Anycast Agility: Network Playbooks to Fight DDoS

0. 서론

IP Anycast는 DNS 및 CDN과 같은 서비스에 사용되어 DDoS 공격을 처리할 수 있는 능력을 제공함. DDoS 공격 동안 서비스 운영자는 Anycast 사이트 간에 트래픽을 재분배하여 사용하지 않거나 더 큰 용량을 가진 사이트를 활용함. 사이트 트래픽과 공격 크기에 따라 운영자는 다른 사이트의 작동을 보존하기 위해 몇몇 사이트에 공격자를 집중시킬 수도 있음. 운영자는 공격 중 이러한 조치를 사용하지만 이를 어떻게 수행하는지에 대해서는 체계적이거나 공개적으로 설명된 바가 없음. 해당 논문은 DDoS 상황에서 BGP를 사용해 트래픽을 전환하는 여러 방법을 설명하고 응답 Playbook이 [1]공격 중에 선택할 수 있는 옵션의 메뉴를 제공할 수 있음을 보여줌. 이 Playbook에서 적절한 응답을 선택하기 위해 우리는 또한 공격 중 운영자의 시야가 불완전함에 불구하고 진짜 공격 크기를 추정하는 새로운 방법을 설명함. 마지막으로 운영자의 선택은 분산 라우팅 정책에 의해 제한되며 모든 정책이 도움이 되는 것은 아님. 우리는 특정 Anycast 배포가 이 Playbook의 옵션을 어떻게 제한할 수 있는지 그리고 다수의 Anycast 네트워크에 걸쳐 얼마나 일반적으로 적용될 수 있는지 처음으로 조사함.

1. 서론

Anycast Routing은 DNS나 CDN과 같은 서비스에서 사용되며 여러 지리적으로 분산된 위치에서 동일한 prefix를[2] 발표하는 경우에 사용됩니다. 1993년에 정의된 Anycast는 초기 2000년대에 DNS 루트에서 광범위하게 배포되었으며 현재는 많은 DNS 제공업체 및 콘텐츠 전달 네트워크에서 사용되고 있습니다. IP Anycast에서 BGP는 각 네트워크를 특정 Anycast 사이트로 경로 지정하여 세계를 수거지로 나눕니다. BGP는 일반적으로 네트워크를 가까운 Anycast 사이트와 연결하여 일반적으로 좋은 대기 시간을 제공[3]합니다. Anycast는 각 사이트가 독립적이므로 하나의 사이트가 DDoS 공격에 압도당할 경우에도 과부하가 아닌 사이트는 영향을 받지 않습니다. DDoS 공격은 점점 커지고 더 자주 발생하고 있습니다. 서로 다른 루트 서버 및 Anycast 서비스는 빈번한 DDoS 공격 이벤트를 보고하고 있습니다. 서로 다른 자동화된 도구들은 공격을 생성하기가 더 쉽게 만들고 있으며 일부는 DDoS-as-a-Service[4]를 제공하여 비전문가 또한 DDoS 공격을 할 수 있도록 서비스를 제공하고 있습니다. DDoS 공격은 시간이 지날수록 증가하고 있으며 2020년 CLDAP 공격은 크기가 2.3 Tb/s를 초과했으며 2021년 VoIP.ms 공격은 5일 이상 지속되었습니다. 공격 소스의 저장소는 수백만 개의 취약한 IoT로 인해 Botnet을 통해 성장하고 있습니다. 운영자들은 Anycast를 이용해 DDoS 공격을 공격 발원지로 돌려 해결합니다. 서비스 운영자는 공격이 지속적으로 유입됨에 따라 대응하기를 원합니다. 아마도 과부하가 걸린 사이트에서 다른 사이트로 부하를 이동시키는 등. 이전의 DDoS 이벤트에 대한 연구들은 운영자들이 이러한 조치를 취한다는 것을 보여주었지만, 최상의 조치가 어떤 공격 크기와 위치에 따라 Anycast 사이트 용량과 비교될 때 어떤 것이 가장 좋은 지 제안되었습니다. 이전 연구들은 대응책을 제안했으며 운영자들이 공격 중에 라우팅을 변경한다는 사실을 알고 있지만, 라우팅 선택이 트래픽을 어떻게 변경하는지에 대한 평가는 아직 제한적입니다. 아주 최근에 path poisoning을 검토한 연구가 있었지만, 공격 중에 라우팅을 사용하는 방법에 대한 구체적인 공개 안내는 없습니다. 본 논문의 목표는 DDoS 공격 중에 Anycast를 통해 트래픽을 균형 있게 조절하는 것을 목표로 합니다. 우리의 첫 번째 기여는 진정한 공격율을 추정하고 응답을 계획하는 새로운 메커니즘을 갖춘 시스템입니다. 먼저, 패킷 손실이 발생할 때라도 실제 공격양을 추정하는 새로운 메커니즘을 제안합니다. 각 사이트의 상대적 부하를 추정하는 것은 방어의 첫 번째 단계이므로, 방어자는 다양한 사이트의 용량에 부하를 맞출 수 있거나 일부 사이트가 가능한 한 많은 공격을 흡수해야 하는지 결정할 수 있습니다. 둘째, 운영자가 다중 사이트 Anycast 시스템에서 어떻게 트래픽을 다시 밸런스 있게 할 것인지 미리 예측할 수 있도록 BGP Playbook을 개발합니다. 이 두 가지 요소는 Playbook에 따라 Anycast 라우팅을 조정하여 DDoS 공격에 자동으로 응답하거나 사람 운영자에게 조치를 권장할 수 있는 시스템을 제공합니다. 두 번째 기여는 다중 홉 TE[5]를 위한

---

1 특정 상황에서 사용할 수 있는 다양한 전략이나 절차들을 모아둔 지침서나 가이드를 의미함.

2 IP를 말함. IPv4 또는 IPv6

3 Anycast로 연동된 서비스는 클라이언트로부터 가장 빠른 서버로 라우팅 되기에 짧은 지연시간을 제공함.

4 DDoS 공격을 제공하는 서비스를 말함. 저렴한 비용에 비전문가도 웹사이트를 통해 쉽게 DDoS 공격을 할 수 있음.

5 다수의 네트워크 장비 또는 라우터를 통해 패킷이 전달되는 상황에서 트래픽을 효과적으로 관리하고 최적 경로를 선택함.

라우팅 옵션의 작동 방식을 이해하는 것입니다: AS prepending[6], community string [7]및 path poisoning[8]. 널리 알려져 있지만 이러한 메커니즘들이 얼마나 이용 가능하고 효과적인지는 잘 모릅니다. 우리는 AS prepending은 거의 어디에서나 이용 가능하지만 community string 및 path poisoning의 지원이 크게 다르다는 것을 보여줍니다. 또한 이러한 메커니즘들의 효과는 크게 다르며, 오늘날의 "더 평면 인터넷[9]" 로 인해 AS prepending은 대부분의 트래픽을 거의 모두 또는 거의 없는 곳으로 이동시킵니다. community string은 더 세분화된 제어를 제공하지만 그 지원이 일관되지 않음을 보여줍니다. path poisoning은 멀리 떨어진 여러 홉에서 제어를 제공할 수 있지만 community string과 마찬가지로 종종 필터링 됩니다. 이러한 요소가 다중 사이트 및 Anycast 시스템의 상호 작용과 결합될 때, BGP Playbook이 운영자를 안내하는 데 중요합니다. TE의 효과는 종종 특정 Anycast 배포의 peer 및 위치에 특정되므로 우리는 결과가 다른 위치 및 Anycast 사이트 수에 얼마나 민감한지 탐색합니다. 마지막으로 우리의 최종 기여는 실제 방어가 성공적임을 보여주는 것입니다. 우리는 실험실에서 실제 공격을 재생하고 TE가 방어할 수 있다는 것을 보여줍니다. 물론 모든 공격에 대해 단일 방어가 가능한 것은 아니지만, 이 예제들은 우리의 접근 방식이 다양한 용량 및 다양한 DDoS 공격에 성공적인 방어를 제공한다는 것을 보여줍니다. 이들은 우리의 알고리즘 및 프로세스 기여(공격 크기 추정 및 Playbook 작성)가 실용적인 응용이 있다는 것을 보여줍니다. 우리의 작업은 공개적으로 이용 가능한 데이터셋을 사용합니다. 우리의 실험의 입력 및 결과에 대한 데이터셋은 무료로 이용할 수 있습니다. 우리의 데이터는 개인이 아닌 서비스에 관한 것이므로 개인 정보 우려가 없습니다.

2. 관련 연구

anycast 라우팅은 라우팅, 성능 및 DDoS 방지 측면에서 오랫동안 연구되어 왔습니다.

트래픽을 조절하기 위한 BGP: 이전 연구에서는 BGP가 링크의 부하를 균형있게 조절하는 데 효과적임을 보였습니다. 그러나 Ballani 등은 anycast가 효과적인 부하 분산을 위해 계획과 주의를 필요로 한다고 보여주었습니다. 다른 연구에서는 패킷 손실, 지연 및 지터를 기반으로 BGP를 조작하는 것을 제안했습니다. 우리는 Ballani의 anycast 계획을 기반으로 하여 BGP Playbook을 제안하고 그 효과를 연구합니다.

Chang 등은 트래픽 엔지니어링을 위해 BGP community를 사용하는 것을 제안했습니다. 최근의 연구에서는 IXPs와 ISP에서 BGP community를 블랙홀 라우팅에 사용하는 것을 조사했습니다. Smith와 Glenn은 링크 혼잡을 해결하기 위해 path poisoning을 검토했습니다. 이러한 방법들은 각각 방어용 라우팅에서 중요한 옵션입니다만, 우리는 운영자가 이들 중에서 선택할 수 있는 시스템을 제시합니다. 단일 방법으로는 모든 공격에 대응할 수 없기 때문에 여러 선택지를 갖춘 시스템이 필요합니다. 예를 들어, 우리는 Tier-1 AS를 poisoning으로 만들었을 때 poisoning이 작동하지 않음을 보여줍니다.

anycast 성능: 대부분의 anycast 연구는 효율적인 전달과 안정성에 중점을 두었습니다. 나중에는 클라이언트의 근접성을 명시적으로 조사했습니다. 일부 연구에서는 anycast를 토폴로지 변경을 통해 개선하려고 시도했습니다. DDoS를 위한 anycast 서비스는 이미 상용 솔루션(e.g., Amazon, Akamai 및 AT&T)에서 사용되고 있습니다. 그러나 이들은 라우팅 조작을 DDoS 방어 메커니즘으로 사용하는 방법에 대해 다루지 않습니다.

anycast catchment 제어를 DDoS 완화 도구로: 우리의 지식으로는 anycast 사이트 간의 부하를 흡수하거나 이동시켜 DDoS 공격을 다루는 아이디어가 처음으로 2016년에 발표되었습니다. Kuipers 등은 그 작업을 보완하여 우리가 §3.4에서 검토했고 실험을 통해 탐구한 트래픽 이동 접근 방법을 정의했습니다. 우리는 응답을 안내하기 위해 BGP Playbook 개념을 발전시켰으며, 공격 크기를 추정하는 새로운 접근 방식을 설명하고, 마지막으로 실제 사례에서 응답이 효과적일 수 있다는 것을 보여줍니다.

상용 및 자동화된 솔루션: 대부분의 출판된 상용 안티-DDoS 솔루션은 트래픽을 방어 인프라로 유도하기 위해 라우팅을 사용합니다. 때로는 모든 사이트가 트래픽 분석을 지원하기 위해 개인 백본을 통해 연결되어야 합니다. 다른 방어 방법으로는 모든 트래픽을 scrubbing

---

[6] AS경로를 길게 보이도록 하여 트래픽을 특정 경로로 유도하거나 특정 경로부터 트래픽을 제외시키는 데 사용됨.

[7] 특정 경로에 대한 추가 설정이나 우선순위를 지정하는데 사용됨.

[8] 경로 메커니즘이 특정 경로를 사용하지 않도록 강제하는 방법임.

[9] 네트워크 구조가 계층적이고 복잡한 형태에서 벗어나

센터로 전환한 다음 좋은 트래픽을 목적지로 터널링하는 BGP가 사용됩니다. 기타 방법으로는 DNS 조작, anycast 프록시가 있으며, 이는 DNS anycast 배포 자체에는 사용할 수 없습니다. 우리는 문제를 외부로 위탁하는 대신에 어떻게 방어할 수 있는지 탐구합니다. 다른 자동화 방어 기법으로는 반응형 자원 관리, 클라이언트-서버 재할당 및 필터링 접근 방식이 있습니다. 우리의 방법은 anycast에서 사용 가능한 리소스를 효율적으로 사용하기 위해 TE 접근 방식을 사용합니다.

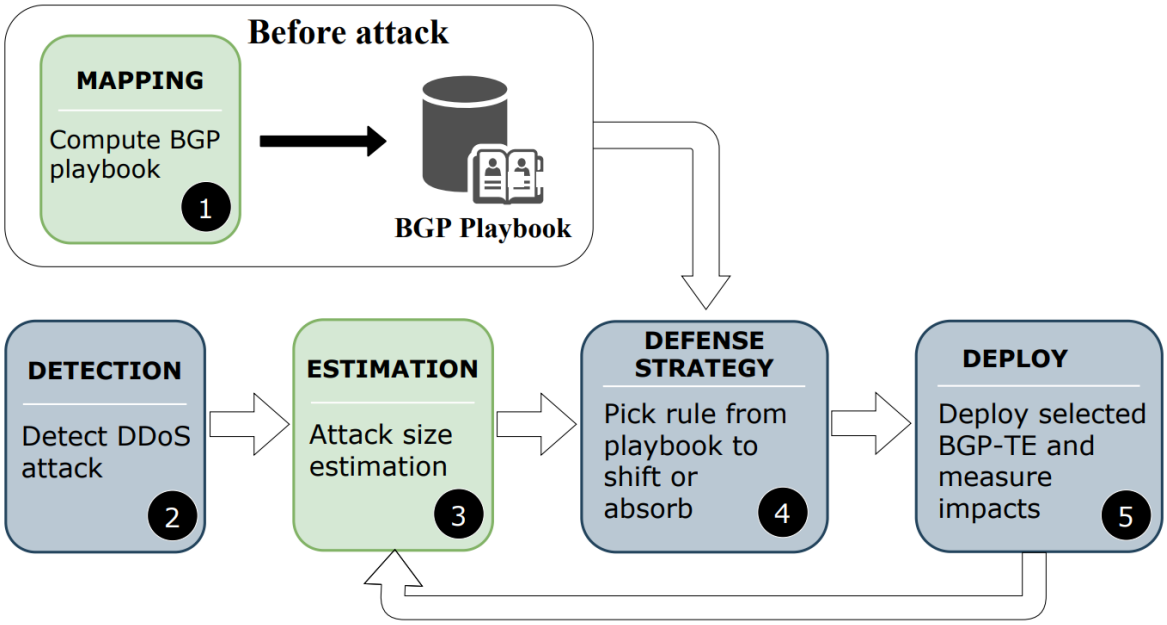


그림 1: 분할 과정 개요

3. DDoS 대응 메커니즘 3가지

이 섹션에서는 BGP 방어 프로세스, BGP Playbook 사전 계산 방법, 공격 규모 추정 및 TE(트래픽 엔지니어링) 응답 선택에 대해 설명합니다.

3.1 개요 및 의사 결정 지원

도 1에서 DDoS에 대한 방어 작업이 어떻게 이루어지는지 보여줍니다. DDoS에 대한 방어는 탐지로 시작됩니다 2, 그런 다음 방어를 계획하고 4 실행하며 공격이 완화되거나 종료될 때까지 이 프로세스를 반복합니다(도 1의 하단 사이클). 공격을 감지하는 것은 큰 공격이 시스템 성능에 영향을 미치므로 간단합니다. 과제는 최상의 응답을 선택하고 빠르게 반복하는 것입니다.

우리는 공격 대응에 두 가지 새로운 구성 요소를 가져왔습니다(도 1의 색칠된 연한 녹색): 공격 전 매핑 및 공격 시작 시 공격 규모 추정입니다. 매핑 1(§3.2에서 설명)은 방어자에게 계획된 응답에 대한 Playbook과 이들이 어떻게 트래픽 믹스를 변경할지에 대한 정보를 제공합니다. 규모 추정 3(§3.3에서 설명)은 방어자가 얼마나 많은 트래픽을 이동해야 하는지 결정하고 Playbook에서 유망한 응답을 선택할 수 있도록 합니다. 이러한 도구들은 특정 사이트에서 트래픽을 줄이는 방법뿐만 아니라 해당 트래픽이 이동할 사이트도 이해하는 데 도움이 됩니다.

이러한 구성 요소들은 측정 및 공격 규모 추정, 방어 선택, 그리고 배포 사이에서 반복되는 우리의 자동화된 응답 시스템(§3.4)에서 결합됩니다. 방어는 매핑 중에 구축된 Playbook을 사용합니다. 우리는 §6.4에서 예제 Playbook을 제공합니다. 이러한 방어가 testbed 실험에서 어떻게 작동하는지를 §8에서 보여줍니다.

우리의 시스템은 특정 anycast IP 주소에서 특정 양의 인프라를 사용하여 운영되며 제3자 스크러빙 서비스를 사용하지 않는 서비스를 대상으로 설계되었습니다. 다중 anycast 서비스, DNS redirection 또는 스크러빙 서비스를 가진 CDN 운영자는 우리의 접근 방식을 사용할 수 있지만 이러한 다른 도구들을 갖고 있습니다. 그러나 많은 운영자는 스크러빙 및 DNS redirection을 사용하지 않거나 사용하지 않는 것을 선호합니다. (모든 DNS 루트 서버, 국가 자주성을 중요시하는 많은 ccTLD 및 스크러빙 서비스 자체의 운영자 등) 우리의 접근 방식은 다른 사이트에서 여분의 용량을 가지고 있는 부피 공격에 대비합니다. DDoS는 서비스의 가용성을 저해하기 때문에 공격 중 사이트

선택의 최적화는 고려하지 않아도 됩니다.

3.2 측정: Anycast 매핑

공격 전에 anycast 서비스의 catchment를 매핑하여 방어자가 공격 중에 신속하게 정보에 기반한 선택을 할 수 있도록 하여 BGP Playbook을 작성합니다(§6.4).

Anycast catchment를 매핑하기 위해 Verfploeter [20]를 사용했습니다. 활성 프로버(ICMP echo request)로서 Verfploeter는 모든 ping 응답 IPv4 /24를 관찰하고 어떤 anycast 사이트가 응답을 받았는지 매핑합니다. Appendix A에서 anycast와 Verfploeter에 대한 자세한 설명을 제공합니다. 매핑은 공격 전에 수행되므로 매핑 속도는 문제가 되지 않습니다.

대신 각 사이트에서 어떤 고객이 시간이 지남에 따라 관찰되는지 또는 RIPE Atlas [3, 73]와 같은 분산된 시각점에서 측정하는 방법으로 트래픽을 매핑할 수 있습니다.(운영자는 이미 이 정보를 최적화하기 위해 수집할 수 있습니다.) 매핑은 현재 catchment뿐만 아니라 공격 중에 우리가 만들 수 있는 잠재적인 변화도 고려해야 합니다. 이러한 전체 매핑은 가능한 변화를 매핑하는 인접한 BGP prefix에서 지속적으로 실행될 수 있는 Verfploeter를 사용하여 수행할 수 있습니다. 이러한 매핑 프로세스는 트래픽이 어떻게 이동할 수 있는지 예측하는 데 중요합니다. 나중에 BGP 제어가 라우팅 정책의 세분화(§6) 및 anycast 사이트의 배치(§7)에 의해 제한된다는 것을 보여줄 것입니다.

경로 대안과 함께 사전 계산된 맵의 한 가지 어려움은 경로가 모든 AS에 의해 영향을 받는다는 것입니다. 따라서 다른 AS의 라우팅 정책 변경으로 인해 맵이 시간이 지남에 따라 변할 수 있습니다. 다행히 이전 연구에 따르면 anycast catchment는 상대적으로 느리게 변한다는 것을 보여줍니다. 또한 우리의 BGP Playbook이 시간이 지남에 따라 안정적임을 보여줍니다(§6.4 및 부록 E).

3.3 공격 규모 추정

공격 감지 후 DDoS 방어의 첫 번째 단계는 공격 규모를 추정하여 얼마나 많은 트래픽을 이동할지를 결정하는 것입니다. 우리의 목표는 제공된 로드, 각 사이트로 보내진(제공된) 트래픽을 측정하는 것입니다. DDoS 중 제공된 로드는 공격과 정당한 트래픽의 혼합으로 팽창되며 서비스 상위에서의 손실은 우리가 직접적으로 실제 제공된 로드를 관찰할 수 없다는 것을 의미합니다. 우리는 후에 실제 DDoS 사건으로 접근을 평가합니다(§4).

