아닐 수 있습니다. 그러나 오염된 노드는 무엇이든 수락할 수 있으며 온전한 노드의 정확성에 전혀 영향을 끼치지 않을 수 있습니다.

정의 (수락, Accept). FBAS 노드 v 는 스테이트먼트 a 를 다음과 같은 경우에만 수락하는데, a 에 모순되는 스테이트먼트를 수락한 적이 없으며

- (1) $v \in U$ 인 쿼럼 U 가 존재하고 U 의 각 멤버가 a 에 투표했거나 a 를 수락하는 것으로 주장하거나,
- (2) 어느 v -차단 집합의 각 멤버가 a 를 수락하는 것으로 주장하는 경우입니다.

올바른 노드는 모순된 스테이트먼트에 투표할 수 없으나, 상기 2번 조건에 따라 노드는 어느 스테이트먼트에 투표한 뒤 나중에 모순된 스테이트먼트를 수락하는 것이 가능합니다.

정리 8. 쿼럼 교차가 적용된 FBAS 에 있는 어느 2 개의 온전한 노드는 모순된 스테이트먼트를 수락할 수 없습니다.

증명. $\langle V, Q \rangle$ 를 쿼럼 교차가 적용된 FBAS 라고 하며 B 를 오염 노드의 DSet이라고 합시다(정리 3에 의해 존재). 어느 온전한 노드가 스테이트먼트 a 를 수락한다고 가정합시다. a 를 수락하는 첫 번째 온전한 노드를 v 라고 합시다. v 가 a 를 수락하는 시점에는, B 에 있는 오염 노드들만이 이를 수락하겠다고 주장할 수 있습니다. 정리 7에 따른 필연적인 결과로, B 는 v -차단이 될 수 없으므로, 조건 1에 따라 v 는 a 를 수락해야 합니다. 따라서 v 는 모든 노드가 a 에 투표했거나 이를 수락한 쿼럼 U 를 식별하게 되며, v 는 a 를 수락하는 첫 번째 온전한 노드이므로 이는 반드시 $U \setminus B$ 의 모든 노드가 a 에 투표했음을 의미합니다. 달리 말하면 v 는 $\langle V, Q \rangle_B$ 에서 a 를 비준했습니다. 이를 일반화하면, $\langle V, Q \rangle$ 의 어떠한 온전한 노드에 의해 수락되는 모든 스테이트먼트는 반드시 $\langle V, Q \rangle_B$ 에서 비준됩니다. B 는 DSet이므로, $\langle V, Q \rangle_B$ 는 쿼럼 교차가 적용됩니다. 더불어 B 는 모든 불량 노드를 포함하므로, 정리 4에 따라 모순된 스테이트먼트들의 비준은 이루어지지 않습니다.

5.4. 수락으로는 불충분

안타깝게도 수락된 스테이트먼트들이 참이라고 가정하는 노드들은 연합 합의 프로토콜에서 보장하는 안전과 활성의 최적 수준에 미치지 못합니다. 이에 따라 안전과 활성에 관한 문제를 다루어야 합니다. 이해를 돕기 위해, 이러한 문제들이 왜 중앙화된 비잔틴 동의보다 FBA에서 더욱 까다로운지 설명하겠습니다.

5.4.1. 안전

유일한 쿼럼이 만장일치로 동의한, 즉 $\forall v, Q(v) = \{V\}$ 인 FBAS $\langle V, Q \rangle$ 를 가정합니다. 이는 안전에 있어서 보수적인 선택이어야 합니다. 즉 모두가 동의하지 않는 한 아무 일도 하지 않습니다. 그러나 모든 노드가 모든 v 에 대하여 v -차단에 해당하므로, 모든 노드는 단독으로 다른 노드를 설득시켜 임의로 스테이트먼트를 수락하도록 만들 수 있습니다.

문제는 수락된 스테이트먼트가 온전한 노드 사이에서만 안전하다는 것입니다. 하지만 제4.1장에서 논의된 바와 같이, 안전을 보장하기에 필요한 유일한 조건은 올바른 노드의 **쿼럼 교차**이며, 일부 올바른 노드가 오염된 경우에도 적용될 수 있습니다. 특히 $Q(v) = \{V\}$ 인 경우, $DSet$ 은 $\emptyset$ 과 V 만이 존재하며, 그 어떤 유형의 노드 오류라도 전체 시스템을 오염시킬 수 있음을 의미합니다. 이와 대조적으로 모든 $B \subseteq V$ 에도 불구하고 쿼럼 교차는 적용됩니다.

5.4.2. 활성

수락된 스테이트먼트의 또 다른 제한사항은 다른 온전한 노드가 이를 수락할 수 없을 수도 있다는 점입니다. 이 가능성은 수락된 스테이트먼트에 대한 의존성이 활성에 문제가 되도록 합니다. 만약 어느 노드가 스테이트먼트를 수락함으로써 스테이트먼트에 작용하는 경우, 다른 노드들은 이와 유사한 방식으로 진행하지 못할 수도 있습니다.

그림 10a에서, v_1 의 스테이트먼트 a 비준 및 수락을 도운 뒤 v_3 가 충돌하는 상황을 보도록 합시다. v_1 은 a 를 수락하나, v_2 와 v_4 는 수락할 수 없습니다. 구체적으로 v_2 의 관점에서 이 상황은 그림 10b와 구분할 수 없는 데, 해당 그림의 경우 v_3 가 $\bar{a}$ 에 투표하여 올바른 노드이나 응답이 느리고, v_1 은 불량 노드이며 $\bar{a}$ 에 투표한 것으로 v_3 에 전달했으나(이로써 v_3 는 $\bar{a}$ 를 수락하게 됨) 동시에 불법적으로 a 에 투표했다고 v_2 에 알린 상황입니다.

그림 10a의 상황과 같이 프로토콜 수준에서 활성의 인식을 지원하기 위해 v_1 은 v_1 이 a 에 작용하기 전 다른 모든 온전한 노드가 a 를 수락하도록 하는 방법을 확보해야 합니다. 이러한 상황에서는 시스템이 a 에 **동의**한다고 말하는 것이 타당합니다.

정의 (동의, Agree). FBAS $\langle V, Q \rangle$ 는 추후 발생하는 사건과 관계없이, 충분한 수의 메시지가 전달 및 처리된 뒤 모든 온전한 노드가 a 를 수락하는 경우에만 스테이트먼트 a 에 **동의**합니다.

5.4.3. 중앙화된 투표와의 비교

상기 문제가 연합 투표에서 왜 발생하는지를 이해하기 위해 쿼럼 크기 T 를 지니는 N 개 노드의 중앙화된 비잔틴 합의 시스템을 생각해 봅시다. 그러한 시스템의 경우 $f_L = N - T$ 이하의 노드 실패 시 **쿼럼 가용성**이 적용됩니다. 모든 2개의 쿼럼은 반드시 최소한 $2T - N$ 개의 노드를 공유하므로, 올바른 노드의 **쿼럼 교차**는 $f_S = 2T - N - 1$ 번의 비잔틴 오류까지 견딜 수 있습니다.

중앙화된 비잔틴 합의 시스템은 일반적으로 $N = 3f + 1$ 및 $T = 2f + 1$ 를 설정하여 $f_L = f_S = f$ 를 얻으며, 이는 안전과 활성이 동일한 오류 허용 수준을 지니는 평형점입니다. 만약 안전이 활성보다 더욱 중요한 경우 일부 프로토콜에서는 T 를 증가시켜 $f_S > f_L$ 가 되도록 합니다[Li and Mazieres 2007]. FBA의 경우 쿼럼이 유기적으로 발생하기 때문에 시스템들은 평형 상태에 도달할 가능성이 낮으며, 따라서 활성 부재 시 안전을 보호하는 것이 훨씬 중요합니다.

이제 어느 중앙화된 시스템에서 노드 실패와 모순된 투표로 인해 일부 노드 v 가 다른 노드에 의해 비준된 스테이트먼트 a 를 비준할 수 없는 상황을 가정합시다. 만약 v 가 $f_S + 1$ 개의 노드에서 a 가 비준되었음을 주장하는 내용을 수신한 경우, v 는 그들 중 최소한 1개는 올바른 노드이거나 모든 안전 보장이 무너졌음을 알게 됩니다. 어느 쪽이든, v 는 안전의 손실 없이 a 를 처리할 수 있습니다. 등가의 FBA는 집합 B 에서도 전달받을 수 있으며, 삭제될 경우 B 는 올바른 노드의 **쿼럼 교차**를 약화시킵니다. 그러한 B 를 식별하는 것은 다음과

같은 3가지 이유에서 어려운데, 첫째로 쿼럼은 동적으로 발견되며, 둘째로 불량 노드가 슬라이스에 대해 거짓을 말할 수 있고, 셋째로 v 는 어떤 노드가 올바른 노드인지 알지 못합니다. 대신 v -차단 집합이 수락하는 경우 a 를 수락하도록 연합 투표를 정의하였습니다. v -차단 속성은 쉽게 확인할 수 있다는 장점이 있으나, 우리가 실제로 원하는 것은 $f_S + 1$ 개임에도 불구하고 중앙화된 시스템에서는 $f_L + 1$ 개의 노드로부터 전달받는 것과 동일한 구조를 지닙니다.

중앙화된 시스템에서 모든 올바른 노드들 간의 합의를 보장하려면, 어떤 스테이트먼트가 비준됨을 인지하는 $f_L + f_S + 1$ 개의 노드만 있으면 충분합니다. 만약 그 중 f_L 개 이상이 실패하는 경우 어차피 활성을 기대할 수 없습니다. 만약 f_L 개 이하가 실패하는 경우, $f_S + 1$ 개의 노드가 여전히 비준을 증언할 수 있음을 알고 있고, 이는 다른 모든 올바른 노드를 설득시킬 것입니다. f_S 개 노드에 대한 의존도는 FBA 모델에서는 간단한 아날로그가 아닙니다. 그러나 흥미롭게도 쿼럼 크기인 $f_L + f_S + 1 = T$ 는 유사한 접근법이 더욱 복잡한 비준화에서도 효과가 있음을 시사합니다.

달리 말하면 어떤 시점에서 노드는 안전에 대한 진위 여부를 맡길 정도로 강하게 어떤 스테이트먼트에 대해 신뢰해야 합니다. 중앙화된 시스템은 어떤 스테이트먼트 a 에 대해 해당 수준에 도달할 수 있는 2가지 방법을 다음과 같이 제시합니다. 직접 a 를 비준하거나, a 가 비준되었음을 주장하는 $f_S + 1$ 개의 노드로부터 역추론하여 이들이 모두 거짓을 말한 경우 안전을 기대할 수 없음을 파악합니다. FBA는 후자의 접근법이 없으므로 올바른 노드에 안전을 확보하기 위한 유일한 방법은 **직접 비준**입니다. 노드는 여전히 비준된 스테이트먼트에 반대하는 투표를 극복할 방법이 필요하기에, **수락**이라는 개념을 소개했으나 이는 온전한 노드만을 대상으로 하는 보다 약한 수준의 일관성만을 제공합니다.

5.5. 스테이트먼트 확정

수락된 스테이트먼트의 두 제한사항은 모두 온전한 노드 S 가 스테이트먼트 a 에 반대하여 투표했음에도 불구하고 a 가 비준되는 경우 발생하는 문제로부터 파생되어 일어납니다. 특히 FBA의 비균일적인 쿼럼을 토대로 바라봤을 때, S 는 일부 온전한 노드가 v 를 비준하는 것을 막을 수도 있습니다. v 에게 반대 투표에도 불구하고 a 를 수락할 수 있는 수단을 제공하기 위해, **수락**의 정의에는 v -차단 집합을 토대로 두 번째 기준을 갖고 있습니다. 그러나 이 두 번째 기준은 비준보다는 약하여, 쿼럼 교차가 적용되는 오염 노드에 대한 보장을 제공하지 않습니다.

이제 스테이트먼트 a 에 어떠한 온전한 노드도 반대투표를 하지 않는다는 속성이 있다고 가정해 봅시다. 이 경우 a 를 수락할 필요는 없으며 대신 노드가 적용되기에 앞서 a 이를 직접 비준할 수 있다고 주장할 수 있습니다. 이러한 스테이트먼트를 **반박 불가**하다고 일컫습니다.

정의 (반박 불가, Irrefutable). 어떤 스테이트먼트 a 는 온전한 노드가 이에 반대하여 투표할 수 없는 경우 FBAS 에서 **반박 불가**합니다.

정리 8은 2개의 온전한 노드가 서로 모순되는 스테이트먼트를 수락할 수 없음을 알려줍니다. 따라서 어떤 온전한 노드에 의해 수락된 스테이트먼트 a 에 반대하여 어떤 온전한 노드가 투표할 수는 있으나, ‘어떤 온전한 노드가 a 를 수락함’이라는 스테이트먼트는 **반박 불가**합니다. 이는 온전한 노드가 a 를 수락하였다는 사실을 비준하기 위해 **두 번째 투표**를 진행할 것을 제안합니다.