아이디어: 우리의 통찰력은 우리가 상위 손실이 있을 때도 서비스에 도착한 일부 알려진 트래픽의 변화를 기반으로 실제 제공된 로드를 추정할 수 있다는 것입니다.

실제 제공된 로드가 서비스에 도착하는 양을 알기 위해서는 어떤 비율의 정당한 트래픽을 추정해야 합니다. 그런 다음 공격 중에 이 트래픽이 얼마나 감소하는지 관찰하여 상위 손실을 추론할 수 있습니다. 불행히도 모든 정당한 트래픽을 결정할 일반적인 방법이 없습니다. 왜냐하면 정당한 보내는 측은 트래픽 비율을 변경하고 공격자는 종종 자신의 트래픽을 정당하게 만듭니다. 우리의 목표는 일부 특정한 정당한 트래픽을 신뢰할 수 있게 추정하는 것이며 다음에 몇 가지 소스를 설명합니다.

트래픽 소스: 알려진 정당한 트래픽의 여러 가능한 소스가 있습니다 – 우리는 알려진 측정 트래픽 및 정기적인 트래픽 소스를 고려합니다. [5] DNS의 경우, RIPE Atlas는 많은 곳에서 보내는 알려진 좋은 트래픽을 제공합니다. RIPE는 약 10,000 개의 공개적으로 사용 가능한 시각점에서 지속적인 트래픽을 생성합니다. RIPE 시각점마다 약 240초마다 쿼리하며 약 2500 쿼리/분의 트래픽이 있으므로 제공된 로드의 좋은 추정치를 제공 할 수 있습니다. (RIPE Atlas는 DNS에 특화되어 있지만 다른 상용 서비스도 종종 유사한 유형의 알려진 모니터링 트래픽을 가지고 있습니다.)

각 사이트에서 알려진 좋은 트래픽을 찾으려면 RIPE 시각점의 catchment와 사전 배포된 RIPE DNS CHAOS 쿼리(예: A-root의 경우 측정 ID 11309)를 사용합니다. 또는 Verfploeter나 anycast 사이트에서 캡처 된 추적을 사용할 수 있습니다. RIPE 트래픽을 사용하는 장점은 서비스에 새로운 부하를 추가하지 않는다는 것입니다.

중요한 히터는 알려진 좋은 트래픽의 추가 소스를 제공 할 수 있습니다. 많은 서비스는 정기적인 트래픽 패턴을 가진 몇 가지 일관된 대용량 사용자가 있으며 시간이 지나도 많은 사용자는 안정적입니다. DNS의 경우 대부분의 중요한 히터가 정기적으로 시간 변화를 갖는 것으로 나타납니다. 우리는 이러한 알려진 변화를 TBATS로 모델링합니다. [19] 공격자는 중요한 히터를 가장하려면 큰 지속적인 투자가 필요합니다.

추정: 우리의 목표는 제공된 로드, Toffered를 추정하는 것입니다. 우리는 액세스 링크에서 관찰된 트래픽 속도, Tobserved를 측정 할

수 있습니다. 액세스 분수 (α)를 액세스되지 않은 트래픽의 분수로 정의합니다. 따라서 Tobserved = α·Toffered입니다. 액세스 분수 (α)를 추정하기 위해 알려진 좋은 트래픽이 서비스에 도착하는 양이 다른 좋은 트래픽 및 공격 트래픽과 동일한 손실을 가진다는 것을 알 수 있습니다. 알려진 트래픽 속도 (RIPE Atlas 측정 트래픽 또는 중요한 히터 또는 둘 다에서)을 Tknown으로 추정합니다. 그런 다음 α·Tknown, offered = Tknown, observed이며 제공된 로드의 추정치는 T^offered = Tobserved·Tknown, offered/Tknown, observed입니다.

3.4 방어 전략으로서의 트래픽 엔지니어링

제공된 로드를 알고 나면 방어자는 트래픽 엔지니어링 결정을 내릴 수 있는 전반적인 방어 전략을 선택할 수 있습니다. 방어자는 먼저 공격이 전반적인 용량을 초과하는지 여부를 결정해야 합니다.

전반적인 용량을 초과하는 공격에 대해 방어자의 목표는 일부 사이트에서 성공적인 서비스를 유지하는 동시에 다른 사이트가 희생자로서 운영되도록 하는 것입니다. 방어자는 몇몇 희생자 사이트에서 트래픽을 이동하여 그들의 고통을 덜어줄 수도 있습니다. 과부하 된 사이트를 로드가 적은 곳으로부터 해제하는 것은 방파제로 인정됩니다.

중간 규모의 공격에 대해서는 방어자가 모든 트래픽을 제공하려고 노력하고 과부하 된 사이트에서 트래픽을 덜 바쁜 사이트로 이동시키기 위해 리밸런싱을 시도해야 합니다. 이러한 경우에 다양한 anycast 네트워크에서 과부하 된 사이트보다 더 많은 용량을 가진 경우 방어 접근 방식이 다를 수 있습니다. 이러한 경우 큰 "슈퍼" 사이트는 작은 사이트에서 트래픽을 유치할 수 있습니다. 중간 규모의 공격에 대해서는 슈퍼 사이트가 트래픽을 처리할 수 있다면 작은 사이트가 닫히는 것이 가장 좋을 수도 있습니다.

공격 규모와 상관없이 트래픽 엔지니어링은 방어자가 희생자 또는 방파제 사이트로 공격 트래픽을 이동시킬 수 있게 합니다. 다음으로 트래픽 엔지니어링 옵션을 설명하고 응답을 자동화하는 방법을 설명하겠습니다. 전적으로 응답을 자동화하고 싶지 않은 운영자를 위해 우리의 시스템은 가능한 조치 및 그 결과에 대한 권장 사항을 제공 할 수 있습니다.

3.4.1 공격 관리를 위한 트래픽 엔지니어링

전반적인 방어 전략(흡수 또는 리밸런싱)이 주어지면 방어자는 트래픽 엔지니어링을 사용하여 트래픽을 이동시킬 것입니다. 이는 자동으로(§3.4.2) 또는 운영자 감독 하에 조언으로 이루어질 수 있습니다.

공용 인터넷으로 연결된 anycast 배포의 경우 BGP [8]가 라우팅을 제어하고 anycast catchment를 영향을 주는 도구로 선택됩니다. 자체 광대역 네트워크를 운영하는 조직은 내부 WAN에서 트래픽을 관리하기 위해 SDN을 사용할 수도 있습니다. [31, 61]. 다행히도 BGP에는 라우팅 정책을 관리하기 위한 잘 알려진 메커니즘이 있습니다. 우리는 논문에서 세 가지 BGP 메커니즘을 사용합니다: AS-Path prepend, BGP community 및 Path Poisoning.

AS-Path Prepending은 라우팅 경로를 덜 선호하는 방법으로 트래픽을 다른 catchment로 보내는 방법입니다. BGP의 AS-Path는 라우트 원천지로 돌아가는 AS 목록입니다. AS-Path는 라우팅 루프를 방지하고 BGP는 더 짧은 AS-Path를 가진 경로를 선호합니다. 추가 AS를 AS-Path에 인위적으로 삽입함으로써 경로 원천지는 다른 사이트보다 하나를 선호할 수 있습니다. Path prepending은 트래픽 엔지니어링을 위한 고른 라우팅 기술로 알려져 있습니다. 우리는 AS-Path prepending이 anycast에 제공하는 제어의 세분성을 §6.1에서 측정합니다.

우리는 Negative Prepending을 하나의 사이트를 다른 것보다 우선하는 방법으로 AS-Path prepend를 사용한다고 정의합니다. AS-Path prepend는 경로 길이를 증가시킬 수 있지만 모든 anycast 사이트를 제어하는 anycast 운영자는 한 사이트를 제외한 모든 사이트에 사전을 추가하여 사실상 그 사이트를 이전보다 더 짧은 AS-Path로 만듭니다. “S 사이트에서 하나에 의한 부정적인 prepending”은 따라서 S를 제외한 모든 사이트에서 prepending하는 것을 의미합니다.

사전 추가로 인한 긴 AS-Path는 prefix를 경로 탈취에 취약하게 만들 수 있습니다. 그러나 이 문제는 anycast prefix에 작은 영향을 미칩니다. 항상 사전이없는 사이트가 있기 때문에 경로 길이가 제한됩니다. 따라서 우리는 사전이없는 경우에도 경로 탈취에 대한 공식적인 방어가 필요하며 그렇지 않으면 사전이 트래픽 엔지니어링에 대한 보다 가치있는 도구가 될 수 있습니다.

BGP community (또는 community string)는 특정 BGP 경로를 32 또는 64 비트의 정보로 라벨링합니다. 이 정보가 어떻게 해석되는지는 AS에 달려 있습니다. 공식적으로 표준화되지는 않았지만 정보의 일부는 AS를 식별하고 나머지는 블랙홀링, prepending 또는 local preference 설정과 같은 정책을 식별하는데 사용됩니다. community string은 ISP가 라우팅 정책 일부 제어를 고객에게 위임할 수 있도

록 널리 지원됩니다.

Path Poisoning은 들어오는 트래픽을 제어하는 또 다른 방법입니다. 이 기술은 AS 경로에 다른 통신사의 AS를 추가하는 것으로 구성됩니다. AS 경로의 여러 부분에서 AS를 반복하는 경로는 라우팅 루프를 나타내며 BGP에서 제거되어야 합니다.

Path poisoning을 사용할 때 우리는 독한 AS와 자신의 AS를 포함하여 경로를 발표합니다 (그렇지 않으면 이웃이 경로를 필터링하여 우리의 발표를 우리로 부터 받은 것이 아닌 것으로 처리 할 수 있습니다). 따라서 독이되는 것은 다른 모든 anycast 사이트에서도 두 번의 사전을 추가해야 합니다. 그렇지 않으면 독이 됨은 또한 더 긴 AS 경로로 이어집니다.

그림 2는 catchment를 수정하기 위해 anycast 시스템에 트래픽 엔지니어링을 적용하는 방법을 보여줍니다. 이 예에서 사이트-1은 공격으로 인해 압도되었습니다. 여분의 용량이있는 사이트-2로 트래픽의 bin을 이동시키려고하는 경우 BGP 발표를 할 수 있습니다. 사이트-1은 AS3를 독합니다, 사전을 추가하고 (AS4에만 표시), AS5에는 not-export BGP community를 사용하여 발표하지 않습니다. 이러한 변경 사항은 사이트-1의 로드를 줄이고 트래픽을 사이트-2로 이동시킵니다.

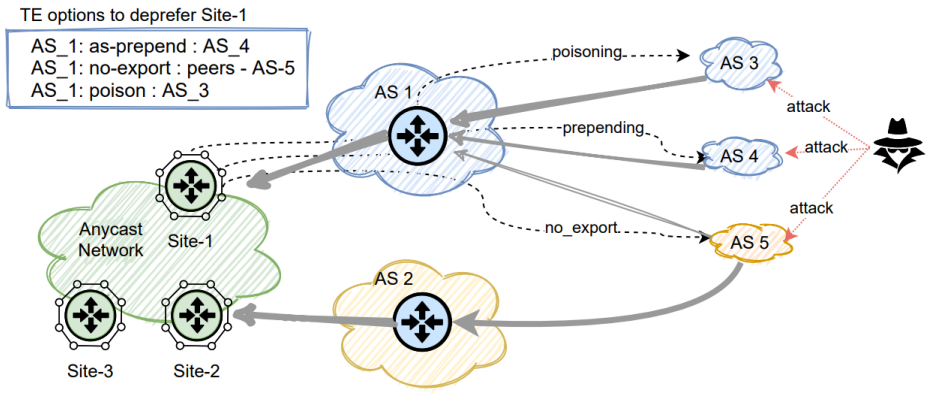


Figure 2: TE techniques to shift traffic from Site-1 to Site-2.

3.4.2 자동 방어 선택

방어를 자동화하기 위해 우리는 중앙 집중식 컨트롤러를 사용합니다. 컨트롤러는 모든 사이트의 관측치를 수집합니다 (외부 측정 또는 사이트에 도달할 수 없는 경우 사이트가 포화되었다고 가정), 그런 다음 필요한 경우 조치를 취합니다 (도 1):

(1) 컨트롤러는 각 사이트에서 예상되는 용량 및 관측된 리소스를 추정된 로드와 비교하여 과용량 사이트를 식별합니다. (2) 컨트롤러는 현재 허용 가능한 사이트를 과부하로 만들지 않고 영향받는 모든 사이트의 부하를 줄일 수 있는 모든 Playbook 옵션을 식별합니다. (3) 모든 실행 가능한 옵션에서 균일 한 분포와 가장 작은 변화를 선호하며 (필요한 경우 임의로 선택) 선택합니다. 모든 변경 사항이 어떤 사이트도 압도하지 못하게 남겨 둔 경우, "가장 나쁜" 시나리오를 선택하거나 운영자 개입을 요청할 수 있습니다.

새로운 라우팅 정책을 배포 한 후에도 결정 기계는 각 사이트의 트래픽 수준을 계속 평가합니다 (도 5). 5 분 후에도 여전히 과부하되는 사이트가 있는 경우 크기 추정, 결정 및 조치를 반복하여 다시 시도합니다. 다음 반복에서 컨트롤러는 이전 반복 (이 결정 프로세스의 단계 (2)에서)에서 고려된 라우팅 옵션 만을 고려합니다. 알림이 전파 될 수 있도록 시도 사이에 시간을 허용하기 위해 시간을 허용합니다. 진동을 피하기 위해 라우트 플랩 억제와 간섭을 피하기 위해 3 회 시도 후 문제를 인간 운영자에게 승격시킵니다. 이 타이머 기간 및 재시도 횟수에 대한 이 값은 진동을 피하기 위한 운영자의 권장 사항에서 선택하며 다른 옵션도 가능합니다. 다음 작업을 탐색할 수 있습니다.

서비스 복귀 : 과부하되는 사이트가없는 기간이 지난 후에는 기본 라우팅이 사용자에게 최상의 서비스를 제공한다는 가정 하에 개입을 자동으로 되돌릴 수 있습니다. 일정 시간 동안 개입을 유지하는 것은 다형적 공격을 처리하는 데 도움이 될 수 있습니다 (§8).

3.4.3 운영자 지원 시스템

우리는 루트 DNS 및 클라우드 서비스의 운영자와 접근 방법에 대한 피드백을 얻기 위해 접근 방법을 논의했습니다. 일반적인 DDoS 이벤

트를 처리하고 비 업무 시간 동안 이벤트를 처리하는 자동 방어에 열광 할지라도 일부 운영자는 인간 감독 (비자동화) 응답을 선호하며 처음 배포 시 운영자 개입을 예상하고 신뢰를 구축합니다.

인간 감독 응답을 지원하기 위해 우리는 자동화의 대안 (또는 전조)으로 운영자 지원 시스템을 설계합니다. 이 시스템은 현재 센서 상태에 기반하여 좋은 옵션을 추천하는 Playbook 조회와 결합 된 웹 기반 인터페이스를 제공합니다.(그림 3). 자세한 내용은 부록 B에 설명되어 있습니다.

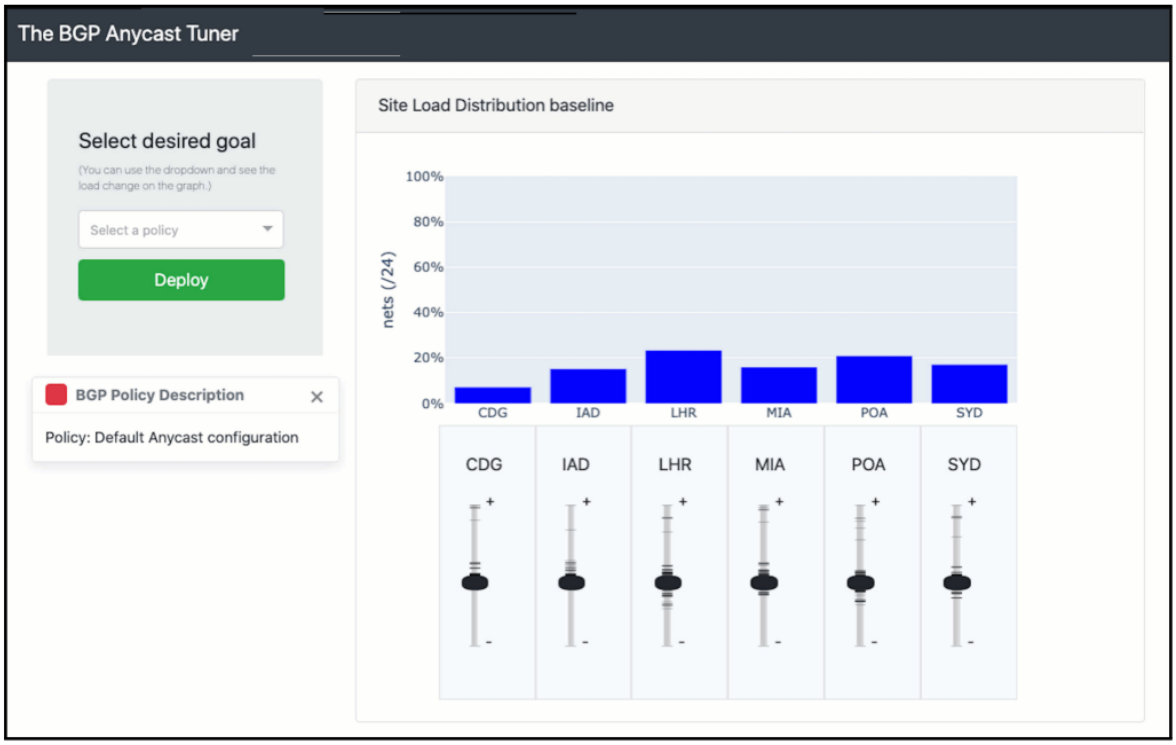


Figure 3: Operator assistance system.

그림 3: 운전자 지원 시스템.

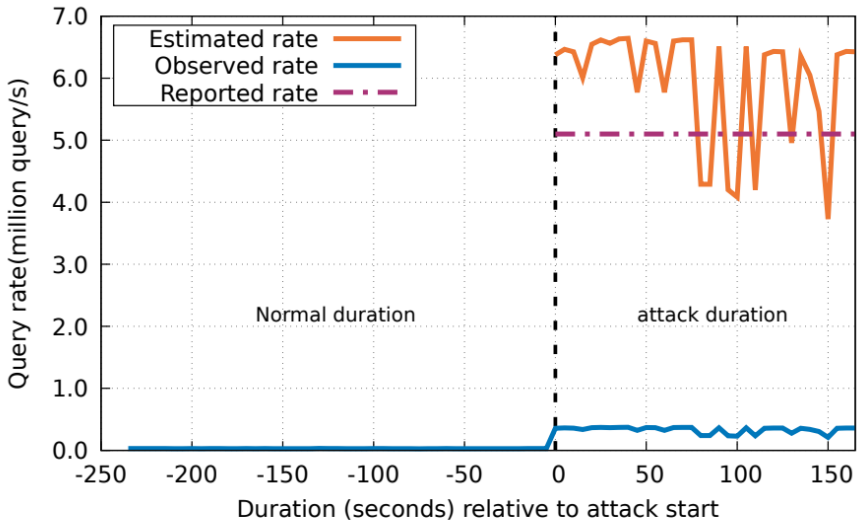


Figure 4: Estimating real-world attack events: estimating Nov. 2015 event with 5.59% access fraction.

그림 4: 실제 공격 이벤트 추정: 2015년 11월 이벤트를 5.59%의 액세스 비율로 추정합니다.

4. 제공된 부하 추정의 평가

다음으로 실제 이벤트를 사용하여 제공된 부하를 추정하는 것을 평가합니다. 시스템 평가는 부록 C에 있습니다.

4.1 사례 연구

우리는 2015년 11월 30일과 2016년 6월 25일의 두 거대한 DNS DDoS 이벤트를 사용하여 접근 방식을 테스트합니다. 2015년 11월 이벤트는 DNS 플러드였으며, 2016년 6월 이벤트는 SYN 및 ICMP 플러드 공격이었습니다. B-root는 이러한 이벤트 모두에서 상당한 상위 전송 손실을 나타냈으므로 B-root에 대한 실제 제공된 부하를 추정하고 다른 루트에서의 관측치와 비교합니다.