정의 (확인, Confirm). FBAS 에 있는 한 쿼럼 U_a 는 $\forall v \in U_a$ 인 v 가 a 를 수락하는 것으로 주장하는 경우에만 스테이트먼트 a 를 **확인**합니다. 노드는 그러한 쿼럼에 포함된 경우에만 a 를 확인합니다.

노드는 ‘어떤 온전한 노드가 a 를 수락’이라는 스테이트먼트의 약어인 $\text{accept}(a)$ 를 사용하여 스테이트먼트 a 를 수락하였음을 표현합니다. a 를 **확인**한다는 것은 $\text{accept}(a)$ 을 **비준**하였음을 의미합니다. 올바른 노드 v 는 어느 특정한 다른 노드가 온전한지를 추정할 수 없기 때문에, v 는 a 를 수락한 다음에만 $\text{accept}(a)$ 에 투표할 수 있습니다. v 자신이 오염된 경우, $\text{accept}(a)$ 은 거짓일 수 있으며 그러한 경우 이를 위해 투표하는 것은 v 의 활성이 필요할 수 있으나, 오염 노드는 어차피 활성을 보장할 수 없습니다.

다음 정리에서는 노드가 최적의 안전을 잃지 않고 **확인된 스테이트먼트**를 신뢰할 수 있음을 보여줍니다. 정리 11은 확인된 스테이트먼트들이 제5.4.2장에서 정의한 **합의(Agreement)** 에 부합함을 보여주는데, 이는 노드가 온전한 노드의 활성에 위해를 끼치지 않으면서도 확인된 스테이트먼트를 신뢰할 수 있다는 의미입니다.

5.3. 수락 스테이트먼트

온전한 노드 v 가 어떠한 스테이트먼트를 비준하였음을 알게 되는 경우, 정리 6에 따라 다른 온전한 노드들이 모순되는 스테이트먼트를 비준하지 않을 것이라는 점을 v 에게 알리게 됩니다. 이는 v 로 하여금 a 를 수락할 충분조건에 해당하나, 필요조건으로는 지정할 수 없습니다. 어느 스테이트먼트를 비준하는 경우 이에 대한 투표가 필요하며, 일부 노드는 이미 모순된 스테이트먼트에 대해 투표했을 수 있습니다. 예를 들어 그림 9에서 v_4 는 다른 3개 노드들이 모순되는 스테이트먼트 a 를 비준했음을 인지하기 전 $\bar{a}$ 에 투표합니다. v_4 는 이제 a 에 투표할 수 없으나, 다른 노드들과 일관성을 유지하도록 a 를 수락하는 쪽이 선호됩니다.

중요한 것은 만약 노드 v 가 온전한 경우, 그 어떤 v -차단 집합인 B 도 오염 노드로만 구성될 수 없다는 점입니다. 이제 B 가 v -차단 집합이며 B 의 모든 멤버가 스테이트먼트 a 의 수락을 주장한다고 가정합니다. 만약 v 가 온전한 경우, B 의 멤버 중 최소 1개는 반드시 온전해야 합니다. 이 온전한 멤버는 a 의 수락에 대해 거짓을 말하지 않습니다. 따라서 a 는 참이며 v 가 수락할 수 있습니다. 물론 v 가 오염되었다면 a 는 참이 아닐 수 있습니다. 그러나 오염된 노드는 무엇이든 수락할 수 있으며 온전한 노드의 정확성에 전혀 영향을 끼치지 않을 수 있습니다.

정의 (수락, Accept). FBAS 노드 v 는 스테이트먼트 a 를 다음과 같은 경우에만 수락하는데, a 에 모순되는 스테이트먼트를 수락한 적이 없으며

- (1) $v \in U$ 인 쿼럼 U 가 존재하고 U 의 각 멤버가 a 에 투표했거나 a 를 수락하는 것으로 주장하거나,
- (2) 어느 v -차단 집합의 각 멤버가 a 를 수락하는 것으로 주장하는 경우입니다.

올바른 노드는 모순된 스테이트먼트에 투표할 수 없으나, 상기 2번 조건에 따라 노드는 어느 스테이트먼트에 **투표**한 뒤 나중에 모순된 스테이트먼트를 **수락**하는 것이 가능합니다.

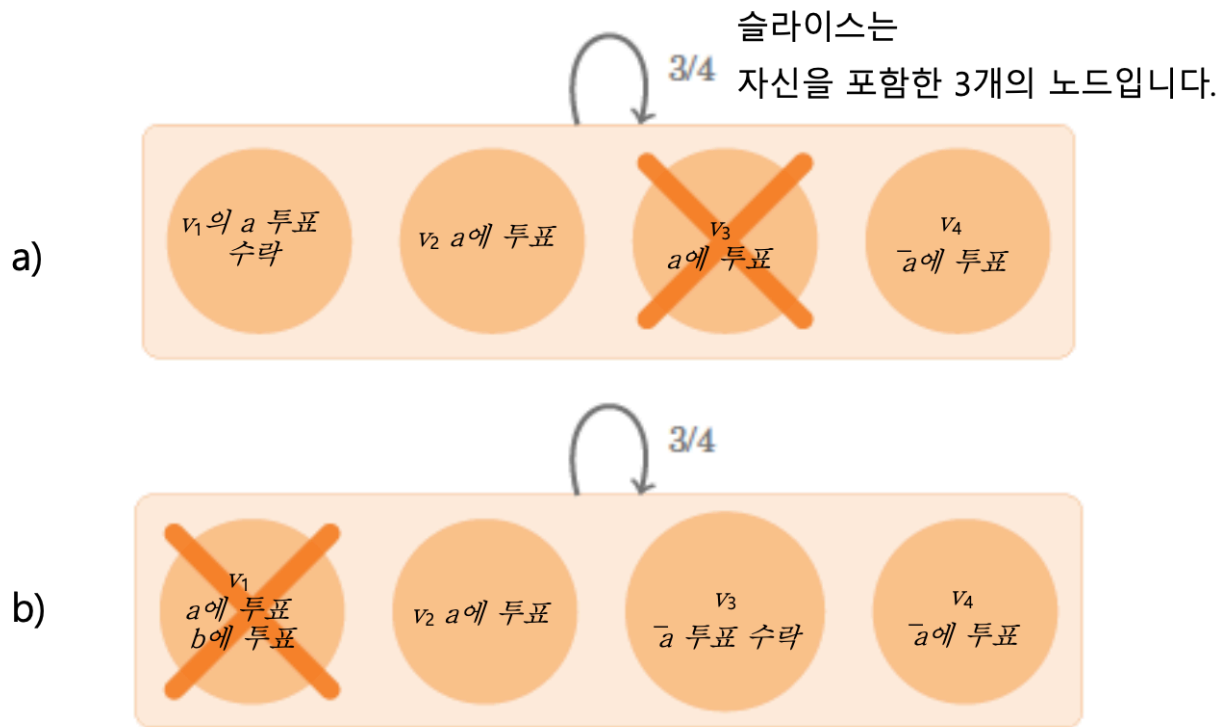


그림 10. v_2 가 강조된 메시지를 확인하지 못하는 경우 v_2 가 구분할 수 없는 시나리오들

정리 8. 쿼럼 교차가 적용된 FBAS에 있는 어느 2개의 온전한 노드는 모순된 스테이트먼트를 수락할 수 없습니다.

증명. $\langle V, Q \rangle$ 를 쿼럼 교차가 적용된 FBAS라고 하며 B 를 오염 노드의 DSet이라고 합시다(정리 3에 의해 존재). 어느 온전한 노드가 스테이트먼트 a 를 수락한다고 가정합시다. a 를 수락하는 첫 번째 온전한 노드를 v 라고 합시다. v 가 a 를 수락하는 시점에는, B 에 있는 오염 노드들만이 이를 수락하겠다고 주장할 수 있습니다. 정리 7에 따른 필연적인 결과로, B 는 v -차단이 될 수 없으므로, 조건 1에 따라 v 는 a 를 수락해야 합니다. 따라서 v 는 모든 노드가 a 에 투표하거나 이를 수락한 쿼럼 U 를 식별하게 되며, v 는 a 를 수락하는 첫 번째 온전한 노드이므로 이는 반드시 $U \setminus B$ 의 모든 노드가 a 에 투표했음을 의미합니다. 달리 말하면 v 는 $\langle V, Q \rangle^B$ 에서 a 를 비준했습니다. 이를 일반화하면, $\langle V, Q \rangle$ 의 어떠한 온전한 노드에 의해 수락되는 모든 스테이트먼트는 반드시 $\langle V, Q \rangle^B$ 에서 비준됩니다. B 는 DSet이므로, $\langle V, Q \rangle^B$ 는 쿼럼 교차가 적용됩니다. 더불어 B 는 모든 불량 노드를 포함하므로, 정리 4에 따라 모순된 스테이트먼트들의 비준은 이루어지지 않습니다.

5.4. 수락으로는 불충분

안타깝게도 수락된 스테이트먼트들이 참이라고 가정하는 노드들은 연합 합의 프로토콜에서 보장하는 안전과 활성의 최적 수준에 미치지 못합니다. 이에 따라 안전과 활성에 관한 문제를 다루어야 합니다. 이해를 돕기 위해, 이러한 문제들이 왜 중앙화된 비잔틴 동의보다 FBA에서 더욱 까다로운지 설명하겠습니다.

5.4.1. 안전

유일한 쿼럼이 만장일치로 동의한, 즉 $\forall v, Q(v) = \{V\}$ 인 FBAS $\langle V, Q \rangle$ 를 가정합니다. 이는 안전에 있어서 보수적인 선택이어야 합니다. 즉 모두가 동의하지 않는 한 아무 일도 하지 않습니다. 그러나 모든 노드가 모든 v 에 대하여 v -차단에 해당하므로, 모든 노드는 단독으로 다른 노드를 설득시켜 임의로 스테이트먼트를 수락하도록 만들 수 있습니다.

문제는 수락된 스테이트먼트가 온전한 노드 사이에서만 안전하다는 것입니다. 하지만 제4.1장에서 논의된 바와 같이, 안전을 보장하기에 필요한 유일한 조건은 올바른 노드의 **쿼럼 교차**이며, 일부 올바른 노드가 오염된 경우에도 적용될 수 있습니다. 특히 $Q(v) = \{V\}$ 인 경우, DSet은 $\emptyset$ 과 V 만이 존재하며, 그 어떤 유형의 노드 오류라도 전체 시스템을 오염시킬 수 있음을 의미합니다. 이와 대조적으로 모든 $B \subseteq V$ 에도 불구하고 쿼럼 교차는 적용됩니다.

5.4.2. 활성

수락된 스테이트먼트의 또 다른 제한사항은 다른 온전한 노드가 이를 수락할 수 없을 수도 있다는 점입니다. 이 가능성은 수락된 스테이트먼트에 대한 의존성이 활성화에 문제가 되도록 합니다. 만약 어느 노드가 스테이트먼트를 수락함으로써 스테이트먼트에 작용하는 경우, 다른 노드들은 이와 유사한 방식으로 진행하지 못할 수도 있습니다.

그림 10a에서, v_1 의 스테이트먼트 a 비준 및 수락을 도운 뒤 v_3 가 충돌하는 상황을 보도록 합시다. v_1 은 a 를 수락하나, v_2 와 v_4 는 수락할 수 없습니다. 구체적으로 v_2 의 관점에서 이 상황은 그림 10b와 구분할 수 없는데, 해당 그림의 경우 v_3 가 $\bar{a}$ 에 투표하여 올바른 노드이나 응답이 느리고, v_1 은 불량 노드이며 $\bar{a}$ 에 투표한 것으로 v_3 에 전달했으나(이로써 v_3 는 $\bar{a}$ 를 수락하게 됨) 동시에 불법적으로 a 에 투표했다고 v_2 에 알린 상황입니다.

그림 10a의 상황과 같이 프로토콜 수준에서 활성의 인식을 지원하기 위해 v_1 은 v_1 이 a 에 작용하기 전 다른 모든 온전한 노드가 a 를 수락하도록 하는 방법을 확보해야 합니다. 이러한 상황에서는 시스템이 a 에 **동의**한다고 말하는 것이 타당합니다.

정의 (동의, Agree). FBAS $\langle V, Q \rangle$ 는 추후 발생하는 사건과 관계없이, 충분한 수의 메시지가 전달 및 처리된 뒤 모든 온전한 노드가 a 를 수락하는 경우에만 스테이트먼트 a 에 동의합니다.

5.4.3. 중앙화된 투표와의 비교

상기 문제가 연합 투표에서 왜 발생하는지를 이해하기 위해 쿼럼 크기 T 를 지니는 N 개의 중앙화된 비잔틴 합의 시스템을 생각해 봅시다. 그러한 시스템의 경우 $f_L = N - T$ 이하의 노드 실패 시 쿼럼 가용성이 적용됩니다. 모든 2개의 쿼럼은 반드시 최소한 $2T - N$ 개의 노드를 공유하므로, 올바른 노드의 쿼럼 교차는 $f_S = 2T - N - 1$ 번의 비잔틴 오류까지 견딜 수 있습니다.

중앙화된 비잔틴 합의 시스템은 일반적으로 $N = 3f + 1$ 및 $T = 2f + 1$ 를 설정하여 $f_L = f_S = f$ 를 얻으며, 이는 안전과 활성이 동일한 오류 허용 수준을 지니는 평형점입니다. 만약 안전이 활성보다 더욱 중요한 경우 일부 프로토콜에서는 T 를 증가시켜 $f_S > f_L$ 가 되도록 합니다[Li and Mazieres 2007]. FBA의 경우 쿼럼이 유기적으로 발생하기 때문에 시스템들은 평형 상태에 도달할 가능성이 낮으며, 따라서 활성 부재 시 안전을 보호하는 것이 훨씬 중요합니다.

이제 어느 중앙화된 시스템에서 노드 실패와 모순된 투표로 인해 일부 노드 v 가 다른 노드에 의해 비준된 스테이트먼트 a 를 비준할 수 없는 상황을 가정합니다. 만약 v 가 $f_S + 1$ 개의 노드에서 a 가 비준되었음을 주장하는 내용을 수신한 경우, v 는 그들 중 최소한 1개는 올바른 노드이거나 모든 안전 보장이 무너졌음을 알게 됩니다. 어느 쪽이든, v 는 안전의 손실 없이 a 를 처리할 수 있습니다. 등가의 FBA는 집합 B 에서도 전달받을 수 있으며, 삭제될 경우 B 는 올바른 노드의 쿼럼 교차를 약화시킵니다. 그러한 B 를 식별하는 것은 다음과 같은 3가지 이유에서 어려운데, 첫째로 쿼럼은 동적으로 발견되며, 둘째로 불량 노드가 슬라이스에 대해 거짓을 말할 수 있고, 셋째로 v 는 어떤 노드가 올바른 노드인지 알지 못합니다. 대신 v -차단 집합이 수락하는 경우 a 를 수락하도록 연합 투표를 정의하였습니다. v -차단 속성은 쉽게 확인할 수 있다는 장점이 있으나, 우리가 실제로 원하는 것은 $f_S + 1$ 개임에도 불구하고 중앙화된 시스템에서는 $f_L + 1$ 개의 노드로부터 전달받는 것과 동일한 구조를 지닙니다.

중앙화된 시스템에서 모든 올바른 노드들 간의 합의를 보장하려면, 어떤 스테이트먼트가 비준됨을 인지하는 $f_L + f_S + 1$ 개의 노드만 있으면 충분합니다. 만약 그 중 f_L 개 이상이 실패하는 경우 어차피 활성을 기대할 수 없습니다. 만약 f_L 개 이하가 실패하는 경우, $f_S + 1$ 개의 노드가 여전히 비준을 증언할 수 있음을 알고 있고, 이는 다른 모든 올바른 노드를 설득시킬 것입니다. f_S 개의 노드에 대한 의존도는 FBA 모델에서는 간단한 아날로그가 아닙니다. 그러나 흥미롭게도 쿼럼 크기인 $f_L + f_S + 1 = T$ 는 유사한 접근법이 더욱 복잡한 비준화에서도 효과가 있음을 시사합니다.

달리 말하면 어떤 시점에서 노드는 안전에 대한 진위 여부를 말길 정도로 강하게 어떤 스테이트먼트에 대해 신뢰해야 합니다. 중앙화된 시스템은 어떤 스테이트먼트 a 에 대해 해당 수준에 도달할 수 있는 2가지 방법을 다음과 같이 제시합니다. 직접 a 를 비준하거나, a 가 비준되었음을 주장하는 $f_S + 1$ 개의 노드로부터 역추론하여 이들이 모두 거짓을 말한 경우 안전을 기대할 수 없음을 파악합니다. FBA는 후자의 접근법이 없으므로 올바른 노드에 안전을 확보하기 위한 유일한 방법은 직접 비준입니다. 노드는 여전히 비준된 스테이트먼트에 반대하는 투표를 극복할 방법이 필요하기에, 수락이라는 개념을 소개했으나 이는 온전한 노드만을 대상으로 하는 보다 약한 수준의 일관성만을 제공합니다.

5.5. 스테이트먼트 확정

수락된 스테이트먼트의 두 제한사항은 모두 온전한 노드 S 가 스테이트먼트 a 에 반대하여 투표했음에도 불구하고 a 가 비준되는 경우 발생하는 문제로부터 파생되어 일어납니다. 특히 FBA의 비균일적인 쿼럼을 토대로 바라봤을 때, S 는 일부 온전한 노드가 v 를 비준하는 것을 막을 수도 있습니다. v 에게 반대 투표에도 불구하고 a 를 수락할 수 있는 수단을 제공하기 위해, **수락**의 정의에는 v -차단 집합을 토대로 두 번째 기준을 갖고 있습니다. 그러나 이 두 번째 기준은 비준보다는 약하여, 쿼럼 교차가 적용되는 오염 노드에 대한 보장을 제공하지 않습니다.

이제 스테이트먼트 a 에 어떠한 온전한 노드도 반대투표를 하지 않는다는 속성이 있다고 가정해 봅시다. 이 경우 a 를 수락할 필요는 없으며 대신 노드가 적용되기에 앞서 a 이 직접 비준할 수 있다고 주장할 수 있습니다. 이러한 스테이트먼트를 **반박 불가**하다고 일컫습니다.

정의 (반박 불가, Irrefutable). 어떤 스테이트먼트 a 는 온전한 노드가 이에 반대하여 투표할 수 없는 경우 FBAS에서 **반박 불가**합니다.

정리 8은 2개의 온전한 노드가 서로 모순되는 스테이트먼트를 수락할 수 없음을 알려줍니다. 따라서 어떤 온전한 노드에 의해 수락된 스테이트먼트 a 에 반대하여 어떤 온전한 노드가 투표할 수는 있으나, ‘어떤 온전한 노드가 a 를 수락함’이라는 스테이트먼트는 **반박 불가**합니다. 이는 온전한 노드가 a 를 수락하였다는 사실을 비준하기 위해 **두 번째 투표**를 진행할 것을 제안합니다.

정의 (확정, Confirm). FBAS 에 있는 한 쿼럼 U_a 는 $\forall v \in U_a$ 인 v 가 a 를 수락하는 것으로 주장하는 경우에만 스테이트먼트 a 를 **확정**합니다. 노드는 그러한 쿼럼에 포함된 경우에만 a 를 확정합니다.

노드는 ‘어떤 온전한 노드가 a 를 수락’이라는 스테이트먼트의 약어인 $\text{accept}(a)$ 를 사용하여 스테이트먼트 a 를 수락하였음을 표현합니다. a 를 **확정**한다는 것은 $\text{accept}(a)$ 을 **비준**하였음을 의미합니다. 올바른 노드 v 는 어느 특정한 다른 노드가 온전한지를 추정할 수 없기 때문에, v 는 a 를 수락한 다음에만 $\text{accept}(a)$ 에 투표할 수 있습니다. v 자신이 오염된 경우, $\text{accept}(a)$ 은 거짓일 수 있으며 그러한 경우 이를 위해 투표하는 것은 v 의 활성이 필요할 수 있으나, 오염 노드는 어차피 활성을 보장할 수 없습니다.

다음 정리에서는 노드가 최적의 안전을 잃지 않고 확정된 스테이트먼트를 신뢰할 수 있음을 보여줍니다. 정리 11은 확정된 스테이트먼트들이 제5.4.2장에서 정의한 **합의(Agreement)** 에 부합함을 보여주는데, 이는 노드가 온전한 노드의 활성에 위해를 끼치지 않으면서도 확정된 스테이트먼트를 신뢰할 수 있다는 의미입니다.

정리/9. $\langle V, Q \rangle$ 를 B 에도 불구하고 쿼럼 교차가 적용된 FBAS 로 하고, B 는 모든 불량 노드를 포함하는 것으로 가정합니다. v_1 과 v_2 를 B 에 없는 노드 2 개로 하고, a 와 $\bar{a}$ 는 모순되는 스테이트먼트로 합니다. v_1 이 a 를 확정하면, v_2 는 $\bar{a}$ 를 확정할 수 없습니다.

증명. 우선 $\text{accept}(a)$ 는 $\text{accept}(\bar{a})$ 와 모순되며, 어떠한 올바른 노드도 둘 다에 투표할 수 없다고 가정합니다. a 를 확정하기 위해서는 v_1 이 반드시 $\text{accept}(a)$ 를 비준해야 함에 유의하십시오. 정리 5에 따라, v_1 는 $\text{accept}(a)$ 를 비준할 수 없으며 따라서 a 를 확정할 수 없습니다.

정리/10. B 를 쿼럼 교차가 적용된 FBAS $\langle V, Q \rangle$ 의 오염 노드로 구성된 집합으로 합시다. U 를 ($U \subseteq B$) 에 해당하는 올바른 노드를 포함하는 쿼럼으로 하고, S 를 $U \subseteq S \subseteq V$ 로 하는 어떤 집합으로 합시다. $S^+ = S \setminus B$ 를 S 의 올바른 노드로 구성된 집합으로 하고, $S_- = (V \setminus S) \setminus B$ 를 S 에 포함되지 않은 올바른 노드의 집합으로 합시다. S^+ 가 v -차단이 되도록 $S_- = \emptyset$ 이거나, 또는 $\exists v \in S_-$ 이 성립합니다.

증명. 만약 S^+ 가 어떤 $v \in S_-$ 에 대해 v -차단이라면, 증명 완료입니다. 그렇지 않다면 반드시 $S_- = \emptyset$ 임을 보여야 합니다. 만약 S^+ 가 어떠한 $v \in S_-$ 에 대해 v -차단이 되지 않는다면, 정리 7에 따라 $S_- = \emptyset$ 이거나 S_- 는 $\langle V, Q \rangle$ 의 쿼럼입니다. 전자의 경우 증명 완료이며, 후자의 경우 다음과 같은 모순이 발생합니다. 정리 1에 따라, $U \setminus B$ 는 $\langle V, Q \rangle$ 의 쿼럼입니다. (정리 3에 따라) B 는 DSet 이므로, $\langle V, Q \rangle_B$ 에는 반드시 쿼럼 교차가 적용되어야 하며, 즉 $S_- \cap (U \setminus B) \neq \emptyset$ 입니다. 이것은 $(U \setminus B) \subseteq S$ 이므로 $S_- \cap S = \emptyset$ 이므로 불가능합니다.

정리/11. 만약 쿼럼 교차가 적용된 FBAS $\langle V, Q \rangle$ 에 포함된 어떤 온전한 노드가 스테이트먼트 a 를 확정하는 경우, 이후 어떤 결함이 발생하더라도 충분한 메시지가 전달 및 처리된 뒤에는 모든 온전한 노드가 a 를 수락 및 확정할 것입니다.

분산된 합의(Distributed Consensus)의 주요 과제는 중앙화의 여부와 관계없이, 시스템이 합의(Agreement)에 도달하기 전 어떤 스테이트먼트가 영구적으로 미결정 상태로 교착될 수 있다는 점입니다. 따라서 프로토콜은 절대로 표면화된 값을 직접 비준하려고 시도해서는 안 됩니다. ‘슬롯 i 의 값은 x ’라는 스테이트먼트가 갇히는 경우, 시스템은 영영 슬롯 i 에 합의할 수 없게 됨으로써 활성을 잃게 됩니다. 해결책은 투표 시 스테이트먼트에 더욱 주의를 기울여 진행하는 것입니다. 갇힌 스테이트먼트는 반드시 정말로 중요한 주제, 즉 슬롯 내용에 관하여 파기할 수 있어야 합니다. 교착된 스테이트먼트를 제거하는 절차를 무효화 라고 일컫습니다.

로컬 상태	a 의 시스템적 상태
미실행	미상(임의)
a 에 투표함	미상(임의)
$\bar{a}$ 에 투표함	미상(임의)
a 를 수락함	갇힘, a -가(價) 또는 a 에 합의
a 를 확정	a 에 합의

그림 13. 온전한 노드가 스테이트먼트 a 의 상태에 대해 알고 있는 내용

더욱 구체적으로 그림 12는 어떤 스테이트먼트 a 가 시스템적으로 지닐 수 있는 상태를 나타냅니다. 최초에 시스템은 양자 결합으로, 이는 모든 온전한 노드가 a 를 수락할 수 있는 가능한 사건들의 연속, 그리고 모든 온전한 노드가 a 를 거부할(즉 a 와 상충하는 어떤 스테이트먼트 $\bar{a}$ 를 수락하는) 사건들의 연속을 의미합니다. 어느 시점에 들어서면 이 두 결과 중 하나는 더 이상 가능하지 않게 됩니다. 만약 온전한 노드 중 어느 것도 a 를 거부할 수 없는 경우, 해당 시스템은 $\bar{a}$ 결합이라고 칭합니다. 반대로 만약 온전한 노드 중 어느 것도 a 를 수락할 수 없는 경우, 해당 시스템은 $\bar{a}$ -가(價)라고 합니다.