우리의 시스템을 적용하기 위해 우리는 알려진 좋은 트래픽을 사용하여 액세스 분율 (α)을 측정합니다. 표 1은 기대되는 일반적인 알려진 좋은 트래픽 ("정상"), 공격 시 관측된 속도 ("관측") 및 계산된 α를 보여줍니다. 여기서 우리는 알려진 좋은 트래픽으로 RIPE Atlas를 사용합니다. 우리는 상위 100 개의 무거운 사용자를 사용할 때도 유사한 결과를 볼 수 있습니다 (공간이 부족하여 생략됨).

그림 4는 관측 된 부하 (아래 파란색 선)를 비교하여 시스템에서 추정 된 제공된 부하 (중간, 변동, 주황색 선)를 다른 루트에서보고 된 공격 속도 (점선 보라색 선)와 비교합니다. 표 1의 제공된 부하 열은 숫자 값을 제공합니다.

공격이 크더라도, 2015 이벤트의 추정 공격 크기가 4-6.5 Mq/s로보고 된 5.1 Mq/s [45,47]와 거의 비슷하다는 것을 알 수 있습니다. 우리는 또한 2016 이벤트 [48]에서도 유사한 결과를 볼 수 있으며, 여기서 총 트래픽의 8-11 Mq/s를 추정하고 보고 된 10 Mq/s 속도와 비교합니다 (그림의 자세한 내용은 §C.2). Testbed 실험 결과를 추가하면 좋은 정확도를 보입니다 (§C.1의 세부 정보).

이 두 이벤트는 상위 전송 손실률이 높더라도 총 제공된 부하를 합리적으로 추정할 수 있다는 것을 보여줍니다. 우리의 결과는 RIPE와 같은 알려진 좋은 트래픽이 2500 개의 쿼리/분을 가질 때 좋은 정확도를 제공하며 추가로 알려진 좋은 트래픽을 사용하면 정확도가 향상됩니다. 그러나 실제로는 대략적인 추정만으로도 직접 관측된 부하를 사용하는 것보다 훨씬 나은 응답이 가능합니다.

우리는 공격 크기 추정이 DDoS 이벤트에 대한 대응 계획을 수립하는 데 충분히 가깝다고 결론짓습니다.

Scenario/ Date	Dur.	known-good traffic normal	observed	α	offered load during attack normal	observed	reported	estimated	α̂	estimated/ reported
2015-11-30	3h	33.08	1.85	0.0559	0.03 M	0.37 M	5.1 M	6.6 M	0.07	1.3
2016-06-25	3h	36.58	0.33	0.0091	0.03 M	0.10 M	10 M	11 M	0.01	1.1
Testbed	5min	425.2	207.0	0.4900	8.5 k	16.3 k	29.2 k	33.2 k	0.56	1.1

Table 1: Estimating sizes of offered load (second from right) based on known-good traffic (second from left) with real-world attacks at `B-root` and testbed experiment. Traffic rates are in queries/second (reporting only the peaks).

표 1: B-루트 및 testbed 실험에서 실제 공격을 통해 알려진 정상 트래픽(왼쪽에서 두 번째)을 기반으로 제공되는 로드 크기(오른쪽에서 두 번째)를 추정합니다. 트래픽 속도는 쿼리/초 단위입니다(최고치만 보고).

Testbed	Used Sites	#
`Peering`	Amsterdam*†(AMS), Boston* (BOS), Belo Horizonte*†(CNF), Seattle* (SEA) Athens* (ATH), Atlanta* (ATL), Salt Lake City* (SLC), Wisconsin* (MSN)	8
`Tangled`	Miami (MIA)*, London (LHR)*, Sydney (SYD)*, Paris (CDG)*, Los Angeles (LAX)*, Enschede (ENS)*, Washington (IAD)*, Porto Alegre (POA)*†	8

Table 2: Testbed and respective sites used in our experiments. Transit providers (*) and IXP (†).

표 2: 실험에 사용된 testbed 및 해당 사이트. 대중교통 제공업체(*) 및 IXP(†).

Experiment	Key Takeaways
Path prepending	Works everywhere to effectively de-prefer a site (§6.1.2), but shifts traffic in large amounts (§6.1.3), and has few traffic levels (Figure 6).
Neg. Prepending	Works everywhere to prefer a site (§6.1.2).
BGP communities	Although widely implemented, well-known communities are not universal (§6.2.1).
	When supported, they provide finer-granularity control than prepending (§6.2.2).
BGP path poisoning	Many Tier-1 ASes drop the announcements when it sees Tier-1 ASes in the paths. (§6.3.1)
	Control over traffic is limited by the filters from other ASes. (§6.3.2).

Table 3: Experiment summarization and findings.

표 3: 실험 요약 및 결과.

5. 평가 방법

다음으로 우리는 TE(§6)의 효과를 평가하고 그 결과가 다양한 배포에 일반화되는지를 설명할 것이다. DDoS에 대응한 트래픽 엔지니어링은 어디에 사이트가 있고 누구와 연결되어 있는지에 따라 다르다. 우리는 두 가지 다른 testbed에서 평가한다. 우리의 접근법(추정, TE 및 Playbook 구성)은 다른 anycast 설정과 함께 어디에서나 적용될 수 있다. 우리는 네트워크 운영자들이 이벤트 전에 우리의 방법을 운영 네트워크와 병렬로 테스트 prefix에서 실행할 것으로 기대한다.

5.1 Anycast testbed

실제 peering 및 배포의 제약 사항을 확인하기 위해 testbed에서 아이디어를 평가한다. 우리는 두 개의 독립적인 testbed인 Peering 및 Tangled를 사용한다. 테이블 2는 각 testbed에 대한 정보를 요약하고 그들의 위치와 함께 지리적으로 분산된 사이트의 고유한 집합을 보여준다(Peering은 더 많은 사이트를 지원하지만 우리는 8개 사이트를 사용했다). 이러한 사이트들은 서로 다른 연결성을 보이며 하나 이상의 transit 및 IXP peer를 가지고 있다. 대부분의 Peering 사이트는 학술적인 transit을 가지고 있지만 Tangled에는 더 많은 상업적인 제공업체가 있다. 우리의 testbed는 거의 절반 정도의 실제 네트워크와 크기가 같다(실제 세계의 네트워크의 거의 절반은 5개 이하의 사이트를 가지고 있다).

5.2 경로 변경 측정

BGP 변경의 효과를 측정하기 위해 우리는 우선 사이트에서 라우팅 알림을 변경하고, 전파되는 데 시간을 주고, 알림이 수락되었는지 확인한 후에 마지막으로 anycast 측정을 시작한다.

- 경로 수렴: 변경 후에 BGP 경로 전파에 시간을 허용한다. 라우팅 및 전달 테이블은 prefix가 업데이트됨에 따라 일관성이 없을 수 있다(루프 또는 블랙홀 발생). 라우팅 업데이트는 보통 5분 이내에 안정되지만 우리의 Playbook을 작성할 때 라우팅이 안정되도록 15분을 기다린다. 공격이 라우팅 정책을 배포한 후에도 완화되지 않으면 5분 후에 다른 접근법으로 이동한다.

- BGP 정책 전파: 정책 필터링은 알려진 경로의 수락을 제한할 수 있지만 실제로 이러한 제한은 우리의 트래픽 엔지니어링에 영향을 미치지 않는다. 가장 좋은 실천 방법은 가장자리 네트워크가 10개 이상의 AS-Path를 거친 경로를 필터링하고 중간 AS는 종종 50개 이상의 AS-Path를 수락한다. 이는 우리가 필요로 하는 준비보다 더 많은 prepend를 나타낸다. RIPE RIS를 사용하여 여러 전역 위치에서의 라우팅 관측을 기반으로 우리의 실험 구성이 AS-Path 길이 필터링으로 인해 다중 항해 지점에서 차단되지 않음을 확인한다.

6. 트래픽 엔지니어링 범위와 제어

공격 부하 추정을 통해 운영자는 BGP를 사용하여 트래픽을 이동시킨다. 우리는 다음으로 세 가지 TE 메커니즘을 평가한다: AS-Path prepending, community string 및 path poisoning. 각각이 언제 작동하고 어느 정도의 제어를 제공하는지 고려한다. testbed(§5.1)에서의 테스트에서의 주요 결과는 표 3에 요약되어 있다. §7에서 일반화를 평가한다.

6.1 경로 prepending으로 제어

먼저 방어 전략으로 AS-Path prepending을 고려한다.

6.1.1 prepending coverage

AS-Path prepending의 지원은 매우 완벽하다 - 상위 제공업체로부터의 명시적 지원이 필요하지 않으므로 우리는 두 testbed 모두에서 모든 사이트에서 prepending이 작동하는 것으로 발견했다. Peering에서는 최대 세 개의 prepending을 사용할 수 있으며 Tangled에서는 최대 다섯 개의 prepending을 사용한다. 이전 연구 [14]에 따르면 5개의 prepend를 사용하는 것이 충분하다고 보여진다. 왜냐하면 활성 AS의 90%가 6개 이하의 AS 항해 지점에 있기 때문이다. 우리는 RIPE RIS를 사용하여 prepend가 적용된 경우 라우팅 가시성을 확인하고 두 testbed에서도 라우팅 전파에 변경 사항이 없다는 것을 확인한다. 그렇지 않으면 이것은 AS 경로 길이 필터링의 존재를 드러낼 수 있다.

6.1.2 prepending이 작동하는가?

AS-Path prepending이 널리 지원되므로 다음으로 이 효과적인 TE 방법을 평가한다.

prepending이 작동하는지 확인하기 위해 유럽(Amsterdam-AMS), 북미(Boston-BOS) 및 남미(Brazil-CNF)의 세 대륙에서 세 사이트를 사용하여 대표적인 시나리오를 사용하여 이 질문을 탐색한다. 다른 구성에 대한 일반화를 §7에서 검토한다. catchment에서 /24 블록을 계산하여 기준을 TE 옵션과 비교한다. (상수가 다른 경우에도 동일한 질적 결과와 모양을 얻기 위해 블록 대신 트래픽을 가중치가 적용된 트래픽으로 탐색하였다, 추가 정보 참조 추가 정보 참조).

이러한 조건에서 각 사이트의 트래픽을 보여주는 그림 5를 확인한다. 각 그래프의 중간 막대는 prepending이 없는 기본 조건인 기준이다. 그런 다음 각 사이트에 prepending을 추가하고, 각 막대의 오른쪽으로 이동하면서 각 사이트에 1, 2 또는 3개의 prepending을 추가한다. 또한 음수 prepending(§3.4.1)을 고려하여 각 막대가 중앙에서 왼쪽으로 이동한다. 그림 5의 모든 세 개의 그래프의 기준(중간 막대)을 먼저 고려한다. 각 막대의 아래쪽 맨 아래 부분에 있는 Amsterdam(AMS)은 트래픽의 약 68%를 받는다. AMS는 두 개의 transit 제공자와 여러 peer를 가지고 있으며 Amsterdam이 전 세계와 매우 잘 연결되어 있기 때문에 BOS 및 CNF보다 더 많은 트래픽을 받는다.

다음으로 각 사이트에서 prepending을 고려한다(중앙에서 오른쪽으로 이동하는 막대). 각 경우에 prepending은 예상대로 사이트에서 트래픽을 이동시킨다. AMS의 경우, 각 prepending은 더 많은 트래픽을 이동시키며, 첫 번째 prepending은 트래픽을 68%에서 37%로 줄이고, 다음은 29%로, 다음은 약 16%로 줄인다. BOS와 CNF는 더 적은 트래픽으로 시작하며 prepending이 더 강한 효과를 보인다. 하나의 prepending이 가장 많은 트래픽을 전송하며(예를 들어, BOS의 경우, 15%에서 7%로), 추가 prepending은 거의 추가 변화를 보여주지 않는다. 이러한 비선형 변화는 prepending을 사용하여 BGP 라우팅이 경로 길이에 따라 변경되기 때문이다. 인터넷의 AS 그래프는 비교적 평평하다.

막대 그래프는 사이트에서 트래픽을 이동할 때 이를 전부 다른 사이트로 보낸다는 것을 보여준다. 이는 라우팅에 따라 달라지며 다른 구성에서의 분할과 비례하지 않을 수 있다. 예를 들어, AMS로의 prepending 한 번 후에, 더 많은 트래픽이 BOS로 가는 대신 CNF로 가게 된다. 이러한 예상치 못한 이동은 공격(§3.2)이 발생할 때 결정을 안내하고 변경의 결과를 예측하기 위해 라우팅 옵션 "Playbook"을 사전에 계산하는 것을 제안하는 이유이다.

또한 음수 prepending이 사이트로 트래픽을 끌어들이는데 성공한다는 것을 볼 수 있다 – 각 경우에 다른 사이트가 prepending되지 않은 사이트에 트래픽이 더 많이 흐른다. AMS는 상대적으로 적은 변화를 보인다(68%에서 89%까지) – 이미 대부분의 트래픽이 있기 때문에, BOS 및 CNF는 각각 최대 68%의 트래픽을 얻는다.

모든 세 사이트는 prepending과 관계없이 해당 사이트에 '붙어있는' 일부 네트워크를 보여준다. 이러한 점유는 해당 교환의 하류에 위치하여 하나의 사이트를 통해만 라우팅될 수 있는 네트워크가 있는 경우에 발생한다. BOS에 붙어있는 두 개의 임의로 선택된 블록에 대한 traceroute를 수행하여 이를 확인한다. Traceroute 및 지리적 위치 확인(Maxmind)을 사용하여 이들이 보스턴의 MIT 및 콤캐스트 네트워크에 있음을 확인한다(끝에서 두 번째 traceroute 홉을 기준으로). 이러한 붙음을 해소하기 위해 local 우선 BGP 속성을 사용했지만, 이러한 옵션의 체계적인 탐색은 미래의 작업이다.

요약하면, 실험은 AS prepending이 작동하고 사이트 간의 트래픽을 이동시킬 수 있음을 보여주지만, 이 트래픽 이동은 균일하지 않다는 것을 보여준다.

6.1.3 prepending이 제공하는 세부 사항

prepending이 트래픽을 이동시킬 수 있음을 입증한 후에는 이것이 얼마나 많은 제어를 제공하는지를 다음에 물어본다. 이 질문에는 사이트로부터 얼마나 많은 트래픽을 밀어내거나 끌어들일 수 있는지, 그리고 최소와 최대 사이에 얼마나 많은 다른 수준이 있는지 두 가지 측면이 있다.

제한: 그림 5는 그 세 사이트와 함께 Peering에서 트래픽이 이동할 수 있는 한계를 보여준다. prepending에 관계없이 AMS, BOS 및 CNF는 항상 약 16%, 7% 및 3%의 블록을 받는다.

그림 6은 우리의 다른 testbed(Tangled)에 5개의 사이트(유럽 두 곳, 북아메리카 한 곳, 남아메리카 한 곳, 오스트레일리아 한 곳)를 배치하여이 결과를 확인한다. X축은 각 사이트에 적용된 prepend의 수로 표시된다. 숫자 0은 기준을 나타내며, 양수 숫자(1-5)는 적용된 prepending의 수이며, 음수 숫자는 음수 prepending을 나타낸다. 그림에서 볼 수 있듯이, 각 사이트는 최대 55-65%의 블록을 포착할

수 있으며 최대 95%의 블록을 떨어뜨릴 수 있다. 따라서 최소와 최대 사이에 세 가지 점만 있으므로 세부적인 제어를 얻지 못한다.

우리는 prepending이 트래픽을 이동시키는 데 유용한 도구 일 수 있지만, 상대적으로 제한된 제어를 제공한다고 결론지었다.

6.2 BGP community를 사용한 제어

다음으로 BGP community string이 가지는 반대의 트레이드오프를 보여준다: 각 사이트에서 지원되는 옵션은 다르지만, 사용 가능할 때 트래픽에 대한 더 세부적인 제어를 제공한다. 각 사이트에서 지원할 수 있는 community string을 사용한다. 동일한 개념에 대한 구체적인 값은 종종 다르다.

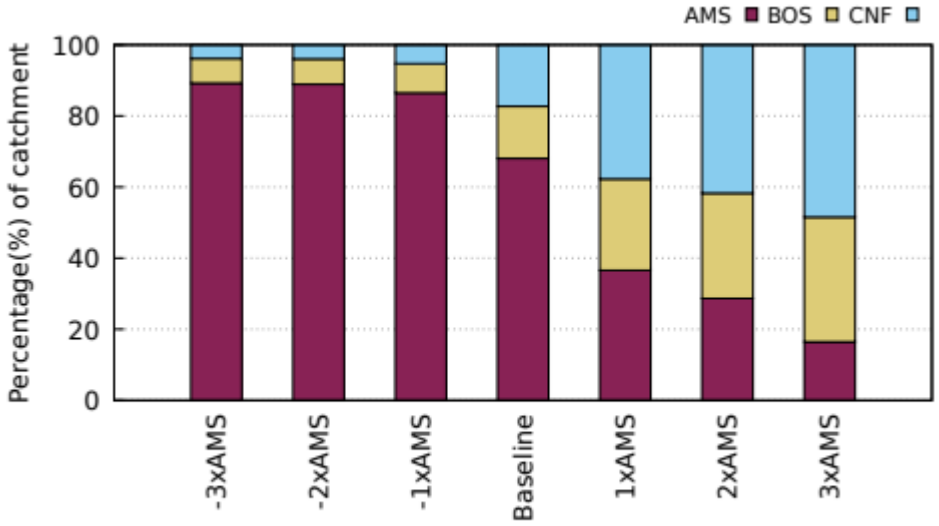
6.2.1 community string coverage

AS(Autonomous System)들은 루프 탐지를 위해 AS 경로가 사용되기 때문에 prepend의 대부분의 지원과 달리 community string을 이웃들과 교환하기 위해 선택적으로 참여해야 합니다. (AS 경로가 루프 탐지에 사용되기 때문에, 명시적으로 필터링되지 않는 한 prepend는 작동합니다). 명시적인 지원이 필요한 이유는 community가 태깅 메커니즘일 뿐이며, 그들이 트리거하는 액션은 peering AS의 재량에 따라 달라지기 때문입니다. 이전 연구에서 community string이 지원하는 다양한 옵션들을 연구해왔습니다. coverage를 평가하기 위해, 우리가 사용하는 testbed에서 BGP community에 대한 지원을 검토합니다. testbed는 다양한 peer를 가진 20여 개의 위치에 대한 정보를 제공합니다. 이 중 각 peer에 대해 이 기능을 지원하는지를 평가했습니다. 표 4에서는 각 사이트에서 지원되는 경로 prepend 및 poisoning 지원 및 어떤 유형의 community string이 지원되는지를 설명합니다. 우리는 광고 옵션 (peer 없음, 고객에게 노출하지 않음, 누구에게도 노출하지 않음), 선택적인 prepend 및 선택적 광고를 지원하는 peer 및 transit을 클래스별로 그룹화합니다. 또한 비 transit peer 및 transit의 수도 표시합니다. peering을 통해 각 사이트의 transit 및 peer에 대한 선택적 발표를 허용하지만, peer 및 transit의 수는 다양합니다. 하나의 transit을 제공하는 많은 사이트는 대안을 제공하지 않습니다. 우리는 AMS에서의 선택적 발표 옵션을 고려했으며, 854개의 peer(2개의 라우트 서버와 748개의 peer를 포함한 106개의 양자 peer) 및 2개의 transit 공급자 [60]가 있습니다. CNF는 한 개의 transit 공급자와 129개의 peer (양자 peer는 6개뿐이며, 다른 peer는 2개의 라우트 서버를 통해 연결됩니다)를 가지고 있습니다. 우리의 Verfploeter 측정을 위해 이중 BGP 세션을 가진 peer와 라우트 서버를 고려합니다. 단일 peer는 Verfploeter 측정에서 주소 공간의 작은 부분을 커버합니다. 일부 peer의 경우 우리가 관측한 coverage가 전혀 없었으며, 이는 우리의 관측을 확인하기 위해 peer에 대한 추가 조사가 필요합니다. 따라서 모든 선택적 발표 옵션은 catchment 분포에 차이를 만들지 않습니다 (12개의 peer로 AMS의 catchment와 Figure 7의 transit-1을 비교하십시오). 표 4의 옵션 열은 이러한 결과를 요약하여 community string을 사용하여 얼마나 많은 라우팅 옵션이 있는지를 보여줍니다. 우리는 Tangled를 두 번째 배포로 평가하여 다른 peer를 제공합니다. Tangled는 클라우드 제공 업체, 크라우드 소싱 transit 제공 업체 및 IXPs(Internet Exchange Point) 위에 anycast 네트워크를 구축했습니다. 모든 transit 제공 업체 및 IXP 사이트에서 표 4에 설명된 대로 community를 지원합니다. Tangled에서, POA 사이트에는 250개의 peer가 있으며 대부분이 community string을 지원합니다. 각 anycast 사이트의 옵션 수는 peer 및 transit과의 연결 수에 따라 다를 수 있습니다. 이 불확실성은 가능한 옵션을 보여주는 Playbook의 필요성을 보여줍니다.