FBAS 전환이 양자 결합에서 a -가(價)로 전환되는 경우, 모든 온전한 노드가 a 를 수락하는 결과가 발생할 수 있습니다. 그러나 이러한 경우가 지속되지 않을 수도 있습니다. 어떤 노드 3개가 쿼럼을 구성하며 PBFT와 유사한 4노드 시스템 $\{v_1, \dots, v_4\}$ 을 가정해 봅시다. 만약 v_1 과 v_2 가 a 에 투표하는 경우, 시스템은 a -가(價)가 됩니다. 어떤 노드 3개도 모순되는 스테이트먼트를 비준할 수 없습니다. 그러나 만약 이후에 v_3 와 v_4 가 a 에 상충되는 $\bar{a}$ 에 투표한 경우, a 를 비준하는 것도 불가능해집니다. 이 경우, a 의 상태는 영구히 미결정 상태, 내지는 교착 상태가 됩니다.

그림 10a에서와 같이 온전한 노드가 a 를 수락한 경우에도 시스템은 a 에 대하여 시스템 단위의 합의에 도달하지 못할 수 있습니다. 그러나 정리 11에 따라, 온전한 노드가 a 를 확정하는 경우 모든 온전한 노드는 결국 수락하게 되며, 따라서 시스템은 a 에 합의하게 됩니다. 그림 13은 온전한 노드가 로컬 상태에서 어떤 스테이트먼트의 전체 상태에 대해 알고 있는 내용을 요약하고 있는 것입니다.

합의의 가능성을 보존하기 위해, 프로토콜은 반드시 모든 스테이트먼트가 반박할 수 없어서 교착되지 않거나, 무력화가 가능하여 갇히더라도 진행을 차단할 수 없도록 해야 합니다. 무력화가 가능한 스테이트먼트를 제작하는 데에는 2가지 잘 알려진 접근법이 존재합니다. 하나는 뷰스탬프(viewstamp)된 복제를 통해 개척된 보기 기반 접근법[Okie and Liskov 1988]으로 PBFT에서 선호하며 [Castro and Liskov 1999], 팍소스(Paxos)에서 개발한 무기명 투표 기반 접근법이 있습니다 [Lamport 1998]. 무기명 투표 기반 접근법은 이해하기에 조금 더 난해할 수 있습니다 [Ongaro and Ousterhout 2014]. 사람들은 주로 뷰스탬프된 복제를 ‘팍소스’라고 칭하거나 두 알고리즘이 실제로는 다름에도 불구하고 동일한 것으로 간주하여, 더욱 혼란을 가중시킵니다. [van Renesse et al. 2014].

보기 기반 프로토콜은 투표의 슬롯을 단조 증가하는 보기 번호와 연결합니다. 합의가 보기 n 의 i 번째 슬롯에 교착된 경우, 노드들은 보기 n 이 i 개 미만의 유의미한 슬롯을 가진 것으로 합의하고 더 높은 번호의 보기로 이동하여 복구합니다. 무기명 투표 기반 프로토콜은 투표의 값을 단조 증가하는 보기 번호와 연결합니다. 무기명 투표가 갇히는 경우, 노드들은 더 높은 수의 무기명 투표로 동일한 슬롯을 다시 시도하며, 기존의 교착된 무기명 투표와 상충하는 값은 절대로 선택하지 않습니다.

이 작업을 진행함으로써 무기명 투표 기반 접근법은 구분된 주요 노드 또는 리더의 개념을 없애는 것이 더욱 용이합니다. 예를 들어 리더 행동은 모방될 수 있습니다 [Lamport 2011b].

6. SCP: 연합 비잔틴 합의 프로토콜 (FEDERATED BYZANTINE AGREEMENT PROTOCOL)

본 장은 스텔라 합의 프로토콜(Stellar Consensus Protocol, SCP)을 소개합니다. 높은 수준에서 SCP는 2개의 하위 프로토콜, 즉 지명 프로토콜과 무기명 투표 프로토콜을 포함합니다.

지명 프로토콜은 슬롯에 대한 후보 값들의 집합을 생성합니다. 충분히 오래 가동된 이후에 해당 프로토콜은 결과적으로 모든 온전한 노드에 대해 동일한 후보 값 집합을 생성하며, 이는 노드들이 슬롯에 대한 단일 합성 값을 생성할 수 있도록 후보 값들을 결정론적인 방식으로 결합할 수 있음을 의미합니다. 그러나 중대한 위험 사항이 2가지 존재합니다. 먼저 노드들은 언제 수렴 지점에 도달하는지 알 수 없습니다. 둘째로 수렴된 이후에도 불량한 노드가 일정 횟수만큼 지명 절차를 초기화할 수 있습니다.

지명 프로토콜이 집합되었음을 노드들이 추정 한 이후에는 무기명 투표 프로토콜을 실행하며, 이는 연합 투표를 도입함으로써 합성 값에 연관된 무기명 투표를 실행 및 중단하게 합니다. 올바른 노드가 무기명 투표의 실행에 합의한 뒤, 무기명 투표와 관련된 값은 문제의 슬롯에 대해 표면화됩니다. 무기명 투표의 중단에 합의한 경우, 해당 무기명 투표의 값은 제외됩니다. 만약 무기명 투표에서 하나 이상의 올바른 노드가 실행 또는 중단할 수 없는 상태로 교착된 경우, 노드들은 더 높은 수의 무기명 투표를 다시 진행합니다. 이 때 교착된 상태의 무기명 투표가 실행되었을 것으로 어떤 노드가 신뢰하는 경우를 대비하여 새로운 무기명 투표를 교착상태의 투표와 동일한 값에 연결합니다. 직관적으로, 모든 교착화 되거나 실행된 무기명 투표가 동일한 값에 연관될

수 있도록 함으로써 안전이 확보됩니다. 더 높은 무기명 투표로 이동함으로써 갇힌 무기명 투표를 무효화할 수 있다는 사실로부터 활성이 발생합니다.

본 장의 나머지 내용에서는 지명 및 무기명 투표 프로토콜을 소개합니다. 각각 개념적 서술을 통해 설명되며, 이후 개념적 서술의 집합들을 나타내는 메시지를 갖춘 구체적 프로토콜이 제시됩니다. 마지막으로 제6.3장은 프로토콜의 정확성을 나타냅니다. SCP는 각각의 슬롯을 완전히 독립적으로 다루며 단일 슬롯 합의 프로토콜의 여러 독립된 인스턴스로 간주될 수 있습니다(이는 팍소스의 ‘단일 결정 총회’와 유사합니다[Lamport 1998]). 후보값과 무기명 투표를 비롯한 개념들은 항상 토의 대부분에서 슬롯을 언급하지 않은 상태로 두더라도 특정 슬롯의 맥락을 토대로 해석이 이루어져야 합니다.

6.1. 지명 프로토콜

슬롯들은 부분적으로만 명령을 내리면 되므로, SCP의 일부 응용 프로그램에는 슬롯 당 타당한 무기명 투표란이 1개밖에 없습니다. 예를 들어 인증서 투명성에 있어, 각 CA는 개별적으로 일련의 슬롯을 지니는 가운데 슬롯마다 정확히 1개의 인증서 트리에 서명할 수 있습니다. 그러나 다른 애플리케이션의 경우 슬롯 당 다수의 타당한 값을 인정할 수 있으며, 이런 경우 가능한 입력값을 축약하는 것이 도움이 될 것입니다. 우리의 전략은 특정 타이밍 추정을 토대로 합의에 도달하는 동기식 지명 프로토콜로 시작한 뒤, 해당 지명 프로토콜의 결과값을 안전 확보에 있어 타이밍이 중요하지 않은 비동기식 무기명 투표 프로토콜로 전달하는 것입니다 [Lamport 2011a]. 그러한 최초의 동기식 단계는 종종 조정자 라고 칭합니다 [Aspnes 2010].

지명 프로토콜은 어떤 한 슬롯에 대하여 일련의 후보 값으로 수렴하는 방식으로 작동합니다. 이후 노드들은 결정론적으로 이 후보들을 해당 슬롯에 대한 단일 합성 값으로 결합합니다. 정확히 값들을 어떤 방식으로 합성할지는 애플리케이션에 따라 다릅니다. 예를 들어, 스텔라 네트워크는 SCP를 사용하여 트랜잭션 집합과 각 슬롯에 대한 원장 타임스탬프를 선택합니다. 후보 값들을 조합하기 위해, 스텔라는 이들의 트랜잭션 집합과 최대 타임스탬프를 취합합니다. (올바르지 않은 타임스탬프를 지닌 값들은 후보가 되기에 충분한 지명을 받지 않을 것입니다.) 다른 가능한 접근법으로는 교차에 의한 집합의 조합이나 가장 높은 해시를 가지는 후보 값을 선택하는 방법 등이 있습니다.

노드는 `nominate x` 스테이트먼트에 대한 연합 투표를 통해 후보 값 x 를 생성합니다.

정의 (후보, Candidate). 노드 v 는 v 가 x 를 지명한다는 스테이트먼트를 확정, 즉 v 가 `accept(nominate x)` 를 비준한 경우 값 x 를 후보 로 간주합니다.

노드 v 가 후보 값을 가지지 않는 한, v 는 애플리케이션 수준의 유효성 점검을 통과한 모든 값 x 에 대해(향후 존재하지 않는 타임스탬프 등) x 를 지명한다는 투표를 실시할 수 있습니다. 사실 후보가 폭발적으로 증가하는 현상을 방지하기 위해 이하에서 논의되는 속도 제한 요소들을 토대로 일반적으로 다른 노드들이 지명하는 값을 재지명해야 합니다. 그러나 v 가 후보 값을 가지는 순간 모든 새로운 값에 대하여 `nominate x`에 대한 투표를 중단해야 합니다. 그러나 (v -차단 집합에 의해 수락된 경우) 계속해서 새로운 값에 대한 지명 스테이트먼트를 수락해야 하며 연합 투표 절차에서 규정된 새로운 지명 스테이트먼트를 확정해야 합니다.

지명 프로토콜에는 시스템에 온전한 노드가 있는 경우(즉 완전 실패가 일어나지 않은 경우) 몇 가지 속성이 적용됩니다. 구체적으로, 각 슬롯에서

- (1) 온전한 노드들은 최소 1개의 후보 값을 생성할 수 있습니다.
- (2) 어떤 시점에 가능한 후보 값의 집합은 성장을 멈춥니다.
- (3) 어떤 온전한 노드가 x 를 후보 값으로 고려하는 경우, 결과적으로 모든 온전한 노드는 x 를 후보 값으로 고려하게 됩니다.

이제 지명 프로토콜이 3가지 속성을 어떻게 달성하는지를 고려해 봅시다. 속성 1은 지명 스테이트먼트가 거부할 수 없으므로 달성됩니다. 노드는 절대로 특정 값을 지명하는 것에 반대하는 투표를 하지 않으며, 첫 번째 후보 값이 확정되기 전까지 온전한 노드들은 어떠한 값이든지 지명하기 위해 투표할 수 있습니다. 따라서 값을 불문하고 x 값이 애플리케이션 수준의 유효성 검증을 통과하는 경우, 온전한 노드는 nominate x 를 확정하기 위해 투표할 수 있습니다. 속성 2는 각 온전한 노드가 최소 1개의 후보 값을 확정하는 경우(이는 유한한 시간 내에 반드시 일어납니다), 어떠한 온전한 노드도 새로운 값을 지명하지 않을 것이므로 보장됩니다. 따라서 후보가 될 수 있는 값들은 온전한 노드에게 투표를 받은 값들만 해당됩니다. 속성 3은 정리 11의 직접적인 결과입니다.

지명 절차는 적용된 값들의 조합 수가 적을 경우 더욱 효율적일 것입니다. 따라서 임시로 우선순위를 노드에 각각 할당하고, 각 노드로 하여금 보다 높은 우선순위를 지니는 노드와 동일한 값을 지명하도록 합니다. 더욱 구체적으로는, H 를 정수의 집합 $\{0, \dots, h_{\max}-1\}$ 을 치역으로 해석할 수 있는 암호화 해시 함수라고 합시다. (H 는 SHA-256일 수 있으며 [National Institute of Standards and Technology 2012], 이 경우 $h_{\max} = 2^{256}$ 입니다.) $G_i(m) = H(i, x_{i-1}, m)$ 을 슬롯 i 에 대하여 슬롯 특이적 해시 함수로 하며, 이때 x_{i-1} 은 i 이전의 슬롯에 대해 선택된 값입니다(또는 부분적인 순서에 따라 슬롯들이 통제되는 경우 슬롯 i 에 인접한 모든 종속 값들의 분류 집합입니다). 슬롯 i 와 어림수 n 이 주어졌을 때, 각 노드 v 는 각 이웃에 대하여 neighbors 및 priority를 다음과 같이 계산합니다.

$$\text{가중치}(v, v') = \frac{|\{q \mid q \in Q(v) \wedge v' \in q\}|}{|Q(v)|}$$

$$\text{neighbors}(v, n) = \{v' \mid G_i(N, n, v') < h_{\max} \cdot \text{가중치}(v, v')\}$$

$$\text{priority}(n, v') = G_i(P, n, v')$$

N 과 P 는 2개의 다른 해시 함수를 생성하기 위한 상수입니다. 함수 가중치(v, v')는 v' 를 포함하는 $Q(v)$ 의 슬라이스 중 일부를 반환합니다. 가중치를 neighbors(v, n)에 v' 가 나타나도록 n 에 대한 확률로 사용함으로써, 신뢰가 그리 높지 않은 노드가 라운드에서 우세를 점할 확률도 감소시킵니다.