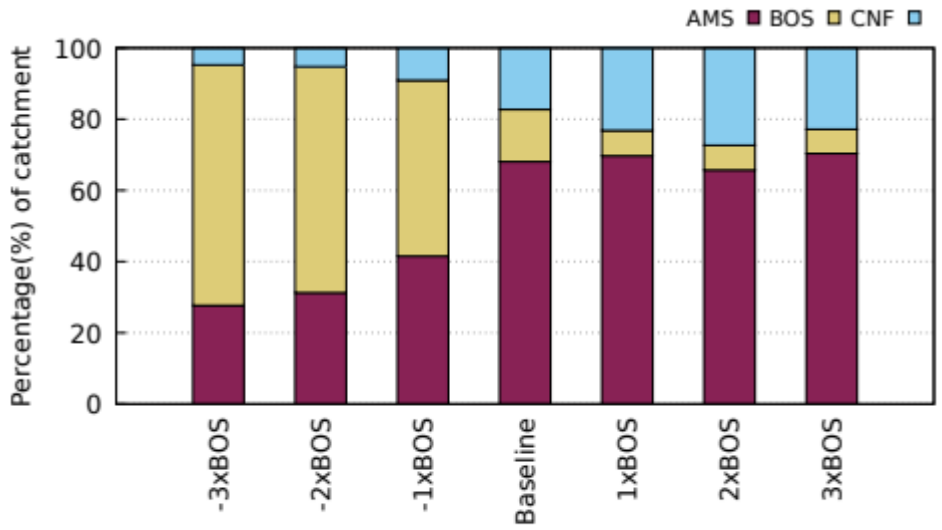
6.2.2 community string이 어떤 세밀한 수준에서 작동하는가?

다음으로 community string이 어떻게 작동하고 어떤 세밀한 제어 수준을 제공하는지 살펴보겠습니다. 우리는 community string을 사용하여 BGP 선택적 발표를 만들어, 우리의 라우트를 특정 transit 공급자 또는 IXP peer에게만 전파합니다. 우리의 실험에서, 우리는 Peering을 사용하여 AMS에서 발표를 변경하고, AMS, BOS 및 CNF에서 anycast가 제공될 때의 트래픽을 관찰합니다(§6.1.2와 동일한 토폴로지). §6.2.1에서 설명한대로 선택적 발표 community string은 AMS 및 CNF에서만 제공되며, AMS에서 여러 peer, 2개의 transit 중 하나, 및 라우트 서버에 영향을 줍니다. 선택적 발표를 위해 대상 AS를 선택하기 위해, AMS 사이트의 모든 작동 peer를 CAIDA의 AS 순위 목록 [9]을 사용하여 고객 콘의 크기를 기준으로 정렬합니다. 그런 다음 6개의 가장 큰 IXP peer와 그 다음 12개를 선택합니다(그림 7의 왼쪽 두 막대). 그런 다음 라우트 서버를 따로 발표하고(다음 막대), 그런 다음 라우트 서버를 포함한 모든 IXP peer를 살펴봅니다. 마지막으로, 각각의 두 transit 공급자에 대한 coverage를 개별적으로 발표합니다. 첫째, AMS가 기준선인 68%의 블록에서 다른 구성으로 53%에서 6%의 블록으로 이동함을 볼 수 있습니다. 둘째, 어떤 조합에서 겹치는 부분이 있음을 볼 수 있습니다. 예를 들어, 각 transit이 AMS에서 도달 가능한 모든 블록의 절반 이상을 도달하므로, AMS에서 도달 가능한 일부 블록이 두 transit 공급 업체에서 모두 도달 가능함을 알 수 있습니다. 따라서 AMS로 라우팅할 블록 수에 대한 어느 정도의 제어가 있지만, 일부 peer는 매우 "강력"하며 prefix를 발표할 경우 많은 블록을 획득할 것입니다. 셋째, 라우트 서버의 중요한 역할을 볼 수 있습니다. 직접 12개의 IXP peer와의 조정은 AMS에서 7%의 블록만 가져옵니다. 그러나 라우트 서버는 AMS가 더 많은 AS에 도달하고 14%의 블록을 단독으로 가져옵니다. 마지막으로, transit 공급 업체가 중요한 역할을 하는 것을 볼 수 있습니다. AMS 사이트에는 두 개의 transit 공급자인 BIT BV (AS12859)와

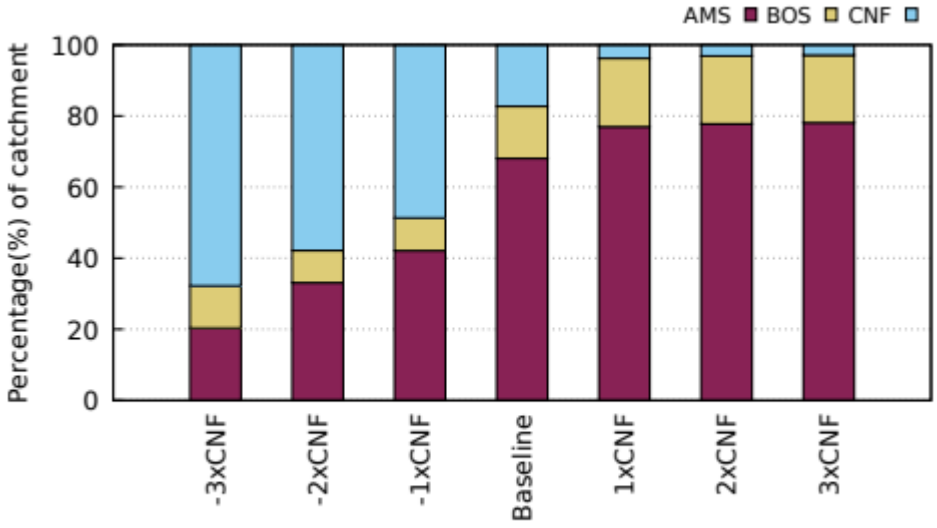
Netwerkvereniging Coloclue (AS8283)가 있습니다. AS8283로 발표하면 AS12859로 발표하는 것보다 AMS로의 트래픽이 더 많이 유도됩니다. 이 두 transit의 AS 관계가 상위 업스트림과 다른 트래픽 분배를 제공합니다. 우리의 실험에서 보여주듯이, AS 경로 prepend와 비교하여 BGP community는 트래픽 분배를 훨씬 더 잘 제어할 수 있습니다. Peering에서 찾은 결과를 일반화할 수 있는지 조사하기 위해 Tangled에서 일련의 실험을 수행했습니다. Peering와 같이, 세 개의 대륙에서 세 사이트를 선택하고(London(LHR), Miami (MIA), Porto Alegre (POA)), LHR에서 선택적 prepend 및 선택적 발표를 위해 community를 사용합니다. 그림 8에서 LHR의 community string을 사용한 후의 catchment 분포를 보여줍니다. 기준선에서 community를 사용하지 않은 경우, LHR은 트래픽의 69%를 처리합니다. 오른쪽에서 왼쪽으로, catchment 분포가 69%에서 33%까지 점진적으로 감소하는 것을 볼 수 있습니다. IXP peer에게 발표 중지는 트래픽을 69%에서 64%로 줄입니다. 그러나 AS2914 (NTT America), AS1299 (Telia Company) 및 AS3356 (Level 3)에서의 prepend 및 no export community를 사용하면 LHR의 catchment에서 30–60%를 얻을 수 있습니다. 두 testbed 모두 모든 사이트에서 community string이 널리 사용되지 않으며, 심지어 잘 알려진 community도 완전히 채택되지 않는다는 것을 보여줍니다. 그러나 community string은 더 세세한 제어를 제공할 수 있습니다. 선택적 발표는 대부분 IXP peer 및 transit이 연결된 수에 따라 더 많은 "유연성"을 제공한다는 것을 발견합니다. 또한 일부 사이트가 예상했던 지원을 제공하지 않는다는 것을 발견했는데, 이는 community string이 명시적인 동의를 위해 transit 공급 업체에 연락할 필요가 있다는 것을 의미합니다.



(a) AMS site.

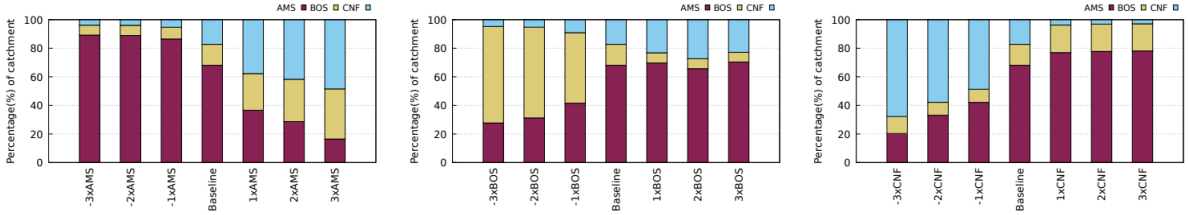


(b) BOS site.



(c) CNF site.

그림 5: peering: 2020-02-24에 AMS, BOS 및 CNF 사이트를 통한 유역 분포에 추가되는 경로의 영향.



(a) AMS site. (b) BOS site. (c) CNF site.

Figure 5: `Peering`: Impact of path prepending in catchment distribution with AMS, BOS and CNF sites on 2020-02-24.

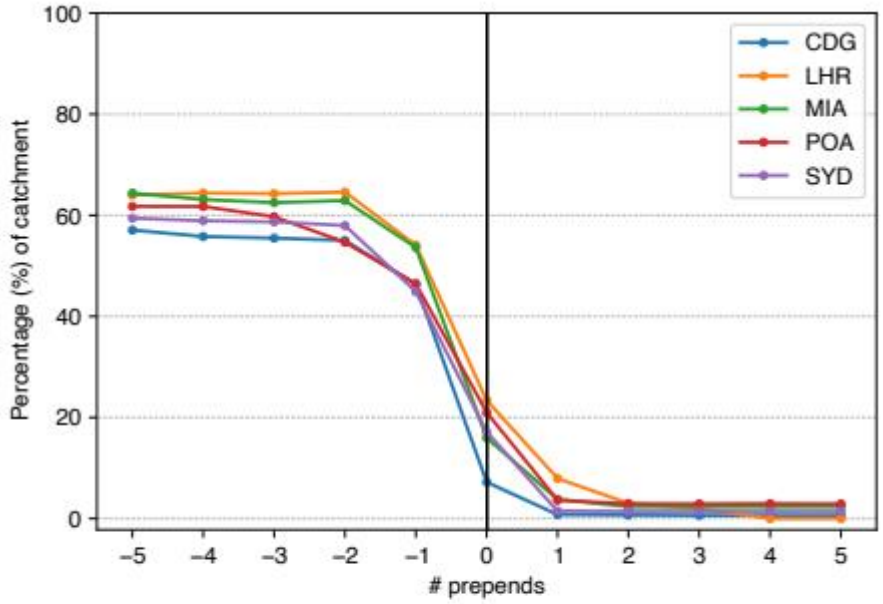


Figure 6: Tangled: Effect of path prepending on catchments.

그림 6: 얽힌: 유역에 추가된 경로의 효과.

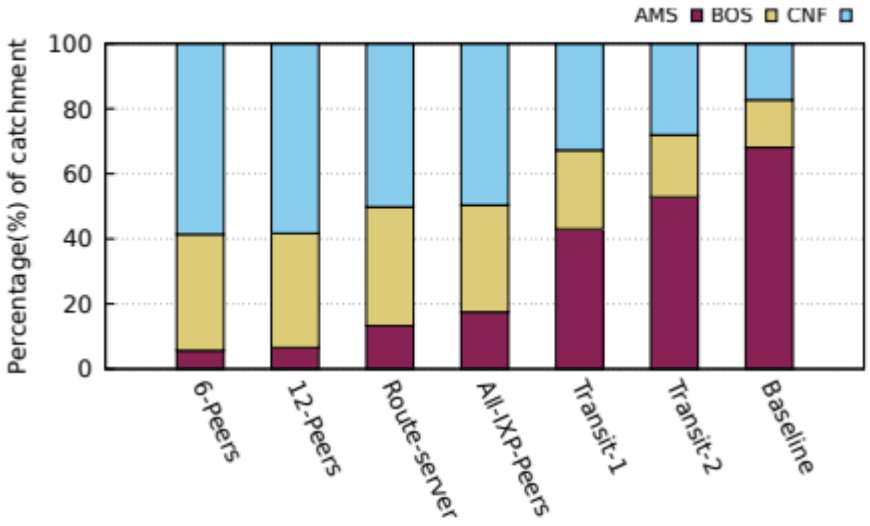


Figure 7: Peering: Community strings (at AMS) on catchments for AMS, BOS, CNF on 2020-02-25.

그림 7: peering: 2020-02-25 AMS, BOS, CNF 유역에 대한 community string(AMS).

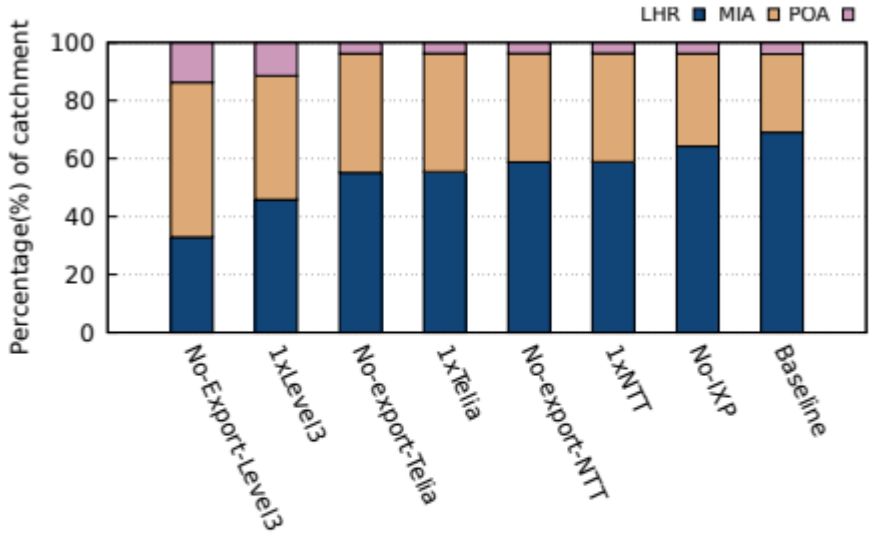


Figure 8: Tangled: using different communities to shift traffic on site LHR on 2020-04-05.

그림 8: 얽힌: 2020-04-05에 LHR 사이트의 트래픽을 이동하기 위해 다양한 community를 사용합니다.

Site:	Peering								Tangled							
Routing policy	AMS	BOS	CNF	SEA	ATH	ATL	SLC	MSN	MIA	LHR	IAD	CDG	LAX	ENS	SYD	POA
AS-path prepend	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
no-peer	✓	–	✓	–	–	–	–	–	✓	✓	–	✓	–	–	✓	✓
no-export	△	–	–	–	△	–	–	–	✓	✓	–	✓	–	–	✓	✓
no-client	–	–	–	–	–	–	–	–	✓	–	–	–	–	–	–	–
Selective prepend	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	–	–	✓	✓
Selective announcement	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	-	✓	✓
Path poisoning	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	–	–	–	–	–	✓
# non-transit peers	854	0	129	0	0	0	0	0	0	0	0	0	0	0	0	250
# transits	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
# options	856	1	130	1	1	1	1	1	1	1	1	1	1	1	1	252

Table 4: Traffic engineering options on each testbed sites. ✓: supported, -: not supported, △: not tested.

표 4: 각 testbed 사이트의 트래픽 엔지니어링 옵션. X: 지원됨, –: 지원되지 않음, M: 테스트되지 않음

6.3 경로 독립성

이제 경로 독립성에 대해 살펴보고, community string과 마찬가지로 coverage와 세분성이 상류 peer에 배포된 라우팅 필터로 제한됨을 보여줍니다.

6.3.1 독립성 coverage

경로 독립성의 지원은 우리가 독립하는 AS와 우리의 상류 AS들이 배포한 경로 필터에 따라 달라집니다. 많은 ISP들, 특히 Tier-1 AS들이 다른 Tier-1 AS를 포함하는 경로에서 독립성을 독립하는 AS 경로를 필터링합니다. Tier-1 AS들은 라우트 누출을 방지하기 위해 경로에

다른 Tier-1 AS를 포함하는 고객의 BGP 발표를 차단하기 위해 이러한 필터를 배치합니다. 이 필터링은 종종 경로 독립을 통제하는 데 효과적이지 못하게 만듭니다.

Tier-1 AS를 독립하는 것이 필터링으로 인해 종종 효과적이지 않다는 것을 확인하기 위해, Peering에서 AMS에서만 발표하는 Tier-1 AS를 독립시키고, 다른 사이트의 영향을 차단하는 유니캐스트 설정을 사용하여 1000개의 RIPE 관측 지점에서 prefix로부터의 트레이스 루트를 생성합니다. 우리의 측정 결과는 Tier-1 AS를 독립할 때 필터의 증거를 보여줍니다 - AS7018 (AT&T), AS6453 (Tata Communications America) 및 AS1299 (Telia Company). 우리는 많은 관측 지점이 경로를 획득하기 위해 Tier-1 AS에 종속되어 prefix에 도달하지 못하는 것을 관찰합니다. 다른 일부는 Tier-1 AS를 피해 경로를 변경합니다. 우리는 또한 RouteViews 텔레스코프를 사용하여 대부분의 Tier-1 AS를 통해 경로가 사라지는 것을 검증합니다.

Tier-1 AS를 독립시키는 것은 종종 효과적이지 않지만, 대부분의 비-Tier-1 AS에서는 독립이 효과적입니다. 안타깝게도, 이러한 AS들은 직접적인 업스트림이 아닐 때는 거의 트래픽을 운반하지 않습니다. 이러한 소규모 AS들을 독립하는 것은 트래픽에 거의 영향을 미치지 않습니다. 우리는 다시 비-Tier-1 AS (AS57866)를 독립한 후에 트레이스 루트를 수행하고, Tier-1 AS가 독립된 경로를 전파하는 것을 관찰합니다. 이것은 Tier-1 및 비-Tier-1 AS와 함께 독립된 경로가 다른 AS에 의해 다르게 처리됨을 증명합니다.

6.3.2 독립성은 어떤 세부성을 제공하는가?

경로 독립성 coverage는 일반적으로 Tier-1 AS를 독립시킬 수 없기 때문에 제한됩니다. 이와 같은 필터링은 독립성이 허용하는 세분성을 제한합니다: Tier-1 AS를 독립하는 것은 허용되지 않으며, 여러 홉 떨어진 비-Tier-1 AS를 독립하는 것은 트래픽을 대표하지 않기 때문에 거의 영향을 미치지 않습니다. 즉시 이웃을 독립시키면 트래픽이 이동할 수 있지만, 그것은 그들에게 발표하지 않는 것보다 더 복잡합니다. 우리는 세부적인 실험으로 이러한 관찰을 확인하지만, 경로 독립성은 일반적으로 트래픽 엔지니어링에 효과적인 도구가 아님을 결론지을 수 있습니다.

6.4 Playbook 작성

prepend, community 및 독립성에 대한 우리의 이해를 바탕으로, 우리는 이 anycast 네트워크를 위한 가능한 트래픽 구성의 Playbook을 구축할 수 있습니다. 실제로, 우리는 BGP에 연결하는 스크립트를 사용하여 Playbook을 자동으로 작성한 다음 다른 BGP 구성으로 반복하고 새로운 catchment를 측정하기 위해 Verfploeter를 실행합니다. Playbook은 각 anycast 배포에 대해 특정해야 하지만, §7에서 프로세스가 일반화된다는 것을 보여줍니다. Playbook을 사용하면 운영자는 단일 "최선의" 접근 방식이 아닌 Playbook의 다양한 접근 방식의 조합이 트래픽 분배를 더욱 효과적으로 제어함을 보장할 수 있습니다.

Playbook은 라우팅 정책의 다양한 변형과 그에 따른 트래픽 분포의 목록입니다. 표 5는 testbed에 대한 Playbook을 보여주며, 사이트의 65% 블록을 나타내는 기준선이 표시되어 있습니다. 우리는 각 사이트에서 다양한 수준의 prepend(양수 또는 음수)을 그룹화하고, 선택한 community string 및 독립성 구성을 표시합니다.