각 노드 v 는 통신할 수 있는 노드 중에서 최초로 priority($0, v_0$)를 극대화하는 $v_0 \in \text{neighbors}(v, 0)$ 에 해당하는 노드를 찾아야 하며, 이후 v_0 와 동일한 값을 지명하기 위해 투표해야 합니다. 만약 $v = v_0$ 인 경우에만 v 는 지명을 위해 새로운 값을 도입해야 하며, v 는 투표 대상인 새로운 지명 스테이트먼트를 결정하기 위해 타임아웃을 사용해야 합니다. n 번의 타임아웃 이후, v 는 priority(n, v_n)을 극대화하는 $v_n \in \text{neighbors}(v, n)$ 의 노드를 찾아 v_n 이 지명을 위해 투표한 모든 값을 지명하기 위해 투표해야 합니다.

변수 의미

- X 노드 v 가 지명하기 위해 투표한 값들의 집합
- Y 노드 v 가 지명된 것으로 수락한 값들의 집합
- Z 노드 v 가 후보 값으로 고려하는 값들의 집합
- N 각 노드로부터 수신한 최신 지명 메시지들의 집합

그림 14. 각 슬롯에 대해 노드 v 에서 유지한 지명 상태

NOMINATE v : $X Y D$

이것은 슬롯 s 에 대해 값을 지명하기 위해 노드 v 에서 보낸 메시지입니다. D 는 v 의 쿼럼 슬라이스 $Q(s)$ 이거나 $Q(s)$ 의 충돌 저장 해시입니다. X 와 Y 는 v 의 상태에서 유도되었습니다. 구체적인 메시지는 다음과 같은 개념적 메시지들을 표현합니다.

- $\{ \text{nominate } x \mid x \in X \}$ (X 의 각 값을 지명하기 위한 투표)
- $\{ \text{accept}(\text{nominate } x) \mid x \in Y \}$ (Y 의 지명을 확정하기 위한 투표)

그림 15. 지명 프로토콜의 메시지

정리 12. 결과적으로 모든 온전한 노드는 동일한 합성 값을 가지게 됩니다.

증명. 해당 정리는 지명 프로토콜의 3가지 속성으로부터 도출됩니다. 각 온전한 노드는 유한한 수의 무기명 투표를 지명하기 위해서만 투표할 것입니다. 불량한 노드의 행동이 부재한 경우, 온전한 노드들은 Z 라고 일컫는 후보 값들의 동일한 집합으로 수렴하게 됩니다. 이러한 수렴을 미연에 방지하기 위해, 불량한 노드는 새로운 후보 값을 도입할 수 있으며, 이를 통해 일정 기간 동안 일부 온전한 노드의 후보로 작용할 수 있습니다. 그러나 그러한 값들은 올바른 노드로부터 투표를 받아야 할 것이며, 따라서 유한 집합으로 한정되는 결과가 일어납니다. 결과적으로 불량한 노드는 시스템을 교란하는 것을 멈추거나 도입할 새로운 후보 값이 고갈되며, 이 경우 온전한 노드는 Z 로 수렴합니다.

6.1.1. 구체적 지명 프로토콜.

그림 14는 노드 v 가 각 슬롯에서 반드시 유지해야 하는 지명 프로토콜을 수록하고 있습니다. X 는 v 가 $\text{nominate } x$ 에 투표한 x 값들의 집합이며, Y 는 v 가 $\text{nominate } x$ 를 수락한 값들의 집합이고, Z 는 후보 값들의 집합, 즉 v 를 포함한 쿼럼이 $\text{accept}(\text{nominate } x)$ 라고 표시한 모든 값입니다. 마지막으로 v 는 각 노드로부터 가장 최근에 수신한 구체적 메시지인 N 을 유지합니다. (엄밀히 따지면 X , Y , 그리고 Z 는 모두 N 에서 다시 연산할 수 있으나, 직접 참조할 수 있도록 하는 것이 더 편리합니다.) 이 4개 영역은 모두 공집합으로 초기 설정됩니다. X , Y , Z 는 모두 시간 경과에 따라 증가한다는 점에 주목하십시오. 노드들은 이 집합들에서 값을 절대로 제거하지 않습니다.

그림 15는 지명 프로토콜을 구성하는 구체적 메시지를 제시하고 있습니다. X 와 Y 는 시간 경과에 따라 단조 증가하므로, 네트워크 인도 순서와 관계없이 D 가 지명 도중 변경되지 않는 한 동일한 노드로부터 들어오는 다수의 NOMINATE 메시지 중 어느 것이 가장 최신인지를 결정할 수 있습니다(그렇지 않은 경우 D 에 버전을 매겨야 합니다). 지명에는 단 하나의 원격 절차 호출(Remote Procedure Call, RPC)만이 필요합니다. 인수는 발신자의 최신 NOMINATE 메시지이며 반환 값은 수신자의 메시지입니다. D 또는 지명된 값이 암호화 해시인 경우, 두 번째 RPC는 필요에 따라 캐시되지 않은 해시의 전상 회수를 허용해야 합니다.

노드는 지명 프로토콜이 언제 완료되었는지를 알 수 없으므로, SCP는 반드시 각 노드에 존재하는 서로 다른 합성 값들에 대처해야 합니다. 그렇다면 최적화로서, 노드들은 후보 값을 가지기 전부터 최종 합성 값의 예측을 시도할 수 있습니다. 이를 위해 합성 값은 $Z \neq \emptyset$ 인 경우 합성(Z)으로 받아들일 수 있으며, 또는 $Y \neq \emptyset$ 인 경우 합성(Y), 또는 $X \neq \emptyset$ 인 경우 합성(X)로 받아들일 수 있습니다. 이는 가장 높은 우선순위의 노드가 제일 일 경우 지명과 동시에 무기명 투표를 개시할 수 있어, 첫 번째 무기명 투표 메시지가 첫 번째 NOMINATE 메시지와 맞물려 준비하도록 함을 의미합니다(아래 설명).

6.2. 무기명 투표 프로토콜.

노드가 합성 값을 확보한 뒤에는 무기명 투표 프로토콜에 참여하게 되나, 이와 병행하여 지명은 계속해서 합성 값을 업데이트할 수 있습니다. 무기명 투표 b 는 $b = \langle n, x \rangle$ 형태의 쌍을 이루며, 이 때 $x \neq \perp$ 이 값이며 b 는 해당하는 슬롯에 대해 x 를 표면화하기 위한 투표입니다. $n \geq 1$ 값은 더 높은 무기명 투표 수가 항상 존재하도록 하기 위한 계수값 역할을 합니다. C 형태의 표기법 $b.n$ 과 $b.x$ 를 사용하여 무기명 투표 b 의 계수값과 값 영역들을 나타내며, 따라서 $b = \langle b.n, b.x \rangle$ 입니다. 무기명 투표는 전적으로 순서가 정해져 있으며, $b.n$ 이 $b.x$ 보다 더욱 중요합니다. 편의를 위해 특수한 무효 공백 값의 무기명 투표 $0 = \langle 0, \perp \rangle$ 은 다른 모든 무기명 투표보다 작으며, 특수한 계수값 ∞ 은 다른 모든 계수 값보다 큼니다.

연합 투표를 사용하여 commit b 및 abort b 스테이트먼트에 합의함으로써 무기명 투표 b 에 대한 실행 및 종단을 함축적으로 표현할 수 있습니다. 제시된 무기명 투표에서 실행 및 종단은 모순되므로 올바른 노드는 최대 1개에만 투표할 수 있습니다. 제5장의 표기법에서 commit b 의 반대는 "commit b "가 되겠지만 abort b 를 사용하는 것이 더 직관적인 표기법일 것입니다.

제공된 슬롯에 대하여 최대 1개의 값을 선택할 수 있으므로 모든 실행 및 교착된 무기명 투표는 반드시 동일한 값을 포함해야 합니다. 대략적으로 볼 때 이는 보다 낮은 번호의 종단되지 않은 무기명 투표와 모순이 발생하는 경우 실행 스테이트먼트가 유효하지 않다는 의미입니다.

정의 (양립 가능, Compatible). 양립 가능한 2개의 무기명 투표 b_1 과 b_2 는 $b_1 \sim b_2$ 으로 표기하며, $b_1.x = b_2.x$ 인 경우에만 성립하고 양립 불가능한 경우 $b_1 \not\sim b_2$ 으로 표기하되 $b_1.x \neq b_2.x$ 인 경우에만 해당합니다. 또한 $b_1 \leq b_2$ 이며 $b_1 \sim b_2$ 인 경우에만 $b_1 \leq b_2$ 또는 $b_2 \geq b_1$ 으로 작성합니다(또는 이와 동일하게 $b_2 \geq b_1$). 이와 유사하게 $b_1 \leq b_2$ 또는 $b_2 \geq b_1$ (또는 이와 동일하게 $b_2 \geq b_1$)은 $b_1 \leq b_2$ 및 $b_1 \sim b_2$ 를 뜻합니다.

정의 (준비됨, Prepared). 무기명 투표 b 는 $\{\text{abort } b_{\text{old}} \mid b_{\text{old}} \leq b\}$ 집합에서 모든 스테이트먼트가 참인 경우에만 준비되었다고 표현합니다.

더 정확하게 말하자면 commit b 는 b 의 준비가 확정된 경우에만 투표를 유효하게 진행할 수 있으며, 이 때 노드는 해당하는 종단 스테이트먼트에 대한 연합 투표를 통해 이를 보장합니다. 이러한 스테이트먼트에 일괄적으로 투표하는 것이 편리하기 때문에 ‘ b 가 준비됨’이라고 작성하는 경우 항상 주변의 환경이 전체 종단 스테이트먼트 집합에 적용됩니다. 특히 노드는 이러한 모든 종단을 각각 투표, 수락 또는 확정하는 경우에만 b 가 준비되었음을 투표, 수락 또는 확정합니다.

SCP 노드는 무기명 투표 b 를 실행하고 해당하는 $b.x$ 값을 표면화하기 위해 우선 b 의 준비를 수락 및 확정된 뒤 commit b 를 수락 및 확정합니다. 첫 번째 온전한 노드가 commit b 에 투표하기에 앞서, 준비 단계는 연합 투표를 통해 결과적으로 모든 온전한 노드가 b 가 준비되었음을 확인할 수 있도록 합니다. 온전한 노드 v 가 commit b 를 수락하는 경우, 이는 결과적으로 $b.x$ 가 선택된다는 의미입니다. 그러나 제5.4.1장에서 논의된 바와 같이, v 는 v 가 오염된 경우를 감안하여 반드시 작용에 앞서 실행을 확정해야 합니다.

6.2.1. 구체적 무기명 투표 프로토콜.

그림 16은 각 노드가 유지하는 슬롯 별 상태를 나타내고 있습니다. 노드 v 는 다음과 같은 사항을 저장합니다. 해당 노드의 현재 상태 φ , 해당 노드의 현재 무기명 투표 b , 가장 최근에 준비한 양립 불가능한 무기명 투표 2개 (p, p'), 실행에 투표했으며 이후 종단을 수락하지 않았을 때(또는 이후 단계에서 실행을 수락 또는 확정하여) 존재하는 경우 가장 낮은 (c) 및 가장 높은 (h) 무기명 투표, 만약 현재 무기명 투표가 실패할 경우 시도할 다음 값인 z , 그리고 각 노드에서 가장 최근에 수신한 메시지(M). 무기명 투표 b, p, p' 및 h 는 단계 내에서는 감소하지 않습니다. 또한 $c \neq 0$, 즉 v 가 commit c 의 비준에 참가했을 수도 있는 경우 코드는 반드시 $c \leq h \leq b$ 를 보장해야 합니다. 이러한 불변은 노드가 현재의 무기명 투표 b 를 준비하기 위해 항상 합법적 투표를 실시할 수 있음을 보장합니다.

변수	의미
φ	현재 상태: PREPARE(준비), CONFIRM(확정) 또는 EXTERNALIZE(표면화) 중 하나
b	노드 v 가 현재 준비 및 실행을 시도하는 무기명 투표($b \neq 0$)
p', p	$p' \leq p$ 로 하여 준비된 것으로 수락된 2개의 가장 높은 무기명 투표들이며, 이 때 그러한 무기명 투표가 없는 경우 $p' = 0$ 또는 $p = p' = 0$ 입니다
c, h	준비의 경우, h 는 준비된 것으로 확정된 가장 높은 무기명 투표이며 존재하지 않는 경우 0입니다. 만일 $c \neq 0$ 인 경우, c 는 v 가 실행으로 투표하였고 종단을 수락하지 않은 가장 낮은 무기명 투표이며 h 가 가장 높은 무기명 투표입니다. CONFIRM의 경우, 각각 v 가 실행을 수락한 가장 낮고 가장 높은 무기명 투표입니다. EXTERNALIZE의 경우, 각각 v 가 실행을 확정된 가장 낮고 가장 높은 무기명 투표입니다.
	불변: $c \neq 0$ 이면 $c \leq h \leq b$ 입니다.
z	다음번 무기명 투표에 사용되는 값입니다. $h = 0$ 인 경우, z 는 합성 값입니다(제6.1장 참조). 그렇지 않으면 $z = h.x$ 입니다.
M	각 노드에서 본 가장 최신의 무기명 투표 메시지 집합입니다

그림 16. 각 슬롯에 대해 각 노드 v 가 유지하는 무기명 투표 상태입니다

PREPARE $v \mid b \mid p \mid p' \mid c \mid n \mid h \mid n \mid D$

슬롯 s 에 대해 노드 v 에서 보내는 메시지입니다. D 가 명시된 $Q(v)$. v 의 상태를 반영하는 다른 영역들입니다. $c.x$ 및 $h.x$ 값들은 $c.n \neq 0$ 인 경우 $c.x = h.x = b.x$ 입니다. 이 구체적인 메시지는 다음과 같은 개념 스테이트먼트들을 나타냅니다.

- $\{ \text{abort } b' \vee \text{accept}(\text{abort } b) \mid b' \leq b \}$ (b 를 준비하기 위해 a 가 투표)
- $\{ \text{accept}(\text{abort } b) \mid b' \leq p \}$ (p 가 준비되었음을 확정하기 위한 투표)
- $\{ \text{accept}(\text{abort } b) \mid b' \leq p' \}$ (p' 가 준비되었음을 확정하기 위한 투표)

그림 17. SCP의 무기명 투표 프로토콜에 존재하는 메시지

그림 17은 3개의 무기명 투표 프로토콜 메시지를 보여주고 있으며, 3가지 중 어느 메시지를 노드가 전송할지를 ϕ 가 결정함을 나타냅니다. 무기명 투표 메시지는 지명 메시지와 중복될 수 있어, $n = 0$ 일 때 노드가 NOMINATE 메시지에 대한 응답으로 z 를 업데이트할 수 있습니다. ' $a \vee \text{accept}(a)$ '는 수락의 정의 중 조건 1에 따라 슬라이스를 충족하기 위해 각 노드가 반드시 행해야 하는 사항이라는 점에 유의하십시오.

편의를 위해 노드 간 상태를 비교할 때는 아래처럼 노드에 속하는 영역들을 식별할 것입니다. 만약 v 가 노드라면 그림 16에서 설명하고 있는 바와 같이, 노드 v 의 상태에서 $b, p, p', \dots$ 의 값을 표현하기 위해 $b_v, p_v, p'_v, \dots$ 로 표기합니다. 이와 유사하게 v_m 에서 m 이 발신자로 표현되고, m 이 연합의 합의되는 v_m 의 상태에서 $b, p, p', \dots$ 에 대응하는 값을 표현하기 위해 $b_m, p_m, p'_m, \dots$ 로 표기합니다.

각 노드는 슬롯에 대한 무기명 투표 상태를 $\phi = \text{PREPARE}$, $z \leftarrow \perp$, $b \leftarrow 0 < z$, $M \leftarrow \emptyset$, 그리고 다른 모든 영역들 (p, p', c, h)는 무효인 무기명 투표 0으로 설정함으로써 초기 상태를 결정합니다. $z = \perp$ 인 경우, 노드는 무기명 투표 메시지를 수신할 수 있으나 발신할 수는 없습니다. $z \neq \perp$ 가 되면, 만약 $b.n = 0$ 인 경우 노드는 메시지 발신을 시작하기 위해 $z \leftarrow b.x$ 로 값 초기 설정을 다시 결정합니다. 이후 노드는 주기적으로 주변 노드와 메시지를 반복적으로 교환하며, ϕ 에 의해 나타난 무기명 투표 메시지를 발신합니다. 새로 수신한 메시지 m 을 M 에 추가함으로써 노드는 다음과 같이 자신의 상태를 업데이트합니다.

- (1) 만약 $\phi = \text{PREPARE}$ 이고 m 이 b' 로 하여금 준비된 b 와 같이 새로운 무기명 투표를 수락할 수 있게 하는 경우, p 와 p' 를 업데이트합니다. 이후 $p \geq h$ 또는 $p' \geq h$ 인 경우 $c \leftarrow 0$ 로 설정합니다.
- (2) 만약 $\phi = \text{PREPARE}$ 이고 m 이 b 로 하여금 새로 준비된 더욱 높은 무기명 투표를 확정하도록 하는 경우, h 를 해당하는 가장 높은 무기명 투표로 상승시키고 $z \leftarrow h.x$ 로 설정합니다.
- (3) 만약 $\phi = \text{PREPARE}$ 이고 $c = 0$, $b \leq h$, 그리고 $p \geq b$ 와 $p' \geq h$ 중 어느 것도 해당하지 않는 경우, $b \leq c \leq h$ 를 충족하도록 가장 낮은 무기명 투표로 설정합니다.
- (4) 만약 $\phi = \text{PREPARE}$ 이고 v 가 1개 이상의 무기명 투표를 실행하기로 수락하는 경우, c 를 해당하는 가장 낮은 무기명 투표로 설정하고 v 가 모든 $[\text{commit } b' \mid c \leq b' \leq h]$ 를 수락하도록 h 를 가장 높은 무기명 투표로 설정하며, $\phi \leftarrow \text{CONFIRM}$ 으로 설정합니다. 또한 h 를 업데이트된 뒤 $z \leftarrow h.x$ 로 설정하며, $h \leq b$ 인 경우를 제외하고 $b \leftarrow h$ 로 설정합니다.
- (5) 만약 $\phi = \text{CONFIRM}$ 이고 수신된 메시지가 b 로 하여금 새로 준비된 무기명 투표를 수락하도록 허용하는 경우, $p \leq h$ 되도록 p 를 수락 및 준비된 무기명 투표 중 가장 높은 값으로 설정합니다.
- (6) 만약 $\phi = \text{CONFIRM}$ 이고 v 가 많은 실행 메시지를 수신하였을 때 h 를 올리도록 하는 경우, v 가 (존재하는 경우) $[\text{commit } b' \mid b \leq b' \leq h']$ 를 모두 수락할 수 있도록 h 를 가장 높은 무기명 투표로 설정합니다. 만약 그러한 h' 가 존재하며 $h' > h$ 라면, $h \leftarrow h'$ 로 설정하고, 필요하다면 v 가 $[\text{commit } b' \mid c \leq b' \leq h]$

를 모두 수락할 수 있도록 c 를 가장 낮은 무기명 투표까지 상승시킵니다.

(7) 만약 $\phi = \text{CONFIRM}$ 이고 v 가 어떤 c 에 대하여 $\text{commit } c$ 를 확정한 경우, 해당 무기명 투표 중 가장 낮고 높은 값인 각각 c 와 h 를 설정하고, $\phi \leftarrow \text{EXTERNALIZE}$ 로 설정하고 $c.x$ 를 표면화한 뒤 중립합니다.

(8) 만약 $\phi \in \{\text{PREPARE}, \text{CONFIRM}\}$ 이고 $b < h$ 인 경우, $b \leftarrow h$ 로 설정합니다.

(9) 만약 $\phi \in \{\text{PREPARE}, \text{CONFIRM}\}$ 이고 $\exists s \subseteq M_v$ 를 통해 발신자 집합 $\{v_m \mid m \in S\}$ 이 차단이 $\forall w \in S, b_m > b_w$ 인 경우, p 와 p' 를 p_m, p'_m 으로 설정하며, 그러한 p 와 p' 가 존재하지 않는 가장 낮은 z 를 계수값입니다. M 을 업데이트한 다음 이전 단계를 반복합니다.

$c = 0$ 인 경우, 상기 프로토콜은 b 가 준비된 것을 확정하기 위해 연합 투표를 구현합니다. $c \neq 0$ 가 되면, 프로토콜은 모든 $c \leq c' \leq h$ 에 대하여 $\text{commit } c'$ 에 관한 연합 투표를 구현합니다. CONFIRM 단계에서 올바른 노드가 $\text{commit } c$ 를 수락한 경우, 해당 노드는 절대로 어떤 $c' < c$ 에 대해서도 $\text{commit } c$ 를 수락하거나 확정하려 시도하지 않습니다. 실행이 확정된 뒤에는 해당 무기명 투표의 값은 안전하게 쿼럼 교차를 가정하여 표면화할 수 있습니다.

특정 노드에서 발신한 모든 메시지는 $\langle \phi, b, p, p', h \rangle$ 를 통해 모든 순서가 결정되며 ϕ 가 가장 중요하고, h 가 가장 덜 중요한 영역입니다. 해당 영역들의 값은 그림 17에서 설명하는 바와 같이 메시지에서부터 결정될 수 있습니다. 모든 PREPARE 메시지는 모든 CONFIRM 메시지를 선행하며, 이는 결과적으로 주어진 슬롯에 대한 달의 EXTERNALIZE 메시지를 선행합니다. 네트워크가 메시지의 순서를 다시 배치할 수도 있으므로, 순서 배치는 M 의 메시지 순서 배치 시점과 관계없이 각 노드에서 가장 최신의 무기명 투표만을 포함할 수 있도록 합니다.

프로토콜 장점 설명에 관한 몇 가지 세부 내용입니다. ‘ $\text{abort } b' \vee \text{accept}(\text{abort } b')$ ’ 형식의 PREPARE 가 함의하는 스테이트먼트들은 v 가 $\text{abort } b'$ 에 대해 투표하는지 또는 확정하는지를 명시하지 않습니다. 수락의 정의에 있어서 이를 구분하는 것은 중요하지 않습니다. 구분에 대한 주석 입력은 v 가 중단 메시지를 수락했다면 실행에 투표했던(따라서 중단에 투표할 수 없는) 기존의 무기명 투표를 고려하지 않아도 됩니다. 실제로 v 가 $c \neq 0$ 일 때 c 를 변형하는 유일한 때는 이전 페이지의 1단계에서 실행에 투표한 모든 무기명 투표에 대해 중단을 수락한 뒤 c 를 다시 0으로 설정하려고 하는 경우뿐입니다. 반면 $c = 0$ 일 때 v 가 c 를 변형하는 유일한 때는 3단계에서 c 를 $c \geq b$ 값으로 설정하는 경우뿐입니다. 노드는 어떠한 $c \geq b$ 에서도 c 를 중단하는 데 투표하지 않으므로, 기존의 어떠한 중단 투표도 $\text{commit } c$ 와 모순되지 않습니다.

정리 11에서는 노드들로 하여금 수락한 내용을 다시 전달하도록 요구하고 있습니다. 이는 어떤 노드가 준비된 것으로 수락한 두 개의 가장 높은 양립 불가능한 무기명 투표가, 해당 노드에서 준비된 것으로 수락한 모든 무기명 투표를 포함한다고 하는 준비의 정의를 토대로 도출된 것입니다. 따라서 모든 메시지에서 p 와 p' 를 포함하는 것은 노드들이 확정 준비된 무기명 투표인 h 로 수렴하도록 합니다. 노드가 준비된 것으로 수락하는 무기명 투표는 반드시 노드가 준비된 것으로 확정하는 무기명 투표의 확대집합이어야 한다는 점에 유의하십시오. 따라서 1단계에서는 새로운 h 가 기존의 c 와 양립 불가능한 경우 $c \leftarrow 0$ 으로 설정할 것이므로 2단계는 $h \neq c \neq 0$ 이 되도록 h 를 설정할 수 없습니다.

v 가 EXTERNALIZE 메시지를 발신하는 시점에는 $\{\text{commit } b' \mid b' \geq c\}$ 를 수락한 상태입니다. 그러나 더욱 중요한 점은 $\{\text{commit } b' \mid c \leq b' \leq h\}$ 를 확정했다는 것입니다. v 는 $Q(v)$ 를 반영하지 않고 확정된 스테이트먼트를 수락함을 주장할 수 있는데, 왜냐하면 이미 슬라이스 중 하나가 만장일치로 합의한 것을 점검한 상태이기 때문입니다. 이는 EXTERNALIZE 의 설명에 존재하는 두 번째 암묵적 CONFIRM 메시지에서 D 를 대신하여 $\{\{v\}\}$ 가 나타나는 것을 설명합니다. D 를 제거하는 경우 도중에 쿼럼 슬라이스가 크게 변경

된 경우에도 하나의 정적 EXTERNALIZE 메시지로 하여금 다른 노드가 향후에 임의로 따라잡는 것을 도울 수 있게 합니다.