표 5의 여러 구성에서 요약하면, 표 6은 각 사이트에서 특정 트래픽 비율로 이어지는 조합을 식별합니다. 이 표의 각 글자는 표 5의 특정 구성으로 다시 연결됩니다. 공격 중에, anycast 시스템이 기준선 구성(q)에서 시작하면, AMS가 과부하가 발생하면 운영자는 표에서 상위의 TE 구성('e', 'g', 또는 'j'와 같은)을 선택할 수 있습니다. 그런 다음 운영자는 그 TE 선택이 다른 사이트에 미치는 영향을 볼 수 있습니다(예: 'e'는 다른 사이트에 부하를 더 많이 유발하고 'g'는 BOS의 부하를 증가시키지만 CNF에서 감소시킵니다).

운영자는 또한 두 가지 이유로 트래픽 부하가 있는 Playbook을 사용할 수 있습니다. 첫째, 대부분의 흥미로운 서비스의 부하는 일일 패턴을 갖습니다. 둘째, 각 /24 prefix의 부하는 각 prefix 뒤에 있는 클라이언트의 수에 따라 다를 수 있습니다(자세한 내용은 부록 F 참조). 부하를 사용하여 Playbook을 작성하는 것은 계산적으로 간단합니다. 운영자는 단순히 동일한 catchment 매핑을 사용하면서 prefix 당 부하를 사용할 수 있습니다.

공격 크기 추정과 함께, 공격은 불확실성과 공격자 위치의 불균등으로 동반됩니다. 그러나 Playbook은 다음과 같은 두 가지 측면에서 "단순히 이전 경험에 의존하는 것"보다 훨씬 더 나은 응답을 제공합니다. 방어자는 TE 조치의 결과를 예측할 수 있으며(즉, 트래픽이 어디로 갈지!), 첫 번째 결과가 불완전하면 다른 가능한 결과 사이에서 선택할 수 있습니다.

Playbook의 유연성과 완성도: 표 6은 이 anycast 배포에서 트래픽 엔지니어링이 우리에게 허용하는 "유연성"을 양적으로 보여줍니다. 이 10% 트래픽 단위를 사용하면, AMS는 9개의 옵션, CNF는 7개의 옵션, BOS는 6개의 옵션만 있습니다. AMS와 CNF는 주로 TE 변경 후

트래픽을 교환하기 때문에 BOS가 연결이 덜 된 경우 3 사이트 구성으로 BOS가 50-60% 범위 내에서 트래픽을 처리할 수 있는 구성은 없으며 3 사이트 구성으로 BOS 또는 CNF를 70% 이상으로 운영할 수 없습니다. 이 분석은 AMS와 같은 중앙에 위치한 사이트가 더 많이 나타나며, AMS의 부하를 공유하기 위해 유럽이나 아시아에 다른 사이트를 추가하는 것이 필요할 수 있다는 것을 나타낼 수 있습니다.

Routing Policy	Traffic to Site (%) AMS	BOS	CNF
(a) 6peers, 12peers	~5	~35	~55
(b) Route-server	15	35	55
(c) All-IXP-Peers/Poison transits	15	35	45
(d) 3xPrepend AMS	15	35	45
(e) 2xPrepend AMS	25	35	45
(f) 1xPrepend AMS	35	25	35
(g) -3xPrepend BOS	25	65	5
(h) -2xPrepend BOS	35	65	5
(i) -1xPrepend BOS	45	45	15
(j) -3xPrepend CNF	25	15	65
(k) -2xPrepend CNF	35	5	55
(l) -1xPrepend CNF	45	5	45
(m)Transit-1	45	25	35
(n) Transit-2	55	15	25
(o) Poison Tier-1/Transit-2	35	25	35
(p) Poison Transit-1	55	25	25
(q) Baseline	65	15	15
(r) 1,2xPrepend BOS	65	5	25
(s) 3xPrepend BOS	75	5	25
(t) 1,2,3xPrepend CNF	75	15	5
(u) -1,-2,-3xPrepend AMS	85	5	5

Table 5: Policies and traffic distribution (in 10% bins); groups sorted by rough fraction of traffic to AMS, and colors showing the traffic compared to the baseline distribution.

표 5: 정책 및 트래픽 분포(10% 구간) AMS에 대한 대략적인 트래픽 비율을 기준으로 정렬된 그룹 및 기준 분포와 비교하여 트래픽을 표시하는 색상입니다.

Traffic to Site (%)	AMS	BOS	CNF
0-10	a	k, l, r, s, u	g, h, t, u
10-20	b, c, d	j, n, q, t	i, q
20-30	e, g, j	f, m, o, p	n, r, p, s
30-40	f, h, k, o	a, b, c, d, e	f, m, o
40-50	i, l, m	i	c, d, e, l
50-60	n, p	–	a, b, k
60-70	q, r	g, h	j
70-80	s, t	–	–
80-90	u	–	–
90-100	–	–	–
Traffic options	9	6	7

Table 6: `Peering` playbook (AMS, BOS, and CNF)

표 6: peering Playbook(AMS, BOS 및 CNF)

7. 배포 안정성과 제약 사항

§6에서는 BGP 기반 TE가 상당한 유연성을 제공함을 보였습니다. Playbook을 작성함으로써 방어자들은 transit 공급자, prepending, community string 및 독립성이 특정 배포에 어떻게 영향을 미치는지를 탐색할 수 있습니다. 다음으로 선택한 사이트와 사이트의 수에 따라 결과가 얼마나 안정적인지 살펴보겠습니다. Playbook의 세부사항은 각 배포마다 다르지만, 우리의 testbed가 모든 가능한 배포를 대표한다고 주장하지 않습니다. 우리의 접근법이 유연하며 다양한 배포에서 공격에 대응할 수 있는 것을 보여줍니다 - 우리의 접근법은 일반화됩니다.

7.1 anycast 사이트 선택의 영향

먼저 사이트가 우리의 Playbook에 어떤 영향을 미치는지 살펴봅니다. 새로운 사이트는 위치와 peering에 의존하기 때문에 catchment를 변경합니다. §6.1에서 우리는 세 개의 대륙에 있는 세 가지 Peering 사이트에서의 catchment를 연구했습니다: 유럽의 대형 상업적 IXP에 위치한 AMS; 브라질의 학술적 백본 transit에 위치한 CNF; 그리고 미국의 학술 사이트인 보스턴에 위치한 BOS. 지리적 위치보다 중요한 것은 사이트 연결성입니다. 여러 transit 공급자는 트래픽 제어와 세분성을 증가시키는 데 더 중요한 역할을 합니다. 대학 네트워크 내에서 연결이 잘 되지 않은 사이트는 일반적으로 트래픽 제어 옵션을 제공하지 않습니다.

prepending 기준선: Figure 9는 세 개의 북미 사이트에 대한 양수 및 음수 prepending의 catchment 크기를 보여줍니다. 이제 기준선 분포가 균형이 맞지 않지만 이전보다는 적습니다. SEA가 50%의 블록을 캡처합니다. 우리는 Peering 운영자들과 SEA의 많은 트래픽을 논의했습니다. 그들은 SEA가 시애틀 IXP에 인접해 있어 많은 상업 공급자로부터 한 홉 떨어진 위치에 있다고 의심합니다. 어떤 사이트가 가장 큰 가시성을 가지는지는 그것의 peering에 따라 달라지며, 배포에서 배포로 다를 것입니다.

prepending coverage와 세분성: 이전 실험과 마찬가지로 트래픽 이동을 보기 위해 prepending을 조정할 수 있습니다. 이 세 가지 사이트에서는 BOS와 SEA가 하나의 양수 또는 음수 prepending 후에 매우 빠르게 트래픽이 이동합니다. SLC는 기준선에서 가장 작은 catchment를 가지고 있기 때문에 음수 prepending 단계마다 더 많은 coverage를 얻습니다. 종종(항상은 아니지만) 학술 사이트는 peer가 적거나 peer가 유사한 연결성을 갖기 때문에 세분성이 낮습니다. 결과적으로 AS-Path 길이의 작은 변경으로 인해 한 사이트가

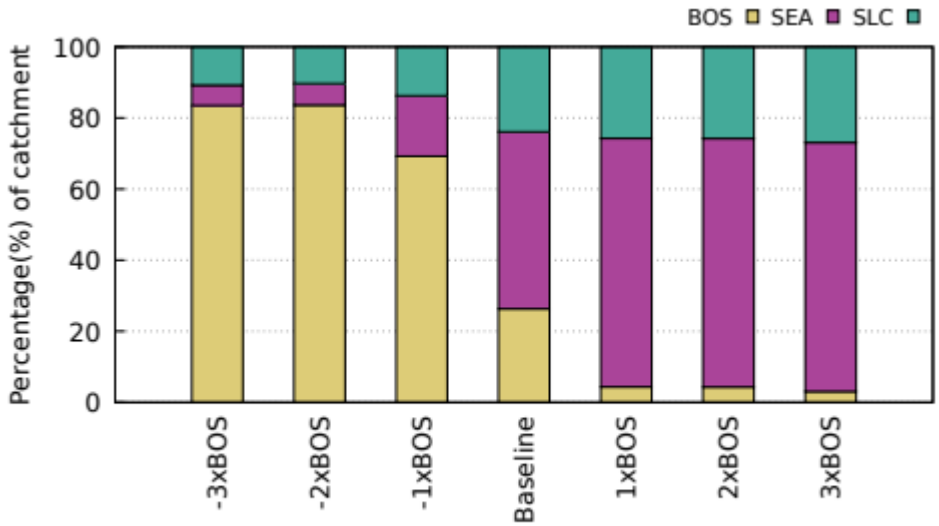
다른 사이트에서 멀어집니다. 또한 이러한 세분화가 낮은 제어는 특정 배포에 특정한 Playbook을 구축하거나 anycast 토폴로지가 변경될 때 중요성을 보여줍니다.

community coverage: community는 IXP와 transit 공급자에서 일반적이지만, 학술 네트워크(NRENs)는 더 간단한 community 세트를 갖습니다. 이러한 학술 사이트 중 어느 것도 community string을 제공하지 않습니다.

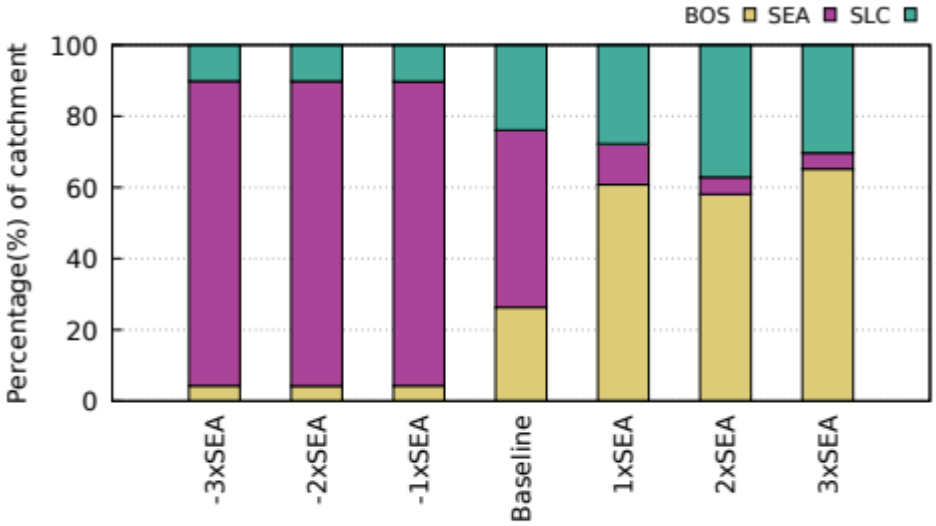
이 관측은 이전의 coverage 관측을 확인합니다: community string 지원이 균일하게 제공되지 않습니다. 우리는 Peering에서 다른 사이트 조합도 조사했고 유사한 결과를 얻었습니다(논문의 확장 버전에서 확인할 수 있습니다).

경로 독립성: 우리는 보스턴, 솔트레이크 시티 및 시애틀에 있는 세 개의 사이트에서 경로 독립성 실험을 반복했습니다. Tier-1 AS는 일반적으로 독립시킬 수 없음을 확인합니다(§6.3.1). 우리는 라우트 유출을 방지하기 위한 필터도 독립성을 방해한다는 것을 볼 수 있습니다.

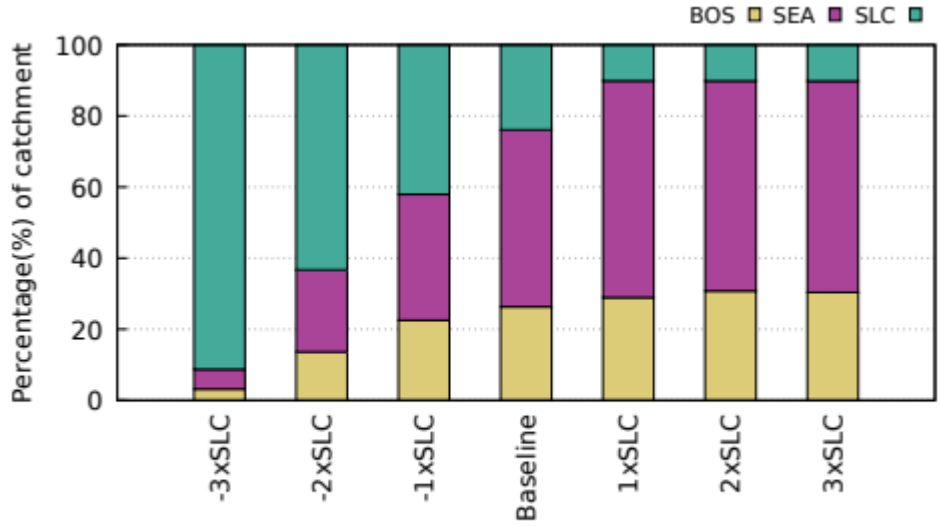
우리의 실험은 catchment가 배포별로 다르지만, 우리의 정성적인 결과는 유지된다는 것을 확인합니다 – prepending은 작동하지만, 그것은 미세한 것이며, community string과 독립성은 어디서나 지원되지 않습니다.



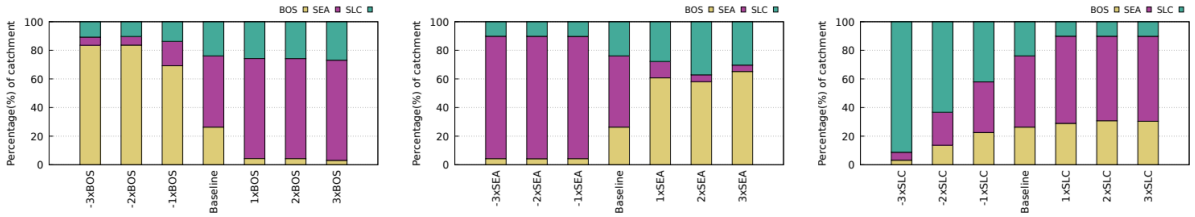
(a) BOS site.



(b) SEA site.



(c) SLC site.



(a) BOS site. (b) SEA site. (c) SLC site.

Figure 9: `Peering`: Impact of choosing BOS, SEA and SLC sites on 2020-02-28

그림 9: peering: 2020-02-28에 BOS, SEA 및 SLC 사이트 선택이 미치는 영향

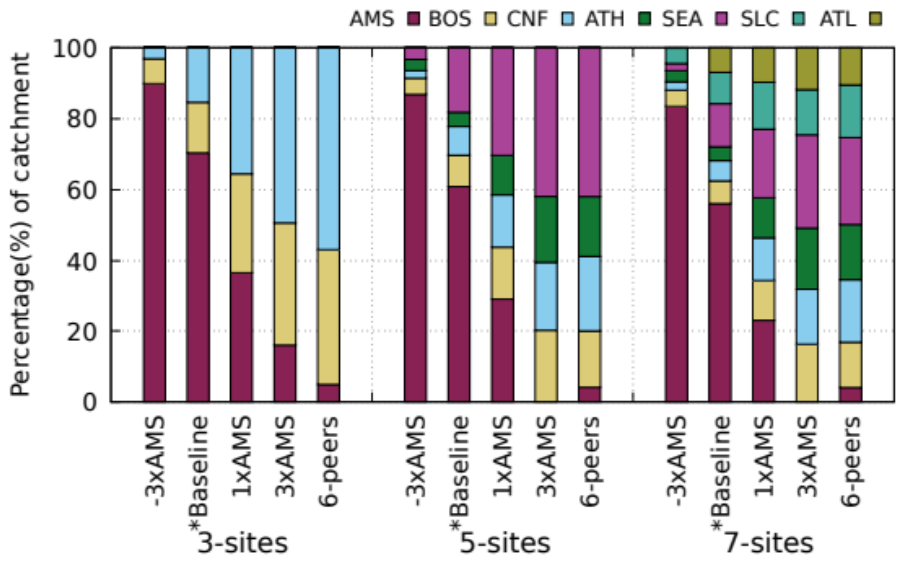


Figure 10: Peering: Impacts of changing the number of anycast sites from 2020-04-07 to 2020-04-10.

그림 10: peering: 2020-04-07에서 2020-04-10으로 anycast 사이트 수 변경이 미치는 영향.

7.2 anycast 사이트 수의 영향

다음으로, 사이트 수를 변경하고 그것이 트래픽 제어에 어떻게 변화하는지를 살펴봅니다. 우리는 각 testbed에서 3, 5 및 7개의 사이트를 선택하고 방어 옵션을 평가하기 위해 Playbook을 작성합니다. Figure 10은 사이트 수별로 그룹화된 선택된 구성을 보여줍니다.

기준선: 더 많은 사이트로 인해 전체 용량이 증가하고 각 사이트의 기준선 부하가 감소합니다. 예를 들어, Figure 10에서 가장 큰 사이트(AMS)의 기준선(별표(*)로 표시됨)은 세 개의 사이트에서 블록의 70%에서 다섯 개의 사이트에서는 61% 및 7개의 사이트에서는 56%로 이동합니다. 작은 사이트는 덜 이동합니다(BOS는 14%에서 6% 및 6%로, CNF는 15%에서 8% 및 6%로 이동). 더 큰 용량과 분포는 전체 서비스를 고갈시키려면 더 크고 분산된 공격자가 필요합니다. 우리는 확장된 논문의 대체 testbed Tangled에서 유사한 결과를 보았습니다.

트래픽 유연성: 더 많은 사이트로 인해 가장 큰 사이트가 보통 가장 큰 변화를 보이며 가장 적은 catchment 크기를 갖습니다. Figure 10에서 기준선을 한 번 prepending과 비교하면, AMS는 세 개의 사이트에서 70%에서 37%로, 다섯 개의 사이트에서는 61%에서 29%로, 일곱 개의 사이트에서는 56%에서 23%로 각각 절반으로 줄어듭니다.

더 많은 사이트를 사용하더라도 일부 블록은 종종 특정 사이트에 "갇혀" 있습니다. 음수 prepending 세 번으로 AMS는 대부분의 트래픽을 얻지만, 세 개의 사이트에서는 90%에 도달하고 다섯 개와 일곱 개의 사이트에서는 각각 87%와 84%에 머무릅니다. 각 사이트에는 고유한 "갇힌 블록" 집합이 있으며 트래픽 엔지니어링으로 이동하지 않습니다.

더 많은 사이트를 사용하면 경로 prepending이 민감하지 않아지므로 BGP community의 세부 제어가 더 중요해집니다. 예를 들어, 5 또는 7개의 사이트에서 AMS에는 community를 사용한 선택적 공지가 필요합니다. 세 번 prepending을 사용하면 모든 트래픽이 이동합니다.

새로운 사이트: 더 많은 사이트를 추가하면 어떻게 Playbook이 새로운 사이트의 배치를 안내하는지 볼 수 있습니다. 새로운 사이트의 트래픽 이동을 예측하는 것은 어렵지만, 테스트 prefix로 실험을 진행하여 사전 배치

Playbook을 작성할 수 있습니다.

7.3 시간에 따른 Playbook 안정성

루팅이 즉시 변경되는 경우 Playbook은 제한적으로 사용됩니다. 링크가 실패하거나 ISP가 새로운 peering을 시작하거나 새로운 transit을 구입할 때 라우팅이 변경됩니다. Playbook이 적용 가능한 기간은 얼마나 될까요?

이 질문에 대한 답변으로, Table 7은 기준 구성에 대한 시간 경과에 따른 각 catchment로 이동하는 /24 블록의 분수를 보여줍니다. 블록의 분수는 일반적으로 매우 안정적이며, 약 5%의 블록이 사이트 내외로 이동합니다. 또한 이전 연구에서는 몇 시간에서 며칠 동안 매우 강력한 anycast 안정성이 확인되었습니다. 우리는 B-root catchment의 안정성을 확인했습니다. 두 주 후에 0.35%의 prefix가 변경되었으며, 한 달 후에는 0.65%의 prefix가 catchment를 변경했습니다(더 많은 내용은 부록 E를 참조하십시오). catchment는 상대적으로 안정적이지만, 운영자들이 주기적으로 Playbook을 새로 고치리라고 기대합니다(아마도 주간 또는 월간).

Months	AMS(%)	BOS(%)	CNF(%)
2020-02	68.1	14.6	17.3
2020-04	70.4	14.2	15.4
2020-06	65.3	14.1	20.6

Table 7: Percent blocks in each catchment over time.

표 7: 시간 경과에 따른 각 유역의 블록 비율.

8. 공격 대응 전략

이 섹션에서는 시스템에서 처리되는 네 가지 실제 공격을 설명합니다. 다양한 유형의 공격에 성공적으로 대응할 수 있는 다른 방법을 보여줍니다.

방법론: 우리는 B-root 서버 운영자, 네덜란드 국립 스크러빙 센터 및 익명의 기업 네트워크에서 발생한 실제 공격을 사용합니다. 이러한 사건은 다형성, 적대적 및 부피 공격을 포함합니다.

우리는 이러한 사건들을 세 사이트 anycast 네트워크에 대해 트래픽 비율을 시뮬레이션하여 평가합니다. 처음 두 이벤트는 §6에서의 우리의 AMS, BOS, CNF 구성을 사용합니다. 마지막 이벤트에서는 §7.1의 BOS, SEA, SLC를 사용합니다.

우리는 시뮬레이션에서 트래픽을 재생하며, 각 anycast 사이트에 트래픽을 실험에서 측정된 catchment를 기반으로 할당합니다. 우리는 점진적인 경로 전파를 시뮬레이션하지 않지만, 대신 라우팅은 변경 후 300초 후에 적용됩니다(보수적인 한계로, 대부분의 라우팅 변경은 해당 시간의 절반 이내에 발생합니다). 그런 다음 각 사이트에서의 트래픽 수준을 평가하고 대상 용량과 비교합니다.

각 공격에 대해 우리 시스템을 방어 운영하며 공격 크기를 추정하고 사전에 계산된 Playbook 응답을 선택합니다. Playbook이 다양한 응답을 허용하므로 선택 옵션이 있는 경우 방어의 다른 방법을 선택합니다: prepending, 음수 prepending 또는 community 스트링 (Figure 11).

2017 다형성 공격: 첫 번째 사건은 2017년 3월 6일에 B-root [51]에서 발생한 DNS 플러드입니다 (Figure 11a). 이 사건은 공격 쿼리가 RANDOM.qycl520.com₩032 (0초부터)와 같은 공통 형식을 가지고 있으며 (4750초에 변경되므로 다형성을 띠고 있음), 부피적으로 다형성 공격이었습니다. 우리는 각 anycast 사이트에서 60k 패킷/초 (30 Mb/s)의 용량을 가정합니다. 이 사건은 B-root가 그 때 활성 상태였을 때 모든 활성 anycast 사이트에서 완전히 포착할 수 있을 정도로 충분히 작았습니다. 이 사건은 약 5시간 동안 지속되었지만,

우리는 처음 2.25시간만 표시합니다. 오늘의 서비스 및 공격 용량은 모두 훨씬 더 큽니다. 우리는 작은 공격을 사용하여 결과가 유사할 것으로 예상됩니다.

Figure 11a에서 AMS 사이트가 용량을 초과하는 100k 패킷/초의 트래픽을 수신하는 것을 확인할 수 있습니다 (적색 줄무늬 영역으로 표시됨). 우리 시스템은 비트레이트 경고에서 공격을 감지합니다. 그런 다음 AMS 과부하를 추정하기 위해 관측된 부하와 액세스 비율을 사용하여 제공된 부하를 계산합니다. 시스템은 네트워크를 각 사이트로의 패킷 수에 매핑하는 데 사전에 계산된 Playbook (Table 6)을 사용합니다. 이 매핑을 사용하여 우리 시스템/운영자는 응답을 선택할 수 있습니다. Figure 11a에서 선택한 라우팅 접근 방법의 영향을 볼 수 있습니다—community 스트링을 사용하여 transit-1에만 알리기. 300초 후에 공격이 완화되었음을 나타내는 줄무늬 영역이 없음을 볼 수 있습니다.

공격자는 쿼리 이름을 4750초에 변경하여이 공격을 다형성으로 만듭니다. 쿼리 이름에 대한 필터링은 반응해야하지만 라우팅 변경은 이러한 유형의 변경과 관계없이 공격을 완화할 수 있습니다.

2021 다형성 가변 길이 공격: 다음으로 2021년 9월 5일에 기업 네트워크에서 시작된 HTTP 공격을 살펴보겠습니다 (Figure 11b). 이 다형성 공격은 세 번의 일시 중지 후에 변경됩니다. 초기 공격은 수백만 개의 HTTP GET (15k 패킷/초)로 이루어지며, IoT 봇넷에서 시작됩니다. 이는 기업 운영자가 IP 기반 필터링을 배치할 때 종료됩니다. 약 1000초 후에 다른 봇넷이 HTTP GET을 사용하여 무작위 경로 (캐싱을 피하기 위해) 및 위조된 TCP ACK를 결합한 다중 벡터 공격을 시작합니다. 그런 다음 소각, 짧은 터지기, 다른 소각 및 끝까지의 터지기를 볼 수 있습니다.

0초에 발생한 초기 공격은 하나의 사이트 (AMS)를 과부하로 만들어 우리의 라우팅 응답을 유발합니다. 추정 후에 우리는 AMS에서 떠나는 경로 이동을 시작합니다. 그러나 공격은 아직 끝나지 않은 상태에서 (90초 후) 경로가 여전히 변경되고 있습니다.

정상 트래픽 소스가 유럽에서 유래하기 때문에 대부분의 트래픽이 AMS로 이동합니다. 1020초에 공격 봇넷이 변경되며, 아시아 및 남미에서의 공격 트래픽이 더 많아집니다(MaxMind의 IP 지리적 위치 정보를 기반으로 함).

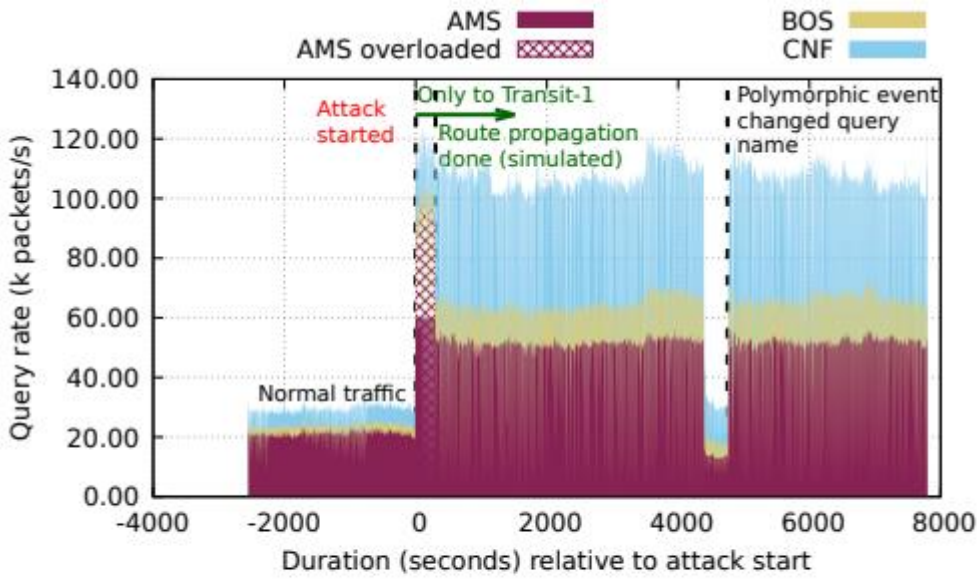
우리의 초기 공격에 대한 라우팅 변경은 여전히 유지되어 있으며, 갱신된 공격은 성공적으로 세 사이트 전체에 분산되어 AMS가 새로운 공격을 견딜 수 있습니다.

이와 같은 공격의 이동은 더 정교한 적대자와 일반적입니다. 라우팅 변경으로 방어하는 모든 접근 방법(우리를 포함하여)은 경로 전파 시간에 의해 제한됩니다. 따라서 이러한 방어의 적용 가능성은 0초에 발생한 것과 같이 짧은 기간 동안 발생하는 공격에 대해서는 제한적입니다. 그러나 트래픽을 분산시키면 이러한 공격에 대해 보호할 수 있습니다. 1000초 후에 발생한 재발되는 공격을 볼 수 있습니다. 이러한 공격의 변형은 공격이 계속되는 동안 방어 효과를 검토하는 것의 중요성을 보여줍니다.

다른 anycast 토폴로지의 예상 공격: 마지막으로 2021년 8월 25일에 네덜란드 국립 스크러빙 센터에서 LDAP 증폭 공격을 고려합니다.

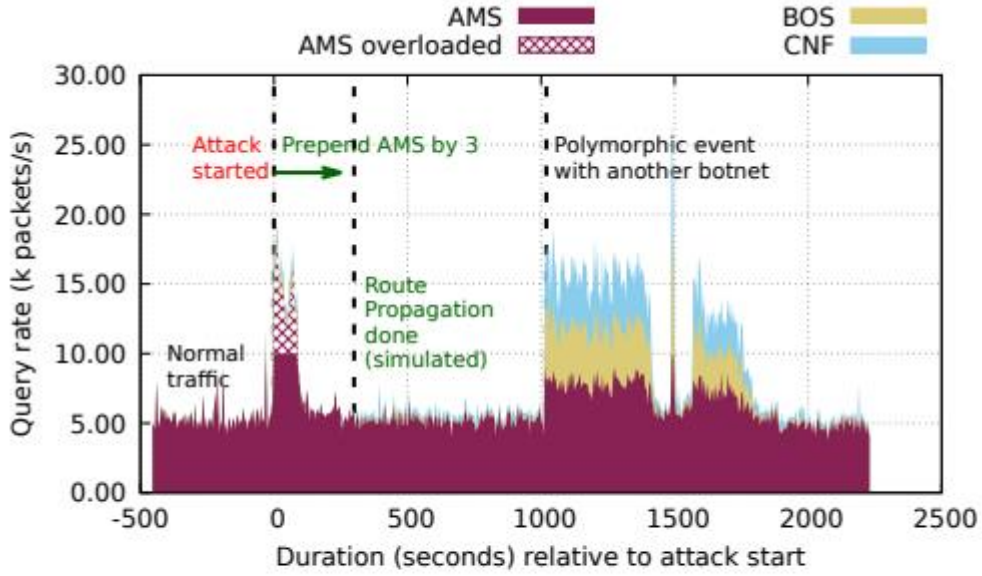
이 경우에는 수퍼 사이트를 BOS에서 시뮬레이션하고, 다른 사이트 (SEA 및 SLC)는 약 절반의 용량을 지원합니다(700k 패킷/초). Figure 11c에서는 SEA, 작은 사이트에서 트래픽이 얼마나 압도되는지를 보여줍니다. 그러나 슈퍼 사이트에서 처리할 수 있습니다. 우리는 음수 prepending으로 응답하고, 300초 후에 트래픽이 BOS로 이동하는 것을 볼 수 있습니다. 이 응답은 공격을 완화합니다(줄무늬 영역이 없음).

기타 공격: 우리는 추가적인 공격을 평가했으며, 그에 대해 부록 G에서 설명합니다. 추가 다형성 및 부피 공격은 라우팅이 경로를 전파한 후에도 성공적으로 처리될 수 있음을 보여줍니다.



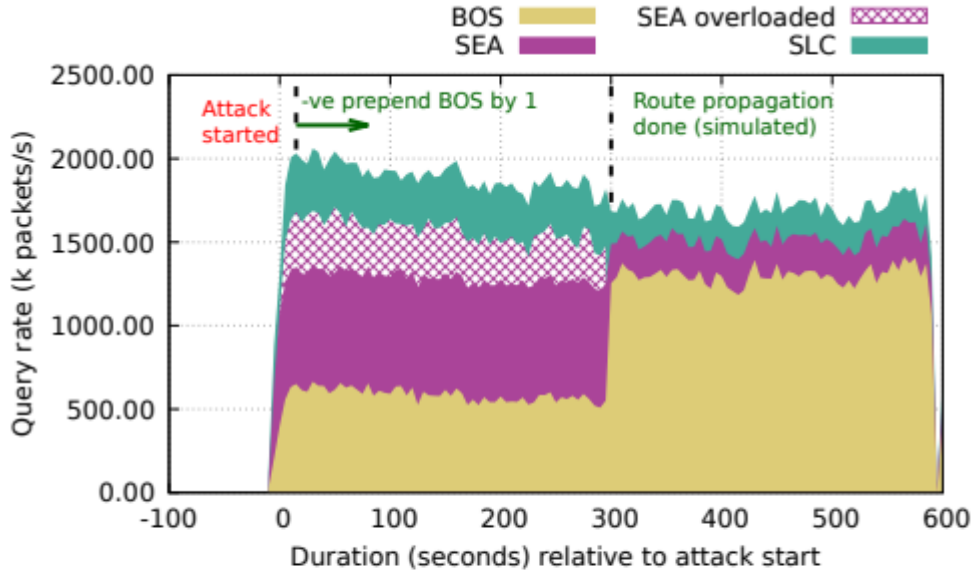
(a) A polymorphic attack at `B-root` defended with community strings.

community string로 방어된 B-루트에서의 다형성 공격



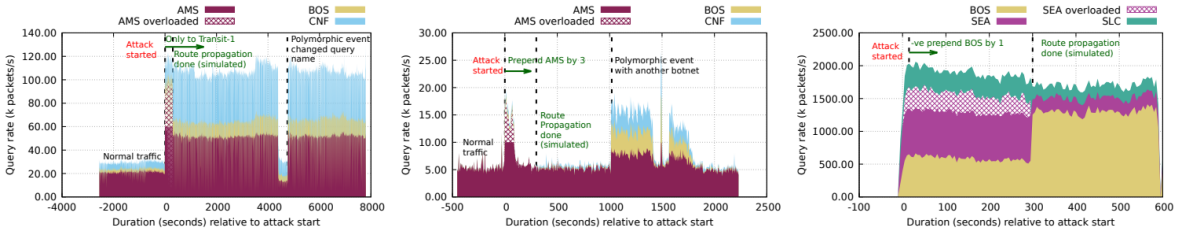
(b) An adversarial event at an enterprise mitigated using positive prepending.

긍정적인 추가를 사용하여 완화된 기업의 적대적 이벤트.



(c) An event captured at the Dutch National Scrubbing Center defended using negative prepending.

네덜란드 국립 스크러빙 센터(Dutch National Scrubbing Center)에서 캡처한 이벤트는 부정 추가를 사용하여 방어했습니다.



(a) A polymorphic attack at `B-root` defended with community strings.

(b) An adversarial event at an enterprise mitigated using positive prepending.

(c) An event captured at the Dutch National Scrubbing Center defended using negative prepending.

Figure 11: Different attacks with various responses.

그림 11: 다양한 대응의 다양한 공격.

9. 한계와 향후 연구

우리의 라우팅 옵션 Playbook(§3)은 많은 공격에 효과적입니다(§8 및 부록 G). 그러나 어떤 방어라도 완벽하지는 않습니다. 다음은 알려진 한계와 향후 연구 분야에 대해 설명합니다.

먼저, 인터넷 라우팅은 분산되어 있어 수렴하는 데 시간이 필요합니다. 라우팅 방어의 효과는 수렴될 때까지 볼 수 없습니다. 우리는 5분보다 빨리 변경하지 않습니다. 라우팅 수렴 시간은 라우팅 변경이 짧은 기간 동안 공격에 제한적으로 적용될 것을 의미합니다 (5분 미만). 라우팅 변경은 서비스에 해를 끼치지 않지만, 이점은 라우팅이 이동할 때까지 나타나지 않을 수 있습니다.

또한, 라우팅 수렴은 트래픽 소스를 빠르게 변화시키는 다형성 공격이 더 효과적일 것을 의미합니다. 라우팅 변경은 방법을 변경하더라도

트래픽 양에 따라 효과를 발휘하는 다형성 공격에 견고합니다. 이는 §8의 사건에서 보여 준 대로, 무엇이든 상관없이 부하를 분산시킬 것입니다. 그러나 라우팅이 수렴하는 것보다 트래픽 위치가 더 빠르게 이동하는 공격을 방어할 때는 가장 많은 트래픽을 처리해야 하는 가장 안 좋은 경우의 볼륨을 각 사이트에 할당해야 합니다. 신속한 변화는 방어를 어렵게 만들지만 불가능하지는 않습니다.

마지막으로, 우리는 기본 서비스의 anycast catchment가 천천히 변화한다고 가정합니다 (일 수록). 우리는 §7.3에서 이러한 가정이 일반적으로 유지된다는 것을 보여 주었습니다.

우리는 공격 중에 부하를 catchment 전체로 균형을 맞추기 위해 라우팅을 변경하지만, 공격 원점을 명시적으로 찾으려고 하지는 않습니다. 향후 연구로써 이러한 정보를 사용하여 방어 선택을 개선할 수 있습니다.

공격 대응은 서비스 운영자와 공격자의 인간 요소에 의존합니다. 이러한 인간 요소를 명시적으로 연구하는 것은 잠재적인 향후 연구입니다. 우리의 현재 작업은 방어 기술적 가능성에 중점을 두었습니다.

10. 결론

본 논문은 DDoS 방어를 위한 여러 anycast 방법에 대한 첫 공개적 평가를 제공합니다. 우리의 시스템은 공격 크기를 추정하고, 사전 계산된 Playbook에서 전략을 선택하며, 자동으로 트래픽 엔지니어링(TE)을 수행하여 부하를 다시 균형잡거나 운영자에게 조언합니다. 우리의 기여는 공격 크기 추정과 Playbook 구축입니다. 우리는 TE 메커니즘을 실험적으로 평가하여, prepending이 널리 사용되지만 제한된 제어만을 제공하고, BGP community와 path poisoning이 반대인 것을 보여줍니다.

감사의 글: ASM Rizvi와 John Heidemann의 이 논문에 대한 작업은 일부로 DHS HSARPA 사이버 보안 부서의 계약 번호 HSHQDC-17-R-B0004-TTA.02-0006-I를 통해 지원됩니다. Joao Ceron과 Leandro Bertholdo의 이 논문에 대한 작업은 네덜란드 과학 연구 기구 (4019020199) 및 유럽 연합의 지평 2020 연구 및 혁신 프로그램 (830927)의 지원을 받았습니다. 우리는 우리에게 측정을 실행할 수 있도록 허용한 Peering 및 Tangled 관리자들에게 감사드립니다. 또한 네덜란드 국가 스크러빙 센터가 DDoS 데이터를 공유해 준 것에 감사드립니다.

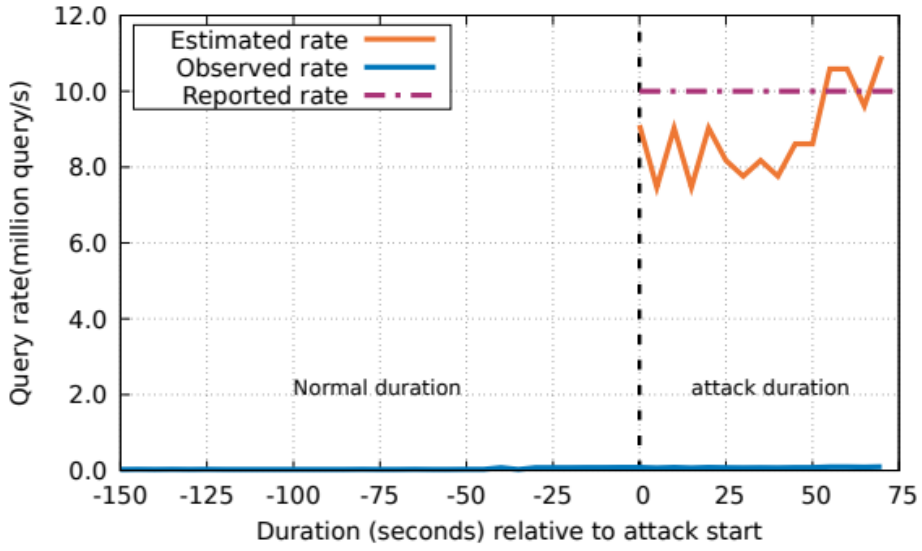


Figure 12: Estimating real-world attack events: estimating June 2016 event with 0.91% access fraction.

그림 12: 실제 공격 이벤트 추정: 0.91% 액세스 비율로 2016년 6월 이벤트를 추정합니다.