무기명 투표 메시지를 교환하려면 하나의 RPC만으로 충분합니다. 인수는 발신자의 최신 메시지이며 반환 값은 수신자의 메시지입니다. NOMINATE에서와 같이, D 또는 무기명 투표의 x 값이 암호화 해시인 경우, 캐시되지 않은 해시의 전송 회수를 위해 별도의 RPC가 필요합니다.

6.2.2. 타임아웃 및 무기명 투표 업데이트

만약 모든 온전한 노드가 동일한 무기명 투표 b 로 시작하는 경우, 이전 페이지의 1단계에서 9단계까지만으로도 충분히 commit b 를 확정하고 $b.x$ 값을 표면화할 수 있습니다. 그러나 지명 프로토콜이 수렴되기 전에 무기명 투표 프로토콜이 시작되는 경우 노드는 z 를 다른 값으로 하여 시작할 수 있습니다. 무기명 투표가 실패하거나 응답하지 않는 노드로 인해 실패할 수 있을 정도로 시간이 오래 소요되는 경우 노드는 타임아웃을 적용한 뒤 더욱 높은 무기명 투표로 다시 시도해야 합니다. 이러한 이유로 인해 노드는 타이머를 다음과 같이 도입합니다. (a) $\phi_v \neq \text{EXTERNALIZE}$ 에 해당하는 노드 v 는 $\exists S \subseteq M_v$ 일 때마다 타이머를 준비시켜, 발신자 집합 $U = \{v_m \mid m \in S\}$ 가 쿼럼이며, $v \in U$ 이고, $\forall m \in S, b_{m.n} \geq b_{v.n}$ 이 되도록 합니다. (b) 만약 타이머가 시작되는 경우, v 는 $b_v \leftarrow \langle b_{v.n} + 1, z_v \rangle$ 로 설정하여 무기명 투표를 업데이트합니다.

각각의 노드는 다양한 시간대에 무기명 투표를 시작시킬 수 있습니다. 그러나 조건 (a)는 쿼럼보다 앞서 노드 v 에서 설정된 타이머를 지연시킵니다. 이와 대조적으로, 이전 페이지의 9단계는 지나치게 뒤쳐진 노드로 하여금 타이머를 기다리는 일 없이 따라잡을 수 있도록 합니다. 종합해 보면 이러한 규칙들은 타이머가 충분히 길게 주어졌을 때 온전한 노드들이 동일한 무기명 투표에서 함께 시간을 보낼 수 있도록 합니다. 더불어 해당 시간은 타이머의 지속 시간에 비례하여 증가합니다. 지연율을 예측하는 일 없이 타임아웃이 충분히 길게 하기 위해, 구현을 통해 $b.n$ 함수로서 타임아웃을 증가시킬 수 있습니다.

6.3. 정확성

SCP 노드는 더 낮은 수를 지닌 양립 불가능한 무기명 투표에 대해 중단을 확정하도록 투표하기 전까지는 commit b 를 확정하는 투표를 할 수 없습니다. 올바른 노드는 상충하는 스테이트먼트들을 수락할 수 없기 때문에(즉 확정을 위해 투표할 수 없음), 이는 주어진 $\langle V, Q \rangle$ 에서, 정리 5는 올바른 노드의 집합 S 가 $V \setminus S$ 에도 불구하고 S 에 쿼럼 교차가 적용되는 한 상충되는 값들을 표면화할 수 없도록 합니다. 이러한 안전은 V 와 Q 가 슬롯 간에만 변경되는 경우 지속되나, 만약 (예를 들면 노드 충돌에 대응하여) 슬롯 중간에 변경되면 어떨까요?

재구성 진행 시 안전을 추론하기 위해 기존 및 신규 쿼럼 슬라이스 집합을 모두 결합하는데, 이 때 노드가 다양한 구성 시대의 메시지 조합을 바탕으로 결정할 수 있다는 사실을 반영합니다. 매우 보수적인 입장을 취할 때, 모든 과거 구성 각각마다 현재 구성의 집계에 적용되는 쿼럼 교차가 필요할 수 있습니다. 그러나 이 문제는 단순히 충돌한 노드와 불법 메시지를 발신한 노드를 분리함으로써 일부 완화할 수 있습니다.

정리 13. $\langle V_1, Q_1 \rangle, \dots, \langle V_k, Q_k \rangle$ 를 단일 슬롯에 대한 합의 도중 FBAS가 경험한 구성의 집합으로 합시다. $V = V_1 \cup \dots \cup V_k$ 와 $Q(v) = \{q \mid \exists j, v \in V_j \wedge q \in Q_j(v)\}$ 가 되도록 합시다. $B \subseteq V$ 를 B가 불법 메시지를 발신한 모든 불량한 노드를 포함하는 집합으로 하며, 이 때 $V \setminus B$ 는 여전히 충돌(무응답) 노드를 포함하도록 합시다. 노드 v_1 및 v_2 는 올바르다고 가정하고, v_1 은 x_1 을 슬롯에 표면화하며 v_2 는 x_2 를 표면화하는 것으로 가정합시다. 만일 $\langle V, Q \rangle_B$ 에 쿼럼 교차가 적용된다면, $x_1 = x_2$ 입니다.

증명. v_1 가 x_1 를 표면화하려면, 반드시 유사 쿼럼인 $U_1 \subseteq V$ 와 협력하여 $\text{accept}(\text{commit} \langle n_1, x_1 \rangle)$ 의 비준을 완료한 상태여야 합니다. 유사 쿼럼이라고 칭하는 이유는 비준화 과정에서 다수의 구성에 걸친 메시지가 개입됨으로 인해 U_1 이 특정 j 에 대해 $\langle V_j, Q_j \rangle$ 에서 쿼럼에 해당하지 않을 수 있기 때문입니다. 그럼에도 불구하고 비준화가 성공하기 위해 $q \subseteq U_1$ 로 하는 $\forall v \in U_1, \exists j, \exists q \in Q_j(v)$ 가 성립합니다. 이는 $q \in Q(v)$ 인 Q 의 구성으로부터 도출됩니다. 따라서 U_1 은 $\langle V, Q \rangle$ 에 포함된 쿼럼입니다. 비슷한 논리에 따라 유사 쿼럼인 U_2 는 반드시 $\text{accept}(\text{commit} \langle n_2, x_2 \rangle)$ 를 비준해야 하며, U_2 는 반드시 $\langle V, Q \rangle$ 에 포함된 쿼럼이어야 합니다. $\langle V, Q \rangle_B$ 의 쿼럼 교차에 따라, 반드시 $v \in U_1 \cap U_2$ 로 하는 몇몇 $v \in V \setminus B$ 이 존재해야 합니다. 추정에 따라 그러한 $v \notin B$ 는 양립 불가능한 무기명 투표를 수락한다는 주장을 할 수 없습니다. v 는 x_1 과 x_2 모두에 대하여 무기명 투표의 실행을 수락하기로 확정했으므로, 반드시 $x_1 = x_2$ 여야 합니다.

노드 v 의 활성화에 대해, 단일 슬롯 내부에서 일련의 구성 $\langle V_1, Q_1 \rangle, \dots, \langle V_k, Q_k \rangle$ 을 FBAS가 진행한 경우 몇 가지 요소에 대해 주의해야 합니다. 첫째로, 안전 위반은 활성을 약화시키며 정리 11에서 쿼럼 교차를 해야 하므로, 정리 13의 안전 전제 조건은 반드시 v 와 v 가 관여하는 노드 집합에 대해 성립해야 합니다. 둘째로 최신 상태의 불량한 노드 집합 $\langle V_k, Q_k \rangle$ 은 v 에 쿼럼을 부정하여 스테이트먼트들을 비준하는 것을 막을 수 있기 때문에 절대로 v -차단 속성을 지녀서는 안 됩니다. 마지막으로 v 의 상태는 스테이트먼트를 수락한다고 거짓으로 주장하는 v -차단 집합에 의해 오염된 적이 절대로 없어야 합니다. 이를 요약하자면 B가 불법적인 메시지를 발신한 노드의 집합이라면, 노드 v 는 다음과 같은 조건들을 만족할 때 점증적으로 온전한 것으로 간주합니다.

- (1) v 가 $\langle V_k, Q_k \rangle$ 의 최신 구성에서 온전하고
- (2) 현재 및 모든 과거 구성의 집합은 B에도 불구하고 쿼럼 교차를 가지며, 그리고
- (3) B는 $1 \leq j \leq k$ 중 어떤 값에서도 $\langle V_j, Q_j \rangle$ 에서 v -차단이 아닙니다.

다음의 몇 가지 정리들은 불량한 노드가 교착 상태로 올바른 노드를 유도할 수 없음을 보여줍니다.

정리 14. 쿼럼 교차를 가지는 FBAS에서, 만약 EXTERNALIZE 단계에 들어간 온전한 노드가 존재하지 않으며 무기명 투표 $\langle n, x \rangle$ 를 가지는 어떤 온전한 노드가 제6.2.2장에서 설명된 대로 타이머를 준비하는 경우, 충분한 소통이 주어졌을 때 모든 온전한 노드 v 는 타이머가 시작하기 전에 $b_v \geq n$ 으로 설정할 수 있습니다.

증명. $S = \{v \mid b_v \geq n\}$ 을 계수값이 n 이상인 노드의 집합으로 합시다. 가정을 토대로 할 때, S 는 온전한 노드를 포함합니다. 더불어 온전한 노드가 타이머를 준비했기 때문에 S 또한 반드시 쿼럼을 포함해야 합니다. S^+ 를 S 의 온전한 부분집합이라고 하고, S^- 를 S 에 포함되지 않은 온전한 노드의 집합이라고 합시다. 정리 10에 따라, $S^- = \emptyset$ (이 경우 정리가 자명함)이거나, S^+ 가 일부 $v \in S^-$ 에 대하여 v -차단에 해당합니다. 페이지 24의 9단계에 따라, v 는 $b_v.n \geq n$ 이 되도록 무기명 투표를 조정합니다. 이 시점에서 $S^- = \emptyset$ 에 해당하는 지점이 될 때까지 $S \leftarrow S \cup \{v\}$ 로 인수를 반복합니다.

정리/15. 타임아웃이 충분히 주어졌을 때, 온전한 노드가 $b.x = x$ 인 CONFIRM 단계에 도달했다면 결과적으로 모든 온전한 노드가 종료됩니다.

증명. 어떤 온전한 노드가 EXTERNALIZE 단계에 도달했다면 어떤 무기명 투표 c 에 대해 commit c 를 확정한 것입니다. 정리 11에 따라, 모든 온전한 노드는 commit c 를 확정한 것이며, 이후 24페이지의 7단계에서 종료될 것입니다. 그렇지 않다면 CONFIRM 단계의 온전한 노드는 $c = \langle n, x \rangle$ 일 때 commit c 를 수락하였습니다. 이에 앞서 어떤 온전한 노드는 c 가 준비됨을 확정하였습니다. 정리 11에 따라, 모든 온전한 노드는 결과적으로 $h \geq c$ 를 가지게 됩니다. 더불어 정리 8에 따라 어떠한 온전한 노드 v 도 abort c 를 수락할 수 없으므로, 어떠한 온전한 노드도 $p \geq c$ 가 되도록 하는 무기명 투표 p 를 준비된 것으로 받아들일 수 없습니다. 따라서 충분한 소통을 거친 뒤, 모든 온전한 노드는 영구적으로 $h \geq c$ 를 갖게 됩니다. 가장 낮은 b 를 가지는 온전한 노드 또는 노드들은 정리 14에 따라, 타이머를 가진 모든 온전한 노드가 동일한 무기명 투표 계수를 가질 때까지 무기명 투표를 증가시킵니다. 이들은 또한 동일한 $z = h.x = x$ 를 가지므로 모두 동일한 무기명 투표를 가질 것입니다. 만약 하나 이상의 온전한 노드가 더 높은 무기명 투표를 가지고 있어 프로토콜을 완료할 수 없는 경우, 더 높은 수의 무기명 투표를 가진 노드에는 타이머가 설정되지 않습니다. 따라서 더 낮은 수의 무기명 투표를 가진 노드는 타임아웃이 설정된 뒤 결과적으로 모든 온전한 노드가 동일한 무기명 투표에서 프로토콜을 완료할 수 있을 때까지 $b \leftarrow \langle b.n + 1, x \rangle$ 를 설정할 것입니다.

정리/16. 과거의 악의적 행동 여부를 불문하고, 악의적인 행동을 하는 노드가 새 메시지를 보내지 않는 시간 초과와 기간이 충분히 길게 주어지는 경우 SCP를 실행하는 온전한 노드는 종료될 것입니다.