부록 A: anycast와 Verfploeter

IP anycast는 들어오는 요청을 다른 위치(사이트)로 라우팅하는 라우팅 방법입니다. 각 사이트는 동일한 IP 주소를 사용하지만 지리적으로 다른 위치에 있습니다. anycast는 인터넷 라우팅과 BGP를 사용하여 사용자를 다른 사이트에 연결하는 방법을 결정합니다. 이것을 사이트의 anycast 수용소라고 합니다. BGP에는 라우팅 정책과 대략적인 거리를 고려하는 표준 경로 선택 알고리즘이 있습니다.

운영자는 BGP를 조작하여 라우팅 결정 프로세스에 영향을 줄 수 있습니다. 우리는 이를 §3.4.1에서 설명하고 있으며, DDoS 공격 중에 부하를 재분배하는 데 사용할 수 있는 TE 기술을 설명하고 있습니다.

우리는 클라이언트를 anycast 사이트 매핑하기 위해 Verfploeter [20]를 사용합니다. Verfploeter를 사용하여 BGP Playbook을 다양한 BGP 변경 사항으로 구축합니다(§6.4). Verfploeter의 주요 개념은 수백만 개의 주소 블록에 대한 핑을 anycast prefix를 소스 주소로 사용하여 보내는 것입니다. 이러한 핑에 대한 응답은 인터도메인 라우팅 시스템에 의해 가장 가까운 anycast 사이트로 라우팅되며, 여기서 주소 블록을 anycast 사이트에 매핑할 수 있습니다.

부록 B: 운영자 지원 시스템

운영자를 지원하기 위해 방어용 인터페이스를 제공합니다(§3.4.3). anycast 네트워크를 반응하고 다시 구성하기 위해 운영자는 이와 유사한 웹 인터페이스를 사용할 수 있습니다. 이 인터페이스에서 운영자는 각 anycast 사이트에서 부하를 증가시키거나 감소시킬 비율을 선택할 수 있습니다. 슬라이더 위치의 가능한 범위는 Playbook 대안 또는 라우팅 정책의 프리셋에 기반합니다. 이 프로세스는 운영자에게 Playbook 복잡성을 숨겨주므로 프로세스가 오류를 범하기 적고 직관적이며, 그러면서도 운영자에게 BGP 라우팅의 완전한 제어를 제공합니다.

그림 3에서는 이 인터페이스의 스냅샷을 시각화할 수 있습니다. 각 슬라이더는 anycast 사이트를 나타내며, 각 사이트에는 "점"으로 표시된 사전 설정이 있습니다. "점"의 위치는 Playbook을 생성하는 데 사용된 모든 측정 결과입니다. 막대 그래프는 측정 프로세스의 결과를 보여주며, 각 위치에 대한 예상 트래픽을 시각화한 후 운영자는 프로덕션 네트워크에 구성을 적용할 수 있습니다.

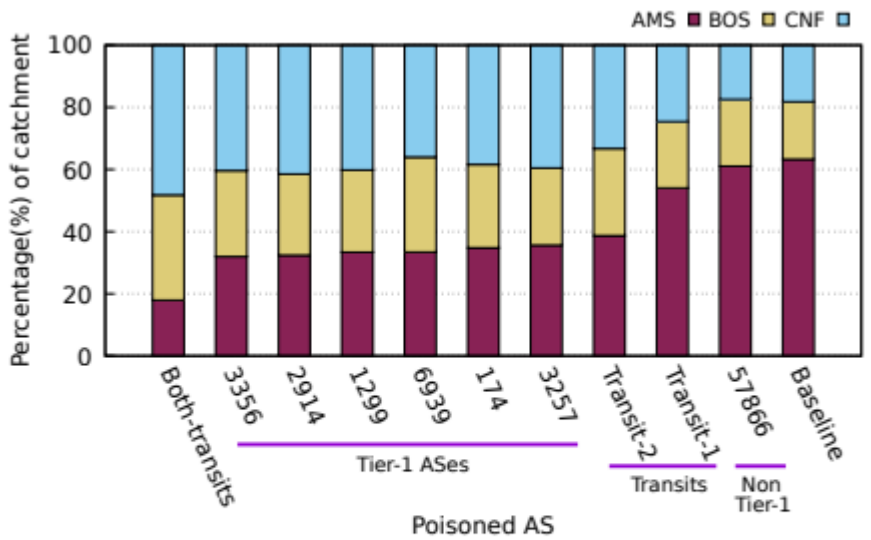


Figure 13: Peering: Impact of path poisoning (from AMS on 2021-04-09).

그림 13: peering: 경로 중독의 영향(2021-04-09 AMS 기준)

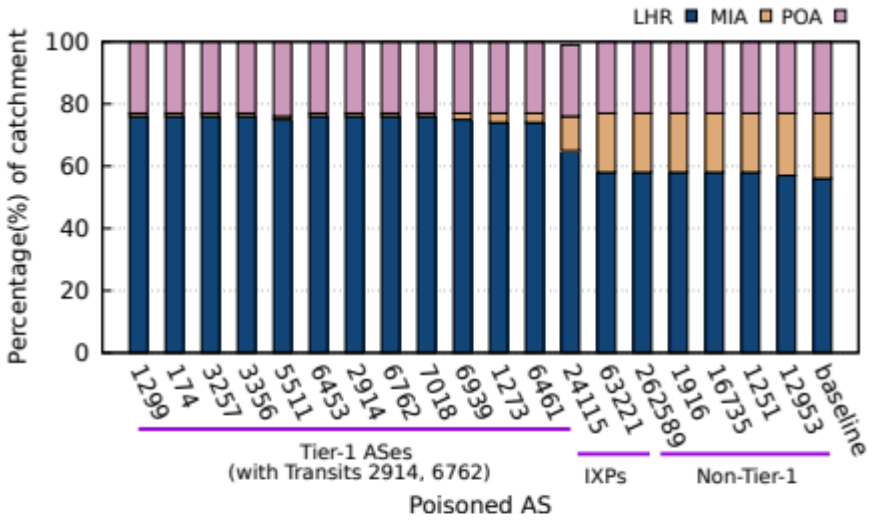


Figure 14: `Tangled`: Impact of path poisoning (from MIA on 2021-04-11).

그림 14: 얽힌: 경로 중독의 영향(2021년 4월 11일 MIA에서).

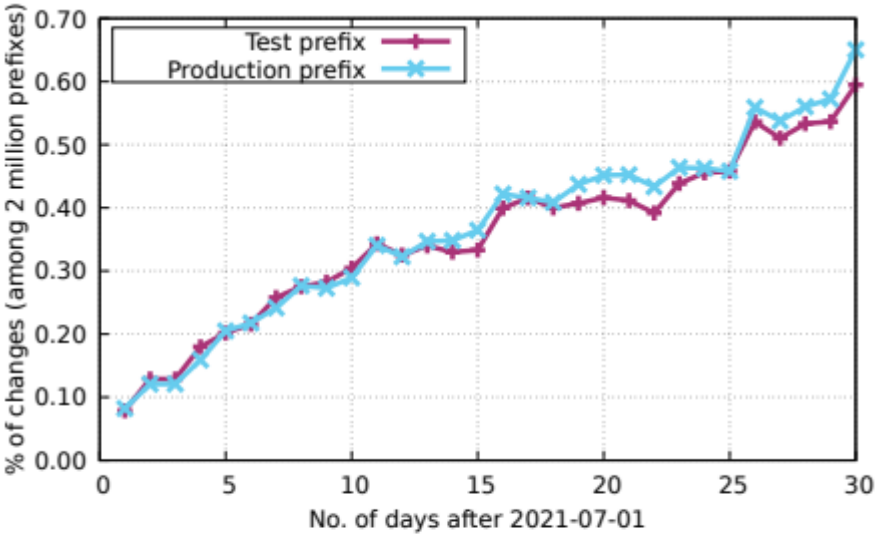


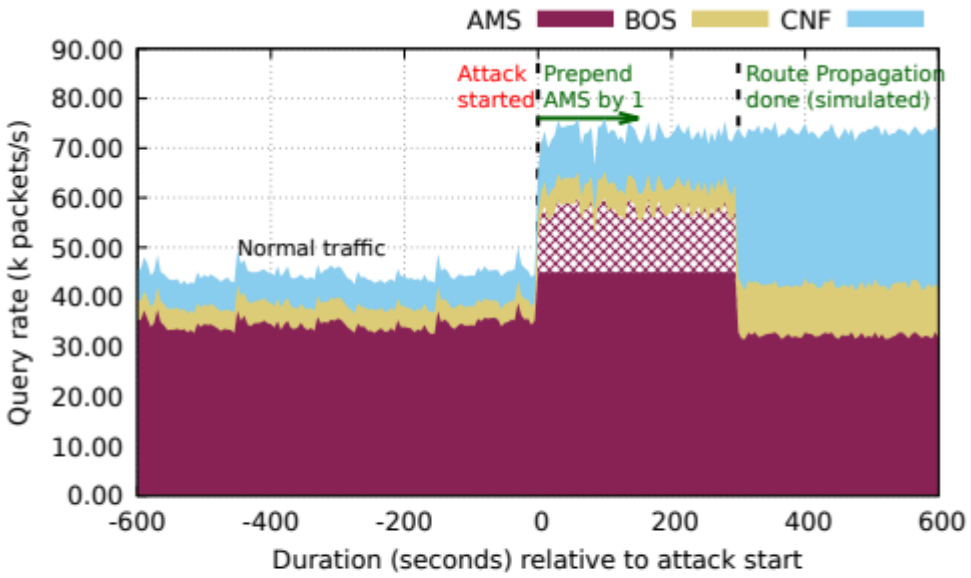
Figure 15: One month of catchment stability in `B-root`.

그림 15: B-루트의 1개월 집수 안정성.

Policy / Day		AMS(%)				BOS(%)				CNF(%)			
		00 GMT	06 GMT	12 GMT	18 GMT	00 GMT	06 GMT	12 GMT	18 GMT	00 GMT	06 GMT	12 GMT	18 GMT
Baseline	Day-1 load	77	84	84	84	10	8	8	7	13	8	8	9
	Day-2 load	77	84	84	80	10	8	7	9	13	8	9	11
	Catchment	68				15				17			
1xPrepend AMS	Day-1 load	43	49	49	58	18	20	18	13	39	32	33	29
	Day-2 load	43	46	46	50	18	18	18	18	39	36	36	32
	Catchment	37				25				38			
1xPrepend BOS	Day-1 load	78	85	83	83	4	3	4	3	18	12	13	14
	Day-2 load	78	85	83	79	4	4	4	4	18	12	13	16
	Catchment	70				7				23			
1xPrepend CNF	Day-1 load	83	88	87	87	11	10	9	8	6	2	3	5
	Day-2 load	83	89	87	85	11	9	9	10	6	2	4	5
	Catchment	77				19				4			
Transit-1	Day-1 load	88	93	92	91	5	4	5	3	6	3	4	5
	Day-2 load	88	93	92	90	5	4	4	5	7	2	4	6
	Catchment	38				24				38			

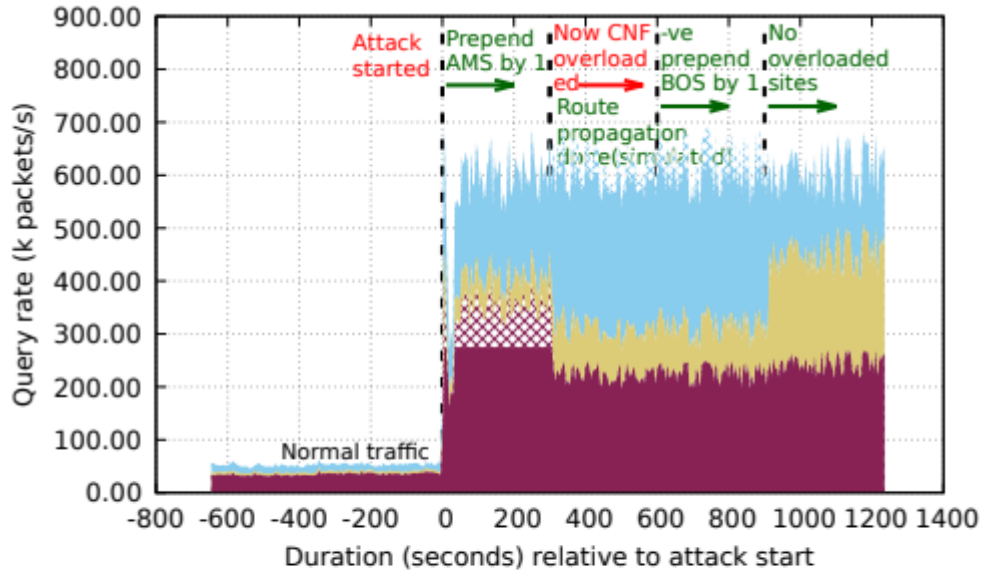
Table 8: Load distribution with `Peering` catchment and `B-root` load. Catchment: 2020-02-24, Load: 2020-02-25 and 2020-02-26 (only showing selected policies). Catchment distribution remains similar over the course of the day showing by a single value.

표 8: peering 유역 및 B-루트 부하를 사용한 부하 분포. 유역: 2020-02-24, 로드: 2020-02-25 및 2020-02-26(선택한 정책만 표시). 유역 분포는 단일 값으로 표시되는 하루 동안 유사하게 유지됩니다.



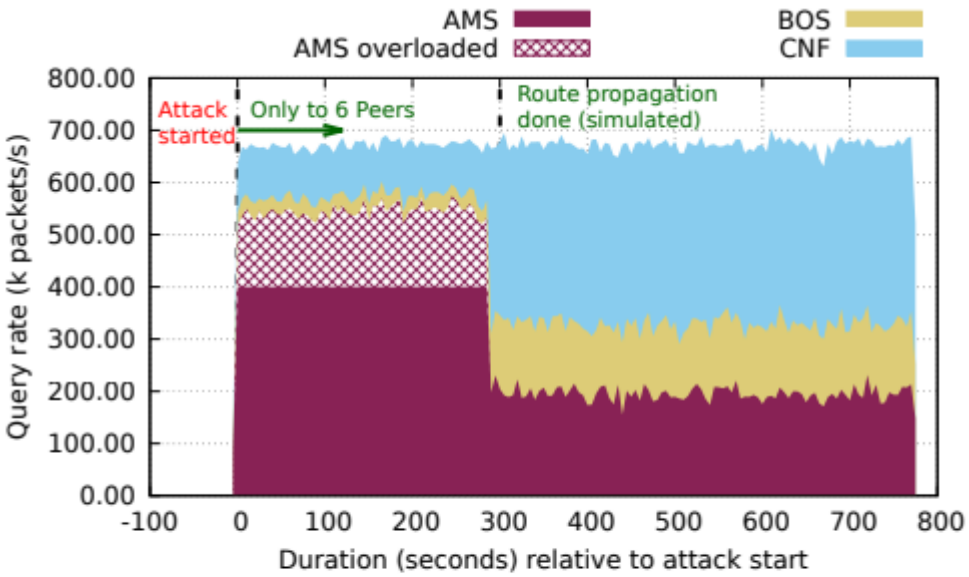
(a) A 2020 event at `B-root` defended using positive prepending.

B-root의 2020 이벤트는 긍정적인 추가를 사용하여 방어되었습니다.



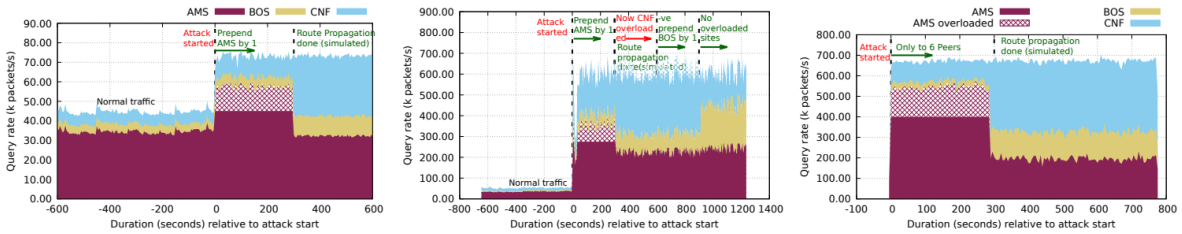
-(b) A 2021 event at B-root defended using negative prepending.

B-root의 2021년 이벤트는 부정 추가를 사용하여 방어되었습니다.



(c) A 2021 event at the Dutch National Scrubbing Center mitigated using community strings.

네덜란드 국립 스크러빙 센터(Dutch National Scrubbing Center)의 2021년 이벤트는 community string을 사용하여 완화되었습니다.



(a) A 2020 event at `B-root` defended using positive prepending.

(b) A 2021 event at `B-root` defended using negative prepending.

(c) A 2021 event at the Dutch National Scrubbing Center mitigated using community strings.

Figure 16: Different attacks with various responses (extended).

그림 16: 다양한 대응의 다양한 공격(확장)

부록 C: 상세한 공격 크기 추정

우리는 실험실 실험과 현실 세계 사건을 통해 우리의 공격 추정치를 검증합니다.

C.1 실험실 실험

우리는 모든 요소를 제어할 수 있는 실험실 (DETER [6])에서 실험을 통해 우리의 모델을 검증합니다. 여기서는 실제 제공된 부하를 추정하고 토폴로지를 고정시킵니다.

우리는 이 논문의 확장 버전 [57]에서 우리의 실험실 실험에 대한 세부 정보를 제공합니다.

C.2 사례 연구: 2016-06-25 이벤트

우리는 §4.1에서 현실 세계의 사례 연구를 보여주었습니다. 여기서는 2016-06-25 이벤트의 다른 사례에 대해 우리의 접근 방식이 작동함을 보여줍니다 (그림 12). 우리는 우리의 추정치(변동하는 주황색 선)가 보고된 선(파선 보라색 선)과 근접함을 관찰할 수 있습니다. 또한 우리의 관찰이 실제 제공된 부하의 작은 부분에 불과함을 알 수 있습니다 (하단 파란색 선).

이러한 결과 모두가 실험실과 실제 세계의 사건에서 우리의 접근 방식의 효과적임을 보여줍니다.

부록 D: BGP 독립성 정밀도

필터로 제한된 독립성 coverage로 인해 (§6.3.1), 우리는 다음으로 제어 범위를 살펴보겠습니다. 우리는 Tier-1 AS를 독립시킬 수 없으며, 소규모 AS는 거의 트래픽을 운반하지 않기 때문에 제한된 범위를 기대합니다.

우리는 Peering과 Tangled에서 세 개의 사이트를 사용하여 경로 독립을 테스트합니다. 예상대로, 우리는 어떤 Tier-1 AS를 독립하든지 Peering에서 AMS에서 30-35%의 부하가 발생하는 것을 관찰합니다 (그림 13). 우리는 Tangled에서도 비슷한 결과를 관찰합니다 (그림 14).

두 개 이상의 홉 떨어진 비-Tier-1 AS를 독립시킬 때, 우리는 트래픽 분배에 작은 변화를 관찰합니다. Peering에서는 AS57866의 독립이 AMS의 트래픽을 약간 줄입니다 (그림 13). 우리는 Tangled에서도 비슷한 결과를 관찰합니다 (그림 14).

우리의 결과는 Tier-1 AS를 독립시키는 것이 필터에 의해 제한되며, 여러 홉 떨어진 비-Tier-1 AS를 독립시키는 것은 트래픽의 작은 부분만 변경됨을 입증합니다.

즉시 상위 스트림을 독립시키는 것은 그들에게 발표하지 않는 것과 동일하므로, 여기서는 해당 사례를 고려하지 않습니다.

부록 E: 수용소 안정성

우리의 통찰력은 수용소가 시간이 지나도 안정적으로 유지됨을 이용합니다 (§7.3). 이를 테스트하기 위해 우리는 테스트 및 프로덕션

prefix와 함께 B-root 수용소 매핑을 한 달 동안 사용합니다. 우리는 B-root 수용소의 안정성을 관찰합니다.

그림 15에서 볼 수 있듯이, 수용소는 시간이 지나도 안정적으로 유지됩니다. 두 주 동안, 약 0.35%의 ㅇprefix가 변하고, 한 달 동안, 약 0.65%의 prefix가 수용소를 변경합니다. 이는 약 2백만 개의 prefix를 고려했을 때 한 달 후에도 수용소를 변경하는 것이 매우 작은 부분임을 보여줍니다. 따라서 매주/매월 한 번씩 Playbook을 작성하는 것이 충분합니다.

또한 하루 중 다양한 시간에 수용소 매핑을 수행했습니다. 우리는 수용소 분포가 하루 중에도 비슷하게 유지됨을 발견했습니다.

부록 F: 부하 분배

우리의 수용소 (§6.4) 분배를 사용하면 트래픽 분배의 적절한 예측을 얻을 수 있으며, 우리는 §8에서 성공적으로 적용합니다. 서비스는 부하에 관심이 있으므로, 다른 경로 변경에서 부하가 어떻게 분배되는지 확인하고 싶습니다.

운영자는 이미 계산된 수용소 매핑을 기반으로 부하 Playbook을 만들 수 있으므로 추가적인 BGP 발표가 필요하지 않습니다.

표 8에서는 서로 다른 시간대의 부하 분배 및 그 영향을 볼 수 있습니다. 낮은 GMT 시간대에는 AMS 사이트에서 부하가 줄어듭니다. 대부분의 유럽이 그 시간에 잠들기 때문입니다. BOS와 CNF는 00 GMT에 더 많은 부하를 받습니다. 이 두 지역에는 바쁜 시간이기 때문입니다. 각 prefix 뒤에있는 클라이언트 수의 차이로 인해 일부 prefix가 더 많은 부하를 제공합니다. 이러한 이유로 BOS prefix(주로 북미 prefix)는 다른 두 사이트의 prefix에 비해 부하가 적습니다. 또한 부하가 서로 다른 날의 같은 시간대에서 안정적으로 유지됨을 볼 수 있습니다 (대부분의 경우 5% 내외로 변동).

상대적인 수용소 분포가 부하 분배를 따르지만 완전히 동일하지는 않습니다. 운영자가 하루 중 다른 시간대에 다양한 부하 Playbook을 고려할 때 결정은 더욱 나아질 것입니다. 수용소 매핑이 안정적으로 유지되므로 여러 부하 Playbook을 작성하는 것이 간단합니다 (부록 E).

부록 G: 추가 공격 및 대응

우리는 B-root와 네덜란드 국립 스크러빙 센터에서 캡처한 더 많은 공격 사건을 평가합니다. 우리는 §8에서 언급한 동일한 방법론을 따릅니다. Peering에서 AMS, BOS 및 CNF 사이트 (§6)로 구성된 같은 Playbook을 사용합니다. 2020년 B-root의 부피 공격: 우리는 2020년 2월 14일 B-root에서 단일 쿼리 이름인 peacecorps.gov을 사용한 단기적인 부피 공격을 관찰했습니다. 이 이벤트는 매우 짧게 3분 동안 지속되었습니다. 실제로 BGP의 전파 지연으로 인해 이러한 단기 공격에 대해 어떤 경로 접근 방식도 작동할 수 없습니다. 우리는 공격이 더 오래 지속되는 경우의 영향을 보기 위해 유사한 트래픽 속도로 이벤트를 연장했습니다.

이 이벤트에서도 AMS는 가정된 용량이 40k 패킷/초인데 60k 패킷/초로 과부하가 발생합니다 (그림 16a). 우리는 AMS를 1만큼 접두합니다. 그 결과, 300초 후에는 AMS에서 과부하가 없는 것을 볼 수 있습니다. 이러한 부피 공격은 루트 서버에서 흔합니다. 경로 기반 접근 방식은 이러한 공격에 대해 방어할 수 있습니다.

2021년 B-root 이벤트에서 우리 시스템의 반복: 우리는 2021년 5월 28일에 B-root에서 발생한 다른 이벤트를 평가합니다. 이 이벤트에서 쿼리는 IP 단편화 (큰 패킷 크기)이었고, 일반적인 쿼리 이름은 pizzaseo.com이었습니다 (이 이벤트가 짧기 때문에 우리는 이를 연장했습니다). 공격이 시작되자, 우리의 시스템은 AMS 사이트가 과부하인 것을 발견합니다 (그림 16b). 우리의 시스템은 AMS에서의 접두가 접근 방식이 AMS에서의 트래픽을 줄이는 가장 좋은 방법이라고 결정합니다. 그러나 AMS를 1만큼 접두하면 CNF 사이트가 가장 많은 redirection 트래픽을받고 과부하가 발생합니다. 리디렉트된 공격 소스는 BOS보다 CNF를 선호합니다. 우리의 시스템이 CNF 사이트가 과부하임을 발견하면 이제 과부하가 발생했으므로 CNF에서 트래픽을 줄일 접근 방법을 배포합니다. 우리의 시스템은 BOS 사이트로의 트래픽을 더 많이 보내기 위해 음의 접두를 배포합니다. 900초 후에는 과부하가 없는 것을 볼 수 있습니다. 이 이벤트는 우리의 시스템이 최적의 경로 접근 방식을 점진적으로 찾아낼 수 있는 방법을 보여줍니다.

community string을 사용하여 방어하기: 다음은 2021년 8월 27일 네덜란드 국립 스크러빙 센터에서 관찰된 공격을 고려합니다. 이 공격은 부피형 DNS 증폭이었습니다. 이 공격에서 AMS는 과부하입니다. Playbook을 참고하여, 우리는 community string을 사용하여 트래픽을 이동시킵니다. AMS에서 6개의 IXP peer를 유지하면서 다른 모든 peer 및 transit을 삭제합니다. 이 변경의 영향은 300초 후에 그림 16c에서 볼 수 있습니다. 공격은 모든 사이트로 성공적으로 확산됩니다. 이 예는 다양한 community string이 트래픽 분배를 제어하는 방법을 보여줍니다. 이 논문의 확장 버전에서 더 많은 사건을 보여줍니다 [57].

References [1] AMPATH. Bgp resources. https://ampath.net/AMPATH_ BGP_Policies.php. [Online; accessed 12-Oct-2021]. [2] APNIC. BGP-stats routing table report—Japan view. https://mailman.apnic.net/mailing-lists/bgpstats/archive/2020/05/msg00001.html, May 1 2020. [3] Vaibhav Bajpai, Steffie Jacob Eravuchira, and Jürgen Schönwälder. Lessons learned from using the ripe atlas platform for measurement research. ACM SIGCOMM Computer Communication Review, 45(3):35–42, 2015. [4] Hitesh Ballani, Paul Francis, and Sylvia Ratnasamy. A measurement-based deployment proposal for IP anycast. In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pages 231–244, 2006. [5] Ran Ben-Basat, Gil Einziger, Roy Friedman, and Yaron Kassner. Heavy hitters in streams and sliding windows. In IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, pages 1–9, 2016. [6] Terry Benzel, Robert Braden, Dongho Kim, Cliford Neuman, Anthony Joseph, Keith Sklower, Ron Ostrenga, and Stephen Schwab. Experience with deter: a testbed for security research. In 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006., pages 10–pp. IEEE, 2006. [7] Leandro M. Bertholdo, João M. Ceron, Wouter B. de Vries, Ricardo de Oliveira Schmidt, Lisandro Zambenedetti Granville, Roland van Rijswijk-Deij, and Aiko Pras. Tangled: A cooperative anycast testbed. In 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pages 766–771, 2021. [8] Matthew Caesar and Jennifer Rexford. BGP routing policies in ISP networks. IEEE Network Magazine, 19(6):5–11, November 2005. [9] CAIDA. AS rank. https://asrank.caida.org/, 2020. [Online; accessed 12-Oct-2021]. [10] CAIDA. CAIDA UCSD BGP community dictionary. https: //www.caida.org/data/bgp-communities/, 2020. [Online; accessed 12-Oct-2021]. [11] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. Analyzing the performance of an anycast CDN. In Proceedings of the 2015 Internet Measurement Conference, pages 531–537, 2015. [12] Mark D Carney, Jeffrey A Jackson, Andrew L Bates, and Dante J Pacella. Method and apparatus for mitigating distributed denial of service attacks, November 24 2015. US Patent 9,197,666. [13] R. Chandra, P. Traina, and T. Li. BGP communities attribute. Technical Report 1997, RFC Editor, 1996. [14] Rocky KC Chang and Michael Lo. Inbound traffic engineering for multihomed ASs using AS path prepending. IEEE network, 19(2):18–25, 2005. [15] Yi-Ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-Bassett, and Ramesh Govindan. Are we one hop away from a better Internet? In Proceedings of the ACM Internet Measurement Conference, pages 523–529, Tokyo, Japan, October 2015. ACM. [16] Danilo Cicalese, Jordan Augé, Diana Joumblatt, Timur Friedman, and Dario Rossi. Characterizing ipv4 anycast adoption and deployment. In Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, pages 1–13, 2015. [17] Danilo Cicalese and Dario Rossi. A longitudinal study of

IP anycast. ACM SIGCOMM Computer Communication Review, 48(1):10–18, 2018. [18] Cloudflare. Famous DDoS attacks | the largest DDoS attacks of all time. https://www.cloudflare.com/learning/ddos/ famous-ddos-attacks/. [Online; accessed 12-Oct-2021]. [19] Alysha M De Livera, Rob J Hyndman, and Ralph D Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association, 106(496):1513–1527, 2011. [20] Wouter B. de Vries, Ricardo de O. Schmidt, Wes Hardaker, John Heidemann, Pieter-Tjerk de Boer, and Aiko Pras. Verfploeter: Broad and load-aware anycast mapping. In Proceedings of the ACM Internet Measurement Conference, London, UK, 2017. [21] Christoph Dietzel, Anja Feldmann, and Thomas King. Blackholing at IXPs: On the effectiveness of DDoS mitigation in the wild. In International Conference on Passive and Active Network Measurement, pages 319–332. Springer, 2016. [22] Ramin Ali Dousti, Frank Scalzo, and Suresh Bhogavilli. Automated ddos attack mitigation via bgp messaging, March 22 2018. US Patent App. 15/273,510. [23] Xun Fan and John Heidemann. Selecting representative ip addresses for internet topology studies. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, pages 411–423. ACM, 2010. [24] Xun Fan, John Heidemann, and Ramesh Govindan. Evaluating anycast in the domain name system. In 2013 Proceedings IEEE INFOCOM, pages 1681–1689. IEEE, 2013. [25] Seyed K Fayaz, Yoshiaki Tobioka, Vyas Sekar, and Michael Bailey. Bohatei: Flexible and elastic ddos defense. In 24th USENIX Security Symposium, pages 817–832, 2015. [26] Ashley Flavel, Pradeepkumar Mani, David Maltz, Nick Holt, Jie Liu, Yingying Chen, and Oleg Surmachev. Fastroute: A scalable load-aware anycast routing architecture for modern CDNs. In 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15), pages 381–394, 2015. [27] Ruomei Gao, Constantinos Dovrolis, and Ellen W Zegura. Interdomain ingress traffic engineering through optimized ASpath prepending. In International Conference on Research in Networking, pages 647–658. Springer, 2005. [28] Vasileios Giotsas, Georgios Smaragdakis, Christoph Dietzel, Philipp Richter, Anja Feldmann, and Arthur Berger. Inferring BGP blackholing activity in the internet. In Proceedings of the Internet Measurement Conference, pages 1–14. ACM, 2017. [29] T. Hardie. Distributing authoritative name servers via shared unicast addresses. Technical Report 3258, RFC Editor, 2002. [30] Lee Hahn Holloway, Srikanth N Rao, Matthew Browning Prince, Matthieu Philippe François Tourne, Ian Gerald Pye, Ray Raymond Bejjani, and Terry Paul Rodery Jr. Mitigating a denial-of-service attack in a cloud-based proxy service, October 7 2014. US Patent 8,856,924. [31] Chi-Yao Hong, Subhasree Mandal, Mohammad Al-Fares, Min Zhu, Richard Alimi, Kondapa Naidu B., Chandan Bhagat, Sourabh Jain, Jay Kaimal, Shiyu Liang, Kirill Mendelev, Steve Padgett, Faro Rabe, Saikat Ray, Malveeka Tewari, Matt Tierney, Monika Zahn, Jonathan Zolla, Joon Ong, and Amin Vahdat. B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in Google's software-defined WAN. In Proceedings of the ACM SIGCOMM Conference, Budapest, Hungary, August 2018. ACM. [32] Geoff Huston. BGP in 2017.

https://labs.apnic.net/ ?p=1102, Jan 8 2018. [Online; accessed 12-Oct-2021]. [33] Team Cymru Inc. Secure Cisco IOS BGP template. https:// www.team-cymru.com/secure-bgp-template.html. [Online; accessed 12-Oct-2021]. [34] Quan Jia, Huangxin Wang, Dan Fleck, Fei Li, Angelos Stavrou, and Walter Powell. Catch me if you can: A cloud-enabled ddos defense. In 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pages 264– 275. IEEE, 2014. [35] Brian Krebs. Krebsonsecurity hit with record DDoS. KrebsOnSecurity, Sept, 21, 2016. [36] Jan Harm Kuipers. Anycast for DDoS. https://essay. utwente.nl/73795/1/Kuipers_MA_EWI.pdf, 2017. [Online; accessed 12-Oct-2021]. [37] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet routing convergence. ACM SIGCOMM Computer Communication Review, 30(4):175–187, 2000. [38] Matt Levine, Barrett Lyon, and Todd Underwood. TCP anycast—don't believe the FUD. Presentation at NANOG 37, June 2006. [39] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Internet anycast: Performance, problems, & potential. In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, pages 59–73, 2018. [40] Ziqian Liu, Bradley Huffaker, Marina Fomenkov, Nevil Brownlee, et al. Two days in the life of the DNS anycast root servers. In International Conference on Passive and Active Network Measurement, pages 125–134. Springer, 2007. [41] Doug Madory and Matt Prosser. Excessive BGP AS path prepending is a self-inflicted vulnerability. Presentation at RIPE 79, October 2019. [42] Marek Majkowski. Memcrashed - major amplification attacks from UDP port 11211. https://blog.cloudflare. com/memcrashed-major-amplification-attacks-fromport-11211/, 2018. [Online; accessed 12-Oct-2021]. [43] Tyler McDaniel, Jared M Smith, and Max Schuchard. Flexsealing bgp against route leaks: peerlock active measurement and analysis. arXiv e-prints, pages arXiv–2006, 2020. [44] Stephen McQuistin, Sree Priyanka Uppu, and Marcel Flores. Taming anycast in the wild Internet. In Proceedings of the Internet Measurement Conference, pages 165–178, 2019. [45] Giovane C. M. Moura, Ricardo de O. Schmidt, John Heidemann, Wouter B. de Vries, Moritz Müller, Lan Wei, and Christian Hesselman. Anycast vs DDoS: Evaluating the November 2015 root DNS event. In Proceedings of the ACM Internet Measurement Conference, November 2016. [46] Priyadarsi Nanda and AJ Simmonds. A scalable architecture supporting QoS guarantees using traffic engineering and policy based routing in the Internet. International Journal of Communications, Network and System Sciences, 2009. [47] Root Server Operators. Events of 2015-11-30. https://rootservers.org/media/news/events-of-20151130.txt, 2015. [Online; accessed 12-Oct-2021]. [48] Root Server Operators. Events of 2016-06-25. https://rootservers.org/media/news/events-of-20160625.txt, 2016. [Online; accessed 12-Oct-2021]. [49] Craig Partridge, Trevor Mendez, and Walter Milliken. Host anycasting service. Technical Report 1546, RFC Editor, 1993. [50] The Canadian Press. Canadian communications company voip.ms hit by cyber attack. https://www.thestar.com/ business/2021/09/21/canadian-communicationscompany-voipms-hit-by-cyber-attack.html/, 09 2021. [51] LANDER project.

Lander:b root anomaly-20170306. https://ant.isi.edu/datasets/readmes/B_Root_ Anomaly-20170306.README.txt, 2019. [Online; accessed 12-Oct-2021]. [52] Bruno Quoitin, Cristel Pelsser, Olivier Bonaventure, and Steve Uhlig. A performance evaluation of BGP-based traffic engineering. International journal of network management, 15(3):177–191, 2005. [53] Bruno Quoitin, Cristel Pelsser, Louis Swinnen, Olivier Bonaventure, and Steve Uhlig. Interdomain traffic engineering with BGP. IEEE Communications magazine, 41(5):122–128, 2003. [54] RIPE. Measurements. https://atlas.ripe.net/ measurements/10310/. [Online; accessed 12-Oct-2021]. [55] RIPE. Root dns observations. Measurement ID 1009 (A-Root), 1010 (B-Root), etc., 2021. [56] RIPE Network Coordination Centre. RIPE - Routing Information Service (RIS). https://https://www. ripe.net/analyse/internet-measurements/routinginformation-service-ris, 2020. [57] ASM Rizvi, Joao Ceron, Leandro Bertholdo, and John Heidemann. Anycast agility: Adaptive routing to manage ddos. arXiv preprint arXiv:2006.14058, 2020. [58] ASM Rizvi, John Heidemann, and Jelena Mirkovic. Dynamically selecting defenses to DDoS for DNS (extended). Technical Report ISI-TR-736, USC/Information Sciences Institute, May 2019. [59] Sandeep Sarat, Vasileios Pappas, and Andreas Terzis. On the use of anycast in DNS. In Proceedings of 15th International Conference on Computer Communications and Networks, pages 71–78. IEEE, 2006. [60] Brandon Schlinker, Todd Arnold, Italo Cunha, and Ethan KatzBassett. PEERING: Virtualizing BGP at the Edge for Research. In Proc. ACM CoNEXT, Orlando, FL, December 2019. [61] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan KatzBassett, Harsha V. Madhyastha, Italo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. Engineering egress with Edge Fabric: Steering oceans of content to the world. In Proceedings of the ACM SIGCOMM Conference, pages 418–431, Los Angeles, CA, USA, August 2017. ACM. [62] Ricardo de O. Schmidt, John Heidemann, and Jan Harm Kuipers. Anycast latency: How many sites are enough? In International Conference on Passive and Active Network Measurement, pages 188–200, Sydney, Australia, March 2017. [63] Thomas Bradley Scholl. Methods and apparatus for distributed backbone internet ddos mitigation via transit providers, February 3 2015. US Patent 8,949,459. [64] A. Shaikh, R. Tewari, and M. Agrawal. On the effectiveness of DNS-based server selection. In Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), volume 3, pages 1801–1810 vol.3, 2001. [65] AX Sharma. Phone calls disrupted by ongoing ddos cyber attack on voip.ms. https://arstechnica.com/ gadgets/2021/09/canadian-voip-provider-hit-byddos-attack-phone-calls-disrupted/, 09 2021. [66] AWS Shield. Aws shield - threat landscape report – q1 2020. https://aws-shield-tlr.s3.amazonaws.com/ 2020-Q1_AWS_Shield_TLR.pdf, 08 2020. [67] R. B. da Silva and E. Souza Mota. A survey on approaches to reduce BGP interdomain routing convergence delay on the Internet. IEEE Communications Surveys & Tutorials, 19(4):2949– 2984, 2017. [68] Daniel Smith. The growth of DDoS-as-a-service: Stresser services. https://blog.radware.com/security/2017/09/

growth-of-ddos-as-a-service-stresser-services/, 2017. [Online; accessed 12-Oct-2021]. [69] Donald J Smith, Michael Glenn, John A Schiel, and Christopher L Garner. Network traffic data scrubbing with services offered via anycasted addresses, May 24 2016. US Patent 9,350,706. [70] Jared M. Smith and Max Schuchard. Routing around congestion: Defeating DDoS attacks and adverse network conditions via reactive BGP routing. In 2018 IEEE Symposium on Security and Privacy (SP), pages 599–617. IEEE, 2018. [71] Job Snijders. Practical everyday bgp filtering with as_path filters: Peer locking. NANOG-67, Chicago, June, 2016. [72] Oliver Spatscheck, Zakaria Al-Qudah, Seunjoon Lee, Michael Rabinovich, and Jacobus Van Der Merwe. Multi-autonomous system anycast content delivery network, December 10 2013. US Patent 8,607,014. [73] RIPE NCC Staff. Ripe atlas: A global internet measurement network. Internet Protocol Journal, 18(3), 2015. [74] One Step. BGP community guides. https://onestep.net/ communities/. [Online; accessed 12-Oct-2021]. [75] Eric Sven-Johan Swildens, Zaide Liu, and Richard David Day. Global traffic management system using IP anycast routing and dynamic load-balancing, March 8 2011. US Patent 7,904,541. [76] Renata Teixeira, Steve Uhlig, and Christophe Diot. BGP route propagation between neighboring domains. In International Conference on Passive and Active Network Measurement, pages 11–21. Springer, 2007. [77] University of Oregon. Route Views Project. http://www. routeviews.org/routeviews/, 2021. [78] USC/ISI. Usc/isi ant datasets. https://ant.isi.edu/ datasets/all.html, 2019. [Online; accessed 12-Oct-2021]. [79] Lan Wei and John Heidemann. Does anycast hang up on you? In 2017 Network Traffic Measurement and Analysis Conference (TMA), pages 1–9, Dublin, Ireland, July 2017. IEEE. [80] Fernanda Weiden and Peter Frost. Anycast as a load balancing feature. In Proceedings of the 24th international conference on Large installation system administration, pages 1–6. USENIX Association, 2010. [81] Curt Wilson. Attack of the Shuriken: Many hands, many weapons. https://www.arbornetworks.com/blog/ asert/ddos-tools/, 2012. [Online; accessed 12-Oct-2021].