증명. 정리 12에 따라, 모든 온전한 노드는 결과적으로 동일한 후보 값을 포함하는 Z 집합을 가지게 됩니다. 이 지점을 지나서 모든 온전한 노드 v 가 동일한 합성 값 $z = \text{combine}(Z)$ 를 가지는 것으로 가정합니다. $b.x = z$ 가 아닌 상태에서 준비된 무기명 투표 b 를 확정하는 온전한 노드가 없는 경우, 최대 1번의 타임아웃이 발생한 뒤 온전한 노드의 모든 신규 무기명 투표는 z 값을 가지며, 충분한 타임아웃이 주어지면 프로토콜을 완료하게 됩니다. 정리 15에 따라, 노드는 또한 어떤 온전한 노드가 준비 단계 이후까지 진행한 경우 완료할 것입니다.

나머지 사례로는 온전한 노드가 $h \neq 0$ 을 가지며 모든 온전한 노드가 $\phi = \text{PREPARE}$ 를 가지는 경우입니다. 정리 14에 따라, 가장 높은 $b.n$ 을 가진 온전한 노드 또는 노드들이 타이머를 준비시킬 때 타이머가 충분히 긴 경우 다른 노드들이 따라잡게 됩니다. 또한 정리 11에 따라, 타이머가 충분히 긴 경우 노드는 다음 번의 타임아웃에 앞서 h (확인된 가장 높은 무기명 투표) 값으로 수렴하며, 이 시점에 모든 온전한 노드는 b 를 동일한 값으로 상승시키고 프로토콜을 완료합니다.

정리 16은 SCP에 막다른 상태가 없음을 확인합니다. 그러나 매우 좋은 타이밍을 지니는 어떤 불량한 노드 집합이 메시지를 지연시켜 SCP 시스템을 영구적으로 선점할 수 있는데, 이 경우 온전한 노드 중 일부는 h 를 타이머가 실행되기 직전에 업데이트하며 나머지는 실행 이후에 업데이트됨으로 인해 온전한 노드들은 다음 번 무기명 투표에 수렴되지 못합니다. 노드는 슬라이스에서 불량 노드를 제거함으로써 그러한 공격으로부터 복구할 수 있습니다.

한 가지 대안은 프로토콜에 임의성을 추가하는 것인데, 예를 들면 24페이지의 2단계를 변경하여 z 를 $1/2$ (내지는 심지어 남은 타이머의 비율에 비례하는) 확률로 업데이트하는 것입니다. 그러한 접근 방식은 확률이 1인 경우 종료되지만 대부분 혹은 모든 노드가 올바르게 작동하거나 고장 및 정지 상태의 일반적인 경우에는 예상 실행 시간이 더 길어지게 됩니다.

7. 제한사항(LIMITATIONS)

SCP는 노드가 충분한 쿼럼 슬라이스를 선택했을 때만 안전을 보장할 수 있습니다. 제3.2장에서는 이를 합리적으로 기대할 수 있는 이유에 대해 설명합니다. 그럼에도 불구하고, 사용자가 구성 가능한 매개변수에 따라 보안이 달라진다면 사람들이 이를 잘못 설정할 가능성이 항상 존재합니다.

사람들이 쿼럼 슬라이스를 올바르게 설정하고 SCP가 안전을 보장하더라도, 안전 그 자체만으로는 연합 시스템에서 발생할 수 있는 다른 보안 문제를 배제하지 못합니다. 예를 들어 금융 시장의 경우, 광범위하게 신뢰받는 노드는 네트워크에서 자신이 가진 위치를 활용하여 정보를 얻음으로써 선취매매(프론트러닝) 또는 기타 비윤리적 활동에 참여할 수 있습니다.

비잔틴 노드는 SCP의 입력 측에서 트랜잭션을 필터링하려고 시도할 수도 있으며 이에 해당하지 않는다면 올바른 결과를 생성하게 됩니다. 올바른 노드가 모든 트랜잭션을 수락하고, 조합 함수가 트랜잭션의 합집합을 취하며, 온전한 노드가 있는 경우 이러한 필터링은 결과적으로 피해자 트랜잭션을 확률 100%로 차단하는데 실패하지만 그럼에도 불구하고 지연을 부과할 수는 있습니다.

SCP의 안전은 최적 상태이지만, 성능과 통신 지연은 그렇지 않습니다. 일반적인 경우로 노드가 현재와 양립 불가능한 무기명 투표를 실행하기 위해 기존에 투표하지 않았다면, 통신 라운드 수를 1단계 줄이는 것이 가능합니다. SCP의 기존 버전에서도 마찬가지로 진행했으나 프로토콜 설명은 더욱 복잡했습니다. 첫째로 실패한 노드가 기존에 발신했던 서명된 메시지를 캐시하고 재전송할 것을 요구했습니다. 둘째로 준비 메시지들에서 중단 스테이트먼트의 투표와 확정에 관한 구분을 적당히 간과하는 것이 더 이상 불가능했으므로 노드는 중단 투표에 대해 잠재적으로 수많은 예외 내역 목록을 발신해야 했습니다.

SCP는 제6.3장에서 논의한 바와 같이 끊임없이 선점에 시달릴 수 있습니다. 해결되지 않은 의문은, 임의성은 배제한 채 상이한 프로토콜이 제한된 통신 지연을 가정하면서도 시간 초과가 발생하는 바로 그 시점에서 악성 메시지를 지속적으로 전송하는 비잔틴 노드를 허용하는 가운데 종료를 보장할 수 있는지의 여부입니다. 그러한 프로토콜은 FLP 불가능성 결과에 의해 배제되지 않습니다[Fischer et al. 1985]. 그러나 동기화를 가정했을 때 종료를 보장하기 위한 다음과 같은 2가지 주요 기술은 FBA 모델에 직접 적용되지 않습니다. PBFT [Castro and Liskov 1999]의 경우 순차 순환 대기 방식으로 리더를 선택하며, 이는 노드가 멤버십에 합의하지 않는다면 직접 적용할 수 없습니다. (제6.1장의 우선순위 항목들에 관한 내용이 조정될 수 있습니다.) 비잔틴 장군 프로토콜(Byzantine Generals Protocol) [Lamport et al. 1982]은 메시지를 중계하여 다양한 정직한 노드에게 여러 다른 내용을 알리는 불량한 노드들을 상쇄시키는데, 이러한 접근법은 노드들이 자신의 슬라이스에 포함된 뚜렷한 불량한 노드에 의존하는 경우 도움이 되지 않습니다. 한편 또 다른 가능성은 무작위성과 동기성을 모두 활용하여 확률 1로 종료하되, 동기성 가정을 하지 않는 Ben Or 식 무작위 프로토콜[Ben-Or 1983]보다 짧은 예상 시간 안에 종료하는 것입니다. 무작위 중앙화된 비잔틴 합의 프로토콜의 속도를 향상시키는 공개적 코인 기술은 연합 한계치 서명에서 일부 암호화 혁신은 예외로 하나, 연합 모델에 적용하기 어려운 것으로 보입니다.

그러나 만약 올바른 노드 v 가 완전히 악의적이면서 공모하는 v -차단 집합을 경험한 경우 실패한 노드를 수용하기 위해 슬롯 중간에 슬라이스를 변경하는 것은 활성에 문제가 있습니다. 다행스러운 점은 S 에 오염된 노드가 있더라도, $V \setminus S$ 에도 불구하고 정리 13을 통해 쿼럼 교차가 적용된 어떤 올바른 노드 집합 S 에 대한 안전이 보장된다는 것입니다. 하지만 올바른 노드가 속아서 잘못된 실행 메시지를 확정하도록 투표한 경우 Q 를 업

데이트하는 것으로는 노드의 차단을 해제하기에 충분하지 않을 수도 있습니다. 그러한 경우 노드는 반드시 기존의 투표를 거부해야 하며, 이는 새로운 노드 이름을 사용하여 시스템에 재가입해야지만 가능합니다. 그러한 복구를 자동화하기 위해서는 다른 노드가 새로 변경된 노드를 인식하고 슬라이스를 자동으로 업데이트하도록 하는 등의 방법이 존재할 수 있습니다.

FBA 모델은 시간 경과함에 따라 참가자의 연속성을 필요로 합니다. 모든 노드가 동시에 영구적으로 떠나는 경우, 합의(Consensus)를 재시작할 때 중앙집중식 조정 또는 사람들의 개입에 대한 합의(Agreement)가 필요합니다. 이와 대조적으로, BTC를 비롯한 작업 증명 시스템은 갑작스럽게 완전한 전환이 일어날 수 있으나 사람이 거의 개입하지 않고도 계속 작동할 수 있습니다. 반면 노드가 반환되는 경우 FBAS는 임의의 긴 중단 기간으로부터 복구할 수 있지만, 작업 증명 체계는 중단 기간 도중에 공격자가 포크를 작업할 가능성에 직면하게 됩니다.

흥미로운 가능성은 SCP를 활용함으로써 구성 매개변수의 변경이나 애플리케이션 프로토콜의 업그레이드에 투표하여 갈등을 촉진하는 것입니다 [Clark et al. 2005]. 이를 실시할 수 있는 한 가지 방법은 매개변수를 업데이트하는 특수 메시지를 지명하는 것입니다. 그 후에 후보 값은 값의 집합과 매개변수 업데이트의 집합을 모두 포함할 수 있게 됩니다. 이 접근법이 지니는 큰 한계점은 안전을 약화시킬 만큼 크지는 않지만 시스템의 쿼럼을 거부할 정도로 충분히 큰 악의적인 노드 집합이 거짓을 전달함으로써 비준된 적이 없는 Y에 구성 변경을 일으킬 수 있다는 점입니다. 전체 쿼럼의 동의를 필요로 함과 동시에 활성을 위태롭게 하지 않는 방식으로 매개변수 변경에 투표할 수 있는 방법은 여전히 해결되지 않은 의문으로 남아 있습니다.

8. 요약(SUMMARY)

비잔틴 합의는 분산된 시스템들로 하여금 효율성, 표준 암호화 보안, 신뢰할 수 있는 참가자들의 지정에 대한 유연성을 확보함으로써 합의에 도달할 수 있도록 오랜 기간 동안 도움을 제공하고 있습니다. 최근에 BTC는 탈중앙화된 합의에 혁신적인 개념을 도입하여, 많은 신규 시스템과 연구에 관한 난제들을 불러일으켰습니다. 본 백서는 비잔틴 합의에서 통상적으로 제공해 온 장점을 유지하는 동시에 탈중앙화된 합의(Decentralized Consensus)에 도달할 수 있도록 하는 모델인 연합 비잔틴 합의(Federated Byzantine Agreement, FBA)에 대해 소개하고 있습니다. FBA와 기존의 비잔틴 합의 시스템 간의 주된 차이점은 바로 FBA가 참가자들의 개별 신뢰 결정을 토대로 쿼럼을 형성함으로써, 인터넷과 유사한 유기적 성장 모델을 허용한다는 것입니다. 스텔라 합의 프로토콜(Stellar Consensus Protocol, SCP)은 FBA가 악의적으로 행동하는 참가자들에 맞서 최적의 안전을 확보할 수 있도록 하는 구조입니다.

감사의 말

Jed McCaleb은 본 백서의 작성에 영감을 불러 일으켰으며 피드백 및 권장 용어를 제공하였고, 다양한 추측을 통한 사고의 진행에 도움을 주었습니다. Jessica Collier는 백서의 작성에 협력하였습니다. Stan Polu는 SCP의 첫 구현을 진행하였으며 이 과정에서 너무나도 귀중한 교정, 권고, 간소화 및 피드백을 제공해 주었습니다. Jelle van den Hooff는 쿼럼 교차와 연합 투표에 관하여 백서의 구조를 재설정함에 있어 핵심

개념을 제공했으며, 용어, 조직 및 발표에 대한 귀중한 조언 또한 제공하였습니다. Nicolas Barry는 프로토콜을 구현하면서 본 백서에서 몇 가지 버그를 찾아내었고, 설명이 필요한 부분들을 식별해 내었습니다. Ken Birman, Bekki Bolthouse, Joseph Bonneau, Mike Hamburg, Graydon Hoare, Joyce Kim, Tim Makarios, Mark Moir, Robert Morris, Lucas Ryan, 그리고 Katherine Tom은 본 백서의 원고와 씨름하며 오류와 혼란의 여지가 있는 내용들을 찾아내 주었고 도움이 되는 조언을 아끼지 않았습니다. Eva Gantz는 도움이 되는 동기와 참고 자료들을 제공했습니다. Winnie Lim은 그림들의 작성을 지도했습니다. 레딧(Reddit) 커뮤니티와 타호-LAFS 그룹은 SCP의 초기 버전에서 검열 관련 취약사항을 지적해 주어, 지명 프로토콜의 개선을 이끌어 주었습니다. 마지막으로 스텔라 팀 모두의 지원, 피드백 그리고 격려에 감사드리고 싶습니다.

면책 공고

본 백서에 Mazières가 기여한 내용은 유급 컨설턴트로서 이루어진 것이며, 스탠포드 대학교 측의 의무나 책임과는 관계가 없습니다.